Systematic Modelling of Anonymity with Application to Cryptocurrencies

Niluka Samanthi Amarasinghe

M.Eng.Mgmt. (Software Engineering) & M.Sc. (Computer Science)

B.Eng.(Hons)(Electrical Engineering)

Submitted in fulfilment of the requirements

for the Degree of

Doctor of Philosophy

Queensland University of Technology

School of Computer Science

Faculty of Science

2022

Keywords

Anonymity, Privacy, Cryptocurrencies, Indistinguishability, Unlinkability, Fully decentralised systems, Game-based security modelling

Abstract

The modern financial world has seen a significant rise in the use of cryptocurrencies in recent years, due to inherent convincing characteristics such as decentralised nature and convenience, and more importantly, perceived privacy and anonymity features.

Despite being considered as the most widespread among all, Bitcoin is claimed to have significant lapses in relation to its anonymity. Many studies have shown that a majority of 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. With the emergence of such new cryptocurrencies, many have attempted to evaluate such claims. These efforts have resulted in various interpretations of anonymity, which are often restricted to a particular currency scheme only. The absence of a common formalised metric for evaluating anonymity has led to much confusion over their claims, making it infeasible to properly compare different systems. More importantly, anonymity in such complex multi-party systems as finance, turns out to be a surprisingly multifaceted notion which needs to be defined and modelled with precision.

In this work, we introduce a common framework, which can be used to evaluate the nature and extent of anonymity in (crypto)currencies and similar distributed transaction systems, irrespective of their implementation. For this purpose, we construct a theoretical model to represent the generic functionality of cryptocurrency schemes across different implementations, by establishing a cryptographically sound and secure foundation. We then develop a comprehensive adversarial model in order to capture different aspects of anonymity around system entities.

Building upon this foundation, we formulate a common template, which is capable of modelling a multitude of different attacker scenarios with respect to various anonymity considerations. With an aim to strengthen the usability of this framework, we provide formal definitions for anonymity notions pertaining to various scenarios. In addition, we investigate the relationships among those definitions and formulate a set of theorems indicating the implications, dependencies and separations among them. Accordingly, this framework, together with the formal definitions and theorems, provides a means for modelling anonymity uniformly across different constructions. As such, the fine-grained systematisation of anonymity resulting from this work highlights the importance of precise definitions for modelling anonymity, which is a surprisingly nuanced concept.

Contents

List of Tables

List of Figures

List of Publications

1. N. Amarasinghe, X. Boyen and M. McKague: A Survey of Anonymity of Cryptocurrencies, ACSW 2019: Proceedings of the Australasian Computer Science Week Multiconference, ACSW 2019, pp. 2:1-2:10. ACM (2019) (Contents included in Chapter 2)

2. N. Amarasinghe, X. Boyen and M. McKague: The Complex Shape of Anonymity in Cryptocurrencies: Case Studies from a Systematic Approach, in International Conference on Financial Cryptography and Data Security, pp. 205-225, Springer, 2021. (Contents included in Chapter 5)

3. N. Amarasinghe, X. Boyen and M. McKague: The cryptographic Complexity of Anonymous Coins: A systematic Exploration, Cryptography, vol.5, no. 1, 2021 (Contents included in Chapters 3 & 4)

List of Abbreviations

BOLT	Blind Off-chain Light-weight Transactions
BTC	Bitcoin
CA	Central Authority
CCA	Chosen Ciphertext Attack
CPA	Chosen Plaintext Attack
CT	Confidential Transactions
DAP	Decentralised Anonymous Payment
DApps	Distributed Applications
ECDSA	Elliptic Curve Digital Signature Algorithm
IND	Indistinguishability
IND-CCA	Indistinguishability under Chosen Ciphertext Attack
IND-CPA	Indistinguishability under Chosen Plaintext Attack
IP	Internet Protocol
ISP	Internet Service Provider
KNOWL	Knowledge
MAC	Message Authentication Code
P2P	Peer-to-Peer
PCN	Payment Channel Network
PPT	Probabilistic Polynomial Time

PRNG	Pseudo Random Number Generator
PoW	Proof of Work
PWR	Power
Tor	The Onion Router
TTP	Trusted Third Party
UC	Universal Composability
ULK	Unlinkability
zk-SNARK	zero-knowledge Succinct Non-interactive Arguments of Knowledge

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Name: Niluka Samanthi Amarasinghe

Signature: QUT Verified Signature

Date: 13/05/2022

This thesis is dedicated to my loving family.

Acknowledgments

Firstly, I would like to extend my deepest gratitude to my supervisory team; Principal Supervisor, Associate Professor Xavier Boyen and Associate Supervisor, Dr. Matthew McKague for their enduring support and guidance provided throughout this journey. If it were not for their expertise, encouragement and patience, this work would not have been possible. I am truly indebted to them for that.

I am sincerely grateful to QUT, for providing me with financial support through the Research Training Program (RTP) scholarship and also for the sponsorship of other finances. Thank you to Xavier, Matt and Ernest for providing me with the opportunities to teach a wide range of courses which helped me immensely to keep my knowledge up to date. I would also like to take this opportunity to thank many others whom I associated in QUT; Leonie, Vicky, Zarah and my colleagues; Udyani, Chathurika, Sriparen, Ali, Mukhtar, Jayamine, Waleed, David, Luke and Ray.

None of this would have been possible if it was not for my loving husband, Damith, who encouraged me to pursue this journey after a long break from studies. Damith, I cannot thank you enough for always believing in me and my abilities to accomplish this goal, for always making time to listen to my grievances, for encouraging me to stay positive, and for being there for me when

I needed support. I wish to extend special thanks to my daughter Samudi and my son Gevindu, for all the support given to me during the past 4 years and also for bearing with me during tough times. I would also like to extend many thanks to my mother, sisters and brother for motivating me to get through this journey. Special thanks to my friends here in Australia and in Sri Lanka for their support and encouragement. Last but not least, I would like to acknowledge my former employer, Central Bank of Sri Lanka, for all the support provided to me throughout my tenure there.

Chapter 1

Introduction

Together with the advancements in technology, the financial sector has embraced many new developments over the years. As a result, ways of conducting monetary transactions have also changed considerably in the recent past. In that connection, various forms of electronic currency schemes have emerged as alternatives to traditional physical currencies and cryptocurrencies hold a major stake among them. Factors such as decentralised nature, flexibility, convenience and more importantly promises of privacy and anonymity, have contributed to the popularity of such cryptocurrencies. In this thesis, we aim to investigate the anonymity considerations in the context of cryptocurrencies.

Anonymity can be considered as a demanding factor, that influences the adoption of cryptocurrency systems by users as naturally one would not want their transaction details to be exposed. Bitcoin, the first and the most widely used cryptocurrency, has attracted much attention with respect to its privacy and anonymity characteristics. It is claimed that Bitcoin has only achieved a level of pseudonymity as the transaction graph is totally public, revealing the payment addresses corresponding to transactions and linking with real world identities eventually [77]. In response to these concerns, many other alternatives have been developed aiming to improve anonymity of Bitcoin or to invent new currencies altogether, thus creating an ongoing battle to meet this demand. Over and above that, many different interpretations of anonymity have loomed, causing much confusion in the community.

Undoubtedly, a satisfactory level of anonymity is pivotal for the sustainability of any cryptocurrency. Hence it is essential that anonymity claims of these currencies be precise, reliable and be comparable across different currency schemes. This demands for a systematic approach to represent anonymity in an apprehensible manner, which facilitates formalised and explicit definitions that can capture different attributes of anonymity and the relationship between various aspects of these definitions. Moreover, such systematisation can be a valuable tool for the designers of future virtual currency schemes in improving anonymity aspects in such constructions. From this perspective, it is vital to have a standardised means for defining what anonymity means in the context of cryptocurrencies.

1.1 Motivation

From a broader perspective, privacy and anonymity can be regarded as two different concepts, yet they often complement each other in some contexts such as cryptographic systems. Privacy, in general, is interpreted as hiding what was performed in a certain action and conversely, anonymity is defined as hiding who (i.e. identity) performed the action [90]. In cryptographic security, both notions are equally important. There also exists a third related notion, steganography, which is to hide that an action was even performed at all. Though it closely relates to privacy and even more so to anonymity, it is perhaps technically less relevant in a networked environment where the very existence of communications taking place is difficult to conceal.

The absence of an acceptable level of anonymity and privacy could hinder the effectiveness of any currency scheme. Many traditional currency schemes are centralised systems where customers depend on another party to preserve the privacy of related information. For example, in a traditional banking model, banks are bound by regulations to preserve the confidentiality of customer information. Cryptocurrency schemes achieve the same through the use of various cryptographic techniques in a decentralised environment.

As a matter of fact, if the transaction history of a particular individual or entity were exposed to an outsider, it could result in many undesirable consequences, from a subjective sense of betrayal, to more concrete abuses such as misuse of that information to gain undue advantages in contract bidding. Even worse, if currency units came attached with transaction histories, that could lead to the blacklisting of specific units based on their use in unlawful activities in the past, or their involvement in boycotted operations, even though the units may have had only uncontroversial uses afterwards. In such a situation, any punishment by blacklisting affects only the current, honest holders of those units rather than the original criminals who have long since received the benefit of those units. As such, it is paramount to have a tolerable level of anonymity in a currency scheme in order to ensure its *fungibility*. By fungibility, we mean that every currency unit in a currency system is replaceable with any other unit within the system.

Bitcoin, despite having the highest market capitalisation at the time of this writing [25, 54], has received much criticism in relation to its anonymity and it has been argued that the current Bitcoin framework only provides 'pseudonymity', in place of anonymity, since transactions are linked to payment addresses in a big graph that is visible to all [24, 44]. Hence payment addresses form a type of identity, and these can in some circumstances be associated with real-world identities. Detailed analyses of public bitcoin transaction data have shown that it is possible to uncover behaviour patterns of Bitcoin users and trace their identities in real life [7, 63, 78].

Ethereum, which emerged subsequently, has become a popular platform for many distributed applications including cryptocurrencies. Ethereum appears to be even worse than Bitcoin with respect to its anonymity, as it compels its users to reuse the same addresses, or accounts, across all transactions, unlike Bitcoin which at least makes an attempt to move funds to new addresses every time a payment is being made [18, 23, 100].

As a consequence of this tension between the need for, and the lack of, effective anonymity in cryptocurrencies, there has been much research 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 fundamentals in the design of new cryptocurrency schemes (e.g. Monero). In spite of many such solutions making claims of "anonymity", some studies have shown that the majority of them could still be subject to deanonymisation under different circumstances [60, 64]. As a result, these claims raise uncertainty around anonymity and leave many questions unanswered, e.g. How likely is the deanonymisation to succeed, what are the factors contributing towards it, what is the extent of deanonymisation etc. Such concerns highlight the importance of assessing anonymity achieved by these currencies in a precise and unambiguous manner. This motivates our main research question;

1. *How can we evaluate and compare anonymity aspects of cryptocurrency schemes in a unified and systematic manner?*

Alongside this ongoing battle to improve anonymity, much energy has been spent in both academia and in the industry to develop means to evaluate the extent of anonymity achieved by those currencies. This has resulted in a multitude of analyses and evaluations in relation to the anonymity of Bitcoin as well as other cryptocurrencies (as presented in Chapter 2). While some of them follow theoretical approaches, many focus on experimental analyses to show the possibilities for deanonymisation of transactions against their original claims. Although many discuss the diversity of anonymity attributes (or implied attributes), they do not provide clear formal definitions [58, 77, 78]. Many such notions are coarse grained and do not capture the full subtlety of a given situation. Even with the very few that attempt to define different anonymity notions, they have been borrowed from elsewhere (e.g. from anonymous communication) and thus leads to multiple different interpretations [24, 44].

Further, relationships among the anonymity notions used in the existing literature are often unclear, which adds to the complexity. Some have attempted to categorise and compare different schemes based on various models, yet their work only provides a high level picture as there is no clear-cut model available to provide a common (i.e. unified) mechanism to represent anonymity at a much granular level. Therefore, the absence of a common framework for modelling anonymity of cryptocurrencies has led to doubts in the credibility of anonymity claims made by existing cryptocurrencies and thus it is not possible to evaluate the extent of anonymity achieved by different currency schemes in a meaningful manner [46, 75]. This leads to the research question given below.

2. *How can we design a common framework to model anonymity aspects of different cryptocurrency schemes in terms of formal anonymity definitions?*

In order to develop a common framework which can be used to model anonymity of cryptocurrencies irrespective of the underlying construction, it is necessary to create a means to model the functionality across different currency schemes. Such model should be secure and correct in terms of functionality, and serves as the first stepping-stone in this study. Accordingly, we formulate the following research question.

3. *How can we develop a means to model the functionality of cryptocurrency schemes commonly across different constructions?*

Therein, we compose our research objectives of this study based on the above research questions in the next section.

1.2 Research objectives and contributions

The motivating objective of this research study is to formulate a strategy to model anonymity in distributed environments such as cryptocurrency schemes in a uniform manner, enabling one to obtain reasonable comparisons across different constructions. In addition, we also aim to investigate the avenues for a simplified approach to understand the security modelling of cryptographic protocols better. In order to achieve our research aims, we pursue a phased approach throughout by setting individual objectives in each phase and we summarise these research objectives and corresponding contributions below.

Objective 1 : Develop a means to commonly represent the functionality of different cryptocurrency schemes.
In order to formulate a unified framework across diverse constructions, there needs to be a common platform which can be modelled as a prototype for such systems. Hence, keeping in line with research question 3, our initial goal is to develop a common means to represent different currency schemes. We consider the typical functionality of a fully decentralised currency scheme for this purpose, thereby defining relevant operations, their correctness and security requirements to ensure the viability and the integrity of the proposed system. We also identify the factors affecting the privacy and anonymity of entities within a cryptocurrency scheme and then aim to develop a comprehensive adversarial model to capture the same.

Contribution 1: Accordingly, our first contribution in this study is a theoretical model that is functionally sound and secure, and can be utilised to model generic functionality, security and correctness of a cryptocurrency scheme, irrespective of the implementation details. We have included this content in our publication [5] and we present a comprehensive description in Chapter 3.

Objective 2 : Design a common framework to model anonymity aspects of different cryptocurrency schemes in terms of formal anonymity definitions
As captured in research question 2, our goal here is to develop a common, parameterised definition of anonymity, which is capable of capturing a wide range of anonymity scenarios based on the adversarial knowledge, adversarial capabilities and the targeted security goal. This framework can further support formalising the anonymity attributes in a comprehensible manner through formalising definitions and relationships among them. Such comprehension enables fine-grained modelling of anonymity, which provides an in-depth understanding of many different aspects of anonymity and the effects of their interdependencies. Further, this also facilitates effective comparisons across various constructions.

Contribution 2: In this connection, our research contributions include a unified framework that can capture a multitude of anonymity notions in the context of cryptocurrencies. We further provide a set of formal anonymity definitions, which can be used to represent different characteristics in currency schemes and compare the same across different schemes. In addition, we also present a set of theorems in order to demonstrate the relationships and interdependencies of anonymity notions resulting from the proposed framework. We have published this material in [5] and we provide further details in Chapter 4.

Objective 3 : Evaluate and compare anonymity aspects of existing cryptocurrency schemes

The ultimate research objective in this work as represented by the research question 1 is to evaluate and compare anonymity aspects of different existing currency schemes in a unified manner. In this respect, we aim to demonstrate the applicability of our framework by analysing a sub set of popular existing cryptocurrencies, which represent a wide range of functionalities, including some which claim strong anonymity guarantees. This allows us to model anonymity across different constructions in a common metric and this enables us to compare them effectively.

Contribution 3: In this contribution, we present the analysis outcomes of several existing cryptocurrency schemes having very different constructions, namely Bitcoin, Zcash, Monero and Mimblewimble, which are compared against a fictitious currency scheme exhibiting very strong anonymity characteristics. Therein, we present a unified view of anonymity landscape of chosen cryptocurrencies. This work is published in [4] and Chapter 5 presents the details of this exercise.

1.3 Methodology

Our research methodology is focused around constructing a common unifying mathematical framework, which is capable of capturing all the multiple security nuances in all existing and future currency schemes. We follow the accepted approach in cryptography, which is to define security properties in terms of "games" or conceptual experiments. For most of our results, we choose game-based definitions over the Universal Composability (UC) framework because UC is a very nice theoretical methodology which is best suited for confined systems whose ideal functionalities may still have a clean description, which is certainly not the case in the context of cryptocurrencies.

1.3.1 Scope of the study

Within the course of this study, we do not intend to consider the particulars of the underlying communication network or the consensus mechanism 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 as intended. However, the communication layer might independently affect the achievable level of anonymity in a particular system, in which case it needs to be addressed separately. We propose some possible workarounds in this regard while reflecting upon the viability of the findings of this work.

1.4 Thesis outline

In this chapter, we presented a synopsis of our research study, while emphasising the motivation behind this work and the significance of resulting research outcomes. To that effect, the chapter unfolds the background to our research problem, leading to the objectives and the methodology. Subsequent chapters of this thesis are organised in the following manner.

Chapter 2: Literature review and background

This chapter summarises the literature relevant to this study and establishes the preliminaries required for the research. In this respect, related literature are presented under theoretical background of anonymity, anonymity considerations of Bitcoin and other cryptocurrencies, and survey studies on anonymity, while highlighting the research gaps and drawing comparisons to our work.

Additionally, we describe other preliminaries required for this study including the notations used.

Chapter 3: An abstract cryptocurrency scheme

Chapter 3 sets forth the foundation of this study by presenting the fundamentals underlying the framework for a generic cryptocurrency scheme. This includes a formal definition for a cryptocurrency scheme, depicting the typical functionality in terms of a set of operational algorithms. Further, the functional correctness requirements of the scheme are also established in this chapter. The chapter also outlines a comprehensive adversarial model in relation to the security requirements of honest functionality of the scheme.

Chapter 4: Anonymity framework

This chapter presents the proposed anonymity framework and resulting anonymity notions. In this connection, a a conceptual experiment is formulated based on the adversarial model developed in the previous chapter. In addition, formal definitions are provided in relation to the anonymity notions corresponding to some chosen attacker scenarios. Further insights are provided into the anonymity definitions and their interdependencies through a set of theorems and corresponding proofs.

Chapter 5: Analysis of anonymity: Case studies

The applicability of the proposed framework is demonstrated in this chapter through a set of case studies. Accordingly, this chapter provides a detailed analysis of anonymity achieved by several existing cryptocurrency schemes including Bitcoin versus Zcash, Monero and Mimblewimble which have become popular for their claims for improved anonymity. A fictitious Trusted Third Party scheme is also studied as a benchmark reference. Additionally, comparisons are drawn wherever appropriate among the schemes.

Chapter 6: Conclusion

This chapter recaps the research outcomes of this study presented in preceding chapters and while emphasising the significance of this work in relation to the systematisation of anonymity considerations of fully decentralised systems in general. To that effect, strengths and limitations of the proposed framework are discussed and recommendations for improvements and future research directions are also identified.

Chapter 2

Literature Review and Background

This chapter establishes the groundwork for our research study. We start by providing an overview of existing theoretical representations of anonymity. Next, we discuss the privacy considerations of cryptocurrencies with an emphasis on Bitcoin, followed by other currency schemes that claim to possess stronger privacy guarantees. Our focus here is to analyse the theoretical foundations used for modelling such anonymity characteristics in cryptocurrencies.

In addition, this chapter also includes other background information including the preliminaries of relevant constructs. To that effect, we present brief descriptions of Game-Based and Universal Composability (UC) framework based security modelling.

2.1 Theoretical representations of anonymity

The notions of privacy and anonymity have been discussed widely in the literature for several decades, mostly with respect to information systems and data communication. While these concepts can be interpreted differently in various contexts, we can distinguish between the two informally as follows. Privacy technically refers to hiding the contents associated with a given outcome, whereas anonymity is linked with identities and thus refers to the hiding of identities of entities involved in an outcome. Although it is difficult to separate the two in a precise manner, both concepts complement each other in many contexts. For example, preserving privacy is important when storing privacy-concerned information such as health data. It is equally important to preserve anonymity of participants when communication takes place over a shared medium (e.g. anonymous donations). Hence, many theoretical frameworks have been developed to model privacy and anonymity in these contexts. In this section, we briefly discuss several frameworks which have been used for this purpose in information systems and anonymous communication.

2.1.1 k -Anonymity

k-Anonymity can be regarded as one of the first notions proposed in the literature, which has been modelled around the idea that the ambiguity of identifiability of an item is greater as the number of possible candidates becomes higher [89]. From a high-level perspective, an item in a set of k items within a system is said to be k-anonymous if that item cannot be differentiated from the remaining $k-1$ items in that set (aka anonymity set). One of the most natural uses of this notion of k-anonymity is in relation with the cryptographic construct of ring signature, where the actual signer is anonymous among the participants of the ring. Sweeney et al. [89] further presented a formal mathematical model to define k-anonymity in the context of information hiding/minimising. For this purpose, an identifier (termed as a quasi-identifier) has been defined as a subset of attributes of a data item, which can uniquely identify a particular data item when combined together. Then the formal definition of k-anonymity follows such that, given a set of data records, the set is said to be k-anonymous only if all available combinations of values of the quasi-identifier set appears at least k times in the given data set.

The theory of k-anonymity has been studied substantially in subsequent literature, and several improvements have been made [62, 92]. There are other studies which propose extensions to overcome the issues of the effectiveness of this model in the presence of an adversary with background knowledge and also when there is only a minute difference between sensitive attributes. For example, l-diversity proposed in [52], requires that sensitive attributes have at least l distinct values so that the said issues of k-anonymity can be overcome. Another study proposes the notion of t-closeness, where the distribution of a sensitive attribute in any indistinguishable set and the distribution of the overall data set should have a distance threshold of t [48]. Both these notions are used extensively in the context of anonymous data disclosure, specifically in data mining to provide a precise notion of anonymity [76, 83].

The concept of k-anonymity has also been referenced in modelling anonymity in cryptocurrencies such as Bitcoin [44, 71]. In [71] for example, the anonymity set consists of the number of active entities in the Bitcoin network at a given time or within a short time interval around the given time instance, thus providing a quantitative measure of anonymity.

2.1.2 A modal logic approach

Tsukada et al. [90] presented a framework for privacy related to information hiding and disclosure, using a modal logic approach. The authors claim that there is a logical structure underlying privacy and anonymity properties. The taxonomy in that work is developed around four properties namely; privacy, anonymity, 'onymity' and identity. In the simplest form, anonymity refers to the hiding of the doer of a particular action whereas privacy refers to the hiding of what that particular action was. On the other hand, onymity refers to the disclosure of the doer of a particular action while identity discloses what the action was.

2.1.3 Anonymity terminology

Several studies have been undertaken to formalise different terminologies related to privacy and anonymity. Pfitzmann et al. [72] proposed a terminology that consists of attributes such as anonymity, unlinkability, undetectability and unobservability, with respect to a set of subjects and objects, which correspond to senders, recipients and messages in a communication network. These terms are defined as follows:

i. Anonymity - Defined with respect to a subject where that subject cannot be identified by an attacker (external to the specific environment) among a set of subjects. The set of subjects are termed as the anonymity set.
ii. Unlinkability - An attacker cannot distinguish whether two or more subjects in a given set are related.
iii. Undetectability - Inability of identifying the existence itself of an item (a subject or an object).
iv. Unobservability - Undetectability of a particular event with respect to all subjects that are not involved in the said event and anonymity with respect to those subjects that are involved in the same event.

Furthermore, these properties are then drilled down further to address senders and recipients separately, e.g. sender anonymity and recipient anonymity etc. This analysis helps identify the relationships among above terms while deducing weaker properties.

2.1.4 Hierarchical frameworks

Bohli et al. [15] attempted to categorise various anonymity definitions in a hierarchical manner and proposed a formal framework for anonymity notions giving consideration to application specific characteristics. Their work presents a privacy model in terms of a set of participants, set of elements (comparable to messages), a usage frequency set indicating how many elements correspond to each participant and a set denoting relationships between elements and corresponding participants. Anonymity notions defined in this model are given below:

i. Strong anonymity - Adversary learns nothing about how elements and participants are linked.
ii. Strong unlinkability with participation hiding - Adversary learns nothing about how elements and participants are linked, except the number of participants.
iii. Strong unlinkability with usage hiding - Adversary knows only about the set of participants.
iv. Weak unlinkability with participation hiding - Adversary knows only about the size of the set of participants and the size of the relationship set.
v. Weak unlinkability with usage hiding - Adversary learns nothing about the frequency set and the relationship set except their sizes.
vi. Weak unlinkability - Adversary knows nothing other than the usage frequency set.
vii. Pseudonymity - Adversary knows only the relationship set.
viii. Anonymity - Adversary learns only the set of participants and the set of messages, but not the links between the two sets.
ix. Weak anonymity - Similar to anonymity but adversary learns how many participants corresponds to how many messages.

Those authors analyse the above notions further and present a hierarchical relationship among them, which helps to identify their dependencies. This framework can be used to construct application specific anonymity notions to model anonymity effectively [15].

2.1.5 Indistinguishability based anonymity notions

Many studies in relation to anonymity are centered around anonymous communication. One such notable framework is presented in [38], which constructs a set of anonymity definitions based around the idea of indistinguishability. These notions are closely analogous to those defined in [72], yet present additional notions in terms of unlinkability, which represents rather weak anonymity characteristics. These definitions consist of the following, and are constructed based on the number and values of messages sent and received by the parties in a system.

i. Sender unlinkability - Hides any relationship between senders and recipients other than what can be determined on total messages sent and special values received.
ii. Receiver unlinkability - Hides any relationship between senders and recipients other than what can be determined on special values sent and total messages received.
iii. Unlinkability - Hides any relationship between senders and recipients based on special values sent and received
iv. Sender anonymity - Hides the number of messages and values sent.
v. Receiver anonymity - Hides the number of messages and values received.
vi. Strong sender anonymity - Stronger than sender anonymity, allows leakage of information on the amount of traffic per receiver at maximum.
vii. Strong receiver anonymity - Stronger than sender anonymity, allows leakage of information on the amount of traffic per sender at maximum.

viii. Sender-Receiver anonymity - Combines above two.
ix. Unobservability - Hides the total number of messages sent/received.

Further, their work also discusses relationships among these notions where it is proven that sender anonymity implies sender unlinkability and the same holds for recipient anonymity. In addition, unobservability is the strongest in the sense that it implies all other anonymity notions. The authors have successfully evaluated several anonymous communication protocols using this framework, with provably secure claims for different aspects of anonymity.

2.1.6 AnoA framework for analysing anonymous communication protocols

Another comprehensive theoretical framework for analysing anonymous communication is the AnoA framework presented by Backes et al. [10], which aims to analyse anonymity aspects in a standardised form. This representation is based on a generalisation of the notion of differential privacy, which means that an attacker is unable to gain non negligible amount of information about an individual data record in a collection of data, despite having background knowledge of some aggregated information on the data set [101]. To that effect, this work formalises several anonymity properties based on indistinguishability coupled with an additive factor and a multiplicative factor representing the winning probability of the adversary and the impact. i.e.

i. Sender anonymity - Hides the sender.
ii. Recipient anonymity - Hides the recipient.
iii. Sender unlinkability - Inability to ensure whether any two given actions were done by the same user.
iv. Relationship anonymity - Hides the sender and recipient such that it is not possible to identify both the sender and the recipient simultaneously.

A noteworthy strength in this work compared to previous studies is that the above definitions are capable of modelling in simulation based composability frameworks such as the UC framework. The analysis of The onion routing (Tor) network demonstrates how this framework can quantify the extent of anonymity perceived by an anonymous communication channel.

2.1.7 Provably secure anonymity

Gelernter et al. [33] presented a rigorous model for anonymity in anonymous communication, based on indistinguishability definitions developed in [38]. Further, they claim that the anonymity notions proposed by them are stronger against a wide range of adversary capabilities as opposed to limited capabilities considered in similar work done previously.

Moreover, these notions are presented in the form of relations with respect to different attacker scenarios such as eavesdroppers, malicious peers and destinations. The achievable level of anonymity is proven under each relation through several exhaustive experiments. This study further provides a definition for *ultimate anonymity* in terms of computational anonymity for sender anonymity and unobservability, and claims that ultimate anonymity leads to inefficiency [33].

2.1.8 Summary

The above studies demonstrate the complexity of the theoretical foundations of anonymity, indicating the multifaceted nature of anonymity. Further, many provide formal definitions for those aspects, thereby facilitating effective comparison between different systems. However, their intentions are directed towards specific application classes such as anonymous data disclosure and anonymous communication. In our study, we focus on constructing a similar framework for distributed systems such as cryptocurrencies and we look at the current approaches towards this in the following section.

2.2 Anonymity aspects of cryptocurrencies

As pointed out at the outset, improved anonymity guarantees perceived by cryptocurrencies, have undoubtedly contributed towards their increased usage. Most cryptocurrency implementations deploy a decentralised architecture to eliminate the need for a trusted third party for transaction verification etc. and they use public ledger technologies instead to achieve such functionality. In addition, various cryptographic techniques are used to improve anonymity characteristics. In this respect, a practically achievable level of anonymity is a deciding factor for the adoption of a particular currency scheme by users. Nevertheless, measuring the level of anonymity achieved by a given scheme is not a straightforward task, even from a theoretical perspective.

With that background, we initiate our investigation into exploring anonymity aspects of cryptocurrencies, by providing a brief overview of the Bitcoin framework and its anonymity considerations. Thereafter, we study several solutions that have been introduced to improve the anonymity of Bitcoin and so-called Altcoins, while trying to capture different methodologies and theoretical frameworks used to model anonymity achieved by these constructions.

2.2.1 Overview of the Bitcoin Framework

Bitcoin is a decentralised cryptocurrency scheme based on a paradigm first proposed by Satoshi Nakamoto [67]. This system relies on an append-only public ledger, instantiated in the form of a "blockchain". Every Bitcoin transaction is stored on the blockchain publicly and this helps prevent spending the same funds (coins) more than once (also known as double spending). Due to the Bitcoin architecture, no central authority is required to manage the transactions and hence the system functions as a decentralised scheme. The base currency unit is termed as Bitcoin whereas the smallest currency unit is known as a Satoshi.

In the Bitcoin system, a user should possess a Bitcoin address in order to be able to receive funds. Users maintain Bitcoin Wallets for this purpose, and these wallets are identified by Bitcoin addresses, instead of users' real identities. Transactions take place between these addresses using cryptographic keys and signatures, without revealing the identities.

