



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

*Distributed
Computing*



Cryptomoney In Real Life

Semester Thesis

Yu Chen

yuchen14@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:

Ye Wang

Prof. Dr. Roger Wattenhofer

January 10, 2021

Abstract

Despite the development of digital payment, an integration between physical and digital money is missing under the current payment environment. This project aims to fill this gap with a new secure transaction system. The system supports a convenient conversion from one money form to another. Unilateral offline payment is realized in the system and a customer-merchant payment architecture is applied. The system is implemented on an Android App. Some experiments are carried out to test the App functionality and its acceptance degree by new users.

Contents

Abstract	i
1 Introduction	1
1.1 Two challenges	1
1.2 Project goal	2
1.3 Structure overview	2
2 Background	3
2.1 From crypto-currency to CBDC	3
2.2 Account-based vs token-based transaction model	4
3 System Design	5
3.1 Physical/Digital money conversion	5
3.1.1 Digital to physical conversion	5
3.1.2 Physical to digital conversion	7
3.2 Contract-based transaction process	8
3.2.1 Contract establishment	9
3.2.2 Payment with contract	9
3.3 Customer-Merchant payment architecture	13
3.3.1 Merchant registration	14
3.3.2 Merchant money generation	15
3.3.3 Merchant money receiving	16
3.4 Security analysis	16
4 App Implementation	18
4.1 Printing interface	18
4.2 Receiving interface	19
4.3 Other interface	19

CONTENTS	iii
5 Experiment	27
5.1 Performance test	27
5.1.1 Test results	27
5.1.2 Result analysis	28
5.2 User test	30
5.2.1 Test results	30
5.2.2 Result analysis	30
6 Conclusion	32
Bibliography	33

Introduction

We live in a world where all kinds of novel payment methods have altered our view of money. Our property becomes a simple digit displayed in our bank account. Many people no longer carry cash with them since credit card or mobile payment has become a better replacement. We enjoy the convenience brought by this ongoing digital transformation, but from time to time we also encounter some little troubles.

1.1 Two challenges

Without doubt, digital payment is now everywhere. There are a lot of advantages of paying digitally, such as its traceability, convenience and low cost. Due to its obvious advantages, some countries even have stopped printing cash.

However, digital payment has also brought up some concerns for some people, especially when talking about its relation with cash. These concerns can be divided into two categories, its social challenge and its technical challenge.

Social challenge

First, with the vast development of digital payment, the social acceptance for cash is getting lower, and this is not good news for the less technically knowledgeable population and children who can't apply for a credit card.

Second, since we seldom use cash, we gradually lose the experience to verify cash as well as the ability to count money quickly.

Also, it always feels that digital money just loses some symbolic meaning than cash. For example, in China, parents give children red packets to represent their best wish in Spring festival, although today we can send digital red packets instead, the digital red packets can't convey the wish as clearly as cash [1].

Technical challenge

Especially for a retailer, the most irreplaceable advantage of cash over digital payment is offline verification. In a digital payment, if the retailer does not have a good Internet connection, the payment can't go on.

The challenges above reveal a general problem, which is that there has not been a good mechanism to coexist the digital and cash payment. It seems better, if people can keep the old habit of using cash while enjoying the convenience brought by digital payment.

1.2 Project goal

This project aims to find a way to mitigate the problems in Section 1.1. To be specific, this project wants to build a new payment mechanism that can achieve the following objects.

- **Convenient physical/digital money conversion:** People can withdraw or deposit money at any place without going to a bank counter or ATM.
- **Unilateral offline payment:** One of the two parties of a transaction can be offline when using this mechanism.
- **User-friendly:** The operation is simple and can be quickly learnt by people of any age.
- **Security:** Security is the fundamental consideration throughout the project. The mechanism guarantees that all sensitive online communication is encrypted and all sensitive data is only one-time effective.

The new payment mechanism will be presented in the form of an Android mobile application. The App allows users to print their own crypto-money with their phones and use them offline safely and conveniently. The design details are covered in Section 4.

1.3 Structure overview

This thesis is structured as follows.

Chapter 1 introduces the project motivation. Chapter 2 gives some background related to digital currency development and transaction model design. Chapter 3 illustrates the System design detail. Chapter 4 demonstrates the real App interface. Chapter 5 analyzes the experiment results regarding the App and the new payment mechanism. Chapter 6 draws the conclusion and considers some future work.

Background

One significant trend throughout money evolution is virtualization. From shell to check, we have been decreasing our independence on the physical currency. From fiat currency to all kinds of crypto-currency, the finance market has witnessed the trend of settlement decentralization and anonymity. At the same time, electronic commerce has also been exploring innovative business models driven by digitization.

2.1 From crypto-currency to CBDC

Digital currency is an inevitable result of currency evolution. It includes commercial crypto-currency and central bank digital currency (CBDC) [2]. The latter has gained increasing attention in recent years due to the maturity of blockchain-based financial technology. It is also developed in order to deal with the existing implications in current fiat money transaction system [3]. Some advantages of CBDC are listed as follows [4].

- **Decreased cost:** Issuance, circulation and audit cost is decreased due to digitization.
- **Increased transparency:** CBDC is more traceable and more transaction data can be generated.
- **Increased stability:** CBDC decreases dependence on the private payment system and increases the central bank's macro-control ability.
- **Increased efficiency:** Transaction of large amount becomes more efficient.

CBDC has a broad design landscape. Some important design parameters are listed as follows.

- **Value:** How much value a CBDC represent and its location in the current financial system.

- **Distribution:** The issuance scope and distribution structure of CBDC.
- **Architecture:** Whether CBDC adopts a centralized or decentralized, or partitioned system architecture.
- **Format:** How CBDC is perceived. Two common choices are account-based and token-based. See Section 2.2 for more details.

2020 is a turning point where the CBDC competition between countries heats up. According to BIS working paper in August[5], many developed countries, including the USA, Europe, Japan have turned their attitudes from cautious observation to active research. The Covid-19 pandemic has also accelerated the work due to the strong demand for social distance. Chapter 3 introduces a CBDC design conducted in China.

2.2 Account-based vs token-based transaction model

As given in Section 2.1, an important design parameter in CBDC is whether the transaction model is account-based or token-based. A common definition of the two models is adopted by Charles Kahn and William Roberds[6]. In general, the account-based model verifies the authentication of the payer, while the token-based model verifies the authentication of the token/transaction itself. Table 2.1 lists some common distinctions between two transaction models.

	Features/Pros
Account-based	intuitive, online, no double-spending issue
Token-based	cash-like, no third party involved, offline-supported

Table 2.1: account-based vs token-based transaction model

Besides, security is more of a concern for tokens. Currently, the security of many token-based systems relies on secure hardware, which is vulnerable to tampering attacks. A security breach also has more severe consequences on tokens than on a personal account.

Bitcoin started the token-based era in digital currency [7]. It uses the UTXO model to record bitcoin transactions. Every transaction needs to clarify the input transaction number indicating where the bitcoin comes from. Therefore, every bitcoin is traceable back to its origin.

Compared with cash, bitcoin is not purely token-based. According to the UTXO model, each bitcoin is bound to a specific account address once it has been mined. There is not a transitive state where a bitcoin does not belong to any address, while a banknote can be spent by anyone.

System Design

An Android App implementing a new crypto-money transaction system is designed to deal with current issues in digital payment. The design adopts a client-server architecture and mainly focuses on functionality and system integrity. Two important goals are secure money conversion and offline payment support at the software level. Convenience and concision are also emphasized throughout the operation.

3.1 Physical/Digital money conversion

This App allows users to perform the conversion at any time. On the one hand, users can print out their money from their phones and pay physical crypto-money offline. On the other hand, after receiving other's physical money, users store them in digital form.

3.1.1 Digital to physical conversion

Two phases are required to convert digital money to physical money.

Phase 1: Generate token

Before printing out the physical money, two elements are needed to construct the money. One is a one-time **token address**, which is used to uniquely identify the money. The other is an **RSA key pair** uniquely bound to each token address. This process requires a stable Internet connection.

Assume a user initially has 100.0 in the account and wants to print out 10.0 physically. After receiving the user request, the App will first generate a new RSA key pair locally, then send the requested amount and RSA public key **token pk** to the server. After receiving the client printing request, the server will perform the following steps.

- **Step 0:** Receive client request amount (10.0) and token pk.
- **Step 1:** Check if the client has enough balance to print 10.0.
- **Step 2:** Generate a unique token address for this 10.0.
- **Step 3:** Deduct 10.0 from the user's account.
- **Step 4:** Record user ID, token address, token pk and free money amount.
- **Step 5:** Send back token address or error message.

By deducting the money from the user's account in advance, the system can avoid the double-spending problem of this money. If a token address is received successfully, the App will store the token address and its RSA private key **token sk**. Fig 3.1 shows the flow chart of this interaction process.

