Privacy-Preserving E-Wallet in Cloud-Era

Interim Report

Cheung Long Sang (3035692797)
Lee Hiu Long (3035687077)
Liao Jiayang (3035637280)

Supervisor: Dr KP Chow
Abstract

Privacy concerns have hitherto been a major barrier to the wide adoption of e-wallets in the market. Transactions on e-wallets may not be confidential. This undermines users’ financial privacy. As a result, there is an increasing interest in privacy coins. Nevertheless, while privacy coins can provide privacy through cryptographic techniques, they are not scalable and have an ambiguous legal status. They are still far from wide adoption. This project aims to develop a e-wallet for fiat money that is both privacy-preserving and scalable using cryptography. Cloud technology is also employed for better security, resilience and synchronism. Thus far, most functions have been designed and implemented, whilst a confidential transaction protocol based on integrated signature and homomorphic encryption, and non-interactive zero-knowledge proof has been proposed. Limitations of the protocol and the project in general are identified. Challenges and possible solutions are also discussed. In the next stage, we will continue implement the e-wallet app, integrate the protocol to the app and deploy the backend to the cloud server. The e-wallet has the potential to be the first privacy-preserving and the first cloud-based e-wallet in the market.

Keywords: E-wallet, Confidential transaction, Cryptography, Cloud technology, E-payment
Acknowledgment

Words cannot express my gratitude to our project supervisor and program coordinator, professor Chow Kam Pui, for his invaluable patience, continuous support, and inspiring feedback. This endeavor would not have been possible without the generous support from the HKU-SCF FinTech Academy.

We are also grateful to our classmates and cohort members, especially our team members, for the editing help, late-night feedback sessions, and moral support. Thanks should also go to the research assistants and study participants from the university, who provided critical data and inspired us.

Lastly, we would be remiss in not mentioning our family. Their belief in us has kept our spirits and motivation high during this process.
## Contents

List of Figures ........................................ 1
Abbreviations .......................................... 2

1 Introduction .......................................... 3
   1.1 Background and Motivation ...................... 3
   1.2 Objectives and Scope .......................... 4
   1.3 Outline of This Report ......................... 5

2 Market Survey .................................... 6

3 Literature Review .................................. 8

4 Methodology ....................................... 10
   4.1 Frontend Development .......................... 10
   4.2 Backend Development ........................... 11
   4.3 Cloud Technology ................................ 11
   4.4 QR Code Technology ............................ 12
   4.5 Confidential Transaction ....................... 12
   4.6 Novel Functions ................................ 14
      4.6.1 Promise ................................... 14
      4.6.2 Collaborative Wallets .................... 14
      4.6.3 Automatic Bill Splitting .................. 15

5 Current Progress ................................ 17
   5.1 Frontend ........................................ 17
      5.1.1 Basic Functions ........................... 17
      5.1.2 Novel Functions ............................ 19
   5.2 Backend ........................................ 21
      5.2.1 Entity Relationship Diagram ............... 21
      5.2.2 API Document ................................ 22
   5.3 Confidential Transaction ....................... 23
      5.3.1 Protocol ................................... 23
      5.3.2 Implementation ............................ 25
      5.3.3 Discussion .................................. 25
   5.4 Limitations ..................................... 26
      5.4.1 Ignorance of Pratical Issues ............... 26
      5.4.2 Compliance with Regulatory Requirements .. 26
   5.5 Difficulties and Possible Solutions ........... 27
6  Next Step  
7  Conclusion  
References  
Appendices  

A  Market Research Survey  
A.1  Original Survey and Raw Data  
A.2  Result Visualization  

B  API Documentation
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Expected improvement of e-wallets</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Demand for multi-devices (Market Survey)</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Attraction of privacy-preserving features in e-wallet adoption (Market Survey)</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>System Design</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Typical scenarios for QR code usecases</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Login UI</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Register UI</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>Terms of Service UI</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>Home UI</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>Sidebar UI</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>User Profile UI</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>Tx History UI</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>Scan UI</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>Payment Code UI</td>
<td>18</td>
</tr>
<tr>
<td>15</td>
<td>Wallets UI</td>
<td>19</td>
</tr>
<tr>
<td>16</td>
<td>Send UI</td>
<td>19</td>
</tr>
<tr>
<td>17</td>
<td>Receive UI</td>
<td>19</td>
</tr>
<tr>
<td>18</td>
<td>Top-Up UI</td>
<td>19</td>
</tr>
<tr>
<td>19</td>
<td>Withdrawal UI</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>Bill Payment UI</td>
<td>19</td>
</tr>
<tr>
<td>21</td>
<td>Promise UI</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>Co-wallets UI</td>
<td>20</td>
</tr>
<tr>
<td>23</td>
<td>Co-wallet Creation UI</td>
<td>21</td>
</tr>
<tr>
<td>24</td>
<td>Co-wallet Joining UI</td>
<td>21</td>
</tr>
<tr>
<td>25</td>
<td>Database ER Diagram</td>
<td>22</td>
</tr>
<tr>
<td>26</td>
<td>Confidential Transaction Phase</td>
<td>24</td>
</tr>
<tr>
<td>27</td>
<td>Project Schedule</td>
<td>28</td>
</tr>
</tbody>
</table>
**Abbreviations**

ACP  
**Auditable Confidential Payment**

ADCP  
**Auditable Decentralized Confidential Payment**

AES  
**Advanced Encryption Standard**

c-o-wallet  
**collaborative wallet**

DCR  
**decisional composite residuosity**

DDH  
**decisional Diffie–Hellman**

DLIN  
**Decision Linear**

e-payment  
**electronic payment**

e-wallet  
**electronic wallet**

ISHE  
**integrated signature and homomorphic encryption**

OCR  
**Optical Character Recognition**

NIZK  
**non-interactive zero-knowledge proof**

P2P  
**peer-to-peer**

PPT  
**probabilistic polynomial-time**

q-SDH  
**q-Strong Bilinear Diffie-Hellman**

UTXO  
**unspent transaction output**

zk-SNARK  
**zero-knowledge succinct non-interactive argument of knowledge**
1 Introduction