The Bitcoin network is a Peer-to-Peer (P2P) broadcast network, where nodes are connected with each other without a central server. All peer nodes in the network share the blockchain. The blockchain is a database structure that consists of a chain of blocks containing transactions and each block is linked to the previous block via its hash. Cryptographic techniques are used to structure the blockchain in a manner such that the order of blocks is preserved and transactions cannot be reversed.

Transactions are created and broadcast to the Bitcoin network, and participating nodes verify them based on the current status of the blockchain. These nodes can add the new transactions broadcast on the network to the existing blockchain by generating a new block. New blocks are constructed by the participant nodes, based on a Proof-of-Work (PoW) system. In a PoW system, participants use their computer processing power and time to produce a target piece of data, which is generally deemed difficult to generate, and this data is used by the other participants in the system to verify the scrutiny of the new data structure [67]. The process of generation of a new block in this manner is termed as mining and the nodes involved in mining are known as miners.

New coins are minted during the block generation process, which are assigned to respective miners as rewards based on the underlying implementation. In addition, transactions can also include transaction fees which act as an incentive for miners to include those transactions in a block. Miners add the new block to the current blockchain and broadcast the new chain to all nodes on the network. Participant nodes accept the longest chain of blocks as the valid blockchain at any given instance. The difficulty of block generation is automatically adjusted based on the total computational power of the miners on the network [12, 44, 77].

2.2.2 Bitcoin privacy

The privacy of Bitcoin users and transaction data has been widely spoken about in both academia and in the industry, showing that Bitcoin transactions are not anonymous. It is claimed that Bitcoin transactions are pseudonymous rather than being anonymous since Bitcoin users are represented by public keys instead of their real identities [7, 36, 44, 57, 63, 77]. Many research studies have proved that Bitcoin transactions can ultimately be linked to real world identities and hence are not even as pseudonymous as was claimed at the outset [7, 35, 36, 44, 63, 77]. The majority of these studies are based on experimental analyses that therefore provide emprirical, quantitative evidence for lack of acceptable level of anonymity in Bitcoin. We summarise some relevant studies below.

Ober et al. [71] analysed the dynamics of the Bitcoin transaction graph and claimed that the usage of multiple public addresses concurrently could pose a threat to the anonymity of the Bitcoin system. In a separate study, Reid et al. [77] constructed two networks; a transaction network and a user network, based on the data extracted from public Bitcoin data and showed that it is possible to gather details about Bitcoin users including their behaviour patterns from publicly available data.

Moreover, Androulaki et al. [7] presented an adversarial model to quantify privacy in the Bitcoin framework in terms of unlinkability of activities and indistinguishability of user profiles. Based on these metrics on the actual Bitcoin network, it has been shown that addresses corresponding to Bitcoin transactions can be extracted from the information that is publicly available on the blockchain and hence Bitcoin does not satisfy the property of unlinkability.

Meiklejohn et al. [57] also claimed that it is impossible to achieve unlinkability in the Bitcoin framework since two different Bitcoins can be easily distinguishable, which is also a violation of the property of fungibility [13]. Hence, it is argued that anonymity can only be considered with respect to the ownership of Bitcoins rather than the coins themselves [12].

Although the majority of above deanonymisation efforts are linked to the application layer, network level deanonymisation is also possible in the Bitcoin network. A recent study in [8] demonstrates such attacks and shows the same are harder to mitigate than the application level attacks.

From these studies, it is apparent that Bitcoin anonymity is compromised as addresses and transactions are linkable by the construction itself. As Bitcoin receives much criticism to that effect, many improvements to Bitcoin have been developed and new currency schemes have emerged with more promising anonymity expectations, which has led to the need for more concrete formalization of anonymity concepts.

2.2.3 Mixing protocols

The simplest solution for the anonymity problem is to mix coins or transactions of multiple users thereby making it difficult to establish a direct link between users and corresponding coins or transactions. Many early solutions were focused on using intermediary mixing services such as **BitLaundry**, **MixCoin** and **Bitcoin tumblers** to achieve anonymity [63]. They are not fully decentralised as the Bitcoin network and they depend on centralised third parties to achieve the mixing, often by charging a fee. If the coins involved in a mix are of similar values, then it makes it somewhat harder to link the coins to respective users, although that alone is not sufficient to hide the link unless new addresses are used [44]. In addition, the level of anonymity also increases with the number of users and transactions involved in the mix, which is the anonymity set. Hence, mixing solutions in general possess a slightly higher level of unlinkability, recipient anonymity and untraceability when compared to Bitcoin [63]. However, users have to trust the central mixing party not to retain any trace of their coins and also to return their coins after mixing [51]. Dependency on a centralised system is also a negative aspect as it could lead to a single-point of failure, to which many solutions have been proposed such as the fair exchange protocol proposed by [12]. Further, decentralised mixing approaches have been proposed and implemented recently as improvements to Bitcoin as well as Altcoins, while addressing these issues [44].

The **CoinJoin Protocol** is a widely accepted implementation that has been proposed to overcome the anonymity issue of Bitcoin and has led to several other anonymous protocols. In this protocol, a client-side application forms a single transaction by combining multiple inputs and outputs from multiple users, which is signed by those users and then sent to the network [56, 96]. In this setup, there is no involvement of a third party and thus it eliminates the need to trust a third party which is a drawback in many mixing protocols. However, if arbitrary amounts are involved, it requires additional pre-processing of the coins, which may introduce further overheads with respect to time and charges [56]. In addition, the availability of users who are willing to combine their coins/transactions at a given time may affect the achievable level of anonymity.

Dash is a cryptocurrency that has been built with a privacy-focus, and has been implemented based on the CoinJoin protocol. Dash uses a secondary network of full nodes on the Bitcoin network to mix transactions without using a centralised third party [26]. Known as the Dash Masternode Network, this network acts as a trust-less implementation to provide the required privacy in mixing. Dash uses a PrivateSend function which extends from the initial CoinJoin protocol and improves on decentralisation and denominations [26]. However, Dash still has some limitations such as requiring at least 3 participants to initiate a PrivateSend transaction and facilitating only common denominations. Further, it is still a centralised system to some extent due to its dependency on the Masternode network, and given enough time, transactions can be linked to respective users.

Research of Maurer et al. [56] claimed that initial CoinJoin transactions having common denominations can be linked to pseudonyms and thus proposed a solution to improve the unlinkability of CoinJoin transactions. Their study examined how knapsack mixing can be implemented to split outputs and how it could be combined with an input shuffling algorithm to reduce the linkability, while allowing different values of coins to be mixed together. Their evaluation results showed an increase in the difficulty of linking transactions in the new framework compared to the original CoinJoin transactions.

CoinShuffle is a decentralised Bitcoin mixing protocol that uses the Dissent (Dining-cryptographers Shuffled-Send Network) Protocol to achieve accountable group anonymity [80]. The Dissent protocol provides anonymous message transmission with shuffling that is consistent with the current Bitcoin framework and does not require a third party to do the mixing. Selij [84] simulated CoinShuffle transactions on a test Bitcoin network and analysed the linkability of those transactions. According to the results of this study, those transactions were identifiable among other transactions when the number of participants is small.

An improved solution, **CoinShuffle^{++}**, was proposed by Ruffing et al. to facilitate unlinkable transactions [81]. It is built on CoinJoin and the DiceMix P2P mixing protocol and achieves faster transaction times compared to CoinShuffle. The DiceMix protocol allows the broadcast of messages sent by a group of users anonymously in a decentralised manner [81]. However, the protocol assumes the existence of at least two honest users within the group to achieve an acceptable level of anonymity.

CoinParty is an improved mixing service which uses decryption mixnets together with threshold signatures to achieve strong anonymity measures [102]. Decryption mixnets consist of a set of mixing nodes, which mix the inputs that are passed through them and provide required encryption and decryption [44]. Threshold signatures enable claiming funds upon agreement of a majority of mixing nodes. In this construction, mixing happens in a distributed manner, thereby eliminating the problems associated with traditional mixing schemes. Further, CoinParty is considered as one of the first mixing protocols to achieve plausible deniability [102]. This construction also assumes the existence of at least two honest mixing nodes to establish the unlinkability of transactions. In addition, it takes longer to complete the mixing and to process the transaction, in comparison to the closest implementation, CoinShuffle.

ValueShuffle is a another protocol which has been developed as an extension to CoinShuffle^{++} protocol with an aim to overcome the problems of early approaches such as difficulties in integrating to the existing Bitcoin framework [79]. The ValueShuffle specification has been constructed from the CoinJoin protocol to achieve sender anonymity through decentralised coin mixing. Confidential Transactions are integrated in the protocol to hide the transaction values. Sender anonymity has been realised through the use of stealth addresses. Further, it is compatible with the Bitcoin architecture and is claimed to achieve a high level of anonymity when integrated with Bitcoin [79]. ValueShuffle transactions have lower transaction sizes compared to other solutions and therefore improves on the efficiency with respect to space and time. In addition, these transactions incur comparatively lower fees among other protocols based on CoinJoin [79]. However, it may still be possible to link input addresses to IP addresses at the network level unless an anonymous payment network is used.

Liu et al. [51] proposed an **unlinkable coin mixing scheme** which provides unlinkability without a trusted third party. This mixing scheme uses ring signatures with Elliptic Curve Digital Signature Algorithm (ECDSA) to achieve anonymity and it can be integrated to the existing Bitcoin framework. ECDSA is the algorithm used to generate and verify signatures in the Bitcoin network. This algorithm generates shorter key lengths and ring signatures provide anonymity of the sender. A mixing server is used to construct mixing transactions. Although this scheme achieves anonymity and scalability, the reliance on a central server degrades the decentralised implementation of cryptocurrencies. **ShareLock** mixing scheme provides a similar mixing solution which makes use of multi-party ECDSA and claims that it ensures anonymity against outsiders, senders and recipients [85].

Wang et al. [93] proposed the concept of a **secure escrow address** to mix Bitcoin transactions without a trusted third party in a decentralised environment. In this scheme, participants use a temporary escrow address generated through a secure distributed key generation process and they transfer some Bitcoins to this address. Then each participant creates a set of output addresses of all participants which are then shuffled to form a list of output addresses. Bitcoins are then transferred from the escrow address to the list of output addresses. Moreover, this scheme also achieves strong deniability. However, malicious participants could have an impact on the system.

Chator et al. [21] proposed a technique to improve the efficiency over mixing protocols through an enhanced means of sampling obfuscating transactions. This solution proposes a recoverable sampling scheme constructed based on programmable hash functions and it is claimed that it can be integrated to many existing cryptocurrency implementations. Such strategies can be used to leverage anonymity properties of existing mixing protocols.

CoinMingle is a recent development which has been proposed based on the CoinJoin protocol as a decentralised mixing scheme [97]. It is claimed that this construction utilises the advantage of several privacy-preserving techniques used by other existing cryptocurrency implementations. This work is developed using ring signatures and stealth addresses on top of the CoinJoin protocol in order to maximise the benefits of both mixing and cryptographic techniques. In addition, a mutual recognition delegation mechanism is used by the users to delegate messages to others.

The above solutions represent only a subgroup of existing mixing solutions and the very existence of such a large number suggests the compelling need for improved anonymity and privacy in cryptocurrencies. This is further supported by the emergence of Altcoins, which claim improved anonymity, as discussed in the following section.

2.2.4 Altcoin solutions

As previously mentioned, mixing protocols have inherent limitations that degrade the achievable level of anonymity. In addition, the involvement of malicious mixes could also significantly affect the anonymity of transactions. Hence, research on anonymity has diverted towards alternative solutions and many Altcoin implementations have emerged as a result.

Zerocoin relies on cryptographic methods to achieve anonymity while eliminating most of the problems faced by mixing protocols. It is regarded as one of the first anonymous electronic cash systems [6, 59]. The Zerocoin protocol uses one-way accumulators for storing values and zero-knowledge proofs for spending coins while breaking the link between transactions [59, 63]. Using one-way accumulators, participants can combine a set of values to generate a single block of data with a constant size and prove that a specific data value exists in this block [59]. A zero-knowledge proof allows a user in a system to prove the existence of some specific information without revealing that information to other users in the system. The Zerocoin implementation improves anonymity considerably in comparison with mixing methods. However, it does not hide transaction values and uses only fixed coin values [6, 82]. In addition, the size of transactions increases due to the cryptographic content that needs to be stored, and therefore transaction processing takes longer than mixing protocols. Further, it requires modifications to the existing Bitcoin architecture to be able to realise its practical implementation [59].

ZeroCash provides improvements to the original Zerocoin e-cash system with enhanced assurance of anonymity. This scheme uses zero-knowledge Succinct Non-interactive Arguments of Knowledge (zk-SNARKs) for verification of transactions and Decentralised Anonymous Payment (DAP) schemes, which enables users to pay directly to each other privately [82]. Using zk-SNARKs, not only can one prove the existence of some specific information, but it is also possible to prove the possession of that information, without revealing it or without interacting with the verifiers. Further, DAP transactions conceal the sender, recipient and also the values of transactions, as opposed to Zerocoin. Moreover, transaction sizes are also smaller compared to Zerocoin (however, they are still much larger than that of Bitcoin), which makes it more efficient than Zerocoin [82]. It also allows transactions of variable amounts. Despite the benefits over Zerocoin, the ZeroCash protocol does not hide IP addresses of end users. In order to achieve this anonymity, an anonymity network such as Tor, which facilitates anonymous transmission of data among users via tunnelling through a large network of distributed nodes, should be used [82]. Considering these facts, Zerocash can be regarded as having a moderate resistance against deanonymisation compared to Bitcoin. Moreover, ZeroCash is not compatible with the existing Bitcoin network and performance of the payment scheme is still affected by the heavy cryptographic computation involved.

Zcash is a virtual currency that was implemented recently based on the ZeroCash framework. This scheme uses a shielded payment scheme with zk-SNARKs to provide an anonymous, decentralised currency system [39]. Transaction values in Zcash can be either shielded or transparent, and it works similar to Bitcoin in the latter case. In comparison with ZeroCash, Zcash uses only the original Bitcoin transaction type with additional operations to handle shielded transactions. Although shielded transactions provide better anonymity, it is claimed that Zcash transactions are still linkable to some extent as it supports both transparent and shielded transactions at the same time [75]. Although Zcash possesses a higher level of anonymity than Zerocash with respect to confidentiality and the size of anonymity set, both Zcash and Zerocash seem to have similar level of resistance to deanonymisation since the majority of Zcash transactions are transparent (unshielded) at present. The study by Zhang et al. [98] supports this claim by proposing improved means for deanonymisation of Zcash transactions based on address clustering.

The **CryptoNote** protocol was developed by Saberhagen [91] with an aim to address the weaknesses of Bitcoin, mainly the inflexibility of the system which resists the addition of new features to the Bitcoin framework. This protocol achieves both untraceability and unlinkability through the use of one-time linkable ring signatures with non-interactive zero-knowledge proofs. Linkable ring signatures allow a user in a group of users (a ring of users) to sign a message on behalf of the group, without revealing the identity of the signer [49]. Hence, every user in that group is equally likely to be the signer for an outsider. This protocol also can be regarded as a form of non-interactive mixing [60]. Instead of one public key, the set of public keys belonging to all the users in the group is used to verify the signature by an external party. Several cryptocurrencies such as ByteCoin, DigitalNote, DarknetCoin and Aeon have been developed based on the CryptoNote architecture [44]. In spite of the claims for unlinkability, Noether [68] mentions that CryptoNote transactions may still be linkable as proven through some targeted attacks. Further, transaction values are not hidden and the transactions are larger in size compared to Bitcoin due to the size of cryptographic keys used. Therefore, Cryptonote can be considered as having similar levels of resistance to deanonymisation as Zcash.

Monero is an Altcoin with strong anonymity features. It was initially built on the CryptoNote protocol [68]. The anonymity of Monero has been strengthened continuously since its inception and the current framework uses linkable ring signatures to hide the senders. This provides a way of mixing Monero coins with garbage coins named “mixins” [60]. Further, Ring Confidential Transactions (Ring CT) are used to hide transaction values which was not possible with CryptoNote [44, 70]. Confidential Transactions (CTs) provide a means of hiding transaction amounts while still allowing the transaction verification by other participants whereas Ring CT incorporates CTs with linkable ring signatures to improve on the original CryptoNote protocol [69]. This introduces a minor overhead in terms of block size in comparison to CryptoNote. In addition, unique one-time addresses (i.e. stealth addresses) are used to hide the recipient. However, it may still be possible to link the stealth addresses to network level IP addresses, which might affect the level of anonymity.

Although Monero is claimed to be one of the most anonymous cryptocurrencies, recent studies have proved that transactions can be de-anonymised [46, 60]. Miller et al. [60] showed that Monero transactions can be linkable due to the mixin sampling strategy used. In a separate study, Kumar et al. [46] developed three attack scenarios and proved that Monero's untraceability feature can be breached. Thus, the resistance of Monero against deanonymisation can be regarded as similar to Zcash.

Mimblewimble is an alternative cryptocurrency protocol which proposes a novel idea to achieve privacy through the aggregation of confidential transactions, without storing individual transaction data. It is the protocol at the core of the implementations Grin and Beam [42, 74]. Although the protocol provides better scalability, unlinkability cannot be assured among the participants who undertake the aggregation [79].

Together with the growth of cryptocurrencies, Payment channel networks (PCNs) have been introduced to address scalability issues related to blockchain technology. Some claim that these do not provide acceptable levels of privacy [27, 53]. Hence, several PCN solutions have emerged aiming to preserve privacy. **Bolt** is one such scheme, which facilitates an anonymous payment channel which could be used by existing anonymous cryptocurrencies. Blind Off-chain Lightweight Transactions (Bolt) provide a basis for defining payment channels that ensure the anonymity of transactions in decentralised cryptocurrencies while storing a part of the transaction data outside the blockchain [34]. However, it is claimed that Bolt provides relationship anonymity (i.e. a sender cannot be linked to corresponding recipient/s) only in the presence of a single node between the participants [27]. Malavolta et al. [53] proposes the use of anonymous multi-hop locks in the design of PCNs to preserve privacy. Multi-hop locks ensure that transactions are updated atomically along a payment path through the channel.

Erdin et al. [27] claim that this approach does not provide sender anonymity although relationship anonymity can still be preserved.

Another aspect of de-anonymisation of cryptocurrency transactions is the linkability to metadata such as IP addresses and real world identities using network and off-network data [9, 99]. The solution proposed by Fanti et al. [29] aims to strengthen the protection of the existing Bitcoin network against these de-anonymisation attacks. They proposed a network protocol, **Dandelion**$^{++}$, a modified construction of a previously proposed protocol Dandelion [16], through the use of randomised routing algorithms and graph topologies, which however is still susceptible to routing attacks originating at the Internet Service Providers (ISPs).

Blockchain technology is now being used as a platform for many applications other than cryptocurrencies. **Ethereum**, is one such platform having an account-based system. The scheme deploys smart contracts and facilitates Distributed Applications (DApps) including the cryptocurrency Ether [22, 30]. Smart contracts are executable programs that are accessible via a particular account. In addition, there are accounts corresponding to users and each account is associated with a balance. A user can invoke a smart contract by creating a transaction with the address of the contract as the recipient [30]. Due to the account-based structure, Ethereum exhibits a weaker level of pseudonymity than Bitcoin [18, 23, 100]. As pointed out by [18], account reuse leads to successful deanonymisation attempts through user profiling despite the introduction of new privacy features such as coin mixers. Another study demonstrates how to compromise pseudonymity in smart contracts through the use of stylometry [50].

2.2.5 Anonymity landscape of cryptocurrencies

As it is evident from the above, numerous solutions have emerged in the race to construct anonymous cryptocurrencies. As a result, there is a pressing need for a reliable means of evaluating and comparing privacy and anonymity considerations of these alternatives. Unsurprisingly, many have attempted to provide reasonable grounds for such comparisons through diverse means. We now explore the outcomes of several similar attempts.

Khalilov et al. [44] conducted a survey into comparing anonymity features of a wide range of existing cryptocurrencies in a comprehensive manner. They attempted to group the underlying constructions of these schemes around three aspects of anonymity: *untraceability*, *hidden values*, and *hidden IP addresses.* However, the study does not formally define these aspects. A similar study was carried out by Conti et al. [24], 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 [3] 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 another survey paper, Alsalami et al. [2] 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 categorisation also, similar to the work in [44], provides a very high level picture of the anonymity levels based on the techniques used by the schemes.

Herskind et al. [37] in their study into the privacy of cryptocurrencies, provides a systematic literature review similar to [44] by analysing existing literature on anonymity and privacy aspects of electronic cash, mainly based on experimental analyses. Their findings provide categorisations in terms of the deanonymisation techniques, in addition to the grouping by anonymisation methods used by different currency schemes, yet no formal privacy notions are considered.

In a separate study, Lee [47] explores the extent of anonymity achieved by a group of chosen cryptocurrency implementations that claim to have improved anonymity characteristics. Their work focuses on evaluating anonymity with respect to *privacy*, *Untraceability* and *Fungibility*, but no formalised definitions are given. According to this grouping, both Monero and Zcash demonstrate very strong anonymity, while satisfying all three properties.

The work in [61] presents a software architecture model for an anonymous cryptocurrency in terms of three layers and defines anonymity at the secret-sharing layer. That study provides a set of anonymity definitions based on [72] in terms of anonymity, unlinkability and pseudonymity, and proposes four attack models to capture security vulnerabilities in cryptocurrency schemes. Although that work provides a set of anonymity definitions based on those four attack models, our work differs in the sense that we consider an exhaustive adversarial model at the transaction layer, which is capable of capturing a plethora of different attacker scenarios. In addition, our work includes a study of interdependencies of anonymity definitions through a set of formalised theorems.

We summarise the existing work discussed above in Table 2.1, while comparing the same with our methodology.

Table 2.1: Comparison of our methodology with similar work

Source/s	Basis/Method	Type	Anonymity notion/s	Applicable cryptocurrencies
[77]	Experimental analysis	Quantitative	No formal notions	Bitcoin
[78]	Experimental analysis	Quantitative	No formal notions	Bitcoin
[58]	Experimental analysis	Quantitative	No formal notions	Bitcoin
[87]	Experimental analysis	Quantitative	No formal notions	BitIodine
[7, 65, 71, 96]	Quantitative analysis	Quantitative	Activity unlinkability and user profile indistinguishability	Bitcoin
[43, 75]	Experimental analysis	Quantitative	Linkability	Zcash
[64, 91, 94, 95]	Implementation techniques	Qualitative	Untraceability, Unlinkability	Cryptonote
[24]	Based on distinct properties	Qualitative	Internal/external unlinkability, untraceability, fungibility, deniability (no formal definitions)	Bitcoin variants, CoinJoin, Cryptonote based variants
[14]	Network analysis	Quantitative	Anonymity degree	Bitcoin, Zcash
[19]	A transaction model based on blockchain semantics	Qualitative	No formal notions	Bitcoin, Ethereum and Hyperledger Fabric blockchains
[2]	Techniques used for anonymisation	Qualitative	Pseudonymity, Set anonymity, full anonymity, confidential transactions (No formal definitions)	Bitcoin and variants, Ethereum, Cryptonote based variants
[3]	Based on existing literature	Qualitative	Unlinkability, fungibility, untraceability, hidden values, unlinkability of IP addresses (No formal definitions)	Bitcoin variants, Cryptonote based variants
[44]	Anonymity/privacy improvement methods	Qualitative	untraceability of input/output addresses and transactions, hidden values, hidden IP addresses (No formal definitions)	Bitcoin variants, CryptoNote based, Mimblewimble
[37]	Based on existing literature	Qualitative	Deanonymisation techniques No formal notions	Bitcoin variants, Cryptonote based variants
[47]	Qualtative analysis	Qualitative	Privacy, Untraceability, Fungibility (no formal definitions)	Bitcoin, Dash, Monero, Verge, PIVX,Zcash
[61]	Software architecture model	Qualitative	Anonymity, unlinkability pseudonymity	Bitcoin, CoinJoin CoinShuffle, Zerocoin Zerocash, Cryptonote Mimblewimble
Our work	Anonymity framework based on a comprehensive adversarial model	Qualitative	Indistinguishability and Unlinkability of senders, recipients, value and metadata (formal definitions)	Decentralised cryptocurrency schemes

2.3 Provable security

Cryptographic security has been studied extensively in the literature, aiming to establish reliable means for modelling security in cryptographic systems. Due to the complexity of security considerations, it is impossible to achieve perfect security in any system. Instead, one needs to resort to a satisfactory security level that can be proven through some manner [45]. Hence, security of these systems are considered with respect to provable security.

According to [88], provable security can be considered under two main categories: unconditional and computational security. While the former is defined in terms of information theoretic primitives representing idealistic security against all possible adversaries, the latter is developed around certain classes of adversaries, who are computationally bounded in some sense. Many different methodologies have been devised to model computational security, and two widely used methods are game-based and UC-framework based security modelling.

2.3.1 Game-based security

One of the popular methods of modelling cryptographic security is through the use of conceptual security experiments that are defined in terms of games. Traditional game-based security modelling involves constructing an experiment around a single security property where a sequence of activities takes place between an attacker (possessing certain capabilities as per the security property) and a challenger [55], which are modelled as polynomial time algorithms. The security goal is defined by a winning condition associated with each experiment. If the adversary meets the winning condition, then the adversary wins the game. The probability of the adversary winning the game is considered as a measure of the security goal specific to that particular experiment [88]. For a given security goal, more than one security experiment can be defined with different capabilities assigned to the adversary in each experiment. These types of security definitions are considered for representing computational security as the parties involved in the game are computationally bounded in some sense, i.e. without unlimited computational powers.

There are several steps in a typical security experiment. Initially, the challenger prepares the setup for the game and may or may not receive data as inputs from the adversary, depending on the modelled scenario. The challenger will almost always give the public parameters that result from the setup to the adversary. Then the adversary, after executing some actions, perhaps in multiple rounds of iteration, gives some outputs to the challenger, who then performs particular action/s with it. The adversary then gets to see the output of the challenger's action and makes a guess as to what the action of the challenger was. The adversary wins the game if the guess is correct or loses otherwise. We say that the security goal is achievable if the probability of adversary winning the game is *negligible*. The term *negligible* in this context means that the probability is either close to *zero* or close to a target probability which would have been the case without any capability (or any additional information) on the adversary. For example, the target probability of an adversary (who does not have access to any additional information) guessing correctly the challenger's choice from the set $\{0, 1\}$, would be $1/2$.

In some complex cryptographic schemes, security proofs are formulated as a sequence of games [86]. Generally, those games are defined in such a way that each game differs slightly from its predecessor. In this case, the goal is to prove that the probability of adversary winning a certain game is very close to the winning probability of the subsequent game and eventually the probability of winning the last game is trivially negligible by construction.

2.3.2 Universally composable security

Universally composable security modelling, in comparison with the game-based approach, is capable of providing a model for the overall security of a certain cryptographic task, even when coupled to another arbitrary system. This methodology involves defining an ideal process to carry out the task in a secure manner and if a protocol emulates this ideal process such that no adversary can tell whether they are interacting with the ideal process or the protocol, then the protocol is said to securely realise the task [20]. In this setup, the ideal process consists of an ideal functionality, which acts as a trusted party, which executes the process in the presence of a simulated adversary in a setting facilitated by an environment. The environment also interacts with the protocol in the same manner. The protocol is considered UC-secure if there exists a simulated adversary such that, no environment can distinguish between interacting with the ideal functionality or the protocol [20].

Although the UC-framework has been widely used in the analysis of security of cryptographic protocols, it places all the subtlety of specifying what constitutes the right notion of ”security” onto the definition of this so-called ”ideal functionality” which, unlike the much simpler definitions afforded in the game-based approach, is far from trivial.

2.4 Research gaps

As it is evident from our findings in Section 2.2, anonymity aspects of existing cryptocurrencies are often evaluated based on experimental analyses, mostly by demonstrating the possibility of deanonymisation of transactions. Such studies have resulted in multiple different interpretations of anonymity due to the diversity of these constructions. Hence, it is not feasible to compare anonymity across different constructions, thus highlighting the absence of a common framework to evaluate anonymity of different cryptocurrencies as a major research gap in this context.

Moreover, precise and unambiguous notions are key to represent anonymity attributes of cryptocurrency schemes in a reliable manner due to their complexity. As summarised in Table 2.1, the notions or properties used to represent anonymity aspects in existing studies are mainly those borrowed from the theoretical representations used in other contexts such as we discussed in Section 2.1. Hence, they do not necessarily facilitate the assessment and comparison of cryptocurrencies in terms of a common, fine-grained, formal qualitative model of anonymity, which indicates another research gap, which is the absence of formal anonymity definitions.

Another noteworthy aspect in this connection is to study the relationships among anonymity notions and their effect on the achievable extent of anonymity. This enables one to understand the strengths and weaknesses of these notions and how anonymity differs in various scenarios. As many existing studies are isolated evaluations, they do not facilitate such analyses. Hence, the lack of comprehensive studies into relationships of anonymity notions opens up another research direction.

In order to address above research gaps in the context of cryptocurrencies, we propose a common framework, which enables such comparison through formal anonymity notions. Table 2.1 compares the terminology used to represent anonymity in above studies with our work. We present the details of the proposed framework in chapters 3, 4 and 5.

Chapter 3

An Abstract Cryptocurrency Scheme

This chapter lays the foundation for this research study by introducing the fundamental building blocks of the proposed framework. We begin this process by constructing an abstract scheme for a generic cryptocurrency system. To accomplish this goal, we define the typical functionality in a cryptocurrency scheme in terms of a set of operations. In addition, we also develop a set of correctness properties in order to establish the correctness of the scheme in the presence of honest functionality. Further, we construct a comprehensive adversarial model to ensure the security requirements relevant to the functionality of the scheme and compose game-based security definitions to address the same.

3.1 Introduction

As evident from studies such as [44, 66] which attempt to compare anonymity aspects of various cryptocurrency schemes, the underlying implementation methods are highly diverse. Even though they use blockchain technology as the foundation, cryptographic primitives and communication mechanisms underpinning their functionality differ significantly between them. As a consequence, it is infeasible to directly compare different constructions in an effective and meaningful manner.

