DOT-M: A Dual Offline Transaction Scheme of Central Bank Digital Currency for Trusted Mobile Devices (Extended Version)

Bo Yang[1,2][0000−0003−0894−5925], Yanchao Zhang[1,2], and Dong Tong[1,2]

[1] National Fintech Evaluation Center, Beijing, China
[2] Research and Development Center , Bank Card Test Center, Beijing, China
{yangbo,zhangyanchao,tongdong}@bctest.com

Abstract. In recent years, many major economies have paid close attention to central bank digital currency (CBDC). As an optional attribute of CBDC, dual offline transaction is considered to have great practical value under the circumstances for payment without network connection. However, there is no public report or paper on how to securely design or implement the dual offline transaction function specifically for CBDC. In this paper, we propose DOT-M, a practical dual offline transaction scheme designed for the mobile device user as either a payer or a payee. Precisely, adopting secure element (SE) and trusted execution environment (TEE), the architecture of trusted mobile device is constructed to protect security-sensitive keys and execution of the transaction protocol. According to the trusted architecture, the data structure for offline transaction is designed as well. On this basis, we describe the core procedures of DOT-M in detail, including registration, account synchronization, dual offline transaction, and online data updating. We also enumerate the exceptional situations that may occur during the dual offline transaction, and give specific handling methods for each situation. Moreover, six security properties of the scheme are analyzed under realistic assumptions. A prototype system is implemented and finally tested with possible parameters. The security analysis and experimental results indicate that our scheme could meet the practical requirement of CBDC offline transaction for mobile users from both aspects of security and efficiency.

Keywords: Central Bank Digital Currency (CBDC), Dual Offline Transaction, Secure Element (SE), Trusted Execution Environment (TEE).

1 Introduction

Currently, a number of central banks are exploring central bank digital currency (CBDC). According to the latest survey conducted by the Bank for International Settlements (BIS) [1] on central banks in 65 countries or economies, about 86 percent have carried out researches on digital currencies. Meanwhile, the proportion of central banks that were performing experiments or developing a proof-of-concept prototype increased from 42 percent in 2019 to 60 percent in 2020. In May 2020, the Digital Dollar Foundation published a white paper: The Digital Dollar Project Exploring a US CBDC [2], which proposed a tokenized U.S. digital dollar, outlined the benefits of a CBDC in the context of the U.S. dollar (USD), and presented potential use cases and pilots. In October 2020, the European Central Bank released the report on a digital euro [3], which analyzed the causes and potential impacts of the launch of the digital euro, and then summarized the core principles and general requirements of the functional design of the digital euro. In July 2021, the People's Bank of China issued the Progress on Research and Development of E-CNY in China [4], which introduced that E-CNY supported offline transactions but did not give technical details.

In addition to CBDCs, many private sectors have launched a variety of so-called cryptocurrencies. According to incomplete statistics, there are over 10,000 kinds of cryptocurrencies whose total value exceeds USD1.3 trillion [5]. The most influential cryptocurrency is bitcoin which adopts blockchain and encryption technology. In addition, some commercial institutions launch so-called "stablecoins", and try to stabilize their values by pegging them to sovereign currencies or related assets. For example, Facebook's cryptocurrency project Libra (renamed Diem) released a white paper [6] on June 18, 2019. However, neither Bitcoin nor Libra supports dual offline transaction.

Just as its name implies, dual offline transaction means the e-payment executed between the payer and the payee, neither of whom could connect to their online bank accounts as online e-payment. There are several papers [7,8,9,10,11,12] that have given some optional solutions by using cryptographic methods or blockchain technology. Actually, to achieve the same user experience as cash used in an environment without network, CBDC is supposed to support dual offline transaction. Nowadays, mobile payment between two e-wallets, such as PayPal consumer to consumer (C2C) model for mobile users, is the mainstream way to implement e-payment as well as CBDC, which is more flexible and user-friendly than the way of POS terminal with bank card or host-based card emulation (HCE). Although dual offline transaction has been implemented for the scenario of money from a bank card to an offline POS terminal, this method is not practicable for the transaction between two equipotent e-wallets on mobile devices. Therefore, designing a dual offline transaction scheme under the CBDC environment for both e-wallets on mobile devices is of practical significance and suitable for various application scenarios. The transactions usually happen as both parties' mobile devices are offline or hardly connect to CBDC's service, for example, when they are on an airplane, in an underground parking lot, in remote mountain areas or even in an earthquake area where network system is destroyed. Undoubtedly, for CBDC, how to implement dual offline transaction on mobile devices in both secure and efficient way is a crucial issue.

For the public, mobile devices are the most convenient medium to reach CBDC or other e-payment services. In recent years, some standardization organizations in payment field have been exploring secure mechanisms and services on smart mobile devices. EMVCo proposes Software-based Mobile Payments [13] that make use of commercially available software protection techniques such as white-box cryptographic, obfuscation, and binary protection. Payment Card Industry (PCI) also puts forward some similar solutions [14,15] that do not require the use of a hardware-based secure element, and emphasizes that they are not suitable for offline payment scenarios. The above solutions of the only software-based protection are mainly designed for online payment scenarios. They heavily depend on the back-end banking system, and are unable to be used to protect the dual offline payment scenarios described in this paper.

1.1 Design Principles

In consideration of CBDC, there should be several specific requirements for dual offline transaction scheme. The following desired design principles are essential to achieve a practical scheme and guide our efforts going forward.

Centralized Banking System. On account of financial regulation, CBDC usually adopts the centralized banking system and related mechanism rather than peer-to-peer electronic cash transfer or distributed ledger technology such as blockchain [9]. In comparison with blockchain, the centralized banking system is more conducive to currency issuing management and monetary policy adjustment for a national central bank. More precisely, this system endows the central bank as the only trusted authority or lawful entity of a country with the highest right to produce and regulate digital currency that could be verified by any others. This centralization is just a logical form that can be implemented in a cluster-based hierarchical approach with distributed back-up system in case of single point of failure or massive concurrency.

Strong Robustness. As a kind of electronic payment to simulate physical cash transfer, dual offline transaction of CBDC requires to reliably guarantee the continuity of transaction activities. Indispensable fault tolerance and information records are needed to ensure the strong robustness.

High Security. Since it is relevant to the money topic which is concerned by any people for protecting their property, the security issue is of prime significance. The basic requirement for dual offline transaction is exactly guaranteeing the correct payer paying the correct value of effective digital currencies to the correct payee. Consequently, the secure cryptographic protocol, specialized mechanism for mobile devices and accurate key protection are necessary. Compared with the online transaction, lots of sensitive computations and data records in offline pattern are moved to front-end devices instead of relying on background banking system, so that the security of devices is crucial. Furthermore, three fundamental security properties should be satisfied for dual offline transaction.

– **Unforgeability.** Unforgeability involves two aspects: unforgeability for identity, and unforgeability for digital currency. Firstly, unforgeability for identity means that the adversary can forge neither the identity of an user nor the wallet of an user who participates the offline transaction. Secondly, unforgeability for digital currency means that the adversary cannot forge the digital currency issued by the central bank and cannot forge its transaction records. Unforgeability ensures the authenticity of a transaction.
– **Preventing Double-spending.** Double-spending problem refers to the phenomenon that the same digital currency is paid for twice or more times repeatedly. The problem should be prevented for the validity of monetary system.

- **Non-repudiation.** Non-repudiation means that the payer and the payee cannot deny their offline transaction behaviors. It ensures the effectiveness of a transaction.

Low Latency. The acceptable efficiency of the system endows the dual offline transaction with availability. For a good user experience, the realized scheme should have a low latency that the time overhead of one transaction is preferable to be controlled in 2 seconds.

1.2 Related Work

Digital Currency. With the continuous development of cryptography, numerous innovations have been created in the research field of electronic cash (e-cash) and e-payment[16,17]. E-cash introduced by Chaum [18], allows users to withdraw digital coins from a bank and to spend them to merchants in an anonymous way, thus perfectly emulates conventional cash transactions. E-payment research based on quantum cryptography has appeared in academia [19,20], and digital currency application scenarios have been commercialized [21] in recent years. Based on initial achievements of research on digital fiat currency (DFC) conducted by the People's Bank of China, Yao explored a closed loop which encompasses DFC issuance, transfer and return in the binary model of the central bank associated with some commercial banks [22]. Nevertheless, few literatures of the technique about dual offline transaction within CBDCs can be found.

