Partial Anonymity In Central Bank Digital Currencies: A Survey

Enrique Aviña, Katharina Gschwind, Felipe Monsalve, Viktor Urvantsev

May 2021

1 Introduction

Central Bank Digital Currencies (CBDCs) are a new public service currently being explored by central banks around the world. As currently envisioned, CBDCs are systems that enable central banks to hold and manage individual user accounts with the purpose of offering users a digital equivalent or extension of a nation's fiat currency.

Although CBDCs are popular as an idea, current literature has not yet provided a standard definition, beyond the aforementioned description, on what exactly a CBDC is and what it would entail on a systems/individual protocol level. In light of this, in our project we summarize relevant related work as a survey where we evaluate initial CBDC proposals and related (non-CBDC) digital currencies. From these works, we synthesize an understanding of the necessary properties to a CBDC and justify their importance.

A major open question in current literature on CBDCs is that of privacy. The United States Federal Reserve, in a paper published on CBDCs, stated that "it will be essential to consider how privacy is respected and how personal data is protected in a CBDC arrangement."[1]. We expect many countries that value individual privacy to follow suit. At the same time, many governments around the world have made it explicit that any CBDC must comply with federal banking regulations, which protect against money laundering and other illegal activities. As such total privacy is not an option, we explore the concept of 'partial privacy', which is not well understood.

In this paper, we (1) contribute an understanding of the necessary properties of a CBDC in **Section 2**, (2) contribute an identification of which models of partial privacy are possible within a CBDC and what technical constraints they would be beholden to in **Section 3**, and lastly, (3) offer a survey of current digital currency work and initial CBDC proposals, particularly in detail in terms of how these protocols implement 'privacy' in **Section 4** and **Section 5**.

2 Parameters of a CBDC

Currently, literature on CBDCs do not provide a clear definition of entities or key properties a CBDC must strive for. In this section, we contribute to current CBDC literature by proposing a model for CBDCs, informed by our readings in digital currencies and CBDC proposals. We outline the concrete entities at play in our CBDC model and pose important design questions of each. We then present key properties of a CBDC system, some which are necessary, and others which might be desired. Our objective is not to propose a specific CBDC implementation, but to formalize entities and desired properties for further discussion.

2.1 Entities

2.1.1 Member

In the most abstract sense, a member is a participant in the CBDC who can influence the state of the CBDC through defined permitted actions detailed below. Concretely, a member would refer to regular people transacting with CBDCs. The actions we permit members to use are as follows:

1. Create an account with the central bank
2. Send money to another user in a transaction
3. Hold cash indefinitely and securely, unless a unit of currency is designed otherwise

There are various design decisions that we will delegate to the CBDC, these include decisions like the following:

1. Are member accounts strongly linked to a unique ID such as a SSN?
2. Is there a distinction between citizen (private) and corporate entities in the system?

2.1.2 Central Bank

The central bank is the central authority on a CBDC system. It's in charge of issuing currency and providing a way of transacting with this currency (whether centralized or not). Some of it's responsibilities in the system are:

1. Managing the CBDC system. This entails keeping some sort of record of currency in circulation (regardless of whether it's tied to an account), and facilitating a (centralized or otherwise) way of transacting with the currency.
2. Regulating the CBDC currency. Similar to physical currency, the central bank should be able to control the supply of digital currency in circulation by controlling the issuing of currency.
3. Providing an API for the private sector to build services overlaying the base CBDC network. For example, there may be space in the private sector to develop more sophisticated ”wallets” or CBDC protocols that could be open to private innovation/development.

There are many design decisions in the central bank's role in the CBDC. Among these, we have:

1. Should the central bank be in charge of approving all transactions that happen through a CBDC?
2. Should the central bank pay interest on issued currency?
3. Should the central bank delegate responsibility to the private sector or is the CBDC a wholly state-run system?

2.1.3 Auditors

Auditors are any entity who is permissioned to audit transactions. Their only action is to perform audits on the transaction ledger, which could be for regulatory purposes, tax collection purposes, or to detect criminal behavior, among others. Based on the implementation of CBDCs, auditors might need to have explicit permission from the central bank, or might be able to perform auditing on public information (potentially asking the central bank for more information in specific cases).

2.1.4 Private Sector

The private sector could be permissioned to adopt some or all of the responsibilities of the central bank. As is commonly done with current central banks, third parties can develop additional overlay services that interface with the CBDC. Different services that third parties could provide could include: hosting wallets, hosting ledgers, providing analytics, etc. These services are analogous to the functions of companies like Paypal.