1.1 Background and Motivation

Electronic wallets (e-wallets), also known as digital wallets or mobile wallets, are mobile applications that allow users to make both online and in-store payments [1], using fiat money deposited therein beforehand\(^1\). Examples include Alipay, WeChat Pay, PayMe, Tap&Go and BoC Pay. Some e-wallets additionally support peer-to-peer (P2P) payment, in which money can be transferred from one wallet to another. They enable users to make instant payments anytime anywhere without transaction fees [3]. In light of such convenience, the portion of Hong Kong people owning an e-wallet surged from 65% in 2017 to 91% in 2020 according to a survey conducted in 2020 [4]. With the tremendous growth of the local e-wallet market in recent years, it is anticipated that e-wallets will become the most common e-payment method in Hong Kong by 2025 [5].

Having said that, Yu [1] argues that there are still barriers to wide adoption. Of particular interest is privacy concerns. Transaction data, such as transaction history and the value of each transaction, is visible to e-wallet providers. They can see every transaction that is made through their e-wallets. Although personal data usage is regulated by the Personal Data (Privacy) Ordinance in Hong Kong, such protection is merely legal but not inherent. That is to say, the ordinance can only deter but not prevent e-wallet providers from abusing user financial data. Users have to trade off their financial privacy to centralized parties for electronic payment (e-payment) services.

On the other side of the e-payment ecosystem, privacy coins, or privacy-preserving cryptocurrencies\(^2\), such as Zcash and Monero, provide transaction confidentiality and anonymity through cryptographic techniques, such as zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) and ring signatures [6], [7]. Nevertheless, due to their decentralized nature, they are not scalable, i.e. having much higher transaction fees and much lower transaction speeds than their centralized counterparts [8]. Moreover, the difficulty in auditing them for regulatory purposes has caused their ambiguous legal status [9]. Consequently, while most consumers still prefer fiat money, privacy coins are far from wide adoption.

\(^1\)The term “e-wallets” are mostly used to refer to stored-value ones that are of interest of this project. For non- stored-value ones, unlike their store-valued counterparts in which deposits work similar as bank deposits, only payment credentials and passwords are stored [2].

\(^2\)Cryptocurrencies, in general, do not offer actual privacy by nature. Most cryptocurrencies, such as Bitcoin and Ether, merely provide pseudonymity.
A natural question is whether we can enjoy privacy without forgoing scalability and auditability. This motivates us to apply several cryptographic techniques used in privacy coins, including integrated signature and homomorphic encryption (ISHE) and non-interactive zero-knowledge (NIZK) proof, to the centralized architecture of e-wallets to enhance its privacy-preserving capability whilst maintaining scalability for practical use and auditability for regulatory requirements.

Moreover, cloud technology is leveraged to strengthen the security and resilience of the e-wallet as well as provide seamless synchronization across multiple devices. To the best of our knowledge, heretofore no existing e-wallet in the market is based on cloud technology.

1.2 Objectives and Scope

This project aims to develop a fully-functional mobile app of a privacy-preserving cloud-based e-wallet, named SuperCloudPay, to enhance consumer e-payment experience in terms of privacy, security and convenience while ensuring validity and allowing auditability. The prototype will include basic functions of an e-wallet, including

1. Online payment
2. In-store payment
3. P2P transfer
4. Top-up
5. Withdrawal
6. Bill payment

as well as novel functions, including:

1. Promise: automatic payment by conditions (see Section 4.6.1)
2. Collaborative Wallets: for groups using decentralised consensus (see Section 4.6.2)
3. Automatic Bill Splitting: automatic bill splitting by OCR (see Section 4.6.3)

and unique features, including confidential transaction, and 2) seamless synchronization.

1. Confidential transaction: end-to-end encryption for wallet and transaction information (see Section 4.5)
2. Seamless synchronisation across multiple devices: users can access SuperCloudPay from any number of devices (such as mobile phones, tablets, computers, smart-watches) simultaneously (see Section 4.3)
1.3 Outline of This Report

The remainder of this report is structured as follows. Chapter 2 analyzes the results of a market survey conducted on e-wallets. Chapter 3 provides a literature review on confidential transaction. Chapter 4 expounds the methodology used for developing the frontend and backend systems as well as unique features of confidential transaction and seamless synchronisation, which includes several cryptographic techniques and cloud technology. Designs of novel functions are also presented. Chapter 5 presents the current progress of the project, which includes implementation of basic and novel functions, backend documentation, and a confidential transaction protocol, and discusses the limitations of the protocol and the app in general, as well as challenges and possible solutions. Chapter 6 shows the planning for future work. Chapter 7 concludes the report.
2 Market Survey

This chapter analyses the motivation and significance of this project from the market point of view.

Recent research on the age of e-wallet users [10] implied that over 80% of e-wallet users are under 30, indicating the primary customer base of e-wallets is young adults. Given the geographical context of the project, the market survey was launched online using Google form (English version) and Tencent questionnaire (Chinese version) in August 2022, with 274 responses collected from Hong Kong university students to explore the demand and possible improvement of e-wallets from the perspective of young people. The survey questions discussed the necessity of introducing privacy protection (Section 4.5) and seamless synchronization (Section 4.3) to e-wallets. The survey result will be visualized and used to support the system design and application implementation.

<table>
<thead>
<tr>
<th>Expected improvements</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy/Security</td>
<td>32.3%</td>
</tr>
<tr>
<td>Withdrawal transaction fee</td>
<td>9.7%</td>
</tr>
<tr>
<td>Internet connection</td>
<td>3.2%</td>
</tr>
<tr>
<td>Convenience</td>
<td>9.7%</td>
</tr>
<tr>
<td>Payment amount limit</td>
<td>12.9%</td>
</tr>
<tr>
<td>Others</td>
<td>32.3%</td>
</tr>
</tbody>
</table>

Figure 1: Expected improvement of e-wallets

From Figure 1, around one-third of university e-wallet users expect improvements in privacy and security aspects, indicating a growing awareness of financial privacy. This shows that privacy protection became a critical concern of e-wallets’ future development. In the meantime, over half of users were found to expect to use e-wallets across multiple devices (Figure 2).
On the other hand, around 65% of survey respondents much valued the privacy-preserving feature and the respondents were possible to adopt other e-wallets for better privacy protection (Figure 3).