Offline Electronic Payment. With regard to existing works, current offline transaction methods [7] generally fall into three categories. The first category is about the offline POS terminal transaction. Li et al. [23] proposed an offline transaction e-commerce system model for an offline POS to make transaction between a customer and a merchant through another mobile device. However, this scenario model is limited and inapplicable for CBDC. The second category is blockchain-based offline situations. The scheme [10] presented a secure, versatile light payment based on blockchain, which exploited the off-chain and offline payments. Dmitrienko et al. [11] provided a solution for Bitcoin payments, which carried out secure payments with Bitcoin in offline settings. VOLGAPAY [12] was put forward as a blockchain-based system to deploy secure offline payment terminal infrastructure. In addition, the paper [24] described the concept of scalable off-chain payments - Lightning Network. Unfortunately, blockchain-based transactions violate the design principle for centralized CBDC system. The third category is the utilization of cryptography to solve the both issues of offline transaction and preserving user privacy. Both Kutubi et al. [8] and Batten et al. [25] presented their secure offline electronic payment scheme respectively by adopting untraceable blind signature (BS). Whereas, the obviously increasing protocol and computation complexity is inadequate to guarantee the robustness and efficiency for CBDC. Additionally, OPS protocol [9] was given for CBDC offline payment system specially towards a two-tier hierarchical infrastructure, but without enough security design or implementation consideration.

Mobile Security. In this field, purely software-based secure techniques, such as software obfuscation and white-box cryptography, are more vulnerable in the face of attacks from bottom system or hardware [13]. Another example is host-based card emulation (HCE) with near field communication (NFC) for mobile payment[26,27] , whose security builds on either cloud services or the local application to simulate a secure element (SE). It cannot support dual offline transaction or satisfy its high security requirement respectively. In comparison, the technique of jointly using SE [28] and trusted execution environment (TEE) [29,30,31,32] on mobile devices endows us with a feasible technical route. SE provides tamper-resistant mechanism to keep core assets secure even if the underlying hardware of mobile device is compromised by hardware-based attacker. Isolated from a rich execution environment (REE) where the guest operation system (OS) runs, TEE aims to protect sensitive codes execution and against both OS layer attacks and other software-based attacks. In general, SE and TEE (or similar forms) are widely deployed and enabled in a great number of today's mobile devices. For example, equipped with both SE and TEE or their similar forms, the secure hardware architecture has been prevailingly applied by recent devices such as Google Nexus and Pixel series, Samsung Galaxy series, Huawei Mate series, OPPO Find series and Apple iPhone. As a prevalent example of providing TEE for embedded devices, ARM TrustZone [33] has been used to execute security-critical services [34]. Based on ARM TrustZone and the divisible e-cash scheme with the best efficiency by Canard et al. [35], Yang et al. [36] proposed AEP-M, a practical anonymous e-payment scheme for mobile devices, which enables an user to spend his digital coins securely and efficiently while preserving his privacy. The paper [29] also presented a secure solution for building the trusted mobile platform and its key management. Although, some vulnerabilities for TEE implementation were found, the manufactures have elaborately fixed them and continuously provide security blanket. In fact, applying both SE and TEE is an appropriate choice for related to the security of a country's monetary sovereignty. Therefore, it is authentically worthwhile and necessary to build that CBDC's mobile scheme on SE and TEE, even if they have not been available for every mobile device so far. To the best of our knowledge, there is no dual offline transaction scheme specially designed for mobile devices using SE and TEE.

1.3 Our Contribution

Based on SE and TEE, we propose DOT-M, a practical dual offline transaction scheme for trusted mobile devices, which enables an user to pay digital currency of CBDC securely and efficiently without Internet to the other equivalent user. This is a complete work to design an efficient dual offline transaction scheme of CBDC integrated with SE and TEE.

For device-centered design, we make following steps towards practical and secure usage:

- For the scenario of dual offline transaction of CBDC, the secure solution for trusted mobile device is specially constructed;
- DOT-M utilizes a series of secret keys, which are derived from a terminal master key reproduced in SE, to protect users' digital currency of CBDC and data;
- The sensitive codes on the user side of DOT-M are isolated and executed in TEE for the possibility that the guest OS is compromised;
- In DOT-M, users could authenticate each other's identity and wallet identity, as well as the authenticity of CBDC in the absence of a network.
- The protocol of dual offline transaction and exception handling are elaborately designed to satisfy users' actual requirements on mobile devices.

We also design the data structure of CBDC in order to split the value of one integrated digital currency flexibly when dual offline transaction is executed. In the meantime, the security properties of the transaction protocol are guaranteed, which include preventing man-in-the-middle attack, preventing intruder attack, preventing malicious payer or payee, preventing double-spending, non-repudiation and unforgeability. Especially, solving double-spending problem is so indispensable for dual offline transaction that ensures each digital currency will only be spent once. Furthermore, we implement a prototype of DOT-M and evaluate its efficiency at the security level of 256-bit. The experimental results show that our scheme is efficient enough for practical usage, even from the perspective of mobile devices.

2 System Model and Assumptions

2.1 Notation

In the paper, we use the notation shown in Table 1.

Table 1. Notation used in this paper

Notation	Descriptions
$y := x$	y assigned as x
$x\|\|y$	Concatenation of x and y
$(y_1, ..., y_j) \leftarrow \mathsf{A}(x_1, ..., x_i)$	An algorithm with input $(x_1, ..., x_i)$ and output $(y_1, ..., y_j)$
$\mathsf{Sign}(k, m)$	Digital signature for a message m using a private key k

2.2 System Model

The system model of DOT-M proposed in this paper is composed of three kinds of participating entities: mobile device $\mathcal{R}$ of payee and mobile device $\mathcal{P}$ of payer, central bank $\mathcal{C}$. $\mathcal{R}$ and $\mathcal{P}$ are bound with payee and payer respectively, and the user behaviors are achieved through the mobile device. The mobile device used for the trader is equipped with security chip for SE and ARM processor chip supporting TrustZone extension technology for TEE. Both of the devices could communicate with each other through NFC or Bluetooth for dual offline transaction. In charge of providing CBDC services and CBDC wallet applications, $\mathcal{C}$ should be a central bank or a third-party agency authorized by the central bank. Generally, $\mathcal{C}$ has multiple data centers to meet the needs of powerful computing and big data storage capabilities. In reality, data centers can be established by many different commercial banks which are able to play more functional roles in the whole ecosystem of CBDC. Figure 1 illustrates the system model for our scheme.

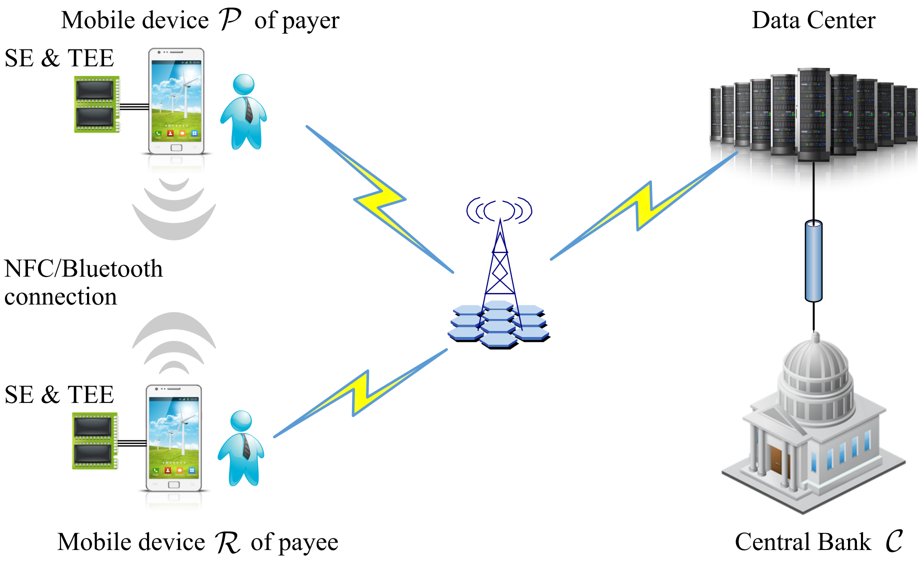

Mobile device $\mathcal{P}$ of payer
SE & TEE
Data Center
NFC/Bluetooth connection
SE & TEE
Mobile device $\mathcal{R}$ of payee
Central Bank $\mathcal{C}$

Fig. 1. System model of DOT-M.

2.3 Security Assumptions and Threat Model

In order to simplify the design of our core scheme, we establish the scheme on the following assumptions:

Assumption 1. The various security mechanisms designed and implemented for SE and TEE by their manufacturers are correct and have not been deliberately planted backdoors.

Assumption 2. All the basically standard cryptographic algorithms are correctly applied and implemented without security risk.

Assumption 3. Mobile device manufactures correctly enable and initialize SE and TEE for each device before delivery. As a consequence, each device has its predefined unique device key with the certificate that could be authenticated by other entities for ensuring it is a trusted mobile device and its critical executions in TEE are trusted.

Assumption 4. The communications between every two entities (including devices and central bank) build on secure transport protocols like TLS, which can provide confidentiality, authenticity and integrity protection for data transmission.