2.1.5 Storage

Storage is not an active actor in the CBDC, but a key element with large implications on system design. In some way the monetary value of a CBDC must be held. Existing digital currencies propose token- and wallet-based systems. In a token-based system the base unit of monetary value is a valued token passed between members. In wallet-based systems, there is instead a system-wide agreement of how wealth is currently distributed across actors, and is globally agreed on over time throughout transactions.

There are many possible CBDC designs, some principal design decisions in developing one include:

1. How does the CBDC represent monetary value? With tokens or by associating value with wallets.?
2. If the CBDC is token-based, is the token stand alone? Can a token exist independent of a wallet? Or a member ID?
3. Does the government provide a wallet for every member? Do they host wallet accounts?
4. Does the government allow the private sector to develop wallets?
5. If token-based, how strongly identifiable is a token? How much of it's transaction history would a token hold?

2.2 Necessary properties of a CBDC

In this section, we identify the properties necessary and desirable for a CBDC informed by current proposals and discussions.

1. **Auditability + Compliance**: If a transaction is found to be fraudulent, can auditors determine which users were involved. A CBDC should not make it easy to launder money or conduct illegal trade. In one form or another, such fraud should be detectable.

2. **Stability**: A large entity should not be able to drastically change the CBDC network states unfairly. The system and ledger protocols should be such that any entity cannot unfairly change the value another member holds–canceling their money–or violate their privacy beyond the design of the system, even if this malicious entity holds control over a significant portion of the CBDC network.

3. **Integrity + Security**: This is primary to a CBDC. The protocol should be iron-clad in the sense that if one member sends a transaction to another it should deterministically go through as intended. There should be no option for man in the middle attacks to steal transactions.

4. **Scalability**: The CBDC should be able to manage a large amount of transactions and should be able to quickly settle transactions.

2.3 Potentially Desired properties of a CBDC

As can be inferred from 2.1, there are a lot of open ends as to how a CBDC can be designed. This section aims to document and detail specific properties which we believe should at least be considered when designing a CBDC. We don't intend to dictate how weak or strong each property should be. The properties follow below.

1. **Extensible**: Open to innovation and to either third parties or the central bank adding services in the future that aren't part of the central service.

2. **Transparent/ Verifiable**: Any user should be able to verify the correct state of the CBDC and ensure that no party is violating its rules.

3. **User-friendly + Inclusive**: Should be easy to use and accessible even for unbanked population and other vulnerable demographics.

4. **Privacy**: The need for privacy complicates technical design, since even the idea of what privacy is needed is not well defined. After exploring some of the tensions between desired properties, we will further explore the question of privacy, exploring different levels of privacy and discussing different protocols that help us achieve them.

2.4 Tradeoffs in CBDC Design

One can see how the potentially desired properties previously discussed can conflict with one another. As a result, any implementation necessitates some loss of one quality to allow for another. We outline some of these tradeoffs in terms of privacy below.

1. **Privacy vs. Transparency/ Verifiability**: The need for privacy directly conflicts with that of verifiability and integrity, as it is hard to keep all data about transactions private while at the same time making it clear for average users that the integrity of the system is intact. While crptographic protocols can protect the integrity of

data while keeping surprising levels of privacy, they might not inspire much trust, as hiding of information makes it very hard for a majority of users to understand how the system works, often limits who can verify the transaction ledger, and ultimately hinders transparency.

2. **Privacy vs. Compliance**: Privacy and compliance are, as we discuss through this paper, at odds. This is because the less information is made available about users, the less auditors can detect irregularities. A lot of privacy-focused systems hide essentially all information that isn't essential to proving the correctness of the system, which might empower bad actors to use it to commit illegal actions. These systems will almost certainly face pushback from regulators as they don't comply with financial regulation laws in many countries.

3. **Scalability vs. Extensibility**: The more a system is made extensible, it will naturally struggle with scalability, since extensibility comes with needing to support a public interface, and the development, security and performance costs that come along with that.

4. **Stability vs. Scalability**: Building a scalable and efficient system often relies on handing large amounts of control to large entities. While this might be acceptable for highly trusted parties, it conflicts with the stability of the system, since an attack on any of the controlling actors might result in an entity using the CBDC network unfairly.

Like these, there are many more tradeoffs that need to be considered in building CBDCs (or other digital currencies, for that matter). In this paper, we will focus on trade-offs around privacy/ anonymity of CBDC users. We will explore how different levels of anonymity can be achieved, and what tradeoffs must be made to achieve these standards of privacy, through the lens of existing protocols/ digital currencies.