A uniform evaluation framework for anonymity requires a common basis that is applicable for every system. This requires the formulation of a unified means for representing the functionality of diverse cryptocurrency schemes in a standardised manner. Hence, we consider a generic cryptocurrency scheme, which can model the common functionality across any currency scheme irrespective of the underlying implementations, as the basis for our anonymity framework.

The plausibility of such a new scheme is paramount for its applicability. For this purpose, the correctness of the functionality needs to be established in an honest operation. For example, fund balances should be updated correctly after a transaction takes place, so that money does not appear in the system out of thin air. Further, there are other considerations in relation to the behaviour of a system and one such requirement is that the states of the ledger should be a partial ordering (symbolised by $\preceq$), so that the most recent qualifying state is chosen as the next state of the system.

Moreover, the proposed scheme must be functionally secure against adversarial influences such as spending the same currency units more than once (i.e. double spending) in establishing the viability of the scheme. Therein, this chapter establishes the fundamentals of the models used in our research.

3.2 Our Contributions

Our major contribution in this chapter is the construction of an abstract cryptocurrency scheme, modelling common functionality demonstrated by fully decentralised cryptocurrency schemes. In this construction however, we do not consider the specific details of the communication mechanisms behind these schemes, which may significantly differ between one another. During this process, the entities in the scheme are identified and its operations are defined based on the required functionality.

In order to establish the functional correctness of the scheme, we define a set of experiments which focuses on the accuracy of individual operations of the scheme, as well as expected honest functionality.

Another vital contribution in this chapter is the formulation of a comprehensive adversarial model in relation to the security aspects of the proposed scheme. This model captures a wide range of attacker scenarios, thereby enabling fine-tuning of security in fully decentralised systems such as cryptocurrencies. Apart from facilitating functional security definitions, this model also forms the basis for our anonymity framework, which is described in detail in Chapter 4.

3.3 Notation

We use the notation given in Table 3.1 in order to represent the entities within the proposed scheme and the terminology used for correctness and security definitions.

Table 3.1: Notation

Description	**Notation**
Security parameter	$\lambda : \lambda \in \mathbb{Z}^+$
System state e.g. current state	p
A set of states	P
p_0 is an earlier state in time than p_1 or $p_0 = p_1$ i.e. p_0 is in p_1's history	$p_0 \preceq p_1$
p_0 is not in the history of p_1	$p_0 \npreceq p_1$
$p_0 \preceq p_1 \quad \wedge \quad p_0 \neq p_1$	$p_0 \prec p_1$
A payment address	a
Public key/Private key of a payment address	$a_{pk},\ a_{sk}$
Ordered tuple of one/more addresses (senders/recipients) of secret keys	$\bar{S},\ \bar{R}$
Ordered tuple of one/more addresses containing only public keys	$S,\ R$
Number of items in a tuple S	$\lvert S \rvert$
Public and private parts of a transaction	$t_p,\ t_s$
Ordered tuples of input and output values of a transaction	$V_{old},\ V_{new}$
Metadata for a transaction	m
Excess value of a transaction (fees + minted value)	V_x
A tuple of addresses of miners	R_m
Concatenation of tuples A, B, Set minus operation of tuples A,B	$A \Vert B,\ A \setminus B$
Empty set, empty tuple	$(\emptyset/\{\}),\ ()$
a_{pk} is an element of tuple R / not an element of R	$a_{pk} \in R \ / \ a_{pk} \notin R$
Every element in tuple $R^{'}$ is in tuple R	$R^{'} \subseteq R$
Return the element in a set with one element H	$H[0]$
Randomly choose an element a from set S with randomness ρ	$a \leftarrow \{S; \rho\}$
If [*condition*] is false after $<$ *statement* $>$, then return 1	$<$ *statement* $>$ [*condition*]
If $\langle$ *condition* $\rangle$ is false after $<$ *statement* $>$, then return 0	$<$ *statement* $>$ $\langle$*condition*$\rangle$
If $a = \perp$ then return c, else return b	$a?b : c$
If $a = \perp$ then return b, else return a	$a?\_ : b$
Return X if y, otherwise return 1	X^y
Standard operations on Associative Arrays	$\texttt{Operation}_{AA}$
Set of all possible system states	$\mathbb{P}$
Set of all possible addresses (both public and secret parts)	$\mathbb{A}$
Set of all possible transactions (both public and secret parts)	$\mathbb{T}$
Set of all possible transaction values i.e. V_{old}, V_{new}, V_m	$\mathbb{V}$
Set of all possible metadata values i.e. m	$\mathbb{M}$
Set of all natural numbers	$\mathbb{N}$

3.4 Proposed Currency Scheme

We define a currency scheme in terms of a security parameter $\lambda \in \mathbb{Z}^+$ and we term the initial state of the system, as the *genesis state.* The system consists of a set of payment addresses, representing senders, recipients and miners in the system, and each address is composed of a private key and a public key (address) or an identity.

A transaction may take place between multiple senders and recipients, and consists of a private and a public part. Once a transaction takes place, its public part is broadcast to the participant network. A minting operation collects unminted transactions at any given point in time and generates a new state. In other words, minting is the process of accepting transactions (including operations such as adding new addresses) into the state. These minting operations are carried out by participating nodes in the system's network (i.e. miners) and more than one candidate state may exist at any given time. New currency units are generated as a result of the minting process, as per the underlying implementation of the scheme.

An adjudicate operation selects the rightful new state of the system from a set of candidate states. A system state p is defined by the implementation and the state typically records 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 now look at the detailed functionality of this generic scheme in the following section.

3.4.1 Functionality

We define the functionality of the proposed scheme based on common operations of a generic cryptocurrency scheme. These functions mainly include operational and other supporting roles such as validation, which facilitates the exercise of defining correctness and security. Table 3.2 summarises the structure of these operations. It should be noted that the symbol ρ here represents the randomness for functions that require it.

Table 3.2: Functionality of the scheme.

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

The initial setup of the scheme is defined by the `Init` algorithm in terms of a security parameter λ and this process generates the genesis state. Payment address creation process, `CreateAddr` takes 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 in the state. An exception occurs if creating an address does not require a change to the state (based on the scheme as in Bitcoin), in which case the transaction can be regarded as empty. Public and private keys can be validated with respect to a given state, p through `IsValidPubAddr` and `IsValidPubAddr` functions. `ExtractID` function can be used to extract the identity of a given address.

The `CreateTxn` function facilitates the creation of a transaction (t_p, t_s) with unspent funds from one or more senders (S) having corresponding input values (V_{old}) and output values (V_{new}) for recipients (R), 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, i.e. `IsValidPubTxn` and `IsValidSecTxn`. 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 (`ExtractSenderPubAddr/RecipientPubAddr/InputVal/OutputVal`).

The `Mint` 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, based on the implementation specifications. `IsValidState` algorithm checks the validity of a given state with respect to a given security parameter. Checkpoint states are considered as descending from the genesis state. A given state can be designated as a checkpoint state through the `CreateCheckpointState` function based on the particulars of the state, which can be retrieved later through the `RetrieveCheckpointState` operation. The genesis state is considered as the first checkpoint and the algorithm `IsGenesisState` can be used to identify the genesis state corresponding to a given security parameter.

It is noteworthy that we model only the generic functionality of a cryptocurrency scheme in this scheme. Hence, we do not consider the specifics of the underlying consensus mechanism or the network in this work. However, there may be additional functionality associated with real world cryptocurrency systems, e.g. Smart contracts with Ethereum. In order to capture such additional features, we define a supplementary function `AdditionalFunctionality`. This enables us realise the security implications of functionality of a scheme that may possess operations outside our base model.

3.4.2 Definition of a Cryptocurrency Scheme

We now formally define an abstract cryptocurrency scheme based on the above functionality as follows:

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

3.5 Correctness of the currency scheme

In this section we model the correctness of a scheme in terms of the functionality. We consider the correctness of individual functions as well as the overall functionality described above, in terms of a set of experiments, each of which must return *true* (or '1' equivalently) upon receiving valid inputs. A summary of the experiments is listed in Table 3.3 while details are discussed later in following sub sections.

Table 3.3: List of experiments for correctness.

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

3.5.1 Generating input data

We define several functions to generate input data required for the correctness experiments in terms of λ and a tuple of arbitrary bit strings $\rho \in (\{0,1\}^*)^*$. Bit strings are mapped to required datasets through separate and arbitrary surjective functions with following mappings:

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

We use the function label $\texttt{DeserialiseS}_{\texttt{W}}$ to represent a repeated execution of the respective $\texttt{Deserialise}_{\texttt{W}}$ function in the instances where a set of data inputs is to be generated instead of a single data item (E.g. we can use this function to expand universally quantified bit strings into universally quantified system states, histories, addresses etc.).

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 3.1). ρ is decomposed to a tuple $\rho = (\rho_1, \rho_2, ...)$ within experiments where ρ_1, ρ_2 etc. are bit strings of arbitrary (finite) length of which the combined length is equal to the length of ρ. i.e. $\sum_i |\rho_i| = |\rho|$ ($|\rho|$ is the length of the bit string ρ, but not the number of bit strings in the tuple). Each ρ_i is used to generate required inputs for the experiments through respective `Deserialise` functions.

```
GenerateState($\lambda; \rho$)
  1. $p \leftarrow \texttt{Deserialise}_{\texttt{P}}(\lambda; \rho)$
  2. if $\neg(\texttt{IsValidState}_{\pi}(p, \lambda))$
  3.     return $\perp$
  4. return $p$

GenerateTxns(p, $\lambda; \rho$)
  1. $T \leftarrow \texttt{DeserialiseS}_{\texttt{T}}(\lambda; \rho)$
  2. for all $(t_p, t_s) \in T$ do
  3.     if $\neg(\texttt{IsValidSecTxn}_{\pi}(t_p, t_s, p))$
  4.         return $\perp$
  5. return $\{(t_p, t_s)\}$

GenerateTxnValues(R, $\bar{\texttt{S}}$, $\lambda; \rho$)
  1. $V_{old} \leftarrow \texttt{DeserialiseS}_{\texttt{V}}(\lambda; \rho)$
  2. $V_{new} \leftarrow \texttt{DeserialiseS}_{\texttt{V}}(\lambda; \rho)$
  3. $m \leftarrow \texttt{Deserialise}_{\texttt{M}}(\lambda; \rho)$
  4. if $(|V_{old}| \neq |\bar{S}|) \vee (|V_{new}| \neq |R|)$
  5.     return $\perp$
  6. return $(V_{old}, V_{new}, m)$

GenerateAddr(p, $\lambda; \rho$)
  1. $X \leftarrow \texttt{DeserialiseS}_{\texttt{A}}(\lambda; \rho)$
  2. for all $(a_{pk}, a_{sk}) \in X$ do
  3.     if $\neg(\texttt{IsValidPubAddr}_{\pi}(a_{pk}, p))$
  4.         return $\perp$
  5.     if $\neg(\texttt{IsValidSecAddr}_{\pi}(a_{pk}, a_{sk}, p))$
  6.         return $\perp$
  7. return $((a_{pk}, a_{sk}))$

GenerateMintAddr(p, $\lambda; \rho$)
  1. $R_m \leftarrow \texttt{DeserialiseS}_{\texttt{A}}(\lambda; \rho)$
  2. $V_m \leftarrow \texttt{DeserialiseS}_{\texttt{V}}(\lambda; \rho)$
  3. for all $a_{pk} \in R_m$ do
  4.     if $\neg(\texttt{IsValidPubAddr}_{\pi}(a_{pk}, p))$
  5.         return $\perp$
  6. return $R_m$
```

Figure 3.1: Functions used to generate input data from bitstring ρ

Further, ρ_i strings are also used to introduce randomness to the operations performed within experiments with the notation, $\texttt{Function}(\textit{parameters} \; ; \; \rho_i)$. In the case where the length of ρ is not sufficient to produce the required number of ρ_i bit strings, the corresponding experiment is terminated by returning 1 (i.e. the experiment terminates with success). In addition, we also introduce several helper functions that help improve the readability of the correctness experiments (Figure 3.2).

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

GenerateSetofStates(k, λ, ρ)
 1. $P_{set} \leftarrow \{\}$
 2. $m \leftarrow |\rho| - k$; $k \leftarrow k+1$
 3. $n \leftarrow (\{1,..,m-1\}; \rho_k)$
 4. for  $j \in \{1,..,n\}$  do
 5.     $k \leftarrow k+1$; $p_j \leftarrow \mathtt{GenerateState}(\lambda; \rho_k)$
 6.     if $\mathtt{IsValidState}_\pi(p_j, \lambda)$, $P_{set} \leftarrow P_{set} \cup \{p_j\}$
 7. return  $(P_{set}, k)$

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

Figure 3.2: Helper functions for correctness

```
$\mathbf{Exp}_\Pi^{\textit{checkpoint-monotonicity}}(\lambda, \rho)$
 1. $p_0 \leftarrow \mathtt{GenerateState}(\lambda; \rho_1)$ $[p_0 \neq \perp]$
 2. $p_0^c \leftarrow \mathtt{CreateCheckpoint}_\pi(p_0, \lambda; \rho_2)[p_0^c \neq \perp]$
 3. $p_1 \leftarrow p_0^c$
 4. $n_c \leftarrow 1$; $k \leftarrow 2$
 5. while $(k < |\rho|)$ do
 6.   $X \leftarrow \mathtt{EvolveState}(p_1, k, \lambda; \rho)$ $[X \neq \perp]$
 7.   $(p_2, \ell) \leftarrow X$; $k \leftarrow \ell$
 8.   $b \leftarrow \{\{0,1\}; \rho_k\}$;  $k \leftarrow k+1$
 9.   if  $b = 1$
10.     $H \leftarrow \mathtt{CreateCheckpoint}_\pi(p_2, \lambda; \rho_k)$
11.     if  $H \neq \perp$, $p^c \leftarrow H$; $n_c \leftarrow n_c + 1$
12.     $k \leftarrow k+1$
13.   else
14.   $p_1 \leftarrow p_2$
15. for  $j \in \{1,..,n_c - 1\}$  do
16.   $p_1^c \leftarrow \mathtt{RetrieveCheckpointState}(p^c)$
17.   if $p_1^c = \perp$, return 0
18.   $p^c \leftarrow p_1^c$
19. return  $p^c \overset{?}{=} p_0^c$

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

$\mathbf{Exp}_\Pi^{\textit{retireve-checkpoint}}(\lambda, \rho)$
 1. $p_0 \leftarrow \mathtt{GenerateState}(\lambda; \rho_1)$  $[\, p_0 \neq \perp \,]$
 2. $p_1 \leftarrow p_0$
 3. while $\mathtt{RetrieveCheckpointState}_\pi(p_1) \neq p_1$ do
 4.     $X \leftarrow \mathtt{RetrieveCheckpointState}_\pi(p_1)$
 5.     if  $X = \perp$,  return 0
 6.     $p' \leftarrow X$
 7.     if  $\neg(\mathtt{IsValidState}_\pi(p', \lambda))$,  return 0
 8.     if  $p' \npreceq p_1$,  return 0
 9.     $p_1 \leftarrow p'$
10. return  $\mathtt{IsGenesisState}_\pi(p_1, \lambda)$
```

Figure 3.3: Correctness experiments 1

Figures 3.3 and 3.4 list all experiments that establish the correctness of the proposed scheme. The property of *checkpoint-monotonicity* ensures that checkpoint states form a partial ordering whereas *adj-monotonicity* preserves

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

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

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

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

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

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

```
$\mathbf{Exp}_{\Pi}^{\textit{extract-id}}(\lambda, d, \rho)$
 1. $p_0 \leftarrow \texttt{GenerateState}_\pi(\lambda; \rho_1)$   $[\,p_0 \neq \perp\,]$
 2. $(a_{pk}, a_{sk}, t_p, t_s) \leftarrow \texttt{CreateAddr}_\pi(p_0, d; \rho_2)$
 3. $d' \leftarrow \texttt{ExtractID}(a_{pk}, p_0)$
 4. $b \leftarrow (d' \overset{?}{=} d)$; $b' \leftarrow (\texttt{IsMintable}_\pi(\{t_p\}, p_0))$
 5. return $b \wedge b'$
```

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

```
$\mathbf{Exp}_{\Pi}^{\textit{genesis-state}}(\lambda, \rho)$
 1. $p_0 \leftarrow \texttt{GenerateState}(\lambda; \rho_1)$   $[\,p_0 \neq \perp\,]$
 2. if  $\texttt{RetrieveCheckpointState}_\pi(p_0) = p_0$
 3.     return  $(\texttt{IsGenesisState}_\pi(p_0, \lambda))$
 4. $X \leftarrow \texttt{EvolveState}(p_1, 1, \lambda; \rho)$   $[\,X \neq \perp\,]$
 5. $(p_1, k) \leftarrow X$;  $b \leftarrow \texttt{IsValidState}_\pi(p_1, \lambda)$
 6. $b' \leftarrow \neg\ \texttt{IsGenesisState}_\pi(p_1, \lambda)$
 7. return $(b \wedge b')$
```

Figure 3.4: Correctness Experiments 2

the ordering of the states while ensuring that the most recent rightful state is always chosen by the `Adjudicate` operation (Figure 3.3). The correctness of the `RetrieveCheckpointState` is ensured through the *retireve-checkpoint* experiment. Figure 3.4 lists the experiments that ensures the correctness of `Init`, `CreateAddr`, `CreateTxn`, `Mint`, `Adjudicate` and `CreateCheckpoint` operations. In addition, the correctness of support operations is established through the experiments *extract-txn-data* and *genesis-state.* Accordingly, the correctness of the proposed scheme is defined as follows:

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* $X \in$ $\{$*init, create-addr, create-txn, extract-txn-data, mint, adjudicate, adj-monotinicity, create-checkpoint, ret- rieve-checkpoint, genesis-state, checkpoint-monotinicity*$\}$, $\mathtt{Exp}_\pi^X(\lambda, \rho)$ *returns* 1.

3.6 Security of the currency scheme

In this section, we establish the security requirements for the currency scheme constructed above, through a game-based approach. We chose game-based definitions over the UC framework because the former are intuitive and can be agreed upon by non-specialists (even non-cryptographers). This is essential as a bridge between theory and applications. Further, UC is best suited for small primitives whose ideal functionalities may still have a clean description, which is certainly not the case in the context of cryptocurrencies.

We define a comprehensive adversarial model to accommodate a wide range of adversarial capabilities. Then we define security requirements for the functionality of the proposed currency scheme. Anonymity aspects, although related to security, are discussed in a separate chapter (Chapter 4) as they are the main focus of this study.

3.6.1 Adversarial Model

We consider several parameters with an aim to define the adversarial model in depth. These parameters represent various levels of adversary capabilities, in terms of knowledge and power. Adversarial knowledge includes the extent of adversary's knowledge of public/secret keys, transaction values, metadata and transactions (ψ). On the other hand, adversarial power is represented by the ability to view and manipulate the state (δ), to influence state initialisation in the experimental setup (α) and to cause minting to fail during the execution of the game (β). These symbols are used with different subscripts denoting which entity is being referred to, as summarised in Table 3.4. These values scale from the least capability (0) to the strongest on the part of the adversary, ranging from passive to adaptive adversaries in game based experiments. Later, the model is further augmented by considering the assigned objective of the game.

The adversary's level of knowledge ψ is modelled in the following manner. When any knowledge parameter has a value of 0, the corresponding entity of that parameter is considered to be hidden from the adversary. We assume that the adversary has oracle access to those hidden entities through opaque handles, 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. At this stage, the adversary is not allowed to create or mint any transactions involving those entities. For certain parameters, the values beyond 0 have a special meaning. For the parameter ψ_t, the public part of the transaction t_p is revealed to the adversary when $\psi_t = 1$. When $\psi_t = 2$, the secret part t_s is revealed as well if any, and with $\psi_t = 3$, the randomness used to generate the transaction 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

Table 3.4: Parameters of the adversarial model

Paramet-er value	Adversarial knowledge					Adversarial power		
	Sender/ Recipient public keys ψ_{pk_s}/ψ_{sk_s}	Sender/ Recipient secret keys ψ_{pk_r}/ψ_{sk_r}	Transact-ion value ψ_v	Transact-ion Meta-data ψ_m	Transact-ion ψ_t	State manipul-ation δ	State initialis-ation α	Cause mint to fail β
0	Hidden	Hidden	Hidden	Hidden	Hidden	Hidden	Hidden randomness honest Init (HIDH)	Not allowed
1	Hidden but revealed at the end	Hidden but revealed at the end	Hidden but revealed at the end	Hidden but revealed at the end	t_p is revealed	Can view the state	Public randomness honest Init (PUBH)	Allowed
2	Access public keys through oracle	Access secret keys through oracle	Chosen by Oracle and known	Chosen by oracle and known	t_s is revealed	Can manipulate the state	Public randomness adversarial Init (PUBA)	-
3	Adversary chooses the identity, the oracle creates addresses	Adversary chooses the randomness, the oracle creates addresses	Adversary chooses ,the values	Adversary chooses metadata	Random-ness 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	-	-	

transaction when $\psi_t = 5$. For other ψ parameters, with a value of 2, relevant information is disclosed to the adversary throughout the game in real time via appropriate oracle access, but in a read-only manner. For values greater than 2, the adversary begins to be granted increasing levels of manipulative access to the internal protocol data referenced by the parameter as explained in Table 3.4.

With this parameterisation, we can capture a wide range of adversaries ranging from passive (with all parameters equal to zero) to static (with δ, $\beta \leq 1$) and adaptive adversaries (with parameter values greater than 1), thus enabling cryptographic security to be apprehended in a meaningful manner. It should be noted that some parameters such as ψ and β may not be useful when merely defining functional security, yet we define them here for completeness, and we utilise them when analysing the anonymity aspects in Chapter 4.

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 3.5). These include $\mathcal{O}_{addr}$ for creating addresses based on a given identity and randomness, $\mathcal{O}_{hidaddr}$ for creating hidden addresses, $\mathcal{O}_{txn}$ for creating transactions and $\mathcal{O}_{mint}$ for minting. Another oracle is defined to generate or alter hidden metadata ($\mathcal{O}_{hidMdata}$). Based on the received parameters, these oracles source required inputs via appropriate function calls to perform relevant functions or may just be disabled altogether, e.g. when $\psi = 0$.

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 lists to store hidden addresses, transactions and metadata. In order to cater for the addresses created with different adversarial inputs, the oracle keeps track of different groups of addresses in $A_{\mathcal{O}_{jk}}$ with binary values j and k, and a value of 0 representing adversarial identity and adversarial randomness, respectively. $\mathcal{O}_{mint}$ sets the flag $f_{\mathcal{O}} = 1$ globally, if a minting operation fails, in which case the adversary loses the game, unless β=1. The adversary has access to all available oracles, unless specifically mentioned with a specific subscript in the games. Table 3.5 summarises the variables used by the oracles.

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

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

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

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

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

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

Figure 3.5: Oracle functions

Table 3.5: A summary of oracle variables

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

Additionally, we also define another set of helper functions to be used in the security games as given in Figure 3.6 to improve the clarity of the games. The `SetupState` function performs the state initialisation based on γ, whereas the `RunAdversary` function executes an instance of the adversary $\mathcal{A}$ denoted by

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

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

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

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

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

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

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

Figure 3.6: Helper functions

different subscripts based on δ, in running security games. `LookupPubAddr` and `LookupSecAddr` functions are used to obtain public keys and private keys from hidden addresses. In addition, `LookupPubTxn` outputs the t_p corresponding to a hidden transaction when $\psi_t = 0$. `GenerateTxnVals` function is used when $\psi_v \in \{0, 1\}$, to generate required input and output transaction values, based on the the maximum transaction value given by the adversary. Further, `LookupMdata` function is used to reveal hidden metadata when $\psi_m \in \{0, 1\}$.

3.6.2 Security Properties

First, we define a set of security properties to ensure the functional security of the proposed cryptocurrency scheme. These are defined by means of game-based experiments around several attributes; *Unforgeability*, *Transaction binding property*, *Spendability*, *Balance property*, *Descendency* and *Anonymity*. Each attribute is treated with respect to attacker's goals and we construct appropriate games to model adversarial behaviour as per the model presented above.

Unforgeability

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. We define a security game to model this property as listed in Figure 3.7.

Attacker's Goal : Attacker's goal here is to create a valid transaction using funds attached to a payment address with hidden secret keys.

Game : In this game, the initial state is setup according to the input parameters and the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs a transaction (t_p, t_s) and the current state $p_{\mathcal{O}}$ based on the capabilities defined by the parameters δ and α. The challenger verifies whether the given state is valid. Subsequently, the challenger extracts the public addresses of the senders from the given transaction and performs a check to see if those addresses were created by the oracle (i.e. to ensure that the adversary does not have the knowledge of any of the secret keys). Further, the challenger also checks whether the transaction was created by the oracle and also whether the transaction is valid. This experiment is listed in Figure 3.7.

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

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

Figure 3.7: Experiment for Unforgeability.

Transaction binding property

This property establishes that the secret part of a transaction t_s cannot be tampered with and ensures that t_s binds with a unique t_p, i.e. a given t_s cannot correspond to two different t_p's[1]. Figure 3.8 lists the corresponding game.

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

Figure 3.8: Experiment for Transaction binding property.

[1]Note that Bitcoin transactions do not have long-lived secret parts (only ephemeral randomness), but Mimblewimble transactions do.

Attacker's Goal : Produce a secret part of a transaction (t_s), which matches a public part of a transaction (t_p) created by the oracle, making the pair (t_p, t_s) a valid transaction.

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 3.8.

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

Spendability

The property of spendability guarantees that the funds associated with a payment address (a) cannot decrease unless the corresponding secret keys are known (Figure 3.9). This property ensures the integrity of fund balances and funds cannot be spent (hence no transaction can be made) except by the owner. The unforgeability property we discussed earlier ensures that a valid transaction cannot be created without the secret keys and hence the spendability property can be regarded as a weaker notion of the former. i.e.
$\texttt{Balance}_{\mathit{aft}}(a) < \texttt{Balance}_{\mathit{bef}}(a)$ only if secret key of a (a_{sk}) is known $\forall a$

Attacker's Goal : Evolve the state such that the balance of a payment address created by the oracle decreases after evolving the state.

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

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

Figure 3.9: Experiment for Spendability.

Game : After the initial setup, the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ 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 3.9). Then the adversary evolves the state from that point onwards and the oracle does not create any new addresses or transactions during that period. The adversary has access to the minting oracle only. Subsequently, the adversary outputs the evolved state p_O together with a public key which they think satisfies the winning condition. The challenger then checks whether the adversary has minted any of the transactions in the initial list of unminted transactions and subsequently checks the balances of this address and compares with the corresponding initial balances stored (Figure 3.9).

Winning condition : Adversary wins if the closing balance of the given address is less than the starting balance.

Balance

This property requires that the fund balances of participants in a transaction are updated correctly upon being minted. Further, the balances of miners' addresses should also be updated correctly with relevant transaction fees and minted values (V_x). These goals can be summarised for a set of transactions T involved in a single minting operation as follows (the notation $v_{old}(s)$ here represents the input value corresponding to the sender s in V_{old} in a transaction t and others follow the same notation):

$$\sum_{t \in T}(\sum_{s \in S} v_{old}(s) - \sum_{r \in R} v_{new}(r)) + new\ units = \sum_{m \in R_m} v_x(m) \tag{3.1}$$

$$\texttt{Balance}_{bef}(a) + \sum_{t \in T}(v_{new}(a) - v_{old}(a)) = \texttt{Balance}_{aft}(a) \quad \forall a \in S, R \tag{3.2}$$

$$\texttt{Balance}_{bef}(a) + v_x(a) = \texttt{Balance}_{aft}(a) \quad \forall a \in \bar{R}_m \tag{3.3}$$

A single experiment is defined to capture all three properties in (Figure 3.10).

Attacker's goal : Create a transaction (t_p, t_s) so that the closing balance of a payment address does not satisfy the equations 3.1, 3.2 and 3.3 above.

Game : In this game, the adversary $A = (A_1, A_2, A_3)$ outputs a tuple of sender addresses $\bar{S}$, a tuple of recipient addresses $\bar{R}$, a tuple of miner addresses $\bar{R}_m$ together with the current state $p_{\mathcal{O}}$. The challenger records the balances of all addresses in the three groups of addresses and the minting history of the oracle $M_{\mathcal{O}}$. Then, the state evolves and the adversary outputs a set of transactions $\{(t_p, t_s)\}$ and the updated state $p_{\mathcal{O}}$. The challenger then records the new minting history M_2 and checks whether only one mint operation has taken place between M_1 and M_2, and also checks whether the minted transactions corresponds to the transactions returned by the adversary. In addition, another check is performed to see if the sender and recipient addresses involved in all transactions are the same as the sender and recipient addresses returned by the adversary. If any of these checks fails, adversary loses the game. For each transaction returned by the adversary, V_{old} and V_{new} values are recorded separately with the corresponding addresses. In addition, V_x values are also recorded with the miners' address details. Finally, the challenger records respective balances of all involved addresses and checks whether above conditions are satisfied for all the addresses.

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

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

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

Figure 3.10: Experiment for the Balance property.

Figure 3.10 lists the corresponding game ($\texttt{Exp}^{balance}_{\pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}$) which demonstrates this property.

Indemnification

The indemnification property requires that fund balances associated with the payment addresses that are not involved in a transaction should remain unchanged. We define this experiment in Figure 3.11.

Attacker's goal : Create a transaction such that there exists a payment address created by Oracle, which is not involved in the transaction, but its balance changes.

Game : In this game, the adversary $A = (A_1, A_2, A_3)$ outputs the current state first. The challenger records the balances of all addresses in $A_{\mathcal{O}}$ in B_{bef} and also records the mint history M_1. Then, adversary evolves the state and outputs a set of transactions $\{(t_p, t_s)\}$ together with the current state $p_{\mathcal{O}}$ and the challenger ensures that there has been only one mint operation since the previous state, and also whether the set of transactions corresponding to that mint operation matches the transactions returned by the adversary. Then the challenger records the sender and recipient addresses corresponding to the given transactions S and R. Subsequently, he checks the closing balances of all addresses in $A_{\mathcal{O}}$ and ensures none of these addresses are in S or R (Figure 3.11).

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

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

Figure 3.11: Experiment for Indemnification.

Positivity

This property ensures that the fund balance corresponding to each payment address in the system is non-negative at all times. Figure 3.12 defines the corresponding experiment for this property.

Attacker's goal : Provide a valid payment address that has a negative balance.

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 3.12).