These assumptions are quite tenable because the mentioned requirements have been generally satisfied in practically industrial activity. Meanwhile, constrained by the market supervision and the force of law, the trust relationships between the central bank and manufactures are easily established and maintained.

Based on the assumptions, DOT-M protects against the following adversary:

- The adversary can attack the scheme itself by attempting to pretend legitimate entities, forge data and manipulate data transmission between entities.
- The adversary can perform software-based attacks which compromise the mobile Rich OS or existing applications running in REE. DOT-M interfaces in REE are also available for the adversary.

- The adversary can physically access the mobile device. He can reboot the device and gain access to data residing on persistent storage.
- The adversary can execute the laboratory-level attack to break SE or mount side-channel attack on SE of the mobile device for obtaining the vital keys.
- The adversary tries to forge the identity of a specific payer or payee to execute the dual offline transaction with others. This breaks the property of unforgeability.
- The adversary tries to forge effective digital currency issued by the central bank. This also breaks the property of unforgeability.
- The malicious payer or payee tries to fabricate an untrusted mobile device or manipulate his trusted mobile device to execute dual offline transaction that deliberately disobeys the process of the proposed scheme.
- The malicious payer tries to double spend his digital currency. This breaks the property of preventing double-spending.
- The malicious payer tries to deny his payment action that has indeed occurred. This breaks the property of non-repudiation.
- The malicious payee tries to deny his receipt action that has indeed occurred. This also breaks the property of non-repudiation.

However, we ignore the malicious behaviors of tampering with the TrustZone hardware. Moreover, privacy-preserving problem is a hot topic for electronic payment, but we do not cover anonymity, unlinkability and untraceability for dual offline transaction in our scheme. On the one hand, several papers have given some optional solutions by using cryptographic methods such as Schnorrs untraceable blind signature [25]. On the other hand, using extra cryptographic methods will obviously increase both computation and protocol complexity of the whole scheme so that its robustness and efficiency may decline for the experience-sensitive dual offline transaction.

3 DOT-M Scheme for Mobile Devices

In this section, we provide a security solution on mobile device for CBDC. Depending on it, the design of data structure for offline transaction and the construction of DOT-M scheme are detailed. The security properties of DOT-M are finally analyzed.

3.1 Security Solution on Mobile Device

The security solution is the terminal foundation for establishing reliable dual offline transaction. Three parts constitute the solution: the specific architecture of trusted mobile device, key derivation with sensitive data management and basic key system of CBDC.

3.1.1 The Architecture of Trusted Mobile Device

Leveraging SE and TEE technology, we design a trusted mobile device architecture for DOT-M. Both $\mathcal{P}$ and $\mathcal{R}$ rely on this architecture to securely execute dual offline transaction. On the basis of the software and hardware capabilities of existing mobile device, we build the security solution that targets at economy, flexibility and extensibility.

Combined with SE, TEE can shield the integrity of the execution process of sensitive programs, the confidentiality and the integrity of sensitive data, which is fundamental to the security of offline transaction. Figure 2 shows the detailed architecture with the way the components interact with each other. The basic functionality of DOT-M in the architecture contains two components: untrusted DOT-M Proxy in normal world (NW) and security-sensitive DOT-M Service in secure world (SW). In reality, SW instantiates TEE, while NW implements REE. The different components are formally described as follows.

DOT-M Proxy. As a bridge between user space and kernel space in NW, it can directly communicate with the normal mobile applications. Waiting for their requests, the proxy handles the parameters and preprocesses them. According to the request type, the proxy would call DOT-M Service for substantive computations of the scheme and finally return the results.

DOT-M Service. It is the core component to perform DOT-M secure computations and operations. The execution of the component codes is under the well protection of TrustZone isolation mechanism. DOT-M Service mainly consists of the following four subcomponents:

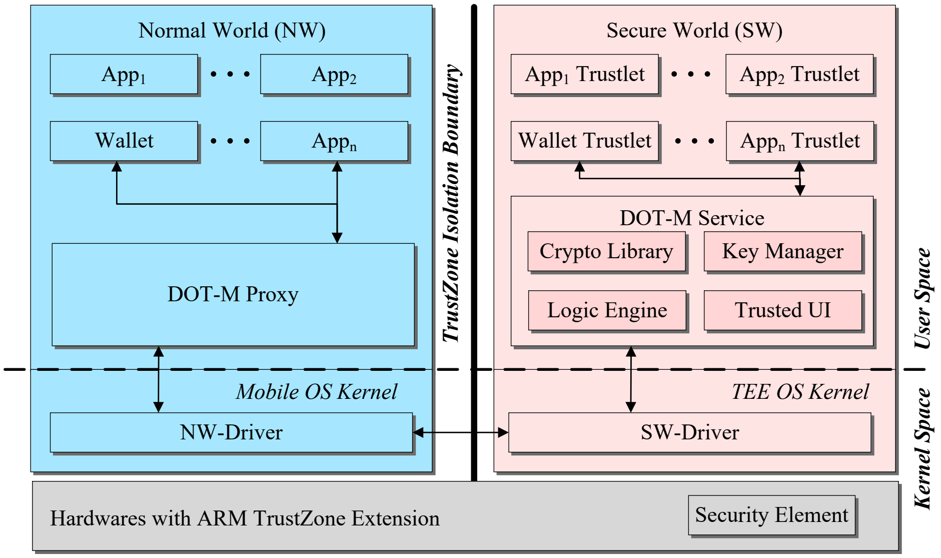


Fig. 2. Architecture of trusted mobile device for DOT-M.

- **Crypto Library:** offers cryptographic algorithm support for Logic Engine. In SW, it integrates hash algorithms, asymmetric and symmetric algorithms and other cryptographic operations.
- **Key Manager:** operates secret keys of the scheme and provides full-lifecycle security protection for these keys. The operations and protections involve the generation, storage, destruction, and other processes of cryptographic keys and their possible certificates.
- **Logic Engine:** executes the computations of security-sensitive parts of DOT-M. Logic Engine reads necessary parameters and data to run operations relying on scheme specification. The operations cover cryptographic computation, communication with SE, data sealing, data unsealing and offline transaction routine.
- **Trusted UI:** takes over the control of mobile device screen as a secure man-machine interface, and protects the authenticity and integrity of input-output data. In our scheme, it is ensured that the sensitive data such as the typed transaction amount or password by user will not be stolen or tampered.

Wallet and Wallet Trustlet. The corresponding application should be launched if the user wants to enjoy the offline transaction of CBDC. For upperlevel interaction, the CBDC application released by $\mathcal{C}$ consists of two parts: an App for NW called Wallet and an App Trustlet for SW called Wallet Trustlet. Wallet provides users with the general GUI, remote service access and other basic functions, while Wallet Trustlet is securely loaded and trusted for handling security-sensitive inputs, data operations and communications with the other device through NFC or Bluetooth during dual offline transaction. In fact, Wallet Trustlet is a trusted application defined in TrustZone. Generally, as well as the customized DOT-M Service in SW, Wallet Trustlet's installation procedure is tightly controlled by the mobile device manufactures because the commercial TEE platforms of the manufactures are quite closed. However, it is feasible to organize the manufactures to support the preinstallation of important components and the reservation of their update entries through uniform specifications, especially in a nationally significant scenario like CBDC. And that is actually what the manufactures are doing.

Components in Hardwares. Protected by TrustZone mechanism, SE is only accessible for SW. With a variety of hardware protection mechanisms for resisting the laboratory-level attack, SE contributes to generate the master key and act as the root of trust in DOT-M.

3.1.2 Key Derivation and Sensitive Data Management

Prior to describing the concrete construction of DOT-M, we show how to derive root keys using SE and how to protect sensitive data. In this scheme, the method of trust chain building refers to GP specification [28].

Master Key Generation. Inside SE, the terminal master key tmk is generated by the true random number generator (TRNG). When the mobile device is powered on for the first time, a fixed-length data is read from the TRNG as tmk, and then it resides in this SE. This key is unique for each SE bound with each mobile device so that it is also considered as the root and never exposed out of SE.

Key Derivation. Before delivery of the mobile device, SE is customized to have the deterministic key derivation function KDF: $\widetilde{\mathcal{K}} \leftarrow \widetilde{\mathcal{S}} \times \{0,1\}^*$, where $\widetilde{\mathcal{S}}$ is the key seed space, and $\widetilde{\mathcal{K}}$ is the derived key space. Using KDF, the storage root key can be derived as $srk \leftarrow \mathsf{KDF}_{tmk}(\texttt{"storage\_root"})$. srk is used by Key Manager in TEE for generating specific storage keys to preserve sensitive data. Note that all the keys derived from tmk in SE are never stored permanently. Besides, other functional root keys, such as identity root key, for building trust chain could be derived similarly by using KDF with different parameters. They are just regained via KDF in the same way when needed.