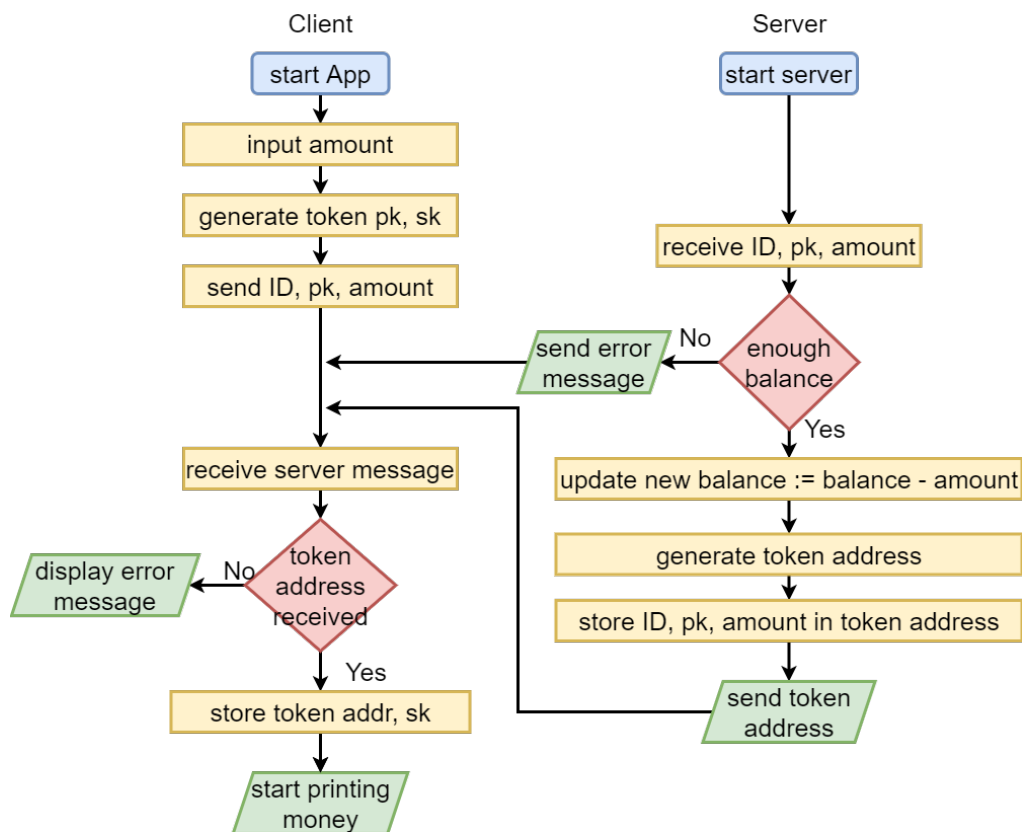


Figure 3.1: Token generation process

Phase 2: Print money

As given in (3.1), the token address and token private key together form the **free money**, which means the money does not belong to any account.

$$\textit{free money} := \textit{token address} || \textit{token RSA private key} \quad (3.1)$$

The free money can be stored in a QR code or an NFC chip, or both. To be specific, the system provides 3 modes to print free money:

- **all QR:** Store all free money data in a QR code.
- **all NFC:** Store all free money data in an NFC chip
- **half-half:** Store the odd part in a QR code and even part in an NFC chip.

In all QR and half-half mode, the App will try to connect to a specified Bluetooth printer and start the printing process automatically. If no printer is found, the QR code will be directly displayed in the App. In both cases, the QR code will also be saved to the system album. In all NFC and half-half mode, the App will notify the user to touch an NFC chip. Different printing choices increase flexibility and free money security. Fig 3.2 illustrates this printing process.

3.1.2 Physical to digital conversion

Free money can be received by anyone if the receiver knows the complete free money data. This can be done by scanning the QR code or/and reading the NFC chip. When the receiver App receives a possible complete free money, it encodes the receiver ID with the private key stored in that free money, and sends the encrypted ID along with the token address to the server. (3.2) gives the final **free money receiving message** M_{rcv_free} sent to the server.

$$M_{rcv_free} := \textit{token address} || \{\textit{receiver ID}\}_{\textit{token sk}} \quad (3.2)$$

After receiving M_{rcv_free} , the server will perform the following steps.

- **Step 1:** Get the token pk and money amount linked to the token address.
- **Step 2:** Decode receiver ID with token pk.
- **Step 3:** Increase receiver's account balance by free money amount.
- **Step 4:** Delete this token address information.

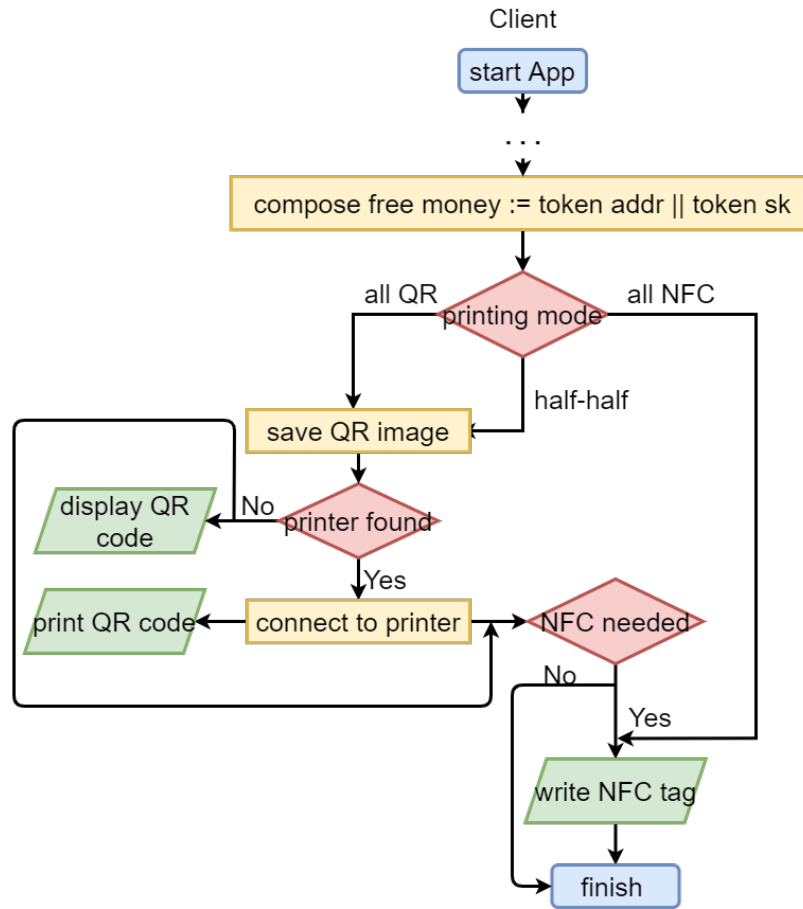


Figure 3.2: Money printing process

Fig 3.3 illustrates the free money receiving process. This process is purely **token-based** since the free money generator is not involved, which also realizes the **payer offline**.

3.2 Contract-based transaction process

The real transaction is more complicated than printing and receiving money. Even for a small transaction in a retail shop, there are different payment scenarios to consider such as money return and amount renegotiation. For convenience and security consideration, a **contract-based** transfer mechanism is provided in the system.

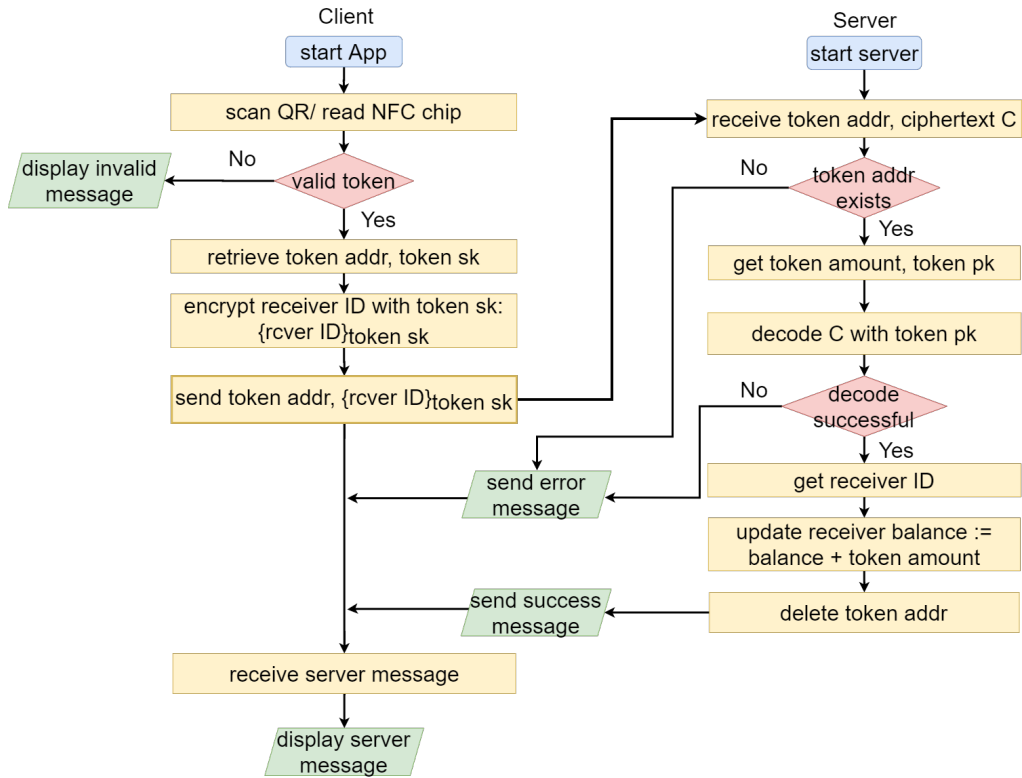


Figure 3.3: Free money receiving process

3.2.1 Contract establishment

Section 4.1.2 only introduces the single free money receiving case, while the contract-based mechanism realizes consequent money receiving process. Before receiving multiple free money, a contract between the receiver and the server is established. Similar to the free money token generation process, a contract is initialized on the receiver's App with a new RSA key pair. The public key **contract pk** is sent to the server along with the contract information such as contract initiator ID, target amount A_{target} and received amount A_{rcv} so far (set to 0 initially). After recording the contract information, the server returns a unique **contract address**.