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

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

Figure 3.12: Experiment for Positivity.

Descendancy

This requires that an adversary should not be able to produce a valid system state, which does not descend from the genesis state. We define this property in Figure 3.13.

Attacker's goal : Produce a valid state which does not descend from the genesis state.

Game : In this game, adversary $A = (A_1, A_2)$ gives a state to the challenger. The challenger retrieves the checkpoint state of the given state and attempts to loop back to the genesis state by retrieving the checkpoint state iteratively (Figure 3.13).

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

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

Figure 3.13: Experiment for Descendancy.

Definition of security of the currency scheme

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

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

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

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

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

3.7 Discussion

In this chapter, we have formulated the foundation for our research study. In this connection, we have developed several building blocks, in aid of the construction of a common framework for modelling anonymity of decentralised systems such as cryptocurrencies.

We initiated this process by constructing an abstract scheme, which depicts the generic functionality of a cryptocurrency system. For this purpose, we considered the common entities and operations of such a system, while allowing for flexibility to cater for diverse implementations. This exercise by itself reveals the kind of complexity we face in such decentralised systems, thus providing an indication for the absence of a common methodology to model such functionality.

As the next step in this process, we analysed how the functional correctness of a currency scheme in the presence of honest functionality can be established in terms of the abstract scheme developed above. This analysis was carried out with respect to individual operations as well as the expected functionality. Accordingly, we constructed several experiments which, upon receiving honest and valid inputs, would return a truth value of '1', thus ensuring the integrity of our model.

Further, we have provided a comprehensive security analysis for the constructed scheme using a game-based approach. We developed an adversarial model, which is capable of addressing an extensive range of adversaries possessing various capabilities. These adversaries are modelled in terms of their knowledge about system entities and their powers to influence the system state. Hence, this model facilitates a large number of attacker scenarios that can provide a clear picture of security in a decentralised system. This also highlights the complexity in modelling security of such a system.

We also investigated the security aspects of this abstract scheme against several properties in this chapter. These properties are mostly in agreement with widely referenced security notions in cryptocurrency schemes. For example, the property of unforgeability can be linked to the double spending property analysed in currency protocols. In addition, new notions such as descendancy addresses the security requirements for the system state (e.g. normal blockchain growth in Bitcoin), which assures that the states preserve time ordering. The balance and positivity properties guarantee that monetary units are created in accordance with the implementation specifications to manage the money supply. As such, the security aspects of the proposed abstract protocol closely agree with real world protocols, and thus facilitate a sound foundation for our study.

The contributions presented in this chapter are included in our publication [5], and form the basis for the anonymity framework, which is the core of this research. We discuss the specifics of this framework and relevant theoretical constructs in the next chapter.

Chapter 4

Anonymity Framework

Having established a sound foundation in the previous chapter, here we present the details of how a solid framework can be built to model anonymity on those grounds. This composition results in a unified approach that allows for fine-grained anonymity notions that can be arranged in a hierarchical structure. These notions stem from the widely known concepts such as indistinguishability and unlinkability, and we define them formally as per the naming conventions in related contexts. This chapter also includes an analysis of the relationships among above notions through security reductions etc., that are interpreted in terms of a set of theorems, supported by corresponding proofs.

4.1 Introduction

As was evident from the literature, no formalised framework exists, that can be used to analyse anonymity in distributed environments such as cryptocurrencies [3, 5]. Hence, we build a novel anonymity framework from the fundamentals based on the abstract cryptocurrency scheme and the adversarial model we defined in

Chapter 3. A game-based approach is utilised for this purpose as opposed to the UC framework due to the complexity of the functionality.

The property of anonymity is often discussed with respect to the participants of a transaction in the context of cryptocurrencies in studies such as [7]. Accordingly, they discuss anonymity characteristics in terms of a wide range of anonymity notions with various interpretations (e.g. the notion of unlinkability has many different perceptions as discussed in Chapter 2). Although many such notions are referenced in the literature, a majority of them are not formally defined in a manner that is commonly applicable to diverse cryptocurrency schemes. Hence, we intend to formalise anonymity notions in the context of cryptocurrencies.

Due to the granularity of the adversarial capabilities considered, our framework yields in a plethora of anonymity notions, each corresponding to a unique attacker scenario. Undoubtedly, it is problematic to work with such a large number and hence it is vital that we categorise them and identify their relationships, which will then simplify the analysis process. Accordingly, we formulate our contributions in this chapter as outlined below.

4.2 Our contributions

The most significant contribution in this chapter is the construction of the Anonymity Framework, which is the core of our research study. This goal is accomplished through a single parameterised game, which is capable of modelling a wide range of anonymity aspects in terms of unique attacker scenarios.

Among other contributions, we present a formalisation of the anonymity notions resulting from this framework, in terms of parametric vectors, each consisting of relevant adversarial goal, knowledge and power. Further, we also compose a set of anonymity definitions similar to conventional cryptographic security notions such as IND-CPA etc., which is useful in analysing anonymity.

Moreover, we also formulate a set of theorems in this chapter, demonstrating the relationships among anonymity notions. These theorems facilitate the anonymity analysis process which is described in detail in Chapter 5, and hence carry an equal importance as the framework itself.

4.3 Anonymity framework

Initially, 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, which stem 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 [73].

We also define a weaker notion of anonymity, *unlinkability*, which is similar to indistinguishability, except that 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 the two transactions has the same value as another transaction of interest.

We define these anonymity notions around a set of entities in a generic currency scheme. These entities include senders, recipients, value and metadata, and can be categorised as topological and non-topological where the former group directly corresponds to the topology of the transaction graph of the scheme. Senders and recipients form the topological category whereas value and other transaction related metadata are categorised as non-topological entities, without having a direct relationship to the transaction graph. We parameterise different scenarios where an attacker can manipulate these entities at various levels and via diverse means.

4.3.1 Anonymity Game

We formulate the *Anonymity Game* using the helper functions and oracle functions defined in Chapter 3 (Figures 3.6 and 3.5 respectively). As explained in detail in section 3.6, the parameter ψ represents the adversary's access to knowledge of the entities related to a transaction. Value of each ψ_x parameter starts from 0 representing a hidden entity x and then increases with the adversary's knowledge. These particulars are summarised in Table 3.4 in Chapter 3.

Moreover, we also introduce another variable, ω, to represent test variable/s for each instance of the game. These variables decide how the transactions are created in the game for a given setting. We define $\omega = (\omega_s, \omega_r, \omega_v, \omega_m)$ with each $\omega_x \in \{0, 1\}$ indicating which entity (i.e. $\underline{s}$ender/$\underline{r}$ecipient/$\underline{v}$alue/$\underline{m}$etadata) is being tested in a given instance of the game; e.g. $\omega = (1000)$ with only $\omega_s = 1$ indicates that the test variable is the sender and thus two transactions are created in the game so that only the senders of the two transactions are different. As such, the adversarial inputs are crafted based on the ω and ψ parameters and transactions are created accordingly.

In order to facilitate the execution of the Anonymity game in a more transparent manner, we define several helper functions as outlined in Figure 4.1. The two main functions are `CheckAdvConditions`, which checks the adversarial conditions of inputs at the start of the game and the `RevealData` function, which reveals data to the adversary at the end of the execution of the game based on the parameter ψ. Moreover, we define additional functions to facilitate the above as well as the oracle functionality as shown in the same figure.

```
RevealData($t_p, \omega, \psi, A^*_{\mathcal{O}}, T^*_{\mathcal{O}}, T_{\mathcal{O}}, T'_{\mathcal{O}}, p_1$)
1. $(\psi_{pk_s}, \psi_{sk_s}, \psi_{pk_r}, \psi_{sk_r}, \psi_v, \psi_m, \psi_t) \leftarrow \psi$; $(\omega_s, \omega_r, \omega_v, \omega_m) \leftarrow \omega$
2. $t_p \leftarrow$ LookupPubTxn($t_p, T^*_{\mathcal{O}}$);  $t_s \leftarrow AA.\text{Lookup}(t_p, T_{\mathcal{O}})$;  $\rho_t \leftarrow AA.\text{Lookup}(t_p, T'_{\mathcal{O}})$
3. $S \leftarrow$ ExtractSenderPubAddr$_\pi(t_p, t_s, p_1)$;  $R \leftarrow$ ExtractRecipientPubAddr$_\pi(t_p, t_s, p_1)$
4. $V_{old} \leftarrow$ ExtractInputVal$_\pi(t_p, t_s, p_1)$;  $V_{new} \leftarrow$ ExtractOutputVal$_\pi(t_p, t_s, p_1)$
5. $m \leftarrow$ ExtractMetadata$_\pi(t_p, t_s, p_1)$
6. $U_s \leftarrow (S^{\psi_{pk_s}}, (\text{LookupSecAddr}(S, A^*_{\mathcal{O}}))^{\psi_{sk_s}})$;  $U_r \leftarrow (R^{\psi_{pk_r}}, (\text{LookupSecAddr}(R, A^*_{\mathcal{O}}))^{\psi_{r_{sk}}})$
7. $U_v \leftarrow ((V_{old}, V_{new})^{\psi_v})$; $U_m \leftarrow (m)^{\psi_m}$;  $U_t \leftarrow (t_p^{\psi_t}, t_s^{(\psi_t=2)}, \rho_t^{(\psi_t=3)})$
8. 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. $(\omega_s, \omega_r, \omega_v, \omega_m) \leftarrow \omega$; $(\psi_{pk_s}, \psi_{sk_s}, \psi_{pk_r}, \psi_{sk_r}, \psi_v, \psi_m, \psi_t) \leftarrow \psi$
2. if $((S_0, S_1 \not\subseteq AA.keys(A_{\mathcal{O}})) \vee (S_0, S_1 \not\subseteq \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}})))) \wedge (\psi_{sk_s}, \psi_{sk_s} < 4)$, return 0
3. if $(S_0, S_1 \cap AA.keys(A_{\mathcal{O}_{11}}) \neq \emptyset) \wedge ((\psi_{sk_s} > 1) \vee (\psi_{pk_s} > 1))$, return 0
4. if $(S_0, S_1 \cap \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}_{11}})) \neq \emptyset) \wedge ((\psi_{sk_s} > 2) \vee (\psi_{pk_s} > 2))$, return 0
5. if $(S_0, S_1 \cap \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}_{00}})) \neq \emptyset) \wedge ((\psi_{sk_s} < 3) \vee (\psi_{pk_s} < 3))$, return 0
6. if $(S_0, S_1 \cap \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}_{10}})) \neq \emptyset) \wedge (\psi_{sk_s} < 3)$, return 0
7. if $(S_0, S_1 \cap AA.keys(A_{\mathcal{O}_{01}}) \neq \emptyset) \wedge (\psi_{pk_s} < 3)$, return 0
8. if $(S_0, S_1 \cap A^*_{\mathcal{O}} \neq \emptyset) \wedge ((\psi_{sk_s} > 2) \wedge (\psi_{pk_s} > 2))$, return 0
9. if $((R_0, R_1 \not\subseteq AA.keys(A_{\mathcal{O}})) \vee (R_0, R_1 \not\subseteq \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}})))) \wedge (\psi_{sk_r}, \psi_{sk_r} < 4)$, return 0
10. if $(R_0, R_1 \cap AA.keys(A_{\mathcal{O}_{11}}) \neq emptyset) \wedge ((\psi_{sk_r} > 1) \vee (\psi_{pk_r} > 1))$, return 0
11. if $(R_0, R_1 \cap \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}_{11}})) \neq \emptyset) \wedge ((\psi_{sk_r} > 2) \vee (\psi_{pk_r} > 2))$, return 0
12. if $(R_0, R_1 \cap \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}_{00}})) \neq \emptyset) \wedge ((\psi_{sk_r} < 3) \vee (\psi_{pk_r} < 3))$, return 0
13. if $(R_0, R_1 \cap \text{LookupSecAddr}(AA.keys(A_{\mathcal{O}_{10}})) \neq \emptyset) \wedge (\psi_{sk_r} < 3)$, return 0
14. if $(R_0, R_1 \cap AA.keys(A_{\mathcal{O}_{01}}) \neq \emptyset) \wedge (\psi_{pk_r} < 3)$, return 0
15. if $(R_0, R_1 \cap A^*_{\mathcal{O}} \neq \emptyset) \wedge ((\psi_{sk_r} > 2) \wedge (\psi_{pk_r} > 2))$, return 0
16. if $(\psi_m < 2) \wedge \neg(m \in D^*_{\mathcal{O}})$, return 0
17. return  1
```

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

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

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

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

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

Figure 4.1: Helper functions for the Anonymity game

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

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

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

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

Figure 4.2: Additional oracle functions

Further, we utilise several oracle functions to carry out the activities in the game based on adversarial capabilities as listed in Figure 4.2 (Note that $\mathcal{O}_{mint}$ was already defined in Chapter 3, yet it is included again here for completeness). The variables used by these oracle functions to keep track of oracle-generated data are listed in Table 4.1.

Table 4.1: 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 (honest) 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

We now define a common game to capture all possible attacker scenarios through above parametrisation. Figure 4.3 illustrates this game.

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

Figure 4.3: Anonymity Game

Execution of the Game The game is initialised by the `SetupState` function (defined in Figure 3.6) based on the parameter α to model various possibilities in setting up the initial state (step 3 in Figure 4.3); i.e. an honest setup with public or hidden randomness ($\alpha = 1$ or 2 respectively), or an adversarial setup with public/hidden randomness ($\alpha = 2$ or 3 respectively) as summarised in Table 3.4. Here we use '$\langle$*condition*$\rangle$' notation to check whether the resulting state is valid. As summarised in Table 3.1, if the condition inside the angle brackets is false, then the game terminates and the adversary loses the game.

Adversarial inputs are given in step 4 via the `RunAdversary` function, based on the test variables (ω), the adversary's knowledge (ψ). The details of how these inputs differ according to ψ_x parameters are given in Table 4.2. In addition, a list of unminted transactions T and miners' addresses R_m also form part of the inputs. The `CheckAdvConditions` function in step 7 checks the validity of the inputs as per the parametrisation of the given instance of the game. The variable $\mathcal{A}_{jk}$, which is listed as an input to this function represents specific associative arrays maintained by the oracle to store address data where $j, k \in \{0, 1\}$ (Table 4.1).

The instance $\psi_t = 5$ is a special case where the adversary creates the two challenge transactions of interest with full control and the challenger randomly mints one of them. A separate check is performed in this case on the transaction inputs t_0, t_1 in steps 10 and 11 to ensure that they are both separately mintable. We use the notation '$\langle \texttt{IsMintable}_\pi(\{t_{p_1}\} \cup T, p_\mathcal{O})^{\bar{\beta}} \rangle$' to represent this check. In this case, when β= 0 (i.e. failed mint operations are allowed), $\bar{\beta} = 1$ and hence the condition is met if $\texttt{IsMintable}()^1 = 1$. Conversely, when $\beta = 1$, $\bar{\beta} = 0$ and hence $\texttt{IsMintable}()^0 = 1$ always and hence the condition is satisfied. In all other cases, the challenger creates two transactions based on the inputs and mints one of them randomly.

If all inputs are validated as above, the adversary continues to evolve the system state through appropriate oracle queries. 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).

In step 17, a set of data is revealed to the adversary through the `RevealData` function based on the ψ parameter, after which the adversary is restricted from making further transactions involving revealed entities. This interpretation closely aligns with the forward security concept discussed in relation to cryptographic protocols, and hence able to model an extended scope of anonymity.

Finally, the adversary 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 $\beta \vee (f_\mathcal{O} \neq 1)$ (i.e. whether failed mint operations are allowed). The adversary wins the game if the guess is correct.

The extent of anonymity is defined based on the adversary's advantage,

Table 4.2: Nature of adversarial inputs

Param-eter value	Adversarial knowledge				
	Sender public/private 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
0	S_0, S_1: handles to hidden public/secret sender addresses created by oracle	R_0, R_1: handles to hidden public/secret recipient addresses created by oracle	V_{old}, V_{new}: maximum values for input/output transaction values, actual values chosen by oracle	m_0, m_1: handles to hidden metadata chosen by the oracle	t_0, t_1: empty, ρ_0,ρ_1: empty
1	S_0, S_1: handles to hidden public/secret sender addresses created by oracle, actual data revealed at the end	R_0, R_1: handles to hidden public/secret recipient addresses created by oracle, actual data revealed at the end	V_{old}, V_{new}: maximum values for input/output transaction values, actual data chosen by oracle,revealed at the end	m_0, m_1: handles to hidden metadata chosen by oracle, actual data revealed at the end	t_0, t_1: empty, ρ_0,ρ_1: empty, t_b created by oracle, t_{p_b} revealed at the end
2	S_0, S_1: public/secret sender addresses created by oracle, known to adversary throughout the game	R_0, R_1: public/secret recipient addresses created by oracle, known to adversary	V_{old}, V_{new}: chosen by Oracle and known to adversary	m_0, m_1: chosen by oracle and known to adversary	t_0, t_1: empty, ρ_0,ρ_1: empty, t_b created by oracle, t_{s_b} revealed at the end
3	S_0, S_1: public/secret sender addresses created by oracle, adversarial identity with ψ_{pk_s} = 3, adversarial randomness with ψ_{sk_s} = 3, known to adversary	R_0, R_1: public/secret recipient addresses created by oracle, adversarial identity with ψ_{pk_s} = 3, adversarial randomness with ψ_{sk_s} = 3, known to adversary	V_{old}, V_{new}: chosen by adversary	m_0, m_1: chosen by adversary	t_0, t_1: empty, ρ_0,ρ_1: empty, t_b created by oracle,randomness of coins revealed to adversary at the end
4	S_0, S_1: public/private addresses created by adversary	R_0, R_1: public/private addresses created by adversary	-	-	t_0, t_1: empty, ρ_0,ρ_1 : random-ness of coins, chosen by adversary, t_b created by oracle, randomness ρ_b
5	-	-	-	-	t_0, t_1: transactions created by adversary, ρ_0,ρ_1: empty

$\mathbf{Adv}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}$, in winning the game in a given setting defined by the relevant parametrisation. A system is secure in a given anonymity notion if a PPT adversary $\mathcal{A}$ has a negligible advantage of winning the game. i.e.

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

Unsurprisingly, there are over 10,000,000 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 formulate a set of anonymity notions while linking them to the relevant attributes discussed in the literature wherever applicable.

4.3.2 Anonymity notions

As previously mentioned, different combinations of the parameters in the *Anonymity Game* yield 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. Therein, we provide formal definitions for the same based on relevant parameter vectors.

We define each notion in terms of a unique adversary, based on the adversary's goal, knowledge and power as GOAL-KNOWL-POWER, which is also represented as a unique parameter vector ω-ψ-(δ, α, β). The goals are formulated with the emphasis on the anonymity aspect (i.e. IND or ULK) and the respective entity or entities of focus, as represented by the parameter ω. For example, if the adversary's goal is to distinguish between two senders, the corresponding goal is termed as S-IND and V-ULK represents the security goal of unlinkability of two transactions bearing the same value. In addition, if all entities are jointly being tested for indistinguishability, then we use ALL-IND to represent that objective. A special scenario is termed NIL-IND where two semantically identical transactions are created during the game, which differ only in the randomness involved and the adversary's goal is to distinguish between the two.

The adversarial capability is represented in terms of the adversary's knowledge and power. The highest level of power is named as FULL power with the ability to manipulate the state initialisation and the state ($\alpha = 3, \delta = 2$), and to make minting to fail ($\beta = 1$). We define an ACTIVE adversary whose power differs from FULL power only by their inability to cause minting to fail (i.e. $\beta = 0$ for ACTIVE power). On the other hand, VIEW power further restricts the adversary by allowing only view access to the system state in an honest state initialisation setting (i.e. $\delta = 1, \alpha = 1, \beta = 0$). The least powerful adversary has no power to view nor influence the state and hence we label as NIL power (i.e. $\delta = 0, \alpha = 0, \beta = 0$).

With respect to the adversarial knowledge, we name the FULL knowledge when the adversary has full control over all entities involved in a transaction (i.e. all knowledge parameters are at their maximum values). Conversely, NIL knowledge represents an adversary without any knowledge of the transaction entities with all knowledge parameters set to value 0. Other intermediate knowledge levels are named according to the restrictions on access to a particular entity. For example, PUBS knowledge represents an adversary having access to the public keys of senders and full knowledge of recipients, value, metadata and the transaction, whereas NILS corresponds to an adversary having no knowledge about the senders (same explanation holds for PUBR and NILR with respect to recipients). PUBSRV on the other hand, represents the public knowledge about the senders and recipients as before, together with the knowledge about the value while having the adversarial control over the metadata and the transaction. PUBSR-NILV is similar to PUBSRV except that the adversary has no knowledge of the value in the former. Likewise, PUBSR-NILM models no access to metadata, despite having the knowledge of public keys of senders and recipients, and adversarial value and transaction. NILS-PUBT represents an adversary having no knowledge of the senders but with public knowledge of the transaction, together with adversarial recipients, value and metadata etc.

Accordingly, the highest level of anonymity modelled by the Anonymity game is the notion ALL-IND-FULL-FULL which represents a fully adaptive adversary with a goal to distinguish between two totally different transactions. Conversely, the weakest is the notion of NIL-IND-NIL-NIL, which models a passive adversary whose goal is to distinguish between two transactions which differ only in the randomness. Accordingly, Table 4.3 lists some useful anonymity notions and their corresponding parameter vectors. We provide formal definitions for the same in the following section.

Table 4.3: Some useful anonymity notions

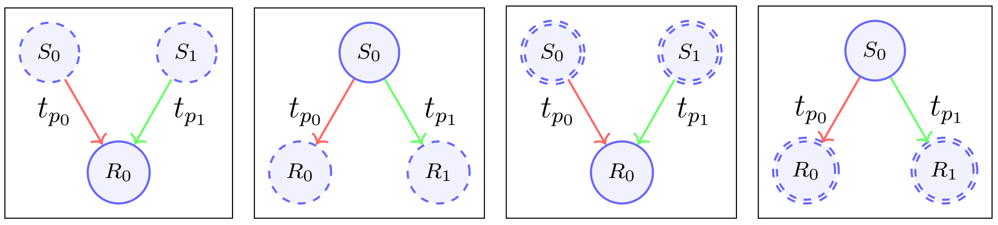
Goal	**Adversarial Knowledge**	**Adversarial Power**	**Parameter vector**
ALL-IND	FULL	FULL	$(1_s 1_r 1_v 1_m)_\omega$-$((4,4)_s,(4,4)_r,3_v,3_m,5_t)_\psi$-$(2_\delta,3_\alpha,1_\beta)$
S-IND	PUBS	ACTIVE	$(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s,(4,4)_r,3_v,3_m,5_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
S-ULK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

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 can be built using publicly available data related to senders and recipients [71, 78]. 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 4.4).

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 4.4(a) illustrates this scenario. In the anonymity game, only the public keys of the senders will be known to the adversary with $\psi_{pk_s} = 3$ and $\psi_{sk_s} = 0$ with same transaction values and other metadata, and the challenger will create two transactions t_{p_0} and t_{p_1} with same value and metadata. Based on the chosen bit b, the challenger mints the transaction t_{p_b} and the adversary gets to see the data related to the minted transaction, based on ψ_t and has to guess the challenger's choice. The knowledge of recipient addresses can vary based on ψ_{pk_r} and ψ_{sk_r}.

We can see that the game represents the strongest attacker scenario when the recipient addresses are fully controlled by the adversary in a setting with an adversarial hidden state initialisation and the ability to manipulate the state, as well as with the highest knowledge of the transaction (i.e. $\psi_t = 5$). However, having β=1 enables the adversary to craft transactions in a manner so that



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

Figure 4.4: Topological anonymity notions
(Dashed outline: addresses with hidden secret keys, double-dashed outline: addresses with hidden public/private keys, Solid outline: both keys known)

failed mint operations can be used to learn about account balances etc., thus revealing the transaction graph, in which case winning the game is trivial. For example, if account balances are known (without any information about corresponding addresses), the adversary could initiate a set of transactions with specially chosen values so that a subset of the challenge transaction minting fails due to insufficient balances to obtain additional information about the addresses based on their initial and ending balances. This is possible in a real system since the adversary could keep track of the transaction flow through the handles to respective addresses even if the senders and recipients are hidden.

Accordingly, the strongest achievable 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 against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameter vector $(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s, (4,4)_r,\ 3_v, 3_m,\ 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible. i.e.*

$$\mathbf{Adv}^{S-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \ \textit{is negligible in } \lambda.$$

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 4.4(c) illustrates, the adversary has to guess the correct transaction as with S-IND scenario, but without knowing either public/private key of the senders. i.e. Senders in this case are hidden with $\psi_{pk_s}, \psi_{sk_s} = 0$ and the adversary has oracle access to these addresses through respective handles (refer to section 3.6.1 in Chapter 3 for further details). The strongest notion in this sense is given by S-ULK-NILS-ACTIVE with the parameter vector "$(1_s0_r0_v0_m)_\omega$-$((0,0)_s,(4,4)_r,3_v,3_m,5_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$"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* $(1_s0_r0_v0_m)_\omega$-$((0,0)_s,(4,4)_r,3_v,\ 3_m,5_t)_\psi$-$(2_\delta,$ $3_\alpha,0_\beta)$ *is negligible. i.e.*

$$\mathbf{Adv}^{S-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

Recipient Indistinguishability (R-IND): This notion is similar to sender indistinguishability, except with recipient addresses. Hence, it is defined to be one's inability to distinguish the correct recipient out of two given recipients in a given situation. As shown in the Figure 4.4(b), public keys of the recipients ($\psi_{pk_r} = 3, \psi_{sk_r} = 0$) are known and the senders could be hidden or known as per the parameters ψ_{pk_s} and ψ_{sk_s}. The two transactions t_{p_0} and t_{p_1} both carry the same sender, values and metadata, yet two different recipients. The adversary needs to guess which transaction out of t_{p_0} and t_{p_1} was minted. The strongest adversarial scenario in this case is R-IND-PUBR-ACTIVE, denoted as "$(0_s1_r0_v0_m)_\omega$-$((4,4)_s,(3,0)_r,3_v,3_m,5_t)_\psi$-$(2_\delta,0_\alpha,0_\beta)$". 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.*

$$\mathbf{Adv}^{R-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \textit{ is negligible in } \lambda.$$

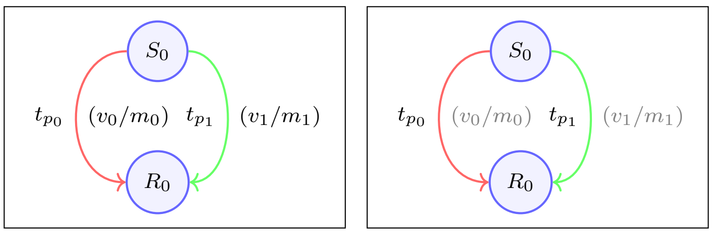
Recipient Unlinkability (R-ULK): This property is referred to as the inability to link a transaction to the correct recipient. Figure 4.4(d) shows the basic setup for this game where the adversary needs to guess the correct transaction out of the two options t_{p_0} and t_{p_1}, without any knowledge about the corresponding recipients, i.e. ψ_{pk_r},$\psi_{sk_r} = 0$. The strongest notion in this setting is represented as R-ULK-NILR-ACTIVE given by the vector "$(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s,(0,0)_r,3_v,3_m,5_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$" and the formal definition is given below.

Definition 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. i.e.*

$$\mathbf{Adv}^{R-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \textit{ is negligible in } \lambda.$$

Non-topological Entities

As opposed to topological entities, non-topological entities such as value and metadata in a currency scheme do not directly affect the structure of the transaction graph. However, if made public, these entities also could hinder the anonymity of users. For example, metadata might include IP addresses, that identify the sender in the real world. Similarly, transaction values can be relatively unique and can potentially be linked with other information, which might reveal corresponding identities. One such instance is where one needs to know the recipient, whom a particular amount was paid to etc. Hence, these entities can also be regarded as equally important in determining the level of anonymity in a currency scheme. In this section, we provide formal definitions for major anonymity notions involving non-topological entities; value and metadata (Figure 4.5).



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

Figure 4.5: Non-topological Anonymity notions

Value Indistinguishability (V-IND): The notion of indistinguishability with respect to transaction values refers to the fact that it is impossible to distinguish the correct value from two given input/output values for a given transaction. In the game, the challenger creates two transactions t_{p_0} and t_{p_1}, with two different sets of values $(V_{\text{old}_0}, V_{\text{new}_0})$ and $(V_{\text{old}_1}, V_{\text{new}_1})$, while having other entities the same (Figure 4.5(a)). Note that the value of ψ_v applies to both V_{old} and V_{new} values. In this case, the adversary knows what the 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 and thus it is a trivial case. Hence the strongest notion in this scenario is given by V-IND-PUBSR-ACTIVE which is represented by the vector "$(0_s0_r1_v0_m)_\omega$-$((3,0)_s,(3,0)_r,3_v,3_m$

$, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$" with the following formal definition.

Definition 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.*

$$\mathbf{Adv}^{V-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

Value Unlinkability (V-ULK): We define the property of unlinkability related to transaction value as the inability correctly identify a transaction of a specific value from two possible hidden (unknown) values. In order to realise this scenario, failed minting operations have to be allowed in the game with the parameter β set to 1, as it would be impossible for the adversary to win the game otherwise. As $\psi_v = 0$, the adversary gives maximum values for V_{new} and V_{old} values from which the challenger generates corresponding values required for the transaction using the `GenerateTxnVals` helper function (Figure 3.6). Further, as in the case of V-IND, we restrict the knowledge of secret keys of senders/recipients as otherwise the transaction is trivial. As shown in Figure 4.5(b) in this context, the challenger creates two transactions t_{p_0} and t_{p_1} with hidden transaction values v_0 and v_1, respectively. The challenger then picks a bit b and mints the transaction t_{p_b} and the adversary makes a guess to identify the correct scenario. Accordingly, the parameters required to achieve the strongest level of anonymity notion V-ULK-PUBSR-NILV-FULL are "$(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s, (3,0)_r, 0_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$" 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.*

$$\mathbf{Adv}^{V-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

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

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

Definition 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.*

$$\mathbf{Adv}^{M-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

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 3.6). Accordingly, we have the corresponding notion M-ULK-NILM-ACTIVE parameterised by, "$(0_s 0_r 0_v\ 1_m)_\omega$-$((4,4)_s, (4,4)_r, 3_v, 0_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$" representing the strongest case in this sense. The formal definition follows.