The links to complete survey, the raw data responses and result visualization are detailed in Appendix A.
3 Literature Review

This section provides a brief literature review on existing methods for the design of a confidential transaction protocol.

The term “confidential transaction” was coined by Maxwell in 2016 [11]. A confidential transaction is a transaction that shields the transaction amount from parties other than the sender and receiver(s) in a way that its validity is still verifiable.

Existing literature tends to focus on the protocols for blockchains [12]. In other words, the underlying architecture is decentralized. As a result, adjustments are required to adapt the results to a centralized architecture.

Meanwhile, most foundational works such as [6], [7], [11] adopt an unspent-transaction-output-based (UTXO-based) model. A new line of research has used an account-based model. Since all the existing e-wallets (for fiat money) adopt an account-based model (due to its accountability, auditability and traceability for regulatory, supervisory and monitoring purposes [13]), we chose to study the ones that adopt an account-based model. Here are some examples of such works.

[14] proposes a confidential transaction protocol on Ethereum via Ethereum smart contract. It harnesses sigma-bulletproofs (proposed in the same paper) to achieve confidentiality. For instantiation, it uses El Gamal encryption scheme with the decisional Diffie–Hellman (DDH) assumption as the underlying harness assumption.

[15] proposes an efficient NIZK scheme for confidential transaction using additive-homomorphic encryption. Its security is guaranteed under the Decision Linear (DLIN) and $q$-Strong Bilinear Diffie-Hellman ($q$-SDH) assumptions.

[16] improves the practicality of [14] and [15] by proposing an NIZK scheme for confidential transaction using Paillier Encryption secure under the decisional composite residuosity (DCR) and $q$-SDH assumptions.

[17] proposes an account-based auditable confidential transaction protocol that supports after-the-fact auditing (including regulation compliance and global supervision) which builds on blockchain. This work won the first prize in 2020 Financial Cryptography contest in China.

The main idea underlying [14]–[17] is as follows:
1. User account balances and transaction amount are encrypted using additive homomorphic encryption (with Encryption algorithm denoted as $\text{Enc}$ and decryption algorithm denoted as $\text{Dec}$) under the same key. For instance, let AliceBalance and alicepk be the account balance and public key of Alice respectively. Similarly we have BobBalance and bobpk. Then, AliceBalance and BobBalance are encrypted as $A := \text{Enc}_{\text{alicepk}}(\text{AliceBalance})$ and $B := \text{Enc}_{\text{bobpk}}(\text{BobBalance})$ respectively. Meanwhile, suppose TxAmount denotes the transaction amount. Then Alice computes $a = \text{Enc}_{\text{alicepk}}(\text{TxAmount})$ while Bob computes $b = \text{Enc}_{\text{bobpk}}(\text{TxAmount})$.

2. Generate the following zero-knowledge proofs:

   (a) The encrypted transaction amounts indeed encrypt the same value by sigma / schnorr’s protocol. For instance, $\text{Dec}_{\text{alice}}(a) = \text{Dec}_{\text{bob}}(b)$ where alicesk and bobsk denote the secret keys of Alice and Bob respectively.

   (b) The encrypted transaction amount $t$ is positive by zero-knowldege range proof.

   (c) The sender balance is non-negative after the transaction. For instance, $\text{AliceNewBalance} := \text{Dec}_{\text{alice}}(A')$ is non-negative where $A' = A - a = \text{Enc}_{\text{alicepk}}(\text{AliceBalance} - \text{TxAmount})$ (equality by correctness of additive homomorphic encryption) by zero-knowldege range proof.

After discreet consideration, this project adopts a variant of the Auditable Decentralized Confidential Payment (ADCP) system proposed in [17] with necessary conversions (from a decentralized architecture to a centralized architecture) for its conceptual simplicity and practical functionality. It is an account-based auditable confidential transaction system that supports after-the-fact auditing for regulatory compliance and global supervision. This echoes the project objective of allowing the transactions to be auditable.
4 Methodology

Figure 4 provides an overview of the system design. Our system is segregated into the frontend and the backend, among which the frontend is responsible for displaying the user interface and allowing the users to interact with the app (local functionalities), while Backend facilitates the database management in cloud servers. The frontend invokes the backend by sending HTTP requests to the backend when it needs to retrieve or modify the data in the database. The backend will then handle the requests, perform the required actions to the database and replies with the results. The communication between the frontend and the backend is under standard security measures. On top of the traditional system design, SuperCloudPay enables the privacy-preserving feature by implementing cloud computing on encrypted data. The details of the system and app design will be presented as follows: Sections 4.1 and 4.2 discuss the technology stack for frontend and backend developments respectively. Section 4.3 and Section 4.4 respectively discuss the use of cloud and QR code technology to enhance the e-wallet capabilities and provide better user experience. Then, Section 4.5 defines the system to be used in designing our protocol and discusses the cryptographic techniques used to construct the system. Finally, novel functions of SuperCloudPay will be introduced in Section 4.6

4.1 Frontend Development

Frontend development refers to the development of the client side of the application. With the frontend, the clients can handle the data through the user interface easily so that they do not need to communicate with the backend directly.

After comparing several native development tools such as Swift and Java and cross-platform development tools including Flutter and React Native, we have chosen React
Native as our development tool, as cross-platform frameworks require lower development costs. In addition to the fact that React Native is more mature than Flutter and our development team is more fluent in JavaScript than Dart.

