Dual-Anonymous Off-Line Electronic Cash for Mobile Payment

Jianbing Ni, *Member, IEEE*, Man Ho Au, *Member, IEEE*, Wei Wu, Xiapu Luo, Xiaodong Lin, *Fellow, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Mobile devices have become near-ubiquitous tools in our daily lives. Following this trend, mobile commence is developed rapidly which in turns stimulates interests in mobile payment. Some prominent examples include Google's Wallet, WeChat Pay, and Apple Pay. Most of these technologies, however, are designed for users to be able to pay conveniently to the business. In other words, they are designed with the business to user model in mind. Besides, an active network connection with an external payment server is required either from payer or payee during transaction. Our work intends to supplement existing solutions, which allows payment to be made in an off-line and dual-anonymous manner. In doing so, a dual-anonymous off-line electronic cash scheme is proposed by utilizing BBS+ signature. The feature of our scheme is dual-anonymous payment, which means that both the payer and the payee in any transaction cannot be identified even all other users and the payment server collude. Through security proof and performance analysis, we also demonstrate that the security of the proposed scheme can be reduced to standard assumptions and it is suitable for applications in mobile commerce.

Index Terms—Mobile computing, electronic cash, mobile commerce, electronic payment, anonymity

1 Introduction

The proliferation of mobile devices and mobile applications is radically changing the way we work and live now [1], [2]. Following this trend, mobile commerce, known as a new generation of e-commerce, is developed rapidly, which allows users to buy and sell goods and services through wireless handheld devices such as smart phones and tablets. It is forecasted that mobile commerce would be worth $250 billion in 2018, a more than 300% growth over a four-year time span. As one of the most significant components in mobile commerce, mobile payment, where mobile devices are payment tools, catches much attention of both industry and academia in the past decade.

Mobile payment enables a mobile device to initiate, authorize, and confirm an exchange of currency in return for goods and services [3]. Various types of payment systems supporting electronic transactions are being used nowadays, including card-based payments, carrier billing, contactless payments and bank transfers [4]. Most of these technologies require the use of credit/debit cards or preregistration at an online payment system, thus bypassing banks and credit card centers altogether. The most commonly used mechanisms are Paypal, Credit Card, and carrier billing. E-wallet is an emerging technology that makes use of mobile devices to store and control users' online shopping information, such as logins, passwords, shipping address and credit card details, and purchase products without showing credit cards. The typical examples of e-wallet are Google's Wallet, Microsoft's E-Wallet, WeChat Pay, and Apple Pay. They offer a convenient and technologically quick method for users to purchase products from merchants or stores.

One distinctive feature of existing mobile payment systems is that transactions are handled by an external payment server. Consequently, there are some restrictions on their applications. For instance, it is unsuitable to be adopted as a payment method for a transfer service serving an area with poor Internet connection. All the transactions should be online that mobile devices should have stable Internet connections with the payment server. However, in reality, it is impossible to guarantee that stable Internet connection is in place when needed, especially at some specific scenarios and areas, such as on a high-speed train, on a cruise ship, at a crowded shopping mall, on rural and remote community. Furthermore, the data service while roaming is expensive. Therefore, off-line payment is quite critical to be a complement of online transaction in mobile payment. In addition, the existing technologies do not offer user's identity privacy, indicating that user's identity would be disclosed to the payment server and other untrusted entities, such as merchants. Consequently, a user's privacy is

- *Jianbing Ni is with the Department of Electrical and Computer Engineering and Ingenuity Labs Research Institute, Queen's University, Kingston, Ontario K7L 3N6, Canada. E-mail: jianbing.ni@queensu.ca.*
- *Man Ho Au is with the Department of Computer Science in the University of Hong Kong, Hong Kong. E-mail: allenau@cs.hku.hk.*
- *Wei Wu is with the School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China. E-mail: weiwu81@gmail.com.*
- *Xiapu Luo is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. E-mail: csxluo@comp.polyu.edu.hk.*
- *Xiaodong Lin is with the School of Computer Science, University of Guelph, Guelph, ON N1G 2W1, Canada. E-mail: xlin08@uoguelph.ca.*
- *Xuemin (Sherman) Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: sshen@uwaterloo.ca.*

Manuscript received 4 August 2020; revised 13 November 2021; accepted 1 December 2021. Date of publication 14 December 2021; date of current version 5 May 2023.
(Corresponding authors: Xiaodong Lin and Man Ho Au.)
Digital Object Identifier no. 10.1109/TMC.2021.3135301

invaded as a large amount of personal information would be disclosed, such as habit, preference, purchase history and account information. Other known payment options include electronic cash (e-cash), which supports off-line transactions and privacy preservation for payers.