3 What is Partial Anonymity?

Levels of anonymity, in the context of a transaction, describe how much information is revealed about transactions and to whom this information is revealed to. Partial anonymity refers to any level of privacy that is not full privacy or total non-privacy. In this section we present possible options of privacy settings across these various axes.

3.1 Privacy Properties

We identify through our literature review the following relevant "privacy properties" that would be key to implementing, or justifying omission of, in a CBDC. These are some of the key properties that make a system's privacy stronger, and we call a system "perfectly private" if it implements all of these properties, or "perfectly visible" if it implements exactly none of them. A "partially anonymous" or "partially private" system implements some, but not all of the follwing properties.

1. **No strongly linked membership**: A simple property to implement, no strongly linked membership refers to the property where a system does not hold a link between a currency holder account (addresses) and a personally identifiable ID. While often important for privacy, as in systems which make ledgers with visible transaction information public, it makes it harder (but certainly not impossible) to identify who an address in the system refers to, which mostly affects compliance.

2. **Metadata/ amount obfuscation**: This property refers to the property where transaction metadata (except for addresses involved in the transaction) are hidden and cannot be viewed by any party in the system. A system implementing this property trades off transparency and compliance for privacy, in that hiding metadata of a transaction makes it significantly harder to understand how currency moves through the system for a user or auditor.

3. **Address obfuscation**: Address obfuscation refers to the property where the addresses involved in a particular transaction are hidden and cannot be viewed by any party in the system. Like metadata/ amount obfuscation, implementing this property trades transparency and compliance for privacy.

4. **No recoverable transaction receipt**: This is a strong privacy property, that refers to no parties, including currency holder accounts, having no proof in the ledger that some accounts were involved in a transaction. This represents a significant tradeoff in user-friendliness and transparency for privacy, as it makes it difficult for users to even verify if their transactions are included in a ledger.

5. **Transaction hiding**: This property refers to having no visible change in the state of the system that's indicative of the occurrence of a transaction. Often achieved through a private ledger, transaction hiding is quite a strong privacy property which affects transparency and auditability.

6. **Non-revocable Privacy**: Non revocable privacy refers to a CBDC not allowing any party to revoke any of the implemented privacy properties defined above. Uncommon in centralized systems, this property has become decently common in cryptocurrencies, sacrificing compliance for privacy.

3.2 Partial Anonymity Levels

Based on the privacy properties mentioned above, we propose various models of partial anonymity. Importantly, this list is not exhaustive and provides a general categorization of the privacy models. While each level may not address *every* privacy property listed above, this list presents a potential categorization of CBDCs that highlights the most important properties of each privacy model.

1. **Transaction-Entities Privacy**: Transaction-entities privacy refers to a system that provides participants in a transaction with recoverable transaction receipts, relaxing the "no recoverable transaction receipt" privacy property. While it does not say anything about any other specific privacy property, the underlying assumption for such a categorization is that most (if not all) of the remaining privacy properties are implemented.

2. **Total Central Visibility**: Total Central Visibility refers to a system where guaranteed privacy policies can be revoked by some party in the system, such as a central bank. Once this privacy is revoked, this information can be visible to some subsection of the involved parties, which can be a group anywhere between a single party and everyone in the system. In short, total central visibility relaxes the "Non-revocable Privacy" property to allow a central authority to revoke privacy, while implementing most, if not all, of the privacy properties defined.

3. **Asymmetric Privacy**: Asymmetric privacy refers to a system that provides a significant number of privacy properties only to *some* parties in a transaction. For example, the identity of a party involved within a transaction might only be known in the scenario where it is the receiver, but not the sender of a particular transaction. Systems with asymmetric privacy do not relax a specific property; rather, they relax different properties in specific scenarios.

4 Existing CBDCs

CBDCs are a nascent area of research. As such, there are not many concrete CBDCs in use, and much is not concretely known about what system designs are best. To the best of our knowledge, the Digital Yuan is the only CBDC already under development and close to general release (with limited testing underway). In general, CBDC proposals and design discussions are just now starting to become publicly available. We present the Digital Yuan solution, the P-Hybrid CBDC proposal, and the University of Calgary Proposal, two notable, recent, CBDC designs, and what levels of privacy they implement.

4.1 Digital Yuan

The digital Yuan is currently still under development, and is currently being tested in multiple cities in controlled settings. Research papers detailing its implementation are not yet available. It seems, however, that some details are already known, especially in terms of the different properties that we would hope to see in a digital USD.