Fig 3.4 illustrates the communication process of contract establishment.

3.2.2 Payment with contract

The App starts receiving free money after receiving the contract address. For each received free money, the App will send an encrypted **contract receiving message** $M_{contract_rcv}$ to the server, which double-encrypts the token address of

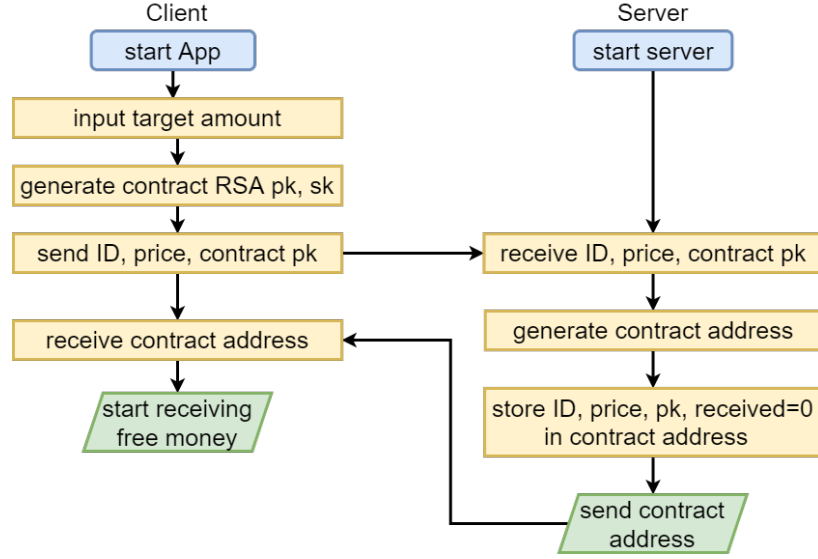


Figure 3.4: Contract establishment process

that free money and the receiver ID with token private key and contract private key **contract sk**. (3.3) shows the construction of M_{ctrct_rcv} .

$$M_{ctrct_rcv} := ctrct\ addr \parallel \{token\ addr \parallel \{rcvrID\}_{token\ sk}\}_{ctrct\ sk} \quad (3.3)$$

After receiving M_{ctrct_rcv} , the server performs the following steps.

- **Step 0:** Receive M_{ctrct_rcv} .
- **Step 1:** Decode token address with contract sk.
- **Step 2:** Decode receiver ID with token sk.
- **Step 3:** Check if receiver ID matches contract initiator ID.
- **Step 4:** Record free money amount and its generator ID linked to the token address under the contract address.
- **Step 5:** Delete this token address information.
- **Step 6:** Update received amount A_{rcv} by increasing the new free money amount.
- **Step 7:** Compare A_{rcv} with target amount A_{target} . If $A_{rcv} \geq A_{target}$, starts change return process, otherwise starts waiting for next M_{ctrct_rcv} .

The goal of establishing a contract is to maintain the state for each transaction so that a consequent receiving process is enabled. Fig 3.5 illustrates the communication after the first free money is received by the receiver under a contract-based mechanism.

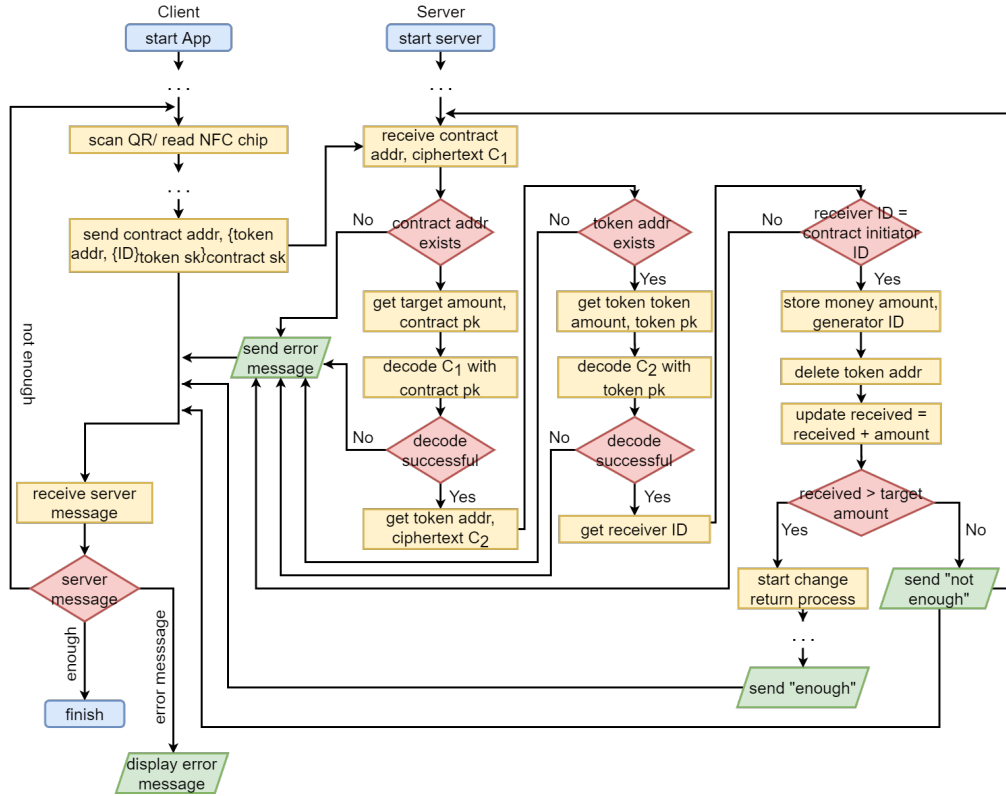


Figure 3.5: First contract-based free money receiving process

As given above, the server starts **change return** process once enough free money has been received under the contract address. The process is given in the following.

Change return

The entire change return process is conducted on the server-side. Since the server has recorded the amount and generator ID of each free money, it is trivial for the server to calculate the change and return it to the account of the last received free money's generator. The process is shown in Fig 3.6.

The mechanism also provides services of **amount renegotiation** and **transaction cancellation**. The explanation is given as follows.

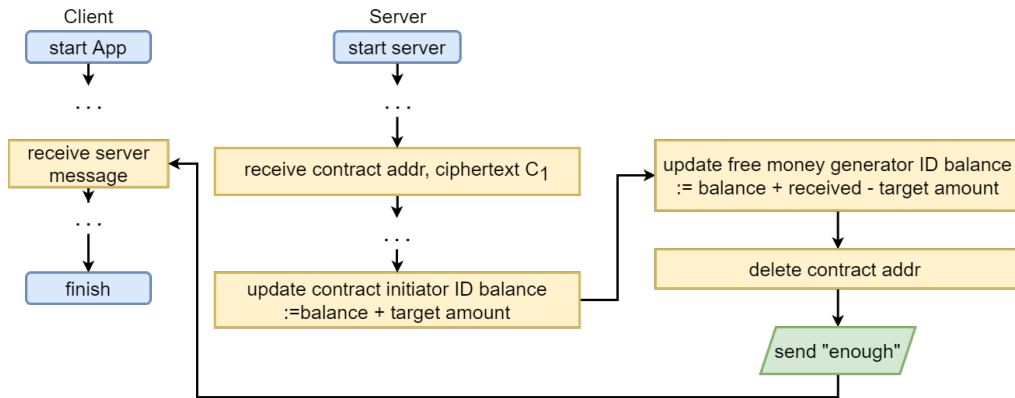


Figure 3.6: Change return process

Amount renegotiation

A receiver can change the target amount during the current transaction. The updated target amount A_{new_target} is sent to the server after encryption with contract sk . When the server receives A_{new_target} , it updates the contract and rechecks if the received amount is enough. Fig 3.7 illustrates this process. Some error handlings are omitted here for simplicity.