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.*

$$\mathbf{Adv}^{M-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \textit{ is negligible in } \lambda.$$

Other useful anonymity notions

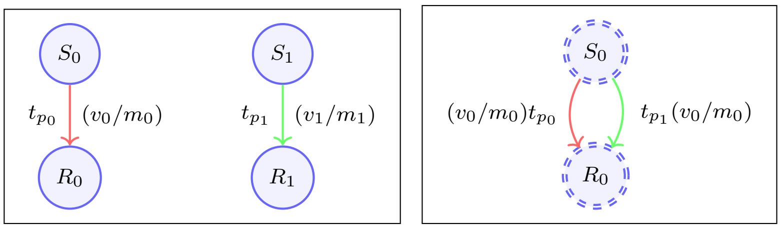
Further to above notions, we also formally define the strongest and weakest anonymity notions modelled in this framework as they are useful in benchmarking the anonymity landscape.

Strongest anonymity (ALL-IND) In this setting, the game models two senders and two recipients. The challenger creates two transactions t_{p_0} and t_{p_1} as before, but each transaction is created using distinct set of data; i.e. different sender, recipient, value and metadata (Figure 4.6 (a)). The strongest adversary in this scenario has the FULL knowledge and FULL power given by ALL-IND-FULL-FULL notion and parameterised by the vector $(1_s 1_r 1_v 1_m)_\omega$-$((4,4)_s,$

$(4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$. This setting models the highest level of anonymity achievable by a currency scheme and can be considered as "absolute fungibility".

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.*

$$\mathbf{Adv}^{ALL-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$



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

Figure 4.6: Strongest and weakest anonymity games

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

Definition 13. ***(S-IND-PUBST-ACTIVE)*** *A currency scheme* Π *is said to be anonymous with respect to Sender Indistinguishability against an adversary* $\mathcal{A}$*, if* $\mathcal{A}$*'s advantage of winning the Anonymity game defined by the parameters* $(1_s 0_r 0_v 0_m)_\omega$-$((3_{pk}, 0_{sk})_s, (4_{pk}, 4_{sk})_r, 3_v, 3_m,\ 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ *is negligible. i.e.*
$\mathbf{Adv}^{S-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}$ *is negligible in* λ.

Definition 14. *(**S-ULK-NILS-PUBT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Sender Unlinkability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(1_s 0_r 0_v 0_m)_\omega$-$((0,0)_s, (4,4)_r, 3_v, 3_m, 1_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible. i.e.*

$$\mathbf{Adv}^{S-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

Definition 15. *(**R-IND-PUBRT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Recipient Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s, (3,0)_r, 3_v, 3_m, 1_t)_\psi$-$(2_\delta, 0_\alpha, 0_\beta)$ is negligible. i.e.*

$$\mathbf{Adv}^{R-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

Definition 16. *(**R-ULK-NILR-PUBT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Recipient Unlinkability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s, (0,0)_r, 3_v, 3_m, 1_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible.*

$$\mathbf{Adv}^{R-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

Definition 17. *(**V-IND-PUBSRT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Value Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s, (3,0)_r, 2_v, 3_m, 1_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible. i.e.*

$$\mathbf{Adv}^{V-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

Definition 18. *(**V-ULK-PUBSRT-NILV-FULL**) A currency scheme Π is said to be anonymous with respect to Value Unlinkability against an adversary $\mathcal{A}$ under a hidden adversarial initialisation, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s, (3,0)_r, 0_v, 3_m, 1_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$ is negligible. i.e.*

$$\mathbf{Adv}^{V-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \text{ is negligible in } \lambda.$$

Further, we consider the weakest adversary that can be modelled in our game. In this case, the game produces two identical transactions as opposed to the strongest scenario above (Figure 4.6 (b)). These transactions differ only in their randomness and the adversary has to identify the correct transaction. Hence, the weakest adversary in this case is a NIL-NIL adversary with no knowledge nor power, which is a passive adversary. This means that even δ=0 , meaning that the scheme has a hidden private state, which however may not be the case for most cryptocurrency schemes. Yet, we provide the following formalisation for comparison.

Definition 19. ***(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.*

$$\mathbf{Adv}^{NIL-IND1}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \textit{ is negligible in } \lambda.$$

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

Definition 20. ***(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_δ*,* $0_\alpha, 0_\beta$*) is negligible. i.e.*

$$\mathbf{Adv}^{NIL-IND2}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} \textit{ is negligible in } \lambda.$$

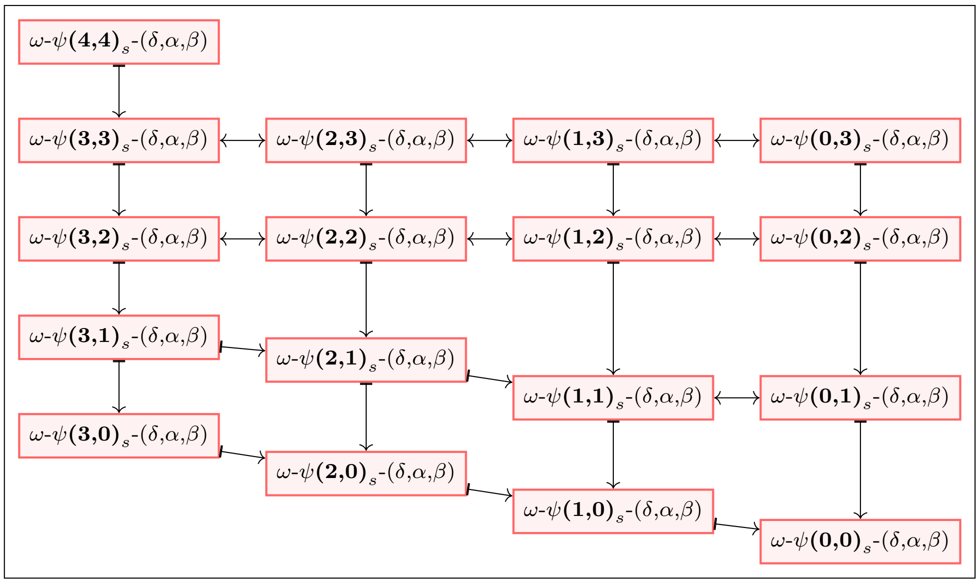

ω-ψ(4,4)$_s$-(δ,α,β)
ω-ψ(3,3)$_s$-(δ,α,β)
ω-ψ(2,3)$_s$-(δ,α,β)
ω-ψ(1,3)$_s$-(δ,α,β)
ω-ψ(0,3)$_s$-(δ,α,β)
ω-ψ(3,2)$_s$-(δ,α,β)
ω-ψ(2,2)$_s$-(δ,α,β)
ω-ψ(1,2)$_s$-(δ,α,β)
ω-ψ(0,2)$_s$-(δ,α,β)
ω-ψ(3,1)$_s$-(δ,α,β)
ω-ψ(2,1)$_s$-(δ,α,β)
ω-ψ(1,1)$_s$-(δ,α,β)
ω-ψ(0,1)$_s$-(δ,α,β)
ω-ψ(3,0)$_s$-(δ,α,β)
ω-ψ(2,0)$_s$-(δ,α,β)
ω-ψ(1,0)$_s$-(δ,α,β)
ω-ψ(0,0)$_s$-(δ,α,β)

Figure 4.7: Relationship of anonymity notions for different adversarial information on sender addresses $(\psi_{pk}, \psi_{sk})_s$.

4.3.3 Relationships between anonymity notions

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

It is interesting to note that as we vary different security parameters in our model, their relationships result in a non-trivial lattice as depicted in figures 4.7 and 4.8. These relations are interpreted as implications, equivalences and separations. The arrow "$\mapsto$" 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 formalise these relationships, we define a set of theorems that will simplify the process of assessing the anonymity of a currency scheme and we describe them below.

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

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

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

iii. given that Π *is secure in* $\omega\text{-}\psi((\mathbf{4,4})_s)\text{-}(\delta,\alpha,\beta)$*,* Π *is also secure in* $\omega\text{-}\psi((\mathbf{3,3})_s)\text{-}(\delta,\alpha,\beta)$*.*
i.e. $\omega\text{-}\psi((\mathbf{4,4})_s)\text{-}(\delta,\alpha,\beta) \rightarrow \omega\text{-}\psi((\mathbf{3,3})_s)\text{-}(\delta,\alpha,\beta)$

where $\omega \in \{1,0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$*,* $\psi_v, \psi_m \in \{0,1,2,3\}$*,* $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.7).*

Proof. Part i. Assume a currency scheme Π which is secure against the anonymity game defined by a given combination of ψ_{pk_r}, ψ_{sk_r}, ψ_v, ψ_m, ψ_t, α, β and with $(\psi_{pk_s}, \psi_{sk_s}) = (3,0)$. This means that senders' addresses are created with respect to identity information controlled by the adversary and senders' public keys are known throughout, yet secret keys are not known. Now consider a scenario with $(\psi_{pk_s}, \psi_{sk_s}) = (2,0)$, while keeping the other parameters fixed. This means that the adversary has access to the public keys of senders through the oracles and the addresses are honestly generated. When compared to the former case, the adversary has less control in the latter scenario. Hence, we can conclude that if

Π is secure against a more powerful adversary, then it is also secure against a less powerful adversary since the more powerful one has the ability to perfectly emulate the less powerful one. i.e. ω-$\psi((3,0)_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((2,0)_s)$-$(\delta,\alpha,\beta)$.

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

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

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

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

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

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

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

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

where $\omega \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.7).*

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. □

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

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

i.e. ω-$\psi((0,1)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((1,1)_s)$-$(\delta,\alpha,\beta)$

ii. *given that* Π *is secure in* ω-$\psi((0,2)_s)$-(δ,α,β)*, then* Π *is also secure in* ω-$\psi((1,2)_s)$-(δ,α,β)*,* ω-$\psi((2,2)_s)$-(δ,α,β) *and* ω-$\psi((3,2)_s)$-(δ,α,β)*, and vice versa.*

i.e. ω-$\psi((0,2)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((1,2)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((2,2)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((3,2)_s)$-$(\delta,\alpha,\beta)$

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

i.e. ω-$\psi((0,3)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((1,3)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((2,3)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((3,3)_s)$-$(\delta,\alpha,\beta)$

where $\omega \in \{1,0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$*,* $\psi_v, \psi_m \in \{0,1,2,3\}$*,* $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.7).*

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

Part ii. Similar to part i, $\psi_{sk_s} = 2$ in this case corresponds to the case where the addresses are honestly generated and secret keys are accessible by the adversary through the oracles during the game. This means that this case is the same irrespective of ψ_{sk_s} in $(0, 2)$, $(1, 2)$, $(2, 2)$ and $(3, 2)$ through the same line of argument as before. Hence the equivalence follows.

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

$\square$

Note that the Theorems 1 and 2 also hold for recipient addresses in a similar manner and hence we do not provide separate theorems for the recipients here.

Separations

Next, we look at several separations between anonymity notions with respect to sender addresses, which are also illustrated in Figure 4.7 (with the arrow $\mapsto$). A separation between two notions A and B given that $A \rightarrow B$, means that there exists a scheme which satisfies B, but not A. We can demonstrate such separations by providing a counterexample for $B \rightarrow A$, assuming that there exists at least one scheme which satisfies B (i.e. $B \nrightarrow A$). In this case, we can say that B is strictly weaker than A (i.e. $A \mapsto B$). Accordingly, we construct a set of theorems to prove these separations, assuming that there exists a scheme, which satisfies the weaker notion in each case. Corresponding proofs are supported by counterexamples.

Theorem 4. *For given combination of ω, δ, α, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t and β*

(with $\psi_t \neq 0$), assuming the existence of a scheme satisfying the weaker notion in each pair of notions, the following separations hold for the varying values of ψ_{pk_s}:

i. ω-$\psi((\mathbf{0,0})_s)$-(δ,α,β) *is strictly weaker than* ω-$\psi((\mathbf{1,0})_s)$-(δ,α,β), ω-$\psi((\mathbf{1,0})_s)$-(δ,α,β) *is strictly weaker than* ω-$\psi((\mathbf{2,0})_s)$-(δ,α,β) *and* ω-$\psi((\mathbf{2,0})_s)$-(δ,α,β) *is strictly weaker than* ω-$\psi((\mathbf{3,0})_s)$-(δ,α,β).

ii. ω-$\psi((\mathbf{1,1})_s)$-(δ,α,β) *is strictly weaker than* ω-$\psi((\mathbf{2,1})_s)$-(δ,α,β) *and* ω-$\psi((\mathbf{2,1})_s)$-(δ,α,β) *is strictly weaker than* ω-$\psi((\mathbf{3,1})_s)$-(δ,α,β).

iii. ω-$\psi((\mathbf{3,3})_s)$-(δ,α,β) *is strictly weaker than* ω-$\psi((\mathbf{4,4})_s)$-(δ,α,β).

where $\omega \in \{1,0\}^4$, $\psi_{pk_r}, \psi_{sk_r} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.7).*

Proof. Part i.

(a). Assume that there exists a currency scheme Π_0 which is secure in ω-$\psi((\mathbf{0,0})_s)$-(δ,α,β). 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_0' derived from Π_0 such that the transaction creation process involves encrypting the transaction details (i.e. all inputs to the transaction). The sender's public key S_{pk}, treated as a bit string, is used as the key for this symmetric encryption and the encrypted data is included in t_p as given below:

```
CreateTxn$_{\Pi_0'}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_0}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    $t_p' \leftarrow (t_p,$ Encrypt$_{S_{pk}}(R, V_{new}, \bar{S}, V_{old}, m))$
    return $(t_p', t_s)$
}
```

All other operations in Π_0' are same as those of Π_0 (and all other operations that take t_p as an input will only consider the t_p portion in the case where t_p is modified). This modified scheme Π_0' is secure when public keys of the sender are unknown (hidden) as Π_0 is secure in ω-$\psi(\mathbf{(0,0)}_s)$-(δ,α,β) and the adversary never knows the keys to decrypt extra transaction information stored in t_p. However, if the adversary gets to know the public keys of the sender (i.e. $(\psi_{pk_s},\psi_{sk_s}) = (1,0)$) at the end of the game, t_p' can be decrypted and information about the transaction will be leaked. Hence, the adversary has a non-negligible advantage over winning the Anonymity game based on leaked information and hence Π_0' is not secure in ω-$\psi(\mathbf{(1,0)}_s)$-(δ,α,β). As shown previously in Theorem 1, ω-$\psi(\mathbf{(1,0)}_s)$-(δ,α,β) implies ω-$\psi(\mathbf{(0,0)}_s)$-(δ,α,β). Hence, we conclude that the latter is strictly weaker than the former.

(b). Consider a scheme Π_1 which is secure in ω-$\psi(\mathbf{(1,0)}_s)$-(δ,α,β). In this case, the adversary gets to know the public keys of the sender at the end of the game, yet is unable to make any transactions with the revealed information. We now consider a modified scheme Π_1' with a revised version of the transaction creation functionality where the transaction details are revealed if the sender's public key is included in transaction metadata as follows.

```
CreateTxn$_{\Pi_1'}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_1}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    for all  $(a_{pk}, a_{sk}) \in \bar{S}$  do
      if  $a_{pk} = m$  then
        $t_p \leftarrow (t_p, (R, V_{new}, \bar{S}, V_{old}, m))$
    return  $(t_p, t_s)$
}
```

Other operations of the scheme are similar to those of Π_1.

Now, Π_1' is secure in ω-$\psi(\mathbf{(1,0)}_s)$-(δ,α,β) similar to Π_1 since the adversary is unable to create any new transactions with the sender's public keys since the public keys are only revealed in the end. However, with ω-$\psi(\mathbf{(2,0)}_s)$-(δ,α,β), adversary has the knowledge of sender's public keys throughout the game in which case, a transaction can be created by including the public key in metadata, causing the function to leak information about the sender set, so that the adversary can identify the senders directly. Hence, Π'_1 is not secure in this case. i.e. ω-$\psi((\mathbf{1,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta)$. Further, Theorem 1 shows that ω-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{1,0})_s)$-$(\delta,\alpha,\beta)$. Hence, the latter is strictly weaker than the former.

(c) Assume that a currency scheme Π_2 is secure in ω-$\psi((\mathbf{2,0})_s)$-(δ,α,β). In other words, Π_2 is secure in the case where public keys of the senders are known to the adversary throughout the game. This scheme is modified to construct a new scheme Π'_2 to have a publicly known special ID (ID_s) such that when this ID is linked with a sender's address, all transaction details are leaked in t_p in the transaction. Hence, the modified transaction creation process is as follows:

```
CreateTxn$_{\Pi'_2}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_1}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    for all $(a_{pk}, a_{sk}) \in \bar{S}$ do
      if ExtractID$(a_{pk}) = ID_s$ then
        $t_p \leftarrow (t_p, (R, V_{new}, \bar{S}, V_{old}, m))$
    return $(t_p, t_s)$
}
```

Other operations of the scheme are similar to those of Π_2.

Now, Π'_2 is secure in ω-$\psi((\mathbf{2,0})_s)$-(δ,α,β) as Π_2 is secure in the same since the adversary does not choose the ID and hence ID_s is used only with low probability.. With ω-$\psi((\mathbf{3,0})_s)$-(δ,α,β), the adversary can choose the identity for the sender's addresses in which case the special ID can be given to create a sender's address. This address can then be used as a sender in a transaction, which will reveal the transaction information. Hence the scheme is insecure in ω-$\psi((\mathbf{3,0})_s)$-(δ,α,β). However, ω-$\psi((\mathbf{3,0})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta)$ from Theorem 1. As ω-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,0})_s)$-$(\delta,\alpha,\beta)$ from above, we can conclude that the former is strictly weaker than the latter.

Part ii. This proof follows the same line of argument as part (i) with the only difference being senders' secret keys revealed in the end and hence counterexamples can be constructed in a similar manner.

Part iii. Assume that a scheme Π_3 is secure in ω-$\psi((\mathbf{3},\mathbf{3})_s)$-$(\delta,\alpha,\beta)$ with the sender's identity and randomness being adversarial, but the address is created honestly. Then, we define a new scheme Π'_3 having a special bit b, which indicates whether the addresses are honestly generated ($b = 1$) or adversarial ($b = 0$). We modify the address creation to indicate whether an address is honestly generated or adversarial, and also the transaction creation process such that when sender's addresses are fully adversarial, all transaction information are included in t_p. These modified functions are given below:

```
CreateAddr$_{\Pi'_3}$$(p, id; \rho)${
    $(a_{pk}, a_{sk}, t_p, t_s)$ ← CreateAddr$_{\Pi_3}$$(p, id; \rho)$
    return  $(a_{pk}, (a_{sk}, 1), t_p, t_s)$
}
```

```
CreateTxn$_{\Pi'_3}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_3}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    for all $a \in \bar{S}$ do
        $(a_{pk}, (a_{sk}, b)) \leftarrow a$
        if $b = 0$ then
            $t_p \leftarrow (t_p, (R, V_{new}, \bar{S}, V_{old}, m))$
        else
            $t_p \leftarrow (t_p, \emptyset)$
    return $(t_p, t_s)$
}
```

Now, Π'_3 is secure in ω-$\psi((\mathbf{3},\mathbf{3})_s)$-$(\delta,\alpha,\beta)$ since the scheme functions similar to Π_3 which is already secure in the above setting. However, the adversary can create an address for a sender themselves of the form $(a_{pk}, (a_{sk}, 0))$, which will cause `CreateTxn` to reveal the transaction details in t_p, thus giving the adversary a non-negligible advantage in the game. Hence, Π'_3 is not secure in ω-$\psi((\mathbf{4},\mathbf{4})_s)$-$(\delta,\alpha,\beta)$, which means that ω-$\psi((\mathbf{3},\mathbf{3})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{4},\mathbf{4})_s)$-$(\delta,\alpha,\beta)$. We already know that ω-$\psi((\mathbf{4},\mathbf{4})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{3},\mathbf{3})_s)$-$(\delta,\alpha,\beta)$. Thus we can conclude that the latter is strictly weaker than the former.

□

We now look at the separations between notions arising from varying the knowledge of secret keys of the senders while keeping other information fixed.

Theorem 5. *For given combination of ω, δ, α, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t and β (with $\psi_t \neq 0$), assuming the existence of a scheme satisfying the weaker notion in each pair of notions, the following separations hold for the varying values of ψ_{sk_s}:*

i. ω-$\psi((\mathbf{1,0})_s)$-(δ, α, β) *is strictly weaker than* ω-$\psi((\mathbf{1,1})_s)$-(δ, α, β).

ii. ω-$\psi((\mathbf{2,0})_s)$-(δ, α, β) *is strictly weaker than* ω-$\psi((\mathbf{2,1})_s)$-(δ, α, β), *and* ω-$\psi((\mathbf{2,1})_s)$-(δ, α, β) *is strictly weaker than* ω-$\psi((\mathbf{2,2})_s)$-(δ, α, β).

iii. ω-$\psi((\mathbf{3,0})_s)$-(δ, α, β) *is strictly weaker than* ω-$\psi((\mathbf{3,1})_s)$-(δ, α, β), ω-$\psi((\mathbf{3,1})_s)$-(δ, α, β) *is strictly weaker than* ω-$\psi((\mathbf{3,2})_s)$-(δ, α, β) *and* ω-$\psi((\mathbf{3,2})_s)$-(δ, α, β) *is strictly weaker than* ω-$\psi((\mathbf{3,3})_s)$-(δ, α, β).

where $\omega \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{1,2,3,4,5\}$, $\beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.7).*

Proof. Part i. This proof is similar to the proof of part (a) of Theorem 4 where the public key is replaced by the secret key. Thus, the modified scheme's transaction process is altered so that additional transaction details are encrypted using the sender's secret keys.

Part ii.
(a) The proof of ω-$\psi((\mathbf{2,0})_s)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi((\mathbf{2,1})_s)$-$(\delta, \alpha, \beta)$ is similar to the proof in part (i) above under the same argument.

(b) The proof of ω-$\psi((\mathbf{2,1})_s)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi((\mathbf{2,2})_s)$-$(\delta, \alpha, \beta)$ is similar to the proof in part (b) of Theorem 4, with the modification such that sender's secret key is included in metadata instead of the public key.

Part iii. The proofs of ω-$\psi((\mathbf{3,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,1})_s)$-$(\delta,\alpha,\beta)$, and ω-$\psi((\mathbf{3,1})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,2})_s)$-$(\delta,\alpha,\beta)$ are similar to the proofs in part (ii) above.

The proof of ω-$\psi((\mathbf{3,1})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,2})_s)$-$(\delta,\alpha,\beta)$ is similar to the proof in Part i (c) of Theorem 1. □

Note here that above separations may not hold in the same manner in all scenarios for recipients' addresses since the knowledge of senders' addresses may leak more information related to transactions compared to the knowledge of recipients. Hence, in the presence of the knowledge of senders, some parametric combinations with respect to recipients may be indifferent. However, there can be obvious separations such as $(0,0)_r \nrightarrow (4,4)_r$ based on a given context. Hence, we do not explicitly formalise such separations here with respect to recipient addresses.

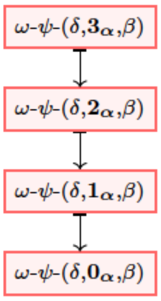
Next, we formulate the relationships between notions resulting from varying adversarial powers; state initialisation, state manipulation and ability to cause minting to fail. We consider implications and separations in this connection as illustrated in Figure 4.8.

Theorem 6. *For a currency scheme Π and for a given combination of ω, δ, ψ and β, the following holds for varying values of α:*

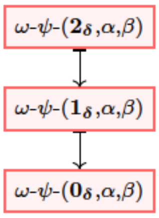
i. *given Π is secure in ω-ψ-$(\delta,\mathbf{3}_\alpha,\beta)$, then Π is also secure in ω-ψ-$(\delta,\mathbf{2}_\alpha,\beta)$. i.e. ω-ψ-$(\delta,\mathbf{3}_\alpha,\beta) \rightarrow \omega$-$\psi$-$(\delta,\mathbf{2}_\alpha,\beta)$*

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

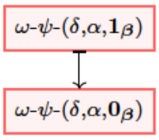
iii. *given Π is secure in ω-ψ-$(\delta,\mathbf{1}_\alpha,\beta)$, then Π is also secure in ω-ψ-$(\delta,\mathbf{0}_\alpha,\beta)$. i.e. ω-ψ-$(\delta,\mathbf{1}_\alpha,\beta) \rightarrow \omega$-$\psi$-$(\delta,\mathbf{0}_\alpha,\beta)$*


ω-ψ-$(\delta,\mathbf{3}_\alpha,\beta)$
ω-ψ-$(\delta,\mathbf{2}_\alpha,\beta)$
ω-ψ-$(\delta,\mathbf{1}_\alpha,\beta)$
ω-ψ-$(\delta,\mathbf{0}_\alpha,\beta)$

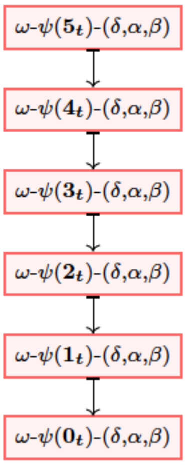
(d) Varying α (Theorems 6 and 12.)


ω-ψ-$(\mathbf{2}_\delta,\alpha,\beta)$
ω-ψ-$(\mathbf{1}_\delta,\alpha,\beta)$
ω-ψ-$(\mathbf{0}_\delta,\alpha,\beta)$

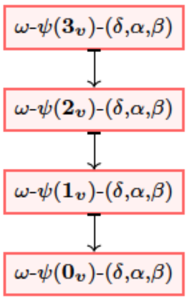
(a) Varying δ (Theorems 7 and 13.)


ω-ψ-$(\delta,\alpha,\mathbf{1}_\beta)$
ω-ψ-$(\delta,\alpha,\mathbf{0}_\beta)$

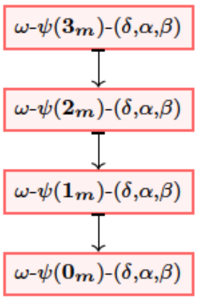
(c) Varying β (Theorem 8 and 14.)


ω-$\psi(\mathbf{5}_t)$-(δ,α,β)
ω-$\psi(\mathbf{4}_t)$-(δ,α,β)
ω-$\psi(\mathbf{3}_t)$-(δ,α,β)
ω-$\psi(\mathbf{2}_t)$-(δ,α,β)
ω-$\psi(\mathbf{1}_t)$-(δ,α,β)
ω-$\psi(\mathbf{0}_t)$-(δ,α,β)

(b) Varying ψ_t (Theorems 11 and 17.)


ω-$\psi(\mathbf{3}_v)$-(δ,α,β)
ω-$\psi(\mathbf{2}_v)$-(δ,α,β)
ω-$\psi(\mathbf{1}_v)$-(δ,α,β)
ω-$\psi(\mathbf{0}_v)$-(δ,α,β)

(e) Varying ψ_v (Theorems 9 and 15.)


ω-$\psi(\mathbf{3}_m)$-(δ,α,β)
ω-$\psi(\mathbf{2}_m)$-(δ,α,β)
ω-$\psi(\mathbf{1}_m)$-(δ,α,β)
ω-$\psi(\mathbf{0}_m)$-(δ,α,β)

(f) Varying ψ_m (Theorems 10 and 16).

Figure 4.8: Relationships among notions based on α, δ, ψ_v, ψ_m, ψ_t and β

.

where $\omega \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}$, $\beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.8(d)).*

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

Hence, the adversary in the latter case is less powerful than the former. Thus it follows that Π is also secure against ω-ψ-$(\delta, 2_\alpha, \beta)$ given that Π is secure against a more powerful adversary in ω-ψ-$(\delta, 3_\alpha, \beta)$.

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

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

Theorem 7. *For a currency scheme Π and for a given combination of ω, α, ψ and β, the following holds for varying values of δ :*

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

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

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{1, 2\}$ (Figure 4.8(a)).

Proof. $\delta = 2$ represents the strongest adversary with the capability to manipulate the state whereas the adversary is only able to view the state when $\delta = 1$ while keeping other capabilities fixed, hence representing a weaker adversary. Similarly with $\delta = 0$, the state is private and hence the adversary is the weakest in this respect, being unable to view the state. Using the same line of argument, if a scheme is secure against a given adversary, it is also secure in a weaker adversary, we can see that ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$ represents a more powerful attacker than ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ and also ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ is more powerful than ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$, hence it is clear that both implications are true. □

Theorem 8. *For a currency scheme* Π *and for a given combination of* ω*,* α *and* ψ *with* $\psi_t > 0$*, given that* Π *is secure in* ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta)$ *then* Π *is also secure in* ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta)$*.*
i.e. ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta) \rightarrow \omega$-$\psi$-$(\delta, \alpha, \mathbf{0}_\beta)$

where $\omega \in \{1, 0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$*,* $\psi_v, \psi_m \in \{0, 1, 2, 3\}$*,* $\psi_t \in \{0, 1, 2, 3, 4, 5\}$*,* $\beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 4.8(a)).*

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

Theorem 9. *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*, the following holds for varying values of* ψ_v*:*

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

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

iii. *given* Π *is secure in* ω-$\psi(\mathbf{1}_v)$-(δ,α,β)*, then* Π *is also secure in* ω-$\psi(\mathbf{0}_v)$-(δ,α,β)*.*

i.e. ω-$\psi(\mathbf{1}_v)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi(\mathbf{0}_v)$-$(\delta,\alpha,\beta)$

where $\omega \in \{1,0\}^4$, $\psi_{pk},\psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v,\psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}$, $\beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.8(e)).*

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

Theorem 10. *For a currency scheme* Π *and for a given combination of* ω*,* δ*,* α*,* $(\psi_{pk},\psi_{sk})_s$*,* ψ_v*,* ψ_t *and* β*, the following holds for varying values of* ψ_m*:*

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

i.e. ω-$\psi(\mathbf{3}_m)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi(\mathbf{2}_m)$-$(\delta,\alpha,\beta)$

ii. *given* Π *is secure in* ω-$\psi(\mathbf{2}_m)$-(δ,α,β)*, then* Π *is also secure in* ω-$\psi(\mathbf{1}_m)$-(δ,α,β)*.*

i.e. ω-$\psi(\mathbf{2}_m)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi(\mathbf{1}_m)$-$(\delta,\alpha,\beta)$

iii. *given* Π *is secure in* ω-$\psi(\mathbf{1}_m)$-(δ,α,β)*, then* Π *is also secure in* ω-$\psi(\mathbf{0}_m)$-(δ,α,β)*.*

i.e. ω-$\psi(\mathbf{1}_m)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi(\mathbf{0}_m)$-$(\delta,\alpha,\beta)$

where $\omega \in \{1,0\}^4$, $\psi_{pk},\psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v,\psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}$, $\beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.8(f)).*

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

Theorem 11. *For a currency scheme* Π *and for a given combination of* ω*,* δ*,* α*,* $(\psi_{pk}, \psi_{sk})_s$*,* $(\psi_{pk}, \psi_{sk})_r$ ψ_v*,* ψ_m *and* β*, the following holds for varying values of* ψ_t*;*

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

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

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

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

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

where $\omega \in \{1, 0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$*,* $\psi_v, \psi_m \in \{0, 1, 2, 3\}$*,* $\psi_t \in \{0, 1, 2, 3, 4, 5\}$*,* $\beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 4.8(b)).*

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

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

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

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

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

Next, we look at the separations that exist between the anonymity notions resulting in varying adversarial power by providing counterexamples.

