Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Sjur Brekke Espedal & Dennis Aleksander Janzso

Design Choices for Offline Transactions in a Norwegian Central Bank Digital Currency

Master's thesis in Communication Technology and Digital Security
Supervisor: Iwona Windekilde
June 2022

NTNU
Kunnskap for en bedre verden

Sjur Brekke Espedal & Dennis Aleksander Janzso

Design Choices for Offline Transactions in a Norwegian Central Bank Digital Currency

Master's thesis in Communication Technology and Digital Security
Supervisor: Iwona Windekilde
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Title:	Design Choices for Offline Transactions in a Norwegian Central Bank Digital Currency
Students:	Sjur Brekke Espedal & Dennis Aleksander Janzso

Problem description:

Central Bank Digital Currency (CBDC) is a growing field of interest among central banks with the potential to disrupt the digital payment provider market. Norges Bank is researching CBDC in the context of national currency in Norway for retail applications. Existing research points to the offline capabilities of a CBDC as essential for the contingency perspective. The offline functionality of a digital currency relies on architectural design considerations, as well as a robust and secure protocol. Offline transactions pose an array of unique challenges, among which are the double spending problem, fair exchange, and anonymity. These problems can not necessarily be solved in an incomplete network, and subsequently, any solution is based on the best effort principle.

In order to design an offline compatible solution, the architecture of the system, the financial risk, and the technical limitations of a possible protocol must be analyzed for optimal choices. Such an analysis will look to existing cryptocurrencies, distributed ledger technologies, cryptographical techniques, secure hardware, and other proposed approaches to the offline problem. Following such an analysis, a system architecture and an offline protocol, designed for a Norwegian context, will be derived and tested.

Date approved:	2022-01-26
Supervisor:	Iwona Windekilde, IIK
Collaborator:	Norges Bank

Abstract

This thesis examines the possibilities and challenges of offline functionality in a Central Bank Digital Currency. A best effort suggestion for a protocol for offline transactions in a Norwegian CBDC context is presented. The protocol is based on existing approaches to the offline problem and gathered requirements from relevant stakeholders, including Norges Bank. The proposed protocol, security mechanisms, and architectural considerations are tested in a CBDC transaction simulation framework developed in Python. The problems relating to offline transactions have not been solved in this thesis, but mitigation mechanisms that can reduce the associated risks are outlined. These mechanisms and spending limits are shown to be effective against the attacks studied, mainly double spending from a sophisticated adversary. The simulation results confirm that increased availability through system architecture can decrease the risks associated with offline functionality.

Sammendrag

Denne oppgaven undersøker mulighetene og utfordringene forbundet med offline funksjonalitet for Digitale Sentralbankpenger (DSP). Et best effort forslag av en protokoll for offline transaksjoner i en norsk DSP kontekst presenteres. Protokollen er basert på eksisterende tilnærminger til offline problemet og formulerte krav fra relevante aktører, inkludert Norges Bank. Den foreslåtte protokollen, sikkerhetsmekanismene og arkitektur aspekter er testet i et transaksjonssimuleringsrammeverk for DSP utviklet i Python. Utfordringene knyttet til offline transaksjoner er ikke løst i denne oppgaven, men mekanismer som kan minimere tilknyttede risikoer er foreslått. Resultatene viser at disse mekanismene og beløpsgrenser er effektive mot de simulerte angrepene, hovedsakelig double spending fra en sofistikert fiendtlig aktør. Simuleringsresultatene bekrefter at økt tilgjengelighet gjennom systemarkitektur kan redusere risikoen forbundet med offline funksjonalitet.

Preface

This Master's thesis concludes our 5-year Master of Science degree in Communication Technology and Digital Security at the Norwegian University of Science and Technology (NTNU).

We would like to thank our supervisor, Iwona Windekilde, for her guidance through our research. Her experience and understanding have been essential throughout the project. Furthermore, we would like to thank the rest of the Department of Information Security and Communication Technology at NTNU for all the help provided throughout our studies.

Next, we would like to thank Peder Østbye at Norges Bank for his contributions and interest in the project and for sharing his expertise in the research area. We would also like to thank Lasse Meholm for taking the time to be interviewed and providing crucial insights into the context of our work.

Finally, we would like to thank our friends and families for their love and support. Particularly, we would like to thank Isabelle Roberts for taking the time to proofread our thesis.

We look to the future and hope the knowledge we have acquired will help build a better tomorrow.

Sjur Brekke Espedal
Dennis Aleksander Janzso
Trondheim 2022

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AML Anti money laundering.

API Application Programming Interface.

BFS Breadth First Search.

CA Certificate Authority.

CAP Collaborative Attack Prevention.

CBDC Central Bank Digital Currency.

CFT Countering the financing of terrorism.

DAG Directed Acyclic Graph.

DAP Distributed Attack Prevention.

DLT Distributed Ledger Technology.

HMAC Hash based Message Authentication Code.

IT Information Technology.

KYC Know your customer.

MCP Maritime Communications Partner.

MCX Mission Critical Services.

NB Norges Bank.

NFC Near Field Communication.

NICS Norwegian Interbank Clearing System.

OPS Offline Payment System.

P2P Peer-to-peer.

PBoC People's Bank of China.

PKI Public Key Infrastructrue.

PoS Proof of Stake.

PoW Proof of Work.

SDN Software Defined Networking.

TA Trusted application.

TCP Transmission Control Protocol.

TEE Trusted execution environment.

TLS Transport Layer Security.

VDES VHF Data Exchange System.

WWW World Wide Web.

Chapter 1
Introduction

Central banks from all over the world are currently researching Central Bank Digital Currency (CBDC) [AKAS+21]. Norges Bank has established a research group to explore the possibilities, challenges, and limitations of a Norwegian CBDC project. Central to such a payment solution is its potential to replace and supplement the current official payment solution, banknotes, and coins, commonly referred to as cash, as well as existing digital solutions. To replace such a system, a payment system able to perform offline transactions and work in any situation would be essential features to be a fully-fledged solution. The main problem with offline payments is the impossibility of a system that is consistent, available, and distributed on a large scale. This problem, known as the CAP-theorem, leads to the possibility of double spending attacks and ways to spend and invent money without central approval. This thesis will cover efforts to detect and mitigate these problems.

1.1 Context of CBDC

Central banks are researching and piloting CBDC projects to provide frictionless payments to the citizens [ACE+20]. CBDC digital tokens of value represent a claim on the central bank, in contrast to bank deposits, which represents a claim on a private bank. Cash usage in Norway is declining and is measured to be 1.3% of the mainland GDP in 2020 by [Ban21c]. The cash usage decline further motivates the project, as the government has no alternative to cash for payments. There are projects both in retail CBDC and wholesale CBDC. Retail is for the general consumer, and wholesale is for interbank payments. The research of Norges Bank, and hence, this project, is focused on the retail CBDC and offline capabilities of such a system.

1.2 Necessity of offline transactions

To meet the existing solution of cash, i.e., a physical national fiat currency, offline availability is essential. With cash, it is hard to counterfeit, transactional privacy is provided, and there is no respendability of the physical object. In many ways, it solves every issue in a contingency situation, given that it is sufficiently hard to counterfeit and sufficiently easy to detect a fraudulent banknote. For a digital solution, this is proven to be a hard task without creating additional risk of accepting counterfeit transactions [ACH21]. With the decline of physical cash, there are few alternatives to payments in a situation where the connection to the private banks would be compromised. Norges Bank lists three main objectives for the development of a CBDC if cash no longer should be a viable available option [Ban18].

1. Need for a credit risk-free alternative to bank deposits.
2. Independent contingency capable solution for the ordinary electronic payment systems.
3. Legal tender functionality.

Of the three research objectives, the first two apply to the offline transaction necessity. A risk-free alternative would suggest a secure alternative, and an independent contingency capable solution would suggest the scenario defined as an offline context in this thesis.

1.3 Objectives

The overall goal of this thesis is to propose a solution capable of providing desirable offline functionality for a Norwegian retail CBDC and analyze its functionality with regard to gathered requirements and security through simulation. To determine what functionality and characteristics should be included and prioritized in the design of such a solution, requirements will be gathered from relevant sources and stakeholders. Relevant literature on CBDCs and offline specific challenges and considerations will be reviewed. Architectural design choices and considerations will then be analyzed in a Norwegian context.

A best effort suggestion for a protocol will be proposed based on existing solutions, and the requirements gathered. Furthermore, a simulation framework will be developed to test the proposed protocol and architecture considerations. The simulation will primarily aim to estimate the effectiveness of the proposed solution from a security perspective. Lastly, the results and findings will be presented and discussed. The simulation framework, the protocol, with its security enhancements,

and the results of the testing and following analysis is the final outcome of the thesis. The overall goals of the thesis are further defined as four specific objectives, as seen below.

1. Formulate requirements for offline functionality of a Norwegian retail CBDC.
2. Determine the impact of architectural design choices on offline functionality and security.
3. Design a protocol for offline transactions based on the relevant requirements and existing solutions.
4. Develop a simulation framework to test the designed protocol's security and other offline considerations of interest.

1.4 Limitations

The research in this thesis focus mainly on the design possibilities for an offline solution. The project is limited in scope to researching, designing, and validating the results of a prototype protocol in a constructed simulation environment. Time and computational constraints limit the possibility of large scale testing of the simulation framework. The architecture and protocol implementation will not be tested beyond the simulation. A user interface will not be created, and there will be no interaction with real users. The design will not be tested on secure hardware, as the access to develop secure applications is limited. Technical aspects beyond offline capabilities and their considerations fall outside the scope of this thesis.

Economic, legislative, political, and social aspects of an offline payment system will not be directly analyzed. The question of who is liable in such a system will be left open, as well as other financial and legislative questions. Only the economic aspects necessary for analyzing the risk and consequences will be covered in this thesis.

1.5 Methodology

This thesis aims to complement Norges Banks' research into offline solutions to retail CBDCs. The objective of the thesis is to propose a viable option for an offline enabling solution in the context of a national CBDC and evaluate it. The solution will be presented by designing an artifact to solve the problem. This is a design problem as the problem is to design an artifact, satisfying requirements from the stakeholder by reaching several goals. The research methodology will be based on the design science framework for design problems as defined by [Wie14].

The first step in designing a solution is to derive the requirements from the stakeholders. The stakeholder requirements define the goals, limitations, and boundaries of the research, which will be collected through the best practices as defined by [WB13]. The first phase of the project will therefore be gathering requirements through reading the reports of central banks, researching existing solutions, and conducting primary research through an interview with Norges Bank. The main focus of this phase is to derive the required functionality for the users, and the system as a whole, with regard to technical possibilities and limitations. An important aspect of this phase is the translation from general requirements to the applied situation of designing the offline protocol and the planned artifact creation.

The next phase is to derive the system architecture of the potential system. The architecture will define the limits and preconditions for a protocol and its design. This phase includes research on the possibilities of using existing solutions to implement a system for offline payments fulfilling the derived requirements. The results from this phase will be used for the validation of the protocol design, as well as provide the fundamentals for an overall system architecture design.

The protocol can then be derived based on the requirements, the preconditions set by the architecture, and the research on previous protocols. The protocol and architecture will be reviewed, and the theory presented will be validated by the simulation. The simulation framework will be developed to objectively test the performance of the protocol under contingency situations and how well the protocol fulfilled the requirements. From the results of the simulation and discussion about the work, a review of the solution can be performed.

The methodology will be divided into three main steps from a research perspective. The steps will be problem comprehension, solution design, and solution evaluation.

Problem comprehension Building a foundational understanding of the offline problem in the CBDC context is essential for the success of any solution. First, the stakeholders' requirements will be derived. This will be a process of analyzing the research of the primary stakeholder, Norges Bank, and the research of similar projects conducted by other central banks. The research is then complemented by interview(s) with Norges Bank to answer uncertainties and specifics to the project requirements. After this research, the derived requirements should be validated by the stakeholders to ensure the correctness of the resulting requirements. The requirement gathering is a continuous cycle, as the comprehension of the problem, the stakeholders' needs, and the feasibility of the solutions could change during the project.

Solution design, implementation, and validation The solution design is dependent on the comprehension of possible solutions and require thorough research

of state-of-the-art literature on similar problems. The design of the architecture is dependent on the existing infrastructure and the possibilities for new infrastructure in Norway, as well as the general population's access to hardware, e.g., smartphones, smartwatches, cards, or other gadgets. Each design iteration should be consistent with the current requirements and, if need be, update the requirements and revalidate with the stakeholders. The protocol will be based on the existing research. Any enhancements to existing research should be validated by the simulation and external reviewers. The implementation will be of minimalistic nature with a proof of concept application in focus, with minimal focus on user-centric features and metrics. The simulation framework possibilities will be discussed and applied for the testing of the protocol in the given architecture environment. The framework will be written in Python. The process will be a cyclic workflow with increments to gradually converge towards a satisfactory result. The validation of the implementation will be from the experiment runs of the simulation.

The simulation model The simulation model will be described in the thesis, with the basis in a Markov model to model states and network failures. The architecture will define the relation between users and network nodes in the system. Both the architecture and the protocol will depend on the simulation context, requirements, and goals of the simulation. An analysis of the dependability of the system should be conducted, as the network would be an extension to the system with a potential for network failure. The impact of the security measures implemented in the protocol should be measured by simulation experiments.

Solution evaluation The evaluation of the proposed solution will be done by evaluating the results of the experiments. How well the results of the experiments in the simulation fulfill the requirements will give a good indication of the success of the solution. Any tradeoffs of the implementation will be discussed in relation to the system and the requirements. The proof of concept testing will be a validation by itself of the possibility for secure offline transactions. The solution needs to be analyzed in the context of the simulation, the purpose of the simulation, and the real-world applicability. The evaluation will also discuss existing and any new risks introduced from the system perspective.

An overall presentation of the steps described above as applied to the project can be seen in Figure 1.1.

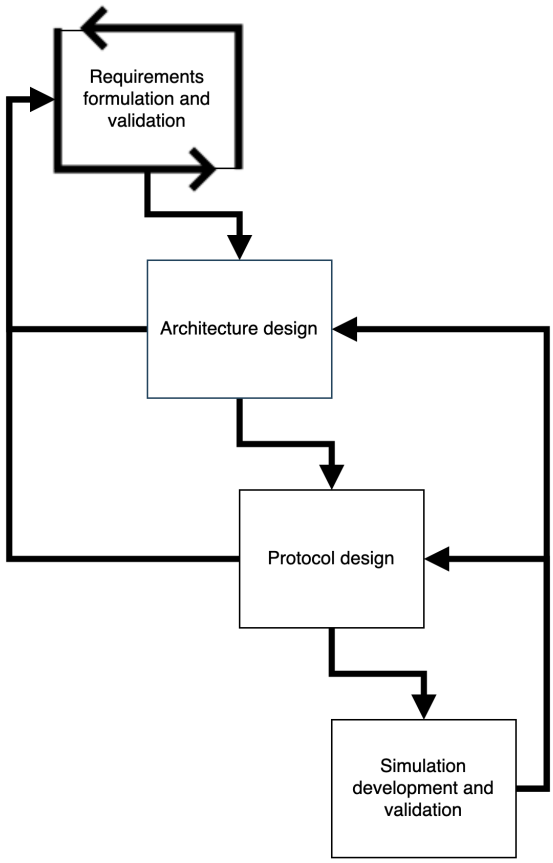


Figure 1.1: Flowchart for overall project progression based on methodology steps

1.6 Outline of Thesis

This thesis is divided into seven chapters, including this introduction.

Chapter 2 – Background presents the theoretical background for the thesis. Existing research, projects, and necessary theoretical material are presented to understand the context and the topics of discussion.

Chapter 3 – Requirements presents the previous requirements of similar projects, discusses and compares the requirements, and conclude with an interview with Norges Bank.

Chapter 4 – Architecture design presents the fundamental possibilities and discusses the principles of centralization and the simulation context for protocol implementation.

Chapter 5 – Protocol Design presents our solution to the protocol design and describes its sub-protocols in depth.

Chapter 6 – Simulation Framework presents the simulation framework developed to validate the performance of our protocol design using the fundamentals from the architecture design.

Chapter 7 – Results and discussion presents the results of our findings and discusses how well the solution fits the requirements.

Chapter 2
Background

This chapter forms the basis of understanding for the challenges posed by offline functionality in a CBDC context, as well as their potential solutions. Existing literature covering a variety of relevant topics will be presented. Some existing CBDC projects will be presented, along with approaches to designing such currencies. The threat model posed by offline functionality will be presented. Primarily we look at the double spending problem, which poses an intractable problem on offline transactions. Some key aspects of digital transactions will be presented with existing technical approaches, forming a foundational problem comprehension for offline functionality. Furthermore, this chapter will review a selection of existing approaches to the offline problem. Solutions specifically applied to CBDCs, as well as more general approaches will be presented. Secure hardware, a popular approach to the problem is analyzed. Lastly, this chapter covers networks and relevant theory to strengthen the offline problem comprehension and provides a foundational understanding of networks relevant for the simulation of transactions.

2.1 Central bank digital currencies

In this section, some existing CBDC projects will be presented. The overall status and designs of the projects will be outlined, along with any potential approaches to offline functionality. Key design considerations of CBDCs, account based versus token based solutions, will also be introduced.

2.1.1 Existing CBDC projects

Bank of international settlements

Regarding offline transactions, the Bank of international settlements' report [oCBoJ+20] discusses the possibility of a distributed system. *"... improving user convenience by making offline and peer-to-peer payments possible would necessitate additional safeguards to counter the risk of fraud, since security features and centralized controls*

(e.g. to "lock" stolen funds or query suspicious transactions) are more difficult to implement on a distributed system. A centralized ledger with a cap on allowable offline transactions is a potential compromise." The article discusses the architecture and ledger design and argues for a centralized approach to provide safeguards both to lock out fraudulent users and lock stolen funds.

e-CNY

One of the more advanced CBDC projects as of yet is the Chinese e-CNY. Developed and issued by the People's Bank of China (PBoC), e-CNY is already in use in multiple Chinese cities through various trials [GAN21]. e-CNY does not rely on blockchain technology but rather on a centralized account based system. The currency is built on a two-tier operational system, with the PBoC issuing e-CNY to authorized commercial banks. The commercial banks handle the exchange and transactions using the currency. [oC21]

The PBoC state in their report that the e-CNY has implemented offline functionality [oC21]. It is, however, unclear exactly how this is achieved from a technical perspective. Various independent analyses take a closer look at this claim and speculate to what extent the currency supports offline transactions and how it is achieved [GAN21][JL21]. The skepticism to the PBoCs claims stems from the intractability of the double spending problem without resorting to a trusted third party, which will not be naturally available in an offline setting. Because of this, it is assumed that the e-CNY offers a sort of pseudo-offline functionality where some transactions are possible in an offline setting, but there are limitations on amounts and number of transactions for each wallet. Different wallets may offer different limitations as a means of managing credit risk. It is also suggested that the e-CNY may utilize forms of local trusted third parties that can mediate exchange in certain "disconnected" scenarios. [GAN21]

The PBoC have taken various steps to improve the offline functionality of e-CNY. Among these are measures taken to reduce the frequency and likelihood of, particularly large scale and long-lasting, offline scenarios, such as natural disasters and communication outages. One such measure is building multiple, geographically separated high-availability data centers facilitating the system. The two-tier structure itself also provides increased availability of the system as its components, and therefore the risk of system failures, are more distributed. [oC21]

2.1.2 E-krona

The Swedish central bank, the Riksbank, is developing a CBDC known as E-krona. The project has gone through two stages of pilot testing, with the latest starting in February of 2021. The results and subsequent discussions of these tests are

outlined in [Rik21] and [Rik22]. The E-krona used in the pilot utilizes a token based architecture based on the Corda Distributed Ledger Technology (DLT) platform. This allows for storage of tokens both in online accounts registered with Riksbanken or other payment providers and locally on a user's device.

Tokens can be transferred to a user's local device wallet. The overall system, i.e., the ledger, then considers these tokens as consumed, and they are now, in effect, offline currency. The tokens can then be spent in offline transactions between users, and the receiver of any such token can redeem it when reconnected to the ledger. The E-krona pilot thereby offers users offline functionality.

It is worth noting that the E-krona does very little to combat the challenges associated with offline transactions, like double spending. The system does not rely on or require users to have any form of secure hardware, meaning all tokens potentially are subject to tampering. Neither does it have any other protection nor security mechanisms. Instead of preventing double spending and other tampering, the E-krona accepts that all offline transactions carry risk and subsequently accepts that no transaction can be final until a connection to the ledger can be reestablished.

Along with the issue of double spending the Riksbank discusses other challenges and considerations surrounding offline transactions. The bank states that limitations on offline balances and transaction volumes might be appropriate to reduce risks associated with such transactions.

The E-krona is built using a two-tier model. As stated, the E-krona utilizes a distributed ledger, allowing approved participants, i.e., third parties, in the network to verify transactions through the blockchain. The Riksbank is operating on a design that relies on third parties to provide services and abstractions on the DLT. These third parties can operate inside the Riksbank network or separately. In the so far conducted phases of testing, it seems the third parties, or intermediaries, are not able to mint tokens or directly process transactions. In other words, it is only the Riksbank itself that can append blocks to the blockchain.

2.1.3 Token based and account based solutions

There are primarily two main models to be considered for a CBDC system: Token based and account based solutions [Ban18]. Hybrid models are possible, and combining the solutions for different purposes could be an important alternative [Ban18]. The main difference between the two solutions is how the balance is stored. Token based (store-of-value) systems are based on the transfer of a payment object between the payer and the payee and depend on the ability of the payee to verify the payment object. Account based systems, on the other hand, depend on the ability of the

participants to verify the identities of the account holders, to be able to link the participants to their accounts and their account history. [KR09]

In an account based solution — also called a register-based solution – the balance is stored in a remote storage. If the user loses their authentication device, the balance is still linked to the user account. Thereby, the user's balance will not be affected as it is stored in a remote storage. Transferal of funds in this model must be done by consolidation with the remote system, and it is safe to assume the flow of funds is traceable. Deposit, withdrawal, and creation of value in the system must all be completed with communication to the remote system. [Rik17]

In a token based solution — also called a value-based solution — the value is stored within the token itself. In many ways, it is similar to today's cash system. If the device storing the tokens is destroyed or lost, the balance is lost. Transferal can be made between parties without consolidation with a remote system. Consolidation with the remote system is still necessary to verify the authenticity of the tokens and validity of the transactions, as any offline transaction can be susceptible to a digital equivalent of counterfeiting or fraudulent transactions. A token based solution can be made anonymous or traceable, depending on the design. [Rik17]

A comparison of the two models from the Riksbank's report [Rik17] can be seen in Table 2.1. From the table, we can see that credit risk is mitigated in both models, as the credit risk is the claim the user has on the money. In a commercial bank, this claim is to the bank, and as the bank has a default risk, there is a credit risk associated with the claim. The risk of a central bank to default is negligible. The financial risk of counterfeit money or fraudulent transactions exists in every model that supports offline transactions, as explained in Section 2.3.2.

In most CBDC projects, it is required to have a central remote party able to record balances and transactions. In an offline situation, the system is without connectivity to the third party for a period of time. For an account based system, an offline transaction would be the same as allowing deferred settlements. The offline transactions can be compared to a digital check. The payee gives the payer credit, and there is an expectation that at a later period of time, one party will connect to the third party and inform the central register of the transaction. The transaction thereby introduces credit risk to the payee. This kind of credit risk is common today for payments with commercial bank money using offline allowed payment cards. [ACH21]

For a token based solution, the risk involved is that the central bank could lose the control of the money supply. To limit this risk, technology and regulations could limit amounts and how many times a token could be respent without consolidating with the central third party. [ACH21]

Properties	Cash	E-krona –value-based	E-krona –register-based	Commercial bank money
Credit risk	No	No	No	Yes*
A store of value	Yes	Yes	Yes	Yes
Payments in real time	Yes	Yes	Yes	No**
Offline function	Yes	Yes, possible for card-based solutions	No	No
Physical presence required	Yes	Yes for card, no for app	No	No
Usability	Works without technical aids	Requires e.g, a card reader or special smartphone technology	Can be managed via apps or online	Can be managed via apps or online

*The deposit guarantee does, however, include holdings up to and including EUR 100,000.
**The exception is payments within the same bank and Swish.
***Cards can have an offline function.

Table 2.1: E-krona properties compared to cash and commercial bank money. [Rik17]

Token based solutions appears to have the same potential for offline payments as an account based solution. [ACH21]

2.2 Offline threat model

In this section, the threat model posed by the offline setting will be introduced. The primary threat relevant in this thesis is that of double spending attacks. These come in different forms depending on system characteristics. Other types of attacks, such as forging attacks, may also be relevant in the offline context and will be briefly discussed.

The double spending problem is an inherent challenge for all digital tokens of value, including currency. The challenge arises from digital tokens' fungibility. Any digital item is a virtual construction that can be copied or forged. Digital tokens of value can, therefore, potentially be spent in multiple transactions by a cheating actor. In physical currency, this is typically not considered an issue as the payee can be certain that the token of value, in this case cash, she receives has not already been given to someone else. However, this cannot be achieved for digital currency without

introducing mechanisms that prevent cheating actors from copying and subsequently double spending a token. [CC21]

The traditional approach to the double spending problem in digital exchange of value is to introduce some central authority that can verify, on behalf of any payer, that the funds have not already been spent. Such a scheme protects any payee from fraudulent transactions. This role is usually taken by banks and payment providers in existing digital monetary systems. Another approach to double spending is found in cryptocurrencies. Originally proposed in [Nak08] cryptocurrencies can, utilizing a DLT, prevent double spending by ensuring that all parties have access to a complete and universally agreed-upon transaction history.

Both the aforementioned approaches to the double spending problem relies on the user being connected, either to a central authority or a distributed network, to avoid being the victim of a double spending attack. In an offline setting then, where such a connection cannot be guaranteed, other additional mechanisms and protections must be included to prevent double spending attacks.

In the literature, multiple types of attacks that more or less fall under the umbrella of double spending are found. There is, of course, the traditional case where the same token of value or the same transaction message is reused in multiple transactions from one payer to one or more payees. In systems that rely on transfers of value between a client and a bank or server, which will be studied in Section 2.4, another type of double spending arises. In such systems, transactions with the bank, such as deposits or withdrawals out of an account, can be replayed. The protocol outlined in [CGK$^+$20] specifically looks at a case where a deposit transaction is sent from a server to a local wallet to add funds to that wallet. In this case, without any protection against such attacks, such a message can be replayed to double deposit the money. Other similar attacks may be possible depending on the structure and rules of any given system or protocol. In this thesis, we will generally refer to all such attacks involving the reuse of transactional messages or receipts to generate illegitimate funds as double spending attacks. A distinction between the different kinds of double spending will be made only where the differing characteristics and mechanisms of such attacks are relevant.

Forging attacks are attacks where funds are created illegitimately either through the unauthorized creation of new funds or by copying existing funds. Such attacks result in a malicious actor obtaining more funds, either through holding more tokens of value or having a higher account balance than she should legitimately have. Forging attacks have many similarities with double spending in that both typically involve copying tokens of value, but forging attacks are not necessarily performed through transactions. [CGK$^+$20]

Any aspect of a CBDC, including offline functionality, will likely be subject to resourceful and sophisticated attackers. Furthermore, as abuse of a CBDC can provide significant monetary gains, incentives for potential attackers are large. Therefore, the threat model posed by offline functionality should assume highly capable attackers in this context.

2.3 Digital transactions

There are a variety of challenges and considerations posed by digital transactions in general. In this section, we will review some of these, as well as their existing and potential solutions. Namely, cryptocurrencies and their underlying blockchain technology, the fair exchange problem, and public key infrastructure will be presented. These topics form a fundamental understanding of digital transactions, how related problems can be solved in an online state, and how traditional approaches may not be available in an offline setting.