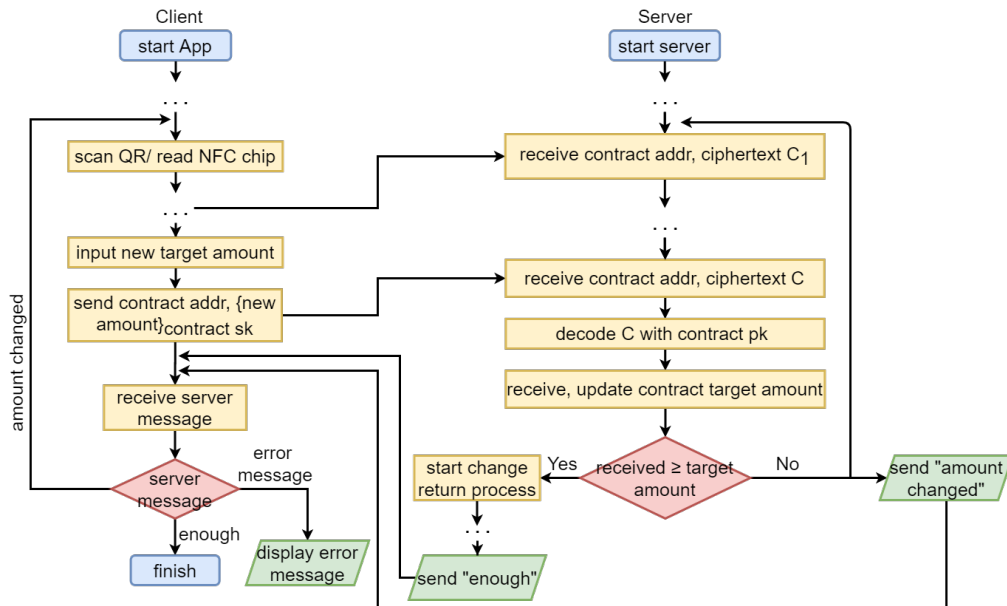


Figure 3.7: Amount renegotiation process

Transaction cancellation

A receiver can terminate the current transaction at any time. When the server receives a cancellation request encrypted in contract sk , it will return already paid free money under the contract address to its corresponding generator. For instance, in one transaction, Alice has paid Charlie 5.0, Bob has paid Charlie 10.0, once Bob cancels the transaction, Alice and Charlie will immediately get back 5.0 and 10.0 respectively in their account balance. The token addresses linked to their free money are also **expired**, which prevents a malicious Charlie from canceling a transaction deliberately and stealing free money later. Fig 3.8 illustrates this cancellation process.

By establishing contracts between the receiver and the server, the system manages to automate as many procedures as possible for the receiver, while also guarantees free money security during the transaction.

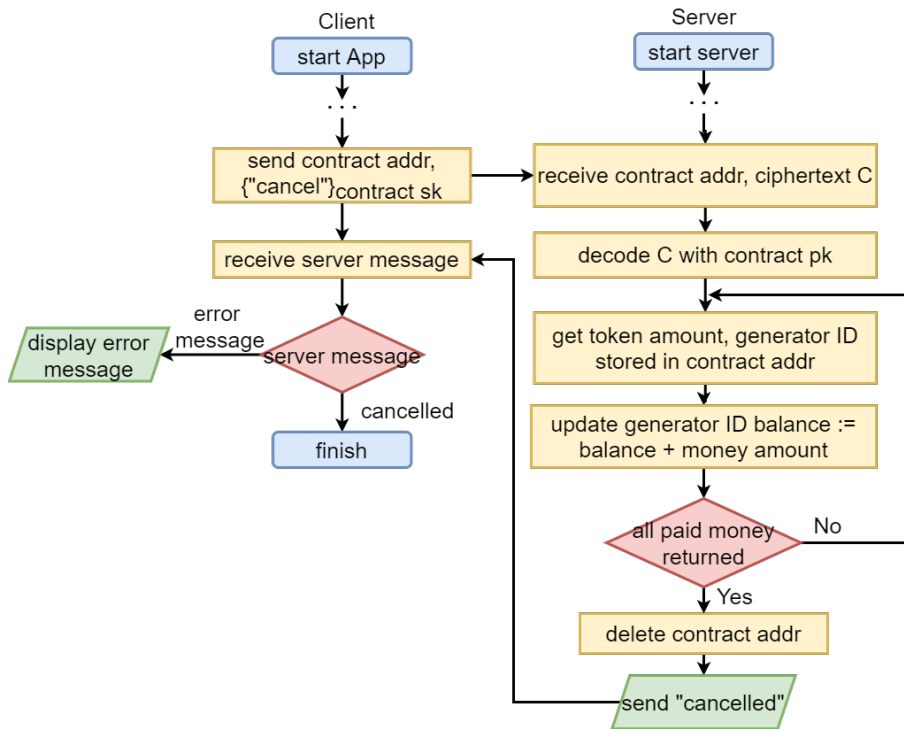


Figure 3.8: Transaction cancellation process

3.3 Customer-Merchant payment architecture

Section 4.1 and 4.2 have realized payer offline since the physical free money receiving process does not require the payer to be online if he/she has printed

out enough free money in advance. The system also realizes the receiver offline in the case of the receiver being a specific retail merchant. The general design idea is to configure a unique long-term RSA key pair for each merchant. The money designated for a merchant is encrypted with that merchant private key **merchant sk** so that only the designated merchant can decode the money with the merchant public key **merchant pk** and receive the money.

3.3.1 Merchant registration

To configure the merchant key pair, the system provides two account types: **customer** account and **merchant account**. When a retail merchant registers a merchant account, the server will generate a long-term merchant key pair and send back the merchant pk. Before any approval, the merchant account registration should be carefully examined to ensure the applicant is indeed the qualified merchant. Fig 3.9 illustrates the merchant account registration process.

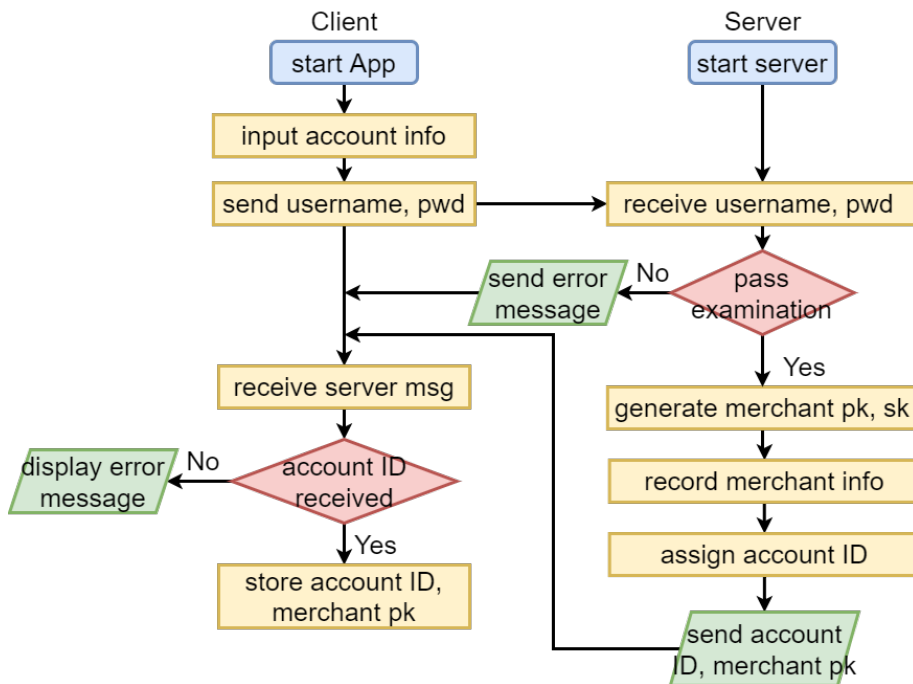


Figure 3.9: Merchant account registration process

After the merchant account registration, customers can generate **merchant money** that can only be recognized and received by the designated merchant.

3.3.2 Merchant money generation

Merchant money can be understood as a voucher for a merchant. When the customer wants to generate merchant money, he/she first selects a merchant, then the App generates a new token key pair, and sends the token pk, money amount along with the selected merchant account to the server.

After receiving the client message, the server will perform the following steps.

- **Step 0:** Receive token pk, money amount and selected merchant account.
- **Step 1:** Check client account balance.
- **Step 2:** Generate token address.
- **Step 3:** Get selected merchant sk.
- **Step 4:** Encrypt token address and money amount with merchant pk to get ciphertext C_m .
- **Step 5:** Encrypt C_m with token pk to get ciphertext C_t .
- **Step 6:** Send back C_t .

(3.4) and (3.5) give the construction of C_m and C_t .

$$C_m := \{token\ addr \ || \ money\ amount\}_{merchant\ sk} \quad (3.4)$$

$$C_t := \{C_m\}_{token\ pk} \quad (3.5)$$

When the customer prints out the merchant money, the App concatenate the token sk at the end of C_t . Therefore, the final merchant money is like (3.6). The generation process is illustrated in Fig 3.10.

$$merchant\ money := C_t \ || \ token\ sk \quad (3.6)$$

The reason why double encryption is used here is to make sure that only the customer who knows the token sk can pay with this merchant money. If the token address is only encrypted with merchant sk and an adversary intercepts the ciphertext sent from the server, then he/she can directly use this token. By performing double encryption, the merchant can check if the payer is authenticated to use this money.