Theorem 12. *For a currency scheme Π and for a given combination of ω, δ, ψ and β (with $\delta \in \{0,1\}$), following separations hold for varying values of α, assuming the existence of a scheme which is secure in the weaker notion in each pair of notions:*

i. ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$ *is strictly weaker than* ω-ψ-$(\delta, \mathbf{3}_\alpha, \beta)$.
i.e. ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, \mathbf{3}_\alpha, \beta)$

ii. ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$ *is strictly weaker than* ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$.
i.e. ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, \mathbf{2}_\alpha, \beta)$

iii. ω-ψ-$(\delta, \mathbf{0}_\alpha, \beta)$ *is strictly weaker than* ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$.
i.e. ω-ψ-$(\delta, \mathbf{0}_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, \mathbf{1}_\alpha, \beta)$

where $\omega \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}, \psi_t,\quad \beta \in \{0,1\}$ *and* $\delta \in \{0,1,2\}$ *(Figure 4.8(d)).*

Proof. Part i. The two notions here differ only based on whether the state initialisation is honest or adversarial. Assume that there exists a currency scheme Π which is secure in ω-ψ-$(\delta, 2_\alpha, \beta)$, where the state is initialised through an honest initialisation based on the randomness r chosen by the adversary. Consider a modified currency scheme Π', where the state is modified to represent the nature of initialisation as (p, b). Here, $b = 1$ represents an honest initialisation and with $b = 0$, the adversary initialises the state, in which case the transaction creation process leaks information in t_p. Modified operations are as follows:

```
Init_{Π'}($1^\lambda; \rho$){
    $p_0 \leftarrow$ Init_Π($1^\lambda; \rho$)
    return  $(p_0, 1)$
}
```

```
CreateTxn_{Π'}($R, V_{new}, \bar{S}, V_{old}, m, p; \rho$) {
    $(p, b) \leftarrow p$;   $(t_p, t_s) \leftarrow$ CreateTxn_Π($R, V_{new}, \bar{S}, V_{old}, m, p; \rho$)
    if  $b = 0$  then return  $((t_p, (R, V_{new}, \bar{S}, V_{old}, m)), t_s)$
    else return  $(t_p, t_s)$
}
```

All other operations of Π' are similar to those in Π. In the case of a ω-ψ-$(\delta, 2_\alpha, \beta)$ adversary, Π' is secure as the state is initialised honestly, resulting in $b = 1$ and hence the scheme executes the honest functionality, which operates similar to Π. Conversely with $\alpha = 3$, the adversary has the ability to perform the initialisation themselves, and hence can choose $b = 0$, which causes the `CreateTxn` to leak transaction information in t_p (here we assume $\delta < 2$). Hence, Π is insecure in ω-ψ-$(\delta, 3_\alpha, \beta)$, proving that the former notion is strictly weaker than the latter.

Part ii. The difference between the two cases of $\alpha = 1$ and $\alpha = 2$ is that in the former, the randomness is public whereas the latter involves adversarial randomness. Assume that there exists a currency scheme Π_0 which is secure in ω-ψ-$(\delta, 1_\alpha, \beta)$ (with $\delta < 2$), where the state is initialised through an honest initialisation based on public randomness r. We consider a modified scheme Π_0' where the adversary is able to set a bit b when the randomness is equal to zero (i.e. $r = 00..0$), in which case the state leaks all transaction information in t_p (similar to the construction in the proof of Theorem 4). Hence, the state initialisation and transaction creation in Π_0' are modified as follows:

```
$\texttt{Init}_{\Pi_0'}(1^\lambda; r)\{$
    $p_0 \leftarrow \texttt{Init}_{\Pi_0}(1^\lambda; r)$
    if  $r = 00..0$  then  $p_0 \leftarrow (p_0, 1)$
    else  $p_0 \leftarrow (p_0, 0)$
    return  $p_0$
}
```

```
$\texttt{CreateTxn}_{\Pi_0'}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(p, b) \leftarrow p$
    $(t_p, t_s) \leftarrow \texttt{CreateTxn}_{\Pi_0}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    if  $b = 1$  then  $t_p \leftarrow (t_p, (R, V_{new}, \bar{S}, V_{old}, m))$
    return  $(t_p, t_s)$
}
```

Now, with the honest initialisation ($r \neq 0..0$), the state will be initialised through $\texttt{Init}_{\Pi_0}$ with the bit 0, thus creating an honest setup, i.e. $\alpha = 1$. Hence, Π_0' is secure in this case as Π_0 is secure against this adversary. However, it should be noted that there may be a situation where $r = 0...0$ even in the honest scenario with a very small probability of $1/2^\lambda$ which is negligible and hence we can assume that Π is secure in ω-ψ-$(\delta, 1_\alpha, \beta)$ in the honest case.

With $\alpha = 2$, the adversary can choose the randomness and thus he chooses $r = 0..0$, which allows him to set the bit $b = 1$. In this case, the state will reveal transaction details in t_p, thus leaking all transaction details. Hence, $\Pi_{0'}$ is not secure in this case. This means that Π'_0 is secure in ω-ψ-$(\delta, 1_\alpha, \beta)$ but not secure in ω-ψ-$(\delta, 2_\alpha, \beta)$, and therefore the former is strictly weaker than the latter.

Part iii. Assume that there exists a scheme Π_2 which is secure in ω-ψ-$(\delta, 0_\alpha, \beta)$ (with $\delta < 0$). Here, an honest state setup takes place with hidden randomness. In comparison, randomness involved in the setup is public when $\alpha = 1$. We consider a modified construction Π'_2 in which the state stores the randomness as shown below:

$\texttt{Init}_{\Pi'_2}(1^\lambda; r) = (\texttt{Init}_{\Pi_2}(1^\lambda; r), r)$

Further, the transaction creation is modified so that the randomness is used to encrypt transaction details, which is then leaked out in t_p. If the adversary has the knowledge of the randomness, then they can decrypt the transaction information from t_p. The modified transaction creation operation is as follows.

```
$\texttt{CreateTxn}_{\Pi'_2}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(p, r) \leftarrow p$
    $(t_p, t_s) \leftarrow \texttt{CreateTxn}_{\Pi_2}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    $t_p \leftarrow (t_p, \texttt{Encrypt}_r(R, V_{new}, \bar{S}, V_{old}, m))$
    return $(t_p, t_s)$
}
```

Now, Π'_2 is secure in ω-ψ-$(\delta, 0_\alpha, \beta)$ when the randomness is hidden as Π_2 is secure in this. However, with $\alpha = 1$, the adversary has the knowledge of the randomness r and thus they are able to decrypt the transactions, which will reveal the transaction information, making the scheme insecure. Hence, This shows that Π'_2 can be secure in ω-ψ-$(\delta, 0_\alpha, \beta)$ but not secure in ω-ψ-$(\delta, 1_\alpha, \beta)$. i.e. the former is strictly weaker than the latter.

□

Theorem 13. *For a currency scheme* Π *and for a given combination of* ω, α, ψ *and* β *(with* $\alpha \neq 0$ *and* $\psi_t > 0$*), following separations hold for varying values of* δ*, assuming the existence of a scheme which is secure in the weaker notion in each pair of notion as follows:*

i. ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ *is strictly weaker than* ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$.
i.e. ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$-$\beta \nrightarrow \omega$-$\psi$-$(\mathbf{2}_\delta, \alpha, \beta)$

ii. ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$ *is strictly weaker than* ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$.
i.e. ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta) \nrightarrow \omega$-$\psi$-$(\mathbf{1}_\delta, \alpha, \beta)$

where $\omega \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{1,2,3,4,5\}$, $\beta \in \{0,1\}$ *and* $\alpha \in \{1,2,3\}$ *(Figure 4.8(a)).*

Proof. Part i. The difference between the two scenarios where $\delta = 1$ and $\delta = 2$ is that with the latter case, the adversary is able to manipulate the state whereas in the former, the adversary can only view the state. Suppose that there exists a scheme Π_1, which is secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$.

We now consider a scheme Π_1', a modified construction of Π_1, similar to the one in Part iii of the proof of Theorem 12, with the addition of a bit $b \in \{0,1\}$ returned along with the randomness in the state initialisation of Π_1' as $(\texttt{Init}_{\Pi_1}(1^\lambda, r), r, b)$. The instance where $b = 1$ represents an honest scenario where all operations of the scheme Π_1' are the same as Π_1. Conversely, when $b = 0$, the scheme executes the modified `CreateTxn` as in the above mentioned proof, which encrypts the transaction details using the randomness r.

With $\delta = 1$, Π_1' is secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ as Π_1 is. Now with $\delta = 2$, the adversary is able to manipulate the state and hence they choose $b = 0$ to execute the modified transaction creation so that the state will leak all transaction information through t_p, making Π'_1 insecure. Hence Π'_1 is secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ but not in ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$ and thus the former is strictly weaker than the latter.

Part ii. When $\delta = 0$, the state is hidden and the state can be viewed with $\delta = 1$. Suppose that a currency scheme Π_0 is secure in ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$. Consider a modified construction of this Π'_0, which is exactly the same as the construction in part iii of the proof of the Theorem 12, with the same state initialisation and the transaction creation. Now, with $\delta = 0$, the adversary is unable to view the state, and hence the randomness is also hidden in the state. As Π_0 is secure when $\delta = 0$, Π'_0 is also secure. However, with $\delta = 1$, the adversary is able to view the state and hence the randomness, which allows them to decrypt the transaction information embedded in t_p, giving them a non-negligible advantage over winning the game. As such, Π'_0 is not secure in this case. Accordingly, we can conclude that ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$ is strictly weaker than ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$. □

Theorem 14. *For a currency scheme* Π *and for a given combination of* ω, α *and* ψ *(* $\delta = 0$*), there exists a scheme* Π *which is secure in* ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta)$ *but not secure in* ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta)$.
i.e. ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta) \nrightarrow \omega$-$\psi$-$(\delta, \alpha, \mathbf{1}_\beta)$ *where* $\omega = \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m \in \{0, 1, 2, 3\}$, $\psi_t \in \{2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ *and* $\delta = 0$ *(Figure 4.8(a)).*

Proof. Consider a currency scheme Π which is secure when failed minting is not allowed, i.e. secure in ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta)$. We construct a new scheme Π' based on Π by modifying the state to be (p, v_s), where a special value v_s is initialised in the state, which is used to encrypt transaction details, which is included in t_p (Assuming $\delta = 0$). Further, the minting operation is also modified so that if a mint operation fails, it will leak the special value v_s together with the error message. Modified operations are given below.

```
$\texttt{Init}_{\Pi'}(1^\lambda;\rho)\{$
    $p_0 \leftarrow \texttt{Init}_{\Pi}(1^\lambda;\rho);$   return   $(p_0, v_s)$
$\}$
$\texttt{CreateTxn}_{\Pi'}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow \texttt{CreateTxn}_{\Pi}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    $t'_p \quad \leftarrow \quad (t_p, \texttt{Encrypt}_{v_s}(R, V_{new}, \bar{S}, V_{old}, m))$
    return   $(t'_p, t_s)$
$\}$
```

```
$\texttt{Mint}_{\Pi'}(T, R_m, p; \rho)$ {
    $T' \quad \leftarrow \{t_p : (t_p, c) \in T\}$
    $C' \quad \leftarrow \{c : (t_p, c) \in T\}$
    $X \quad \leftarrow \texttt{Mint}_{\Pi}(T', R_m, p; \rho)$
    if $X = \perp$ then
      return   $(\perp, c', v_s)$
    $(p', V_x) \ \leftarrow X$
    return   $((p', v_s), V_x)$
}
```

Now, Π' is secure in ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta)$ similar to Π. However, when $\beta = 1$, the adversary can cause minting to fail and when minting fails, the resulting error leaks the special value v_s and the encrypted transaction data. Hence, the adversary can decrypt the transaction information, which makes Π' insecure against this adversary. Thus, we can prove that ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta)$ is strictly weaker than ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta)$. □

It is noteworthy that the above relationship can be valid for many other instances based on system specific parameters. We proved this only for one such scenario with $\delta = 0$.

Theorem 15. *For a currency scheme* Π *and for a given combination of* ω, δ, α, $(\psi_{pk}, \psi_{sk})_s$, ψ_m, ψ_t *and* β *(with* $\psi_t > 0$ *and* $\omega_v = 1$*), the following separations hold for varying values of* ψ_v *and for sufficiently long transaction values, assuming the existence of a scheme satisfying the weaker notion out of each pair:*

i. ω-$\psi(\mathbf{0}_v)$-(δ, α, β) *is strictly weaker than* ω-$\psi(\mathbf{1}_v)$-(δ, α, β) *given that the sum of* V_{old} *is sufficiently high.*

ii. ω-$\psi(\mathbf{1}_v)$-(δ, α, β) *is strictly weaker than* ω-$\psi(\mathbf{2}_v)$-(δ, α, β)

iii. ω-$\psi(\mathbf{2}_v)$-(δ, α, β) *is strictly weaker than* ω-$\psi(\mathbf{3}_v)$-(δ, α, β)

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_t, \psi_m \in \{0, 1, 2, 3\}$, $\beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 4.8(e)).*

Proof. Part i. The difference between the two scenarios is that with $\psi_v = 0$, transaction values are unknown to the adversary whereas when $\psi_v = 1$, the values will be revealed at the end of the game. Assume that a scheme Π_0 exists such that it is secure in ω-$\psi(\mathbf{0}_v)$-(δ, α, β). Consider a modified scheme Π_0' with a modified transaction creation operation, which involves encrypting transaction details using the sum of transaction values ($v = \sum_{v_i \in V_{old}}$) and then including it with t_p. The modified transaction creation is given below.

```
$\texttt{CreateTxn}_{\Pi_0'}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow \texttt{CreateTxn}_{\Pi_0}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    $t_p' \leftarrow (t_p, \texttt{Encrypt}_v(R, V_{new}, \bar{S}, V_{old}, m))$
    return $(t_p', t_s)$
}
```

Here we want the possible range of v to be sufficiently high so that there is enough entropy in v to ensure that the adversary cannot easily guess the value. Further, when $\psi_v < 2$, the adversary chooses an upper bound V_{max} for the values and we should have $v_i \geq 2^{\lambda} : v_i \in V_{max}$ in order to ensure that the adversary does not gain a non-negligible advantage (and we also assume that V_{max} is large enough). Other functions of the scheme are the same as in Π_0.

Now, when $\psi_v = 0$ the adversary does not have access to the transaction values, and hence Π_0' will function similar to Π_0, which is secure against this adversary. In the case of $\psi_v = 1$, the adversary can view the transaction values at the end of the game, which enables them to decrypt all the transaction details from t_p using the key v calculated from the transaction values. This provides a non-negligible advantage to the adversary to win the game and hence Π_0' is insecure in this case. Therefore, we can conclude that ω-$\psi(\mathbf{0}_v)$-(δ, α, β) is strictly weaker than ω-$\psi(\mathbf{1}_v)$-(δ, α, β), given that v is sufficiently high.

Part ii. Consider a scheme Π_1 similar to the construction defined in the proof of Theorem 4 part (i) b, which is secure in ω-$\psi(\mathbf{1}_v)$-(δ, α, β). This scheme is slightly modified in Π'_1, where the modified `CreateTxn` function reveals all transaction details if the sum of the transaction values v is in metadata (instead of the public keys of the sender). Note that in this case, the scheme should allow for sufficiently long metadata so that it is not trivial for the adversary to detect the value.

With this construction, Π'_1 is secure in ω-$\psi(\mathbf{1}_v)$-(δ, α, β), as Π_1 is secure in the same by our assumption, given that transaction values have a high enough entropy, which makes it harder for the adversary to guess the values. In the case of $\psi_v = 2$, transaction values are known to the adversary throughout the game which enables them to craft a transaction by embedding the sum v in metadata. As this leaks information about the transaction details, Π'_1 is not secure against this adversary, i.e. ω-$\psi(\mathbf{2}_v)$-(δ, α, β). Hence, we can conclude that ω-$\psi(\mathbf{1}_v)$-(δ, α, β) is strictly weaker than ω-$\psi(\mathbf{2}_v)$-(δ, α, β) under the given conditions.

Part iii. The two scenarios in this case differs in how the transaction value is chosen between honest and adversarial choices. This proof can be constructed similar to the proof of Theorem 4 Part (i) c, with a modification to the `CreateTxn` function so that if a special transaction value v_s (instead of the special ID in that proof) appears in a transaction, all transaction details are revealed. As in the previous parts however, this value v_s should have a sufficiently high entropy so that v_s is not chosen accidentally except with negligible probability. Accordingly, we can show that ω-$\psi(\mathbf{2}_v)$-(δ, α, β) is strictly weaker than ω-$\psi(\mathbf{3}_v)$-(δ, α, β) subject to the above condition. □

Theorem 16. *For a currency scheme Π and for a given combination of ω, δ, α, $(\psi_{pk}, \psi_{sk})_s$, ψ_v, ψ_t and β (with $\psi_t > 0$, $\delta > 0$), the following separations hold for varying values of ψ_m and for sufficiently long metadata values, assuming the existence of a scheme satisfying the weaker notion in each pair of notions:*

i. ω-$\psi(\mathbf{2}_m)$-(δ, α, β) is strictly weaker than ω-$\psi(\mathbf{3}_m)$-(δ, α, β)

ii. ω-$\psi(\mathbf{1}_m)$-(δ, α, β) *is strictly weaker than* ω-$\psi(\mathbf{2}_m)$-(δ, α, β)

iii. ω-$\psi(\mathbf{0}_m)$-(δ, α, β) *is strictly weaker than* ω-$\psi(\mathbf{1}_m)$-(δ, α, β)

where $\omega \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{1,2,3,4,5\}$, $\beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.8(f)).*

Proof. (Sketch)
The proof follows the same line of argument as the proof in Theorem 15 subject to the condition that the schemes allow for sufficiently long metadata (i.e. $|m| = \text{poly}(\lambda)$) so that there is no trivial information leakage to the adversaries. □

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 following separations hold for varying values of* ψ_t, *assuming the existence of a scheme satisfying the weaker notion in each pair of notions:*

i. ω-$(\psi(\mathbf{0}_t))$-(δ, α, β) *is strictly weaker than* ω-$(\psi(\mathbf{1}_t))$-(δ, α, β) .

ii. ω-$(\psi(\mathbf{1}_t))$-(δ, α, β) *is strictly weaker than* ω-$(\psi(\mathbf{2}_t))$-(δ, α, β) .

iii. ω-$(\psi(\mathbf{2}_t))$-(δ, α, β) *is strictly weaker than* ω-$(\psi(\mathbf{3}_t))$-(δ, α, β).

iv. ω-$(\psi(\mathbf{3}_t))$-(δ, α, β) *is strictly weaker than* ω-$(\psi(\mathbf{4}_t))$-(δ, α, β) .

v. ω-$(\psi(\mathbf{4}_t))$-(δ, α, β) *is strictly weaker than* 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 4.8(c)).*

Proof. Part i. The two scenarios represented by the notions in this case differ only in terms of whether the adversary has knowledge of the transaction. With

$\psi_t = 0$ the transaction is hidden whereas when $\psi_t = 0$, t_p is revealed at the end of the game.

Consider a scheme Π_0, which is secure in ω-$(\psi(\mathbf{0}_t))$-(δ, α, β). This means that the scheme is secure when the transaction (t_p) is hidden. We construct a new scheme Π'_0 from this with a modified transaction creation operation, which leaks all transaction information in t_p as below.

```
CreateTxn$_{\Pi'_0}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_0}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    $t'_p \quad \leftarrow \quad (t_p, (R, V_{new}, \bar{S}, V_{old}, m))$
    return $(t'_p, t_s)$
}
```

Now, Π'_0 against an ω-$(\psi(\mathbf{0}_t))$-(δ, α, β) adversary, as t_p is hidden and Π'_0 operates in the same manner as Π_0. When $\psi_t = 1$, the adversary gets to view t_p at the end of the game, which leaks all transaction details as per the modification above, thus making the scheme insecure against this adversary. Hence, Π'_0 is not secure in ω-$(\psi(\mathbf{1}_t))$-(δ, α, β) although the scheme is secure in ω-$(\psi(\mathbf{0}_t))$-(δ, α, β). Thus, we conclude that the latter is strictly weaker than the former.

Part ii. The two cases here differ in the adversary's knowledge of t_p or t_s. We consider a currency scheme Π_1 which is secure in ω-$(\psi(\mathbf{1}_t))$-(δ, α, β). Hence, Π_1 is secure when t_p is leaked to the adversary. We now consider a modified construction of Π_1, Π'_1 with the following transaction creation operation so that t_s always leaks all transaction information.

```
CreateTxn$_{\Pi'_1}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_1}$$(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    $t'_s \quad \leftarrow \quad (t_s, (R, V_{new}, \bar{S}, V_{old}, m))$
    return $(t_p, t'_s)$
}
```

In the case where $\psi_t = 1$, t_p is revealed to the adversary at the end of the game and Π'_1 is secure in ω-$(\psi(\mathbf{1}_t))$-(δ, α, β) since Π_1 is secure here as t_p does not reveal any additional information. When $\psi_t = 2$, t_s is revealed to the adversary which leaks all transaction details, thus giving the adversary a non-negligible advantage in the game. Hence, Π_1' is not secure against this adversary. Therefore, ω-($\psi(\mathbf{1}_t)$)-(δ, α, β) is strictly weaker than ω-($\psi(\mathbf{2}_t)$)-(δ, α, β).

Part iii. With $\psi_t = 3$, the randomness of the actual coins involved in the transaction is revealed to the adversary as opposed to revealing just the t_s when $\psi_t = 2$. For this proof, we consider a scheme Π_2, which is secure in ω-($\psi(\mathbf{2}_t)$)-(δ, α, β). i.e. Π_2 is secure against an adversary who has the knowledge of t_s. Then, we construct a new scheme Π_2' from Π_2 by modifying the transaction creation process where transaction is created using the hash of the randomness r and use r as a symmetric key to encrypt transaction details, which is then included in t_p as below.

```
$\texttt{CreateTxn}_{\Pi_2'}(R, V_{new}, \bar{S}, V_{old}, m, p; r)$ {
    $(t_p, t_s) \leftarrow \texttt{CreateTxn}_{\Pi_2}(R, V_{new}, \bar{S}, V_{old}, m, p; \texttt{Hash}(r))$
    $t_p' \quad \leftarrow \quad (t_p, \texttt{Encrypt}_r(R, V_{new}, \bar{S}, V_{old}, m))$
    **return** $(t_p', t_s)$
}
```

Now Π_2' is secure when $\psi_t = 2$ as Π_2 is secure against an adversary having the knowledge of t_s given that the hash function is one-way (i.e. the adversary may be able to recover the hash of r from t_s, but not r). When the adversary is able to view the randomness r of the coins in the transaction (i.e. $\psi_t = 3$), they can decrypt the transaction details leaked in t_p using the key r and hence can gain a non-negligible advantage in the game. Hence, it makes Π_2' not secure against this adversary. Thus, Π_2' is secure when $\psi_t = 2$ but not when $\psi_t = 3$. i.e. ω-($\psi(\mathbf{2}_t)$)-(δ, α, β) is strictly weaker than ω-($\psi(\mathbf{3}_t)$)-(δ, α, β).

Part iv. The difference between the two notions in this case is that the adversary gets to choose the coins (i.e. randomness of the coins) with $\psi_t = 4$. Here we consider a scheme Π_3, which is secure against an adversary who has the knowledge of the randomness of the coins in the transaction with $\psi_t = 3$ and $\delta > 0$. Then we construct a modified scheme Π_3' similar to the above except that if a special randomness r_s is used, all transaction details are leaked in t_p as follows:

```
$\texttt{CreateTxn}_{\Pi'_3}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$ {
    $(t_p, t_s) \leftarrow \texttt{CreateTxn}_{\Pi_3}(R, V_{new}, \bar{S}, V_{old}, m, p; \rho)$
    if  $\rho = r_s$  then
      $t'_p \quad \leftarrow \quad (t_p, (R, V_{new}, \bar{S}, V_{old}, m))$
      return $(t'_p, t_s)$
    else return  $(t_p, t_s)$
}
```

Π'_3 is secure in ω-$(\psi(\mathbf{3}_t))$-(δ, α, β) as Π_3 is. However, in the case where $\psi_t = 4$, the adversary has the ability to choose the randomness, and hence can choose the special randomness r_s, which causes t_p to reveal all transaction details. Hence, Π'_3 is not secure against this adversary although it is secure when $\psi_t = 3$, thus it is proven that ω-$(\psi(\mathbf{3}_t))$-(δ, α, β) is strictly weaker than ω-$(\psi(\mathbf{4}_t))$-(δ, α, β) for $\delta > 0$.

Part v.

In these two notions, the adversary is able to create the transaction when $\psi_t = 5$ whereas they can only choose the randomness when $\psi_t = 4$. Consider a scheme Π_4 which is secure against an adversary in ω-$(\psi(\mathbf{4}_t))$-(δ, α, β).

We then construct a new scheme Π'_4 with the state initialised as $\texttt{Init}_{\Pi'_4}(1^\lambda; \rho) = (\texttt{Init}_{\Pi_4}(1^\lambda; \rho), 0)$ (i.e. state will be of the form $(p, 0)$) and t_p is modified as (t_p, b) with $b \in \{0, 1\}$. When $b = 1$, the minting operation leaks the details of all minted transactions (i.e.$\{t_p\}$) in the state. We also assume that $\delta > 0$ in this case. Accordingly, the modified minting operation is as follows while all other functions operate the same as in Π_4:

```
$\texttt{Mint}_{\Pi'_4}(\{(t_p, b)\}, R_m, (p, 0))$ {
    $p_1 \leftarrow \texttt{Mint}_{\Pi_4}(\{t_p\}, R_m, p)$
    if  $b = 1$  then return  $(p_1, \{t_p\})$
    else return  $(p_1, 0)$  }
```

Now, Π'_4 is secure when $\psi_t = 4$, since Π_4 is secure against an adversary with the ability to choose the randomness of the coins for the transaction. However, when the adversary is capable of creating the transaction (i.e. $\psi_t = 5$), they can choose $b = 1$ (i.e. return $(t_p, 1)$ as an input in the game) so that the minting operation leaks the information of all minted transactions. In this case, the adversary can gain additional information about the minted transactions by looking at the minted state (as $\delta > 0$) and thus has a non-negligible advantage in winning the game. Hence, Π'_4 is not secure against this adversary and thus this proves that ω-$(\psi(\mathbf{4}_t))$-(δ, α, β) is strictly weaker than ω-$(\psi(\mathbf{5}_t))$-(δ, α, β).

□

It should be noted that in some cases the separations are not known to hold for all values of the unspecified parameters. Further, some relationships depend on what information is contained in each entity and how they can be interpreted together with other relevant parameters, so that no obvious links can be defined.

Based on the above theorems, we also define the following corollaries.

Corollary 1. *Given that a currency scheme* Π *is secure in the strongest anonymity notion (i.e. secure against the strongest possible adversary), then* Π *is also secure in any other notion (any other adversary).*

i.e. $(1_s 1_r 1_v 1_m)_\omega$-$((4,4)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta) \rightarrow \omega$-$\psi$-$(\delta, \alpha, \beta)$
where $\omega \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}$, $\beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 4.7).*

Proof. (sketch) This follows from the above theorems as illustrated in figures 4.7 and 4.8 since this notion is the strongest among all. Accordingly, we name this notion as *Absolute Fungibility* or as the notion of ALL-IND-FULL-FULL. □

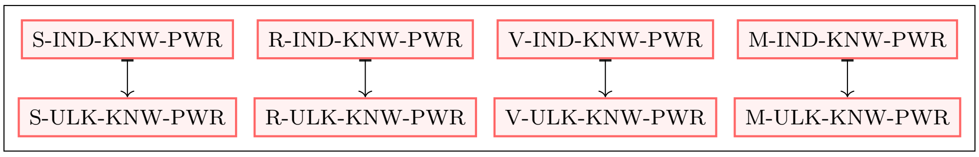


Figure 4.9: Relations between indistinguishability and unlinkability (Corollary 2).

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

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

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

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

iv. *given* Π *is secure in M-IND-KNW-PWR for a given adversarial knowledge KNW of senders, recipients and value and given adversarial power PWR, then* Π *is also secure in M-ULK-KNW-PWR.*
i.e. M-IND-KNW-PWR $\rightarrow$ *M-ULK-KNW-PWR*
(Figure 4.9)

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.

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

Part iii. Follows from Theorem 9.

Part iv. Follows from Theorem 10. $\square$

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 19). Note that this notion is trivial in that no adversary can ever win the corresponding game with greater success than a blind guess since the transactions t_0 and t_1 are, aside from randomness (which is also hidden), identical.