Sensitive Data Management. Utilizing the storage keys derived from srk and the generic methods for data protection by TrustZone, DOT-M Service could seal the sensitive data of the scheme as blobs to permanently store with protecting their confidentiality and integrity. When needed, these data could be recovered through unsealing operation. Considering that srk is derived from tmk in SE to act as a hardware-based trusted root, the protection for the sensitive data has higher security in comparison with merely using keys generated in TEE.

3.1.3 Basic Key System of CBDC

Aside from the assumption about bottom protocols of transportation in Sect.2.3, the entire secure foundation of CBDC for upper layer is also established through the public key certificate infrastructure (PKI). Actually, central bank needs to assign digital certificates of public keys to entities, ensuring that a relationship of trust can be built within the CBDC system. Table 2 shows the definitions of different certificates and keys in our scheme for offline transaction.

Table 2. The definitions for certificates and keys.

Symbol	Definition	Origin
$pk_{\mathcal{C}}$	Public key for verifying digital currency of CBDC	Issued by $\mathcal{C}$
(usk_i, upk_i)	Private-public key pair of user i	Generated in TEE
$ucert_i$	Certificate of user i	Issued by $\mathcal{C}$
(wsk_j, wpk_j)	Private-public key pair of wallet for user j	Generated in TEE
$wcert_j$	Certificate of wallet for user j	Issued by $\mathcal{C}$
pwd	Payment password of wallet	Set by user

Preset in CBDC wallet application, the unified public key $pk_{\mathcal{C}}$ is used for verifying the digital currency issued by the central bank $\mathcal{C}$. When a wallet application is started on a mobile device by its user for the first time, (usk, upk) and (wsk, wpk) are generated in TEE through DOT-M Service and sent to $\mathcal{C}$, then $\mathcal{C}$ issues $ucert$ and $wcert$ respectively for the user. For the same user, his $wcert$ is recorded and associated with his $ucert$ in $\mathcal{C}$'s database. Subsequently, the user is required to set a payment password pwd through TUI on his device for future transactions.

3.2 Design of Data Structure

In this section, we propose the design of data structure for offline transaction of CBDC.

3.2.1 Data Structure of Currency Block

Figure 3 shows the data structure of currency block for offline transaction. The structure consists of form of presentation, offline transaction record, remaining amount of value , number of times of offline transaction. The instance of this data structure represents a certain amount of digital currency as a whole with its information. The offline transaction about this currency block is the procedure to consume the digital currency until the remaining amount of value turns into 0. During one transaction, several currency blocks would be used to pay simultaneously.

The form of presentation describes a basic piece of digital currency of CBDC that could be paid and verified. It is able to define any amount of original value of digital currency in a single string issued (i.e. signed) by the central bank $\mathcal{C}$. Its data save the original currency information that would never be changed during the offline transaction. The data structure of offline transaction record is linked list that records all the necessary contents of each offline transaction involved this currency block. Remaining amount of value represents the currently spendable amount of value of the digital currency and would be deducted after each transaction. Number of

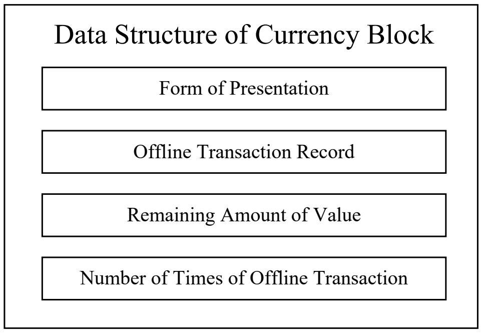

Data Structure of Currency Block
Form of Presentation
Offline Transaction Record
Remaining Amount of Value
Number of Times of Offline Transaction

Fig. 3. Data structure of currency block for offline transaction of CBDC.

times of offline transaction increments by 1 after each transaction and is used for the maximum limit of dual offline transaction for the sack of efficiency and security.

The form of presentation is formally a string shown in Figure 4. The form contains at least the unique serial number, amount of value, original owner and signature of $\mathcal{C}$. In practice, the string can have many other extended fields that are beyond the scope of this paper.

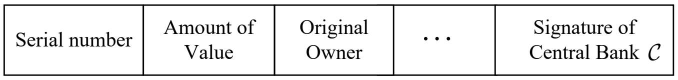


Fig. 4. Form of presentation for the digital currency of CBDC.

Figure 5 shows data structure of offline transaction record which is a linked list. Recording the information of a single offline transaction, each node contains transaction amount, transaction timestamp, $\mathcal{P}$'s certificate $ucert_{\mathcal{P}}$, $\mathcal{R}$'s certificate $ucert_{\mathcal{R}}$, signature of $\mathcal{P}$.

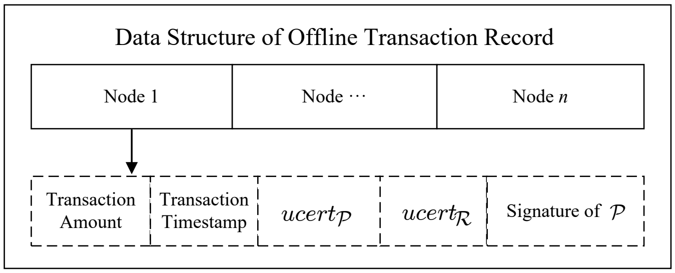


Fig. 5. Data structure of offline transaction record.

3.2.2 Data Structure of Wallet

Figure 6 shows the data structure of the wallet for mobile user, which contains data structure of currency block, wsk, $wcert$, $pk_{\mathcal{C}}$, total remaining amount of value in wallet. The key $pk_{\mathcal{C}}$ is for verifying the signature of $\mathcal{C}$ for each digital currency. In one instance of the data structure of wallet, there could be several instances of currency blocks due to how much digital currency the user has.

3.2.3 Data Structure of User

This paper also defines and applies the user data structure described in Table 2, including usk, $ucert$ and pwd. In our scheme, the data structure of user with its keys is used to identify the behaviors of different transaction users bound with their real back-end accounts, while the data structure of wallet with its keys is to identify the user's exact wallet for paying or receiving the digital currency during the dual offline transaction.

3.3 The Details of Dual Offline Transaction Scheme

The scheme description of this paper only considers the case in which the user has one wallet application with one instance of its data structure. In real life, an user would have several wallet applications on his device, and

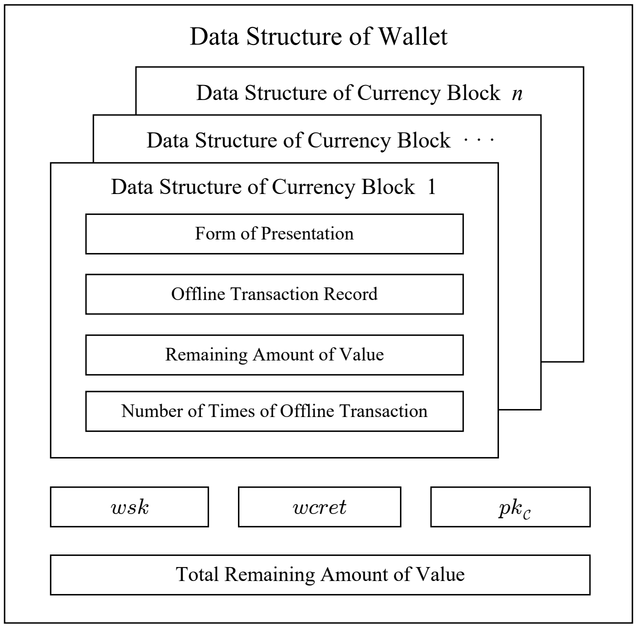

Data Structure of Wallet
Data Structure of Currency Block n
Data Structure of Currency Block $\cdots$
Data Structure of Currency Block 1
Form of Presentation
Offline Transaction Record
Remaining Amount of Value
Number of Times of Offline Transaction
wsk
$wcret$
$pk_{\mathcal{C}}$
Total Remaining Amount of Value

Fig. 6. Data structure of wallet.

it is one of the reasons why there is a need to distinguish wallet keys with user keys. In view of this situation, the procedure of the scheme is a little different where the wallet identifier that represents the unique identity of a wallet, should be added to the data structure of wallet. When making a transaction, the user selects the corresponding wallet identifier as the transaction objective.

The dual offline transaction scheme mainly consists of 4 procedures: registration, account synchronization, dual offline transaction and online data updating. Exception handling is also necessary for the scheme's validity and availability.