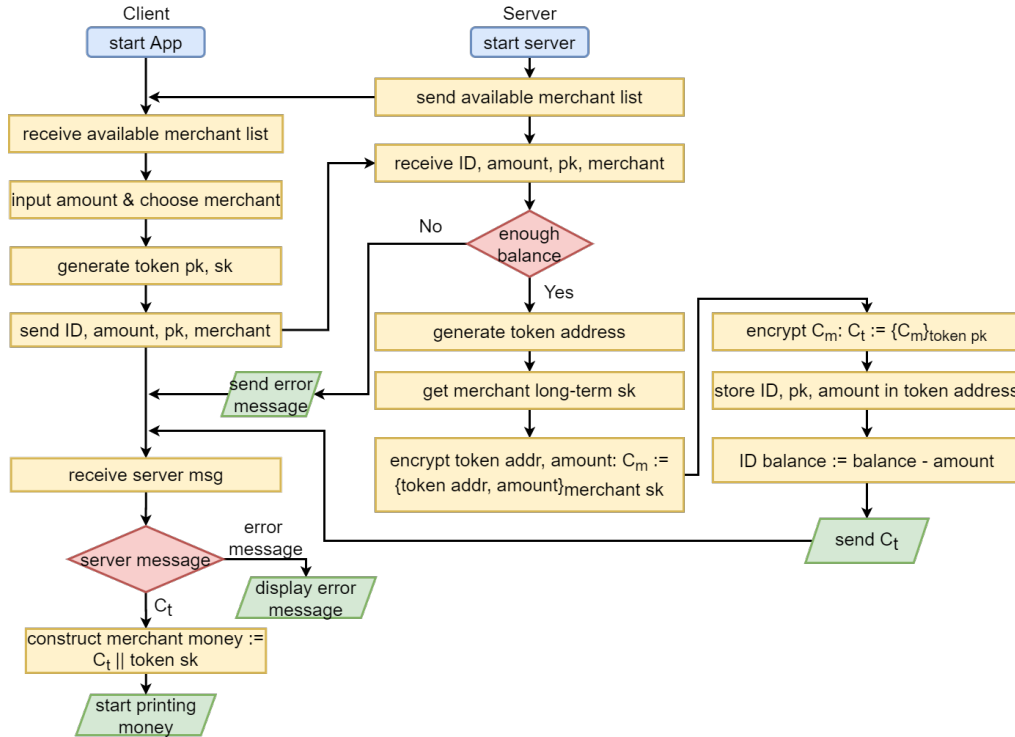


Figure 3.10: Merchant money generation process

3.3.3 Merchant money receiving

The encrypted merchant money can be verified offline. When the merchant receives merchant money, the App first uses the token sk stored in the money to decode C_t . Then it decodes C_m with stored merchant pk. If both decryptions succeed, the merchant now gets the token address and amount. To determine whether the token address is expired, the App maintains a local database that records all token addresses it has seen before. When the merchant is back online, the App automatically sends the received money to the server, as is does when receiving free money. Fig 3.11 illustrates the merchant money receiving process.

3.4 Security analysis

The system has the following security properties.

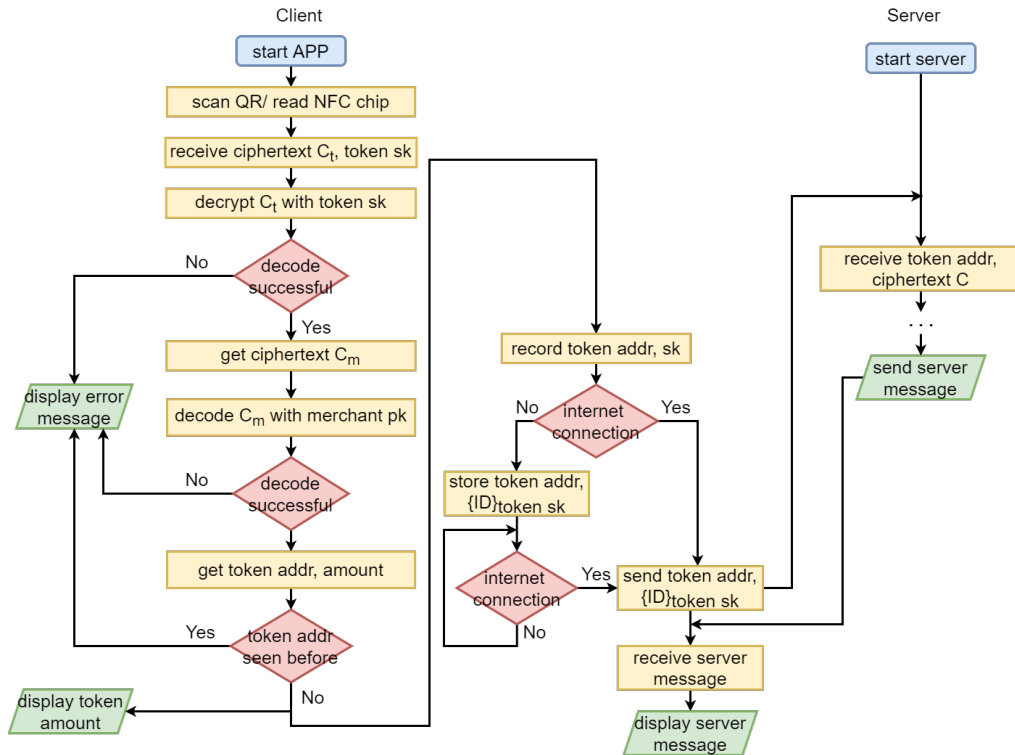


Figure 3.11: Merchant money receiving process

Prevent interception attack

Throughout the two-way conversion, the token private key, which is the key to receive the free money, is never shown on the Internet. The only two places where the private key stores are the generator phone and its physical carrier (QR code or NFC chip). This prevents online eavesdropping.

Prevent double-spending attack

Once free money is received by someone, this free money is immediately expired. This along with in advance balance deduction prevents the double-spending problem of this free money.

Forward/Backward security

Since all token/contract key pairs are only used one-time, stealing a key pair does not breach the free money security before or after. The only long-term key is the merchant key pairs, which could be improved by updating the key pairs regularly in the future.

App Implementation

The real App implementation adopts all the system design covered in Chapter 4. It also reduces user operation steps and makes the system intuitive as much as possible. The interface is simple since this App is still in the prototype stage and functionality is a more important aspect this project focuses on.

4.1 Printing interface

The App allows users to either generate and print new free/merchant money, or print a past free/merchant money which has been generated before and not expired.

Print new money

Users can print new free money or merchant money on the same page. Fig 4.1 illustrates the operation steps of printing new money. The printed QR part of free money is like Fig 4.2.

Print past free money

Users can print past free money whose token key pair and token address has been generated before. Users can choose a different printing mode than the previous one for this money. Fig 4.3 illustrates the operation steps of printing past free money.

Print past merchant money

Users can print past merchant money. However, users cannot change the merchant name. Fig 4.4 illustrates the operation steps of printing past merchant money.

4.2 Receiving interface

The App allows users to receive free money of any amount or request a specific amount of free money based on the contract-based mechanism. A merchant can receive merchant money after offline verification.

receive free money

Users can receive free money by directly scanning the QR code or/and reading an NFC chip. The App will automatically recognize whether an NFC part is required after scanning the QR part. For all-NFC mode, users should manually skip the QR scanning process. Fig 4.5 illustrates the operation steps of the free money receiving process.

request free money

Based on the contract-based mechanism, users can request a specific amount of free money and return the change to the payer after completing the transaction. Users can also cancel the transaction during the transaction, in which case all received money will be given back to the payer. Fig 4.6 illustrates the free money request process.

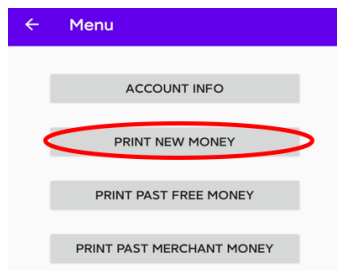
receive merchant money

A merchant user can verify whether the merchant money for him/her is valid/expired offline and receive the money when he/she is back online. Fig 4.7 illustrates this merchant money offline verification process.

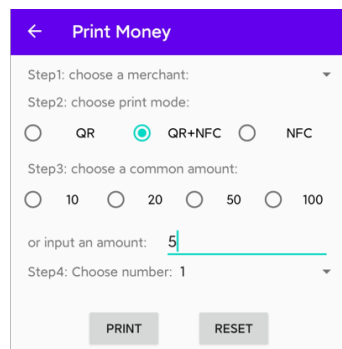
The merchant money can also be verified online. Fig 4.8 illustrates the operation steps.

4.3 Other interface

Users need to choose an account type when registering an account. The registration page is shown in Fig 4.9.