2.3.1 Blockchain and cryptocurrencies

A blockchain is a method of storing data in such a way that, if executed correctly, the content and order of content appended to the blockchain is unaltered since its original entry. Blockchain thus represents a way of storing data that, after adding it to the blockchain, is immutable – unchanged, unmodified, and permanent. [DP17]

A blockchain achieves this immutability by adding data in blocks that include a hash of the previous block, such that all blocks are dependent on the block before it. When hashed sequentially in this manner, the blocks, once added to the chain, cannot be altered without altering all following blocks, as any change would propagate through the chain through the previous hash fields. [DP17]

Given the immutability characteristics of blockchains, blockchain technology has a range of use cases and applications. Whenever it is necessary to maintain a permanent and unalterable record, blockchains provide an elegant solution. The most prevalent such application is cryptocurrencies. First suggested in [Nak08], along with the popularization of blockchains themselves, cryptocurrencies utilize the data structure for storing and sharing transactional data. The sum of all the transactions provides the current state of the system and, consequently, the balance of all accounts. A blockchain used for this purpose is often referred to as a ledger. [MA18]

Along with the blockchain as an immutable record, cryptocurrencies utilize Peer-to-peer (P2P) networks to form a distributed ledger. This ledger can be verified by any desiring party to ensure that a potential counterpart in any transaction has the necessary funds available. Cryptocurrencies utilize a consensus algorithm

to ensure that the network is in agreement on the current state of the system by applying a protocol describing how new blocks should be added to the chain and by whom. [BMZ18]

The most widely used consensus algorithms are Proof of Work (PoW) and Proof of Stake (PoS), but there exists a range of alternatives and variations. PoW selects who in the network will add, or mine, the next block in the chain by making competing actors, or miners, solve a puzzle. A common application of this, like the one found in Bitcoin, makes the hashing of each block the puzzle by requiring the resulting hash of a new block to be smaller than a given number. As the hashing algorithm used is *puzzle-friendly*, meaning one cannot easily achieve the desired output by manipulating the input, solving the Bitcoin PoW puzzle requires guessing a large number of inputs and calculating their resulting hashes. The first miner to solve the puzzle for the given block is collectively entrusted with adding the next block to the chain, containing new transaction data. [BMZ18][Nak08]

PoS similarly selects one from a pool of miners to mine the next block in the chain. Instead of the miners racing to solve a puzzle, one is typically selected at random from a set of miners with a proven stake in the system. Such a stake can be having more than a certain amount of the given currency, but other mechanisms also exist. In cryptocurrencies, a reward is typically provided to the miner of a block to incentivize participation in the consensus mechanism. [BMZ18]

2.3.2 Fair exchange problem

The fair exchange problem is central to any digital exchange of value between two or more parties. In any such transaction, each party must trust that the other(s) will fulfill their obligation according to a predetermined agreement. As a simple example, the transfer of a good owned by an agent A to an agent B in exchange for a payment from B to A. If A delivers the good before B has completed its payment, A will have to trust B to fulfill its obligation to pay after the fact. Agent A will, in this case, have no recourse if B decides not to fulfill its promise. Similarly, in the opposite example, B may find itself having to trust A to deliver the good after receiving the payment. This, in short, is known as the fair exchange problem. More generally, a *fair exchange* can be defined as

An exchange is fair if at the end of the exchange, either each player receives the item it expects, or neither player receives any additional information about the other's item. [Aso98]

This problem is largely circumvented in traditional transactions, where the two parties are typically in close proximity to each other and able to visually confirm the other party's ability to fulfill their obligation to the agreed-upon transactions. The

real life, physical scenario typically also offers a possible recourse for both parties, should one prove not to honor the agreement after the fact. However, in digital transactions where the parties are typically not in close physical proximity and may have very little knowledge of the other party, this is not as trivial. [PVG03]

The formal specification of a fair exchange is a state based system involving two parties, P and Q, whom both have an item, i_P and i_Q. Both parties have a desired description of what they want and expect from the exchange, d_Q and d_P, and the parties will be satisfied if their description matches the received item ($desc(i_Q) = d_Q$ for P and $desc(i_P) = d_P$ for Q). This definition assumes that there exists some description function $desc()$ that is capable of mapping an item's properties in a way that can be evaluated against the parties' desired descriptions. [PG+99]

For an exchange to be considered fair it must have at least the following properties

- Effectiveness: If both participants behave correctly and the transaction is completed P will have i_Q that satisfies $desc(i_Q) = d_Q$, and Q will have i_P that satisfies $desc(i_P) = d_P$.
- (Strong) Fairness: If the transaction is completed, both parties will have their desired results (as in the above point), or neither party will have gained any additional information about the other party's item.
- Timeless: The transaction will be completed at some point. At this point the state of the system is final, or any changes of the state will not change the level of fairness in the transaction.

The fair exchange problem is usually solved by introducing some mutually trusted third party. It is in fact proven that a fair exchange, as defined above, is impossible without a trusted third party [PG+99]. Such a trusted third party can give assurances to both parties in the transaction that the other party is not cheating in some way. There exists a variety of possible implementations of a trusted third party in existing fair exchange protocols.

The fair exchange problem is an essential consideration in the design of any offline solution for a CBDC. In an online state, any actor intending to take part in a transaction can rely on the system, e.g., a DLT or a central database, as a trusted third party. The system can confirm whether or not the payer has sufficient funds to complete the transaction and offer a recourse procedure in the event the payee does not uphold their obligations. In the offline scenario, however, where one or both of the involved parties cannot consult the system, ensuring a fair exchange becomes more challenging.

2.3.3 Public key infrastructure

A Public Key Infrastructrue (PKI) allows for distribution and use of certificates with security and integrity [Wei01]. With a hierarchical structure in the trust model, the entities can trust other entities with certificates signed by a central authority, a Certificate Authority (CA). A public key infrastructure is based on asymmetric cryptography and provides a foundation for secure communication between strangers [AL99]. By using a trusted third party, the CA, two participants can trust that the other part is whom they say they are (authentication).

RSA is a cryptosystem for securing communication, as first outlined by Rivest, Shamir, and Adleman in their patent [US4]. In the RSA system, there are two keys, a private and a public key. Using the private key, one can decrypt ciphertext encrypted with the public key and vice-versa. The encryption is performed as follows:

$$c = m^e \mod n$$

In the equation, c is the ciphertext, m is the message, e and n is the public keys of the receiver of the message. The decryption of the ciphertext is performed as follows:

$$m = c^d \mod n$$

In the equation, d and n is the private key of the receiver. This method is used for creating signatures on messages to prove the integrity and authenticity of the message. Using a PKI with certificates and signatures on messages can provide traceability on messages with authentication and integrity.

Public key infrastructure ensures the authentication in a system where every user trusts the CA. The CA signs the certificate of each user and ensures that their public key, used to decrypt ciphertext encrypted with their private key, can be used to authenticate identity in the infrastructure.

2.4 Existing approaches to offline transactions

The problem of offline transactions gets some recognition from central banks, as we have seen and shall further review in Section 3.1. However, most central banks in the early stages of CBDC research seem to avoid tackling the offline problem head-on. Offline functionality is stated to be a desired and maybe even necessary feature. However, the technical solutions required to achieve that are generally viewed as an area where more research is needed. In this section, we will review the existing research and approaches to offline transactions and cover some of the variations of solutions that have been proposed.

2.4.1 Offline Payment System

One of the more advanced approaches to an offline protocol to date is proposed in [CGK+20]. This approach utilizes a combination of a central certificate authority, like a central bank, and secure hardware to overcome the challenges posed by the offline problem. The article proposes a complete protocol, referred to as the Offline Payment System (OPS), for user registration, offline transactions, and any required post-offline settlement. The protocol requires all users to have access to a Trusted execution environment (TEE). The TEE is essentially an implementation of secure hardware that allows for the execution of a Trusted application (TA). Specifically it uses the GlobalPlatform standardization for TEEs, which will be further reviewed in Section 2.5. The TEE provides guarantees of confidentiality and integrity, which ensures that no unauthorized actor can access the data of the TA or modify the code of the TA. This is the protocol's main protection against double spending attacks. The TEE uses a counter to prevent replay/double spending attacks. [CGK+20]

The OPS protocol requires users to complete a registration procedure to establish the necessary cryptographic keys and certificates. This procedure registers the users' TEE with the central server and synchronizes a counter to keep track of transactions between the client and the server. Following the registration, users can deposit online funds for offline use. This involves the client sending a deposit request to the server, which checks that the client has sufficient online funds and charges the amount from the client's online account. The server then sends a signed confirmation of the deposit to the client. This allows the client to fill up her offline funds in the TA. The server and the TEE maintains counters to prevent replay attacks of this message. A withdrawal protocol doing the opposite is also included in the OPS. [CGK+20]

The OPS payment protocol describes how a transaction is performed. Every transaction is initiated by the payee, who sends a payment request to the payer. If the payer approves the transaction, the payer's TA checks that sufficient offline funds are available and, if so, deducts them from the payer's offline balance. A payment confirmation signed by the payer is then sent to the payee as proof of the transaction. The payee's TA validates the confirmation. All users also maintain a payment log to detect and prevent double spending attacks, which the payment is checked against. When the payee reestablishes the connection to the server, a claim protocol is used to transfer the funds received while offline back into online funds. This involves the payee sending the payment confirmation to the server, which checks its validity and converts the appropriate funds. [CGK+20]

The OPS protocol(s) provide a complete system for offline transactions. It does, however, have certain requirements that may be undesirable for a CBDC. The protocol relies on secure hardware to prevent malicious actors from abusing the protocol through double spending. The security of the protocol is therefore

limited by the security of the secure hardware and execution environment. The logs implemented in the protocol do not protect against double spending to multiple payees. The protocol also requires users to have access to secure hardware, a TEE enabled device, to partake in the protocol. Furthermore, the protocol also proactively requires users to deposit funds for offline use. These requirements degrade the security and availability of the protocol.

A similar approach can be found in [LWZ+21]. Here a similar protocol, relying on a TEE is outlined.

2.4.2 Pure Wallet

Another proposed solution to the offline problem is presented in [IDLK21], referred to as Pure Wallet architecture. This approach focuses on Bitcoin use in relatively short lasting offline scenarios. To ensure offline functionality, offline tokens are introduced as a claim on online funds to be exchanged in an offline setting. Tokens are issued by a token manager while users are online in exchange for online funds, in this case Bitcoin. A payer can obtain a token by initializing a transaction on the blockchain for a set amount. The token manager then transfers a token matching this amount to the payer. The payer can then send this token to the payee during an offline session. When connectivity is restored, the payee can present the token manager with the token to be rewarded with the funds involved in the transaction. If the token is not used, funds are returned to the payer's Bitcoin wallet. [IDLK21]

Each token issued by the token manager has a predetermined time to live. It can only be redeemed within this period and should therefore only be accepted by a payee within this period. This also puts a constraint on how quickly the payee has to redeem the value of the received token. The protocol also limits a token to single-use, meaning it can only be used once in one transaction. Upon such an exchange, the token is to be deleted from the payer's device. [IDLK21]

The Pure Wallet architecture has several limitations. Its protection against double spending is based on tokens being removed from the payer's device upon use, presumably relying on secure hardware and a token timeout to prevent large scale fraud. The architecture's intended use case is temporary offline scenarios with users in close proximity to each other, like on an airplane. In such a scenario, it is possible that the relatively weak protections against double spending are sufficient. However, the architecture as described does not scale with a threat model posed on a CBDC. It is also subject to the requirement that users need to proactively allocate funds for use in an offline setting.

2.4.3 Chaum's protocol

One approach to the offline problem that does not rely on secure hardware of any kind is presented in [OS14]. This proposal builds on an older protocol for offline transactions first presented in [CFN90], hereafter referred to as Chaum's protocol, and enhances it to mediate the double spending threat. The protocol described is looking at three parties to partake in any transaction, a payer, a payee, and a trusted third party, e.g., a bank. The banknotes to be used in offline transactions are constructed by the payer while online against a deposit of online funds with the bank and can be represented as a sequence of triplets:

$$T_i = (h(a_i, c_i), h(a_i \oplus (u||C(K_i)), d_i), K_i) \quad , \quad i \in 1, 2, ...n \tag{2.1}$$

Here d_i and c_i are randomly selected numbers the payer uses as a password. u is the ID of the payer, also known to the bank, and a_i is a random number the payer uses to hide u. K_i is a one-time RSA public key, and C is the payer's certificate used to sign the keys, K_i. This key must be preregistered with the bank before any bank notes can be made. The size of n decides the security of the system.

The creation of such a banknote starts with the payer selecting $2 * n$ triplets, T_i, and creates matching obfuscation coefficients r_i. The client calculates obfuscated hashes, $r_i^e * h(T_i)$, where e is the bank's public key and sends it to the bank. The bank then selects n of these hashed triplets and requests the corresponding r_i, a_i, c_i and d_i from the payer. The bank then confirms that these parameters are correct in regards to the obfuscated hashed triplets. It also checks that all triplets contain the payer's certificate signed to the key, K_i, belonging to the payer. If so, the bank signs all the remaining triplets with its private key, d, multiplies them together, and returns them to the payer so that the payer receives:

$$I = \prod (r_i^e * h(T_i))^d \mod n \quad , \quad i \in L \tag{2.2}$$

Here L is the set of triplets signed by the bank selected from the $2 * n$ triplets first sent from the payer. I is the obfuscated and signed triplets. The payer can now deobfuscate these triplets by reapplying the obfuscation coefficient:

$$Z = I * \prod r_i^{-1} \mod n = \prod h(T_i)^d \mod n \quad , \quad i \in L \tag{2.3}$$

The client now holds a banknote comprised of n triplets, T_i, and their signature by the bank, Z.

To perform a transaction, the payer sends the signed banknote, Z, and the triplets, T_i, to the payee. The payee confirms that the signature is correct using the public key of the bank. This verifies that the banknote is legitimately issued by the bank. The payee then sends a challenge, Y, to the payer. The challenge is a sequence of bits of length n. Some of the bits correspond to an ID for the payee, and some are selected at random. The payer signs the challenge with its one time keys:

$$R = K_1(K_2(...K_n(Y)...)) \tag{2.4}$$

and returns this signature for validation by the payee. Next the payer return, for each bit in the challenge, either a_i and c_i if the ith bit is 0, or $a_i \oplus (u||C(K_i))$ and d_i if the bit is 1. The payee confirms that the received values correspond to the earlier received values for the triplets, T_i.

To redeem a banknote, when the payee has reestablished a connection to the bank, the payee sends Z, T_i, Y, R and the payer's response to the challenge to the bank. The bank is then able to validate the transaction in the same way the payee was when performing the transaction. If the bank cannot validate the transaction, that is, the response to the challenge, (a_i, c_i) and $(a_i \oplus (u||C(K_i)), d_i)$, does not correspond when hashed to the information in the triplets T_i, the bank knows that the fault is on the payee as she has accepted an invalid banknote.

If the banknote is valid, the bank checks its logs to see if the banknote has been spent before. If it has not, the transaction is approved by all parties and the bank transfer the appropriate online funds deposited by the payer while online to the payee. If the banknote has previously been spent, there are primarily two possible scenarios. The first is that an identical transaction with the same triplets and challenge has already been posted to the bank. Because each transaction should contain a unique, randomly generated challenge, this would suggest that the payee is being fraudulent by attempting to reclaim the same transaction multiple times. The other case is a transaction using the same banknote, but a different challenge is posted to the bank. In this case, it is likely that the payer has double spent the banknote as only the payer is capable of creating such a transaction.

A key feature of this transaction scheme is that it maintains anonymity for all users who have not double spent their banknotes. If used correctly, the bank and the payee will not be able to determine the identity of the payer as the payer ID, u, is hidden using the random variable a_i. When a banknote is double spent, however, it is highly likely that for at least one position in the different challenges, the bit is different. Knowing about both of these transactions would then provide you both (a_i, c_i) and $(a_i \oplus (u||C(K_i)), d_i)$ for the same i. It is then possible to reveal u and

identify the fraudulent payer. A similar argument holds for the certificate $C(K_i)$, thus proving that the payer committed fraud. [OS14]

Chaum's protocol provides a solution to offline transactions that does not rely on secure hardware to detect double spending attacks. In addition to this, it also allows for anonymity for all non-fraudulent users. It does, however, not provide any prevention of double spending in the act. That is, the payee has no ability to tell if the payer is double spending until it can reestablish the connection to the bank, given that the banknote was first spent in a transaction with a different payee. This makes the finality of transactions impossible without compromising the security of the protocol. The protocol also has no support for offline respendability of the same banknote, as the payee has no means of reconstructing the banknote.

Chaum has further made research into how to issue a CBDC in his paper *"How to Issue a Central Bank Digital Currency"* [CGM21]. The paper suggests a token based approach with an expiration date is implemented, and a Chaum-style blind-signature protocol is presented. The paper guarantees perfect, quantum resilient transaction privacy while being within the Anti money laundering (AML) and Countering the financing of terrorism (CFT) regulations. To ensure a coin has not been double spent, the payee in the transaction must deposit the coin to the central bank, and the bank checks that the coin has not been spent before. This approach ensures privacy but does not protect against double spending from the payee's perspective during a long offline period.

2.5 Secure Hardware

As we have seen, several approaches to the offline problem rely on some form of secure hardware. That is, an environment to execute a trusted application in a way that prevents tampering with the application itself or the related data. Such environments may be provided by specialized, dedicated hardware but can also be found in more widespread devices such as some smartphones. In this section, some existing systems that provide this functionality are reviewed. Other similar approaches to secure hardware, such as the safe storage provided by BankId will be presented.

2.5.1 Trusted Execution Environment

As stated, TEEs allows execution of TAs and provides a high level of trust in the application being executed correctly and that the belonging data has not been modified or manipulated by other processes. TEEs require authorization of any application before it can run in the environment. Any data belonging to a TA is also isolated so that only that TA may read or modify it. [Fel19] In their 2015 white paper [Glo15] GlobalPlatform outlines their TEE standardization. According to GlobalPlatform themselves,

the TEE is "*an isolated execution environment that runs alongside the Rich OS and hosts trusted services offered to that rich environment.*" [Glo15] GlobalPlatform is one of several such standards that have been adopted by ARM TrustZone which means it is available in many Android phones among other devices [CGK+20]. The environment includes separate hardware resources and only allows specific programs, TAs, to run on the hardware. The TEE allows for the execution of TAs in a way that provides confidentiality, authenticity, and integrity of the execution. A TEE has a fixed Application Programming Interface (API) for interacting with the TA. The interface can be reached from untrusted applications to execute the public functions of the TA. [Glo15]

A variety of other standardization of TEEs exists. Many hardware manufacturers have also implemented TEE support, such as Intel's SGX (Software Guard Extensions), the mentioned ARM's TrustZone, AMD's Secure Execution Environment, and Apple's Secure Enclave. [GANAHHJ18]

2.5.2 BankId

BankId is a collaboration of banks and payment service providers, and telecom providers in Norway [Col22]. BankId on the phone was developed by Telenor and the banks in the BankId collaborations and was launched in 2009. The application is specific for Norway and stores the private key of the user on the sim card. This application requires specific hardware, specifically specialized sim cards, and is an example of specialized infrastructure distributed in a country.

2.6 Simulation and networks

In this section, some of the existing literature relating to networks is reviewed. Foundational theory regarding modeling networks is presented, and their characteristics and limitations are explained. A concept of networks and their properties is necessary to comprehend the fundamental challenges posed by being offline, as defined in this thesis. Furthermore, an understanding of network modeling is essential for creating and applying the simulation framework that will be presented in Chapter 6.

2.6.1 Graph theory

Graphs are commonly used as representations of real world situations. *"A graph G is an ordered triple* $(V(G), E(G), \psi_G)$ *consisting of a nonempty set of vertices V(G), a set E(G), disjoint from V(G) of edges, and an incident function* ψ_G *that associates with each edge of G and unordered pair of (not necessarily distinct) vertices of G."* [BM+76]

Links in a network can be directed or undirected. A network is called undirected if all the links in a network are undirected. The vertices are often called nodes, while the edges are often called links. In the context of the internet, the nodes are a representation of routers or end users, while the links are internet connections. The graph is typically undirected, as the communication is two-way. [Bar13]

Network science uses the concepts from graph theory to deal with randomness and use the organizing principles from statistical physics to simulate networks. Thereby, network science can extract information from incomplete and noisy datasets. [Bar13]

A path is a route traversing links between the nodes in the network. The path's length is the number of links the path contains. The shortest path between two nodes, i and j, is the path with the shortest length. The shortest path is often called the distance, denoted d_{ij}. In an undirected graph, the shortest path between i and j is the same as between j and i. Finding the shortest path in an undirected unweighted graph is often done using the Breadth First Search (BFS) algorithm. In a complete graph, each node is connected to every other node. In a computer network, this is rarely the case. [Bar13]

A Directed Acyclic Graph (DAG) is a type of directed graph [TS11]. A depth first search can be used for a topological sort of a DAG. The sort assumes there is no difference between the nodes in the graph and sorts the graph in linear time. [CLRS22]

2.6.2 Disconnected Graph problem

A disconnected graph is a graph where there are subgraphs that do not have a path between them. A disconnected graph is, per definition, also an incomplete graph. In a disconnected graph, there is no way of guaranteeing complete information, as the two network may diverge without synchronization of information.

In a shared memory space, the data should have consistency and availability as long as network partitions do not exist. Out of the properties consistency, availability, and tolerance to network partitions, there can only be at the most two of these properties for any shared-data system [Bre00]. Another way of saying this is: *"the impossibility of guaranteeing both safety and liveness in an unreliable distributed system"* [GL12].

One strategy is forfeit consistency, ensuring availability and tolerance to network partitions. This is an optimistic approach, resulting in conflicts that need resolution in the future. One way of ensuring some sort of consistency or correctness of data is data expiration, assuming it is possible to reconnect to the rest of the network. [Bre00]

Forfeiting availability also allows tolerance to network partitions. This can be done with pessimistic locking of data and making minority partitions unavailable. [Bre00]

Forfeiting both availability and consistency leads to a tradeoff resulting from the CAP theorem. As a disconnected graph suggests network partitions, the only way to ensure availability is to compromise on consistency. By explicitly handling partitions, the designer of the system can optimize the consistency and availability and achieve some tradeoffs in all three [Bre12]. This theorem can be applied to state that one cannot achieve the finality of transactions and consistency of value in the system in an offline setting. This point will be further discussed throughout this thesis.

2.6.3 Random Networks

Networks are represented through a graph to visualize elements and their interactions. Many systems can be described as a network with different topologies depending on their complexity, size, interactions, and density. The World Wide Web (WWW) is such a network connecting clients and servers through the internet. Large networks self-organize into a scale-free state, meaning a vertex in a network interacts with a decaying number of other vertices following a power-law distribution [BA99]. Following this, larger network sizes do not necessarily imply that each vertex has a higher node degree. Real networks are, to some extent, random.

Traditionally, the random graph theory of Erdös Rényl (ER) was used for complex network topologies[ER$^+$60]. Watts and Strogatz (WS) is used to simulate a small world model [WS98]. In both ER and WS, the number of vertices is fixed and then randomly connected. In most real-world networks, the network is open and grows by adding new vertices in the system, increasing through the lifetime of the network [BA99]. Both the ER and the WS models assume the probability of two vertices being connected is uniform and randomly distributed.

In a real network, some vertices have more connections than others, and it is more probable of connecting to a vertex that already has several connections. These two concerns are addressed in the Albert-László Barabási preferential attachment model (BA) [Bar13] where the amount of vertices is variable, and the preferential attachment ensures vertices with more edges are preferred for new connections [JNB03]. Barabási concludes that real networks are not random and may be the wrong model for most real systems, even though they remain relevant for network science [Bar13].

2.7 Key takeaways

We have, in this chapter, covered a variety of relevant topics relating to CBDCs, digital transactions, and offline functionality. Double spending makes it challenging

to facilitate offline transactions securely. Other issues like fair exchange further complicate the problem. Existing CBDC projects have different approaches to offline functionality both in technical terms, such as the system architecture, but also in terms of expectations, requirements, and the prioritizing of requirements. We have reviewed some existing approaches to offline functionality, outlining the technical possibilities and limitations.

Chapter 3
Requirements

The goal of this chapter is to derive the requirements and limitations for an offline protocol and a system architecture for a retail CBDC in a Norwegian context. Norges Bank (NB) has created several reports deciding various non-functional requirements and system goals. However, their requirement list is not complete. Therefore, this chapter consists of the phase of gathering specific requirements. This is divided into three parts, gathering relevant requirements from existing CBDC projects, gathering requirements from NB in an interview with the bank, and then using the data to derive appropriate final requirements. The various central bank projects define requirements on different levels depending on the purpose of their reports. Business level requirements will be used to derive the functional and non-functional requirements. The purpose of gathering requirements is the use for designing, testing, and verification of the best effort solution.

3.1 CBDC requirements from literature

3.1.1 Requirements from Norges Bank

Norges Bank's research into a Norwegian CBDC is summarized over three reports published in 2018 [Ban18], 2019 [Ban19] and 2021 [Ban21a]. In these reports, a set of requirements and desirable characteristics for a CBDC are listed. The characteristics change somewhat between the different reports.

In the first report, [Ban18], the characteristics, displayed in Table 3.1, are mainly high level requirements providing a description of desired functionality and effects. An important aspect described is the desire to use the currency in a contingency situation without describing the specific use case. The report mentions several non-functional requirements as design principles for software development. The possible non-functional requirements are scalability, interoperability, availability, security, and flexibility.

Characteristic	Description
Efficiency	The banking system should be efficient, fast, secure at low costs matching the population's needs.
Payment usage	Means of payment and payment instruments must cover a range of different needs, as payments are made in many different situations and users may have differing priorities.
Redundancy	Backup solutions must be available that can operate effectively in the event of a service stoppage.

Table 3.1: Relevant requirements from Norges Bank first report [Ban18]

In the second report, [Ban19], the characteristics are separated into essential characteristics, displayed in Table 3.2 and desirable characteristics, displayed in Table 3.3. In this report, the descriptions and explorations from the first report are formalized into requirements. The separation between essential and desirable is dependent on the choice of architecture and methods of the development of the CBDC. The essential characteristics are focused on high-level design goals. Sufficient speed and technical autonomy are necessary for a well functioning product with good user experience and compliance with sound Information Technology (IT) architecture principles. These aspects are important for a reliable and maintainable product. The desirable characteristics include security, a non-functional requirement important for the trust in the Norwegian payment system. DLT compatibility is a feature to be able to have a stable system, as this is the foundation for a redundancy solution. Offline payment is a desirable feature as this provides a use case in contingency situations.

In the third report, [Ban21a], system requirements defined in the second report are repeated, but the technical solutions are discussed. In the report, the desirable characteristics from the second report are combined with the essential characteristics in a combined characteristics table. Every characteristic is necessary to consider in the final solution of a CBDC. In the report, it is specified that a solution must be modular to be used in a payment situation. All payments should be immediate and final. Offline payments are defined as *"direct payment between end users and their payment instruments in situations where there is no contact between the register or account system and the user interface."*. Offline capabilities should be possible when users are in close physical proximity to each other. [Ban21a]

3.1.2 Requirements from Sweden's Riksbank e-krona

Sweden's central bank, the Riksbank has created several reports on the development of e-krona, the Swedish CBDC, with different requirements on different levels. The

Essential requirements

Characteristic	Description
Administrative	Controlled by Norges Bank
Speed	CBDC payments are immediate and final
Capability	Capable of functioning as legal tender
Technical Autonomy	Satisfy requirements relating to technical autonomy
IT Architecture	Compliant with sound IT architecture principles

Table 3.2: Relevant essential requirements from Norges Bank second report [Ban19]

Desirable Characteristics

Characteristic	Description
Security	Provision of the desired degree of data protection
Offline payment	Depending on the technical autonomy required, this may be a necessary characteristic, although the option to make offline payments is considered desirable in any event.
DLT compatibility	Distributed Ledger compatibility
Third-party compatible	The objective is that third-party stakeholders should be able to innovate and build services on top of the CBDC.

Table 3.3: Relevant desirable requirements from Norges Bank second report [Ban19]

goal of the project is to create a digital complement to cash to support the Riksbank in the task of promoting a safe and efficient payment system, as stated in the report [Rik17]. The basic relevant characteristics from the report are listed below.

- It is electronically available 24 hours a day, 7 days a week, 365 days a year and available in real time or close to real time.
- It is available to the general public, i.e. is more broadly available than traditional central bank deposits in RIX, to which only the banks have access.

The main goal is thereby to create an available and public currency for use in day-to-day transactions. The Riksbank's requirements differ from the Norwegian requirements, as the practice of cash as a legal tender is not compulsory in Sweden. This motivates the development of e-krona, while the report concludes that the decision of e-krona as a legal tender should be made by the legislator.

Technical considerations of the Riksbank include considerations and technical properties regardless of the design choices. This includes scalability, interoperability, reliability, and accessibility. These requirements are non-functional requirements and similar to the requirements of Norges Bank. The rapid development at the time of writing prevented the bank from concluding on a choice of technology other than the founding principles.