3.3.1 Registration

This procedure is specifically divided into the following parts:

1. User downloads the CBDC wallet application on the mobile device and its two parts (i.e. Wallet and Wallet Trustlet) are respectively installed and activated.
2. User starts the application for the first time, and it connects to central bank $\mathcal{C}$, and verifies that $\mathcal{C}$ is real.
3. User sets the account and login password of the application. User's mobile device generates (usk, upk) in TEE, and sends upk to $\mathcal{C}$. $\mathcal{C}$ issues $ucert$, and sends $ucert$ to user's mobile device where $ucert$ is well protected by sealing described in Sect.3.1.2.
4. User's mobile device generates (wsk, wpk) in TEE, and sends wpk to $\mathcal{C}$. $\mathcal{C}$ issues $wcert$, and sends $wcert$ to user's mobile device where $wcert$ is also protected.
5. User sets pwd of wallet for payment by TUI.

3.3.2 Account Synchronization

When the mobile device is connected to the Internet, the wallet application connects to $\mathcal{C}$ for account ledger synchronization. The information synchronized with the ledger mainly includes the wallet data structure so that the user gets his pre-existing digital currency from his online account and put it into the wallet application on his mobile device which is prepared for offline transaction. DOT-M Service seals and saves the synchronized data of wallet as sensitive data. The method of exchanging digital currency with cash or money from a bank account is the foundational function of CBDC, therefore it is omitted here.

3.3.3 Dual Offline Transaction

Before the dual offline transaction starts, $\mathcal{R}$ and $\mathcal{P}$ need to complete the following preparations:

1. $\mathcal{R}$ requires the payee to input the payment amount on the device through TUI.

2. $\mathcal{P}$ requires the payer to input the wallet payment password *pwd* on the device through TUI, and validate it.
3. $\mathcal{R}$ and $\mathcal{P}$ establishes a secure channel through NFC or Bluetooth.

Next, the dual offline transaction begins, and there are 9 major steps in the protocol as shown in Figure 7.

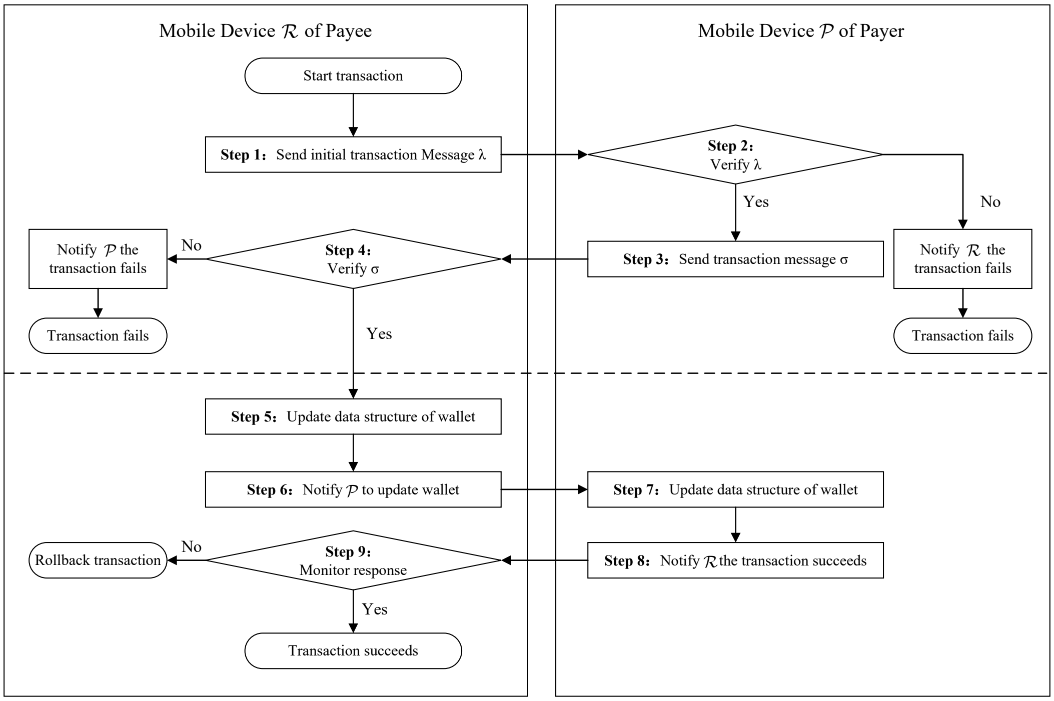

Mobile Device $\mathcal{R}$ of Payee
Mobile Device $\mathcal{P}$ of Payer
Start transaction
Step 1: Send initial transaction Message λ
Step 2: Verify λ
Yes
No
Notify $\mathcal{P}$ the transaction fails
No
Step 4: Verify σ
Step 3: Send transaction message σ
Notify $\mathcal{R}$ the transaction fails
Transaction fails
Yes
Transaction fails
Step 5: Update data structure of wallet
Step 6: Notify $\mathcal{P}$ to update wallet
Step 7: Update data structure of wallet
Rollback transaction
No
Step 9: Monitor response
Step 8: Notify $\mathcal{R}$ the transaction succeeds
Yes
Transaction succeeds

Fig. 7. Dual offline transaction protocol.

Step 1.$\mathcal{R}$ builds the initial transaction information λ, and sends it to $\mathcal{P}$.

1. Use $usk_{\mathcal{R}}$ to output a signature $\alpha \leftarrow \mathsf{Sign}(usk_{\mathcal{R}}, tsn)$, where *tsn* is the unique serial number of transaction.
2. Use $wsk_{\mathcal{R}}$ to output a signature $\beta \leftarrow \mathsf{Sign}(wsk_{\mathcal{R}}, v)$, where v is the amount of value for offline transaction.
3. Send $\lambda := (\ tsn || \ v \ || \ \alpha \ || \ ucert_{\mathcal{R}} \ || \ \beta \ || \ wcert_{\mathcal{R}})$ to $\mathcal{P}$.

Step 2. $\mathcal{P}$ verifies λ through the following steps.

1. Obtain $upk_{\mathcal{R}}$ from $ucert_{\mathcal{R}}$, and verify α.
2. Obtain $wpk_{\mathcal{R}}$ from $wcert_{\mathcal{R}}$, and verify β.
3. Check whether v and payee's identity indeed accord with the payer's original intention by showing the contents for the payer through TUI.
4. Check whether the remaining amount of value of its wallet is greater than the offline transaction amount.

If the above sub-steps fail, $\mathcal{P}$ would notify $\mathcal{R}$ the transaction fails, and close the transaction.

Step 3. $\mathcal{P}$ builds the transaction message σ, and sends it to $\mathcal{R}$.

1. Choose one or a cluster of applicable currency blocks as needed, generate the transaction record nodes for the corresponding blocks, append all generated nodes to the blocks, and assemble the transaction data *tsd*.
2. Use $wsk_{\mathcal{P}}$ to output a signature $\gamma \leftarrow \mathsf{Sign}(wsk_{\mathcal{P}}, tsd)$.
3. Send $\sigma := (tsd \ || \ \gamma \ || \ wcert_{\mathcal{P}})$ to $\mathcal{R}$. More clearly, the data structures of *tsd* and σ are shown in Figure 8.

Step 4. $\mathcal{R}$ verifies σ through the following steps.

1. Obtain $wpk_{\mathcal{P}}$ from $wcert_{\mathcal{P}}$, and verify γ.

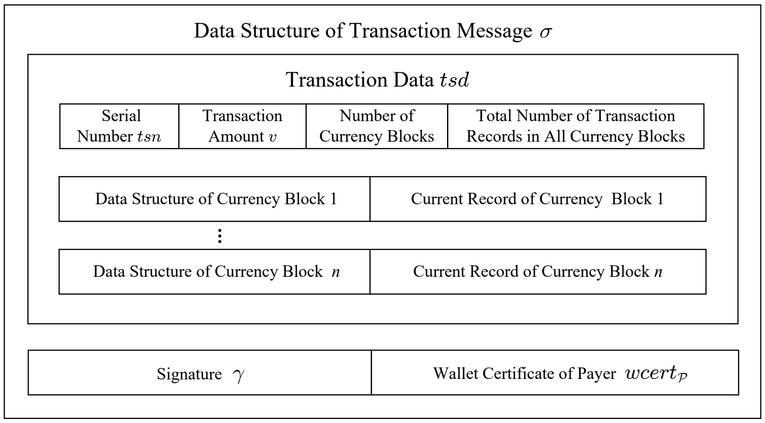

Data Structure of Transaction Message σ
Transaction Data tsd
Serial Number tsn
Transaction Amount v
Number of Currency Blocks
Total Number of Transaction Records in All Currency Blocks
Data Structure of Currency Block 1
Current Record of Currency Block 1
Data Structure of Currency Block n
Current Record of Currency Block n
Signature γ
Wallet Certificate of Payer $wcert_{\mathcal{P}}$

Fig. 8. Data structure of transaction message.