E-cash was introduced by Chaum [5] as an electronic counterpart of physical money. Extensive research [6], [7], [8], [9], [10], [11] has been done on the subject since then. In its simplest form, an e-cash system consists of three parties (the bank B, the user U and the merchant M) and four main procedures (account establishment, withdrawal, payment and deposit). The user U first performs an account establishment protocol with the bank B. The currency circulating around is quantized as coins. U obtains a coin by performing a withdrawal protocol with B and spends the coin by participating in a payment protocol with M. To deposit a coin, M performs a deposit protocol with B.

A secure and practical e-cash possesses three properties, namely, *anonymity*, *balance* and *exculpability*. *Anonymity* (also referred to as privacy), is a distinctive feature of cash payments that offer a user. It means that payments do not leak the users' whereabouts, spending patterns or personal preferences. *Balance* means that no collusion of users and merchants together can deposit more than they withdraw without being detected. Finally, *exculpability* refers to the fact that honest spenders cannot be accused to have double-spent. Spending the same coin twice, also known as *double-spending*, is a prominent example of misbehavior. Existing e-cash schemes tackle this dilemma by incorporating mechanisms such that spending a coin twice provides sufficient information for everyone to compute the user's identity.

Many e-cash schemes [12], [13], [14], [15], [16], [17], [18], [19] are designed to achieve secure electronic payment with privacy protection for payers and detection of double spenders. However, privacy guarantee on both payer and payee is offered by physical cash. In reality, the identity of the payee, who receives the e-cash, is prevented from being disclosed to others. For example, when a buyer purchases goods, the buyer does not concern the identity of the merchant but whether the goods he/she bought match the money paid. The merchant's identity is preserved, as long as the merchant gives the equivalent goods to the buyer. Therefore, the difference between physical cash and electronic cash has not been bridged. Furthermore, the existing e-cash schemes are designed for one-way transaction and it is unknown how a user can be acted as a payee. Nevertheless, in a transfer, the payee's privacy should be drew the same attention as the payer's. The leakage of payee's identity would expose a large amount of sensitive information about the payee, such that an attacker can link a transfer with a payee, and thereby predict the incentive of the transfer and the personal information about the payee. Therefore, to preserve the payee's privacy is crucial for the development of electronic cash.

Physical cash is able to prevent payee's identity disclosure because of its unique property that receivers can transfer cash with privacy protection without direct involvement of banks. However, in traditional e-cash, the payee has to contact the bank for deposit after a transaction, such that the bank is easy to identify the payee's identity colluding with the payer. Although transferable e-cash allows users to be able to transfer coins between each other multiple times before deposit, as happens with physical cash, the bank can distinguish the identity of the last payee, who deposits the transferred e-cash. Moreover, transferred e-cash grows in size, such that the communication and storage efficiency is low. Therefore, how to prevent colluding payers and bank from linking the transactions and the identities of payees is an open problem [15].

Our Contribution. In this paper, we propose an efficient and dual-anonymous off-line electronic cash scheme to bridge the difference between physical cash and electronic cash. The innovation is the design of a randomization protocol which allows a user to re-randomise his coin without revealing the identity, so that when the coin is deposited back to the payment server, it cannot be linked to any transaction. Therefore, the proposed scheme is an efficient solution for the open problem demonstrated by Blanton *et al.* [15] in the off-line manner. Furthermore, the coin is "transferable" without the increase of its size as long as the payee fleshes the received coin with the payment server by executing the randomise protocol at his convenient time in an anonymous way. In addition, our scheme allows a payer and a payee to make a transaction at the area with poor Internet connection via device-to-device communication, such as Bluetooth, and the payee can exchange the transferred coins with the payment server when the Internet is connected. We also formalize the security model for such a scheme and prove that our construction achieves payment balance, payer anonymity, payee anonymity and exculpability under the random oracle model. Finally, we demonstrate our scheme is efficient and practical to be adopted in mobile electronic commence.

Paper Outline. The remainder of this paper is organized as follows. In Section 2, we review the related works about e-cash and mobile payment approaches. In Section 3, we formalize syntax and the security requirements for dual-anonymous off-line e-cash scheme. In Section 4, we present cryptographic tools that are used in our construction. Then we propose the details of our scheme in Section 5, followed by security proofs in Section 6. We analyze the performance of our scheme in Section 7 and draw the conclusion in Section 8.

2 Related Works

In this section, we review the existing anonymous payment systems, including centralized and decentralized systems.