It appears that the People's Bank of China (PBoC) has developed a protocol for completely offline use of the digital Yuan. Digital Yuan may be loaded onto a physical card-wallet that can be scanned at registers for transactions, and no wifi or mobile network connection is needed to off- or onload digital yuan from the card. The director of the PBoC explains the digital yuan would 'provide more privacy than commercial payments products like bank cards, WeChat or Alipay, which are tied more closely to the banking system.' He goes on to explain that via digital Yuan, "the central bank could observe and monitor transactions taking place while the transacting parties would remain private. But it still allows the PBoC to analyze transactions to monitor crimes." He further explains that digital yuan wallets use "ID anonymization technology" such that personal information would be concealed from "counterparties, operating agencies and other commercial institutions."2

The PBoC claims that the digital Yuan needs a phone number to establish an account. Phone numbers in the PRoC do full under a real-name registration system, establishing strongly ID-linked CBDC accounts, and although anonymization protocols are used to hide an individual's identity to other actors on the network, user identity and transaction history are strongly known to the government central bank. This model, overall, is a textbook example of a model with total central visibility as a privacy model.

4.2 Canadian CBDC Proposal: University of Calgary

As part of the Model X project hosted by the Bank of Canada, a group of researchers at the University of Calgary submitted a proposal for a CBDC [5], which details their best CBDC design. In this proposal, the designers sought for multiple facets of privacy: a "defense-in-depth" strategy, where all components would have to fail in unison for an attack to be successful, mechanisms for attack recovery and protection mechanism evolution. This acts in combination with "strong privacy protection", which comes in the form of not only privacy of current and past transactions, but also immutability and integrity of the history, alongside updatability of algorithms as possible threats grow and change. These measures assist in creating a CBDC that has integrity, scalability and all kinds of privacy except for perfect privacy considered in its construction. This particular proposal structures their CBDC with the aid of financial intermediaries (FIs) and Cash+ tokens for offline work.

These FIs will be responsible for providing identity and fraud protection, supply of hardware and software 'wallets' for online and offline versions of the currency, eliminate double spending risk, and more. This allows for particular measures to be put in place that could allow for increased security, integrity, stability and compliance, since the distribution of power allows for each involved entity to do less, while also giving each entity the ability to manage their subsection with more information than a government has. The modularity and dependence on the FIs for providing the service leads to effectively any privacy level to come out of this scheme, even if the government itself keeps the information it houses under perfect privacy.

Cash+ tokens would come in different flavors, with different use cases and different tradeoffs. Some coins would only support single hops (only allowed to be used in one transfer), some could be limited by the amount of time it can exist, and various other designs could be used to give purpose to the token offline. Generally though, the paper insinuated an inverse relationship between the amount of things that can be done with a token and the risk of some sort of unintended use case occuring with it, like duplication or fraud.

4.3 P Hybrid

One notable, recent, CBDC proposal worth mentioning is the P-Hybrid Asymmetric Privacy CBDC proposal[7]. This was presented as an architecture in the Bank of Canada's recent Model X Challenge for CBDC architecture proposals, and was recognized as one of the top three proposed architectures.

In the P-Hybrid protocol, CBDC member accounts are strongly ID-linked, so that every citizen and corporate entity could have only one, publicly linked account. In this proposed CBDC, any given transaction is "asymmetrically" private. It is designed to keep the sender of a transaction perfectly private, while the receiver of a transaction is known to a subset of permissioned actors. Therefore, this protocol conforms to our definition of *Asymmetric privacy.* Receivers' identities are hidden while ensuring authenticity using zero-knowledge proofs, making it statistically impossible to determine which CBDC user is the sender for an incoming transaction to all. Receivers are "semi-public": although they are not hidden mathematically like senders are, the CBDC is designed to authorize a minimal set of actors the permission to see transaction receipts. These would be actors such as tax collection agencies. For the general public, both senders and receivers hold the *address obfuscation* property; but this property is broken for receivers by permissioned authorities for auditability purposes.

In addition to zero-knowledge proofs, the authors introduce the idea of "private tokens" as a means of hiding user identity. This is a hybrid approach between token-based and wallet-based digital currency models. Token-based digital currency is a common model that prevents against falsifying currency, since authorized token IDs are known, and doesn't require digital currency users to identify with a particular account, similar to traditional cash. The downside to token-based digital currencies is the potential for tracking individual coins and its history potentially leaking user information. The proposed CBDC does not model individual valid tokens of currency with identities. Instead, the authors propose "private tokens" that do not have a known identity or history to establish authenticity, and instead only represent monetary value, but bridge the gap between account-based digital currency and token-based digital currency. Using private tokens prevents the sender-account from needing to be known to establish authenticity, in this way the CBDC is not wallet-based. This proposal would provide *metadata/amount obfuscation.* To ensure authenticity, private tokens can only be sent to ID-linked accounts, i.e., that receipt of private tokens must be known by the permissioned entities, so the system is not entirely wallet-based. This prevents falsifying money as all tokens have established endpoints and the same coin cannot be double spent.