Expo Cli is used for building the project as it allows the developer to run the app before building the project so that the developer can test the app more efficiently. Though React Native Cli provides more IOS and Android APIs, our app does not use those APIs that are not provided in Expo, Expo is still preferred.

4.2 Backend Development

The backend consists of the database and the server for managing the data and allowing the client to access the data conveniently and securely.

PostgreSQL, which is a relational database management system is used, as referential integrity can be achieved by the key constraints in the relational database.

Restful API endpoints are designed as routes for the users to communicate with the server. The endpoints are standardized, and they provide many functionalities like registration, login, and making transactions.

The backend is implemented using Django, as developers can develop the backend rapidly with the mature libraries provided by Django. It also provides many features to enhance security like Cross-site scripting (XSS) protection which can prevent injections of code from the client side. There are also libraries for authentication and atomic requests for Django so that authentication and database locking can be implemented easily.

Token authentication is used where a token is generated at random with a time limit when the user logins with their username and password. The server uses the token for authentication so that the users can only access the authorized data.

4.3 Cloud Technology

Our e-wallet adopts a cloud-based system for the backend. AWS Cloud will be used for the cloud server for its reliability, scalability and cost-effectiveness [18]. Harnessing cloud technology, our e-wallet app can be seamlessly synchronized with and accessed on multiple devices (such as smartphones, tablets, laptops, desktops and smartwatches) simultaneously. With data stored in the cloud, there is no fear of losing data due to the
loss of devices or the destruction of physical servers. This strengthens the security and resilience of the e-wallet [19].

4.4 QR Code Technology

In order to facilitate the in-store payment and P2P transfer in e-wallets, SuperCloudPay adopts QR code technology to standardize and streamline the process. First, in SuperCloudPay, each wallet possesses a unique wallet address, and a wallet could be linked to one or more users (such as a collaborative wallet, see Section 4.6.2). Hence, a generalized solution is to generate a QR code based on the wallet address, instead of user-related information. Second, for in-store payment scenario, the QR code is used to pay a bill, which would be scanned by an external terminal. Thus, this QR code should be a dynamic code and would be expired after a short period of time for security concerns. Considering both one-time and short expiration requirements, the “payer” QR code would be created based on both wallet address and a random number related to time of code generation and wallet balances. Typically, for P2P transfer scenario, the payer would be the one to scan and the payee would be the one to display the QR code. Since the QR code is used to receive payments, it could be static and directly generated from the wallet address. Other typical examples are included in Figure 5. In real implementation, the wallet address is assigned to each wallet in creation, and information of dynamic QR code would be computed before each request in the local device for security reasons. Then, a QR code would be generated locally by utilizing a mature react native library named react-native-qrcode-svg.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Payer</th>
<th>Payee</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-store Payment</td>
<td>QR Code/Bar Code (Dynamic)</td>
<td>Scanning Terminal</td>
</tr>
<tr>
<td></td>
<td>Mobile Camera</td>
<td>QR Code (Static)</td>
</tr>
<tr>
<td>Online Payment</td>
<td>Mobile Camera</td>
<td>QR Code (Dynamic)</td>
</tr>
<tr>
<td>P2P Payment</td>
<td>QR Code (Dynamic)</td>
<td>QR Code (Dynamic)</td>
</tr>
<tr>
<td></td>
<td>Mobile Camera</td>
<td>Mobile Camera</td>
</tr>
</tbody>
</table>

Figure 5: Typical scenarios for QR code usecases

4.5 Confidential Transaction

Our project adopts an Auditable Confidential Payment (ACP) System, a variant of the ADCP system defined, constructed and instantiated in [17], for our confidential transaction protocol.

An ACP system is formally defined as follows:
Definition 4.1 (Auditable Confidential Payment System). An Auditable Confidential Payment (ACP) system is a 9-tuple (Setup, CreateWallet, RevealBalance, CreateCTx, VerifyCTx, UpdateCTx, JustifyCTx, AuditCTx, OpenCTx) of probabilistic polynomial-time (PPT) algorithms such that

- \((pp, sp) \leftarrow \text{Setup}(1^\lambda)\): The setup algorithm Setup ran by the server takes as input a security parameter \(1^\lambda\), outputs a public parameter \(pp\) and a secret parameter \(sp\).
- \((pk, sk, \hat{C}) \leftarrow \text{CreateWallet}(\hat{v}, sn)\): The wallet creation algorithm CreateWallet ran by a user takes as input an initial balance \(\hat{v}\) and a serial number \(sn\), outputs a public key \(pk\), a secret key \(sk\) and an encrypted balance \(\hat{C}\).
- \(\hat{v} := \text{RevealBalance}(sk, \hat{C})\): The balance reveal algorithm RevealBalance ran by a user takes as input a secret key \(sk\) and an encrypted balance \(\hat{C}\), outputs the plaintext balance \(\hat{v}\).
- \(ctx \leftarrow \text{CreateCTx}(sk_s, pk_s, pk_r, v)\): The confidential transaction creation algorithm CreateCTx ran by a sender takes as input a sender secret key \(sk_s\), a sender public key \(pk_s\), a receiver public key \(pk_r\), and a transaction amount \(v\), outputs a confidential transaction \(ctx\).
- \(bit \leftarrow \text{VerifyCTx}(ctx)\): The confidential transaction verification algorithm VerifyCTx ran by the server takes as input a confidential transaction \(ctx\), outputs 0 if it is valid, 1 otherwise.
- \(\text{UpdateCTx}(ctx)\): The balance update algorithm UpdateCTx ran by the server takes as input a confidential transaction \(ctx\), updates the sender and receiver balances accordingly.
- \(\pi \leftarrow \text{JustifyCTx}(pk, sk, \{ctx\}, f)\): The confidential transaction justification algorithm ran by a user takes as input a public key \(pk\), a secret key \(sk\), and a set of confidential transactions \(\{ctx\}\) that it participated in and a policy \(f\), outputs a proof \(\pi\) for \(f(pk, \{ctx\}) = 1\) for auditing.
- \(bit \leftarrow \text{AuditCTx}(pk, \{ctx\}, f, \pi)\): The confidential transaction audit algorithm ran by an auditor takes as input a public key \(pk\), a set of confidential transactions \(\{ctx\}\), a policy \(f\) and a proof \(\pi\), outputs 0 denoting accept or 1 denoting reject.
- \(v := \text{OpenCTx}(sp, ctx)\): The confidential transaction open algorithm ran by a supervisor takes as input a secret parameter \(sp\) and a confidential transaction \(ctx\), outputs the plaintext transaction amount \(v\).