Step1: Click PRINT NEW MONEY



Step2: To print a free money, select the empty merchant, to print a merchant money, choose the desired merchant name

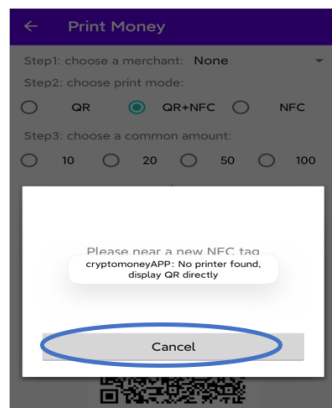
Step3: Choose a printing mode

Step4: Choose or input a valid amount

Step5: Choose printing number (default 1)

Step6: Click PRINT

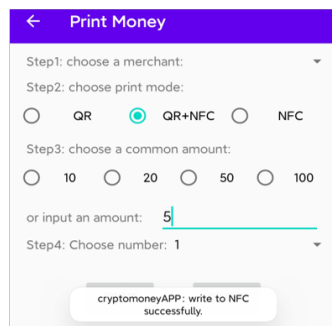
RESET: Clear all your choices in this page



Step6: Print/Display QR code (for QR and QR+NFC mode)

Step7: touch an NFC chip (for NFC and QR+NFC mode)

Cancel: Cancel writing NFC tag. Cannot revoke this free money generation, but can revoke later money if printing number > 1



Step8: Write to NFC successfully (for NFC and QR+NFC mode)

Figure 4.1: New money printing interface

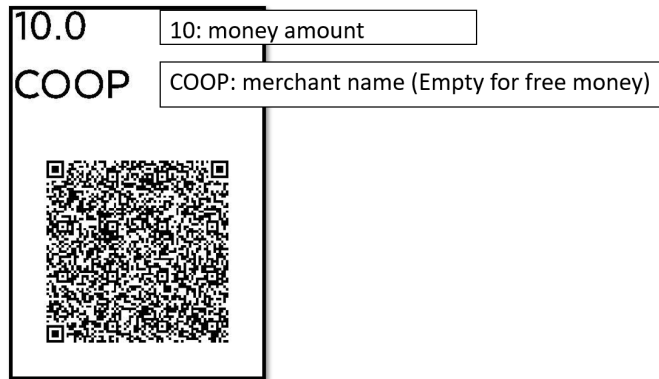
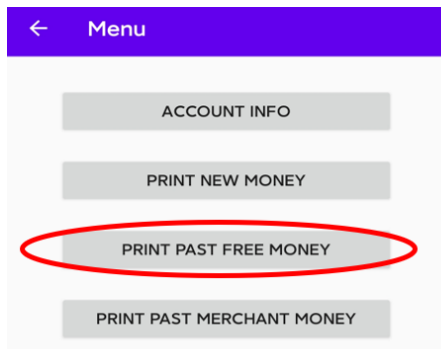
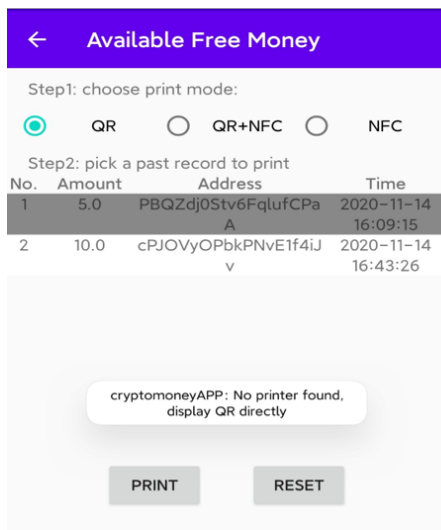


Figure 4.2: A merchant money example



Step1: Click PRINT PAST FREE MONEY

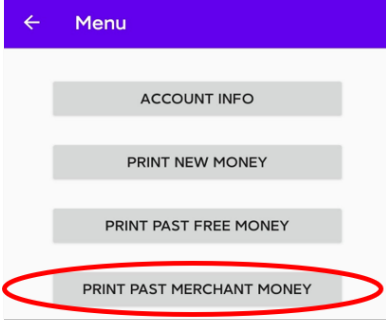
Step2: Choose a printing mode to print past free money



Step3: Choose a past free money record to print (pull down to refresh)
Amount: free money amount
Address: the token address of this money
Time: time when this money is generated

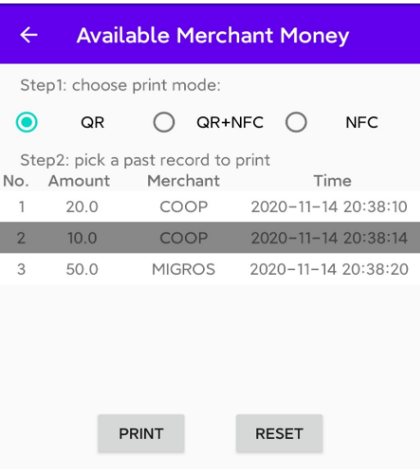
Step4: Click PRINT

Figure 4.3: Past free money printing interface



Step1: Click PRINT PAST MERCHANT MONEY

Step2: Choose a printing mode to print past merchant money



Step3: Choose a past merchant-specific money record to print (pull down to refresh)
Amount: merchant money value
Merchant: the specified merchant of this money
Time: time when this money is generated

Step4: Click PRINT

No.	Amount	Merchant	Time
1	20.0	COOP	2020-11-14 20:38:10
2	10.0	COOP	2020-11-14 20:38:14
3	50.0	MIGROS	2020-11-14 20:38:20