The authors explain that their model is designed on 'open protocols' based on blockchain. Any CBDC member can see that transactions are happening, although they can't know the details on a transaction, and have the API to verify transactions or the current CBDC state.

5 Existing Digital Currencies

While there's limited literature about CBDCs and their privacy models, recent years have seen widespread adoption of private (not central bank issued) digital currencies. As a result, we have seen quite a bit of literature appear on privacy-focused digital currencies. In this section, we will explore several such digital currencies and their approach to anonymity.

5.1 CryptoNote

CryptoNote[12], an application layer protocol which focuses on traceability draws many ideas from Okamoto and Ohta's "Universal Electronic Cash" paper[13], the first first ideal untraceable electronic cash system which claims that in order for a system to be fully anonymous, "the relationship between the user and his purchases must be untraceable by anyone." CryptoNote claims that to satisfy this requirement, which we defined as perfect privacy above, a digital currency must have the following properties:

1. **Untraceability**: For each incoming transaction, all possible senders are equiprobable. This definition arises from CryptoNotes' use of ring signatures.

2. **Unlinkeability**: For any two outgoing transactions, it is impossible to prove they were sent to the same person.

Monero, a digital currency, uses the CryptoNote protocol, which seeks to provide full anonymity as defined by aforementioned section. It uses an obfuscated public ledger, meaning anyone can send or broadcast transactions, but no outside observer can tell the source, amount or destination. CryptoNote uses a variety of techniques to provide privacy: one-time ring signatures, stealth addresses (through the Dual-Key Stealth Address protocol), and a non-interactive zero-knowledge proof protocol called Bulletproofs, and "Dandelion++" transaction broadcast propagation which passes many transactions through a single intermediate node, obscuring IP addresses.

A deeper discussion about ring signatures is warranted, as standard "vanilla" ring signatures are vulnerable to "temporal analysis" attacks. Ring signatures to consider include group signatures, adhoc group signatures, linkeable ring signatures, and traceable ring signatures, each of which provide different privacy guarantees.

5.1.1 Digital Signatures

Group signatures help hide the identity of a particular signer. Ring signatures offer perfect anonymity while the other signature schemes offer some level of partial anonymity. Below we will describe the characteristics of various group digital signature schemes; implementers of a CBDC can choose a signature scheme depending on their particular needs.

1. **Group signature**: Traditional group signatures have various good privacy properties, namely: only members of a group can sign messages and the receiver of a group signature can verify that it is a valid signature, but cannot discover which group member made it. However, group signatures use a group manager, which is a central trusted authority who has the power to revoke anonymity of a singer. This scheme can be used to implement *total central visibility*, because individual signers are hidden publicly until the group manager reveals their identities.

2. **Ring signatures**: Ring signatures are similar to group signatures but differ in two key ways: first, no trusted coordinater is needed, groups are formed ad-hoc. Secondly, there is no way to revoke the anonymity of an individual signer. There is also no way of knowing if any two signatures were produced by the same signer. This digital signature scheme can be used to implement any privacy model where the identity of one or more parties is hidden.

3. **Traceable Ring Signatures**[13] Traceable ring signatures have the aforementioned privacy properties, but add the 'linkeable' property which allows one to determine whether any two signatures were produced by the same member. This is particularly useful in offline e-cash systems or systems where we need to limit "excessive" anonymity. We note that anonymity is still preserved as long as a signer does not sign on two different messages with the same group. Therefore, this scheme could be used in a revocable privacy model, where one of our described privacy models could be broken by fradulent user activity or by the user providing a de-anonymizing "view key". In normal operation, a signer wouldn't sign two transactions with the same group and *perfect*, however identifying this action could point to fraudulent activity.

5.1.2 Stealth Addresses

Monero generates one-time stealth addresses which hide the address of the recipient using the Dual-Key

Stealth Address Protocol (DKSAP). This stealth addresses is generated by two pairs of cryptographic keys, namely a 'scan key' pair and a 'spend key' pair, and computes a one-time payment address per transaction, as detailed below:

1. The receiver has two private/public key pairs `(s, S)` and `(b, B)`, where `S = s·G` and `B = b·G` are 'scan public key' and 'spend public key', respectively. Here G is the base point of an elliptic curve group.