4.4 Discussion

Our main goal in this chapter is to develop a framework to model anonymity in decentralised systems such as cryptocurrencies. The anonymity game we constructed in this connection is capable of capturing a plethora of unique anonymity notions (the game itself supports 23,040,000 unique parameter vectors out of which over 600,000 represent meaningful realisations) based on an exhaustive adversarial model. Further, our model facilitates modelling of anonymity with respect to many entities, without being restricted to the participants of a transaction only as focused as is the case in many existing studies. This type of analysis may provide useful insights on how privacy could be compromised in many different ways. For example, learning the details of transactions having a given value may leak valuable information about the flow of funds in a transaction graph. Hence, our framework enables systematic modelling of anonymity along multiple dimensions as opposed to existing methodologies, which are mostly based on the functional level as outlined in Chapter 2. It is noteworthy that this framework has been built as a general framework, in order to achieve a unified means for modelling a wide range of cryptocurrencies. For this purpose, the framework approximates implementation details, without referring to currency-specific information. Hence, we emphasise here that the generality of the framework is merely by design and not a limitation.

The absence of formalised definitions for various anonymity notions poses a barrier for effective comparison of different constructions. This has also resulted in various interpretations of anonymity aspects across diverse implementations. Conversely, our solution supports the formalisation of each parametric vector in terms of a unique anonymity definition as explained through a set of examples in Section 4.3.2. We define these notions around the two privacy notions; indistinguishability and unlinkability, so that these can be compared with many existing anonymity representations. In addition, the naming convention we use for these formal anonymity definitions closely follows the conventional cryptographic security definitions, so that the notions reveal relevant adversarial capabilities related to each notion. This simplifies the representation of parametric vectors corresponding to each attacker scenario, while indicating the differences between them.

Another important contribution presented in this chapter is the investigation of relationships among the anonymity notions represented by different parametric vectors in the game. While it is paramount to understand the implications and dependencies of different attributes within the entities in a given system, it is equally important to formalise the same so that it simplifies the analysis of anonymity across different constructions. Accordingly, we have presented a set of theorems, which formalise the implications, equivalences and separations among anonymity notions resulting from our framework, with respect to individual parameters. During this process, we have identified that some anonymity parameters are directly influenced by the transaction graph whereas others have indirect dependencies which are not obvious. Hence, these theorems provide a useful means of evaluation and comparison of anonymity of currency schemes effectively and precisely. More importantly, this strenuous exercise reveals the complexity of anonymity in the multi-dimensional parametric space we have constructed through our framework, which would not have been apparent otherwise.

We have since published the contributions in this chapter in [5]. We demonstrate the applicability of this framework in real-world settings through analysis outcomes obtained from several case studies in the following chapter.

Chapter 5

Analysis of Anonymity: Case Studies

This chapter provides insights on the application of the anonymity framework we developed in the previous chapter. We present several case studies based on real world cryptocurrency schemes to demonstrate how this framework can be utilised to evaluate anonymity aspects of cryptocurrency schemes effectively and draw useful comparisons among them.

5.1 Introduction

As argued in Chapter 4, the concept of anonymity can be interpreted in many different ways and hence anonymity in a given context depends on a number of factors. We have encapsulated these elements in a common template which can be applied to a wide range of cryptocurrencies. Now, we are in a position to examine how to apprehend anonymity properties of specific cryptocurrencies through the eyes of this template.

In this section, we study a toy example, a Trusted Third Party (TTP) scheme to obtain a taste of the highest level of anonymity that can be modelled through our framework. In contrast, Bitcoin is perceived to have a very weak level of anonymity as claimed in studies such as [71, 77, 100]. Ethereum on the other hand, is even weaker with respect to anonymity due to its account-based model as presented in [18, 50, 100]. In this context, we explore the anonymity aspects of Bitcoin, together with several other schemes that claim to possess higher levels of anonymity under our anonymity framework.

It should be noted however that our intention here is not to prove any particular anonymity property, but to consider them as axioms taken on faith from primary sources and translate claimed security properties into our notation. Further, to consider all anonymity notions resulting from our framework would be infeasible within the scope of this study. Instead, we focus on specific notions of interest, which could be linked to the attributes discussed in the literature, and also be of use in providing meaningful comparisons within them.

We summarise the contribution included in this chapter as follows.

5.2 Our Contributions

The focus of this chapter is to present the findings of several case studies conducted with respect to a group of cryptocurrency schemes using the constructed anonymity framework. For this purpose, we present case studies for a Trusted Third Party Scheme and the Bitcoin, Zcash, Monero and Mimblewimble currency schemes, all having diverse implementations. Further, we also provide comparisons of chosen anonymity notions across these schemes.

5.3 Analysis of anonymity

As mentioned above, here we focus on specific notions of interest towards our purpose of demonstrating how our framework can be deployed to characterise anonymity properties of actual cryptocurrencies, precisely. We consider *Indistinguishability* (IND) and *Unlinkability* (ULK) notions related to sender (S), recipient (R) and value (V), in a bid to provide a meaningful comparison across real-world currency schemes. It should be noted that we do not consider the anonymity notions with respect to metadata here, since they may represent different information in each scheme, thus making comparisons unproductive.

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

We adopt the following process during the analysis of the above currency schemes. We start by providing a brief description of the functionality of the scheme being studied, followed by a recap of how the parameters of the adversarial model can be used to model the functionality of the scheme. Next, IND and ULK notions are examined with respect to senders, recipients and value, while identifying the strongest level of anonymity satisfied by the scheme under the given circumstances.

5.3.1 A Trusted Third Party (TTP) scheme

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

Table 5.1 illustrates how the variables in the TTP scheme are modelled in terms of the notation in our model.

Table 5.1: Modelling the variables of the TTP Scheme

Notation in our model	**Variables in the TTP scheme**
$a_{pk} = a_{sk}$	User address
V_{old}, V_{new}	Input/Output transaction values
m	Transaction metadata
t_p	No public transaction data
t_s	All transaction data

Adversarial capabilities

In the TTP model, the CA can have its own state variables outside the challenger and the adversary, and thus is not required to accept the adversarial state. Also, the initial state will be an empty list of transactions, accounts etc., allowing any method of state initialisation possible. Hence we can allow the adversary to take any value for δ and α in our adversarial model (Table 3.4). Further, we assume that transactions are encrypted with an asymmetric system using the CA's public key, and hence can be revealed in the end without revealing any information. We model user identities in terms of a single address thereby setting $a_{pk} = a_{sk}$ in our model. To enable the adversary to supply sender/recipient addresses to the challenger, we provide access to an additional oracle `DelegateAccess` to transfer the authority of the addresses controlled by the adversary to the challenger. Thus, the challenger is able to create the transactions required for different scenarios. Note that this oracle is only specific to the TTP functionality, and is reminiscent of how ideal functionalities must be explicitly augmented with player corruption functions in the UC model.

Analysis of anonymity

First, we consider a FULL power adversary (denoted by $(2_\delta$,3_α,$1_\beta))$, who has the complete knowledge of recipients, value and metadata, but knows senders only by public keys and provides the input transactions to the game (named as PUBS knowledge denoted by $((3,0)_s,(4,4_r),3_v,3_m,5_t)_\psi)$ against the goal of S-IND. We name this adversary as S-IND-PUBS-FULL, who in this case cannot learn any new information about the sender corresponding to the minted transaction as the state is private, and thus has negligible advantage of winning the Anonymity game (given by the parameter vector $(1_s000)_\omega$-$((\mathbf{3},\mathbf{0})_\mathbf{s},(4,4)_r,3_v,3_m,5_t)_\psi$-$(2_\delta,3_\alpha,1_\beta))$. Hence, the TTP scheme is secure against S-IND-PUBS-FULL adversary and also against a S-ULK-NILS-FULL adversary having no knowledge of senders (NILS knowledge) represented by $(1_s000)_\omega$-$((\mathbf{0},\mathbf{0})_\mathbf{s},(4,4)_r,3_v,3_m,5_t)_\psi$-$(2_\delta$,$3_\alpha$,$1_\beta)$ by implication. Similar anonymity notions hold for R and V as well. Accordingly, we can say that the scheme is secure even against an adversary with FULL-FULL capabilities, for all entities; i.e. ALL-IND-FULL-FULL setting, as the scheme does not leak any information to the adversary. This is modelled by the vector

$(1111)_\omega$-$((4,4)_s,\ (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta$,$3_\alpha$,$1_\beta)$, which depicts '*absolute fungibility*' demonstrating the strongest possible level of anonymity in our model.

5.3.2 Bitcoin

We discussed the anonymity considerations of Bitcoin in detail, as investigated in the literature, in Chapter 2. This peer-to-peer cryptocurrency relies on a public blockchain where transaction data are public. Users are identified via public addresses and they initiate transactions using their private keys to spend funds (unspent transaction outputs). Transaction inputs include references to unspent transaction outputs and a set of new outputs with corresponding values, which later becomes inputs to another transaction. In addition, transactions also contain additional data which help in the verification and are broadcast on the network. Participating network nodes compete to create new blocks (mining) to include new transactions in the blockchain and a qualifying block is accepted by the network based on a Proof-of-Work system. Table 5.2 summarises how our notation can be used to model the variables in the Bitcoin scheme.

Table 5.2: Modelling the variables of the Bitcoin Scheme

Notation in our model	**Variables in the Bitcoin scheme**
a_{pk}, a_{sk}	Public key and secret key of user addresses
V_{old}, V_{new}	Input values of unspent transactions/Output values
m	Transaction metadata
t_p	All transaction data
t_s	No secret transaction data (empty)

Adversarial capabilities

All adversarial capabilities (i.e. all values) can be considered in the context of Bitcoin, yet some capabilities do not make any meaningful outcomes due to the functionality. Thus we avoid such capabilities from our consideration. There is no secret transaction part of a transaction t_s under normal operating conditions as all Bitcoin transactions are public. Hence, we do not consider the scenario $\psi_t = 2$ here. Further, Bitcoin functionality also requires public access to the state information, and thus we assume that all adversaries have access to the state and only consider $\delta > 0$. The Bitcoin network was initialised honestly using an honest public random value, and hence in practice the adversary cannot have $\alpha = 0$ which models an honest hidden initialisation, even though the protocol allows for it in principle. Hence, no current adversary can influence the initialisation.

Analysis of anonymity

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

Conversely, consider a weak adversary in our game against an empty test variable, who has no information of the transaction (NIL knowledge), but can view the state setup and the state (VIEW power), denoted by NIL-IND-NIL-VIEW and parameterised by $(0000)_\omega$-$((0,0)_s, (0,0)_r, 0_v, 0_m,\ 0_t)_\psi$-$(1_\delta$,$1_\alpha$,$0_\beta)$. Here, the adversary has to distinguish between two identical transactions carrying same data, except with different randomness. Despite the public transaction history, the adversary cannot identify the correct transaction with a substantial probability, thus making the Bitcoin system secure against this attacker. However, if we increase at least one capability, the scheme becomes insecure. Thus, we conclude that Bitcoin only satisfies an extremely weak notion in our model, which only provides anonymity against two identical transactions that only differ in the randomness. It should be noted however that we make this claim subject to the computational and operational assumptions of the Bitcoin construction. In fact, the only way to make the scheme anonymous is to make the state private (i.e. $\delta = 0$), which would render Bitcoin useless for its intended application as a decentralised currency.

5.3.3 Zcash

As mentioned in Chapter 2, Zcash emerged as a result of the efforts to improve the anonymity of Bitcoin. We consider the Zcash Sapling specification for this study [40]. Here we only consider the transactions between shielded addresses (referred to as addresses hereafter) as transparent addresses and related transactions operate similar to Bitcoin [39]. Each address has a private spending key that allows the owner to spend the coins (notes) sent to that address. Each note is coupled with a unique nullifier, which is generated using the spending key, and remains hidden until it is spent. A note also has a unique note commitment, which is publicly revealed when the note is created. Without the private key, it is infeasible to link a note commitment to its nullifier. An unspent note in Zcash is a note with a publicly revealed commitment and a hidden nullifier. When a shielded transaction is created, nullifiers of input notes and commitments of output notes (i.e. newly created) are revealed. In addition, the value of a shielded transaction is also hidden, and is revealed through value commitments related to input and output notes, and relevant balancing operations are carried out as homomorphic operations. Further, zk-SNARK primitives are used for functions such as proving the ownership of notes, verifying and validating transactions [40]. In order to facilitate these zero-knowledge proofs, a common reference string is generated at the initial setup phase of the scheme [11, 17, 31]. Nonetheless, if an adversary is able to manipulate the randomness associated with the reference string, then they can generate false proofs to prove ownership etc. and hence the randomness has to be hidden [11, 28]. A model of the Zcash functionality in terms of our notation is given in table 5.3.

Table 5.3: Modelling the variables of the Zcash Scheme

Notation in our model	**Variables in the Zcash scheme**
a_{pk}, a_{sk}	Public key and private spending key of user addresses
V_{old}, V_{new}	Values of input notes/output notes
m	Transaction metadata
t_p	All public transaction data (nullifiers of input notes, commitments of output notes, value commitments, zk-SNARK proof)
t_s	Secret transaction data (nullifiers of output notes etc.)

Adversarial capabilities

Similar to Bitcoin, we can model Zcash addresses through the payment addresses (a_{pk}, a_{sk}) in our model. As the state is public under normal operation conditions, it makes sense to only consider the adversarial capabilities with $\delta > 0$. As the adversary may gain a non-negligible advantage with the knowledge of the randomness associated with the setup space, we only consider the instances with hidden randomness, i.e. $\alpha \notin \{1, 2\}$. In shielded transactions, senders and recipients correspond to the nullifiers of input notes and to the commitments of output notes respectively. Further, the values of input/output notes are also concealed as value commitments in the transaction. t_p represents public transaction data such as nullifiers of input notes, output note commitments, value commitments and zk-SNARK proof data whereas actual input/output notes, nullifiers of output notes and other relevant private data can be modelled by t_s. The knowledge of secret keys (i.e. $\psi_{sk_s} > 0$) is required to link the nullifiers of input notes to their owners (senders) and the private keys of recipients (i.e. $\psi_{sk_r} > 0$) should be known to link the note commitments of output notes to their owners (recipients).

Analysis of anonymity

We begin by analysing the unlinkability property. Although the linkability of Zcash transactions is explored in literature such as [75] with respect to transactions involving both shielded and transparent addresses, we only consider shielded addresses here. Consider an adversary for S-ULK (i.e. distinguish between two unknown senders aka sender unlinkability) who has all powers except to cause minting to fail (ACTIVE power), and has full knowledge of recipients, metadata and values and public transaction data (input/output note commitments, value commitments etc.), except the senders (NILS-PUBT knowledge), which we capture in a parameter vector $((0,0)_s, (4,4)_r, 3_v,\ 3_m, 1_t)_\psi$-($2_\delta$,$3_\alpha$,$0_\beta$). The adversary cannot obtain any additional knowledge of the minted transaction as the note commitments do not leak any information about the sender. The knowledge of the recipients also does not provide any advantage as the output notes do not carry any information about the sender. Hence, this adversary has a negligible advantage over winning the game. Hence, the Zcash scheme is secure in S-ULK-NILS-PUBVT-ACTIVE (i.e. sender unlinkability with unknown senders, known value and transaction, adversarial recipients and metadata, with ACTIVE power). If the adversary is given more powers to cause minting to fail (i.e. FULL power), then he may gain additional information about account balances etc. based on unsuccessful minting attempts, making the system insecure with respect to S-ULK-NILS-PUBVT-FULL. Further, for any adversary having $\psi_t > 1$, the adversary has access to t_s, which provides additional knowledge about sensitive transaction data that compromises security. Similarly, we can also show that Zcash is secure in R-ULK-NILR-PUBVT-ACTIVE (i.e. recipient unlinkability with hidden recipients, adversarial senders and metadata, and with the knowledge of the value and the transaction), but not in R-ULK-NILR-PUBVT-FULL.

The scheme also satisfies S-IND-PUBSVT-ACTIVE security (i.w. sender indistinguishability with the knowledge of public keys of senders, value and transactions, with adversarial recipients and ACTIVE power), as the knowledge of senders' public keys and public transaction data (PUBSVT knowledge) does not reveal any information about the nullifiers of input notes (i.e. $((3,0)_s, (4,4)_r, 2_v, 3_m, 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta))$. Yet, with the same reasoning as with S-ULK, Zcash fails in S-IND-PUBSVT-FULL. Similarly, Zcash is secure in R-IND-PUBRVT-ACTIVE (i.e. recipient indistinguishability with the knowledge of recipients' public keys, transaction value and transaction data), but not in R-IND-PUBRVT-FULL.

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

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

5.3.4 Monero

Monero is another cryptocurrency that claims improved anonymity based on several cryptographic primitives such as linkable ring signatures and stealth addresses to achieve anonymity with respect to senders and recipients [95]. In addition, Ring Confidential Transactions (RingCT) are used to conceal transaction values through value commitments [95]. Each user has two pairs of private/public keys as spend and view keys. A sender creates a one-time public key (stealth address) for each output using recipients' public keys. The sender mixes the actual inputs with a set of additional random public keys (known as mixins) using linkable ring signatures, to produce a signature for the ring of inputs. The one-time public key, the signature and the public keys of inputs (in the ring) are submitted to the network along with other transaction data [1, 95]. A sender can include an (optional) pre-agreed, encrypted payment ID, enabling respective recipients to identify the sender using their private keys. The recipients can retrieve outputs using both their private/public view keys and can spend them using the spending keys. Outsiders can only view the public keys in the ring (of probable senders), with each being an equally probable input to the transaction.

Table 5.4: Modelling the variables of the Monero Scheme

Notation in our model	**Variables in the Monero scheme**
a_{pk}, a_{sk}	Public keys of spend/view keys, Private keys of spend/view keys
V_{old}, V_{new}	Values of inputs/outputs
m	Mixin data (i.e. public keys in the ring)
t_p	All public transaction data (Value commitments, Public keys of the participants in the ring)
t_s	Secret transaction data (transaction ID, values, senders, recipients etc.)

Adversarial capabilities

Similar to others, the Monero Blockchain state is also public under normal operation, thus we assume that the adversary can view the public state, which allows $\delta > 0$ in our model as before. However, most of the transaction data (e.g. actual senders, recipients, values etc.) are hidden from the public and we model them in t_s. We use a_{sk} to represent private keys of both spend and view keys collectively, and a_{pk} to model public keys of the same. In addition, mixin data are modelled by metadata in our model, i.e. m. Further, the adversary may gain a non-negligible advantage if the secret keys of the senders are known (i.e. $\psi_{sk_s} > 0$). If the adversary knows the secret key of the recipient (i.e. $\psi_{sk_r} > 0$) and t_s (i.e. $\psi_t > 2$), then they can guess the sender. Moreover, if a transaction (i.e. t_s) consists of a payment ID, the secret key of the recipient can be used to deanonymise the sender; i.e. $\psi_{sk_r} > 0$.

Analysis of anonymity

First we look at the unlinkability property of Monero, which is analogous to the notion of traceability of Monero, referred to in [46, 95]. We consider the S-ULK-NILS-PUBT-ACTIVE notion as with Zcash, without the knowledge of likely senders, with the other parameters being adversarial (i.e. $((0,0)_s, (4,4)_r, 3_v, 3_m, 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta))$. The state only reveals the public keys of a possible set of senders, but not the recipients nor the value. Yet, if the adversary chooses the mixins (i.e. dummy public keys in the ring), then they can easily identify the sender as sender's public key is the only unknown key in the ring. Thus, Monero cannot be secure if the adversary has the knowledge of the mixins in the ring, meaning that Monero is not secure against the above adversary. Hence, we define a weaker notion by setting $\psi_m = 0$ in our model, which represents an adversary who has no knowledge of the sender or metadata (NILSM-PUBT knowledge), against which Monero is secure; i.e. S-ULK-NILSM-PUBT-ACTIVE adversary modelled by $(1_s000)_\omega$-$((0,0)_s, (4,4)_r, 3_v, 0_m, 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$.

In the case of sender indistinguishability (S-IND) where an adversary has some knowledge of the possible senders (i.e. $\psi_{pk_s} > 0$ or $\psi_{sk_s} > 0$), with $\omega_s = 1$, security of Monero depends on how the mixins are chosen. The problem occurs if one of the possible senders (S_0 or S_1) is not included in the mixins, then the adversary can trivially guess who the sender is. If the keys in the ring are chosen by the adversary, then they can manipulate the situation and win the game. On the other hand, if the keys are chosen randomly, then the above scenario happens with some probability (based on the total number of keys in the scheme and the ring size), in which case the adversary's advantage in winning the game increases with the total number of keys (assuming the ring size is constant). Accordingly, with the goal of S-IND, security can be compromised unless the mixins are chosen optimally, knowing the adversary's goal. Further, t_p also independently leaks information about the mixins and against any adversary having the knowledge of t_p (i.e. $\psi_t > 0$), the scheme is not secure. Hence we can see that Monero is secure against the S-IND-NILMT-PUBS-ACTIVE adversary (the adversary has no knowledge about the mixins and the transaction) given that mixins are chosen randomly. This setting can be represented in terms of the following parametric vector: $(1_s000)_\omega$-$((3,0)_s,(4,4)_r,\ 3_v,0_m,0_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$.

With recipient anonymity, we can see that Monero complies with R-ULK-NILR-PUBT-ACTIVE (hidden recipients with the knowledge of t_p) as funds are received by stealth addresses which can be claimed only by the recipient with the matching private key. This notion of unlinkability closely relates to the notions described in [46, 95]. Similarly, Monero is also secure in R-IND-PUBRT-ACTIVE ($(01_r00)_\omega$-$((4,4)_s,(3,0)_r,\ 3_v,3_m,1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ as the knowledge of recipients' public keys or mixin data do not reveal anything about the ownership of stealth addresses. Further, as values are hidden, Monero's anonymity with respect to values reduces to the knowledge of the secret keys of the senders/recipients similar to Zcash and hence it satisfies V-ULK-NILV-PUBSRT-FULL and V-IND-PUBSRT-ACTIVE notions. As with Zcash, S-IND, S-ULK, R-IND, R-ULK and V-IND goals fail against a FULL power adversary with the information leakage from failed minting. Thus, we can see that the maximal anonymity level satisfied by Monero is the ALL-IND-NILMT-PUBSR-ACTIVE security (i.e. the adversary has the knowledge of the public keys of senders and recipients except the mixins and the transaction) given by the parameter vector $(1111)_\omega$-$((3,0)_s$, $(3,0)_r$, $3_v,0_m,0_t)_\psi$-$(2_\delta$,3_α,$0_\beta)$.

5.3.5 Mimblewimble

The Mimblewimble protocol focuses on improving anonymity and scalability through confidential transactions and transaction aggregation [32, 42]. We study the Grin implementation of this protocol for this analysis [41]. A coin in this is a commitment, $C = vH + rG$ where v is the value, r is the randomness (hence the private key of the coin), and H, G are generators of a discrete logarithm [32]. The opening of the commitment of a coin is necessary to spend that coin, which requires the corresponding secret key (r). The sender sends the input coins (commitments) to the recipient over an authenticated channel, who then adds the commitments to the output coins (by including individual private keys) and a partial signature for the transaction (using a random nonce), which is sent back to the sender. The sender validates the received signature and generates their portion of the signature and broadcasts the transaction on the network, which is verified (via the relevant public key generated through public transaction data) and minted by the network nodes subsequently. Transactions are included in the blockchain, subject to transaction aggregation which hides the actual transaction graph [41]. A typical transaction consists of input/output coins (commitments) and relevant range proofs (proving that values are positive), transaction fee and a signature. This functionality can be expressed in terms of the parameters in our model as summarised in Table 5.5.

Table 5.5: Modelling the variables of the Mimblewimble Scheme

Notation in our model	**Variables in the Mimblewimble scheme**
a_{pk}, a_{sk}	Public keys of coins/randomness of coins
V_{old}, V_{new}	Values of inputs/outputs
m	Transaction metadata
t_p	All public transaction data (public coin data, transaction signature, range proof, value commitments etc.)
t_s	Transaction history (before aggregation)

Adversarial capabilities

As it was the case with other currency schemes, the Mimblewimble state also is public. However, transactions hide the senders, recipients and the values while revealing only the commitments required to validate a given transaction by any third-party. The knowledge of the secret key (r) of the coins is required to produce a valid signature for a transaction, allowing the rightful owners to spend the coins. Hence we model the secret inputs and outputs as a_{sk_s} and a_{sk_r}. Then the adversary's knowledge of these is ψ_{sk_s} and ψ_{sk_r} as in all other cases. The public keys corresponding to the inputs and outputs are modelled as a_{pk_s} and a_{pk_r} and we model the adversary's knowledge corresponding to these as ψ_{pk_s} and ψ_{pk_r}. As the sender initiates a transaction by communicating with relevant recipients, when the adversary knows any of the secret keys, there is no anonymity among the participants in the transaction (i.e. when $\psi_{sk_s}, \psi_{sk_r} > 0$). Further, we use t_s to represent the transaction history since it is not publicly available after the aggregation. If the adversary has access to the transaction history, then there is an added advantage for the adversary in the game. Hence, we assume that the adversary does not have the knowledge of the transaction history for this analysis.

Analysis of anonymity

Consider the S-IND-PUBSRT-ACTIVE notion, which is parameterised by $(1_s000)_\omega$-$((3,0)_s, (3,0)_r,\ 3_v, 3_m, 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ (i.e. sender indistinguishability with the knowledge of public keys of senders, recipients and the transaction). Despite learning the value, metadata and public transaction data, the adversary is not able to distinguish between any sender, as secret keys are not known, thus making Mimblewimble secure against this adversary. However, any further leakage of information (i.e. private keys of recipients) would compromise anonymity. Similarly, the notion of S-ULK-NILS-PUBRT-ACTIVE (sender unlinkability with hidden senders, recipients known by their public keys with the knowledge of t_p) denoted by $(1_s000)_\omega$-$((0,0)_s,(3,0)_r,\ 3_v,3_m,1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ is also satisfied by implication. With a similar argument, we can show that it also satisfies R-IND-PUBSRT-ACTIVE and R-ULK-NILR-PUBST-ACTIVE. With value indistinguishability, we can see that it is secure in V-IND-PUBSRT-ACTIVE as the value is hidden similar to Zcash and Monero, and hence also secure in V-ULK-NILV-PUBSRT-FULL. Thus, we can conclude that the strongest anonymity notions that Mimblewimble satisfies is with respect to ALL-IND-PUBSRT-ACTIVE, denoted by the vector $(1111)_\omega$-$((3,0)_s,(3,0)_r,3_v,3_m,1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$.

5.4 Comparison of findings

While anonymity on the surface looks like an atomic notion, it is evident from the above analysis that our framework provides more nuance. In this section, we provide a comparison of anonymity achieved by different cryptocurrencies as we have unfolded in above case studies. In order to visualise these findings clearly, we deploy graphical interpretations as illustrated in figures such as Figure 5.7. Accordingly, we compare various anonymity aspects in these schemes in subsequent sections.

5.4.1 Sender anonymity

First we consider the anonymity levels achieved by the above currency schemes with respect to senders. In the case of sender indistinguishability (i.e. S-IND), the TTP scheme demonstrates the strongest level of anonymity as expected since no details of the transactions are public, and hence is secure against a S-IND-PUBS-FULL adversary. On the contrary, Bitcoin has the weakest anonymity as sender details are visible in the blockchain. Hence, Bitcoin does not have an acceptable level of anonymity with respect to senders. On the other hand, Zcash, Monero and Mimblewimble have much stronger levels of sender anonymity compared to Bitcoin, but still fall behind the much stronger TTP scheme. While Zcash is secure in S-IND-PUBST-ACTIVE, both Monero and Mimblewimble have slightly lower level of anonymity with respect to senders, but in different dimensions. For example, Monero is only secure in S-IND-NILMT-PUBS-ACTIVE as its strongest level due to the mixin information attached with the transaction, whereas Mimblewimble satisfies the highest anonymity with S-IND-PUBSRT-ACTIVE due to the pre-communication that takes place between the parties involved in the transaction. This comparison is illustrated in Figure 5.1.

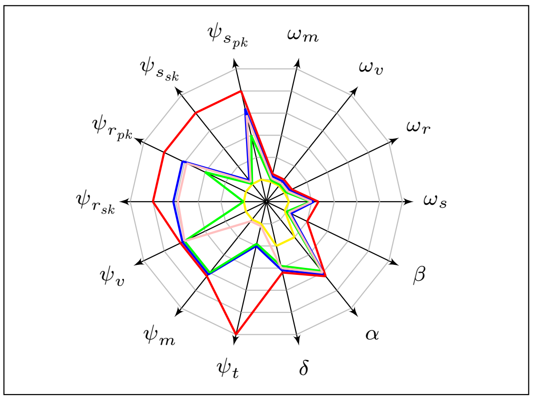


Figure 5.1: Sender indistinguishability [TTP(red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)]

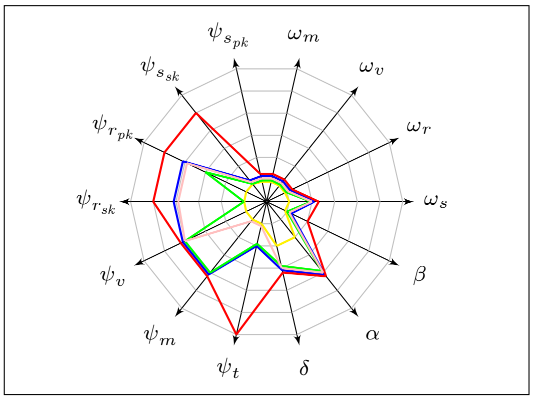


Figure 5.2: Sender unlinkability [TTP(red), Bitcoin (yellow), Zcash (blue), Monero (pink), Mimblewimble (green)]

Sender unlinkability notion in our framework closely correlates with the untraceability notion derived in [91] in the sense that for a given transaction, all possible senders are equally probable to be the corresponding sender. However, our definition focuses on a wider scope providing a broader view of this notion of anonymity. Figure 5.2 shows the S-ULK behaviour of these schemes and thus reveals that, S-ULK security follows the same pattern as in S-IND.

5.4.2 Recipient anonymity

With regards to the recipient anonymity, Zcash and Monero demonstrate similar levels of R-IND and R-ULK anonymity as evident from Figures 5.3 and 5.4. Mimblewimble however, satisfies a slightly weaker notion due to the pre-communication. TTP scheme has the highest level of recipient anonymity with Bitcoin at the other end having the weakest level.