Figure 4.4: Merchant money printing interface

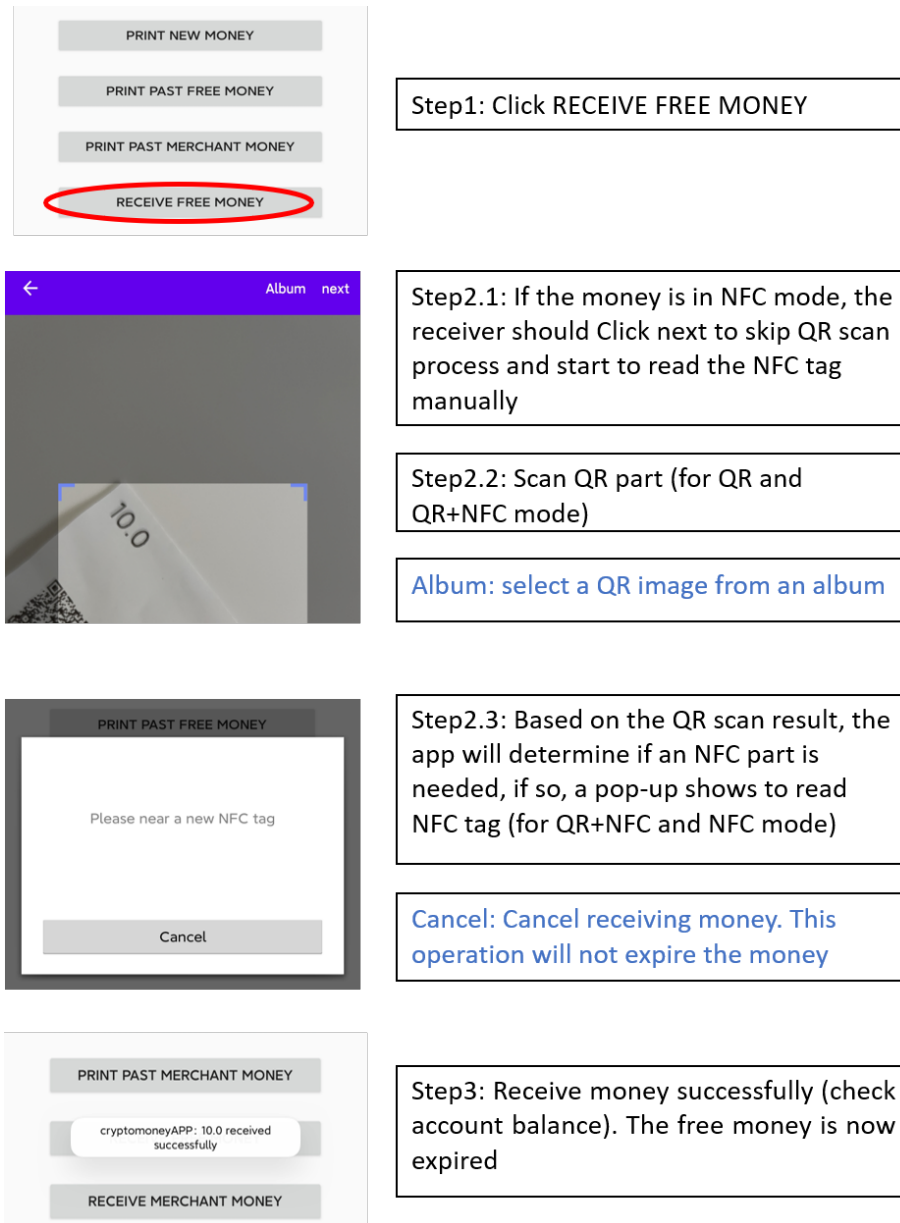


Figure 4.5: Past free money receiving interface

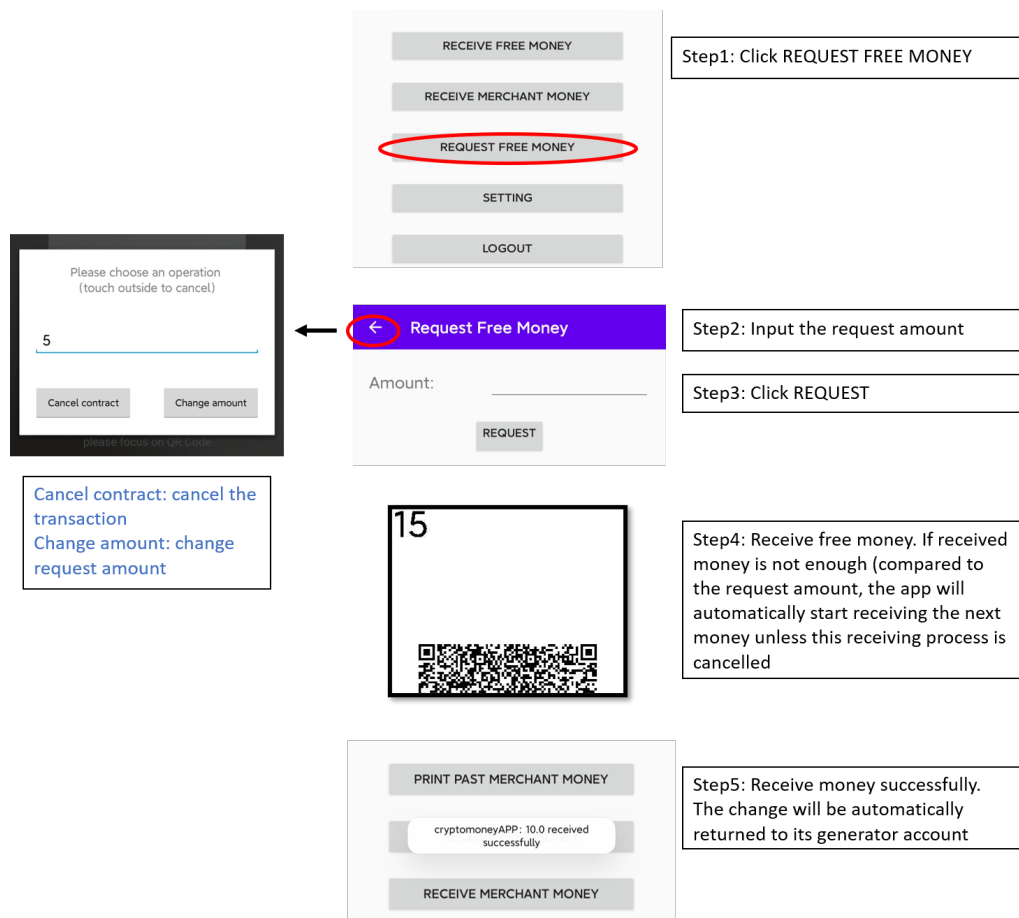


Figure 4.6: free money request interface

← cryptomoneyAPP

Account: COOP

Password:

Remember password

LOGIN REGISTER

Response:

Offline verification merchant: KFC

VERIFY

Response: amount= 20.0

PRINT PAST MERCHANT MONEY

cryptomoneyAPP: 20.0 received

REQUEST MONEY

Step1: Choose a merchant account to verify

Step2: Click VERIFY

Step3: Receive merchant money

Response: displays the verification result of the merchant money

Step4: Receive merchant money when logging in to the account next time

Figure 4.7: Merchant money offline verification interface

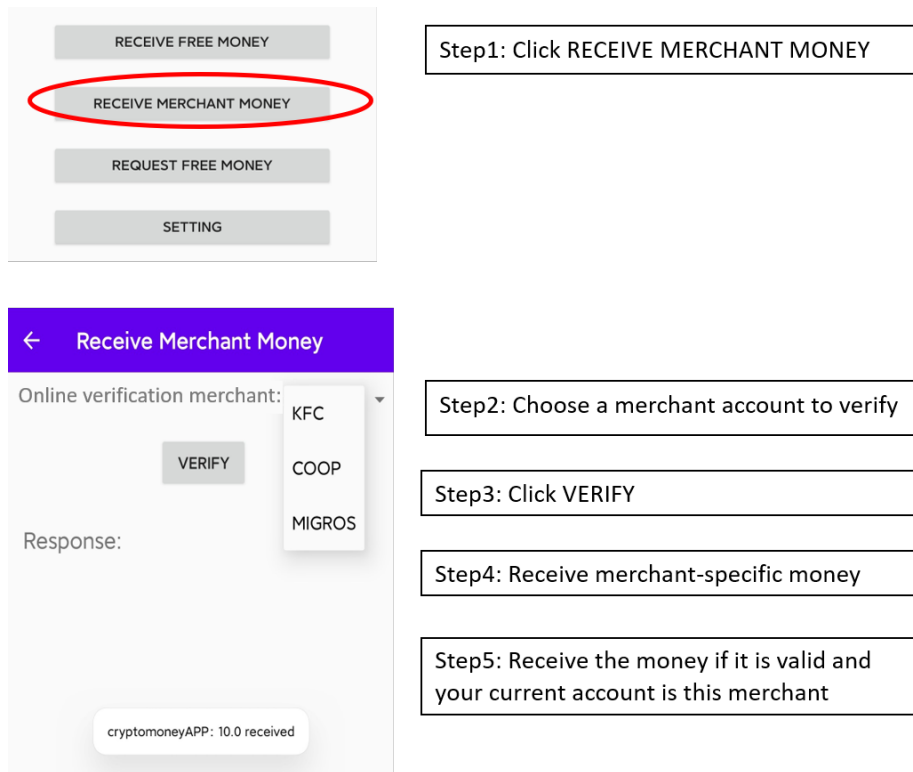


Figure 4.8: Merchant money online verification interface

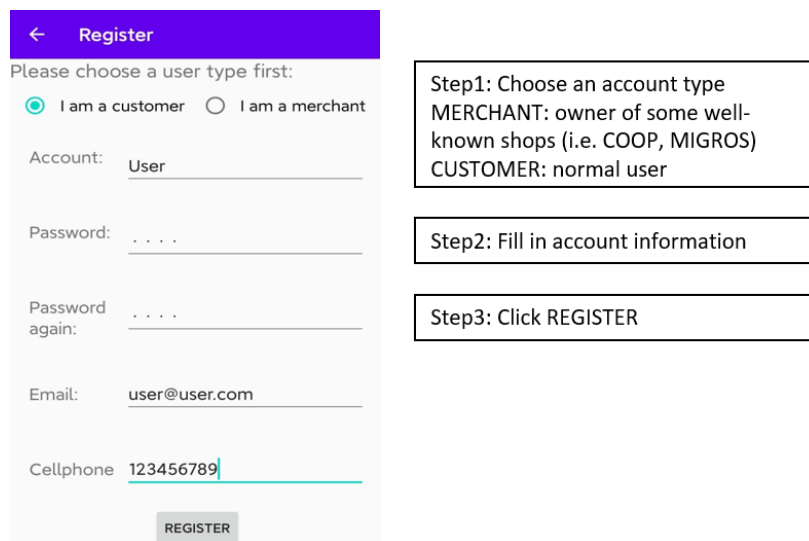


Figure 4.9: Registration interface

Experiment

A performance test and a user test are carried out in this project. The goal of the performance test is to compare and analyze the functionality and system response speed among different Android devices. The user test is to study users' acceptance of the system and the learning time taken to be familiar with the App. The test results are significant for future improvement.

5.1 Performance test

The performance test uses 4 Android phones of different Android versions released between 2015 and 2020. 7 tests are chosen to test the money printing and receiving speed in different modes. Each test is conducted six times to reduce the randomness impact.

5.1.1 Test results

The 7 tests are listed as follows. A MEMOBIRD Bluetooth printer is used to print QR codes. Due to the length of the free money content, the capacity of the NFC chip must be larger than 400 bytes. The project uses a 888 bytes NFC chip.

- **Test 1:** Print 5 5.0 free money in QR mode.
- **Test 2:** Print a 10.0 merchant money in QR+NFC mode
- **Test 3:** Receive a 5.0 free money generated in Test 1.
- **Test 4:** Request 18.0 free money, receive 4 5.0 free money generated in Test 1.
- **Test 5:** Offline verify/receive a 10.0 merchant money generated in Test 2.
- **Test 6:** Print a 5.0 merchant money in NFC mode.

- **Test 7:** Online receive a 5.0 merchant money generated in Test 6.

The tests above cover almost all main system functions. Fig 5.1 gives the complete performance test result. All results only include the necessary operation time related to the test itself. For instance, in Test 1, the timing is started from clicking the PRINT NEW MONEY button and ends at the last free money is printed out. The login time and the time to connect to the printer are not covered.

5.1.2 Result analysis

Although the QR code scanning speed and the NFC chip response speed are partially affected by the phone hardware, the result shows that on average the difference is not much obvious. There is some NFC response time difference between different phone models, which is mainly revealed in Test 6 and 7, but the difference is within the allowable range since the timing environment is not under very strict control.

The most significant difference appears in Test 6, where HUAWEI Mate 8 took 20 seconds more than the other 3 models to finish receiving 4 free money in all-QR mode. This result is largely caused by one abnormal data whose cell is highlighted in blue. This abnormal data is generated because the QR codes are complicated (due to the long length) and two QR codes are not clearly printed by the printer so that scanning takes a long time and sometimes the scanning result is invalid (in this case the App will re-scan the QR code without human interaction). Another abnormal data is highlighted in yellow. Redmi K30 Pro spends more than twice as long as normal on reading the free money stored in the NFC chip due to a slow Internet connection.

Test 1, 2, 6 test the printing speed of three printing modes. The results show that printing single money in all-QR mode takes around 8 seconds (39 seconds for 5 consequent money), compared with 16 seconds in QR+NFC mode and 12 seconds in all-NFC mode. Test 3, 4, 5, 7 test the receiving speed of different modes. The results show that all-QR and all-NFC modes have similar receiving speeds (around 8-10s). in QR+NFC mode the time is around 16 seconds since both a QR code and an NFC chip need to be received in this mode.