2.1 Centralized Payment Systems

The typical anonymous centralized payment system is e-cash [5], in which a centralized bank plays the role of account management and coin transaction. The practical e-cash system should be at least offline and payer anonymity. Most e-cash schemes [12], [13], [14], [15] are based on the blind signatures or their variations. The bank can sign on the information associated with the coins in a blind way in coin withdrawal and the zero-knowledge proofs are utilized to prove the validity of the blind signatures without exposing payers' identities during transactions. However, payer anonymity [20] results in the difficulty of tracing the misbehaving payers who double-spend the coins. Therefore, coin

traceability of anonymous payers [6], [9], [10], [11], [12], [13], [14] is achieved to prevent double-spending and many other crimes without threatening the privacy of honest payers. Moreover, the digital counterpart of cash offers desired features over traditional paper cash, so as to be classified into different categories, such as compact e-cash [11], [12], [17], [21], [22], divisible e-cash [7], [8], [9], [10], [13], [14], [23], [24], [25], and transferable e-cash [26], [27], [28], [29], [30].

The goal of compact e-cash is to improve the efficiency of coin withdrawal and coin storage by enabling a user to withdraw multiple coins in a single operation. Camenisch *et al.* [22] proposed the first compact e-cash scheme that reduces the computational complexity of coin withdrawal of 2^n coins to $O(n+k)$ and keeps the storage space of the coins to $O(n+k)$ bits, where k is a security parameter. Due to this appealing feature, compact e-cash attracted many follow-up research that aim to further improve the efficiency. Au *et al.* [12] constructed a compact e-cash scheme from the bounded accumulator, which supports more efficient coin tracing. Au *et al.* [21] later proposed a compact spending protocol and a batch spending protocol to allow the user to spend multiple denominations of coins efficiently without performing the spend protocol a number of times. Subsequently, Märtens [17] presented a compact e-cash scheme which allows users to withdraw a wallet with arbitrary number of coins, while the spending protocol is of constant time and space complexity. Recently, Lian *et al.* [11] proposed a compact e-cash scheme with full-tracing to improve the efficiency of coin tracing without the trusted third party. The computational complexity is reduced from $O(k)$ to $O(1)$ with low storage cost of coins. These compact e-cash schemes save storage cost of coins and improve computational efficiency of spending and coin tracing, but they only guarantee the payer privacy.

To improve the efficiency of coin spending, Okamoto [23] proposed the first practical divisible e-cash, in which the user can withdraw a coin of monetary value 2^n and then spend this coin in several times by dividing the coin value. The user can efficiently spend a coin of 2^l, where $0 < l \leq n$, such that divisible spending is more efficient than repeating the spending protocol 2^l times. Canard and Gouget [13] constructed the first anonymous divisible e-cash system based on complex non-interactive zero-knowledge proofs. Au *et al.* [14] proposed a more efficient scheme at the unconventional security model which only ensures that the bank would not loose money "on average". An improvement was proposed by Canard and Gouget [24] with efficient spending, but the utilized tree structure suffers from the downside of efficiency. To improve the efficiency of tree structure, Canard *et al.* proposed the first efficient divisible e-cash system [7] secure in the standard model based on a public global tree structure and an anonymous divisible e-cash scheme [8] with constant-time withdrawal and spending from advanced tree representation. In recent years, Pointcheval *et al.* [9] and Bourse *et al.* [10] got rid of the tree structure and designed the divisible e-cash systems from a unique coin's structure and constrained pseudo-random functions, respectively. Deo *et al.* [25] further proposed the first compact e-cash system based on lattices that achieves the desirable properties of balance, anonymity, traceability, and strong exculpability.

The transferable e-cash can improve the efficiency of coin deposit because it allows a user to re-transfer the obtained coin offline, instead of depositing it at the bank once receiving. Canard and Gouget [26] proposed the first anonymous transferable e-cash scheme in which the user cannot recognize a coin she has already obtained when receiving it again. Fuchsbauer *et al.* [27] designed a more practical one, which suffers from the weakness that all users have to give up their anonymity if one of the users double-spends the received coin. Zhang *et al.* [28], [29] proposed a transferable conditional e-cash scheme, which is not truly anonymous [13] as a trusted third party is required to reveal the identity of a double-spender. This weakness is fixed by Baldimtsi *et al.* [30]. They proposed a generic construction of anonymous transferable e-cash schemes, but the instantiation is inefficient since the transferred e-coin grows in size, and the server has to hold a number of global information. Therefore, they are not suitable for the applications in mobile commerce environments.

However, most of aforementioned anonymous e-cash schemes only achieve the anonymity of the payers. Blanton *et al.* [15] demonstrated the issue of preserving the anonymity of the payee and presented an open problem that how to preserve the payee's identity against the collusion of payers and the bank. They proposed a trivial solution to achieve payee anonymity in which the payee presents the transcript to the bank in exchange for physical cash directly. Nonetheless, this requires an anonymous physical channel. Baldimtsi *et al.* [30]'s transferable e-cash can provide both payer privacy and payee privacy, but it is inefficient in computation and coin storage. In this work, we propose a dual-anonymous off-line e-cash scheme that achieves the anonymity of both payers and payees, and is highly efficient than the existing ones. To demonstrate the advantages of our scheme clearly, we give the comparison about several important features, including off-line payment, dual anonymity, efficiency, security assumption and security model, with the existing work in Table 1. Concurrent with our work, Bauer *et al.* [31] demonstrated the weaknesses of anonymity and efficiency in the scheme [30] and proposed the first concrete construction of transferable e-cash based on bilinear groups.