2. The sender then generates a ephemeral (one-time) key pair `(r, R)`, where `R=r·G` and transmits it along with the transaction.

3. Both sender and received compute a shared secret c using the ECDH: `c=H(r·s·G)=H(r·S)=H(s·R)` where `H(·)` is a hash function.

4. The sender uses `c·G+B` as the 'ephemeral' destination address for sending the payment.

5. The receiver actively scans the blockchain and checks where some transaction has been sent to the address `c·G+B=(c+b)·G`. If the sender finds a transaction, they can use their private key `c+b`. We note that `c+b` can only be computed by the receiver.

In this scheme, private spending keys are not exposed or stored; combined with the use of changing ephemeral addresses, the DKSAP protocol protects users privacy against network analysis attacks. Because stealth addresses hide the *receiver* identity and randomize receiver addresses, they can be used to achieve our *address obfuscation*, *traceability*, and *membership privacy* properties.

5.1.3 Dandelion++

The Dandelion++ protocol is a network layer anonymity solution which improves upon Bitcoin's P2P network security. The origin of a transaction message and its IP address can be mapped by third-party observers if they control enough nodes. An early attempt to combat this is a technique called *diffusion*, where each node spreads transactions with exponential and independent delays to its neighbors to mitigate de-anonymization attacks; however, recent studies [3] have proven this technique does not offer adequate anonymity protection.

Dandelion++[6] improves upon this through its two phases:

1. **Stem Phase**: In the stem phase, rather than a node broadcasting a transaction to all of its connected peers, it relays the transaction message through a privacy graph to a single random peer based on an algorithm. Subsequently, that node then only transmits the transaction message to another single peer, and the pattern continues until eventually (and randomly) one of the nodes broadcasts the message in the typical format of diffusion to the rest of the network.

2. **Fluff Phase**: After the stem phase, once a single node broadcasts the message using the diffusion method, the transaction message is propagated to a majority of nodes in the network quickly. However, it becomes much more difficult to trace back to the original node since the transaction message was transferred to many individual nodes through a privacy graph before being propagated in a manner that would allow an observer to map it to a single node. Instead, an observer could only map the spread of transactions back to the several nodes where the message was transferred in the stem phase, thus muddling the actual identity of the sender. In effect, this is abstractly similar to how a ring signature obfuscates the actual signer of a transaction.

Therefore Dandelion++ is used to meet *address obfuscation* and *traceability* properties by randomizing the broadcast patterns of transactions.

5.2 ZCash

ZCash is the implementation of the research done in the ZeroCash [9] and ZeroCoin [8] papers. It is similar to Bitcoin in a few ways due to the decentralized blockchain structure that is used by both, and ZeroCoin was actually built on top of Bitcoin before ZeroCash made the concept into its own separate currency. The biggest difference is that ZCash has an even greater emphasis on privacy than Bitcoin, and allows users, both senders and receivers, to maintain any transaction's secrecy during its creation and fulfillment, and during its time on the ledger after the transaction is completed.

ZCash uses decentralized autonomous payment schemes (DAP schemes), which utilize the more well known zero-knowledge proofs and zero-knowledge succinct non-interactive arguments of knowledge (zk-snarks), to do so. As a result, this doesn't only make transaction details safe from other users of the system, it also makes the transactions safe from even the hosts of the ledger. As a result, it makes for a private and secure platform that protects users from essentially any actor in the system. It's important to note that, even with the strength of the privacy, there are ways for transactions to be revealed. Each user of the platform has a viewing key, which can be used to reveal information about the transactions if compliance or auditing is necessary for the user's transactions. This additional layer of revocable privacy places this cryptocurrency's privacy level under Transaction-Entities Privacy umbrella. It is important to note that the power to reveal is left entirely up to the user, and secure, offline storage of this key yields relative safety from attacks using this medium.

The currency's most recent innovations tackle illegal coin replication (due to zk-snarks and the fact that the coins can't be tracked on the ledger publicly), where they added a multi user key shard generation scheme, known as a "Ceremony" in initial iterations, so that if even one person acts responsibly and destroys their key shard after a transaction, the private key that was used can't be rebuilt. This is one example of how the currency has evolved since its initial release, and it goes to show that this design is one that's still evolving, even to threats that are well known.