2. Obtain $upk_{\mathcal{P}}$ from $ucert_{\mathcal{P}}$, and verify the authenticity of all current offline transaction records. Similarly, $\mathcal{R}$ can verify the authenticity of all previous transaction records via linked lists inside different currency blocks.
3. Use $pk_{\mathcal{C}}$ to verify that the digital currency of CBDC is exactly issued by $\mathcal{C}$.

If the above sub-steps fail, $\mathcal{R}$ would notify $\mathcal{P}$ the transaction fails, and close the transaction.

Step 5. $\mathcal{R}$ updates the data structure of wallet, mainly involving linking each current node to the corresponding record inside its currency block, increasing the number of times of offline transaction in each currency block, and renewing remaining offline amount of wallet.

Step 6. $\mathcal{R}$ notifies $\mathcal{P}$ to update the data structure of wallet by sending a notification message signed using $usk_{\mathcal{R}}$.

Step 7. $\mathcal{P}$ verifies the message using $upk_{\mathcal{R}}$ from **Step 2** and updates its data structure of wallet.

Step 8. $\mathcal{P}$ notifies $\mathcal{R}$ that the transaction is successful.

Step 9. $\mathcal{R}$ monitors the response from $\mathcal{P}$. If $\mathcal{R}$ receives the response from $\mathcal{P}$ within the due time, $\mathcal{R}$ closes the successful transaction and shows the message, otherwise it rollbacks transaction, and restores the data structure of wallet to the state before the start of this transaction. Only when the payee sees the success message on the screen of $\mathcal{R}$, he admits this effective transaction.

3.3.4 Online Data Updating

Whenever $\mathcal{R}$ connects to Internet, it submits the previous offline transaction data of the wallet to the CBDC system server for account ledger synchronization. Similar, $\mathcal{P}$ also does online updating for his account synchronization. However, once $\mathcal{C}$ finds the updating data from $\mathcal{R}$ are inconsistent with those from $\mathcal{P}$, $\mathcal{C}$ would give priority to $\mathcal{R}$'s data and execute corresponding exception handling described in Sect.3.3.5.

3.3.5 Exception Handling

Exceptional situations that have bad impacts may occur during the procedure of the dual offline transaction, such as when the mobile device crashes, the device is out of power, NFC communication fails and so on. The occurrence of exception means the procedure is terminated at once in the position where it happens. As a consequence, the exception handling should be considered and triggered timely in case of the failure for synchronization of either critical data or the changing amount of digital currency. The part below the dotted line of Figure 7 shows the prepared design for the exceptional situation. The whole procedure of dual offline transaction can be categorized into 6 kinds of exceptional situations according to where exceptions occur and which mobile device causes them. Each situation corresponds to a specific handling method.

Exceptional Situation 1. Exception occurs anywhere before **Step 5** during the whole procedure. Under this circumstance, no matter whose mobile device causes the exception, the transaction would immediately stop and there is no substantive change for either $\mathcal{P}$ or $\mathcal{R}$. The payer and payee do not realize transferring digital currency.

Exceptional Situation 2. Exception is caused by $\mathcal{R}$ and occurs after **Step 5** and before **Step 6**. At this point, $\mathcal{R}$ has already updated its wallet (i.e., received the digital currency), but $\mathcal{P}$ cannot update his wallet because $\mathcal{P}$ would never receive the notification from $\mathcal{R}$'s **Step 6**. When $\mathcal{R}$ reboots and recovers its wallet application, the wallet could detect that $\mathcal{R}$ did not finally execute **Step 9** in the last transaction. Then it would automatically rollback the transaction so that the wallet deducts the amount of the received digital currency which actually does not paid successfully by $\mathcal{P}$. As a result, neither the payer nor the payee realizes transferring digital currency.

Exceptional Situation 3. Exception is caused by $\mathcal{R}$ and occurs after **Step 6** and before **Step 9**. At these points, $\mathcal{P}$ would receive the notification from $\mathcal{R}$'s **Step 6** and update its wallet, but $\mathcal{R}$ would finally rollback the transaction in the way similar to that in **Exceptional Situation 2**. Consequently, $\mathcal{R}$ does not receive the digital currency while $\mathcal{P}$ pays out the currency. This is a rather particular case. The method to handle it relies on online account synchronization with $\mathcal{C}$. When $\mathcal{R}$ executes online data updating sometime later, $\mathcal{C}$ would confirm that the offline transaction is failed. Then by tracking historical data from both $\mathcal{R}$ and $\mathcal{P}$, $\mathcal{C}$ could construct the same amount of digital currency and give them back to $\mathcal{P}$.

Exceptional Situation 4. Exception is caused by $\mathcal{P}$ and occurs after **Step 5** and before **Step 7**. At these points, $\mathcal{P}$ would never update its wallet, and $\mathcal{R}$ would finally rollback the transaction because of receiving no response. Therefore, the payer and payee do not realize transferring digital currency.

Exceptional Situation 5. Exception is caused by $\mathcal{P}$ and occurs after **Step 7** and before **Step 8**. At this point, $\mathcal{P}$ has already updated its wallet, but $\mathcal{R}$ would finally rollback the transaction because of receiving no response. Similarly, $\mathcal{R}$ does not receive the digital currency while $\mathcal{P}$ pays out the currency. Hence the handling method is identical to that in **Exceptional Situation 3**.

Exceptional Situation 6. Exception is caused by $\mathcal{P}$ and occurs after **Step 8** and before **Step 9**. At this point, $\mathcal{P}$ has already updated its wallet, and $\mathcal{R}$ could receive the successful notification from $\mathcal{P}$. As a consequent, the transaction could actually finish and succeed.

Except for the 6 kinds of exceptional situations mentioned above, there would be several extreme cases. For example, both mobile devices of $\mathcal{P}$ and $\mathcal{R}$ crash simultaneously, or either $\mathcal{P}$ or $\mathcal{R}$ crashes just at the critical point during the period of executing one step. In practice, these cases may happen with a very low frequency. And to tackle with these, the method of software engineering, such as the instance and crash recovery from program execution backup, can play an important role to help two sides of mobile devices come back to one of above 6 exceptional situations. Additionally, similar to missing physical cashes in real life, there is a risk that the digital currency in the process of dual offline transaction could be also lost. For example, when a mobile device completely breaks down just after it receives new digital currency and has not synchronized them to online account, the currency may be lost for its user temporarily. A mechanism that the central bank periodically summarizes the global data, could help recovering the missing digital currency.

3.4 Security Analysis

We provide an informal security analysis to argue that DOT-M satisfies the six security properties, three of which are described in Sect.1.1 as the design principles, if the security assumptions given in Sect.2.3 hold, the digital signature algorithm we use is EUF-CMA secure [37], the bottom message authentication code (MAC) is unforgeable against chosen message and chosen verification queries attack (uf-cmva) [38], and the bottom asymmetric and symmetric cryptography algorithms are IND-CPA secure. Thus, the basic confidentiality and authenticity of underlying protocols and system could be proved in the standard way. Below, we interpret why DOT-M satisfies the specific security properties.

3.4.1 Preventing Man-in-the-middle Attack

Because there is the bottom secure communication channel between legitimate entities according to **Assumption 4**, the adversary hardly obtain valuable plain texts or forge effective data by exploring communication channel. Even if the adversary tries to simply forward data to spend other's currency, the transaction message σ clearly records and represents the exact identities of payer and payee. Protected by digital signature, the message σ cannot be forged or tampered by the adversary.

3.4.2 Preventing Intruder Attack

Enhanced by SE and TEE technology, the mobile device becomes a trusted terminal on our trusted architecture design in the light of **Assumption 1**. Although the adversary is able to intrude the device through REE interfaces or even execute the laboratory-level attack on SE, it is impossible for him to acquire the vital keys or impact the program execution of the core transaction protocol. The significant program process is executed by Wallet Trustlet and DOT-M Service which are issued by the central bank and also protected by TEE. Moreover, simulating a transaction program process in REE to execute a malicious transaction is unworkable. According to **Assumption 3**, the predefined trusted relationship through SE and TEE cannot be bypassed. Thus, the adversary cannot attack the dual offline transaction by intruding a legitimate user.

3.4.3 Preventing Malicious Payer or Payee

Because of the similar theory, acting as payer or payee, an originally legitimate but currently malicious user hardly manipulate the significant procedure of the dual offline transaction even on his own device to execute a malicious transaction. As a result, the normal transaction protocol cannot be disobeyed by a malicious user. The only feasible means from the user is to deliberately shut down the device when it executes the important steps of the transaction. However, it will be considered as one of exceptional situations described in Sect.3.3.5 to tackle with.