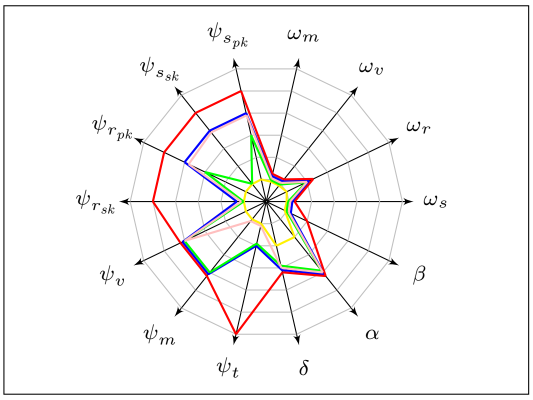


Figure 5.3: Recipient indistinguishability
[TTP(red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)]

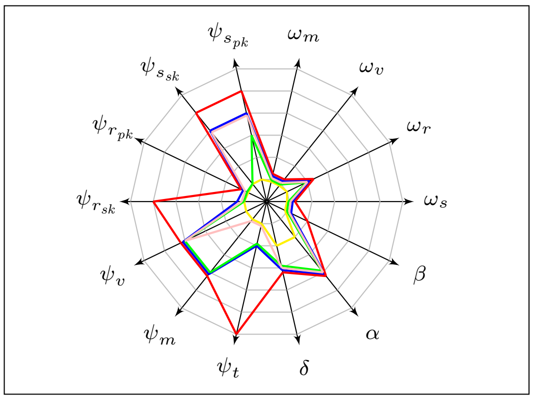


Figure 5.4: Recipient unlinkability
[TTP (red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)]

Our definition of recipient unlinkability corresponds to the property of unlinkability mentioned in [91]. While this definition does not mention about the effect on unlinkability from other factors, here we consider how other parameters from the adversary's capabilities could affect this attribute.

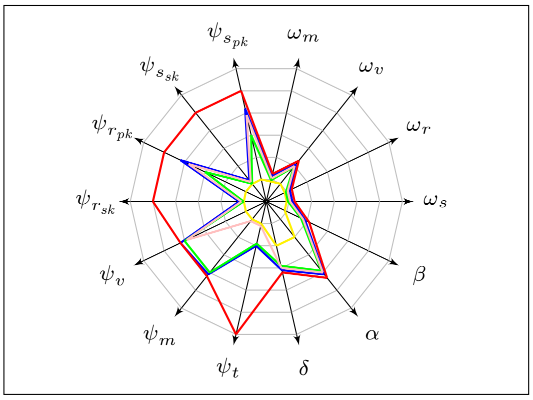


Figure 5.5: Value indistinguishability [TTP (red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)]

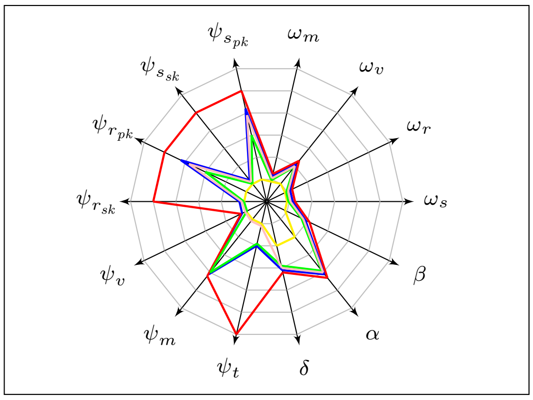


Figure 5.6: Value unlinkability [TTP (red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)]

5.4.3 Value anonymity

In the case of anonymity with respect to transaction value, Zcash, Monero and Mimblewimble demonstrate the same level of anonymity with V-IND-PUBSRT-ACTIVE due to the concealment of value through various techniques in these schemes (Figure 5.5). A similar analogy is visible with V-UNL notion as illustrated in Figure 5.6. Meanwhile, Bitcoin offers no anonymity with respect to value.

5.4.4 Highest level of anonymity

Figure 5.7 compares the schemes with respect to the highest level of anonymity achieved by them. Compared to the TTP scheme, the other four schemes show weaker anonymity prospects, proving that they do not meet the criteria for "absolute fungibility". As expected, Bitcoin demonstrates the weakest anonymity of all and is secure only in NIL-IND-NIL-VIEW. Conversely Zcash, Monero and Mimblewimble demonstrate considerably higher anonymity with minor deviations among them. Zcash shows the highest level while Mimblewimble shows weaker anonymity with respect to the participants of a transaction and in Monero, anonymity is compromised when details of the choice of mixins are leaked to the adversary. Yet, it should be noted that we considered only shielded transactions in Zcash for this analysis and unshielded transactions may hinder these anonymity levels significantly. Nevertheless, the knowledge of the randomness of the coins (i.e. $\psi_t > 1$) hinders the anonymity in all three schemes above.

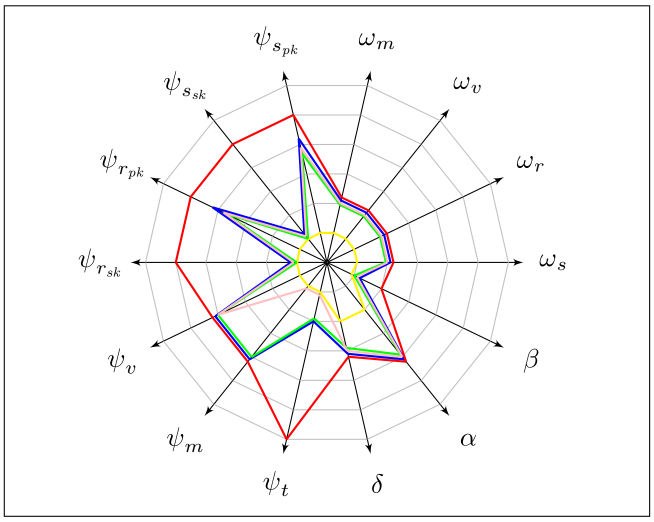


Figure 5.7: Maximal anonymity notions
[TTP(red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)]

5.4.5 Summary of findings

Above findings can be summarised in tabular form as presented in Table 5.6, which lists the strongest anonymity notion achieved by each currency with respect to sender, recipient and value anonymity.

Table 5.6: Comparison of anonymity

Currency scheme	Strongest notion satisfied by the currency scheme			
	Sender Anonymity	Recipient Anonymity	Value Anonymity	Maximal Anonymity
TTP	S-IND-PUBS-FULL	R-IND-PUBR-FULL	V-IND-PUBV-FULL	ALL-IND-FULL-FULL
Bitcoin	None	None	None	NIL-IND-NIL-VIEW
Zcash	S-IND-PUBSVT-ACTIVE	R-IND-PUBRVT-ACTIVE	V-IND-PUBSRT-ACTIVE	ALL-IND-PUBSRT-ACTIVE
Monero	S-IND-NILMT-PUBS-ACTIVE	R-IND-PUBRT-ACTIVE	V-IND-PUBSRT-ACTIVE	ALL-IND-NILMT-PUBSR-ACTIVE
Mimblewimble	S-IND-PUBSRT-ACTIVE	R-IND-PUBSRT-ACTIVE	V-IND-PUBSRT-ACTIVE	ALL-IND-PUBSRT-ACTIVE

5.5 Discussion

In this chapter, we have demonstrated how our anonymity framework is capable of providing a very precise landscape of anonymity in cryptocurrency schemes as opposed to various categorisations given in studies such as [3, 24, 44, 66]. In that connection, we have presented a qualitative evaluation of a subset of real world cryptocurrency schemes as our major contribution in this chapter, thus revealing how anonymity levels differ in diverse dimensions.

One may wonder why we need such granularity in modelling anonymity in the context of cryptocurrencies, yet the findings from our case studies show how a minute change such as varying one value along a single dimension, could drastically affect the level of anonymity. Hence, these outcomes emphasise the fact that claims for anonymity cannot be made lightly in the presence of such granularity.

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

Accordingly, our anonymity framework has been successful in producing a precise and exhaustive recount of true anonymity achieved by cryptocurrency schemes despite their diverse implementations.Furthermore, this analysis has demonstrated how anonymity can be visualised in multiple dimensions, yet in a unified view. We have published these findings in [4].

Chapter 6

Conclusion

Our main focus in this final chapter is to summarise our contributions while emphasising their research significance. In this respect, we look at the research objectives established in Chapter 1 and evaluate how well we have achieved these goals in our contributions together with future research directions.

As established at the outset, anonymity is key to the success of a cryptocurrency scheme. Hence, it is imperative that the extent of anonymity achieved by any cryptocurrency can be represented using a set of unified definitions. From a user's perspective, such a common framework for evaluating anonymity can facilitate meaningful comparisons between different schemes. On the other hand, from a designer's perspective, this type of framework could provide a useful guide to understand how different attributes influence anonymity, which allow them to refine anonymity in existing as well as new constructions. Accordingly, our overall research objective is to formulate a mechanism to achieve a uniform, fine-grained systematisation of anonymity modelling for decentralised systems such as cryptocurrencies and we summarise our contributions aligning with relevant research objectives as presented below.

Contribution 1 : Develop a means to commonly represent the functionality of different cryptocurrency schemes.

As discussed in Chapter 3, the foundation for a common anonymity framework consists of an abstract model of a cryptocurrency, which is representative of various cryptocurrency schemes. In order to address this, we constructed a generic cryptocurrency scheme by identifying the common functionality across different currency types. We defined a set of operations to represent such common functionality in terms of a set of entities (e.g. senders, recipients, transactions etc.) and typical operations including address and transaction creation, minting and adjudication. Further, our model caters for additional entities and functionality, and these can be defined based on the requirements of specific implementations.

In order to define the correctness and security of this model, we defined the correctness of functionality through a set of experiments which ensures expected operations in an honest setting. Further, we identified the security requirements for our model, which were captured through a set of game-based experiments. The security goals achieved through these game-based definitions allow us to define the security of cryptocurrencies with respect to issues such as transaction forgery and double-spending. For this purpose, we developed a comprehensive adversarial model, which looks at adversarial knowledge and power within a wide range of entities from zero knowledge to a fully active adversary.

The model we formulated here is capable of representing the functionality of many different constructions, as it is evident through the analysis presented in Chapter 5. In addition, the model itself enables one to understand the basic principles behind similar decentralised environments. The operations in our model, provide a clear understanding about the interactions between different entities within the system. Further, the correctness and security requirements provide useful insights into understanding how such schemes operate. More importantly, the adversarial model developed in this work demonstrates how the security of complex systems such as fully-decentralised environments can be modelled from first principles, thereby capturing a wide range of attacker scenarios.

It should be noted however that we did not consider the underlying communication mechanisms or the details of the consensus mechanism explicitly in our model, as they may be different in each construction. Yet, in some scenarios we may be able to model such additional information present in the communication layer (e.g. IP addresses) through the existing parameters in our model such as metadata.

Contribution 2 : Design a common framework to model anonymity aspects of different cryptocurrency schemes in terms of formal anonymity definitions

We addressed this issue in Chapter 4 as our main contribution in this work. As established in Chapter 1 and further supported in Chapter 2, the absence of a common mechanism to model anonymity in a unified manner limits our ability to compare this aspect of different cryptocurrencies. Hence, we constructed a common template in a game-based setting to address this by providing a qualitative recount of anonymity. We built this template using the adversarial model we defined in Chapter 3, along with a set attacker scenarios addressing different aspects of anonymity (eg. sender anonymity). Each scenario modelled in this game is a unique combination of adversarial knowledge, power and a security goal, related to a system entity (e.g. sender/recipient/transaction value, metadata). Each scenario is represented by a unique parametric vector representing the relevant adversarial capabilities and the entities being targeted.

This template results in a numerous number of adversarial settings, modelling different aspects of anonymity. We formalised these notions in terms of indistinguishability and unlinkability with respect to system entities and provided definitions for some useful notions. Further, we examined the relationships among these anonymity notions to understand their implications, equivalences and separations. In doing so, we formulated a set of theorems to formally establish these relationships. However, there may be other relationships between notions in addition to what we have presented, depending on specific circumstances. We have presented only those theorems that can be applied irrespective of the construction.

Accordingly, our framework 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 emphasise that these definitions arise naturally from considering the possible interactions between the adversary and the cryptocurrency. Indeed, our notions generalise many security notions familiar to cryptographers such as known vs. chosen plaintext, forward security, indistinguishability, active vs. passive adversaries, and so on. The fact that we consider all of these security dimensions simultaneously multiplies the number of definitions, but also allows us to meaningfully understand and compare the anonymity of systems that differ along multiple dimensions. As a consequence, this work provides a means for modelling anonymity in a precise and qualitative manner.

Contribution 3 : Evaluate and compare anonymity aspects of existing cryptocurrency schemes

In order to demonstrate how our framework can be applied in a real world setting, we have analysed a subset of existing cryptocurrency schemes (Bitcoin, Zcash, Monero and Mimblewimble and a TTP scheme representing an ideal scheme). We considered the indistinguishability and unlinkability of senders, recipients and transaction values for this analysis. We used the formal definitions and corresponding parametric vectors in order to evaluate and compare anonymity in these systems and provided graphical interpretations to facilitate the comparisons. Chapter 5 contains the relevant details.

The findings of this exercise indicate that some currency schemes may possess strong anonymity levels in one aspect, but very weak in another. Further, a minute change in one parameter could drastically affect the achievable extent of anonymity, which could not have been captured with existing anonymity notions.

As such, this analysis demonstrates how the proposed framework can be applied in real-world settings through findings presented. Our analyses outcomes allow us to directly compare the anonymity levels achieved by the currency schemes studied thereby revealing the potential of this framework as opposed to different categorisations presented in studies such as [24, 44]. In addition, comparisons between different systems highlights how minute variations of anonymity levels exist between very similar constructions, indicating the importance of fine-grained systematisation of anonymity. Constructing a formal proof for anonymity of an existing currency scheme in this framework would further establish the usefulness of this work.

Bibliography

[1] K. M. Alonso, "Zero to Monero," 2020. `https://src.getmonero.org/library/Zero-to-Monero-1-0-0.pdf`.

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

[3] N. Amarasinghe, X. Boyen, and M. McKague, "A survey of anonymity of cryptocurrencies," in *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW 2019, (New York, NY, USA), pp. 2:1–2:10, ACM, 2019.

[4] N. Amarasinghe, X. Boyen, and M. McKague, "The complex shape of anonymity in cryptocurrencies: Case studies from a systematic approach," in *International Conference on Financial Cryptography and Data Security*, pp. 205–225, Springer, 2021.

[5] N. Amarasinghe, X. Boyen, and M. McKague, "The cryptographic complexity of anonymous coins: A systematic exploration," *Cryptography*, vol. 5, no. 1, 2021.

[6] E. Androulaki and G. O. Karame, "Hiding transaction amounts and balances

in bitcoin," in *Trust and Trustworthy Computing* (T. Holz and S. Ioannidis, eds.), (Cham), pp. 161–178, Springer International Publishing, 2014.

[7] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *International Conference on Financial Cryptography and Data Security*, pp. 34–51, Springer, 2013.

[8] M. Apostolaki, C. Maire, and L. Vanbever, "Perimeter: A network-layer attack on the anonymity of cryptocurrencies," in *Proceedings of the 25th International Conference on Financial Cryptography and Data Security (FC'21)*, 2021.

[9] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Large-scale network attacks on cryptocurrencies," *arXiv preprint arXiv:1605.07524*, 2016.

[10] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi, "Anoa: A framework for analyzing anonymous communication protocols," *The Journal of Privacy and Confidentiality*, vol. 7, no. 2, 2017.

[11] A. Banerjee, M. Clear, and H. Tewari, "Demystifying the role of zk-snarks in zcash," in *2020 IEEE Conference on Application, Information and Network Security (AINS)*, pp. 12–19, IEEE, 2020.

[12] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better - how to make bitcoin a better currency," in *Financial Cryptography and Data Security. FC 2012. Lecture Notes in Computer Science*, vol. 7397, pp. 399–414, 2012.

[13] A. Berg, "The identity, fungibility and anonymity of money," *Economic Papers: A journal of applied economics and policy*, vol. 39, no. 2, pp. 104–117, 2020.

[14] A. Biryukov and S. Tikhomirov, "Deanonymization and linkability of cryptocurrency transactions based on network analysis," in *2019 IEEE*

European Symposium on Security and Privacy (EuroS P), pp. 172–184, June 2019.

[15] J.-M. Bohli and A. Pashalidis, "Relations among privacy notions," *ACM Trans. Inf. Syst. Secur.*, vol. 14, pp. 4:1–4:24, June 2011.

[16] S. Bojja Venkatakrishnan, G. Fanti, and P. Viswanath, "Dandelion: Redesigning the bitcoin network for anonymity," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 1–34, 2017.

[17] S. Bowe, A. Gabizon, and I. Miers, "Scalable multi-party computation for zk-snark parameters in the random beacon model.," *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 1050, 2017.

[18] F. Béres, I. A. Seres, A. A. Benczúr, and M. Quintyne-Collins, "Blockchain is watching you: Profiling and deanonymizing ethereum users," 2020.

[19] C. Cachin, A. De Caro, P. Moreno-Sanchez, B. Tackmann, and M. Vukolic, "The transaction graph for modeling blockchain semantics.," *IACR Cryptology ePrint Archive*, vol. 2017, p. 1070, 2017.

[20] R. Canetti, "Universally composable security: a new paradigm for cryptographic protocols," in *Proceedings 2001 IEEE International Conference on Cluster Computing*, pp. 136–145, Oct 2001.

[21] A. Chator and M. Green, "How to squeeze a crowd: Reducing bandwidth in mixing cryptocurrencies," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pp. 40–49, April 2018.

[22] T. Chen, Z. Li, Y. Zhu, J. Chen, X. Luo, J. C.-S. Lui, X. Lin, and X. Zhang, "Understanding ethereum via graph analysis," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, no. 2, pp. 1–32, 2020.

[23] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses," *ACM Comput. Surv.*, vol. 53, June 2020.

[24] M. Conti, S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, 2018.

[25] R. de Best, "Distribution of the biggest cryptocurrencies from 2015 to 2020, based on market capitalization," 2021.

[26] E. Duffield and D. Diaz, "Dash: A privacycentric cryptocurrency," 2014.

[27] E. Erdin, S. Mercan, and K. Akkaya, "An evaluation of cryptocurrency payment channel networks and their privacy implications," *arXiv preprint arXiv:2102.02659*, 2021.

[28] H. A. Estensen, "A comparison of monero and zcash," 2018. `koclab.cs.ucsb.edu/teaching/cren/project/2018/Estensen.pdf`.

[29] G. Fanti, S. B. Venkatakrishnan, S. Bakshi, B. Denby, S. Bhargava, A. Miller, and P. Viswanath, "Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, p. 29, 2018.

[30] S. Ferretti and G. D'Angelo, "On the ethereum blockchain structure: A complex networks theory perspective," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 12, p. e5493, 2020.

[31] G. Fuchsbauer, "Subversion-zero-knowledge snarks," in *Public-Key Cryptography – PKC 2018* (M. Abdalla and R. Dahab, eds.), (Cham), pp. 315–347, Springer International Publishing, 2018.

[32] G. Fuchsbauer, M. Orrù, and Y. Seurin, "Aggregate cash systems: A cryptographic investigation of mimblewimble," in *EUROCRYPT*, 2019.

[33] N. Gelernter and A. Herzberg, "On the limits of provable anonymity," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, (New York, NY, USA), pp. 225–236, ACM, 2013.

[34] M. Green and I. Miers, "Bolt: Anonymous payment channels for decentralized currencies," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, (New York, NY, USA), pp. 473–489, ACM, 2017.

[35] D. Hellwig, G. Karlic, and A. Huchzermeier, *Privacy and Anonymity*, pp. 99–121. Cham: Springer International Publishing, 2020.

[36] J. Herrera-Joancomartí, "Research and challenges on bitcoin anonymity," in *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance* (J. Garcia-Alfaro, J. Herrera-Joancomartí, E. Lupu, J. Posegga, A. Aldini, F. Martinelli, and N. Suri, eds.), (Cham), pp. 3–16, Springer International Publishing, 2015.

[37] L. Herskind, P. Katsikouli, and N. Dragoni, "Privacy and cryptocurrencies—a systematic literature review," *IEEE Access*, vol. 8, pp. 54044–54059, 2020.

[38] A. Hevia and D. Micciancio, "An indistinguishability-based characterization of anonymous channels," in *International Symposium on Privacy Enhancing Technologies Symposium*, pp. 24–43, Springer, 2008.

[39] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification," tech. rep., Tech. rep. 2016-1.10. Zerocoin Electric Coin Company, 2016.

[40] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification version 2020.1.3," tech. rep., Electric Coin Company, 2020.

[41] “Introduction to mimblewimble and grin,” Aug. 2020. `https://github.com/mimblewimble/grin/blob/master/doc/intro.md`.

[42] T. E. Jedusor, “Mimblewimble,” 2017. `https://scalingbitcoin.org/papers/mimblew-imble.txt`.

[43] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, “An empirical analysis of anonymity in zcash,” *CoRR*, vol. abs/1805.03180, 2018.

[44] M. C. K. Khalilov and A. Levi, “A survey on anonymity and privacy in bitcoin-like digital cash systems,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[45] N. Koblitz and A. J. Menezes, “Another look at” provable security”,” *Journal of Cryptology*, vol. 20, no. 1, pp. 3–37, 2007.

[46] A. Kumar, C. Fischer, S. Tople, and P. Saxena, “A traceability analysis of monero’s blockchain,” in *Computer Security – ESORICS 2017* (S. N. Foley, D. Gollmann, and E. Snekkenes, eds.), (Cham), pp. 153–173, Springer International Publishing, 2017.

[47] J. Lee, “Rise of anonymous cryptocurrencies: Brief introduction,” *IEEE Consumer Electronics Magazine*, vol. 8, pp. 20–25, Sep. 2019.

[48] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and l-diversity,” in *2007 IEEE 23rd International Conference on Data Engineering*, pp. 106–115, 2007.

[49] K. Li, R. Yang, M. H. Au, and Q. Xu, “Practical range proof for cryptocurrency monero with provable security,” in *Information and Communications Security* (S. Qing, C. Mitchell, L. Chen, and D. Liu, eds.), (Cham), pp. 255–262, Springer International Publishing, 2018.

[50] S. Linoy, N. Stakhanova, and S. Ray, "De-anonymizing ethereum blockchain smart contracts through code attribution," *International journal of network management*, vol. 31, no. 1, 2021.

[51] Y. Liu, X. Liu, C. Tang, J. Wang, and L. Zhang, "Unlinkable coin mixing scheme for transaction privacy enhancement of bitcoin," *IEEE Access*, pp. 1–1, 2018.

[52] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 3–es, 2007.

[53] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability." Cryptology ePrint Archive, Report 2018/472, 2018. `https://ia.cr/2018/472`.

[54] V. Martin, "Cryptocurrencies-reshaping the financial industry," *PaKSoM 2020*, p. 187, 2020.

[55] U. Maurer, "Constructive cryptography–a new paradigm for security definitions and proofs," in *Joint Workshop on Theory of Security and Applications*, pp. 33–56, Springer, 2011.

[56] F. K. Maurer, T. Neudecker, and M. Florian, "Anonymous coinjoin transactions with arbitrary values," in *2017 IEEE Trustcom/BigDataSE/ICESS*, pp. 522–529, Aug 2017.

[57] S. Meiklejohn and C. Orlandi, "Privacy-enhancing overlays in bitcoin," in *Financial Cryptography and Data Security* (M. Brenner, N. Christin, B. Johnson, and K. Rohloff, eds.), (Berlin, Heidelberg), pp. 127–141, Springer Berlin Heidelberg, 2015.

[58] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, (New York, NY, USA), pp. 127–140, ACM, 2013.

[59] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*, pp. 397–411, May 2013.

[60] A. Miller, M. Moeser, K. Lee, and A. Narayanan, "An empirical analysis of linkability in the monero blockchain," *arXiv preprint arXiv:1704.04299*, 2017.

[61] T. Miyamae and K. Matsuura, "Privacy analysis and evaluation policy of blockchain-based anonymous cryptocurrencies," *arXiv preprint arXiv:2012.10563*, 2020.

[62] N. Mohammed, B. C. M. Fung, and M. Debbabi, "Anonymity meets game theory: secure data integration with malicious participants," *The VLDB Journal*, vol. 20, pp. 567–588, Aug 2011.

[63] L. Morris, *Anonymity Analysis of Cryptocurrencies.* PhD thesis, Rochester Institute of Techology, 2015.

[64] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan, *et al.*, "An empirical analysis of traceability in the monero blockchain," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, pp. 143–163, 2018.

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

[66] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of cryptocurrency systems," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pp. 745–752, Dec 2016.

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

[68] S. Noether, "Review of cryptonote white paper," 2014. `https://downloads.getmonero.org/whitepaper_review.pdf`.

[69] S. Noether, "Ring signature confidential transactions for monero." Cryptology ePrint Archive, Report 2015/1098, 2015. `https://eprint.iacr.org/2015/1098`.

[70] S. Noether, A. Mackenzie, and the Monero Research Lab, "Ring confidential transactions," *Ledger*, vol. 1, no. 0, pp. 1–18, 2016.

[71] M. Ober, S. Katzenbeisser, and K. Hamacher, "Structure and anonymity of the bitcoin transaction graph," *Future Internet*, vol. 5, no. 2, pp. 237–250, 2013. Copyright - Copyright MDPI AG 2013; Last updated - 2014-07-30.

[72] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management." Http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, Aug 2010. V0.34.

[73] A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity—a proposal for terminology," in *Designing privacy enhancing technologies*, pp. 1–9, Springer, 2001.

[74] A. Poelstra, "Mimblewimble," 2016. `https://download.wpsoftware.net/bitcoin/wizardry/mimblewimble.pdf`.

[75] J. Quesnelle, “An analysis of anonymity in the zcash cryptocurrency,” Master’s thesis, University of Michigan-Dearborn, 2018.

[76] K. Rajendran, M. Jayabalan, and M. E. Rana, “A study on k-anonymity, l-diversity, and t-closeness techniques,” *IJCSNS*, vol. 17, no. 12, p. 172, 2017.

[77] F. Reid and M. Harrigan, “An analysis of anonymity in the bitcoin system,” in *Security and privacy in social networks*, pp. 197–223, Springer, 2013.

[78] D. Ron and A. Shamir, “Quantitative analysis of the full bitcoin transaction graph,” in *Financial Cryptography and Data Security*, (Berlin, Heidelberg), pp. 6–24, Springer Berlin Heidelberg, 2013.

[79] T. Ruffing and P. Moreno-Sanchez, “Valueshuffle: Mixing confidential transactions for comprehensive transaction privacy in bitcoin,” in *Financial Cryptography and Data Security*, (Cham), pp. 133–154, Springer International Publishing, 2017.

[80] T. Ruffing, P. Moreno-Sanchez, and A. Kate, “Coinshuffle: Practical decentralized coin mixing for bitcoin,” in *Computer Security - ESORICS 2014* (M. Kutyłowski and J. Vaidya, eds.), (Cham), pp. 345–364, Springer International Publishing, 2014.

[81] T. Ruffing, P. Moreno-Sanchez, and A. Kate, “P2p mixing and unlinkable bitcoin transactions.,” *IACR Cryptology ePrint Archive*, vol. 2016, p. 824, 2016.

[82] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, “Zerocash: Decentralized anonymous payments from bitcoin,” in *2014 IEEE Symposium on Security and Privacy*, pp. 459–474, May 2014.

[83] Y. Sei, H. Okumura, T. Takenouchi, and A. Ohsuga, “Anonymization of sensitive quasi-identifiers for l-diversity and t-closeness,” *IEEE transactions on dependable and secure computing*, vol. 16, no. 4, pp. 580–593, 2017.

[84] J.-W. Selij, "Coinshuffle anonymity in the block chain," *Dostupno na: http://rp. delaat. net/2014-2015/p77/report. pdf (20.8. 2018.)*, 2015.

[85] O. Shlomovits and I. A. Seres, "Sharelock: Mixing for cryptocurrencies from multiparty ecdsa.," *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 563, 2019.

[86] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs.," *IACR Cryptology ePrint Archive*, vol. 2004, p. 332, 2004.

[87] M. Spagnuolo, F. Maggi, and S. Zanero, "Bitiodine: Extracting intelligence from the bitcoin network," in *International Conference on Financial Cryptography and Data Security*, pp. 457–468, Springer, 2014.

[88] D. Stebila, "An introduction to provable security," July 2014. `https://www.douglas.stebila.ca/files/teaching/amsi-winter-school/Lecture-2-3-Provable-security.pdf`.

[89] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[90] Y. Tsukada, K. Mano, H. Sakurada, and Y. Kawabe, "Anonymity, privacy, onymity, and identity: A modal logic approach," in *2009 International Conference on Computational Science and Engineering*, vol. 3, pp. 42–51, Aug 2009.

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

[92] L. von Ahn, A. Bortz, and N. J. Hopper, "K-anonymous message transmission," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS '03, (New York, NY, USA), pp. 122–130, ACM, 2003.

[93] Q. Wang, X. Li, and Y. Yu, "Anonymity for bitcoin from secure escrow address," *IEEE Access*, vol. 6, pp. 12336–12341, 2018.

[94] D. A. Wijaya, J. Liu, R. Steinfeld, and D. Liu, "Monero ring attack: Recreating zero mixin transaction effect," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*, pp. 1196–1201, Aug 2018.

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

[96] D. A. Wijaya, J. K. Liu, R. Steinfeld, S.-F. Sun, and X. Huang, "Anonymizing bitcoin transaction," in *Information Security Practice and Experience*, (Cham), pp. 271–283, Springer International Publishing, 2016.

[97] M. Xu, C. Yuan, X. Si, G. Yu, J. Fu, and F. Gao, "Coinmingle: A decentralized coin mixing scheme with a mutual recognition delegation strategy," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, pp. 160–166, Aug 2018.

[98] Z. Zhang, W. Li, H. Liu, and J. Liu, "A refined analysis of zcash anonymity," *IEEE Access*, vol. 8, pp. 31845–31853, 2020.

[99] T. Zhao and T. Zhang, "Revisiting anonymity and privacy of bitcoin," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1275–1280, 2021.

[100] Y. Zhou, J. Wu, and S. Zhang, "Anonymity analysis of bitcoin, zcash and ethereum," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp. 45–48, March 2021.

[101] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, *Differential privacy and applications*. Springer, 2017.

[102] J. H. Ziegeldorf, R. Matzutt, M. Henze, F. Grossmann, and K. Wehrle, "Secure and anonymous decentralized bitcoin mixing," *Future Generation Computer Systems*, vol. 80, pp. 448 – 466, 2018.