2.2 Decentralized Payment Systems

With the development of Bitcoin, many decentralized payment systems are designed from Bitcoin and enhance the privacy preservation for users from pseudonymity to full anonymity. Miers *et al.* [32] introduced Zerocoin, an anonymous distributed e-cash from Bitcoin to allow fully anonymous currency transactions. Ben-Sasson *et al.* [33] proposed a decentralized anonymous payment system named Zerocash from Bitcoin, which constructs a full-fledged ledger-based digital currency that achieves the anonymity of both payers and payees. Subsequently, by extending Zerocash, Garman *et al.* [34] designed new decentralized anonymous payment systems that can enforce compliance with specific transaction policies. Jivanyan [35] proposed a new anonymous payment system, called Lelantus, which ensures transaction confidentiality and user anonymity with small proof sizes, short verification cost, and no trusted setup.

TABLE 1
Comparison With Related Work

Features	Off-Line	Dual Anonymity	Security Assumptions	Security Model	Efficiency	Stateless[1]	Cheater Detection (no TTP)[2]
[10]	√	×	PRF	Random Oracle	High	√	√
[11]	√	×	S-RSA, DDH, q-DDHI	Random Oracle	High	√	√
[15]	√	♮[3]	LRSW and DDH	Random Oracle	High	√	√
[18], [19]	√	×	PUF and RSA	Random Oracle	High	√	×
[20]	×	×	LRSW and DDH	Random Oracle	High	√	√
[25]	√	×	LWE	Random Oracle	High	√	√
[27]	√	×	CDH, HLIN, and q-DHSDH	Standard	Low	√	×
[28], [29]	√	♮[4]	SXDH, q-ADH-SDH and AWFCDH	Standard	Low	√	×
[30]	√	√	DDH and DLIN	Standard	Low	×	√
[31]	√	√	SXDH, q-ADHSDH, and AWFCDH	Standard	Low	√	√
[32], [35]	×	√	Strong RSA and DL	Random Oracle	High	√	√
[33], [34]	×	√	CDH and DDH	Standard	Low	√	√
[36]	×	√	DL and RSA	Random Oracle	High	√	√
[44]	×	√	DL and CDH	Standard	Low	×	√
[46]	×	√	PDL	Random Oracle	High	×	√
[48]	√	×	Secure Hardware	Random Oracle	High	√	√
Ours	√	√	q-SDH and DDH	Random Oracle	High	√	√

[1]*The payment server should maintain global variables, including a user list, a coin list and a deposited coins list. The user list keeps all information about users, keys and certificates. The coin list keeps all information about the state of coins in the system, such as withdrew, corrupted, transferred. The deposited coins list keeps the list of the deposited coins.*

[2]*A trusted third party (TTP), such as judger or authority, is required to identify double spenders.*

[3]*[15] mentioned payee anonymity could be achieved in a trivial manner if the payee presents the transcript to the bank in exchange for physical cash directly. Nonetheless, this requires an anonymous physical channel. Since this involves physical properties, no security model or proof is given.*

[4]*[28], [29] is not truly anonymous as it relies on a trusted party, capable of revealing identity of any spender, to handle double-spending.*

Tewari and Hughes [36] proposed a fully anonymous transferable e-cash scheme from the blind signature and the blockchain [37], which preserves the identities of payers and payees. Lin *et al.* [38] reconstructed the signatures used in the transactions to achieve conditional anonymity and designed a decentralized conditional anonymous payment to enable participant anonymity and regulation. Monero [39] is a new cryptocurrency built from the ring signature and the Bulletproof. It provides the level of anonymity by hiding the source accounts from which the coins are sent among a set of other accounts. Due to the appealing feature of strong privacy preservation compared with Bitcoin, Monero is further extended to support accountability [40] and optimized to achieve a logarithmic proof size in the size of the ring [41]. Mixing is also a popular approach to provide strong anonymity in Bitcoin-based cryptocurrency. For example, CoinJoin [42] allows a group of users to randomly permute their coins without a trusted third party, and Mixcoin [43] adds an independent cryptographic accountability layer to facilitate anonymous payments using Bitcoin. Recently, Fuchsbauer *et al.* [44] designed an aggregate cash system based on Mimblewimble [45] that can aggregate transactions and protect traction values. It keeps track of available coins in the system via a ledger, so it supports online payment. Tewari *et al.* [46] introduced a transferable e-cash system using blockchain that enables users to continuously use the transferable coins for payment.