Sweden's central bank's CBDC project is one of few with a finished phase 1 prototype. Their phase 1 pilot was a token based solution with a DLT. The prototype did not include offline functionality, but it is an emphasized research area of interest, and the architecture was designed to support the feature [Rik21]. The Riksbank published a report specifically on the possibility of a cash-like CBDC [ACH21]. The report states the requirement for offline payment to be non-anonymous by the EU regulations in the case where transactions are stored remotely, regardless of whether the system is token based or account based. With locally stored tokens, transactions up to 150 EUR can be made without the need for individuals to identify themselves according to the AML regulations.

3.1.3 Requirements from other central banks

The Bank of Canada, the European Central Bank, the Bank of Japan, Sveriges Riksbank, the Swiss National Bank, the Bank of England, the Board of Governors of the Federal Reserve, and the Bank for International Settlements have collaborated in creating a report to decide on the foundational principles. Core features of a CBDC [oCBoJ+20]. The report points out that the feasibility of achieving each characteristic depends on the technologies available and the overall design goals of the system.

In their report, they divided the core features into three topics: instrument features, system features, and institutional features. The relevant features for the purpose of technical requirements are instrument features and system features. The relevant features from their report can be seen in Table 3.4. The table includes most of the requirements mentioned by other central banks, with several general features divided into more specific features. The report provides a general framework for the development of a CBDC on the principles of complementing traditional money alternatives, providing stability, motivating innovation, and efficiency.

The central bank of Iceland has researched the potential for Rafkrona and developed similar general technical requirements as a foundation as the previously discussed central banks [oI18]. The bank emphasizes the need for an available currency for everyone.

In China, the PBoC, who are already running trials of their CBDC, have noted

Instrument features	
Convenient	CBDC payments should be as easy as using cash, tapping with a card or scanning a mobile phone to encourage adoption and accessibility.
Accepted and available	A CBDC should be usable in many of the same types of transactions as cash, including point of sale and person-to-person. This will include some ability to make offline transactions (possibly for limited periods and up to predetermined thresholds).
Low cost	CBDC payments should be at very low or no cost to end users, who should also face minimal requirements for technological investment.
System features	
Secure	Both the infrastructure and participants of a CBDC system should be extremely resistant to cyber attacks and other threats. This should also include ensuring effective protection from counterfeiting.
Instant	Instant or near-instant final settlement should be available to end users of the system.
Resilient	A CBDC system should be extremely resilient to operational failure and disruptions, natural disasters, electrical outages and other issues. There should be some ability for end users to make offline payments if network connections are unavailable.
Available	End users of the system should be able to make payments 24/7/365.
Throughput	The system should be able to process a very high number of transactions.
Scalable	To accommodate the potential for large future volumes, a CBDC system should be able to expand.
Interoperable	The system needs to offer sufficient interaction mechanisms with private sector digital payment systems and arrangements to allow easy flow of funds between systems.
Flexible and adaptable	A CBDC system should be flexible and adaptable to changing conditions and policy imperatives.

Table 3.4: Selected core CBDC features of [oCBoJ$^+$20]

certain objectives and visions for the e-CNY. One of these is *"... to support fair competition, efficiency and safety of retail payment services"*. [oC21] Furthermore, the PBoC expands on this objective, stating that *"In addition, it supports off-line transactions and is settled upon payment"*. [oC21] For the Chinese CBDC then, offline functionality is a clear requirement. As we have seen, such offline functionality is already present in the existing versions of the currency. The reasoning of the PBoC for prioritizing offline functionality to this degree seems to be based on its ability to ensure availability and usability for all users in a variety of settings, which is seen as key to the currency's proliferation. [GAN21]

The PBoC also states multiple other objectives and requirements concerning the e-CNY. The most relevant of these are their requirements with regard to privacy and anonymity and AML regulation. Regarding anonymity, the PBoC utilizes a managed anonymity scheme. This involves preserving user anonymity for typical retail use, i.e., small value transactions. High-value transactions, on the other hand, should be traceable to combat illegal activities and misuse. This allows the currency to comply with China's AML and CFT requirements. [oC21]

3.2 Interview with Norges Bank

3.2.1 Preparation requirements discussion

Before the interview the following technical requirements were proposed as discussion points.

- Overall system architecture (DLT, hierarchy)
- Credit risk associated to offline transactions
- Requirements to offline transactions (type of transaction and use cases)
- Privacy and anonymity

Norges Banks second report discuss the requirements, purpose, and possibility of a national CBDC [Ban19] as elaborated in Section 3.1.1. Technical autonomy is an essential requirement from the second report, and from this, the question is raised of how an overall system architecture could be designed. DLT compatibility is a desirable characteristic, and a DLT could enable a distributed architecture.

The fair exchange problem[1] and the disconnected graph problem[2] leads to a potential for financial risk if offline transactions. Norges Bank's willingness to take

[1] See Section 2.3.2
[2] See Section 2.6.2

responsibility and the consequences of the existence of risk in the system are important considerations for the system design. There is always a risk with offline transactions, and it is unclear to what extent it is tolerable. The use cases of the offline protocol and offline environment are important for a potential simulation. The possibility of enforcing limitations on transaction limits and transaction amounts could be important for limiting the financial risk. In Section 3.1.3 it is suggested that the e-CNY has such offline limitations depending on the user and user group. A spending limit would limit the amount of profit a potential adversary could gain from attacking the system. A use case of day-to-day transactions for offline use could implicate the same parties would be able to do offline transactions as they usually do in the online setting. With the limitation of not allowing new trading parties in the offline setting, the risk of encountering a fraudulent user could be minimized. The privacy and anonymity of the user are discussed in Norges Banks third report [Ban21a] along with different technical solutions. As Norges Bank's potential CBDC is proposed to be a cash-like alternative, anonymity is a topic of interest. In an offline setting, it is interesting to know the importance of the anonymity feature. A CBDC could provide more anonymity than traditional banking, and it could also help reduce money laundering and other illegal activities if it is cash-equivalent.

3.2.2 The interview with Norges Bank

The interview with Norges Bank was conducted on 01.02.2022 remotely. The transcript of the interview can be found in Section A.1 in Norwegian, as this was the spoken language during the interview. The participants from Norges Bank were Peder Østbye, special advisor for the department of Financial Stability, and Lasse Meholm, project coordinator of the experimental testing of CBDC. Through the interview, several interesting talking points were discussed, and in the following sections, the main points will be summarized.

Norges Bank is open to several architectural designs regarding the CBDC. If this is centralized, semi-decentralized, or fully decentralized has not been decided, and further investigation is needed. This will depend on the costs and benefits and will be discovered in their current experimental testing phase.

An offline solution should be designed to be indifferent to the underlying technology. With a modular design, each module could be replaced if the underlying technology is outdated without redesigning the entire system.

A blockchain could be a part of the final design if appropriate, but this is not decided. A traditional database could serve the same purpose, depending on the architecture.

In their testing phase, NB will use an ERC-20 token standard or an ERC-20 token standard equivalent as a standardized unit of operations.

The use of existing cryptocurrency projects as a basis can be helpful. These projects have large ecosystems and a community of developers.

Third parties are welcome to participate in the project, as Norges Bank does not envision itself taking a larger part of the economic ecosystem than at present. Several roles could be assigned or opened for third parties if the contribution of the third parties contributes to the system goals.

Norges Bank wishes to supply the core functionality of the system, the underlying infrastructure, and the register of transactions. There is functionality NB needs to be responsible for — mainly creation and destruction of money and control of the money supply — while other actions such as validation of transactions or customer wallet solutions could be outsourced to third parties.

In offline payments, the receiver should not be the one taking the counterpart risk. The risk could be covered by, for example, an insurance policy.

Any potential risk in offline payments should be mitigated as much as possible. The remaining risk could be acceptable. The current money system with physical cash has a counterfeiting problem. If the new risk is lower than the counterfeiting risk, the bank will be content.

Limited amount of transactions, limited amount per transaction, or time-based spending limits are types of spending limits that could be considered for mitigating risk. Regulatory and legal factors could limit the transactions based on the anonymity offered. The limits should be adjustable to be a valuable tool in different contingency cases.

The monetary value represented in the CBDC would represent today's cash. This implicates a user could not have credit in the form of CBDC, but third parties could give credit to the user and this could be exchanged for units of the CBDC. Third parties could give credits based on their evaluation of the credit capabilities of the user.

An offline solution should be able to cover the whole spectrum of offline use cases. This includes a few days of disconnected functionality and up to a few months of no communication with the central system. In contingency situations, the use cases should at least cover essential products, such as food and medicines. Whether specialized hardware is required or not is yet to be determined, as well as whether this should be required for businesses.

Ideally, an offline part of the system should work without any external dependency, including telecommunications and electricity. The requirement for electricity is tolerable, but it should at least be independent of telecommunication, i.e. the internet, for a period of time.

3.3 CBDC requirements in Norwegian context

In this section, the requirements for the design of an offline protocol will be presented. These are based on the various requirements gathered from different actors in the CBDC research space, with an emphasis on those collected from NB. The requirements are categorized as business-, functional-, and non-functional requirements. This set of requirements will provide a scope and limitations for the following design process.

Other kinds of requirements, such as user requirements, will not be specified. The design process is a best-effort attempt at making a high-level prototype solution to the offline problem. Considering this, constructing requirements that outline the end user experience and use cases should be left to further research.

Some of the following requirements have contradicting implications for the design choices for the protocol. Some requirements, as we will see, also have technical contradictions and infeasibilities within themselves. Any such contradictions, and the following trade-offs, will be further discussed in Chapter 5.

3.3.1 Stakeholders

The stakeholders in a Norwegian CBDC solution are mainly the payment solution providers, Norges Bank, and the Norwegian consumers in combination with regulatory and policy responsible authorities. In the Norwegian money market, Norges bank is already a dominating party in collaboration with commercial banks [Ban21b]. In addition, finance companies and insurance companies have a potential stake in the CDBC solution, depending on the intended use and implementation. As a CBDC is considered to be a modern continuation of cash, the main parties would be NB, the end users, and businesses.

Cash is a legal tender in Norway, and a consumer has the right to settle an obligation in cash [Ban21b]. With the possibility of a CBDC, every business that today is required to accept physical cash could have an interest in a legally supported digital equivalent.

Payment systems are often divided into interbank payment systems and systems for payment services. A retail CBDC would be a system for payment services, and depending on the implementation; it could need a clearing system [Ban21b]. The core of the interbank payment system in Norway is the Norwegian Interbank Clearing

System (NICS), responsible for the balances of every bank based on the customers' transactions. A retail CBDC could have this interbank clearing included as the same system would be used independently of the bank, with each deposit of the consumer being directly in Norges Bank.

The financial payment system for retail payments in Norway is dominated by BankAxept, as seven out of ten payments in Norway are made with a BankAxept card. Visa is a large stakeholder in the global payment market, but BankAxept is chosen automatically by most payment terminals in Norway, reducing Visa's share of transactions. [Ban21b]

The prospect of a CBDC in a Norwegian context is, in this thesis, from the perspective of Norges Bank. This extends to the relationship between Norges Bank and their proposed system and the end user.

3.3.2 Business requirements

In NB's second report [Ban19], the main business goal of a potential CBDC is formulated as the main question.

- Ensure that Norway maintains a secure and efficient payment system.
- Provide confidence in the monetary system.
- Function as a contingency solution in case of failures in bank payment systems.

3.3.3 Functional requirements

The functional requirements are constructed based on various requirements gathered from NB and other central banks, as well as existing literature on offline transaction protocols and systems presented in Section 2.4. These requirements are subsequently formulated based on what is technically possible and functionally desirable for a Norwegian CBDC's offline capabilities.

FR1 Transactions can be made even if one or both of the transacting parties are not connected to a central server or the system as a whole.

FR2 It shall not be possible to create and use tokens of value not already in existence through forging, double spending, or any other means.

FR3 No user should be able to spend tokens of value that are not in their possession through a valid exchange.

FR4 Money transferred in an offline setting shall be reusable without connecting to a central server.

FR5 Transactions completed in an offline context should be final so that they cannot be disputed by the transacting parties after the fact.

FR6 The system shall be available for use at any moment, given that both parties have the required equipment and preloaded funds.

FR7 Any illegal transaction should be traceable and identifiable so that the origin of illegitimate funds can be determined.

FR8 It should not be possible to change the total supply of money or value in the system by creating or destroying funds or by any other means.

3.3.4 Non-functional requirements

Through the discussion with NB, summarized in Section 3.2.2, the non-functional requirements were indirectly specified and mentioned. Through their report [Ban19], several non-functional requirements were mentioned and discussed. The resulting list is a combination of the previous research.

Availability The payment solution should be available for use in any situation at any time.

Confidentiality The individual use of the system, and the resulting data, should not be accessible or traceable by unauthorized parties.

Integrity The integrity of the transactions should be preserved. This means that the data entered into the system should be accurate and unchanged.

Interoperability The system should be able to interact with other systems in order to exchange data and provide an interface for third parties.

Traceability The bank should be able to view transaction logs and keep track of the changes that have been made. This would allow for better auditing, accountability, and fulfilling AML regulations.

Fault Tolerance A fault in the system should not affect the system's ability to perform uninterrupted.

Graceful Degradation Any abuse of the system should not propagate in a matter that amplifies the severity of the abuse. Any abuse should be detected at the earliest possible opportunity, imposing limitations on further abuse.

Modularity Each part of the system should support a modular design to provide easier maintainable software modules. This would reduce the cost of replacing a module if requirements or the threat model changes.

Reliable The system should be able to handle large amounts of transactions and be able to perform consistently. An offline context should not limit the reliability of the system.

Privacy The system should be able to protect the privacy of the users.

Fairness The protocol should strive to facilitate fair exchange characterized by effectiveness, fairness, and timelessness as specified in Section 2.3.2.

Lasting security The protocol should be able to maintain a secure state of operation, including supply conservation, in a lasting offline state.

3.3.5 Requirements validation

The requirements stated above were reviewed by the CBDC research group of Norges Bank. Their conclusion was that the requirements outline the necessary and satisfactory functionality and behavior of CBDC in a Norwegian context.

3.4 Key takeaways

In this chapter, existing CBDC projects have been reviewed to clarify common general requirements for a CBDC. The results of an interview with Norges Bank are presented along with requirements outlined in the bank's reports. The relevant stakeholders for a Norwegian CBDC are also presented. A set of business-, functional-, and non-functional requirements for a Norwegian CBDC are formulated and validated by Norges Bank.

Chapter 4
Architecture design

In this chapter, the architecture necessary for the model and simulation will be derived. The simulation context is an approximate model of offline scenarios possible in Norway defined in Section 4.1. The connectivity to the internet will be explored in Section 4.1.1. With this research, we can define the connectivity of the end users and thus estimate the number of connections to a backbone network each user has. We will then explore which network topologies are the best suited for the simulation. This chapter will explain the foundations for the research on an optimal degree of decentralization in the system. The level of centralization in a CBDC system is discussed to give a suggestion for the general system architecture to be used in a simulation. The modeling of a processing node is discussed in the last part of the chapter to outline how an architecture with stateful nodes can be simulated and derive the availability of such a system.

4.1 Simulation context

The definition of offline can vary with the purpose and goal of the study. In this thesis, offline will be defined as the following: Offline is disconnected from all parts running the server side program of the payment solution. Users are assumed to have appropriate devices and electricity. The timeframe is assumed to be anywhere between a few minutes to a few months. The functionality the retail CBDC should provide is a point of sales offline solution with physical proximity between payer and payee.

4.1.1 Network access in Norway

The Norwegian Communication Authority states in their 2020 article on network coverage in Norway that the internet access has, in general, good coverage [AK20]. 98% of all households have access to at least 30mbp/s download capacity, and 89% of all households have access to 100Mbit/s download speed. There are approximately only a few hundred households that lack access to a minimum of least 10Mbit/s

download capacity. On a national level, 98% of the population has access to cabled broadband, almost 100% have access to radio-based broadband or equivalents, and 97% have access to satellite broadband. When disregarding satellite connectivity, 99.98% of all households have access to 10 Mbit/s broadband [AK20]. With this, we can conclude that there is well-established connectivity during normal circumstances with broad access to alternative communication mediums such as satellite connectivity.

The backbone network for universities, colleges, and research institutions in Norway is Uninett [GHHK10]. This network delivers network connectivity through universities in Norway and stretches through the country. Within this backbone, there exists a high correlation coefficient, and high autocorrelation in some failure processes [GHHK10].

On the sea, the VHF Data Exchange System (VDES) is to be an enabling technology for the extended ship-to-ship communication and general data transmission [LRW+19]. This enables a network of ships to communicate with land stations and other ships through VDE and a mix of VHF and satellite connections.

Tampnet and Maritime Communications Partner (MCP) are important for the infrastructure at sea. Tampnet delivers high-speed connectivity to offshore installations through fiber. Tampnet delivers the fastest way out of Norway, to Europe, without passing through Sweden [LBJH+15]. MCP is a mobile carrier for the ocean and is operating in maritime areas in the whole world [LBJH+15].

Norway has a backup solution for essential communication, the emergency public safety network. This network is built to be separate from commercial internet access and is mainly used by emergency services. It has about 60 000 users, and covers 86% of the land area in Norway [SHM21]. The network is going to switch from the current radio technology to 5G network slicing for Mission Critical Services (MCX) [RTW+19].

The backbone network in Norway is delivered by Telenor, and there are not enough alternatives for communication if there is a failure in this network [LBJH+15]. We can assume a failure in the backbone will lead to several parts of the country being without network connectivity for the duration of the failure period. Unfortunately, the architecture of the transport network is not public knowledge, and an exact model is therefore impossible.

The 5G network in Norway will be available for everyone in 2024 and provides specialized solutions for different services with increased flexibility [Aut22]. The 5G base stations provide an autonomous transport network, enabling local communication within a base station's area and communication between connected base stations. Software Defined Networking (SDN) enables flexibility in routing and increases the

flexibility in case of failures. Low orbit satellite systems are on the rise, and several projects will, in a few years, be able to provide satellite broadband access to end users in Norway. [Aut22]

Uninett develops and manages the Norwegian research network. Uninett is independent of the transport network delivered by Telenor and uses Global Connect, formerly known as Broadnet, for access to fiber infrastructure. The network has high reliability, with three sets of fiber between each part of the country [Aut17]. Even though this network is independent of the core network, the topology is public knowledge. In addition, the research institutions are located in the largest cities, making the network a good estimator of the core network.

An internet user can select several access technologies. 99% of all households can choose between three or more access technologies with more than 10Mbit/s download capacity [AK20]. In addition, there are several existing alternative systems for achieving communications in contingency situations in Norway. With this, we can conclude that the general network connectivity in Norway is well established. A potential backup solution could come in the form of compatibility with the Norwegian emergency network or another form of communication.

4.1.2 Mediums of payment

Wearable technology is anticipated to be the future of proximity mobile payments [LLT+22]. Physical cards are the dominating form of payment, requiring no form of electricity [RW14]. Near Field Communication (NFC) technology enables payment though mobile phones [PVP15]. The same technology is used in wearables to enable payment through rings, bracelets, clothes with integrated chips, or other potential wearable devices. Cryptocurrencies, in general, are a new method of payment used in the digital world [AD19]. The adoption by end users is growing exponentially, but the mediums of payments have stayed the same in a point of sales situation.

In the context of a Norwegian CBDC for offline usage, we are assuming a more complex medium than a card or a chip without a connection to electricity. This could be a phone or a wearable object with the possibility to process transactions and make computations.

4.2 Network topologies to simulate Norway

As discovered in Section 2.6.3, real networks are not random but either constructed or created by several evolving factors. A network of social interaction is often created by people interacting with both friends and new connections. To create such a topology, the Barabàsi Albert preferential attachment model (BA) creates a topology where

new nodes attach to an amount m of new nodes preferentially weighted after node degree. This topology will serve well to simulate a local P2P payment network.

From Section 4.1.1, the network access in Norway is generally connected by the core network with few alternatives. This is assuming the exclusion of satellite connectivity, as this requires specialized hardware, e.g., antennas and modems. To simulate a point of sales situation, if we include peer-to-peer transactions, a BA model can be adequate for the purpose. For a complete simulation of Norway, Uninett could be used as an abstraction of the backbone network in Norway, with BA graphs to simulate the users and the interconnections. The Uninett graph is close to a tree topology, and the endpoints could append a BA graph with varying sizes to represent cities. For a simulation purpose, the abstraction of a backbone network serves little to no purpose, as several runs with a BA graph connected to a central node would provide the same simulation as several independent BA graphs loosely interconnected. This assumes no interconnection between the BA graphs. Because we assume physical proximity to do transactions, there would be no need for interconnectivity between BA clusters. Therefore, a BA graph with a separate network representing internet connectivity would be sufficient for the modulation.

4.3 Level of centralization

One of the most important architectural design considerations of any CBDC is its level of centralization. As we have seen in Section 2.1 several approaches to the level of centralization can be found in existing CBDC projects and research. The level of centralization covers the overall architecture of a CBDC, that is, whether it is a distributed network in any form or a more traditional server-client architecture. Among the distributed network approaches, there are variations on who has access, both to the ledger itself but also to append to the ledger. In a DLT-based solution, only those actors capable of writing to the ledger can directly process transactions.

The level of centralization and the infrastructure with which the CBDC system is available determines the availability of the CBDC. With regard to the offline problem, ensuring a high level of availability and reliability is a preferable alternative, given the many difficulties associated with offline transactions. Furthermore, long-lasting and large-scale system outages and disconnectivity make it particularly difficult to ensure security. Any measure taken to reduce the occurrence of such scenarios will therefore reduce the risk associated with offline functionality. Essentially, the architecture of a CBDC can remedy some of the risks posed by offline functionality by reducing the frequency and longevity of offline usage.

Examples can be found in existing CBDC projects of this principle being utilized. China's CBDC, the e-CNY, utilizes geographically separated high-availability data

centers to reduce the need for offline functionality. The Bahamian CBDC, the Sand Dollar, has built designated infrastructure to ensure access to the CBDC system even without a regular internet connection. [WG20]

4.4 Distribution degree in simulation context

In the simulation, one goal is to make a run with a varying degree of intermediaries and the number of intermediaries each node is connected to. With this, the goal is to measure the marginal benefit per extra intermediary, i.e., an entity capable of processing transactions, representing an internet connection for users, and preventing offline situations. The most efficient way of solving the impossibility of a 100% reliable offline transaction problem is to reduce the probability of being offline. A more reliable system is a more decentralized system with several path redundancy mechanisms and backup solutions if the primary solution fails. Intermediaries in several nodes with different connections to end users are one way of securing active backups.

4.5 Modeling processing nodes

To model the architecture, it was concluded in Section 4.2 that varying amounts of connectivity from each node in a naturally formed group would be sufficient for simulating connectivity to an online context. Depending on the experiment, an intermediary could have direct access to the primary execution of processing online transactions or forward this to the appropriate server. Effectively these types of nodes have two states, up and down, for the purpose of the model. Queuing delay and congestion mechanisms are out of scope for this model, and a two-state Markov model with an M/M/1 queue would be a good model. The arrival of faults in the system is assumed to occur with a Poisson process with the rate λ. The restoration time in the system is assumed to be an exponentially distributed process with the parameter μ. A state diagram of the two-state Markov model is presented in Figure 4.1. For the M/M/1 Markov chain, the transient solution can be described by the equation:

$$p_0(t) = \frac{\lambda}{\mu + \lambda} \cdot e^{-(\lambda+\mu)t} + \frac{\mu}{\mu + \lambda} \tag{4.1}$$

as outlined in [EHHP09], where λ represents the failure rate of the node, $p_0(t)$ is the probability of being in state 0 at time t, and μ represents the recovery time of the node. The stationary solution to this problem occurs when time goes to infinity, resulting in the equation:

$$p_0(\infty) = \frac{\mu}{\mu + \lambda} = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda} + \frac{1}{\mu}} \tag{4.2}$$

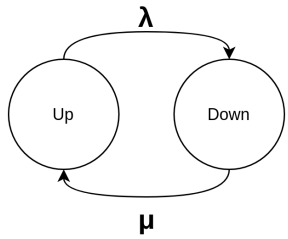


Figure 4.1: Two state Markov model

This solution is the expected time in state 0, divided by the total time in both states. This results in the equation forming the total percentage of time in an up state. Each node represents a system of independent paths to the processing of transactions. Therefore, a single node up would be sufficient for processing the transaction, and the transaction is in an online context.

For a user connected to multiple processing nodes, each node represents an independent system with independent failure and recovery rates. In the model, we assume each intermediary would fail independently, as they represent independent systems or entities with standalone access to a central server. The recovery rate is assumed to be independent based on the aforementioned argumentation. In Figure 4.2 a diagram of n independent sources with the previously mentioned conditions are shown. If a single client is connected to the following processing nodes in parallel, a single up state is sufficient for an online state. From the reliability block diagram in Figure 4.3 we can see that a single node is sufficient for connectivity in this model. With the parallel logic, we can formulate the total availability of n nodes based on μ and λ. The equation for the total availability of n identical parallel components can be computed by the equation:

$$A_{parallel} = 1 - \Pi_{i=1}^{n}(1 - A_i) \tag{4.3}$$

The availability of a single processing node is a transient solution to the steady-state equation of the up state. We can insert the availability in Equation 4.3 with the transient steady-state Equation 4.2 to form the total availability of n parallel components in the system in Equation 4.4.

$$A_{parallel} = 1 - \Pi_{i=1}^{n}(1 - \frac{\mu}{\mu + \lambda}) = 1 - (1 - \frac{\mu}{\mu + \lambda})^n \tag{4.4}$$

For the testing purposes of a completely simultaneously disconnected state, it is convenient with a single point of failure. The additional node would fail independently

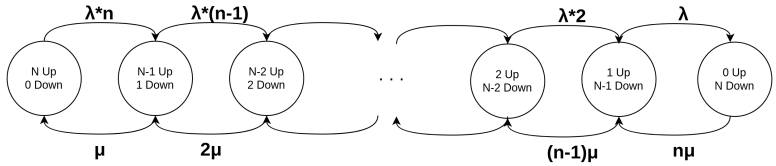


Figure 4.2: State transition diagram of a model with n independent sources

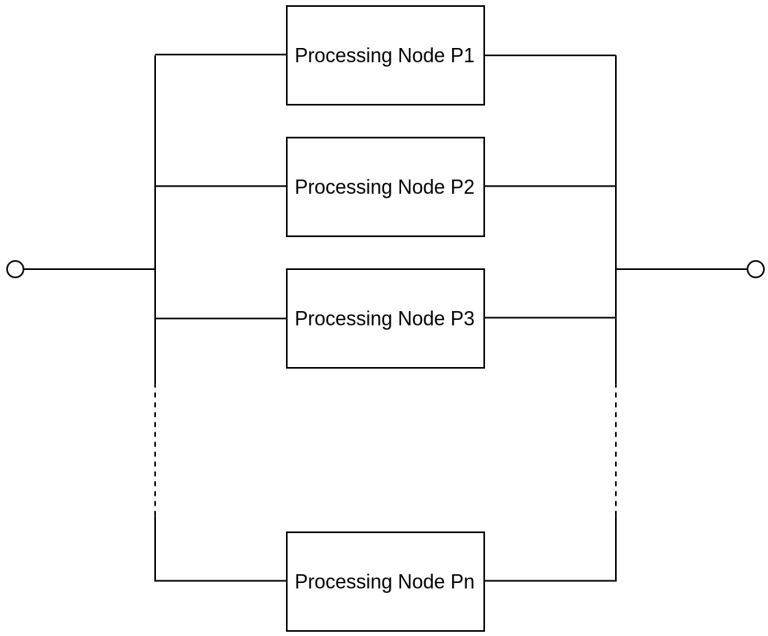


Figure 4.3: Reliability block diagram of a parallel system

of the processing nodes and will be referred to as the central node. The central node fails with the probability λ_c and recovers with the probability μ_c. The state model is equivalent to a single node, two-state Markov model as displayed in Figure 4.1. The processing nodes fail with the probability λ_p and recover with the probability μ_p. The resulting system's reliability block diagram is displayed in Figure 4.4. The two-state Markov model has the steady-state solution derived in Equation 4.2. The availability of the processing nodes and that of the central node is given by the solution of the steady-state model with its respective parameters. This leads to a total steady-state availability of the system of the two availabilities combined. The total uptime of the system is when both systems are up, as seen in the block diagram, and the resulting availability is derived in Equation 4.5.