The correctness and security model of an ACP system follow from those of an ADCP system by removing requirements associated with decentralized functionalities.
An ISHE scheme and NIZK proof systems are used for construction. Informally, an ISHE scheme is a scheme that combines a digital signature scheme and a homomorphic (public-key) encryption scheme such that a single key pair is used in both signature (for authenticity) and encryption (for confidentiality). Meanwhile, an NIZK system can be informally described as a system allowing a prover to prove a statement to a verifier without leaking any information and interacting with the verifier. Exploiting these two cryptographic objects, the validity of a transaction can be verified without revealing the transaction amount and user wallet balances.

The instantiation in [17], which utilizes Twisted El-Gamal Encryption, is also used in our project.

4.6 Novel Functions

4.6.1 Promise

A wallet can make a 'promise' to another wallet by specifying the payment amount and payment conditions in the 'promise'. The result is a logical AND of payment conditions. An example of a promise would be "Alice to Bob, HKD 500, after 2023-02-08 and when Alice has received HKD 1000 from Cathy". Upon 'Make Promise', the payment logic is checked for satisfiability such that the promise is successfully made if and only the formula is Boolean satisfiable.

4.6.2 Collaborative Wallets

Besides holding one personal wallet, SuperCloudPay users can hold multiple collaborative wallets (co-wallets). Collaborative wallets utilise a decentralised consensus logic called 'M out of N authorisation' for transaction authorisation. Cloud technology is used for such a collaboration.

*M out of N authorisation.* In a collaborative wallet, whilst deposit does not require authorisation, usage and withdrawal of money requires the consent from a predefined number of users (hereinafter referred to as ‘M out of N consensus’). Once the consent condition is satisfied, the money can then be spent or withdrawn. The condition is a logical OR of a set of logical statements specifying the number of consents required from a set of users. For instance, A creates a shared wallet for himself as well as B, C, D, E, F, G, H and I, with consensus logic "2 out of \{A,B,C,E,G\} or 1 out of \{D,E\} or 3 out of \{F,G,H,I\}". All of them can deposit money into the shared wallet. Only D and E can use the money inside the shared wallet without asking for consents from other users. For B, it has to seek consents from A,C,E,G. For H, it has to seek consents from...
any 2 users from F, G and I. Besides "M out of a set of users", for user convenience, there are some predefined logical statements that can be used by a simple click, such as "\( \text{math.ceil}(N/2) \) out of \( N \)\), meaning that consents from more than half users are required to reach the consensus. For example, "1 out of \( \{A\} \) or \( \text{math.ceil}(N/2) \) out of \( N \)" is a valid consensus logic. Only A can use the money without seeking for consents. For other users, besides themselves, they have to seek consents from 4 other users out of the 9 users including A to use the money. The consent condition can be modified by the wallet creator (i.e. the user who creates the wallet) to be approved by a \( M \) out of \( N \) consensus.

There are two types of collaborative wallets – shared wallets and organisation wallets.

**Shared Wallet.** Shared wallets are typically for family, friends and classmates. Unverified wallets have a user limit of 50, a balance limit of $5,000 and an annual transaction limit of $25,000. Verified wallets enjoy unlimited wallet balance and annual transaction limit. To verify a wallet, specify a verified user as the representative. The KYC information of the representative is then binded with the wallet. The representative is obliged to bear legal responsibility for any money laundering or illicit activities associated with the wallet.

**Organisation Wallet.** Organisation wallets are typically for clubs, SMEs and community groups. While there is no user limit, balance limit and annual transaction limit, they require identity verification. KYC documents required include the registration documents. The organisation is obliged to bear legal responsibility for any money laundering or illicit activities associated with the wallet. There are three roles in an organisation wallet – creator, administrator and participants. Administrators are appointed by the creator. They enjoy the same privileges as the creator except the right to appoint administrators. There are two types of organisation wallet – private wallet and public wallet. In private wallets, transactions and wallet balance are encrypted as in personal wallets and shared wallets. In public wallets, transaction records and wallet balances are revealed to the public. SuperCloudPay users can join a public wallet by invitation or via an invitation link. When joining the wallet, a user can choose to reveal its identity (username or phone number) or hide. The consensus process is private, i.e. the voting is anonymous.

### 4.6.3 Automatic Bill Splitting

Upon payment, a wallet can initiate a bill-splitting request with its contacts or other wallets by selecting them from its contact list, or specifying their wallet addresses, or scanning (one-time) QR Codes (see Section 4.4). The payment is made and confirmed
if and only if every participant of the splitting has paid. A transaction is regarded as failed if it is not confirmed within 15 minutes. The amount paid is refunded afterwards. The bill-splitting request can be made both manually and automatically. The former is done by inputting the amount of each bill-splitting participant manually. The latter is done by first scanning the receipt of the meal so that each dish and its associated price is automatically recognized by Optical Character Recognition (OCR), then selecting the corresponding wallets manually. Service fee can also be considered in the bill-splitting request by one click.
5 Current Progress

The overall progress is deemed satisfactory. Every task is on track. To date, background research, market survey and a literature review on confidential transactions have been conducted. They are presented in Chapter 1, Chapter 2 and Chapter 3 respectively. For frontend, basic and novel functions have been designed and implemented. They are shown in Section 5.1. For backend, an Entity Relationship Diagram and API document have been completed and presented in Section 5.2. In addition, a confidential transaction protocol has been devised and implemented. The protocol is presented and discussed in Section 5.3. After that, limitations of the protocol as well as the project in general are identified in Section 5.4. Difficulties in the deployment of app into cloud server will be highlighted with a possible solution in Section 5.5.