In summary, the following conclusion can be drawn from the performance test.

- all-QR and all-NFC mode have a similar printing/receiving speed, but the speed of all-NFC mode are more affected by the phone hardware.
- QR+NFC mode takes the most time to print/receive money, although it is also the most secure mode.
- Randomness is inevitable in all modes.

performance test (sec)	MI 10	HUAWEI P20	RedMI K30 Pro	HUAWEI Mate 8	avg	
test 1	ex1	37.33	40.06	39.54	40.48	39.35
	ex2	54.01	35.60	38.15	40.63	42.10
	ex3	40.65	33.38	37.13	46.18	39.34
	ex4	35.32	35.49	44.04	41.25	39.03
	ex5	42.24	33.43	38.03	36.27	37.49
	ex6	37.72	34.73	34.86	39.90	36.80
	avg	41.21	35.45	38.63	40.79	39.02
test 2	ex1	22.53	12.49	11.52	22.37	17.23
	ex2	15.84	13.43	15.85	13.14	14.57
	ex3	13.05	12.22	13.78	21.63	15.17
	ex4	10.83	24.28	25.04	13.24	18.35
	ex5	12.29	25.18	11.67	15.33	16.12
	ex6	14.39	20.52	10.48	19.59	16.25
	avg	14.82	18.02	14.72	17.55	16.28
test 3	ex1	8.98	8.46	12.20	9.55	9.80
	ex2	12.16	7.49	7.55	11.60	9.70
	ex3	13.09	8.82	7.84	9.33	9.77
	ex4	13.20	6.28	6.77	10.13	9.10
	ex5	8.96	15.56	21.34	13.72	14.90
	ex6	17.29	5.43	4.99	12.83	10.14
	avg	12.28	8.67	10.12	11.19	10.57
test 4	ex1	24.12	21.06	18.79	28.03	23.00
	ex2	31.63	23.11	23.91	26.41	26.27
	ex3	33.49	33.03	21.64	57.64	36.45
	ex4	38.61	30.50	39.67	47.20	39.00
	ex5	21.90	43.91	31.00	100.77	49.40
	ex6	19.65	19.69	16.94	20.25	19.13
	avg	28.23	28.55	25.33	46.72	32.21
test 5	ex1	11.18	12.11	12.62	27.59	15.88
	ex2	11.15	49.58	9.60	11.94	20.57
	ex3	12.13	13.64	10.30	13.07	12.29
	ex4	10.81	13.37	31.60	12.54	17.08
	ex5	19.02	14.55	13.66	16.46	15.92
	ex6	8.65	10.03	23.90	21.25	15.96
	avg	12.16	18.88	16.95	17.14	16.28
test 6	ex1	10.63	13.43	13.56	17.94	13.89
	ex2	18.28	8.88	11.00	16.38	13.64
	ex3	12.04	14.28	11.35	16.12	13.45
	ex4	9.56	19.28	13.91	25.35	17.03
	ex5	14.57	10.38	9.72	10.76	11.36
	ex6	9.61	13.05	11.42	10.75	11.21
	avg	12.45	13.22	11.83	16.22	13.43
test 7	ex1	6.50	8.88	8.36	9.72	8.37
	ex2	6.46	7.81	7.53	10.94	8.19
	ex3	5.33	14.10	10.12	7.9	9.36
	ex4	6.79	6.87	10.18	8.12	7.99
	ex5	7.80	5.78	9.66	10.35	8.40
	ex6	8.14	5.50	20.47	11.16	11.32
	avg	6.84	8.16	11.05	9.70	8.94

Figure 5.1: Performance test result

5.2 User test

Due to Covid-19, the user test is only performed among roommates. The test aims to investigate whether a new user could accept and be familiar with the new transaction system in a short time. This is mainly done by recording the time each user takes to complete the required tasks.

5.2.1 Test results

Three phases are included when testing each user: learning, training and testing. The specific process is explained as follows.

- **Learning:** A brief user guide is read by each user before operating the App. The user guide covers the motivation of the whole project and the main concepts related to the system such as free money and merchant money. The guide also illustrates the detailed operation instructions for printing and receiving money in different modes. The reading time for each user is recorded.
- **Training:** Before the formal test, each user walks through hands-on training. In this phase, the user will be taught to register a new account, connect to the printer and be familiar with the NFC operation. Each user is also asked to print and receive free/merchant money, following the user guide instructions. Any questions related to the system will also be answered. The goal of training is to eliminate bias impact and make sure all users have the same understanding of the system before the formal test.
- **Testing:** Each user is required to complete Test 1-5 listed in Section 6.1.1 without hint or assistance. The time for each test will be recorded with the same rules applied in the performance test. Problems that occurred during the tests will also be recorded.

Fig 5.2 gives the user test result. Fig 5.3 analyzes some common issues during the user test.

5.2.2 Result analysis

The comparison between Fig 5.1 and Fig 5.2 shows that after learning and training, a new user can finish each test within 1-2 times of the time used in the performance test, which can be regarded as the ideal time to finish each test. The most significant differences occur in Test 4 and Test 5, where the average user time is twice as much as the ideal time. As analyzed in Fig 5.3, the main reason is that despite being trained before the test, new users cannot find the best

pilot test	reading(min)	training(min)	test 1(sec)	test 2(sec)	test 3(sec)	test 4(sec)	test 5(sec)
user 1	4.27	12.21	42.12	19.06	10.43	67.07	17.60
user 2	8.23	7.51	54.91	25.52	6.59	21.40	37.90
user 3	4.54	10.43	65.05	25.45	6.07	44.86	27.89
user 4	2.47	13.55	39.20	11.91	20.29	98.12	36.46
user 5			43.22	46.62	11.40	87.60	48.66
user 6	8.21	10.51	46.40	19.05	6.52	48.85	24.83
avg	5.54	10.84	48.48	24.60	10.22	61.32	32.22

Figure 5.2: User test result

No	user test issues
①	QR scan slow (wrong angle, QR code not clear)
②	user not familiar with NFC operation
③	printer print duplicates sometimes
④	NFC read fails sometimes
⑤	write NFC after QR code printing

	test 1	test 2	test 3	test 4	test 5
user 1				①	
user 2	③	④			④
user 3	③	④		①	⑤
user 4			④	①④	④

Figure 5.3: User test issues

position to scan a bit complicated QR code or to communicate with an NFC chip, which also occurs sometimes in the performance test. It is interesting to find that the average user time in Test 3 is almost the ideal time (the data colored in green in two figures). This could be explained by simple probability. The probability for an expert user and a new user to successfully scan a single QR code is very close, but the probability of successive success decreases dramatically.

During the user test, some questions from users also help improve the project. For instance, several users are not fully understanding the merchant money. After further communication with users, it is found that using the voucher analogy is easier to be understood than explicitly introducing all functions of merchant money.

Conclusion

In summary, this project designs a novel transaction system, implements the system on the Android App and tests it. The main goal of the system is to provide a secure and convenient integration between physical and digital money, which is missing in today's most digital transaction environment. The system also realizes the fully unilateral offline and partially bilateral offline payment via asymmetric encryption. Besides the practical part, an investigation on the development of CBDC is also included in this project.

Due to time limitations, several improvements that could be considered in the future are listed as follows.

- **Fully bilateral offline:** The current system does not support change return or transaction cancellation when both sides are offline. Future implementation could refer to the practice DC/EP where a delayed verification is implemented.
- **Other money formats:** Chapter 6 has shown that using a QR code or an NFC chip could occur some inevitable recognition failure. A new materialization of free/merchant money may solve this problem.
- **TLS communication:** Currently, the communication between the client and the server is manually encrypted by the protocol. A TLS protocol could add another layer of security.

Bibliography

- [1] H. Shen, C. Faklaris, H. Jin, L. Dabbish, and J. I. Hong, “i can’t even buy apples if i don’t use mobile pay?” when mobile payments become infrastructural in china,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–26, 2020.
- [2] R. Ali, J. Barrdear, R. Clews, and J. Southgate, “Innovations in payment technologies and the emergence of digital currencies,” *Bank of England Quarterly Bulletin*, p. Q3, 2014.
- [3] C. Boar, H. Holden, and A. Wadsworth, “Impending arrival—a sequel to the survey on central bank digital currency,” *BIS Paper*, no. 107, 2020.
- [4] C. Barontini and H. Holden, “Proceeding with caution—a survey on central bank digital currency,” *Proceeding with Caution-A Survey on Central Bank Digital Currency (January 8, 2019)*. *BIS Paper*, no. 101, 2019.
- [5] R. Auer, G. Cornelli, J. Frost *et al.*, “Rise of the central bank digital currencies: drivers, approaches and technologies,” *Bank for International Settlements Working Papers*, no. 880, pp. 1–41, 2020.
- [6] C. M. Kahn and W. Roberds, “Why pay? an introduction to payments economics,” *Journal of Financial Intermediation*, vol. 18, no. 1, pp. 1–23, 2009.
- [7] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” Manubot, Tech. Rep., 2019.