However, the transactions of these cryptocurrencies would be online since miners must scan the public ledger to approve the validity of coins. The proposed scheme supports off-line payment, which is pretty important for a payment system serving an area with poor Internet connection. It enables a payee to receive coins from a payer through device-to-device communications at a region without the Internet, and then exchange the coins with the bank when the Internet connection is covered. Currently, the central banks of many countries, e.g., China, Canada, Brazil, India, and Sweden, are developing their central bank digital currencies (CBDC) [47]. The development of CDBC has to consider the usage for minority that stay in areas with poor network coverage, e.g., at mountains. The proposed e-cash scheme is meaningful to provide an off-line solution for these countries to develop their centralized digital currencies. Dmitrienko *et al.* [48] proposed the first cryptocurrency built upon Bitcoin that requires neither payers nor payees to be online during payment. However, this cryptocurrency does not achieve full anonymity, but pseudonymity, and is based on secure hardware, such as ARM TrustZone or Intel SGX.

3 Syntax

There are two types of entities in the system, namely, a payment server and users including payers and payees. Requirements are as follows.

- Users are required to register with the payment server. They can act as payers or payees in the system.
- The payment server is not required to participate in the interaction between a payer and a payee.
- The payment server and the payee cannot identify who the payer is in a transaction. The payment server and the payer cannot identify the payee in a transaction.

A dual-anonymous off-line e-cash scheme is a tuple of eight polynomial time algorithms (Setup, KeyGen, Account Establishment, Withdrawal, Payment, Randomise, Finalise, Cheater Detection).

- Setup. This algorithm is executed by the payment server to set up the whole system. On input an unary string 1^λ, where λ is the security parameter, this algorithm outputs a key pair consisting of the public key bpk, the secret key bsk and the system parameters

$param$. The payment server keeps bsk privately and releases bpk and $param$ to the public.

- Account Establishment. This protocol is executed between the payment server and a legitimate user to establish an account. The user inputs the system parameters $param$ to generate the public-secret key pair and interacts with the payment server. At the end of successful execution, the user obtains an account that is kept privately and thereby is eligible for conducting transactions in an anonymous form.
- Withdrawal. This protocol is executed between the payment server and a legitimate user to withdraw a fresh coin. The user's input is the account and the public key. At the end of successful execution, the user obtains a valuable coin from the payment server, which is kept privately and can be used to transact with others.
- Payment. This protocol is executed between two legitimate users on behalf of a payer and a payee to transfer a coin. The payer inputs a coin and generates a proof of the coin and the payee inputs the account to convince the payer that the identity is legitimate. At the end of successful execution, the payee obtains a valid transcript, including a proof of the coin and some auxiliary information.
- Randomise. This protocol is executed between the payment server and a payee later to randomise the transcript. The payee's input is the transcript and the account. At the end of successful execution, if the coin is not double spent, the payee obtains a fresh coin that can be used for payment; Otherwise the payment server outputs the identifier of the payer, along with the proof that the payer has spent a coin twice in transactions.
- Finalise. This protocol is executed between the payment server and a payee to deposit a coin. The payee's input is the transcript and the public key. At the end of successful execution, if the coin is not double spent, the payee deposits the coin into the account; Otherwise the payment server outputs the identifier of the payer and the proof of double-spending.
- Cheater Detection. The algorithm can be executed by everyone to check if a user indeed spent a coin twice in transactions. On input two transcripts that generated from the same coin, the algorithm outputs the identifier of the double-spending payer.

3.1 Security Notions

We now describe security requirements and formal security models of a dual-anonymous e-cash scheme. The security goals are introduced as follows.

- Balance. It means that no one can deposit more than they withdraw without being detected, even users collude together. This is the most significant requirement from the perspective of the payment server. It ensures that the number of the coins circulating on the market is equal to the quantity of the coins withdrew minus deposited. More precisely, we require that the collusion of payers and payees cannot deposit more than n coins back to the payment server without being identified, if they have withdrawn n coins. That is, if they store $n+1$ coins to the payment server, this double-spender must be identified.
- Payer Anonymity. It is required that an honest payer of a given transaction cannot be identified, even all other users and the payment server collude. In particular, spending of the same payer cannot be linked.
- Payee Anonymity. It is required that an honest payee of a given transaction cannot be identified, even all other users and the payment server collude. In particular, receptions of the same payee cannot be linked.
- Exculpability. It means that no one can convince others the guilt of an honest user, even all other users and the payment server collude.

A dual-anonymous e-cash scheme is deemed to be secure if balance, payer anonymity, payee anonymity and exculpability are satisfied.

We formally define security models with respect to the aforementioned requirements. The capacity of the adversary $\mathcal{A}$ can be modeled as the following oracles.