5.2.1 Zero-Knowledge Proofs

It's worth having a short explanation of the security primitives that ZCash uses, so this subsection should be a brief explanation on what a zero-knowledge proof is. A zero-knowledge proof is a proof where the verifier can verify some sort of secret or proof without having knowledge of what that proof is. The three primary properties of a zero-knowledge proof are completedness (an honest verifier can be convinced by an honest prover), soundness (an honest verifier cannot be convinced by a dishonest prover with non-negligible probability) and, of course, zero knowledge (the verifier has no idea what the thing they are proving actually is).

A brief example follows. Let's say a server holds and verifies passwords. If the server holds a password in plaintext and queries a user to send their password and compare against their plaintext, it isn't zero knowledge, since the server has the password in plaintext. Let's now say that the server holds some hash of the password, and doesn't have access to a decryption oracle. To prove, a user would send their password, hashed, to the server, and the server would compare the two ciphertexts. If they match, the user gets access, if not, then the user is denied. This proof is zero-knowledge because the proof has completedness (user can convince the server that they know the password by sending them this matching hash), soundness (no one can really guess the password with any high probability, not even the verifier) and zero-knowledge (the verifier doesn't know what the password is, only the hash. It doesn't know what function is used, what other parameters it took, and can't reasonably replicate it without some other breach of security).

5.2.2 Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (ZK-Snarks)

ZK-Snarks build upon the qualities of a zero-knowledge proof, but add in non-interactiveness and succinctness. The biggest reason to do this is for scalability and efficiency, since with zero-knowledge proofs, both the prover and the verifier have to be online at the same time in most basic cases. In order to make something non-interactive, both sides should minimize communication with one another (thus comes succinctness), so that the process of verification doesn't rely upon multiple pings across the network. The example in the zero-knowledge proofs section actually satisfies this well enough, since there only has to be one message sent from the prover and verifier each.

ZK-Snarks have three functions as a part of their general operation. It has a generator, which is essentially an initialization function. It puts out two public keys, a prover key pk and a verifier key vk. It has a prover function which essentially turns the input text that needs to be proven into a message which reveals nothing about the original input, and takes in three parameters: the input text, the prover key published by the generrator, and a publicly available random input of some sort. The final function is a verification function, which takes in a proof, the virtual key, and the same random input from before, and returns True or False based on whether or not the proof passes or fails verification respectively.

5.3 MimbleWimble

Mimblewimble is a blockchain protocol that focuses on providing increasing privacy and scalability to blockchain-based ledgers.[11] To do so, Mimblewimble offers a protocol for signing and verifying transactions that doesn't reveal addresses or quantity transacted, and merges transaction blocks, leveraging cryptography to remove past transaction data without compromising security (a technique known as transaction "cut-through"). Like Monero, Mimblewimble's transaction structure originates from work on "confidential transactions", verifying transactions through Pedersen commitments, which hide amounts being sent in a transaction. It also uses properties of elliptical curve cryptography to combine two signatures, which hides individual addresses involved in a transaction. To date, the two best known implementations of Mimblewimble at alt-coins Grin and Beam, though Litecoint developers have completed the code for a Litecoin Mimblewimble Extension Block upgrade earlier this year.

5.3.1 Confidential Transactions [10]

Mimblewimble, along with other blockchain protocols, use confidential transactions to hide information about transactions, allowing verifiers to confirm that transactions don't create money out of thin air without knowing the values involved in the transaction. Based on elyptic curve cryptography, every input or output value v in a transaction on Mimblewimble is obscured by embedding $r * G + v * H$ instead, where r is a blinding factor (and is also used as a private key), and G and H are generator points in the eliptic curve. We call $r * G + v * H$ a Pedersen commitment, and because of cryptographic properties of elyptic curves, neither v or r can be deduced. However, because of cryptographic properties of elyptic curves we also know that if inputs v_1 and outputs v_2 are equal:

$$\begin{aligned}
&(r_2 * G + v_2 * H) - (r_1 * G + v_1 * H) \\
&= (r_2 - r_1) * G + (v_2 - v_1) * H \\
&= (r_2 - r_1) * G + 0 * H
\end{aligned}$$

Since blinding factors r are used as private keys, they can't be shared by the recipients, who generate a secret r for the output of a transaction (anyone else who knows r for this output could spend the output amount in another transaction). Therefore, senders and recipients collectively build a (schnorr) signature for the excess value $(r_2 - r_1)$. Now, with a commitment to the excess value, a verifier can check that the Pedersen commitments of the outputs minus the commitments of the inputs are equal to the commited excess value, without having any idea of the values in a transaction.