5.1 Frontend

5.1.1 Basic Functions

Upon entering SuperCloudPay, the login page (Figure 6) is shown. If the user has not registered for a SuperCloudPay account, it will have to browse to the register page (Figure 7) prior to login to the app. A successful registration must involve a valid username, password, phone number (with verification) and an assent to the terms of service (Figure 8).

![Figure 6: Login UI](image1)
![Figure 7: Register UI](image2)
![Figure 8: Terms of Service UI](image3)

After successfully logging in to the app, the homepage (Figure 9) will be displayed. The homepage provides an entrance to many functions. First, the icon on the top left corner
can be clicked to show the sidebar (Figure 10) in which some wallet information and a navigation are shown. For instance, when the user clicks ”My Profile”, it will navigate to the user profile page (Figure 11) where user account information and a wallet dashboard are shown.

The user can return to the homepage via the bottom navigation bar. In fact, the bottom navigation bar provides navigation to the transaction history (Figure 12), scan (Figure 13), payment code (Figure 14) and wallet dashboard pages (Figure 15).
From the small buttons in the middle of the homepage (Figure 9), users can navigate to the pages of other basic functions, namely Send (Figure 16), Receive (Figure 17), Top-Up (Figure 18), Withdrawal (Figure 19) and Bill Payment (Figure 20).

5.1.2 Novel Functions

The last two functions to demonstrate are two of our novel functions – Promise and Collaborative Wallet, whose descriptions are detailed in Section 4.6.1 and Section 4.6.2.
On the collaborative wallet page, by clicking the first and second bottom buttons, one can navigate to Co-wallet Creation (Figure 23) and Co-wallet Joining (Figure 24) pages respectively.
5.2 Backend

5.2.1 Entity Relationship Diagram

Figure 25 is the entity relationship diagram illustrating the relationships, attributes, and cardinality of the entities in our database. It is drawn in a hybrid of Chen notation style, Crow’s Foot style, and some self-defined style.

Rectangles are entity sets, diamonds are relationship sets, double rectangles are weak entity sets, double diamonds are weak relationship sets, inverted triangles represent specializations, ellipses represent attributes of relationship sets, underline attributes are primary keys, directed lines represent mapping cardinality one, undirected lines represent mapping cardinality many, double lines represent total participation and single line represent partial participation.

As the diagram is a bit complex, this subsection explains some important parts only. As shown in Fig 25, the transaction entity has attributes id, datetime, amount, and type, where id is the primary key. The double line and the arrow connecting the transaction entity, from relationship and wallet entity, signify that a transaction must be initiated by
a wallet, while the single line means a wallet can participate in zero or more transactions.

As a condition number is unique among all conditions in a promise but not unique among all conditions, we can uniquely identify a condition by the condition number together with the corresponding promise id. This entity is a weak entity with a weak relationship with promise. A wallet must be a personal wallet or a collaborative wallet but not both. This is represented by a disjoint specialization block in the ER Diagram.

5.2.2 API Document

An API document has been completed and attached in Appendix B. The API Document lists all the endpoints provided by the Django server. The frontend can send HTTPS requests to the endpoints with the method, params, and body specified in the API documentation to do the desired action.

For example, the frontend can register an account by sending a request to the endpoint `/user/` with a Post method and a body consisting of the username and password.
The server will then respond with a body describing the user id and username. All the parameters, request body, and response body are in JSON format.

5.3 Confidential Transaction

5.3.1 Protocol

Our confidential transaction protocol adopts an Auditable Confidential Payment (ACP) system (Setup, CreateWallet, RevealBalance, CreateCTx, VerifyCTx, UpdateCTx, JustifyCTx, AuditCTx, OpenCTx). For the definitions, constructions and instantiations of each algorithm, please refer to Chapter 4.5. There are three main roles in the protocol, including Sender, Receiver and Cloud. Sender refers to any sender who would like to initiate a transaction. Receiver refers to the receiver(s) of the transaction. Cloud refers to the back-end server built on cloud. The protocol consists of four phases, namely, Setup, Wallet Creation, Balance Checking and Confidential Transaction. The procedures for each phase is as follows:

**Setup.** The Setup phase is entered once only when we initiate our e-wallet system. It consists of two steps. First, the setup algorithm Setup is ran by the server to obtain a set of parameters for the whole system. Then, some initializations are done to optimize the efficiency of the system.

**Wallet Creation.** The Wallet Creation phase is entered when a user would like to create a wallet. As mentioned in Section 4.6.2, there are two main types of wallets in SuperCloudPay, namely personal wallet and collaborative wallet. Creating a personal wallet is only possible upon account creation. It consists of three steps. First, the user enters its username, email address and password to the app. Then, the wallet creation algorithm CreateWallet is triggered to create a personal wallet associated with the username, a public key, a secret key and an encrypted balance (an encryption of 0). Finally, the username, email address, password, public key and encrypted balance are sent to Cloud.

Meanwhile, the creation of a collaborative wallet also consists of three steps. First, the user enters the name and description of the wallet to the app. Then, the wallet creation algorithm CreateWallet is triggered to create a personal wallet associated with the name, the description, a public key, a secret key and an encrypted balance (an encryption of 0). Finally, the name, description, public key and encrypted balance are sent to Cloud.
**Balance Checking.** The Balance Checking phase is entered when a user requests the information of a wallet. It consists of only one step of running the balance reveal algorithm `RevealBalance` to obtain the plaintext balance.

**Confidential Transaction.** The Confidential Transaction phase is entered when **Sender** initiates a confidential transaction. It consists of six steps as illustrated in Figure 26:

1. **Sender** gets the receiver public key $pk_r$:
   - (a) by scanning a QR code,
   - (b) by redirecting from an online store,
   - (c) from the contact book, or
   - (d) etc.