- Acco-Estab oracle. $\mathcal{A}$ acts as a user and engages in the Account Establishment protocol to establish an account. In the end, $\mathcal{A}$ obtains an account and the oracle stores the public key of the user in a database.
- Withdrawal oracle. $\mathcal{A}$ presents the public key and engages in the Withdrawal protocol acting as a user. In the end, $\mathcal{A}$ obtains a coin.
- Payment_1 oracle. $\mathcal{A}$ now acts as a payee and requests payers to transact with him.
- Payment_2 oracle. $\mathcal{A}$ now acts as a payer and spends coins to payees.
- Randomize oracle. $\mathcal{A}$ acts as a payee and engages in the Randomise protocol to randomise transcripts. In the end, $\mathcal{A}$ obtains fresh coins and the oracle stores the transcripts that $\mathcal{A}$ randomizes.
- Finalise oracle. $\mathcal{A}$ acts as a payee and engages in the Finalise protocol to finalise transcripts. In the end, the oracle deposits the transcripts that $\mathcal{A}$ finalizes.
- Hash oracle. $\mathcal{A}$ can ask for the values of the hash function for any input.

We require that the responses from the oracles are indistinguishable from the view as perceived by the adversary in real world attacks.

Definition 1. (Game Balance)

- Initialization Phase. *The challenger $\mathcal{C}$ takes a sufficiently large security parameter λ and executes* Setup *algorithm to generate the public key bpk and the secret key bsk. $\mathcal{C}$ keeps bsk privately and sends bpk to the adversary $\mathcal{A}$.*
- Probing Phase. *$\mathcal{A}$ can ask for a polynomially bounded number of queries to the Acco-Estab, Withdrawal, Payment_1, Payment_2, Randomize, Finalise, Hash oracles in an adaptive manner.*
- End Game Phase. *Let q_{AE} be the number of queries to the Acco-Estab oracle, q_W be the number of queries to the Withdrawal oracle, q_{P_1} be the number of queries to*

the Payment$_1$ oracle, q_{P_2} be the number of queries to the Payment$_2$ oracle, q_R be the number of queries to the Randomize oracle, q_F be the number of queries to the Finalise oracle, $\mathcal{A}$ wins the game if $q_F > q_W + q_{P_1} - q_{P_2}$ and the Cheater Detection *algorithm does not output any user presented in the Acco-Estab query.*

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

Definition 2. (Game Payer Anonymity)

- Initialization Phase. *Given a sufficiently large security parameter λ, the challenger $\mathcal{C}$ generates the system parameter $param$ and sends to the adversary $\mathcal{A}$. $\mathcal{A}$ generates bpk and bsk, keeps bsk privately and sends bpk to $\mathcal{C}$.*
- Probing Phase. *$\mathcal{A}$ is only allowed to ask for Hash oracle, since $\mathcal{A}$ possesses bsk.*
- Challenge Phase. *$\mathcal{C}$ chooses two public keys PK_0 and PK_1 and sends them to $\mathcal{A}$. $\mathcal{C}$ runs the* Account Establishment *protocol with $\mathcal{A}$ acting as the payment server to obtain accounts A_0 and A_1 on behalf of PK_0 and PK_1, and $\mathcal{C}$ also runs the* Withdrawal *protocol to acquire several coins $\{C_{00}, \ldots, C_{0n}\}$ and $\{C_{10}, \ldots, C_{1n}\}$ with respect to two accounts A_0 and A_1, respectively. Then, $\mathcal{A}$ acts as a payee to ask $\mathcal{C}$ to spend coins. $\mathcal{A}$ is allowed to specify which coin to be spent, with the restriction that it cannot ask $\mathcal{C}$ to double-spend any coin. Finally, $\mathcal{C}$ randomly chooses an unspent coin C_0 from the set of A_0 and an unspent coin C_1 from the set of A_1, picks a random bit $\delta \in \{0, 1\}$ and runs the* Payment *protocol using the coin C_δ*
- End Game Phase. *The adversary $\mathcal{A}$ outputs $\hat{\delta}$, and it wins the game if $\hat{\delta} = \delta$.*

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins minus $\frac{1}{2}$.