3.4.4 Preventing Double-spending

In our scheme, the unique series number of digital currency signed by $\mathcal{C}$ and the transaction message signed by $\mathcal{P}$'s wallet ensure and differentiate the payment of each digital currency block. In addition, the offline transaction record within data structure of currency block is also signed by $\mathcal{P}$'s $wsk_{\mathcal{P}}$. Because the keys and vital operations are protected by SE and TEE inside the mobile devices, the payer himself hardly tampers with those sensitive data. Meanwhile, according to **Assumption 2** and **Assumption 3**, the digital signature from the exact wallet to pay out the currency could not be forged, falsified or refused. Even if $\mathcal{P}$ pays out the same digital currency repeatedly, $\mathcal{C}$ could detect the double-spending attack when the same currency is synchronized and submitted by the same or different payees.

3.4.5 Non-repudiation

Similarly, DOT-M utilizes security mechanisms such as digital signature and hardware-based mobile technology described in Sect.3.1.1. During the dual offline transaction protocol, in **Step 3** $\mathcal{P}$ signs the payment behavior and the node of transaction record using his $wsk_{\mathcal{P}}$ and $usk_{\mathcal{P}}$. And in **Step 6**, $\mathcal{R}$ signs the receipt behavior using his $usk_{\mathcal{R}}$. Thus, combined with the verifiable transaction message in protocol, the trusted mobile device and the verifier would not accept user's malicious behaviors, although they are willing to repudiate their operations.

3.4.6 Unforgeability

Firstly, for the unforgeability of identity, SE and TEE can protect not only the private key of the user and the wallet, but also the signing executions performed by the authorized user. The detailed description of the security mechanism can be found in Sect.3.1. Moreover, the signature algorithm [39] we use makes sure that the adversary cannot obtain the private key of the user and the wallet, and then is unable to forge the identities of them. Secondly, for the unforgeability of digital currency, the digital currency is signed by the central bank's private key which is well preserved, and every mobile device could verify the digital currency by $pk_{\mathcal{C}}$. Similarly, each node of transaction record is signed by $\mathcal{P}$ and would be verified one by one by $\mathcal{R}$. As a consequence, the forged digital currency or transaction records are hardly accepted by other payees' devices.

4 Implementation and Evaluation

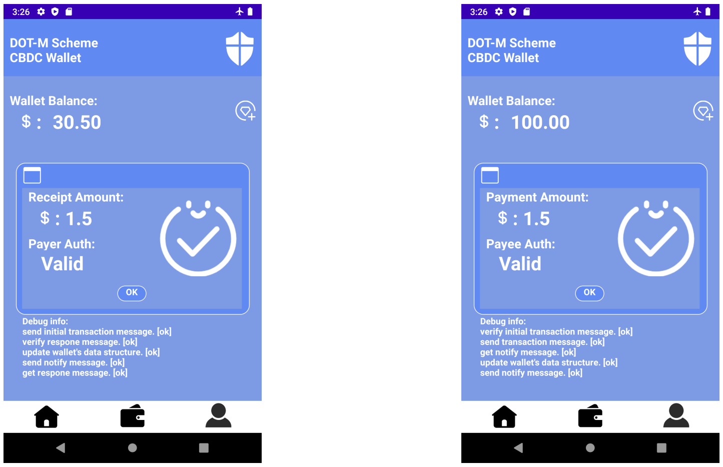
In this section, we first present the prototype of DOT-M from both aspects of hardware and software. Afterwards, we show the efficiency and performance evaluation of the proposed scheme based on our prototype system.

4.1 Implementation

For simulating the complete dual offline transaction process, we use one PC platform as the central bank server and implement the CBDC background service on this platform, which can apply two-way authentication with the wallet application, generate or destroy digital currency, and issue certificates and wallet parameters etc. Moreover, the implementation also involves the simulation of mobile device, which is described as follows.

Hardware Platform. We utilize two Hikey-960 development boards whose operating systems are Android 9.0 to simulate the mobile devices of two parties. The HiKey-960 development board is based around the Kirin-960 processor with four ARM Cortex-A73 and four Cortex-A53 cores. The board is equipped with 3GB of LPDDR4 SDRAM memory and 32GB of UFS 2.0 flash storage. The function of TEE is supported and enabled by the hardware on the board. Limited by the lack of NFC hardware resources on the HiKey-960, in order to simulate NFC communication that conforms to the ISO14443 protocol with the maximum transmission rate of 424 kbps, we choose to use serial communication at the same rate as NFC to transmit data between devices. In addition, we adopt STM32F103 module to act as the security chip (i.e. SE).

Software Implementation. For the software implementation of DOT-M on the mobile device, we respectively develop Wallet in Android, the trusted services and Wallet Trustlet in OP-TEE which is compliant with GP's TEE Specifications [40,41,42]. For the cryptographic algorithms used in our scheme, such as the SM series algorithms, we implement them under the Specifications [39,43,44] based on GmSSL 3.0 [45] static library and at the security level of 256-bit. The establishment of secure communication channel between mobile devices or between the device and the central bank, is developed by using the TLS1.2 protocol from GmSSL 3.0 library. 7053 lines of code (LOC) in C language totally comprise our components and auxiliary functions in TEE and SE. Besides, we program one test application that could execute upon DOT-M scheme. It contains 896 LOC for Wallet running in NW and 736 LOC for Wallet Trustlet in SW. Figure 9 shows a runtime instance of our implementation at the point of accomplishing a dual offline transaction for both payee and payer. Due to the maturity of TUI technology in the current mobile industry, we omit its implementation when simulating the dual offline transaction.



DOT-M Scheme
CBDC Wallet

Wallet Balance:
$: 30.50

Receipt Amount:
$: 1.5
Payer Auth:
Valid

Debug info:
send initial transaction message. [ok]
verify respone message. [ok]
update wallet's data structure. [ok]
send notify message. [ok]
get respone message. [ok]

DOT-M Scheme
CBDC Wallet

Wallet Balance:
$: 100.00

Payment Amount:
$: 1.5
Payee Auth:
Valid

Debug info:
verify initial transaction message. [ok]
send transaction message. [ok]
get notify message. [ok]
update wallet's data structure. [ok]
send notify message. [ok]

Fig. 9. A runtime instance of the implementation for the payee and payer respectively.

4.2 Efficiency and Performance Evaluation

Since the resource-constrained mobile device is the performance bottleneck as well as the focus of user's attention, we measure the performance of DOT-M on the prototype system revolving around mobile devices especially for the frequent process of dual offline transaction. In our scheme, the time costs of the bottom establishment process for secure communication channel between devices is basically fixed and limited. Then a dual offline transaction requires the payer and payee to interact with each other twice and complete the 9 steps together. The transaction time overhead is significantly affected by three operations: signing, verifying, and data communicating. Decided by the data length, the time overhead of communicating does not fluctuate dramatically because the speed of

NFC or Bluetooth is quite enough for transferring necessary data in offline transaction. However, the number of signing and verifying operations are positively correlated with both the number n_b of currency blocks used in the transaction and the total number n_d of nodes contained in the linked lists of all the offline transaction records. In theory, the time overhead of the transaction may obviously grow along with the increasing number of these two cryptographic operations. Table 3 illustrates the theoretical numbers of signing and verifying in different critical steps.

Table 3. Numbers of signing and verifying in different critical steps.

	Signing	Verifying
Step 1.	2	/
Step 2.	/	2
Step 3.	1+ n_b	/
Step 4.	/	1+n_b + n_d
Step 6.	1	/
Step 7.	/	1

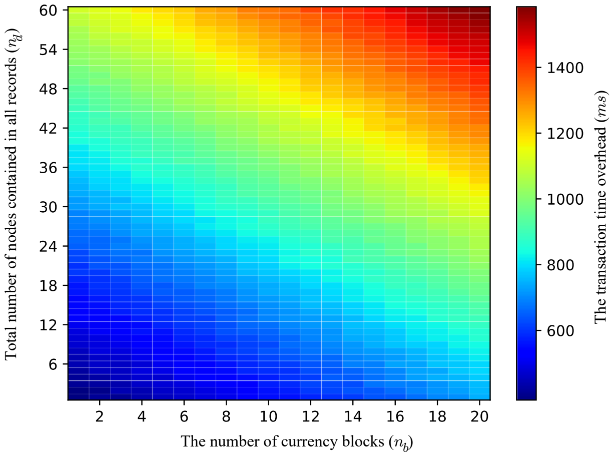


Fig. 10. The colormap of the transaction time overheads (ms).