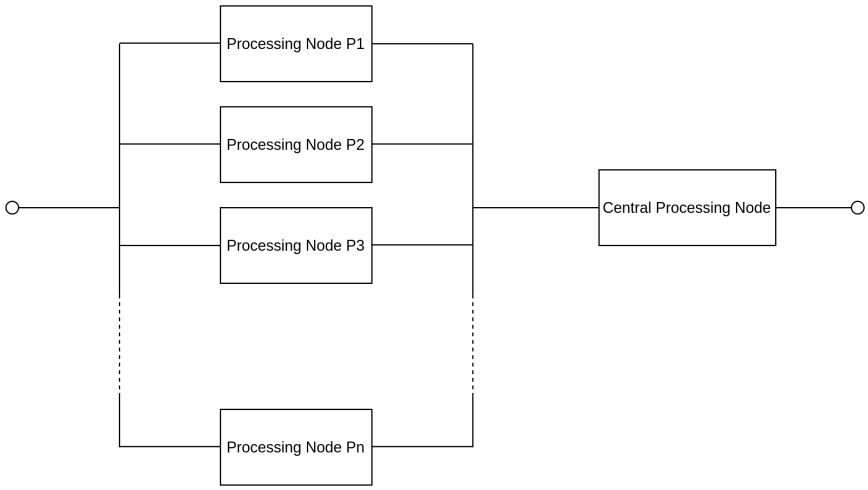


Figure 4.4: Reliability block diagram of a parallel system with a central point of failure

$$A_{total} = A_{parallel} \cdot A_{centralnode} = (1 - \Pi_{i=1}^{n}(1 - \frac{\mu_p}{\mu_p + \lambda_p})) \cdot \frac{\mu_c}{\mu_c + \lambda_c} \tag{4.5}$$

4.6 Key takeaways

In this chapter, the architectural design characteristics of a Norwegian CBDC have been discussed. General architecture considerations are also reviewed. The network context for a simulation of a realistic architecture and the provided network connectivity is discussed. A state-based model for the availability of processing nodes to be used in a simulation context, and the equations for computing the availability of such a system, have been derived.

Chapter 5
Protocol design

This chapter will present our protocol solution based on the best effort approach to an offline protocol for a Norwegian CBDC. The solution is based on the requirements gathered in Chapter 3 and the current state-of-the-art research presented in Chapter 2. The chapter begins with a broader discussion of the technical possibilities, challenges, and trade-offs posed by the requirements and research presented so far. Specifically, a few key considerations will be outlined, and the design choices made in the protocol with regard to these will be substantiated. The protocol will then be presented, first through a general overview, then a more detailed explanation of its individual components including the security mechanisms.

5.1 Considerations and trade-offs

5.1.1 Token vs account based solutions

As we have seen in Section 2.4 and 2.1.3, there are various approaches to how the storage of value and transactions should be arranged in the existing literature. One particular consideration is the difference between token based and account based solutions. Token based solutions, such as those presented in [IDLK21] and [PG+99], employ tokens of value that are themselves exchanged. Ownership of the tokens is controlled by who can present the correct knowledge about their cryptographic construction and characteristics. In this design, the central authority typically signs individual tokens of value and provides them to the payer. The payer then transfers the tokens by some means to the payee. As the payee now has the necessary knowledge of the token, she can present these to the central authority for settlement.

The other solution, account based systems, like the ones presented in [CGK+20] and [LWZ+21], do not operate with discrete tokens of value but rather with accounts that have a modifiable balance. With these types of solutions, the current balance of any account has to be stored in a manner that is tamper-proof as any entity, without such protections, is capable of creating universally accepted funds. This kind of

system typically operates by all parties signing transactions from their accounts as proof that they have agreed to the transactions. Further security measures, such as counters to maintain chronologicity, are commonly implemented.

This consideration is certainly a trade-off, as both solutions offer different advantages and disadvantages. Token based solutions have inherently better security in the sense that they do not necessarily require tamper-proof systems and can rely on cryptography alone to combat double spending and forging attacks. They can, however, as discussed in Section 2.4.3, not prevent double spending in real time while offline if multiple payees are targeted. This means that any token based design still requires some form of secure hardware or other enhancements to prevent double spending in real time, which is a necessity if the finality of transactions is desired. Furthermore, we have seen that the most sophisticated token based approach, Chaum's protocol, does not allow for offline respendability of tokens. It is conceivable that such a solution could allow for offline respendability, but based on existing research, such an enhancement without negatively affecting the security of the protocol is not achievable.

Account based solutions are less complex in the sense that they require less cryptographic calculations and operations and, consequently, less computational overhead. This is a major consideration with secure hardware, as memory and computational power often are limited. Account based designs' relative simplicity also allows for a simpler protocol, in more general terms, which is a benefit in a system that will, in all likelihood, be used and implemented by multiple entities and providers.

Another consideration is the atomicity of funds, meaning at what granularity they can be transferred. In any token based solution, the minimum amount of any transaction is the value of the token. This problem can be functionally circumvented by making the value of each token minuscule, but this would dramatically increase the computational overhead for cryptographically demanding solutions. Account based solutions do not share this problem, as the balance of any account can be altered by any amount. A hybrid solution in which tokens are created without a predetermined value but are ascribed a fixed value when needed in a transaction is conceivable and would share some similarities to a traditional checkbook. Technically, the border between token and account based solutions gets a little blurry in such a solution, and it might represent a suitable compromise between the two. However, more research into this approach is needed.

In conclusion, the advantages provided by a token based design are not sufficient to counter the increased complexity. Furthermore, both solutions will require additional security mechanisms to prevent double spending in real time. However, due to the

relative simplicity of account based solutions it can be argued that such appendages are more easily implemented with that architecture, though this depends to some degree on the mechanism in question. As both solutions require secure hardware to satisfy the requirements of an offline protocol for a CBDC an account based solution is, based on existing research, the more convenient approach, from a technical perspective, and thus the one that will be used in the protocol proposed in this thesis.

5.1.2 Traceability and accountability

Even in a CBDC offline solution protected by secure hardware, fraud, and attempts at such, are still likely to occur. The potential gain for any malicious actor by breaking secure wallet protection is enormous. Therefore, one can assume that significant efforts will be put in to bypass the system's security mechanism, placing a need on contingency procedures and mechanisms necessary when the secure hardware is penetrated. One significant technical challenge in designing an offline solution for a CBDC is how to ensure transaction traceability, accountability, and non-repudiation. In the simpler approaches reviewed in Section 2.4 the central authority has no means of definitively knowing which user committed fraud, even though they may easily detect the fraud after the fact. As presented in Section 2.4.3, solutions exist with traceability and accountability built-in. Here, the matter in which each token is spent will not only reveal if fraud has taken place but also which party in the transaction acted fraudulently.

The account based solutions reviewed in Section 2.4 have little or no support for traceability or accountability in the event of secure hardware or general protocol failure. In the design presented in this chapter, mechanisms are introduced to provide this. Specifically, a system of user maintained logs would be utilized to ensure traceability and accountability, in addition to other preventative security measures. The contents of these logs are continuously shared with other users through transactions, ensuring all users maintain as complete a picture of the state of the offline system as possible. Furthermore, by relaying these logs to the central authority upon regaining connection, the central authority can determine, in case fraud was committed, who the responsible party is. There are potential issues with such an approach, mainly that users can frame other users by manipulating their own logs, making others look fraudulent. This is, however, remedied by the victim of such fraud informing the server of its own logs, which will, through a system of certificates and signatures, prove its innocence and restore correct accountability to the system. The exact mechanisms with which this is done will be further explained later in this chapter.

5.1.3 Respendability

Respendability is the reuse of funds received in an offline context in the same offline context. This opens up for propagation of faults through legit transactions. If funds are later discarded as fraudulent, the following transactions might be considered discarded as well. Disallowing this would break with the requirements in Section 3.3 and would potentially harm legitimate users. This implies the liability of faulty transferred funds for the legitimate user is with the issuer of the funds or the liable part of the system. The responsible party for the system security would then be obligated to retrieve the falsely created funds from the responsible person, as the protocol provides traceability of funds and non-repudiation of sent transactions.

5.1.4 Transaction limits

Transaction limits, such as a limited number of transactions per client per day, or limits on the volume allowed in each transaction, can and should potentially be implemented in the offline functionality of a CBDC. Such limitations can significantly decrease the financial risks associated with the technical risks of the system. It is clear that no offline CBDC solution exists without at least some risk of abuse and fraud due to technical limitations. Any measure taken to limit the potential damage caused by such abuse or fraud will decrease the financial risks of the users and providers of the system.

For this reason, transaction limits should be implemented in any CBDC offline solution. It is, however, not included in the protocol here presented, as implementing such rules in an existing protocol is fairly trivial. Furthermore, such limits also do little to prevent fraudulent transactions but rather limit their potential when an attacker has already circumvented other security measures of the system.

5.1.5 Privacy

Another consideration in the design of an offline protocol is the privacy of individual users. Chaum's protocol, presented in Section 2.4.3, allows for complete anonymity of any non-fraudulent user in the sense that transactions cannot be attributed to any one identity. Other solutions allow for a lesser degree of privacy but mostly require a central authority to have complete information of all transactions. Generally, in the literature, there seems to be a trade-off between privacy and security. In an offline state, users cannot rely on a trusted third party for fair exchange and double spending prevention. Thus, they must individually assess the legitimacy of each transaction through some set of procedures or mechanisms. It can be argued that the more information about other users and their transactions are hidden and concealed, the harder, and thus less effective, such assessments become. This trade-off, its effects, and its potential solutions will be further discussed in Section 7.4.

Privacy and confidentiality are derived non-functional requirements. Despite this, they are not prioritized in the protocol suggested in this thesis. This is a design choice made to limit the scope of the design process due to the overall limitations of the thesis. We here prioritize security and functionality, as well as other specific requirements such as finality of transactions, and delegate privacy to further research. We recognize that double spending and similar attacks are the most pressing challenge associated with offline transactions and that any potential solution should first and foremost aim to combat this. It is conceivable that measures can be added to the protocol here suggested to provide some level of privacy.

5.2 Protocol overview

Based on the gathered requirements and considerations deliberated above, a detailed protocol will in this section, be presented. The protocol consists of mainly four sub-protocols, namely deposits, payment, withdrawal, and synchronization. These will be covered in depth in the following sections. To define the behavior and mechanisms of the protocol, we limit its scope to a closed system with a central server and an arbitrary number of users or clients. It is assumed that all clients have access to secure hardware, which will be further outlined in Section 5.6.1. The protocol covers transactions between clients only in settings where both parties are offline, as defined in Section 4.1. A client being offline, in this case, means that she does not have a connection, and therefore not the ability to communicate, with the central server and the wider CBDC system. The client still needs a connection to other clients to perform a transaction with said clients. The protocol imposes no limitations on what kind of connection clients have with each other and should provide functionality for a range of communication methods and use cases.

The protocol draws considerably on the ideas of the OPS presented in [CGK+20]. There are, however, enhancements added to improve the protocol to satisfy the requirements. The OPS does not provide traceability in the event of fraud. In addition, it affords no protection against abuse of the protocol in lasting offline scenarios or in the event of secure hardware breaches. Mechanisms to mitigate lasting and propagating faults are added in the proposed protocol, along with other changes, adaptations, and improvements.

The OPS paper, [CGK+20], does very little estimation of the security of the OPS protocol. There is no testing or simulation of any kind to evaluate the effectiveness of the security measures implemented. The paper is not peer-reviewed, submitted to any journal, and generally has few citations. We therefore have a poor understanding of the level of security of the foundation of the protocol here presented. From the article, it is clear that secure hardware is the main method of combating double spending. Thus, any estimation of the security of secure hardware, as presented

in the paper, could serve as an entry point to any analysis of the risks posed by applying the OPS protocol.

The protocol is based on a certificate hierarchy, where the central server provides client certificates for each client. Each client can then prove that the certificate is from the server, and this provides the basics for authentication. By using the certificate for signing messages with a signed integrity check, using a Hash based Message Authentication Code (HMAC) signed with the client keys, the integrity and authenticity of the message is provided, if the client also provides their certificate.

The protocol requires each client to have two sets of keys with the certificates. One for the secure hardware and one for the client itself. A message signed by the secure hardware should not be editable by anyone after the creation. The combined integrity and authentication check proves this for anyone who can verify the server certificates with the server's public key. The client has a separate certificate as the client should not have access to the secure hardware's keys, and the client also needs to sign messages in the protocol.

The protocol assumes that messages and transactions sent are received. That is, the protocol has no functionality to remedy the loss of messages in transit. Similarly, the protocol assumes that there are no other errors in the execution or communication, apart from malicious activity that will be discussed throughout this chapter. In a real-world implementation, this type of functionality must either be appended to the protocol or provided by some other underlying communication protocol. Furthermore, the protocol has no confirmation messages sent in response to a received transaction. Replay of messages should be provided by the underlying transport protocol, for example, TCP with the TLS extension for handling both integrity, confidentiality, and authenticity by using the client and server certificates. A secure version of UDP could also be utilized with an extension to this protocol for asking to resend packages. As this protocol does not require an acknowledgment of received packages, this opens the possibility of clients denying having received funds, even if provable by the sending party.

Each client's secure wallet maintains a set of counters. One counter to track its own transactions, and additional counters to track all other clients whom it engages in transactions with, including the server. The counters are utilized to maintain chronologicity inside the secure wallet. This is done as a security measure in and of itself, allowing each secure wallet a way of detecting replayed transaction messages, but it is also necessary for the sorting and maintaining of logs. Each client's individual counter, that is, the counter of its own transactions, is updated with every transaction sent and received. The counters each client stores for other clients are updated based on the last received counter value for any given client.

The protocol uses an account setup where each client has three separate accounts. The first is the online CBDC account. The protocol is agnostic with regard to what technology is used for online accounts, i.e., the overall architecture of the CBDC. Secondly, each client has an account with the server, in this case representing the central bank, payment providers, or other intermediaries, that stores their offline balance. Therefore, this is only available to the clients when online and holds the online equivalent funds to be used while offline. The last account is the offline, secure hardware wallet stored on the client's device. This account mirrors the offline account stored on the server and will have the same balance if the client is online. This secure wallet synchronizes with the client's offline account with the server through the deposit, synchronization, and withdrawal sub-protocols.

Each client maintains a log of all offline transactions it is involved in. The log is sent with each transaction from the payer and then appended to the payee's log. This is done as a security measure to detect and prevent double spending and fraudulent transaction through a mechanism that will be explained in Section 5.6.2. Due to the technical limitations of secure hardware solutions, the logs are not stored in the secure hardware. The logs are stored as a directed graph. Each client maintains her own directed graph with a complete transaction history of her own transactions, as well as the transaction history of all other clients she has received payments from. Upon receiving a transaction, the payee will begin the validation procedure. After validating the log and the transaction, the payer's log is added to the payee's graph to avoid double storage of the data. The detailed means with which this is done will be explained in Section 5.5.

5.3 Setup and deposits

5.3.1 Client setup

The client needs the application installed on an appropriate device with the required secure hardware capabilities. The client setup needs to be run once per client as an initial setup. The server key pair should be approved and preinstalled with the application. The client will then create a local key pair and request signing of the signing key from the server. The server verifies the identity of the client and creates the client certificate based on the identity and the public key. The server keeps the register of all created clients and their respective signing key, along with a counter for each client and their offline and online balances. The client then provides a TEE-attestation, a proof that the secure hardware is not tampered with, and sends the public key of the TEE. The server attests the authenticity of the TEE and the TA registers with the server the created key pair.[1]

[1]The TEE and TA registration is done as described in depth in the article [CGK+20]

Protocol variables

Variable	Description
S	Server
(skS, pkS)	Signing key pair of server
S.reg	Server's register of all approved offline clients
SiC_i	Server's counter for client i
C_i	Client i
C_i.log	Log of transactions kept by client i
BC_i	Online account of client i
TC_i	Offline account of client i stored at server
SC_i	Secure hardware wallet of client i
BC_i.bal	Online balance of client i
TC_i.bal	Offline balance of client i stored at server
SC_i.bal	Secure hardware balance of client i
SC_i.i	Counter of secure hardware of client i
S$C_i.cl_j$	Log of counter of j kept by secure hardware of client i
(skC_i, pkC_i)	Signing key pair of client i
(sh.skC_i, sh.pkC_i)	Signing key pair of secure hardware of client i
C_i.prev_hash	Hash of previous transaction
C_i.blacklist	Blacklist kept by client i received from server

Table 5.1: Protocol variables with descriptions

Protocol actions

Action	Description	Scope
sig(singing key)	Produces a signature on a message using a given key	Any
vsig(transaction, public key)	Validates signature of a message	Any
log_tx(transaction, counter)	Logs complete transaction data with received log and internal counter	Client
rollback_log(transaction)	Remove a transaction and associated log from log	Client
transfer(from, to, amount)	Performs an online transaction	Server
withdraw(amount)	Triggers withdrawal protocol from secure hardware	Client
validate_log(log)	Validates that received log is legitimate	Client
broadcast_log(log)	Broadcasts log	Client
H(*)	Standardized hashing algorithm	Client

Table 5.2: Protocol actions with descriptions

Protocol messages

Messages	Description	Sender
req_dx[from, to, amount, signature]	Requests a deposit to offline account	Client
dx[from, to, amount, counter, signature]	Deposit transaction	Server
wx[from, to, amount, counter, sh_signature]	Withdraw transaction	Client
tx[from, to, amount, counter, sh_signature, log.index, signature]	(Payment) transaction	Client

Table 5.3: Protocol messages with descriptions

5.3.2 Deposits

Before a client can engage in any offline transaction, she will need to make a deposit to her offline account. This is done by transferring funds from her online account to an account held by the server dedicated for the client's offline use. Despite the fact that the account is an online account registered in the client's name, transactions from this account, i.e., withdrawals, can only be done with the authorization of the central server. If a blockchain based online architecture is used, this can be achieved by utilizing accounts that require multiparty signatures to perform a transaction. If a more traditional server architecture is used, implementing this account limitation is trivial. The withdrawal process will be outlined in Section 5.4.

A deposit transaction is initiated on the client's device by sending a deposit request (req_dx) to the server. This request triggers the server to perform a transaction between her online account and her offline account stored with the server. Depending on the broader architecture of the CBDC, this might require additional signatures for the client's online account. Upon receiving this request, the server confirms that the client is registered with the server and therefore approved for offline transactions and that the client has sufficient funds in her online account. If these requirements are satisfied, an online transaction is performed to transfer funds from the client's online account to the client's offline account. The server then sends a deposit transaction to the client. This transaction includes the sender, i.e., the client's online account, and the receiver, i.e., the client's offline account. It also includes the amount and a counter. The counter is included to prevent replaying the deposit message and creating illegitimate funds. Each deposit from the server to any given client increases the counter for that client. Lastly, the transaction message is signed by the server to confirm that it is a legitimate transaction from the server and to prevent any malicious actor from changing the content of the message in transit.

Upon receiving the deposit transaction, the client passes the transaction to her secure wallet, which verifies the signature and the correctness of the message. The client's secure wallet also verifies that the received counter value is larger than those previously received from the server. This step is not performed for the very first deposit transaction made to that client, as no earlier counter value will be stored. The new counter is stored on the client's secure wallet, and the balance of the secure wallet account is increased accordingly. Finally, the client appends the deposit transaction to her log.

Pseudocode for the deposit protocol is shown in Algorithm 5.1.

Algorithm 5.1 Deposit Protocol

C_i sends req_dx[BC_i, TC_i, x, $sig(skC_i)$] to S
if $C_i \in$ S.reg & $vsig(req\_dx, pkC_i)$ & BC_i.bal $\geq$ x **then**
  $transfer(BC_i, TC_i, x)$
  S sends dx[BC_i, TC_i, x, SiC_i, $sig(skS)$] to C_i
  if $vsig(dx, pkS)$ & SiC_i > S$C_i.cl_S$ **or** SiC_i **not in** S$C_i.cl_S$ **then**
    SC_i.bal ← SC_i.bal + x
    S$C_i.cl_S$ ← SiC_i
    SC_i.i ← C_i.i + 1
    $log\_tx(dx, SC_i.i)$
  end if
end if

5.4 Withdrawals

Once a client has reestablished a connection to the server, it can redeem the funds it may have received during an offline session. This is done through the withdraw sub-protocol. This protocol largely does the opposite of the deposit protocol. The protocol is summarized in Algorithm 5.2.

Any withdrawal begins with the client executing the *withdraw*() function, specifying a sum, which triggers a process in the client's secure hardware. The client's secure hardware wallet checks if there are sufficient funds to withdraw the desired amount. The amount is then detracted from the balance of the wallet. The secure hardware wallet then generates a withdraw transaction, wx, containing the from and to account, in this case the client's offline balance stored at the server and the client's online account respectively. The transaction also includes the client counter and a secure wallet signature. Outside the secure wallet the client appends a hash to the transaction to maintain the client's hashchain. The client log is also added as a proof of validity of the claim the client is making to the server. Finally, the transaction is signed using the client's keys and sent to the server. This transaction, as it is considered an outgoing transaction, is added to the client's log.

Upon receiving the withdraw transaction, the server confirms the validity of the transaction, i.e., the signatures, counter, hash, and log, and confirms that there are sufficient funds in the client's offline account with the server to cover the transaction. The funds are then transferred to the client's online account.

5.5 Payment

For typical transactions, such as payments and transfers, the payment sub-protocol is used. This protocol facilitates the transfer of funds from one client to another and

Algorithm 5.2 Withdraw Protocol

C_i exec $withdraw(x)$
if SC_i.bal $\geq$ x **then**
  SC_i.bal $\leftarrow$ SC_i.bal - x
  SC_i.i $s \leftarrow$ SC_i.i + 1
  C_i sends wx[TC_i, BC_i, x, C_i.i, $sig(sh.skC_i)$, $H(this.wx, C_i.prev\_hash)$, C_i.log, $sig(skC_i)$] to S
  C_i exec $log\_tx(wx, SC_i.i)$
  if $vsig(wx, sh.pkC_i)$ & $vsig(wx, pkC_i)$ & TC_i.bal $\geq$ x **then**
    $transfer(TC_i, BC_i, x)$
  end if
end if

is algorithmically described in Algorithm 5.3. The protocol is concerned only with the transfer itself and imposes no limitations on the prelude to that transaction. For instance, a typical payment might be initiated by the payee as a request for funds, yet the protocol only outlines the transfer as initiated from the payer's side. It is therefore agnostic with regard to the wider payment system and transaction rules and flexible in the sense that it can be used in a variety of scenarios. To perform a transaction, the payer must know the account address of the payee. This can be known in advance or discovered through some other preluding protocol to the payment. The payer triggers the payment on her device, which again triggers the process in her secure wallet. The secure wallet checks if there are sufficient funds to complete the transaction. If this is the case, the amount is detracted from the balance in the secure wallet. A transaction message is then constructed in the secure wallet of the payer. This transaction includes sender and receiver accounts, amount, and the payers counter. This is signed by the secure hardware signing keys of the payer to prove to other clients that the transaction is originating from an approved secure wallet. Outside the secure hardware, the log of the payer is appended along with the transaction hash. The transaction is then signed with the payer's key. The payer sends this signed transaction to the payee.

Upon receiving the transaction, the payee validates the signatures, both the secure hardware signature and the client signature. The payee then performs a check for the payer in her own blacklist. If the payer is not blacklisted, the received log is appended to the existing log, and the entire log is broadcasted. The log is then validated. The details regarding how these steps are performed will be outlined in the following section. If the above mentioned security mechanisms do not cancel the transaction, it is checked in the payee's secure hardware for the correct expected counter value.

Upon completing all security measures, the transaction is accepted by the payee,

and the balance and counters are updated accordingly. If the validation of the transaction fails, the transaction is rejected, and the received log, already appended to the payee's log, is rolled back as the transaction is invalid.

Algorithm 5.3 Payment Protocol

C_1 initiates transaction to C_2 of amount x
if SC_1.bal $\geq$ x **then**
  SC_1.bal $\leftarrow$ SC_1.bal - x
  SC_1.i $\leftarrow$ SC_1.i + 1
  C_1 sends tx[TC_1, TC_2, x, C_1.i, $sig(sh.skC_1)$, $H(this.tx, C_1.prev\_hash)$, C_1.log, $sig(skC_1)$] to C_2
  if $vsig(tx, sh.pkC_1)$ & $vsig(tx, pkC_1)$ **then**
    if C_1 **not in** C_2.blacklist **then** ▷ Server blacklist
      C_2 exec $log\_tx(tx, SC_i.i)$
      C_2 exec $broadcast\_log(C_2.log)$ ▷ Collaborative security
      validation_result $\leftarrow$ C_2 exec $validate\_log(C_2.log)$
      if C_1 **not in** validation_result **then** ▷ Client prevention
        if C_1.i > $SC_2.cl_1$ **or** C_1.i **not in** $SC_2.cl_1$ **then** ▷ SH security
          SC_2.bal $\leftarrow$ SC_2.bal + x
          $SC_2.cl_1 \leftarrow C_1$.i
          SC_2.i $\leftarrow$ SC_2.i + 1
        end if
      else
        C_2 exec $rollback\_log(tx)$
      end if
    end if
  end if
end if

5.6 Security measures and enhancements

5.6.1 Secure hardware

Secure hardware is used to ensure a safe execution environment with a signed program, ensuring no modifications to the program before or during the run. A secure hardware is described in Section 2.5. The secure hardware allows for secure software execution, and we can assume it to be close to tamper-proof. The hardware often has a limited capacity in regard to storage and processing, and the program should therefore be simple. The secure hardware increases the complexity of tampering with the system and reduces the probability of a successful attack on the system. Without secure hardware, it would still require specialized knowledge to inject code into a software program, depending on the program and execution environment. The secure hardware does not provide a safe protocol by itself, as it is still assumed to be

breakable in the future. The protocol needs further attack detection and prevention to reduce the consequences of an attack.

5.6.2 Distributed attack prevention (DAP)

In addition to secure hardware, to ensure protection against double spending, logs are added as a measure to detect and trace such attacks. Due to the massive potential gains to be had by circumventing or breaking the secure hardware protections, it is conceivable that malicious actors occasionally will succeed in doing so. Therefore, additional measures, like log-keeping, to detect and prevent forging and double spending attacks should be included. The purpose of clients maintaining and sharing their own log of transactions is to detect other clients who, by bypassing the secure hardware limitations, overspend the balance of their accounts or double spend the same transactions.

Upon receiving the transaction message and the log from the payer, before the transaction is passed to the payee's secure wallet, the payee performs the log validation process. The first step of this process, which is in its entirety outlined in Algorithm 5.4, is to validate the signature on the log. This confirms that the payer guarantees that the contents of the log are validated and approved by the payer. The log is then sorted in topological order so that all transactions are in chronological order. This allows for traversing the graph and accounts for all funds in the enclosed sub-system represented by the log. A list of each client represented in the log, including their counter and balance, is created. For each transaction discovered in the traversing of the graph, the signature for that transaction is validated. This confirms that whomever that transaction is supposed to be from has indeed sent that transaction. The counter sent with that transaction is checked against the last stored counter for that client to ensure that no client has respent the same transaction or forged transactions with a lower than true counter value.

Each transaction includes a hash of the transaction data and the previous transaction hash, forming a hashchain for each client's transactions. The purpose of this hashchain is to assure non-repudiation on the order of transactions for any client. The payee confirms that the payer has sent her complete output transaction log by recalculating all hashes. If the payee cannot completely recalculate the hashchain she knows that the payer has omitted at least one transaction from her logs and will subsequently discard the transaction. This also ensures that the payee cannot alter the logs of the payer after receiving them without breaking the hashchain. A malicious actor can still create valid but illegitimate hashes by using the same previous hash in multiple transactions, essentially creating multiple forks of her own hashchain. This would only be uncovered by the payee if she already knows about one of the other forks. However, it will provide clear evidence to the server that a client has created

illegitimate transactions when it is presented with multiple, inconsistent hashchains created by the same client. It is conceivable that the counter already implemented in the secure wallet can be used in a similar manner as the hashes here described and that this measure is therefore redundant. This would, however, impose strict limitations on counter chronologicity and behavior. It would then not be possible to increment the counter on receiving transactions, which is necessary to sort the logs. Furthermore, as the counter is implemented in the secure wallet, it is desirable that this functionality remains as simple and free of computational and memory overhead as possible. Hashes are therefore used to ensure this non-repudiation of the order of transactions.

The last stage in the log validation is accounting for the balances of all clients present in the log. If the log is complete and no fraudulent transactions have occurred, all clients should, at all times through the traversing of the graph, have a non-negative balance.