Definition 3. (Game Payee Anonymity)

- Initialization Phase. *Given a sufficiently large security parameter λ, the challenger $\mathcal{C}$ generates the system parameter $param$ and sends to the adversary $\mathcal{A}$. $\mathcal{A}$ generates bpk and bsk, keeps bsk privately and sends bpk to $\mathcal{C}$.*
- Probing Phase. *$\mathcal{A}$ is only allowed to ask for Hash oracle.*
- Challenge Phase. *$\mathcal{C}$ chooses two public keys PK_0 and PK_1 and sends them to $\mathcal{A}$. $\mathcal{C}$ runs the* Account Establishment *protocol with $\mathcal{A}$ acting as the payment server to obtain accounts A_0 and A_1 on behalf of PK_0 and PK_1, respectively. $\mathcal{C}$ is required to run the* Payment *protocol with $\mathcal{A}$ acting as a payer to receive coins, and $\mathcal{A}$ can specify the account that receives coins. $\mathcal{C}$ also run the* Randomise *protocol with $\mathcal{A}$ acting as the payment server to randomise the transcripts, and $\mathcal{A}$ still can specify the account that $\mathcal{C}$ is used to interact. Finally, $\mathcal{C}$ randomly chooses a bit $\delta \in \{0, 1\}$ and runs the* Payment *protocol with $\mathcal{A}$ to get paid a new coin using the account A_δ generated from the identifier U_δ, and then randomises it by executing the* Randomise *protocol.*
- End Game Phase. *The adversary $\mathcal{A}$ outputs $\hat{\delta}$, and it wins the game if $\hat{\delta} = \delta$.*

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins minus $\frac{1}{2}$.

Definition 4. (Game Exculpability)

- Initialization Phase. *Given a sufficiently large security parameter λ, the challenger $\mathcal{C}$ generates the system parameter $param$ and sends it to the adversary $\mathcal{A}$. $\mathcal{A}$ generates bpk and bsk, keeps bsk privately and sends bpk to $\mathcal{C}$.*
- Probing Phase. *$\mathcal{A}$ is only allowed to ask for Hash oracle.*
- Challenge Phase. *$\mathcal{C}$ runs the* Account Establishment *protocol with $\mathcal{A}$ acting as the payment server to obtain an account and runs the* Withdrawal *protocol to acquire several coins $\{C_0, \ldots, C_n\}$. $\mathcal{A}$ then acts as a payee and asks for spending from $\mathcal{C}$. $\mathcal{A}$ is allowed to specify which coin to be used, with the restriction that it cannot request $\mathcal{C}$ to double-spend any coin. $\mathcal{A}$ also acts as the payment server and requests $\mathcal{C}$ to randomise the transcripts. In addition, $\mathcal{A}$ can corrupt any user who has account and valid coin by possessing the account and the coin.*
- End Game Phase. *$\mathcal{A}$ runs* Finalise *protocol twice or* Finalise *protocol once and* Randomise *protocol once or* Randomise *protocol twice with $\mathcal{C}$. $\mathcal{A}$ wins the game if* Cheater Detection *algorithm on these two protocols can point to an honest user that has been involved in any of the* Acco − Estab *protocol.*

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

A dual-anonymous off-line e-cash scheme is secure if no adversary can win the Games *Balance*, *Payer Anonymity*, *Payee Anonymity* and *Exculpability* with non-negligible advantage in probabilistic polynomial time.

4 Preliminaries

Notions. If S is a non-empty set, $|S|$ represents the cardinality of S and $a \in_R S$ denotes a is randomly chosen from S. $s_1||s_2$ means the concatenation of binary strings s_1 and s_2. We say that a function $g(\lambda)$ is a negligible function, if for every positive polynomial $f(x)$, there exists an integer $N > 0$ such that for all $x > N$, $g(x) < \frac{1}{f(x)}$.

Bilinear Map. Let $(\mathbb{G}, \mathbb{G}_T)$ be cyclic groups with the same prime order p and g be a generator of $\mathbb{G}$. $\hat{e} : \mathbb{G} \times \mathbb{G} = \mathbb{G}_T$ is the bilinear map if the following properties are satisfied:

- (Bilinearity). For all $x, y \in \mathbb{Z}_p$, $\hat{e}(g^x, g^y) = \hat{e}(g, g)^{xy}$.
- (Non-Degeneracy). $\hat{e}(g, g) \neq 1_{\mathbb{G}_T}$, the identity element in $\mathbb{G}_T$.
- (Efficient Computability). $\hat{e}(g^x, g^y)$ is efficiently computable for all $x, y \in \mathbb{Z}_p$.

Mathematical Assumptions. We review two cryptographic assumptions that are related to the security of our construction.

Definition 5. (Decisional Diffie-Hellman). *The Decisional Diffie-Hellman (DDH) problem in $\mathbb{G}_T$ is defined as follows:*

Given a tuple $(D, D^a, D^b, D^c) \in \mathbb{G}_T^4$, *output* yes *if* $c = ab$ *and* no *otherwise. We say that the DDH assumption in* $\mathbb{G}_T$ *holds if there is no algorithm can solve the DDH problem in* $\mathbb{G}_T$ *with non-negligible advantage in probabilistic polynomial time.*