We design an experiment to evaluate our scheme's total time overheads of the whole transaction between the payer and payee under the conditions of different n_b and n_d which also decide the data length to communicate, as shown in Figure 10. Each average experimental result is taken over 100 test-runs. In the figure, the points of time overhead located in the top right corner have more time-consuming than those in the bottom left corner. The statistical results show that as n_b and n_d increase respectively, the time overheads of completing a transaction gradually increase and n_b has more influence than n_d on transaction time. It is recommended to select the combination of parameters n_b and n_d to make the points of time overhead locate in the more cooling blue area in the figure. Generally, the users would be rather satisfied if the transaction time is less than 1000 milliseconds (ms). The optional parameter combinations of (n_b, n_d) such as (6, 42), (10, 36), (16, 24) and (20, 18) all meet the requirement of normal use. In some extremely bad cases such as combination (20, 60), the time overhead is 1590 ms which is also completely acceptable for the mobile users. Furthermore, from our experiment, when the combination changes from (1, 1) to (20, 60), we find that the time overhead of data communicating just grows from 112 ms to 568 ms. It is rather stable. According to our efficiency analysis and experimental results, DOT-M can be considered as a reasonably efficient scheme for dual offline transaction on mobile devices.

5 Conclusion

In this paper, we propose DOT-M, a complete and practical dual offline transaction scheme for mobile devices using SE and TEE. DOT-M takes both security and efficiency specially for mobile users. The scheme supports transactions that can be completed when both the mobile devices of payer and payee are offline. Additionally, the exceptional situations are analyzed, and specific processing methods for each situation are given. In the meantime, six necessary security properties of the transaction protocol are guaranteed. Our implementation and evaluation convince that DOT-M is quite practical for dual offline transaction through mobile devices.

References

1. Ready, steady, go?-Results of the third BIS survey on central bank digital currency, https://www.bis.org/publ/bppdf/bispap114.htm/. Last accessed 2 Feb 2022
2. The digital dollar project exploring a US CBDC, https://www.banking.senate.gov/imo/media/doc/Giancarlo%20Testimony%20Addendum%206-30-202.pdf. Last accessed 4 Feb 2022
3. Report on a digital euro, https://www.ecb.europa.eu/pub/pdf/other/Report_on_a_digital_euro~4d7268b458.en.pdf/. Last accessed 4 Feb 2022
4. Progress on research and development of E-CNY in China, http://www.pbc.gov.cn/en/3688110/3688172/4157443/4293696/2021071614584691871.pdf. Last accessed 2 Feb 2022
5. Today's cryptocurrency prices by Market Cap, https://coinmarketcap.com. Last accessed 15 Jul 2021
6. Libra white paper, https://wp.diem.com/en-US/wp-content/uploads/sites/23/2020/04/Libra_WhitePaperV2_April2020.pdf/. Last accessed 10 May 2022
7. Gupta, Y.K., Jeswani, G., Pinto, O.: M-Commerce offline payment. SN Computer Science **3**(1), 1–11 (2022)
8. Kutubi, M.A.A.R., Alam, K.M.R., Morimoto, Y.: A simplified scheme for secure offline electronic payment systems. High-Confidence Computing **1**(2), 100031 (2021)
9. Christodorescu, M., et al.: Towards a two-tier hierarchical infrastructure: an offline payment system for central bank digital currencies. arXiv preprint arXiv:2012.08003 (2020)
10. Zhong, L., Wu, Q.H., Xie, J., Li, J., Qin, B.: A secure versatile light payment system based on blockchain. Future Generation Computer Systems **93**, 327–337 (2019)
11. Dmitrienko, A., Noack, D., Yung, M.: Secure wallet-assisted offline bitcoin payments with double-spender revocation. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 520–531. (2017)
12. Ivanov, N., Yan, Q.B.: System-wide security for offline payment terminals. In: International Conference on Security and Privacy in Communication Systems, pp. 99–119. Springer (2021)
13. EMV Mobile Payment software-based Mobile Payment Security Requirements version 1.4, https://www.emvco.com/
14. Payment Card Industry (PCI) contactless payments on COTS (CPoC^{TM}) security and test requirements version 1.0, https://listings.pcisecuritystandards.org/documents/CPoC_Program_Guide_v1.0.pdf?agreement=true&time=1575480166972
15. Payment Card Industry (PCI) software-based PIN entry on COTS (SPoC^{TM}) magnetic stripe readers annex security and test requirements version 1.1, https://docs-prv.pcisecuritystandards.org/SPoC/Standard/SPoC_MSR_Annex-v1.1.pdf
16. Bhattacharyya, A., Setua, S.K.: Design of ECSEPP: elliptic curve based secure e-cash payment protocol. In: Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, pp. 337–345. Springer (2016)
17. Islam, A., et al.: An online E-Cash scheme with digital signature authentication cryptosystem. In: Sustainable Communication Networks and Application, pp. 29–39. Springer (2021)
18. David, C.: Blind signatures for untraceable payments. In: Advances in cryptology, pp. 199–203. Springer (1983)
19. Xie, S.C., Niu, X.F., Zhang, J.Z.: An improved quantum e-payment system. International Journal of Theoretical Physics **59**(2), 445–453 (2020)
20. Wen, X.J., Chen, Y.Z., Fang, J.B.: An inter-bank e-payment protocol based on quantum proxy blind signature. Quantum information processing **12**(1), 549–558 (2013)
21. Liu, W.C.: The impact of digital currency on foreign exchange management and countermeasures. In: International Conference on Application of Intelligent Systems in Multi-modal Information Analytics, pp. 87–92. Springer (2021)
22. Yao, Q.: Experimental study on prototype system of central bank digital currency. Journal of Software **29**(9), 2716–2732 (2018)
23. Li, S.N., et al.: Research on offline transaction model in mobile payment system. In: International Conference on Frontier Computing, pp. 1815–1820. Springer (2018)
24. Poon, J., Dryja, T.: The bitcoin lightning network: scalable off-chain instant payments. (2016)
25. Batten, L., Yi, X.: Off-line digital cash schemes providing untraceability, anonymity and change. Electronic Commerce Research **19**(1), 81–110 (2019)
26. Micallef, S., Konstantinos Markantonakis, I.S.G.: Mobile payments using Host Card Emulation with NFC: security aspects and limitations. ISG MSc Information Security thesis series (2018)
27. Armando, A., Merlo, A., Verderame, L.: Trusted host-based card emulation. In: 2015 International Conference on High Performance Computing & Simulation (HPCS), pp. 221–228. IEEE (2015)
28. GlobalPlatform Technology secure element protection profile version 1.0, https://globalplatform.org/specs-library/secure-element-protection-profile/

29. Yang, B., Feng, D.G., Qin, Y., Zhang, Y.J.: Secure access scheme of cloud services for trusted mobile terminals using TrustZone. Journal of Software **27**(6), 1366–1383 (2016)
30. Yang, B., Feng, D.G., Qin, Y.: A lightweight anonymous mobile shopping scheme based on DAA for trusted mobile platform. In: 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 9–17. IEEE (2014). https://doi.org/10.1109/TrustCom.2014.6
31. Yang, B., Yang, K., Qin, Y., Zhang, Z.F., Feng, D.G.: DAA-TZ: an effcient DAA scheme for mobile devices using ARM TrustZone. In: International Conference on Trust and Trustworthy Computing, pp. 209–227. Springer (2015)
32. Zhao, S.J., Zhang, Q.Y., Hu, G.Y., Qin, Y., Feng, D.G.: Providing root of trust for ARM TrustZone using On-Chip SRAM. In: Proceedings of the 4th International Workshop on Trustworthy Embedded Devices, pp. 25–36. ACM (2014)
33. ARM Security Technology building a secure system using TrustZone technology, https://documentation-service.arm.com
34. Proxama, http://www.proxama.com/platform/. Last accessed 23 October 2020
35. Canard, S., Pointcheval, D., Sanders, O., Traore, J.: Divisible e-cash made practical. In: Public-Key Cryptography (PKC), pp. 77–100. Springer (2015)
36. Yang, B., Yang, K., Zhang, Z.F., Qin, Y., Feng, D.G.: AEP-M: practical anonymous e-payment for mobile devices using ARM TrustZone and divisible e-cash. In: International Conference on Information Security 2016. LNCS, pp. 130–146. Springer (2016)
37. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing **17**(2), 281–308 (1988)
38. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: EURO-CRYPT 2012, pp. 355–374. Springer (2015)
39. ISO/IEC 14888-3:2018 IT Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms. (2018)
40. GlobalPlatform Technology TEE system architecture version 1.2, https://globalplatform.org/specs-library/tee-system-architecture/
41. GlobalPlatform Technology TEE client API specification version 1.0, https://globalplatform.org/specs-library/tee-system-architecture/
42. GlobalPlatform Technology TEE internal core API specification 1.3.1, https://globalplatform.org/specs-library/tee-internal-core-api-specification/
43. ISO/IEC 10118-3:2018 IT Security techniques — Hash-functions — Part 3: Dedicated hash-functions. (2018)
44. ISO/IEC 18033-3:2010/AMD 1:2021 Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers — Amendment 1: SM4. (2018)
45. Guan, Z.: The GmSSL Project, http://gmssl.org. Last accessed 15 Apr 2021