If any of the above mentioned steps fail, the payee will look at what client in the logs caused the irregularity. If the client responsible for the failure of validation is the sender of the transaction, the payee rejects the transaction and removes all received log entries from her log. In this case, the payee will know that the payer has acted fraudulently as the protocol has not been followed. If an irregularity is detected in the log but not originating with the payer, the transaction is accepted. Because the payer, in this case, had no way of knowing it had received fraudulent transactions, the payee accepts this transaction to provide finality of transactions.

5.6.3 Centrally organized blacklisting

A lockout mechanism, such as blacklisting, is an efficient way of blocking out fraudulent users from the system. The server detects fraud by checking the signature of the transaction, checking the counter of the transaction, and the balance of the sender. If a sender has a negative balance after sending funds, the sender has done a fraudulent transaction. A server side blacklist could minimize the consequences of a fraudulent user, as the user could be efficiently blocked from interacting with any other user until the dispute is settled. By synchronizing the server blacklist with the users of the system, the users will be blocked from interacting with a previously known fraudulent user, and this feature increases the graceful degradation and fault tolerance of the system.

5.6.4 Collaborative attack prevention (CAP)

Based on the requirements outlined in Section 3.3 offline functionality should also extend to situations where users are offline for extended periods of time. This type of scenario poses additional challenges as the main means of detecting fraud,

Algorithm 5.4 Validate log

```
if vsig(incoming log tx, pkC_sender) then
    topology sort log
    fraud_clients = [ ]
    sums = {node: [balance = 0, counter = 0]}
    for tx in log do
        if vsig(tx, sh.pkC_tx.from) & vsig(tx, pkC_tx.from) then
            if sums[tx.from].counter ≥ tx.counter then
                fraud_clients ← tx.from
            end if
            sums[tx.from].counter ← tx.counter
            sums[tx.to].balance ← sums[tx.to].balance + tx.x
            if tx is not deposit then
                if tx.hash != H(tx_−1.hash, tx) then
                    fraud_clients ← tx.from
                end if
                sums[tx.from].balance ← sums[tx.from].balance - tx.x
                if sums[tx.from].balance < 0 then
                    fraud_clients ← tx.from
                end if
            end if
        end if
    end for
    return fraud_clients
end if
```

connecting to the central server, is unavailable. Fraudulently acquired funds can therefore propagate more extensively, and fraudulent agents have more opportunities to spend illegitimate funds. Because of this, a mechanism to detect and prevent double spending in extended offline cases is proposed as an extension to the protocol.

This mechanism relies on offline clients collaborating to detect and subsequently prevent fraudulent agents from spending illegitimate funds. The general idea of this mechanism is that clients, who cannot communicate with the central server but can communicate with each other, share information about transactions they are aware of so that they can detect seemingly fraudulent behavior. As stated, each client holds a log of all transactions it knows about, including her own as well as any sent to her. As explained in the previous section, this alone provides each client with some protection against double spending attacks. To amplify the effect of this type of detection, a mechanism to broadcast transaction logs to any clients willing and able to receive them is added. The precise means with which this is done is not specified, but it can conceivably be achieved using any short to medium ranged communication medium and protocol.

Upon receiving a payment transaction, the client will, in addition to all the steps already explained, broadcast her complete log, including the recently received transaction. The client will broadcast even the illegitimate part of any received log, if any, as proof to others that a client is attempting fraud.

This enhancement to the protocol prevents fraudulent users and illegitimate transactions from propagating and limits their potential damage to a system in an extended offline scenario. Clients can more easily detect fraudulent actors by having more data about transactions between other nodes. Attacks in which a client manipulates her own hashchain or counter to double spend the same funds to multiple parties will have a higher likelihood of detection, as each client has a more comprehensive image of the state of the offline system. The exact way in which this enhancement is implemented can be varied to account for limitations and desired characteristics of the protocol and its prerequisites. For instance, the broadcast can be extended to a two-way exchange where both clients share their logs. This would increase the effectiveness of the mechanism and even allow for real-time double spending prevention for certain attack scenarios. It would also, however, depending on the number of clients capable of responding to any broadcast, dramatically increase the communication and computational overhead of the protocol. Conversely, suppose these overhead costs prove too extensive given the proposed layout. In that case, a delay mechanism can be added so that clients begin broadcasting and listening for broadcasts only after they have been offline for an extended period.

5.7 Log synchronization

As stated, all clients send their complete logs with every transaction. Logs may also be broadcasted upon receiving a transaction. As any received log is added to the client's existing log, the logs of all clients will grow rapidly in any offline setting. To combat this, a mechanism to reset each client's log is introduced. This mechanism involves clients performing a synchronization procedure with the server whenever the connection to the server is reestablished after an offline session. This procedure, which is outlined in Figure 5.1, essentially creates a checkpoint transaction that allows the client to clear all previous transactions from her log.

The synchronization procedure is initiated by the client once the online status is resumed by sending a synchronization request to the server. Upon approval by the server, the client sends a withdraw transaction to the server for her entire offline balance. The server then immediately returns a deposit transaction with the same amount. The withdraw and deposit transactions used are the same as those outlined in Section 5.3.2 and 5.4, but they are triggered with the total offline amount of the user and happen automatically. Upon receiving the deposit transaction from the server, the client can clear her log, keeping only the deposit. The log will now be

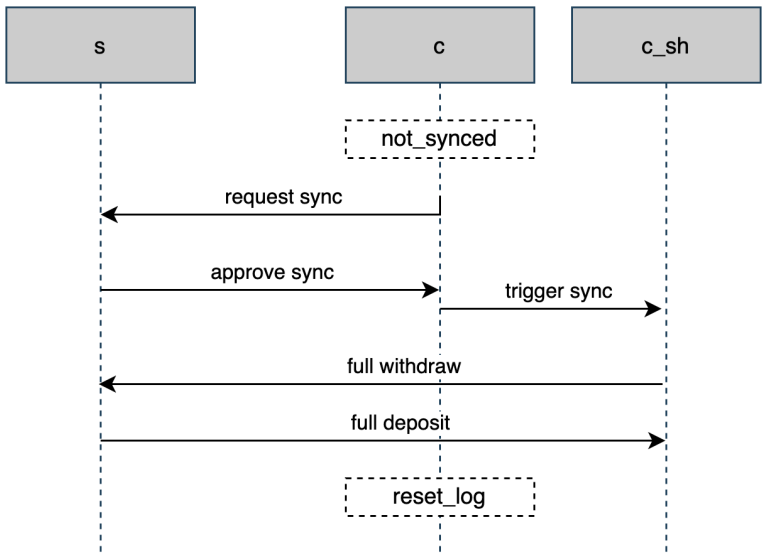


Figure 5.1: Sequence diagram of collect procedure

practically similar to any client who has never performed any offline transaction apart from an initial deposit. The client may not reset her counter and hash. The deposit should be distinguishable from other deposits so that any other clients are aware that a synchronization has taken place upon receiving the now cleared log.

5.8 Partially offline transactions

Payment transactions can conceivably happen in scenarios where one of the parties is online, i.e., has a connection to the server, and the other is offline. In this case, there is a significant difference between whether it is the payer or the payee that is offline. In the case where the payer is offline, it is hard to gain any advantage from the fact that the payee is online. Even though the payee, in this case, could organize an online transaction with the server, a transaction from the payer would still require the payer to authenticate it. As the payer is not connected to the server, such an authentication would have to be relayed through the payee. It is conceivable that a protocol that securely allows for such a transaction can be constructed. However, due to the increased overhead and complexity of introducing another protocol, the simpler approach is to treat this as an offline transaction. The payer would then, as described in the previous sections, send an offline payment transaction to the payee, who would treat it like any other offline transaction upon receiving it. The payee can then immediately synchronize with the server.

In the other case, where it is the payee who is offline, it is easier to take advantage

of the pair's partial connection. The payer can, in this case, perform an online transaction and request a receipt from the server of said transaction. Presenting this receipt to the payee, given that the payee has means of verifying its origin and authenticity, should satisfy the payee that she has been paid.

5.9 Key takeaways

In this chapter, a best effort solution to offline functionality for a CBDC has been presented. The proposed protocol uses an account based architecture with funds separate from the main CBDC. Sub-protocols for deposits and withdraws have been formulated. A payment sub-protocol along with a host of security measures, primarily aimed at preventing double spending, are also explained. Secure hardware, server blacklisting, Distributed Attack Prevention (DAP) and Collaborative Attack Prevention (CAP) are mechanisms introduced to prevent abuse of the protocol. To apply these mechanisms, the protocol utilizes a transaction log scheme. The means for synchronizing and validating these logs have also been presented.

Chapter 6
Simulation Framework

To test the protocol design, it is necessary to be able to simulate the protocol to estimate the effectiveness of the security measures. In this chapter, the simulation framework design and implementation will be described. A simulation is a quantitative method applied where it is not possible to achieve an optimal solution. A model is a mathematical way of formalizing the structure and relationship of a real system [ASW+18]. The simulation is the execution of the model to achieve an as close as possible estimation of the real problem [ASW+18]. The simulation itself will be created to simulate the model of the system, as the simulation itself will provide results to check the effects of protocol design and architecture design choices.

A digital system is often best described by a discrete event simulation with discrete time and space, as the events can be described as logically separate processes in a discrete time [Fis01]. The simulation in a programming language is, in general, an abstraction of the real program, and by implication, it is often in some way a simplification of the real problem [Fis01]. The simplifications done for an efficient simulation will be explained in Section 6.1.6. When developing a simulation tool, there is a choice between using a general-purpose programming language or a special-purpose simulation language. In this thesis, the general-purpose language Python was selected.

To explain the simulation framework, first the simulation model is presented, then the node types are presented to introduce the roles in the model. The node types have states and events to transition between the states. The network to describe the context is presented along with the parameters that can be adjusted in the simulation framework. The threat actors in the system are described, and then the simplifications and abstractions of the model are explained. Lastly, the simulation experiments designed to test the architecture and protocol design are presented. The simulation framework in its entirety is available on GitHub [EJ22].

6.1 Simulation model

The simulation model is based on a network of nodes, defining each connection. This limits whom a node can communicate with and how the network coverage is defined. The network architecture discussed in Chapter 4 is based on a network of user nodes with an overlay of the connection to the processing nodes. In the model, edges are assumed never to fail, but the network nodes have up and down states. The simulation model used is based on a Poisson distributed generation of events.

6.1.1 Node types

For the natural point of sale simulation model, each node in the network has a role, i.e. a node type. The types used for this model are:

- User (user node)
- Fraudulent User (user node)
- Intermediary (processing node)
- Central Bank (processing node)

User The user performs transactions in the model. The user is offline if there is no connection to an intermediary. The user sending the funds is the one deciding if the transaction is online or offline. If the user is offline, the user node attempts offline payments. If the user is online and the receiver is online, an online transaction is attempted.

Fraudulent User The fraudulent user is a highly technically capable adversary. This user can send fraudulent transactions, manipulating both the software and the secure hardware. The fraudulent user can achieve double spending by replay attacks and selective rollback of secure hardware. The fraudulent user can not edit signed messages or successfully manipulate signatures.

Intermediary The intermediary provides access to online transactions if directly connected. An intermediary is connected to the central bank, and if both the intermediary and the central bank are online, an online transaction can be conducted. The intermediary is in the processing node group for potential events.

Central Bank The central bank processes the transactions, and no transactions can occur if this node is offline. The central bank writes completed transactions to a blockchain and checks that each user has sufficient funds at the point of transaction.

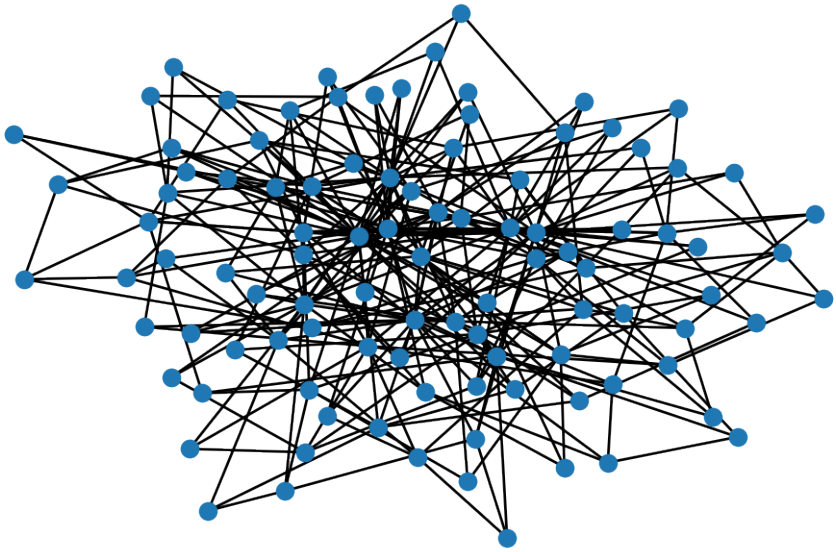


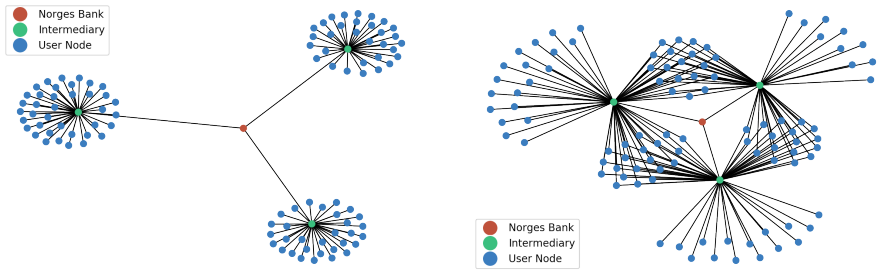
Figure 6.1: An example of interconnection of the user nodes. BA-graph with interconnection parameter of 3, 100 nodes.

6.1.2 Network

The network is a Barabàsi Albert preferential attachment model (BA). The amount of connections per node decides the different transaction partners the node can have during a simulation. The network does not append edges to the network in the simulation after creation. To simulate movement in time and space, the average amount of connections per node is set to a high number. This gives each node multiple connection partners, and the connected partners are likely to share connected nodes. The nodes in the BA graph with their connections are illustrated in Figure 6.1 with the connection preference of 3 and 50 nodes. It is observed some nodes are more interconnected than others, similar to how some nodes will be more active users of the system or conduct more transactions than others.

The network connectivity is represented by the intermediaries, with the condition to be connected to a minimum of one active intermediary to have network access. In Figure 6.2 the connection to the intermediaries is visualized with varying amounts of intermediary connections per node. With an increased amount of intermediaries per user, each user has higher connectivity to the central bank.

The central bank is connected to each of the intermediaries. Each intermediary represents a separate path to the central server, mitigating the offline situation to an



(a) Each node connected to one intermediary

(b) Each node connected to 1.5 intermediaries on average

Figure 6.2: Network with only User node to intermediary connections. Three intermediaries, 100 nodes, varying intermediaries per node.

online situation.

6.1.3 Parameters

The parameters used for the modelling and simulation of the desirable experiments are listed in Table 6.2 and Table 6.1. The parameters listed in Table 6.1 do not vary through the simulation experiments. The experiments are described in Section 6.2. The static parameters could vary for simulations with a different purpose than the described experiments, but for the purpose of the experiments here presented, with the limitations of the hardware, varying these parameters was considered out of scope and not directly necessary for the experiments. In this section, each of the parameters are explained in relation to the simulation framework.

Nodes The amount of nodes is set to 50. This was because of the computational complexity of increasing the number of nodes. An optimal amount of nodes could be a higher amount, but due to hardware limitations and time constraints of the project, the simulation run must be limited.

Intermediaries For the testing, the number of intermediaries should be equal to or higher than the number of intermediary connections per node. This ensures the existence of enough intermediaries for the minimum amount of intermediaries per node. The intermediaries are connected to the central bank, there is always one, and only one central bank.

Barababàsi Albert preferential attachment parameter The parameter defines the preferential attachment parameter in the graph, defining how likely an additional node is to connect to an existing node. In this model, the parameter defines the average amount of connections each node in the network will have, the node

degree between user nodes. As mentioned in Section 6.1.2 the parameter can be set high to simulate the potential of movement through time in the graph where each edge represents the potential connections a node has during the simulation. The parameter is set to 20 in this simulation to give a large potential for connections with few recurring transaction partners to increase the damage potential of the fraudulent users.

Fraudulent Node Percentage The fraudulent node percentage represents how many of the user nodes in the network that turns into fraudulent nodes. The parameter is set static to 0.1, 10% of the nodes. This is set artificially high compared to a realistic scenario to see comparable results to a large scale attack. The percentage in a real population is assumed to be very low, with a minuscule percentage of the population able to manipulate software and the protocol mechanisms. With secure hardware, this percentage of users able to manipulate the program is assumed to decrease even further. Using this variable, it is possible to implicitly test the effects of secure hardware, as the assumed result of a secure hardware implementation versus a non-secure hardware implementation would be a decrease in the fraudulent adversaries.

Transaction rate The transaction rate defines how often a transaction is conducted. With a transaction rate of 5, a transaction is executed every 5 ticks per node. With 50 nodes, an average of 10 transactions will be executed per tick. The transactions are generated through a Poisson distribution to simulate how transactions are sent with varying frequencies.

Transaction rate fraudulent user The transaction rate of fraudulent users is set to the same as the transaction rate of legitimate users. The intention of having a separate parameter is to test scenarios where an organized attack is performed by a sophisticated adversary. With a distributed short term attack, the transaction rate of fraudulent users could be much higher than the transaction rate of legitimate users to create the most damage before getting caught.

Transaction volume mean and std The mean transaction volume is the average transaction volume for legitimate clients in both online and offline scenarios. The exact amount is determined by a normal distribution with the transaction volume mean and transaction volume standard deviation as inputs.

Initial balance mean and std The mean and standard deviation of the initial balance is set to 10000 and 200, respectively. The value for each user node is determined by a normal distribution. The balance is first distributed to nodes by an initial deposit to the central bank.

Offline balance preference mean and std The offline balance preference is how much each node prefers to have in their offline account. After the initial deposit, each node transfers its preferred amount to its offline account. Only the balance transferred to the offline account with a deposit transaction before the client becomes offline can be used in the offline setting.

Offline balance preference fraudulent user mean and std This value corresponds to the amount each fraudulent user prefers to have in their offline wallet. This is set to a maximum amount value to ensure the fraudulent users have enough funds in their initial deposit to attempt a maximum amount of fraud in the offline setting.

Broadcast coverage The broadcast coverage parameter is used for the CAP extension as described in Section 5.6.4. The parameter defines how many of the node's neighbors are reachable for the broadcast of logs. This parameter is set to 0.2, 20%, as the preferential attachment parameter is set to 20. The 20% then corresponds to, on average 4 neighbors. The parameter would vary with the density of users in proximity of a transaction and the technology used for the broadcast.

Random seed The random seed for the simulation is used to determine the variation of a simulation run that should be used for the statistical models. This parameter allows for the recreation of the simulations, as each simulation can be recreated by using the same parameters again. The parameter is also used for variations of the same simulation with different choices in the statistically dependent procedures.

Offline amount limit The offline amount limit is the amount limit for offline transactions on a per-transaction basis. The limit does not prevent multiple consecutive transactions to the same person but enables the possibility of such a constraint.

Recovery rate of the central bank The recovery rate of the central bank is a parameter in an exponential distribution determining the ticks to recover. Each tick – a time element – in the simulation counts up and when it reaches the determined ticks to recover, the node transitions to an online state. With an average of 20 for the recovery rate, this would correspond to an average of 20 ticks before transitioning. With a transaction rate of 0.2, this would result in about 100 transactions conducted in an offline setting. Variations in this parameter could simulate short and long offline settings.

Failure rate of the central bank The failure rate of the central bank is how often the users in the simulation become completely offline.

Name	Description	Value
User Nodes	Amount of user nodes in the graph	50
Intermediaries	Amount of intermediary nodes in the graph	10
Barabàsi Albert preferential attachment parameter	The BA attachment parameter	20
Fraudulent Node Percentage	Percentage from 0-1 of the user nodes that becomes fraudulent nodes	0.1
Transaction rate	The frequency of transactions, input in a Poisson distribution for generation of events	5
Transaction rate fraudulent user	The frequency of transactions for fraudulent users, input in a Poisson distribution for generation of events	5
Transaction volume mean	Average amount per transaction	100
Transaction volume std	Standard deviation in the amount per transaction	40
Initial balance mean	Initial balance of each user node	10000
Initial balance std	Standard deviation of the initial balance of the user nodes	2000
Offline balance preference mean	Mean preferred offline balance for user node	1500
Offline balance preference std	Standard deviation of the offline balance preferred by the user node	500
Offline balance preference fraudulent user mean	The mean balance preferred by the fraudulent users	1 000 000
Offline balance preference fraudulent user std	The standard deviation of the balance preferred by the fraudulent users	0
Broadcast coverage	Percentage of neighbors, a broadcast of logs can cover	0.2

Table 6.1: Static Simulation Parameters

Recovery rate of the intermediary The recovery rate of the intermediary defines the exponential parameter, how many ticks to an active state from a failure state.

Failure rate of the intermediary The failure rate of the intermediary defines how often each intermediary transitions to a failure state.

Server Blacklist This parameter enables or disables the server created blacklist of users not to do transactions with. The feature is discussed in Section 5.6.3.

DAP This parameter enables the DAP security measure as described in Section 5.6.2.

CAP The parameter enables the CAP security measure as described in Section 5.6.4.

6.1.4 Events

The simulation model is visualized in a simplified activity diagram in Figure 6.3. In the diagram, the event generation is simplified to user nodes and processing nodes. In the simulation, the events are generated in a loop, with the number of events generated dependent on the random seed. Each run in the simulation is based on a number of ticks, the time unit of the simulation. For each tick, the user nodes can execute a transaction with the connection probability, and the processing nodes can become online or offline, with their corresponding distributions depending on the simulation parameters.

Transaction execution The user nodes, including users and fraudulent users, have one activity, performing a transaction. The fraudulent users only perform offline transactions, while regular users check with their connected intermediaries if at least one is online. If at least one intermediary is online and the central bank is online, the user is able to perform an online transaction and send the transaction to the payee. The transactions are generated in a Poisson distributed frequency. The user nodes perform payment transactions in accordance with the protocol description in Section 5.5. If the payer node is online, the transaction is attempted as an online transaction, while if the payer node is offline, the transaction is attempted as an offline transaction. For this reason, the payer is always the one deciding what type of transaction is attempted, allowing the fraudulent users to attempt an offline transaction every time. If the payee is online while the transaction is attempted offline, the payee is able to check the balance of the payer with the intermediary before the payment is attempted. With the assumption that both participants in the transaction should have the same connectivity, and therefore the online balance in the offline account should be up-to-date, the payee can reject the payment if

Name	Description	Value
Random seed	The random seed of each simulation	1 - 42
Transactions per node	Average amount of transactions per node in a simulation. Defines the length of the simulation	1 - ∞
Offline amount limit	Max amount spent per offline transaction	1 - ∞
Recovery rate of central bank	Parameter in an exponential distribution, determines the amount of ticks before the central bank is back in an online state a failure	0 – 200
Failure rate of central bank	Parameter in a Poisson distribution, determines how often the central bank transitions to an offline state	0 – 200
Recovery rate of the intermediary	Parameter in an exponential distribution, determines the amount of ticks before the intermediary is back in an online state after a failure	0 – 200
Failure rate of the intermediary	Parameter in a Poisson distribution, determines how often the intermediary transitions to an offline state	0 – 200
Intermediary connections per node	User node redundancies	1 – 20
Server Blacklist	Determines if the server blacklist prevention feature is active	True/ False
DAP	Determines if the DAP prevention feature is active	True/ False
CAP	Determines if the CAP prevention feature is active	True/ False

Table 6.2: Simulation Testing Parameters

the payee does not have sufficient funds in the online mirror of the offline balance. This approach is pessimistic and will potentially stop legit payments, but it is an assumption of this implementation of the protocol.

Processing node failure and recovery events The processing nodes are both the intermediaries who have direct access to the central bank and the central bank itself. A failure in an intermediary leads to a service failure for connected nodes if there are no other intermediaries they can connect to. The failures occur in a Poisson distributed event generation, and when the failure has occurred, the nodes recover in an exponentially distributed amount of ticks. If there is a failure in the central bank, this leads to an immediate service failure of all nodes in the network,

and every transaction after this will be an offline transaction. This case is used to simulate long and short offline contexts, while selective failure in intermediary nodes simulates partial service failures.

6.1.5 Threat actors

The fraudulent users are the main threat in the model. The fraudulent users will always try to spend their maximum limit of funds. If there is no transaction limit in place, this would mean the users would spend all of their balance in the first transaction. As the users try to spend the most amount of value before being detected by the system, they try to maximize their spending in the shortest amount of time. To achieve this, their offline preference is set to higher than their maximum balance. Thus they will always have all their money in their offline account. To escape most of the security measures put in place by the protocol and mitigate their personal risk of being detected without making a profit, the users act like legitimate users until they are out of funds. When the fraudulent users are out of funds, they reset their offline balance, clear their transaction log of every transaction except the deposit, and reset their previous hash to the initial hash. All of these measures combined are a highly advanced method and are deemed to be the most efficient way of achieving fraud in the protocol. Simpler attacks do exist, but for the purpose of this simulation framework, the most advanced adversaries could create the most amount of illegitimate spending with the corresponding countermeasures. Without the DAP and CAP mechanisms, very simple attacks could do increasing damage, as it would not be necessary to initially have the balance to send the amount.

6.1.6 Simplifications and abstractions

To create a simulation framework, the model of a real system has been abstracted and simplified, as a real system is close to impossible to simulate without such simplifications. The computational complexity is an important limiting factor of the simulation, as this limits how many runs can be done in any amount of time. The implemented features in the framework should all be directly relevant for the testing purpose, as the implementation itself is not the point of a simulation. The desired experiments are listed in Section 6.2, and the simulation framework is designed to test the experiments with an implementation of the protocol defined in Chapter 5.

Certificate implementation The certificate structure, creation, distribution, and signatures of messages are not implemented in the framework. This is because, for the testing of the security measures, this was not deemed as necessary as the concept and methods have a large consensus among researchers to provide authenticity and integrity when used for digital signatures. The process of creating keys and encrypting/ decrypting with RSA/DSA is a relatively quick process. However, the

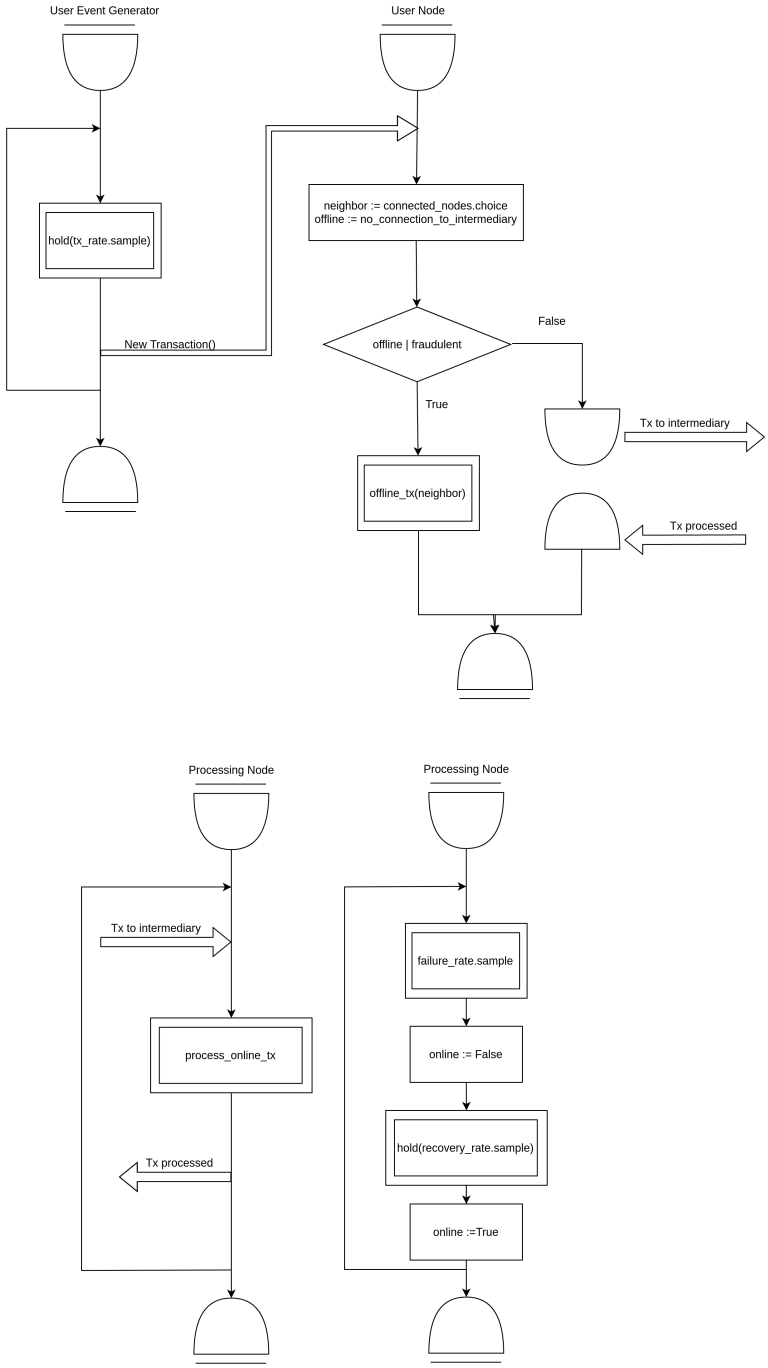


Figure 6.3: Activity Diagram for the main events in the simulation framework

implementations would provide few, if any, advantages to the model and additional overhead for each run.