2. **Sender** initiates a confidential transaction $ctx$ of amount $v$ and sends it to **Cloud** by running $ctx \leftarrow \text{CreateCTx}(sk_s, pk_s, pk_r, v)$ where $ctx = (sn, memo, aux)$, $memo = (pk_s, pk_r, C_s, C_r)$ is a concatenation of the sender public key $pk_s$, receiver public key $pk_r$, transaction amount ciphertext encrypted under the sender public key $C_s$, transaction amount ciphertext encrypted under the receiver public key $C_r$ and $aux = (\pi_{legal}, \sigma)$ provides auxiliary information for the transaction.

3. **Cloud** forwards $(pk_s, C_r)$ to **Receiver** as a transaction request.

4. **Receiver** decrypts $C_r$ by its secret key $sk_r$ to obtain a plaintext value $\tilde{v}$, and checks if it is the transaction amount agreed upon. Then, it tells **Cloud** whether it accepts the transaction request. In case it does not, the phase is aborted.
5. **Cloud** checks if the transaction is valid, i.e. satisfying the following properties: 1) the wallet balance of **Sender** is sufficient; 2) $C_s$ and $C_r$ correspond to the same plaintext transaction amount $t$; and 3) $t$ is non-negative, by **VerifyCTx**. If not, it sends an error message to both **Sender** and **Receiver**, and aborts the phase.

6. Finally, **Cloud** runs **UpdateCTx** to decrease the encrypted sender wallet balance and increase the encrypted receiver wallet balance. The state of each wallet balance is increased by 1. After that, a message denoting successful transaction is sent to both the sender and the receiver.

Lastly, for regulatory and supervisory purposes, we have the following special phases:

**Auditing.** The Auditing phase is entered when a regulatory requests to check the satisfiability of certain confidential transactions $\{ctx\}$ of a user towards a regulation policy $f$. It consists of five steps.

1. **Cloud** requests the user to generate a proof for $\{ctx\}$ with respect to $f$.

2. The user runs the confidential transaction justification algorithm **JustifyCTx** for $\{ctx\}$ with respect to $f$ to obtain a corresponding proof $\pi$.

3. The user sends $\pi$ to **Cloud**.

4. **Cloud** forwards $\pi$ to the regulator.

5. The regulator runs the confidential transaction audit algorithm **AuditCTx** for $\{ctx\}$ with respect to $f$ to check whether $\{ctx\}$ complies with $f$. If not, subsequent regulatory actions may be performed by the regulator.

**Supervising.** The Supervising phase is entered when a supervisor requests to inspect a confidential transaction $ctx$. It consists of only one step of the supervisor running the confidential transaction open algorithm **OpenCTx** to get the plaintext transaction amount. Any implication of the amount may suggest a subsequent legal action.

### 5.3.2 Implementation

The protocol has been implemented in C++ with subroutines from https://github.com/yuchen1024/Kunlun.

### 5.3.3 Discussion

The proposed protocol offers privacy by design. Throughout the whole protocol, the transaction amount is end-to-end encrypted. In other words, it is only known to **Sender**
and Receiver, and the medium of transfer (i.e. Cloud) does not have access to it. This can ensure the confidentiality of the transaction, and can thus preserve user privacy.

Moreover, the protocol can ensure the validity of each transaction whilst preserving user privacy. Utilizing NIZK proof by running \texttt{VerifyCTx}, the three conditions stated in the fifth step in the confidential transaction phase can be checked without knowing the actual wallet balance of Sender and the actual transaction amount. This can protect user privacy. Meanwhile, checking those three conditions can prevent malicious users from creating money for free by collusion, which is critical in a payment system.

Meanwhile, although the supervising phase violates privacy by design, the auditing phase still preserves privacy at its best. More specifically, no transaction amount is revealed during the auditing phase despite the satisfiability check.

Furthermore, the protocol does not consist of any decentralized elements. The centralized architecture underlying the protocol may allow it to be much more scalable than privacy coins, which adopts a decentralized architecture. Compared to the original adoption of ADCP in [17], this adoption (with necessary modifications) can theoretically allow confidential transactions to be conducted at a much higher speed and at a much lower cost. Removing the decentralized elements can also allow the use of cloud database, for which the advantages are discussed in Section 4.3. In addition, it provides a more feasible solution to upgrade existing e-wallet products in privacy protection, with minor changes in the current data structure, compared to the decentralised UTXO-based model.

As confidential transactions are the core of a privacy-preserving e-wallet, the protocol has built a useful foundation for the ongoing development of privacy-preserving e-wallets.

5.4 Limitations

5.4.1 Ignorance of Practical Issues

The proposed confidential transaction protocol has not yet been integrated to the app. Hence, its efficiency is yet to be evaluated. Practical issues, such as system crashes and inconsistency, are not taken into consideration.

5.4.2 Compliance with Regulatory Requirements

Regulated by the Hong Kong Legislative Council and Monetary Authority, the marketization of e-wallets is facing increasingly more regulations and compliance requirements, which mainly consist of two parts.
Part 1: Capital Requirements. E-wallet providers should ensure at least 25 million HKD before entering the market. According to the Payment Systems and Stored Value Facilities Ordinance (Cap. 584) by HKMA, “a store value facility (SVF) licensee must have a minimum paid-up share capital of HK$25 million and evidence of sufficient working capital for protecting the float.” This has posed a challenge for the introduction of SuperCloudPay into the market, potentially limiting the impact of our project.

Part 2: Licensing Requirements. Since the commencement of the SVF licensing regime (2016), only a total of 16 SVF licensees\(^3\) have come on board in Hong Kong, showing relatively complicated procedures for SuperCloudPay being licensed.

5.5 Difficulties and Possible Solutions

After migrating the database to cloud servers, the consistency between the local copy and cloud storage of user data may significantly increase the visiting frequency of cloud servers, which could potentially overwhelm the central server and lead to inefficient scalability. A possible solution to this is to perform complexity analysis to evaluate the amount of cloud service to purchase.