By including rangeproofs (zero-knowledge proofs which allow us to prove a value is within a certain range without revealing any more information about the value) in a transaction kernel, a verifier can additionally check that the outputs to a transaction can't be negative. This is important because we assume output values, are positive, but negative ones would not be detected with the confidential transaction protocol above, since the outputs minus the inputs could still be equal to 0.

Confidential transactions, as used in Mimblewimble and other blockchain protocols, provide *metadata/ amount obfuscation* and *address obfuscation*, and make a strong push towards no strongly linked membership (since there are no permanent addresses). Because of this, it makes tracing transactions to the network very hard, and since blinding factors (which double as private keys) can be randomly chosen at each transaction, there isn't much data that can be used to link an account to individuals.

5.3.2 Cut-Through

An important part of the Mimblewimble protocol is aggregation and cut-through. The basic idea is that, like transactions, blocks consist of input and output commitments. To validate a block, it's enough to "add all the output commitments together, then subtracts all input commitments, k*G values, and all explicit input amounts times H"[11], which is essentially the same as validating a single transaction. Similarly, we can combine transactions from different blocks in the same way, resulting in a valid transaction with input and output commitments. In this case, there will be outputs commitments that exactly match input commitments, which we can delete while still having a valid transaction. This idea becomes very powerful when we consider that starting at the genesis block, every single spent output commitment has a matching input commitment. What remains is explicit inputs, along with unspent outputs, which can be validated as if it was a single transaction. This is significant in terms of scalability, since it means the ledger grows with users as opposed to transactions, but also provides no recoverable transaction receipts and non-recoverable privacy (since it cuts significant data from the ledger).

6 Conclusion and Future Work

In this paper, we proposed a formal definition for a CBDC, and formalized necessary and desired properties. We then presented various privacy properties that CBDCs can exhibit, and proposed models of privacy that implement these privacy properties, defining partial anonymity and describing which privacy properties current and proposed digital payment systems exhibit. By exploring the partial anonymity space and proposing different policies that can be implemented with currently deployed methods, we contribute to literature on CBDCs. Before a CBDC is released to the public, policy surrounding the privacy of CBDCs must be developed. It is our belief that policy and implementation bounds will dictate the efficacy and level of privacy of a CBDC.

Future work on CBDCs should investigate methods of authenticating actors to ensure that permissioned entities like auditors don't overstep the bounds of their privacy limits. Secondly, future developments could look to unite the techniques used by current digital currencies to propose a protocol that achieves all/some of our privacy properties.

eom □

References

[1] Cheng, J., Lawson, A. N., & Wong, P. (2021). Preconditions for a general-purpose central bank digital currency. *FEDS Notes, 2021* (2839). Link

[2] Kwon, O., Lee, S., & Park, J. (2020). Central Bank Digital Currency, Inflation Tax, and Central Bank Independence. *SSRN Electronic Journal*. Published. Link

[3] Biryukov, A. (2014, June). Deanonymisation of clients in Bitcoin P2P network. Link

[4] MIT DCI. *(2020, June 12)*. MIT DCI response to Bank of England CBDC discussion paper. MIT Media Lab. Link

[5] Kyoung Jin Choi, Ryan Henry, Alfred Lehar, Joel Reardon, and Reihaneh Safavi-Naini. University of Calgary's *A Proposal for a Canadian CBDC: Model X Final Report.*

[6] Fanti, G., Venkatakrishnan, S. B., Bakshi, S., Denby, B., Bhargava, S., Miller, A., Viswanath, P. (2019). Dandelion++. ACM SIGMETRICS Performance Evaluation Review, 46(1), 5–7. Link

[7] Tinn, Katrin and Dubach, Christophe, Central Bank Digital Currency with Asymmetric Privacy (February 11, 2021). Available at SSRN: https://ssrn.com/abstract=3787088 or Link

[8] I. Miers, C. Garman, M. Green and A. D. Rubin, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 397-411, doi: 10.1109/SP.2013.34.

[9] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, Madars Virza, *Zerocash: Decentralized Anonymous Payments from Bitcoin*, proceedings of the IEEE Symposium on Security and Privacy (Oakland) 2014, 459-474, IEEE, 2014

[10] *Mimblewimble - Grin Documentation.* (2021). MimbleWimble. Link

[11] Tom Elvis Jedusor, *Mimblewimble*, 2016. Retrieved May 9, 2021 from Link

[12] Saberhagen, Nicholas. (2013, August 17). CryptoNote v2.0. Link

[13] Fujisaki, E., & Suzuki, K. (2007). Traceable Ring Signature. Public Key Cryptography – PKC 2007, 324–337. Link