Rehashing of transactions With a similar argument as the certificate and signature checks, the transactions in the log are not rehashed by each client when verifying the authenticity but are simply checked that the expected hash is equal to the hash stored in the transaction. As we do not assume any node can successfully compute a fraudulent hash or hashes, a re-computation would simply be extra overhead.

Reset of logs The reset of logs after a resynchronization with the server, as outlined in Section 5.7 is not implemented in the framework. It was concluded that the simulation framework would disregard any memory limitations of each client and, thus, the reset of logs was not needed for the simulation purposes.

Continuous network restructuring To simulate movement in a network, each node could be able to move around and create new edges in the network. For this model, it was chosen to increase the number of edges in the generation of the network and then disregard most of the neighbors during parts of the simulation.

Malicious adversary variations To simulate a complex threat model, different malicious actors could be implemented. Each actor could have a different technical capability and thereby a different manipulation degree of the software. This could, in the case of no-preventions, boost the total amount for the adversaries, as they would not have to follow any rules to extract the most amount of value from the system. Furthermore, the variants would work to a varying degree against each of the security measures. However, in this framework, only one actor was created, optimized for the attack against the DAP preventions.

Secure hardware Secure hardware was not used in the framework, as this requires extra effort to create programs and requires specialized hardware and specialized programming languages. To simulate the hardware, it would be sufficient to decrease the number of fraudulent nodes, as the number of persons with the knowledge to operate and manipulate such hardware would be decreased.

Transmission Layer In the framework, we assume all communication is secure and reliable. Transmission Control Protocol (TCP) with Transport Layer Security (TLS) is a proven communication transport protocol for secure and reliable communication and could be implemented. As the design of the protocol does not limit the possibilities for the underlying technology, the simulation also has no limitations or implementation of such.

Routers and network connectivity All routers and network connectivity in the framework are simplified down to a few state machines. This is due to the fact that software components in simulations can be accumulated to the collective availability of the combined components, and nodes that do not append additional simulation capabilities to the framework are unnecessary.

Space complexity The space complexity on the client level was explicitly and intentionally not a priority, as the primary focus was computational complexity and the simplicity of implementation. As it is not assumed that the transaction log is stored in the secure hardware part of the software, it is assumed there is sufficient storage for a log with exponential growth.

6.1.7 Assumptions and biases

To expand on the assumptions already covered and those evident from the practical implementation of the simulation, we will, in this section, expand on some key assumptions and biases in the presented simulation model.

The threat model implemented assumes that double spending is the main potential risk posed by the protocol. It does not account for other potential attacks and abuses that may be possible given the implementation. Any results might therefore fail to account for the complete attack surface exposed by the offline protocol. Assumptions regarding the technical capacities of users, specifically with regard to logs, may also affect the general security achieved in the simulation result. Attacks targeting these logs, which will be discussed in Section 7.2.4, might significantly affect the overall risk assessment of the protocol, but they are assumed not to be attempted in the simulation.

6.2 Simulation Experiments

In this section, the simulation experiments will be presented. The experiments represent the simulation runs executed on the simulation framework, and the results will be presented in Chapter 7. The purpose of the experiments and the parameters used will be presented.

6.2.1 Intermediary density

The effect of the intermediary density will be tested in this simulation experiment. The goal of the experiment is to analyze the impact on rates of offline transactions for varying connection redundancies. The degree of redundancy will affect the steady state availability of the system and, consequently, the time the system is in an offline state. The purpose of the experiment is to test the availability equation derived in

Section 4.5 for such a model and how well the theory applies to the architectural model in the simulation.

The simulation model operates with an input of average intermediary connections per node that can be varied. The number of connections for each client is the same as this number for all integer inputs but will vary between floor and ceiling values with any decimal inputs. In the simulation framework, each client conducts transactions after a Poisson process. If the sending client, the payer, is offline, the transaction is counted as an offline transaction. Otherwise, the transaction is considered an ordinary online transaction and not counted in the results presented. Recall that transactions with an online payer and an offline payee are trivial to solve, as discussed in Section 5.8.

To analyze the effect of intermediary density, i.e., how many intermediaries are available per client, the simulation was run 11 times for each intermediary density, with varying seeds as input for the random generator. The intermediary densities used in the simulation are integers 1 – 10, as well as 1.5.

The values used in the simulation for network failure rate and recovery rate are 20 and 10, respectively. These inputs mean that each intermediary will fail on average once every 20 ticks and recover with an average delay of 10 ticks. Similarly, for the central node, the failure rate is 100, and the recovery rate is 5. These values are selected to enable the simulation to simulate multiple offline scenarios quickly. However, they are far higher (compared to the rate of transactions) than what we could realistically expect for a network such as the Norwegian core internet. Therefore, any results of this simulation are exaggerated compared to any realistic scenario but should demonstrate principles that would hold for more realistic networks. The complete set of testing parameters used for this experiment can be found in Table 6.3.

6.2.2 Protocol security mechanisms

To test the effectiveness of each protocol security enhancement proposed in Chapter 5, a series of simulations are to be run. The main goal of the experiment is to test whether each mechanism provides the intended prevention against fraudulent transactions and attempt to quantify the effectiveness of each measure. Thus a realistic analysis of the risk posed by offline functionality can be carried out. A variety of scenarios will be simulated to test differing conditions and offline scenarios. Simulations will run with four different levels of prevention mechanisms enabled. The first is the no prevention scenario. In this case, no preventions are enabled, except that an online node checks the offline balance of the payer node against the intermediary. The second level of prevention enables server blacklisting as outlined in Section 5.6.3. The third level of prevention enables only DAP as outlined in Section 5.6.2. The fourth

Name	Values
Transactions per node	200
Offline amount limit	1000
Recovery rate of central bank	5
Failure rate of central bank	100
Failure rate of central bank	20
Recovery rate of intermediary	10
Server Blacklist	False
DAP	False
CAP	False
Random seed	1, 2, 3, 42, 5, 6, 7, 8, 9, 10
Intermediary connections per node	1, 1.5, 2, 3, 4, 5, 6, 7, 8, 9, 10

Table 6.3: Intermediary density experiment variables

and final level enables DAP and CAP as outlined in Section 5.6.4. The mechanisms are tested individually to evaluate their specific contribution to the security of the system, except for CAP, which will have close to no effect without also including DAP. The server blacklist could also be included in the fourth run, as the features are capable of complementing each other but is left out in this experiment.

In addition to the four levels of prevention, the recovery rate of the central system is varied so that different types of offline scenarios are simulated. Increasing the recovery rate of the central system increases the longevity of offline periods. Simulations were run with recovery rates of 10, 20, 50, and 100, with the length of each simulation being ten times the recovery rate worth of ticks. The length of the simulation is set through the transactions per node parameter and is set to two times the recovery rate of the central bank. Each of the simulated scenarios was run three times with varying seeds for the random generator. A complete set of the parameters varied in this experiment is displayed in Table 6.4.

6.2.3 Transaction limits

To test the effects of transaction limits, the maximum amount to be sent in each transaction is varied in these simulations. A set of simulations will be run for differing security measures. Each simulation has 50 clients, with an average of 10% being fraudulent. Each client starts with an average of 10 000 funds. With a fraudulent user spending their entire balance in each transaction, limited by the transaction limit for each transaction. In this experiment, the offline amount limit varies from as

Name	Value
Offline amount limit	1000
Failure rate of central bank	4
Recovery rate of the intermediary	1
Failure rate of the intermediary	100
Intermediary connections per node	1.5
Random seed	1, 2, 3, 42, 5, 6, 7, 8, 9, 10
Transactions per node	20, 40, 100, 200
Recovery rate of central bank	10, 20, 50, 100
Server Blacklist	True/ False
DAP	True/ False
CAP	True/ False

Table 6.4: Protocol security mechanisms experiment variables

low as 10 up to twice the average initial balance. In this experiment, it is expected that the amount of fraud will be limited with a smaller offline amount limit per transaction, and the effect will decrease when we surpass the node's balance. The initial node balance in this experiment implicitly represents a deposit limit for the offline account. Without the deposit limit, the effect of the transaction limit would be expected to be linear with the amount of successful fraud volume. By varying the security measures in this run, we will see the effects of each of the security measures on lower and higher transaction limits and how well a combination of both fraud mitigation techniques and the transaction limit, limiting the consequence of the fraud, works together. The exact variation in parameters of each experiment run can be seen in Table 6.5.

6.3 Key takeaways

In this chapter, the simulation framework has been presented in detail. The simulated network, types of nodes, and events are explained. The simulation framework implements architectural considerations discussed in Chapter 4 and the protocol proposed in Chapter 5. Simulation parameters are explained. Key model simplifications and abstractions are covered, and the threat actors are introduced in the simulation model. Lastly, the relevant experiments designed to test the effect of intermediary densities, protocol security measures, and transaction limits are outlined.

Name	Value
Transactions per node	50
Recovery rate of central bank	20
Failure rate of central bank	4
Recovery rate of the intermediary	5
Failure rate of the intermediary	4
Intermediary connections per node	1.5
Random seed	1, 2, 3
Offline amount limit	10, 100, 1000, 5000, 7500, 10000, 12500, 15000, 17500, 20000
Server Blacklist	True/ False
DAP	True/ False
CAP	True/ False

Table 6.5: Transaction limits experiment variables

Chapter 7
Results and discussion

This chapter contains the results of the simulation experiments and a discussion of the results obtained through the simulations. The chapter further discusses the implications of the results and how well the protocol fulfills the requirements from Chapter 3. Risks introduced by the protocol, the practicality of implementation, and a discussion of alternative solutions follow to give a clear evaluation of the aspects of the protocol. A simulation evaluation then follows to discuss the validity of the results. Privacy is then discussed. The chapter is concluded with a discussion of further research in the field and a conclusion on the project.

7.1 Simulation Results

In this section, the results of the simulation experiments from Section 6.2 will be presented and discussed.

7.1.1 Intermediary density

The parameters used in this experiment is presented in Section 6.2.1. As discussed in Chapter 4, the architecture design choices play a crucial part in reducing the risks associated with offline usage of a CBDC. In this section, simulation results regarding intermediary density and its effect on system availability will be analyzed. Based on the simulation model and the network characteristics presented in Section 4.5, we have expected values for the uptime, i.e., time spent with connection to at least one intermediary, of each node. Using the formula for availability in the designed architecture, Equation 4.5, we can calculate the expected uptime for each node in the simulation for differing numbers of intermediary connections. The results of this can be seen in Table 7.1.

The results of the experiment can be seen in Figure 7.1. The plot displays the maximum and minimum observation for each intermediary density, as well as the second and third quartile and the median value. The dotted line shows the expected

values as calculated using the Equation 4.4. The average for all the simulation runs for all numbers of intermediaries is displayed in Table 7.1.

The intermediary densities used in the simulation are selected to study the reduction in offline transaction rate by increasing the number of connections per client. As we see, inputs of increasing numbers of intermediaries per client quickly trend towards an online transaction rate of 1. For this reason, no inputs higher than 10 were included, as we can confidently assume this trend continues. The intermediaries per node value of 1.5 is included to study the gradual change at lower values, as this is where the gain of adding more intermediaries per client is the largest.

The rate with which the online transaction rate trends towards 1 with increasing connections per client is determined by the intermediary failure and recovery rates. We recall from Equation 4.4 that the availability, i.e., the time spent with at least one intermediary connection, of each client is dependent on the number of connections per client, the failure rate of the intermediaries, and the recovery rate of the intermediaries. This equation is plotted in Figure 7.2 for varying levels of intermediary failure and recovery rates. As we see, the trend toward 1 happens more rapidly with decreases in the failure rate (i.e., higher λ) and reduced recovery rate. Intuitively, the more failure prone the network is, the more intermediaries per client are necessary to achieve high levels of availability.

From Figure 7.1 we see that the results of the simulation run largely conform with the expected rate of online transactions. There are some variations in the results, and thus some deviations from the expected values, but this is expected given the relatively short runs, 1000 ticks, and the small sample size, ten runs total. Despite the results only accounting for network availability for a given set of failure and recovery rates, they show that the model presented in Section 4.5 holds for the network simulated. Increasing each client's access to the CBDC system will decrease the frequency of offline transactions. Thus, the CBDCs architecture, and the infrastructure with which that architecture is provided, can reduce the risks associated with offline use by reducing the frequency and longevity of offline scenarios.

7.1.2 Protocol security mechanisms

The parameters of this experiment are described in Section 6.2.2. The results of all protocol simulation runs can be seen in Figure 7.3 and Figure 7.4. These plots both show the same data, but the former compares prevention mechanisms with each other for each level of network recovery, while the latter displays each prevention mechanism's effectiveness for varying longevities of offline scenarios. The figures display the fraud success rate, i.e., what fraction of attempted fraudulent transactions were successful, for each simulation.

Average intermediaries per client	Expected uptime	Simulated online transaction rate
1	0.63492	0.63379
1.5	0.76909	0.75247
2	0.84656	0.85626
3	0.91711	0.92247
4	0.94062	0.95127
5	0.94846	0.94951
6	0.95108	0.95905
7	0.95195	0.96799
8	0.95224	0.96366
9	0.95233	0.94406
10	0.95237	0.96398

Table 7.1: True and expected availability for intermediary densities

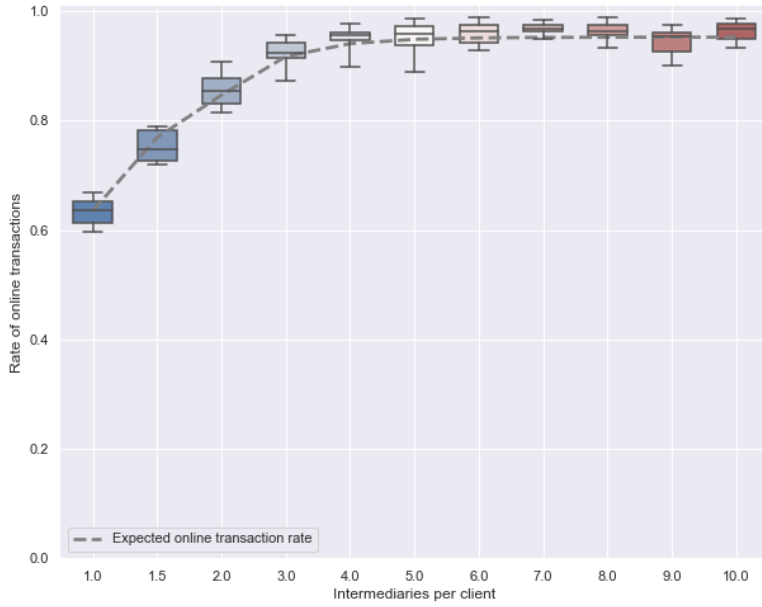


Figure 7.1: Simulated rate of transactions processed online against intermediary connections per client

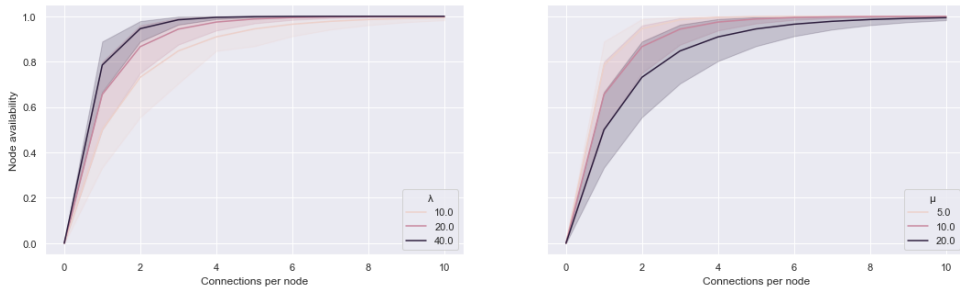


Figure 7.2: Expected rate of transactions processed online against intermediary connections per client for changes in intermediary failure rate (λ) and recovery rate (μ) based on Equation 4.4

From Figure 7.3 we see primarily that increasing the level of prevention complexity reduces the success of fraudulent clients. Server blacklisting alone provides a significant improvement over the no prevention scenario. The DAP provides yet another leap in the security of the system. The difference between DAP and CAP, which also has DAP enabled, is less pronounced, but we see a slight improvement (6.1% against 5.3% averaged for all runs, respectively). These results suggest that the various prevention mechanisms proposed in the protocol, on the whole, have the desired effect.

The effectiveness of the differing prevention levels is relatively consistent across the different intermediary recovery rates. In Figure 7.4 we see an increase in fraudulent success for higher intermediary recovery rates for the no prevention scenario. As there are no preventions enabled in this scenario, this suggests that more fraudulent transactions will happen in the simulation runs with higher recovery rates. This can partly be explained by the fact that in those scenarios, the rate of offline transactions compared to online transactions is higher.

The fraud success rate with server blacklisting is decreasing with increased intermediary recovery rates. Once again, this can partly be attributed to the fact that longer intermediary recovery rates give longer simulations, providing more time for the prevention mechanism to work. This might be counterintuitive in the sense that we could expect server blacklisting to be most useful in scenarios where there are many shorter offline periods, giving the server a more up-to-date image of the network than with longer intermediary recovery times. Given the workings of the simulation and the parameters used, however, the server blacklisting seems to gain effectiveness in the runs with longer recovery rates.

The DAP and CAP scenarios, on the whole, provide much better protection against fraudulent transactions than server blacklisting. CAP is also, as mentioned,

Prevention	Mean fraud success rate	Reduction from no prevention (%)
No prevention	0.8901	-
Server blacklist	0.3092	65.3
DAP	0.0607	93.2
CAP	0.0529	94.1

Table 7.2: Mean fraud success rate and percent-wise reduction compared to no prevention

slightly more effective and slightly more consistent in its results. These results show that the DAP and CAP prevention mechanisms, even when fraudulent users have bypassed the secure hardware protections, significantly reduce the success rate of attacks in the system. A summary of the results of the simulations can be seen in Table 7.2. Here we see the mean fraud success rates for the different preventions and their percent-wise reduction compared to the no prevention scenario. Based on its characteristics and input parameters, the simulation suggests that DAP and CAP can reduce the success of fraudulent actors by 93.2% and 94.1%, respectively.

This relatively small difference between DAP and CAP indicates that the broadcast mechanism may not be worth the increased communication and computation overhead it puts on the protocol. It is worth noting, however, that, due partly to limitations of the simulation framework explained in Section 6.1.6, as well as the relatively narrow scenarios and parameters tested for, the simulation does not cover the full range of conceivable offline scenarios. The CAP is explicitly designed to amplify the effect of the DAP in long-lasting offline scenarios with a high degree of mobility for clients. Such scenarios are not extensively tested in these simulation runs. The lack of synchronization and, therefore, the resetting of logs may also contribute to the lack of difference between DAP and CAP in the results.

7.1.3 Transaction limits

A commonly discussed measure to limit the risk associated with potential offline solutions is transaction limits. Different kinds of limits are proposed, namely limits on volume per transaction and limits on the number of transactions per client. Some central banks also note that different types of accounts and wallets may require different transaction limits. Transaction limits should be imposed in such a way that, even if a fraudulent user were to bypass all other security measures, no legitimate users would accept transactions breaching that limit. It is also worth noting that a limit only on the volume of each transaction can be bypassed by performing multiple

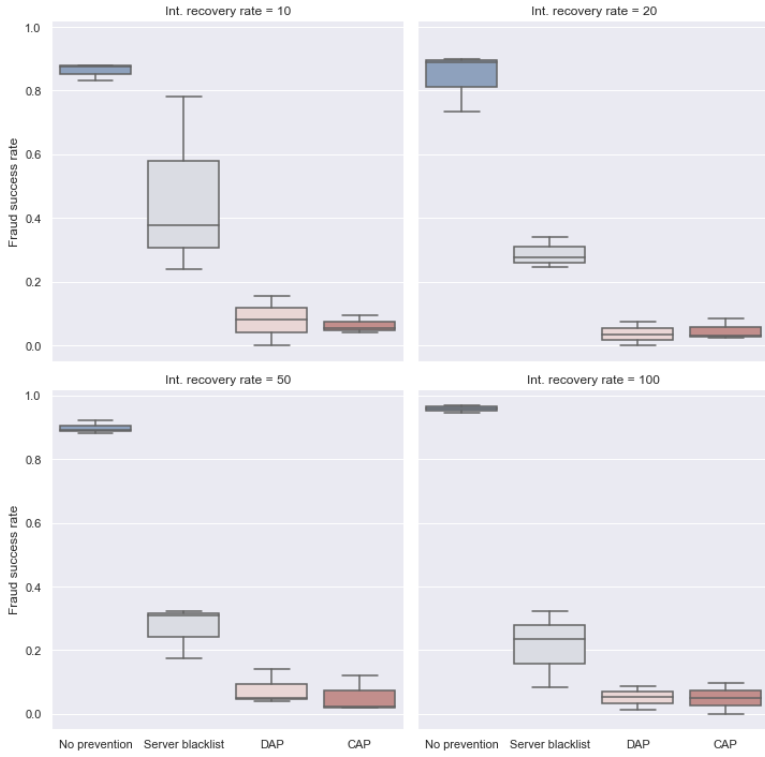


Figure 7.3: Success rate of fraudulent clients for differing security mechanisms

smaller transactions, and any real world implementation should therefore include a combination of the variants.

The expected effect of transaction limits is relatively intuitive. If implemented correctly, we can expect such limits to linearly reduce the volume of illegitimate funds fraudulent users can spend. However, certain attacks where conspiring fraudulent users bypass the transaction limit to produce funds can be conceived. In such cases, the system will rely on other security measures preventing them from spending these funds in transactions with legitimate users. Specifically, in the context of the protocol presented in this thesis, the DAP and CAP mechanisms should protect legitimate users from accepting funds generated in this manner.

Exact levels of transaction limits and how they are imposed on different types of users are, in addition to being a technical security consideration, a discussion of system characteristics, user experience, and policy. Therefore, determining exact limits should be left to more holistic analyses in the field. In this study, a range of limits on volume per transaction is simulated to demonstrate the effect of transaction limits.

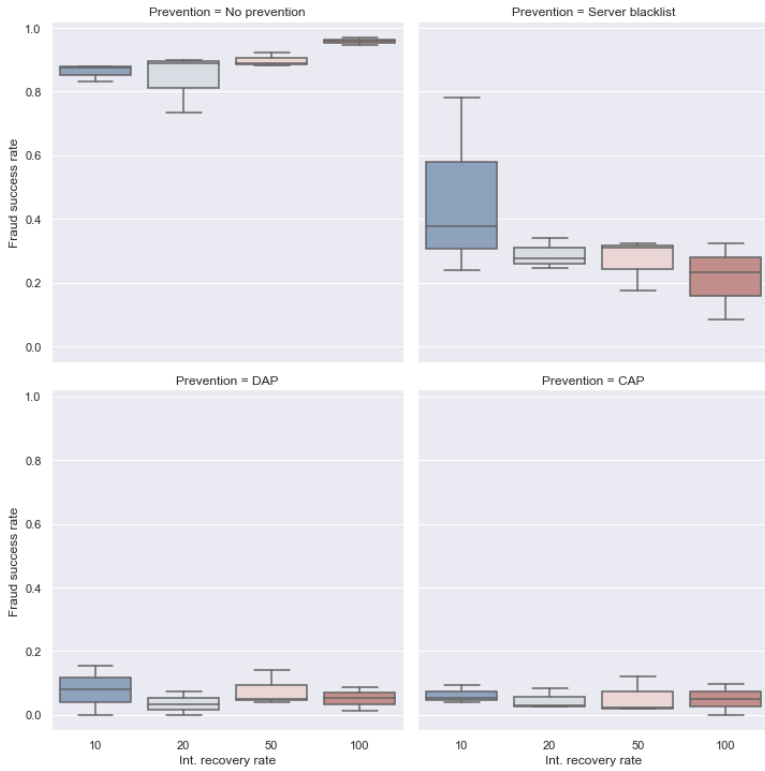


Figure 7.4: Success rate of fraudulent clients for differing intermediary recovery rates

The simulation is run with and without other security measures. The simulation experiment and the specific testing parameters are presented in Section 6.2.3.

The results of the simulation of transaction limits can be seen in Figure 7.5. Fraudulent volume sent, i.e., the total volume of illegitimate funds sent and received in the simulation, is plotted on the vertical axis. Varying levels of per transaction volume limits are plotted descendingly on the horizontal axis.

From the results, we see that the transaction limits have an effect on the volume of fraudulent funds sent. This is most apparent in the simulation runs without any other security measures. We see the most dramatic effect of transaction limits when the limit is lower than the average initial funds given to each client. This is to be expected as, to double spend, fraudulent users still need to send a log with the initial deposit transaction from the server. This deposit, and other transactions received, imposes an upper limit on the volume of transactions a fraudulent user can get away with. This also demonstrates, more generally, that transaction limits much larger

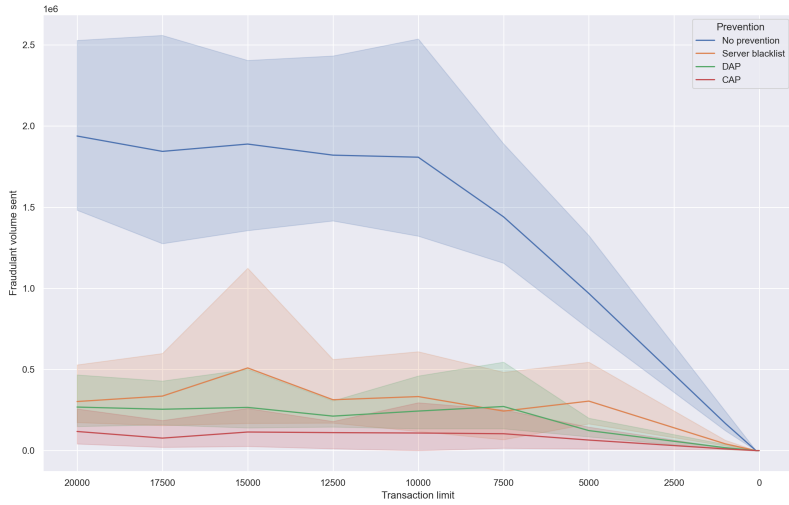


Figure 7.5: Volume sent through fraudulent transactions for varying transaction limits

than the expected volume for each transaction will have a smaller effect on preventing fraud.

Generally, transaction limits are less effective when other security measures are included, as these will prevent the majority of fraud. The transaction limit may still reduce the fraudulent volume sent if sufficiently low to reduce the overall transaction volume in the simulation. This is, however, a slightly moot point, as any such overly low transaction limit would severely decrease offline functionality and usability of the CBDC.

7.2 Protocol evaluation

In addition to testing the proposed protocol through simulation, we will, in this section, evaluate and discuss the protocol in light of the requirements gathered in Chapter 3. Points of interest related to the protocol, such as its practicality of implementation and alternative solutions, will be briefly discussed for a complete evaluation of the protocol.

7.2.1 Requirement satisfaction

With regard to the business requirements outlined in Section 3.3.2, the offline protocol can aid Norges Bank in achieving all three. Specifically, the protocol can increase the security and, consequently, the confidence in a CBDC. The suggested protocol also provides a contingency solution for failures in the banking system that would result in offline scenarios for the users of a retail CBDC.