\(^3\)SVF licensees in Hong Kong include all onboarding e-wallet and prepaid card payment services providers with value storage features.
6 Next Step

<table>
<thead>
<tr>
<th>Phase</th>
<th>Month</th>
<th>Tasks</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aug 2022</td>
<td>• Brainstorm ideas for FYP and confirm the FYP topic  &lt;br&gt;• Conduct background research on e-wallets  &lt;br&gt;• Conduct market survey  &lt;br&gt;• Enhance knowledge and skills in cryptography and app development</td>
<td>2 Oct 2022  &lt;br&gt;• Detailed project plan  &lt;br&gt;• Project web page</td>
</tr>
<tr>
<td>1 Inception</td>
<td>Sep 2022</td>
<td>• Devise main features  &lt;br&gt;• Design user interface  &lt;br&gt;• Literature on cryptographic techniques and protocols for confidential transactions  &lt;br&gt;• Research on cloud platform to use  &lt;br&gt;• Design backend database</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oct 2022</td>
<td>• Implement Login, Register, Homepage, Sidebar, User Profile, Wallet Dashboard (basic)  &lt;br&gt;• Design confidential transaction protocol</td>
<td>22 Jan 2023  &lt;br&gt;• Preliminary implementation  &lt;br&gt;• Detailed interim report</td>
</tr>
<tr>
<td>2 Elaboration</td>
<td>Nov 2022</td>
<td>• Implement Bottom Navigation Bar, Transaction History, Scan, Payment Code (basic)  &lt;br&gt;• Design confidential transaction protocol</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dec 2022</td>
<td>• Implement Send, Receive, Top-up, Withdrawal, Bill Payment (basic)  &lt;br&gt;• Implement confidential transaction protocol</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jan 2022</td>
<td>• Implement Promise (novel)  &lt;br&gt;• Fine tune the user interface  &lt;br&gt;• Test and debug the entire prototype  &lt;br&gt;• Write interim report</td>
<td></td>
</tr>
<tr>
<td>3 Construction</td>
<td>Feb 2022</td>
<td>• Implement Collaborative Wallet (novel)</td>
<td>18 Apr 2023  &lt;br&gt;• Finalized tested implementation  &lt;br&gt;• Final report</td>
</tr>
<tr>
<td></td>
<td>Mar 2022</td>
<td>• Implement Automatic Bill Splitting (novel)  &lt;br&gt;• Integrate cryptographic protocols into the backend</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Apr 2022</td>
<td>• Connect backend to the cloud server  &lt;br&gt;• Finish implementation of whole app  &lt;br&gt;• Test and debug the entire app  &lt;br&gt;• Write final report</td>
<td></td>
</tr>
</tbody>
</table>

Figure 27: Project Schedule

Figure 27 shows our project schedule. Having completed the inception phase, we have done relevant research on both practical and theoretical levels. The corresponding findings were presented in Chapter 1, Chapter 2 and Chapter 3 respectively. We are currently at the end of the elaboration phase. As detailed in Chapter 5, we have designed and implemented basic functions (i.e. login, register, terms of service, homepage, sidebar, user profile, transaction history, scan QR code, show payment code, wallet dashboard, send, receive, top-up, withdrawal, bill payment) as well as two out of three novel functions (i.e. promise and collaborative wallet), and devised ER Diagram, API document and a confidential transaction protocol based on the findings and methodologies introduced in Chapter 3 and 4. Our next action will be continue following the project schedule to
develop the app, including implementing automatic bill splitting and integrating the confidential transaction protocol to the app. Upon completion of implementation of frontend and backend, we will deploy the backend to the cloud server for the unique feature of seamless synchronization. Finally, the whole app will be tested and debugged.
7 Conclusion

This project develops a cloud-based privacy-preserving e-wallet by leveraging cryptography and cloud technology. The project has completed the design and implementation of basic and novel functions, backend design and documentation and a confidential transaction protocol, which is based on an ACP system and various cryptographic techniques, including ISHE and NIZK proof. Harnessing cryptography, transaction validity and confidentiality can be guaranteed simultaneously. The centralized architecture used in the protocol may enable transaction efficiency, achieving scalability. However, the protocol does not take into consideration practical issues, and the potential impact of SuperCloud-Pay may be limited by real-world regulatory requirements. Meanwhile, some challenges of deployment of backend onto the cloud server are foreseen and a complexity analysis is suggested.

In future work, following the proposed project schedule, automatic bill splitting will be implemented and the confidential transaction protocol will be integrated to the app. After that, the backend will be deployed on the cloud server. Finally, the whole app will be tested and debugged.

At a theoretical level, SuperCloudPay proposed may outperform traditional e-wallets in terms of privacy and security, and prevail over privacy coin transaction protocols in terms of scalability and auditability. At a practical level, SuperCloudPay can enhance the e-payment experience of consumers by improving convenience, privacy and security while preserving auditability for regulators. Upon completion of the project, the e-wallet has the potential to be the first privacy-preserving and the first cloud-based e-wallet in the market.
References


Appendices

A Market Research Survey

A.1 Original Survey and Raw Data

- Google Form (English version): https://forms.gle/yinvsjr64Ey6HK2n8
- Combined raw data of survey (English Chinese version): https://docs.google.com/spreadsheets/d/1RgB_YkYmp47-WGftuuqDDfLhNr89V6zB/edit?usp=sharing&ouid=106265686999015997461&rtpof=true&sd=true

A.2 Result Visualization

https://drive.google.com/file/d/1RP3VmBU07RFKLKvapMSfSQx9fFZzitj/view?usp=sharing

B API Documentation

https://connect.hku.hk/MySharepoint.com/:b:/g/personal/u3568707_connect_hku_hk/EaMc4A0YeshJhC8wUr7ygpAB00GTkvX-6tpIXVnrz20I3A?e=VUI16N