The functional requirements, outlined in Section 3.3.3, are partially covered. FR1, FR4, FR6, and FR7 are considered to be completely satisfied as they are design features of the protocol. FR2 and FR3 require that no form of forging or double spending attacks can be performed. This is, under the proposed protocol, not guaranteed. As we have seen, none of the reviewed approaches to the offline problem can make such a guarantee without compromising on other requirements and features. Specifically, we recognize that, in order to completely prevent forging and double spending, finality and reusability in an offline setting is technologically infeasible. FR2 and FR3 are, therefore, given the current technical limitations, contradicting FR4 and FR5. Which of these requirements are satisfied will therefore be a matter of priority and a design choice.

FR5 is partially satisfied. The protocol holds that a transaction conducted in good faith by a non-cheating party will be final. However, this is at the expense of FR8, as one cannot allow finality without accepting that illegitimately created funds will propagate the system and increase the total supply of funds in the system. The security measures discussed throughout this thesis aim to reduce both the risk of, as well as the potential scale of, such illegal creation of funds. However, as long as the double spending problem for an offline setting cannot be completely solved, one cannot have both finality of transactions and consistency of value in the system.

The protocol has some prerequisites that contradict the availability requirement. Especially, the need for secure hardware degrades the availability of the suggested protocol. Beyond this limitation, the protocol increases the availability of a potential retail CBDC by making transactions possible in a range of offline scenarios.

The protocol opposes the confidentiality requirement by making offline transactions available to other observers. More on this issue, along with a wider discussion of privacy considerations, can be found in Section 7.4. The integrity requirement is partially maintained but voided in the case of illegitimate transactions to reduce the potential creation of illegitimate funds.

The protocol largely meets the interoperability requirement in the sense that it is more or less agnostic to the wider CBDC system and can, with minor modifications, be applied to other CBDC solutions than the one here envisioned. Traceability is

provided by the protocol and a necessity for some of its security mechanisms. The protocol is in line with the requirements of fault tolerance and graceful degeneration, as it provides offline functionality and maintains operability even when abused by a malicious actor.

The modularity requirement is difficult to evaluate without a wider understanding of the CBDC system. It is conceivable that offline functionality, as here presented, can be applied in a modular way to the wider system. The system largely meets the reliable requirement. There are potential issues with the scalability of some solutions in the protocol, namely the growth of logs in lasting offline scenarios, which will be further discussed in Section 7.2.3. Although tested through simulation, there remain uncertainties regarding the protocol's reliability in untested conditions. The privacy requirement is another requirement that is hard to evaluate the protocol against specifically. This point is further discussed in Section 7.4.

With regard to the fair exchange problem, we know from Section 2.3.2 that providing such fairness in an offline setting is impossible. However, by providing finality of transactions for non-cheating actors, a best effort approach to the problem is implemented. This ensures that even if the payer has previously forged or double spent funds involved in a transaction, the payee remains compensated. This satisfies the effectiveness and timeless properties of a fair exchange for the non-cheating party.

The lasting security requirement is not met with regard to supply conservation, as discussed above. The protocol does, however, provide a functional alternative for lasting offline scenarios.

7.2.2 Level of risk

Through the simulation results presented, along with further evaluation of the protocol, we attempt to analyze the risks associated with offline functionality. Risks, in this context, refers to the risk of abuse and misuse of the protocol resulting from its technical limitations, causing financial harm to the system and its stakeholders, reducing trust and confidence in the system, reducing central bank control of the money supply, or having other undesirable consequences. Analyzing such risks with any level of precision is notoriously difficult given the large complexity of the system, the environment in which it runs, and the threat landscape it is exposed to. Any number of factors included or excluded from the simulation model can have large effects on the validity of results. Analyzing the risks associated with offline functionality of a retail CBDC through such means will give, at best, weak estimates.

Despite this, the results presented above suggest that the mechanisms proposed to prevent abuse of the system largely have the desired effect. The results from the intermediary density experiments support the idea that increasing the availability of

the CBDC (online) system through architecture design choices and infrastructure will reduce the threat posed by offline vulnerabilities. Relatively moderate levels of redundancy can significantly reduce the frequency and longevity of offline periods. Such a reduction in the probability of occurrence will reduce the risk associated with offline functionality as any offline vulnerabilities are less exposed. Therefore, decreasing offline usage is an effective first line of defense against the risks posed by offline functionality.

However, as long as one cannot unconditionally guarantee a connection to the system, additional security measures are needed to provide offline functionality at an acceptable level of risk. The experiments testing transaction limits suggest that such limitations on the use of the system can curb the consequences of successful abuse of the system. As discussed, such limitations increase in effectiveness with the restrictiveness of which they are implemented. The reduction in risk provided by such a measure is, therefore, dependent on what level of restriction is deemed acceptable to place on users of the system.

The experiments testing the specific prevention mechanisms of the protocol suggest that they are effective at reducing the probability of fraud attempts being successful and, therefore, the overall risk associated with the system. Simulation runs suggest that DAP and CAP may prevent more than 90% of attempts at double spending, but this estimate is, as mentioned, very much subject to model parameters and assumptions. More realistic scenarios with sophisticated attackers familiar with the limitations of the prevention mechanisms will, in all likelihood, be far more successful in their attempts to bypass the system's security measures. The simulation experiments make no effort to analyze the effects of secure hardware on the security of the protocol. However, we know from existing literature that such systems can, and have, been penetrated.

With such an unclear understanding of the risks involved in an offline solution that provides finality of transaction and offline reusability of funds, further research and particularly more extensive testing of existing solutions is needed before any accurate estimate of the potential costs of such solutions can be determined.

7.2.3 Practicality of implementation

An important aspect of the protocol design is its potential to be practically implemented. A CBDC should ideally be available to a large part of the population and, as far as possible, not require unusual or specialized hardware or tools. As we have seen, however, achieving satisfactory security for offline transactions is very difficult without relying on secure hardware. This secure hardware nevertheless dramatically increases the difficulty of adopting and using the system, as it may require acquiring new devices. As we have seen, many device manufacturers are already implementing

TEEs in their devices, and the accessibility of secure hardware may therefore be a manageable concern.

Another potential issue with any implementation of the protocol presented in this thesis is the storage and exchange of logs. Although logs may be reset upon synchronization with the server, long-lasting, or even shorter offline periods with high transaction frequencies, may quickly produce large logs for clients to maintain. The sharing of logs, particularly when logs are being broadcasted, may quickly exhaust client storage capacities. This issue is further complicated by the fact that logs cannot easily be deleted as new ones are received. For this reason, contingency solutions like pseudo-intermediaries that allow client synchronization in lasting offline scenarios may be necessary.

7.2.4 Risks introduced by the protocol

Although the aim of the protocol is to reduce the risks associated with offline transactions, it is conceivable that certain aspects of the protocol may introduce risks to the system. As discussed, the protocol does not entirely prevent double spending and forging but takes various steps to reduce the occurrence rate and effectiveness of such attacks. If the security and prevention mechanisms prove less effective than here assumed, they can therefore provide a false sense of security. Furthermore, scenarios not simulated and attacks not envisioned may prove to allow fraudulent users to create and use large sums of illegitimate funds. In this case, the central bank could lose control over the money supply, and the high volume of fraudulent funds in the system would probably severely reduce the confidence and trust in the system and subsequently the currency.

Another potential issue with the proposed protocol is potential tampering with logs. As clients exchange logs of other clients' transactions, it is possible for a malicious client to alter the transaction history of others in their log. This can make non-fraudulent users seem fraudulent in the eyes of others who have only learned their transaction history from a malicious third party. As any non-fraudulent client that falls victim to such an attack can clear themselves of any accusations of wrongdoing by providing their own legitimate log, there is some protection against this type of attack. There may, however, be unexplored scenarios and implementations, specifically with regard to server blacklisting, where this can become an issue. Scenarios where multiple users collude to create fraudulent transaction records are also hard to simulate and test due to the large potential complexity of such attacks. Further testing of this type of attack should therefore be undertaken before any such log sharing scheme is implemented.

As discussed, the log sharing mechanism of the protocol can also pose an issue if clients' logs grow beyond the storage capacities of their environments or devices.

This can happen through ordinary use of the protocol. However, it is also conceivable that malicious actors can purposefully generate large numbers of transactions to perform a denial of service attack. Both through DAP and CAP, such an attack can relatively easily exhaust other clients' storage capacities by propagating large logs.

The underlying mechanisms and systems of the protocol may be less secure than here assumed. For instance, the protocol relies on a central server issuing certificates for the signing of transactions. If this server is breached or the certification scheme in other ways can be manipulated, the majority of prevention and protection mechanisms used in the protocol may be compromised. Similarly, we assume that secure hardware provides a high degree of protection from attempts at fraudulent transactions. If this assumption were to no longer hold, even with the other double spending preventions, the volume of fraudulent funds could quickly increase.

7.2.5 Alternative solutions

As we have seen in the evaluation of the protocol with regard to the requirements and simulation results, there are aspects of the protocol that can be improved. As discussed in the previous section as well as in Section 5.1, the different existing technical approaches satisfy different requirements. There are, however, some technical limitations, such as the impossibility of having both finality of transactions and consistency of money supply without solving the double spending problem, that constrain how well any solution to the offline problem can satisfy the outlined requirements.

As discussed in Section 5.1.1 one major design consideration in any offline protocol is whether to implement an account or a token based solution. We have reviewed both types of solutions. Although there certainly are technical differences, no solutions seem to provide a significantly better architecture than the other with regard to the solution's ability to prevent double spending in real time. Due to the intractability of the double spending problem, any solution will be a best effort approach that strives to prevent such attacks as much as possible within reasonable complexities and any design priorities.

It is conceivable that the protocol here presented could utilize a token based solution without any significant change to its effectiveness or complexity. As seen, specifically with Chaum's protocol, such token solutions can offer functionality that account solutions in and of themselves may not provide. Using tokens that, through an exchange, inherently provide traceability and non-repudiation can reduce the need for additional measures such as logs and DAP and CAP here proposed. Such a solution can also be implemented without secure hardware and still maintain an acceptable level of security through traceability. Real time double spending prevention, reusability of funds offline, and finality of transactions would still need

to be ensured through other mechanisms, or the lack of these features would have to be accepted as a design and technical limitation. Alternative solutions that do not conform with the broader design characteristics presented in this thesis may also exist.

7.3 Simulation evaluation

The simulation largely serves its purpose in that it is capable of estimating the effectiveness of the proposed protocol and other considerations of the offline functionality of a CBDC. As we have seen, the simulation provides results for key experiments that provide an insight into measures and mechanisms discussed throughout this thesis. A general critique to the simulation results and experiments is its limitations in amount of runs, large variance between each run, a short simulation period and few testing parameters.

For the experiments, the testing parameters are few and, to some degree, arbitrarily picked. This implicates a biased result, with the outcome highly dependent on the experiment design. The model design is based on the researchers' conception of a complex system, and the outcome of such a simulation model should be discussed only after addressing the assumptions and the presumed context of the model. Whether this context is correct or not is impossible to settle without further research, and an external review of this context would be necessary to give undisputed validation to the results.

The secure hardware assumption of the protocol is not implemented, as mentioned in Section 6.1.6. As it is assumed that secure hardware would limit the number of adversaries capable of abuse in the system, the actual effects of this are yet to be seen, and any additional complexities associated with such an implementation are not discussed as it is still unobserved.

The transaction rate is not varied in the experiment runs, and the real rate and transaction volume could be significantly different from the assumed rates in the simulation. It would be fair to assume this could vary for different hostile adversaries in a real world environment and with different users depending on the use cases defined for the system.

The simulated fraudulent user only covers one attack. A more thorough simulation would include different ways to exploit the system than the ones thought of in this thesis could be possible.

The mobility of a user is another factor not included in the simulation. If an adversary were to completely change its position in the network, as this is possible in a physical setting, several of the preventions may be compromised. A variety

of grouping density would be realistic for most public areas in a city and as the experiment runs only contain 50 nodes, the effects of larger groups is yet to be seen.

The simulation length was determined by the number of ticks, depending on how many user transactions were to execute during the simulation. For the testing of the experiments with different offline periods, the simulation length was set to be a factor of the recovery rate. A better approach could be to end the simulation after an offline scenario is complete to be sure of if online consolidation made one of the features increasingly efficient. In general, longer simulations with a larger quantity of runs, testing an increased amount of random seeds and parameters could improve the accuracy and reliability of the results.

The simulation did not implement the clearing of logs according to the resynchronization described in Section 5.7, based on the rationale in Section 6.1.6. This should be implemented in a fully functional run of the simulation to mitigate the rapid growth in the logs in long runs. The memory issues would prevent larger scale simulations if this extension to the protocol is not implemented.

The simulation framework is created in a modular way and is able to simulate all the simulation experiments. This is possible even as the framework is not specifically created for any of the experiments but as a general transaction simulation framework for nodes in the given architecture design. The simulation model provides a framework for testing the experiments in a quantitative fashion with measurable results. The model simulates the transactions and preventions in the provided graph in a clear and undisputed way, with an implementation of the provided protocol. Even in simulation runs with fewer nodes, the simulation outputs clearly trending results for most variations, yielding a reliable proposition for the protocol's effects and security enhancements. The simulation runs are easily comparable, and it would be reasonable to assume that the effects seen in a smaller scale would hold in a larger scale.

As an overall conclusion of the simulation, it is created based on the requirements and the architecture derived, giving verifiable support for the protocol enhancements suggested in the thesis. Whether the assumptions hold is discussable, and the simulation model could be tested with a broader range of more realistic parameters.

7.4 Privacy

There are key considerations in the design of offline functionality for a CBDC, as well as for the system as a whole, that have largely been neglected in the protocol design and testing in this thesis. One major discussion around CBDCs and offline functionality is what level of privacy and anonymity can and should be provided

to the users. This is both a technical issue, meaning what levels of privacy can be achieved given other requirements and policy considerations. In the central bank's lead research into CBDCs, there are approaches that hold the privacy of users as a key requirement and others where the anonymity of transactions seems almost undesirable. Most banks also hold that different levels of privacy and anonymity should be available to different types of users and for different types of transactions.

Retail CBDCs are often compared to their cash equivalent in the research, and several central banks, therefore, find the anonymity level of cash a desirable level of anonymity for digital currency as well. At the same time, considerations of AML and CFT, as well as principles like Know your customer (KYC), may call for more transparency and traceability at the cost of privacy.

In traditional cryptocurrencies and digital transactions, a differentiation is made between identity privacy and transaction privacy. The former referring to the system's ability to keep its users, and their usage, anonymous and untraceable. Transaction privacy refers to whether the data going into, and potentially out of, any transaction can be kept hidden from an unauthorized observer. Providing identity privacy might be attempted through pseudonyms. Such pseudonyms allow identification of a user without revealing any information about the user's real identity. Non-persistent pseudonyms, i.e., changing pseudonyms with every interaction, may also be implemented to reduce the traceability of a user for any observer. Pseudonyms, however, have been proven to provide, at best, weak anonymity. Even the most sophisticated implementations can be de-anonymized, given enough effort. In offline scenarios, transactions will likely also include close proximity between the involved parties, making the protection of pseudonyms weaker. Other aspects of a CBDC might also reduce identity privacy, such as the fact that any interaction with the system over the internet can tie a user to her IP address. There exist solutions to partly remedy this, but in general, providing complete identity privacy is hard. [ACE+20]

Transaction privacy, similarly, poses some technical challenges. The content of each transaction can be encrypted to hide information such as amounts from observers. This, however, limits the system's ability to check that no user is breaking the rules. In an online setting, a CBDC, if using a central server architecture or a permissioned blockchain, can utilize a trusted third party, such as a payment provider or the central bank itself, to verify the legitimacy of every transaction, even if their contents are encrypted. In offline settings or solutions that, for other reasons, do not rely on a trusted third party, other approaches are necessary. The most common approach to this problem is the use of zero-knowledge proofs, which provide a way for a payer to prove that she is not overdrawing her balance without revealing what that balance is. This type of solution, however, is technically complex and introduces significant computational overhead. They also pose some restrictions on how the

currency, protocols, and transactions can be structured. [ACE+20]

The protocol presented in this thesis provides little privacy in and of itself. The protocol, however, can, and depending on the desired level of privacy in a Norwegian CBDC, should implement pseudonyms. The certificates and account addresses used in the protocol can be created without directly providing any information about the person who owns the account. In terms of transaction privacy, however, the protocol is incompatible with privacy. The security mechanisms applied in the protocol to prevent double spending require users to know the content of transactions from and between other users. This content can, therefore, not be hidden through encryption, and any observer of the system will know all data about every transaction they receive through logs. It is conceivable that mechanisms such as zero-knowledge proofs can be appended to the protocol to allow for encryption of transaction data while still maintaining the same level of security and traceability, but this requires further research. Such an enhancement will probably also require a significant change in the design of the protocol.

7.5 Further research in the field

Offline solutions for retail CBDCs is certainly an area where more research is necessary. We have seen, in the existing literature, that there is no clear consensus on what technical approaches best provides such functionality. There exists little knowledge about the extent to which abuse such as double spending and forging attacks can and will be undertaken in the space. The lack of such knowledge makes it difficult for central banks in the design and research phases of CBDC projects to embrace offline solutions.

Further testing, both through more extensive simulation and real world pilot programs, is necessary to strengthen the understanding of the risks associated with offline functionality. The analysis of risks performed in this thesis provides an indication that the measures suggested can be effective. However, far more extensive research is needed to confidently and accurately estimate the level of risk posed by providing offline functionality for a CBDC. A clarification of requirements, expectations, and priorities with regard to technical possibilities and limitations is also lacking in most CBDC projects to date, and more effort should be put into navigating the general trade-off that exists between functionality and security.

As we have seen, the evaluation of the protocol suggested some measures introduced require more in depth analysis to evaluate potential vulnerabilities. Specifically, increasing log sizes in lasting offline scenarios and log tampering are areas where more research is needed. Similarly, we have seen that the simulation framework provided is limited both through its assumptions and by its simplicity. More extensive simulation

and testing of the protocol suggested and other measures tested is therefore necessary to validate the finding of this thesis.

As long as central problems such as double spending, fair exchange, and the CAP-theorem exists, a combination of preventative measures will likely be necessary to provide offline functionality at an acceptable risk. Defining this combination and the measures it includes is an area in which more openly available research is needed.

7.6 Conclusion

In this thesis, the problem of offline transactions has been discussed in the context of a national Norwegian CBDC. The problem has not been solved, but mitigation techniques have been presented with a protocol for offline transactions. A simulation framework, [EJ22], has been developed for the purpose of validating the protocol. A requirement analysis has been conducted and gathered the core functional and non-functional requirements for a Norwegian CBDC project. Increasing availability of the system has been proven to be an essential feature for the mitigation of offline scenarios as a whole, and the simulation validates the expectations. The simulation further validates the possibility of minimizing the consequences of abuse by implementing transaction limits and shows the effects of the security enhancements in the protocol. The transaction limit efficiently decreases the consequences of abuse in the system when set low enough, independently of if security enhancements are in place. The security enhancements lower the probability of abuse of the system by a measured 94.1%. The combination of security measures derived in the thesis greatly improves the probability of secure offline transactions in a Norwegian CBDC.

In this thesis, various design considerations of a CBDC, with a focus on offline functionality, have been reviewed. Mechanisms to prevent double spending and other types of abuse have been presented for both token based and account based approaches. We have seen that these designs come with different advantages and disadvantages and that although an account based approach is adopted in this thesis, both solutions provide a capable alternative. Chaum's algorithm, and comparable token based solutions, have some neat advantages in terms of fraud detection. However, they struggle to provide real time prevention of fraud, especially in a contingency situation. Both account and token based solutions require additional security mechanisms to prevent double spending in real time in a way that may reduce risks to tolerable levels. Such additions may be secure hardware, blacklisting, logs and log sharing, and other variants.

References

[ACE+20] Sarah Allen, Srdjan Capkun, Ittay Eyal, Giulia Fanti, Bryan Ford, James Grimmelmann, Ari Juels, Kari Kostiainen, Sarah Meiklejohn, Andrew Miller, Eswar Prasad, Karl Wüst, and Fan Zhang. Design choices for central bank digital currency: Policy and technical considerations. IZA Discussion Papers 13535, Bonn, 2020.

[ACH21] Hanna Armelius, Carl Andreas Claussen, and Isaiah Hull. On the possibility of a cash-like cbdc. Sveriges riksbank staff memo, Sveriges Riksbank Staff memo, Stockholm, 2021.

[AD19] Saeed Alzahrani and Tugrul U Daim. Analysis of the cryptocurrency adoption decision: Literature review. In *2019 Portland International Conference on Management of Engineering and Technology (PICMET)*, pages 1–11. IEEE, 2019.

[AK20] Amun Govil Lie Amund Kvalbein, Harald Wium Lie. Bredbåndsdekning i norge 2020. *Norwegian Communications Authority*, 1(1), Sep 2020.

[AKAS+21] Pauline Adam-Kalfon, Henri Arslanian, Klara Sok, Benoît Sureau, Haydn Jones, and Yanjie Dou. Pwc cbdc global index. Apr 2021.

[AL99] Carlisle Adams and Steve Lloyd. *Understanding public-key infrastructure: concepts, standards, and deployment considerations*. Sams Publishing, 1999.

[Aso98] Nadarajah Asokan. *Fairness in electronic commerce*. PhD thesis, University of Waterloo, 1998.

[ASW+18] David R Anderson, Dennis J Sweeney, Thomas A Williams, Jeffrey D Camm, and James J Cochran. *An introduction to management science: quantitative approach*. Cengage learning, 2018.

[Aut17] Norwegian Communications Authority. Robuste og sikre nasjonale transportnett - målbilder og sårbarhetsreduserende tiltak. Technical Report 2, Regjeringen, Apr 2017.

[Aut22] Norwegian Communications Authority. Robuste transmisjonsnett for norge mot 2030. Technical Report 1, NKom, Jan 2022.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[Ban18] Norges Bank. Central bank digital currencies - no 1 | 2018. *Norges Bank*, 2018. (Accessed on 1/17/2021).

[Ban19] Norges Bank. Central bank digital currencies - no 2 | 2019. *Norges Bank*, 2019. (Accessed on 1/17/2021).

[Ban21a] Norges Bank. Central bank digital currencies - no 1 | 2021. *Norges Bank*, 2021. (Accessed on 1/17/2021).

[Ban21b] Norges Bank. Norway's financial system 2021. *Norges Bank*, 2021. (Accessed on 21/03/2021).

[Ban21c] Norges Bank. Sedler og mynter Årsrapport 2020. Jan 2021. (Accessed on 10/12/2021).

[Bar13] Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

[BM+76] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[BMZ18] L. M. Bach, B. Mihaljevic, and M. Zagar. Comparative analysis of blockchain consensus algorithms. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1545–1550, 2018.

[Bre00] Eric A Brewer. Towards robust distributed systems. In *PODC*, volume 7 of *10.1145*, pages 343477–343502. Portland, OR, 2000.

[Bre12] Eric Brewer. Cap twelve years later: How the" rules" have changed. *Computer*, 45(2):23–29, 2012.

[CC21] Usman W. Chohan and Usman W. Chohan. The double spending problem and cryptocurrencies. *Available at SSRN 3090174*, January 2021.

[CFN90] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, pages 319–327, New York, NY, 1990. Springer New York.

[CGK+20] Mihai Christodorescu, Wanyun Catherine Gu, Ranjit Kumaresan, Mohsen Minaei, Mustafa Özdayi, Benjamin Price, Srinivasan Raghuraman, Muhammad Saad, Cuy Sheffield, Minghua Xu, and Mahdi Zamani. Towards a two-tier hierarchical infrastructure: An offline payment system for central bank digital currencies. *CoRR*, abs/2012.08003, 2020.

[CGM21] David Chaum, Christian Grothoff, and Thomas Moser. How to issue a central bank digital currency. *Available at SSRN 3965032*, 2021.

[CLRS22] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[Col22] BankId Collaboration. Bankid, about us. https://www.bankid.no/en/private/about-us/, 2022.

[DP17] Massimo Di Pierro. What is the blockchain? *Computing in Science Engineering*, 19(5):92–95, 2017.

[EHHP09] Peder J Emstad, Poul E Heegaard, Bjarne E Helvik, and Laurent Paquereau. Dependability and performance in information and communication systems. *Tapir Akademisk Forlag, Nardovegen*, 12:7005, 2009.

[EJ22] Sjur Brekke Espedal and Dennis Aleksander Janzso. CBDC offline framework, 6 2022. https://github.com/sjure/CBDC-offline-sim.

[ER+60] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[Fel19] Don Felton. What is a trusted execution environment (tee)? 2019.

[Fis01] George S Fishman. *Discrete-event simulation: modeling, programming, and analysis*, volume 537. Springer, 2001.

[GAN21] Geoffrey Goodell and Hazem Danny Al-Nakib. The development of central bank digital currency in china: An analysis. *https://arxiv.org/pdf/2108.05946.pdf*, 2021.

[GANAHHJ18] Joaquin Garcia-Alfaro, Guillermo Navarro-Arribas, Hannes Hartenstein, and Jordi Herrera-Joancomartí. *Data privacy management, cryptocurrencies and blockchain technology*. Springer, 2018.

[GHHK10] Andres J Gonzalez, Bjarne E Helvik, Jon Kåre Hellan, and Pirkko Kuusela. Analysis of dependencies between failures in the uninett ip backbone network. In *PRDC*, pages 149–156. Citeseer, 2010.

[GL12] Seth Gilbert and Nancy Lynch. Perspectives on the cap theorem. *Computer*, 45(2):30–36, 2012.

[Glo15] GlobalPlatform. The trusted execution environment: Delivering enhanced security at a lower cost to the mobile market, 2015.

[IDLK21] Ikechi Saviour Igboanusi, Kevin Putra Dirgantoro, Jae-Min Lee, and Dong-Seong Kim. Blockchain side implementation of pure wallet (pw): An offline transaction architecture. *ICT Express*, 7(3):327–334, 2021.

[JL21] Jiaying Christine Jiang and Karman Lucero. Background and implications of china's central bank digital currency: E-cny. *https://ssrn.com/abstract=3774479 or http://dx.doi.org/10.2139/ssrn.3774479*, 2021.

[JNB03] Hawoong Jeong, Zoltan Néda, and Albert-László Barabási. Measuring preferential attachment in evolving networks. *EPL (Europhysics Letters)*, 61(4):567, 2003.

[KR09] Charles M Kahn and William Roberds. Why pay? an introduction to payments economics. *Journal of Financial Intermediation*, 18(1):1–23, 2009.

[LBJH+15] Olav Lysne, Kristine Beitland, Åke Holmgren Janne Hagen, Einar Lunde, Kristian Gjøsteen, Fredrik Manne, and Sofie Nystrøm Eva Jarbekk. Digital sårbarhet – sikkert samfunn — beskytte enkeltmennesker og samfunn i en digitalisert verden. *NOU*, 13, 2015.

[LLT+22] Xiu-Ming Loh, Voon-Hsien Lee, Garry Wei-Han Tan, Jun-Jie Hew, and Keng-Boon Ooi. Towards a cashless society: the imminent role of wearable technology. *Journal of Computer Information Systems*, 62(1):39–49, 2022.

[LRW+19] Francisco Lázaro, Ronald Raulefs, Wei Wang, Federico Clazzer, and Simon Plass. Vhf data exchange system (vdes): an enabling technology for maritime communications. *CEAS space Journal*, 11(1):55–63, 2019.

[LWZ+21] Rujia Li, Qin Wang, Xinrui Zhang, Qi Wang, David Galindo, and Yang Xiang. An offline delegatable cryptocurrency system. *CoRR*, abs/2103.12905, 2021.

[MA18] Mahdi H. Miraz and Maaruf Ali. Applications of blockchain technology beyond cryptocurrency. *CoRR*, abs/1801.03528, 2018.

[Nak08] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 2008.

[oC21] People's Bank of China. Progress of research & development of e-cny in china, Jul 2021.

[oCBoJ+20] The Bank of Canada, European Central Bank, Bank of Japan, Sveriges Riksbank, Swiss National Bank, Bank of England, Board of Governors of the Federal Reserve, and Bank for International Settlements. Central bank digital currencies: foundational principles and core features. *Bank for International Settlements*, 1(1), Oct 2020. (Accessed on 17/01/2022).

[oI18] Central Bank of Iceland. Special publication no 12: Rafkróna? *Central Bank of Iceland*, Sep 2018. (Accessed on 11/13/2021).

[OS14] Marek R Ogiela and Piotr Sułkowski. Protocol for irreversible off-line transactions in anonymous electronic currency exchange. *Soft Computing*, 18(12):2587–2594, 2014. (Accessed on 11/13/2021).

[PG+99] Henning Pagnia, Felix C Gärtner, et al. On the impossibility of fair exchange without a trusted third party. Technical report, Citeseer, March 1999.

[PVG03] Henning Pagnia, Holger Vogt, and Felix C. Gärtner. Fair Exchange. *The Computer Journal*, 46(1):55–75, 01 2003.

[PVP15] Debajyoti Pal, Vajirasak Vanijja, and Borworn Papasratorn. An empirical analysis towards the adoption of nfc mobile payment system by the end user. *Procedia Computer Science*, 69:13–25, 2015.

[Rik17] Sveriges Riksbank. The riksbank's e-krona project: Report 1. *Sveriges Riksbank*, Sep 2017. (Accessed on 17/01/2022).

[Rik21] Sveriges Riksbank. E-krona pilot phase 1. *Sveriges Riksbank*, Apr 2021. (Accessed on 13/11/2021).

[Rik22] Sveriges Riksbank. E-krona pilot phase 2. Technical report, Sveriges Riksbank, April 2022.

[RTW+19] Mark Roddy, Thuy Truong, Paul Walsh, Mustafa Al Bado, Yanxin Wu, Michael Healy, and Sean Ahearne. 5g network slicing for mission-critical use cases. In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 409–414. IEEE, 2019.

[RW14] Marc Rysman and Julian Wright. The economics of payment cards. *Review of Network Economics*, 13(3):303–353, 2014.

[SHM21] Morten Stenstadvold, Per-Trygve Hoff, and Tom E. Markussen. Etterevaluering av tetra nødnettprosjektet. Technical Report 1021245, NTNU/Forskningsprogrammet Concept, The address of the publisher, 1 2021.

[TS11] Krishnaiyan Thulasiraman and Madisetti NS Swamy. *Graphs: theory and algorithms*. John Wiley & Sons, 2011.

[US4] A US4405829. Cryptographic communications system and method/ rl rivest, a. shamir, lm adleman (usa); patent holder massachusetts institute of technology.

[WB13] Karl Wiegers and Joy Beatty. *Software requirements*. Pearson Education, 2013.

[Wei01] Joel Weise. Public key infrastructure overview. *Sun BluePrints OnLine, August*, pages 1–27, 2001.

[WG20] Central Bank Digital Currencies Working Group CBDC WG. Implementing a cbdc: Lessons learnt and key insights policy report. 2020.

[Wie14] Roel J Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

Appendix

Appendix A

A.1 Interview with Norges Bank

Intervju 01.02.2022, 08:30-09:30, Microsoft Teams

Deltakere:
Peder Østbye, Spesialrådgiver, Norges Bank
Lasse Meholm, Prosjektkoordinator Eksperimentell Testing, Norges Bank
Sjur Brekke Espedal, NTNU
Dennis Aleksander Janzso, NTNU

Intervju Transkripsjon:

(...) Introduksjon av deltakere. Kuttet.

Dennis: Det første punktet vi har sendt er overordnet om system arkitekturen. Veldig interessant for oss for å prøve å avgrense hva vi ser på. Det er mye løsninger som er diskutert der ute, så det vi lurer på i første omgang er om dere har noe mer avgrensninger om arkitektur nå enn det som er tilgjengelig i rapportene. Tenker spesielt på om det er en DLT-teknologi dere ser for dere og i hvilken grad vil den være sentralisert og distribuert. Har dere noe mer oversikt eller noe mer dere kan komme med der?

Peder: Jeg kan begynne, så kan du legge til Lasse. Vi har ikke kommet så mye lengre i hva som blir den endelige arkitekturen i DSP. Det vil denne eksperimentelle testing vise. Vi har kommet så vidt i gang med denne eksperimentelle testing, i hvert fall i planleggingsfasen. Det som er klart er at vi kommer til å teste en type token-teknologi også kan den være basert på elementer av DLT infrastruktur eller ikke. Man kan ha en token infrastruktur både sentralisert og desentralisert, og det

vi antageligvis kommer til å teste er ERC-20 token som en standard. Dere trenger ikke nødvendigvis bruke den, men vi har en hypotese om at det vi lærer om en ERC-20 token kan være overførbart selv om man ikke skulle bruke en ERC-20 token. Så finnes det mye utviklingsmiljøer rundt ERC-20 token som vi kan dra nytte av i test arbeidet. Jeg vil tro at det vil bli noe testing med den standarden. For offline delen, og nå tenker jeg litt høyt her, så vil det være et spørsmål om en slags offline modul kan kobles på en ERC-20 token standard og være modulært i den betydning at man kan legge penger eller enhetene i offline som kan konsolideres med et register senere. Den konsolideringen vil være mer komplisert hvis du har et DLT-system ettersom da må kommunikasjonen gå ut i et DLT-netverk. Man kan tenke seg en offline løsning på toppen av Bitcoin. Dersom en låser enheter i en offline løsning for å sende ut i nettverket senere, ville det blitt komplisert. Først måtte det blitt sendt ut desentralisert og det ville vært usikkert når det blir lagt inn i blokkjeden. Dette er kanskje ikke like stort problem for en DSP der det eksisterer et register vi har kontroll over der vi kan korrigere for problemstillingene som kan oppstå ved konsolidering mot registeret. Der kunne vi løst slike kommunikasjon problemstillinger. Foreløpig er vi agnostiske til underliggende teknologi og da er det fint med en offline løsning som er robust som kan tilpasses ulike register løsninger og som eventuelt kan skiftes ut. La oss si at vi får en offline løsning som går ut på dato eller viser seg å ikke være kvante sikker eller lignende. Da er det praktisk om vi kan skifte ut offline modulen uten å skifte ut hele registerløsningen. Det samme gjelder omvendt hvis offline løsningen er veldig god, men register løsningen må justeres, så er det fint om offline løsningen fortsatt kunne brukes.

Dennis: Veldig gode poeng. Jeg vet ikke om din kollega vil kommentere noe mer?

Lasse: Essensen er at vi ikke har bestemt oss enda, det kan godt hende vi ender opp med å ikke bruke blokkjede men heller tradisjonell teknologi. Det underliggende i det Peder sier er at vi ikke ønsker å komme i en situasjon der vi har en token som kun én teknologi løser. Vi ønsker å være i en situasjon der sentralbanken kan velge teknologi i fremtiden uten å bli låst inn. Der er ERC-20 et eksempel på det. Det finnes også DLT-teknologier med et økosystem rundt seg. Det er svært attraktivt i den situasjonen vi er i nå der vi skal teste. Dersom sentralbanken går videre med å etablere DSP, så kan det godt være noe helt annet enn det vi tester nå. Grunnen til at vi tester nå er å legge et datagrunnlag for en god beslutning neste sommer.

Dennis: Litt oppfølging der, vi har vært litt innom det allerede. Vi ser mye at det diskuteres i forhold til hierarki i slike løsninger med forskjellige “tiers”. Er det sentralbanken ene og alene som har full tilgang til å se og skrive til en ledger eller en database? Eller er det distribuert med tredjeparter som kan skrive til systemet. Har dere tenkt på dette eller er det mer åpent?

Peder: Vi har det fortsatt åpent, men vi har en hypotese der vi kommer til å administrere den underliggende infrastrukturen og registeret. Tredjeparter som kobler seg på, la oss si at de kjøper DSP fra en bank, da kommuniseres det til det registeret som administreres av Norges Bank. Visse funksjoner og egenskaper med registeret må vi ha tilgang på, for eksempel utstedelse av penger, destruksjon og pengemengde. Man kan tenke seg at andre aktører kan utføre visse funksjoner desentralisert, for eksempel validere transaksjoner og validere smart kontrakter. Foreløpig hypotese er at det er et register vi administrere og alle transaksjoner går inn mot det registeret. Så finnes det andre løsninger som har vært foreslått, for eksempel der banker eller andre institusjoner har et eget register så det ikke skjer en kontinuerlig konsolidering med registeret sentralbanken har. Det kan være en mulig løsning. Kanskje det dukker opp noe der. Den foreløpige hypotesen er at det er et register vi administrere.

Dennis: Det neste vi kan gå litt videre på er når det kommer mer til offline, så er det veldig mange løsninger som er tilgjengelig der ute som baserer seg på en eller annen form for kreditt, depositum eller debit. Der får brukerne en slags kreditt godkjenning slik at de kan bruke penger i en offline setting avgrenset innenfor en kredittramme for eksempel. I en norsk kontekst, vil det være noe Norges Bank kan tilby eller er det mer en rolle for tredjeparts tilbydere. Er dette noe dere har vurdert og sett på?

Peder: Nå ble det flere spørsmål på en gang her, men la oss tenke oss en enkel situasjon. Du har en wallet og en DSP. Så kan en manuelt legge DSP inn i en offline modul eller så kan en tenke seg at dette var automatisert for å sikre at brukerne har noe tilgjengelig offline. Dersom man selv må reservere noe for en offline løsning, kan det være man ikke gjør det, glemmer det, eller tenker at en offline situasjon ikke kommer likevel og dermed har man ikke de pengene offline når det trengs. Det kan derfor tenkes det er noe som blir integrert med ulike wallets, som kan tilbys av tredjeparter. For eksempel kan et krav til wallet tilbydere være funksjonalitet for å gjøre midler tilgjengelig for offline løsningen. Dette har vi ikke bestemt, men dette er et vanskelig punkt. Hvordan kan man sikre at brukere har DSP offline, når den primære løsningen er et register. Dette er ikke noe vi har løst, men forslag fra litteraturen dere har referert til, kan tenkes. Det kan bli løst ved automatisk blir et visst beløp reservert og tilgjengeliggjort til offline løsning. Når det kommer til kredittrisiko som dere har skrevet i eposten. Hvis dere tenker på kortsystemet i dag så har de en offline løsning, der jeg ikke kjenner detaljene. Men der kan man gjennomføre en viss mengde transaksjoner offline. Det er noen begrensninger der jeg ikke kjenner helt, men det vil senere konsolideres med et register. Dersom det ikke var dekning må noen dekke det tapet. Da er den primære løsningen at den som betalte må dekke beløpet. Dersom du betaler med penger du ikke har i dagens offline løsning så vil det oppstå et gjeldsforhold, da vil du måtte dekke dette senere. På butikkene er det sånn jeg har forstått ingen kredittrisiko knyttet til handelen

ettersom det er en forsikringsordning for den situasjonen. Hvis man tenker at det som eksisterer i dag er det enkleste så må det være det samme med DSP hvor det er noen begrensninger, og hvis noen har klart å hacke systemet så skal ikke det være en kredittrisiko som butikken utsettes for alene. Her må man kombinere juridiske eller regulatoriske virkemidler med teknologi. Det vil være et problem dersom løsningen som var såpass dårlig at det er mulig å tilgjengeliggjøre store beløp ved å hacke systemet. Det er en av de tingene vi lurer på, som vi ikke vet svaret på. Hvor sikkert kan det bli og hvor enkelt kan det bli å hacke systemet. Det er noe av det dere ser på som dere kanskje finner svar på med simuleringen deres.

Lasse: Jeg er enig med deg Peder. Butikken løper ikke en kredittrisiko ved offline betalinger. Det forplikter blant annet å signere manuelt, dersom terminalen er offline. Det blir ryddet opp i så fort terminalen er online, det kan ta minutter eller timer. Det er viktig at sentralbanken ikke tar på seg større rolle i samfunnet en det den har i dag. Ønsker ikke sentralbanken å gjøre det, så ønsker sentralbanken at bankene gjør det. Det er ikke en rolle sentralbanken ønsker å ta i fremtiden og ta en enda større del av økosystemet slik at ansvaret blir tynget ned. Det er betaler og ikke mottaker som er ansvarlig for at pengene går igjennom. Dersom betaler utfører noe bevisst, så er det betaler som løper den risikoen og ikke de som mottar pengene. Det er noen land som har innført DSP, der du har DSP i din digitale wallet, samtidig så har du i tillegg et eksternt kort for offline betalinger. Så legger du kortet på mobilen og laster over penger med NFC for å så bruke kortet helt offline. Det er også en mekanisme som er mulig her. Sjur: Er det viktig at det vil være bankene eller en tredjepart og ikke Norges Bank som tar eventuell kreditrisiko?

Lasse: Jeg tror ikke sentralbanken ønsker å ta den.

Sjur: Er det aktuelt med en transaksjons eller beløpsgrense i offline?

Lasse: Det er et konsept med dagens løsning. Det er en beløps eller transaksjonsgrense per tid. Det er mange mekanismer som kan fungere og betalingssytemet har fungert i 30 år allerede. Problemet i et DLT-system kan komme når systemet kommer online igjen, så er det mange systemer som må på plass.

Dennis: Litt videre, det ble nevnt kort med transaksjonsbegrensninger i ideene som har vært offline. I en norsk retail CBDC, er det naturlig at det eksisterer en transaksjonsbegrensning og hva vil den eventuelt være? Det er avhengig av bruksområdet, men er dette noe som burde eksistere online eller offline med DSP?

Peder: Dette med transaksjonsbegrensninger dukker opp i mange sammenhenger i form av ulike begrensninger. Man kan tenke seg begrensninger i forhold til finansiell stabilitet, for å unngå bank runs. Så kan det være begrensninger knyttet til personvern og anonymitet, begrensninger for å oppfylle regulatoriske krav men samtidig kunne

tilby personvern eller full anonymitet eventuelt. Innenfor offline løsninger er det også aktuelt, både for å oppfylle regulatoriske begrensninger og for å hindre misbruk eller begrense tap hvis noen klarer å manipulere systemet. I så fall vil det være viktig at den som manipulerer systemet ikke kan manipulere transaskjonsbegrensingen. Dette er temaer vi ikke vet så mye om, spesielt ved offline løsninger hvordan man kan ha slike begrensninger.

Dennis: Vi lurer også på, hva ville vært størrelsesordenen på en slik begrensning. Hva kunne vært et minimum nødvendig transaksjonsvolum eller antall transaksjoner per dag for eksempel?

Peder: Dette kommer an på formålet med begrensninger. For å fremme finansiell stabilitet så har vi ikke noe beløp der. Når det kommer til regulatoriske begrensninger så finnes det et E-penge regelverk. Du kan, kunne i hvert fall før, kjøpe et e-penge kort som var anonymt på narvesen. Det har kommet noen regler som stadig er under utvikling hvor store beløp som er akseptabelt for en sånn type løsninger. De reglene kan være en pekepinn for transaksjonsbegrensninger for anonyme betalinger. For offline løsninger, hvis man skal legge noen begrensninger så kan et utgangspunkt være de begrensningene som finnes i dag på kort. Vi har ikke gjort noen konkret vurdering på hvor stor begrensningene skal være. Det tror jeg heller ikke er så lurt, det viktigste er at det er fleksibelt teknologisk. Hvis man skal hardkode noen begrensninger som skal gjelde for alltid vil det være lite fremtidsrobuste. Det viktigste er at løsninger er fleksible og at vi kan implementere de grensene som er i henhold til regelverket.

Lasse: Det kan være et utgangspunkt hvor mye man kan ta ut fra minibank i uka. Det kan også være forskjellig hvis du er bedrift eller privatperson. Jeg og Peder har ikke snakket om noen grense her.

Peder: Det viktige her å justere for robusthet, at kan til enhver tid kan justere for regelverket og det som er hensiktsmessig.

Dennis: Det er gode pekepinner vi kan jobbe ut i fra i hvertfall.

Sjur: I forhold til dagens løsninger finnes det kredittkort med forskjellig kredittgrense avhengig av en kredittvurdering av brukeren. Er det noe som kunne vært aktuelt i DSP eller burde det være tilgjengelig for alle?

Lasse: Det vil i så fall være banken og ikke sentralbanken som gjennomfører en slik kredittvurdering.

Peder: Det er viktig at DSP ikke vil være kredit, men som kontanter. Du vil ikke kunne bruke penger du ikke har. På et kort så får du kreditt som banken eller kredittkortselskapet gir det. Det kan man tenke seg at vil være mulig i DSP også. Vi

tenker oss at det ikke er kreditt involvert i DSP og Norges Bank kommer ikke til å gi kreditt til enkeltpersoner, det er i mot formålet. Det vil være opp til private aktører om det skal bli gitt kreditt i DSP eller ikke. Situasjoner som kan oppstå, er at noen klarer å manipulere systemet. Da vil det ikke være snakk om kredittrisiko, men heller en teknologisk risiko for butikkene. I den situasjonen må vi vurdere hvilken rolle sentralbanken skal ta, hvem som skal dekke tapet dersom systemet manipuleres. Dette har vi ikke tenkt så mye gjennom annet enn at vi ønsker å ta minst mulig ansvar. Vi ønsker at dette i hovedsak skal tas av tredjeparter, men dette er noe vi må vurdere nærmere.

Dennis: Det virker som om dere er åpne for å akseptere en viss risiko for misbruk av systemet, stemmer det?

Peder: Det er ikke noe man ønsker, men det er en tradeoff man må gjøre. Ingen offline løsninger helt sikre og ingen systemer er helt manipulasjonsfrie. Hvis man skal tilby en offline løsning så man akseptere at det er teknologisk risiko tilknyttet denne. Det blir så et spørsmål om hvordan gjøre denne minst mulig og hvordan fordele ansvar hvis dette gjøres. Det har også et anonymitet aspekt. Hvis du skal kunne betale anonymt med DSP og kunne manipulere dette for anonyme betalinger så blir det vanskeligere å følge enn hvis alle offline løsninger er knyttet til identitet. Dette var noe vi skrev om i tidligere rapporter. Det kan tenkes at anonymitet og offline er en dårlig kombinasjon nettopp fordi det da kan manipuleres lettere. Selv om kanskje intuitivt for offline betalinger skulle anonymitet være hensiktsmessig, vil det kunne ha noen regulatoriske svakheter.

Lasse: Det kan godt være at noen ønsker å for eksempel konfiskere penger som har vært med i en illegal virksomhet selv om det har vært offline. Da har du en del utfordringer som Peder snakket om.

Sjur: Ettersom dere nevner at det ikke skal være noe kreditt, må vi anta at alle brukere har token, altså en slags DSP token før systemet går offline for å kunne bruke penger.

Lasse: Du kan ikke ha kreditt i kontanter i dag, skal du låne må du låne i vanlige penger og så ta ut penger i minibanken. Det er det egentlig lånte penger selv om kontantene ikke er lånte penger. Det blir det samme her. Banken kan yte kreditt, men DSP er tilsvarende en elektronisk form av kontantene.

Sjur: Med debitkort i dag, har en kreditt selv om det vil være tilsvarende DSP.

Peder: Man kunne tenke seg at det private vil tilby noe sånt, tilby en mengde kreditt, men sentralbanken vil ikke gi kreditt. Så kan det oppstå en krisesituasjon og det viser seg at personer ikke har lastet penger over i offline løsninger hvis det

er et krav. Da kunne en tenke seg at ut i fra samfunnsforhold vil man få kreditt for å dekke minimale behov. Det er ikke sikkert det er sentralbanken som skal tilby kreditten, det er noe som kan tilbys av det offentlige. Det er noe det offentlige kan tilby, å bruke et visst beløp en krisesituasjon uavhengig av hva du hadde på forhånd hvis du skulle trenge det, det kan så bli dekket i ettertid av å gå på den som har brukt pengene. Dersom den ikke har pengene må det dekkes gjennom en annen ordning. En sånn løsning vil ikke være det samme som at sentralbanken gir kreditten. Det vil si at vi har en løsning som muliggjøre at alle borgere har et visst beløp på offline løsningen i en krisesituasjon for eksempel.

Lasse: Det som har skjedd tidligere etter krigen var at staten trykte kuponger. Du kunne få 10 kuponger på hvetemel og 20 for sukker for eksempel. Så kunne du betale med papirkupongene på butikken. Det er mulig i en digital verden.

Peder: Analogien til det Lasse, virker litt virkelighetsfjernt nå, men å kunne gå til utdelingspunkter for å fylle på penger på kortet.

Dennis: La oss gå videre til å prøve å definere offline. Vi lurer på hva en offline løsning skal dekke. Det er et spekter fra små hverdagslige transaksjoner opp til full systemsvikt hvor hele nettverket er tilgjengelig over lengre tid. Hva er det mest interessante? Burde det være forskjellige offline løsninger for forskjellige tilfeller? Er det nødvendig at det dekker små hverdagslige offline settinger? Hva er mest interessant fra deres perspektiv?

Peder: Vi kan begynne overordnet. Vi tenker oss en offline situasjon. I utgangspunktet får du DSP i en online situasjon fra banken din. Da vil du være avhengig av banken og online systemene for å få DSP. I betalingssituasjonen er det viktig å kunne gjennomføre offline betalinger. Det vil da være utgangspunktet. Hva skal man ikke være avhengig av for å gjennomføre en offline transaksjon. Vi har tenkt at man ikke må være avhengig av internett eller teleinfrastruktur, en opplagt case, for da må man være online. Så er spørsmålet om man skal kunne betale uten tilgang til elektrisitet for eksempel. La oss si det er knyttet til en wallet i mobiltelefonen din, så må du lade telefonen for å kunne bruke walleten. Vår foreløpige hypotese er at offline dekker den første situasjonen, altså mangel på tele og internett forbindelse men avhengig av strøm på en eller annen måte. Det må så vurderes om det skal lages en løsning som er uavhengig av strøm. Det kan da tenkes å lage kort som er robuste mot forstyrrelser i elektrisitet forsyningen. Først tenker vi offline, uten telenettet og internett. Bruksområdet er i betalingssituasjoner, hensikten er å få tak i medisiner, mat og dekke grunnleggende behov. En offline løsning må i hvert fall kunne virke for å kjøpe de grunnleggende tjenestene. Så kommer spørsmålet om teknologien støtter den avgrensningen. En kunne tenke seg at butikker og utsalgssteder blir pålagt å ha en løsning for å ha offline løsninger tilgjengelig hvis det er spesialutstyr. Det kan

være for mye å kreve for alle butikker og utsalgssteder for å kunne motta offline betalinger. Ideelt sett ville det beste vært om man kunne bruke mobiltelefonen eller en chip og at alle kan gjennomføre offline betalinger uten omfattende investeringer for å kunne motta offline betalinger.

Dennis: Der det blir teknisk vanskelig er en total systemsvikt over en lengre tid. Hvis man har en ledger teknologi så må systemet synkroniseres regelmessig for å kunne fungere. Er det et krav at DSP skal kunne fungere dersom tilgangen til Norges Bank og nettverket som en helhet er nede i dagesvis om gangen i krisesituasjoner? Er det rimelig å forvente at systemet skal fortsette å fungere i en slik offline setting?

Peder: Her burde jeg visst bedre hvilke krav som settes til banker og betalingsaktører i dag, men det kan være et naturlig utgangspunkt. Riksbanken opererer i sin rapport med en ganske lang tid. Vi har ikke gjort noe konkret vurdering av hva som er hensiktsmessig tid hvor man kan være offline. Jeg antar det er en slags trade-off, desto lengre en har en offline løsning, desto større teknologisk usikkerhet og større sikkerhetsutfordringer får du. Jeg antar det er mulig å lage en løsning for å gjøre tradeoffen når den oppstår, ikke hardkodede grenser, men å kunne gjøre en avveining om man vil ta risikoen løpende. Her håper jeg det finnes fleksible løsninger for å slippe å bestemme konkrete tall i forkant, men at man kan justere etter hvert og gjøre avveininger for ulike risikoer mot hverandre.

Lasse: Det er greit å sette litt opp mot kontanter her. Man kan si hva man vil om kontanter, men det er veldig bra i krisesituasjoner. En trenger hverken strøm eller nett eller noenting. Det fungerer alltid. Det er en god målestokk, hvis man klarer å lage en digital erstatning så hadde det vært glimrende.

Peder: Med kontanter er det også en forfalskningsrisiko selv om den er liten. Man kan tenke seg at pengene går innom sentrale punkter der de blir kontrollert. Desto lengre intervallet ble, desto større er sjansen for falske kontanter i omløp. Det har skjedd, det har kommet falske kontanter i omløp fordi man ikke har den infrastrukturen for å sjekke gyldigheten av kontanter. Det har da oppstått spørsmål om Norges Bank skal gå god for kontantene av samfunnshensyn. Nå har ikke Norges Bank gjort det, men det belyser at det er en avveining mellom ulike risikoer.

Lasse: En annen ide jeg har diskutert for mange år siden er et digitalt sjekkhefte. Det ville løst offline på en litt annen måte også.

Dennis: Når er en transaksjon fullført og hvordan definerer man det? Spesielt i en offline setting blir dette vanskelig. Er det slik at Norges Bank ønsker å være ansvarlig for finality av hver transaksjon, altså når den er gjennomført, eller kan det distribueres til tredjeparter. Vil tredjeparter kunne ha offline løsninger der de verifiserer transaksjoner til en eventuell blokkjede hvis det blir teknologien?

Peder: Finality er i hovedsak et regulatorisk-juridisk spørsmål og det er særlig viktig ved systemisk viktige betalinger. Dersom det kommer en stor betaling for verdipapirer og det blir usikkerhet på om pengene er overført eller ikke så vil det kunne få systemiske konsekvenser. I sånne situasjoner er finality svært viktig. Også i forbruker situasjoner er dette viktig ettersom det dreier seg om når et konkursbo kan dra inn penger som er betalt til en betalingsmottaker. Det vi har som utgangspunkt hvis det er en registerløsning er endelighet i registeret. Den vanskelige situasjonen er den dere er inne i, ved offline, når noe skal konsolideres med et register senere. Det kan da tenkes at det oppstår konflikter om endeligheten av en transaksjon. Hvis noen gjennomførte en offline betaling, som ikke var konsolidert med registeret og før det ble online igjen gikk vedkommende konkurs. Da er det et spørsmål om konkursboet kan trekke inn pengene igjen eller om de er overført. Dette tenker jeg vil være i hovedstad regulatoriske spørsmål som kan løses ved å definere visse krav til når en betaling anses som endelig, også i en offline løsning. En kan tenke seg som i dag, at det eksisterer sikringsordninger for at ikke mottaker skal ha noe risiko selv om det ikke skulle være endelig. Det er dermed hensiktsmessig med en teknisk løsning for å se om penger er betalt eller ikke. Det kan dermed være klarere når man skal lage et regulatorisk regelverk om hvilken betaling som er endelig. Det må kunne knyttes klart til hvilken betaling som er endelig i den tekniske infrastrukturen eller gjøres på en måte så det ikke fortsatt er uklart.

Lasse: Det blir veldig vanskelig hvis mottaker av pengene er usikker på om den har fått pengene. Mottaker vil gjerne bruke pengene rett etterpå. Hvis en blir usikker på om pengene faktisk er mottatt vil tilliten til mekanismen bli vanskelig å kommunisere.

Peder: Det er viktig med sammenhengen mellom det regulatoriske og det tekniske, den som råder over pengene er også eier av pengene, regulatorisk sett.

Dennis: Hvordan ser dere for dere at systemet skal være hosted. Ved en sentralisert ledger, vil det være noe Norges Bank har hos seg eller vil det være distribuert utover Norge. Hvis vi skal simulere et nettverk, kan man regne med at en versjon av ledgeren vil være tilgjengelig i de fleste geografiske områder eller kommer den til å være sentralisert?

Peder: Alle sånne systemer er til en viss grad distribuert ved at man har backuper i tilfelle en går ned. Det er systemene til sentralbanken og bankene i dag. De har reserveløsninger og reserve-reserveløsninger. Ledgeren vil ligge på forskjellige databaser, så er spørsmålet om den vil ligge sentralt og det bare er reserveløsninger eller om man skal ha forskjellige databaser rundt om kring som til enhver tid konsolidering mot hverandre, at det er distribuert, men at man ikke har et desentralisert nettverk. Man kan også tenke seg at det er desentralisert der ulike aktører har noder og kan

validere i nettverket. Vi har ingen sterke føringer der, det kan være mulig med et permission nettverk der banker eller andre aktører har en valideringsrolle. Disse to spørsmålene henger sammen med hverandre, hvis man må gjøre avveininger så er det viktig å få belyst dem. Dersom en offline løsning er mye vanskeligere hvis man skal ha gevinster ved desentralisering så må man gjøre en avveining om man skal ha en god offline løsning eller gevinster ved et offline nettverk. Så dette kan være fint om det blir belyst dersom denne tradeoff eksisterer.

S.B. Espedal & D.A. Janzso

Design Choices for Offline Transactions in a Norwegian Central Bank Digital Currency

NTNU
Kunnskap for en bedre verden