Definition 6. (q-Strong Diffie-Hellman). *The q-Strong Diffie-Hellman (q-SDH) problem in* $\mathbb{G}$ *is defined as follows: Given a* $(q+2)$ *tuple* $(g, g_0, g_0^x, g_0^{x^2}, \ldots, g_0^{x^q}) \in \mathbb{G}^{q+2}$, *output a pair* (A, c) *such that* $A^{(x+c)} = g_0$ *where* $c \in Z_p^*$. *We say that the q-SDH assumption in* $\mathbb{G}$ *holds if there is no algorithm can solve the q-SDH problem in* $\mathbb{G}$ *with non-negligible advantage in probabilistic polynomial time.*

Proof of Knowledge. In a proof-of-knowledge protocol [49], a prover convinces a verifier that he knows a witness w satisfying some kind of relation R with respect to a known string x. That is, the prover can convince the verifier that he knows some w satisfies the relation $(w, x) \in R$. If the prover can convince the verifier in a way that the latter cannot learn anything except the validity of the relation, this protocol is called a zero-knowledge proof-of-knowledge (ZKPoK) protocol [50]. Currently, a plethora of ZKPoK protocols have been proposed, in which Σ-protocols are a special type of three-move ZKPoK protocol. They can be transformed into non-interactive Signature Proof-of-Knowledge (SPK) protocols or signature schemes that can be proven secure in random oracle model. Σ-protocols are able to be converted into 4-move perfect zero-knowledge proof-of-knowledge protocols [51].

For instance, $PK\{(x) : y = g^x\}$ denotes a Σ-protocol that proves the knowledge of discrete logarithm. That is, a prover convinces a verifier that he possesses the knowledge of $x \in \mathbb{Z}_p$ such that $y = g^x$ with respect to some $y \in \mathbb{G}$ without exposing the actual value of x. The values on the left of the colon denote the knowledge that the prover aims to prove, and the values on the right of the colon denote the publicly known values. The signature of knowledge for message $m \in \{0,1\}^*$ that is transformed from the above Σ-protocol is denoted as $SPK\{(x) : y = g^x\}(m)$, which is secure under the random oracle model due to Fiat-Shamir heuristic.

BBS+ Signature. Here we briefly review the BBS+ signature due to [52], which can be utilized to sign multi-block messages and thereby construct anonymous authentication protocols.

Let $g_0, g_1, \ldots, g_{\ell+1}$ be generators of $\mathbb{G}$. Choose $u \in_R \mathbb{Z}_p$ as the secret key of the signature scheme, and compute the public key as $G = g_0^u$.

A signature on messages $m_1, m_2, \ldots, m_\ell$ is (A, e, s), where $e, s \in_R \mathbb{Z}_p$ and $A = (g_0 g_1^{m_1} g_2^{m_2} \cdots g_\ell^{m_\ell} g_{\ell+1}^s)^{\frac{1}{u+e}}$.

This signature can be verified as: $\hat{e}(g_0 g_1^{m_1} g_2^{m_2} \cdots g_\ell^{m_\ell} g_{\ell+1}^s, g_0) \stackrel{?}{=} \hat{e}(A, G g_0^e)$.

The BBS+ signature can be proved to be unforgeable against adaptive chosen message attack under the q-SDH assumption and a ZKPoK protocol can be constructed from the BBS+ signature that allows the signer to prove the possession of message-signature pairs.

5 Construction

In this section, we describe the construction of the proposed dual-anonymous off-line e-cash scheme. The blind version of the BBS+ signature is used to establish accounts and withdraw coins for users. The bank signs on the secret key of the user in a blind way by using the BBS+ signature during Account Establishment and Withdrawal. The BBS+ signature in Account Establishment serves as the user's account and the BBS+ signature in Withdrawal is the withdrew coin of the user. To spend the obtained coin, the payer needs to prove the validity of the withdrew coin in her account based on the ZKPoK protocol without exposing the coin and the payer's identity in Payment. After obtaining the payment transcript, the payer can randomize the received coin at bank in the convenient time in Randomise, instead of directly depositing the coin at bank. In Randomise, the payee can display the payment transcript and obtain a new coin from the bank without exposing the identity. The payee can deposit the randomised coin to the bank by paying it to herself or use it for payment again, such that the bank cannot identify the payer in Finalise. The double-spending detection is based on the unique transaction identifier and the coin series number. If the coin with the same series number is spent more than once, the identifier of the spender can be identified.

5.1 High Level Description

We first provide a high level description of the proposed dual-anonymous off-line e-cash scheme, which is designed from the underlying e-cash schemes due to [12], [22].

- Setup. The payment server generates its public-secret key pair, which consists of two public-secret key pairs of the BBS+ signature, one is used to establish the accounts for users and the other is prepared for generating coins. Let $\mathbb{G}_T$ be a cyclic group with a prime order p such that DDH assumption holds and G, H be two elements in group $\mathbb{G}_T$.
- Account Establishment. A user setups an account with the payment server by executing the ZKPoK protocol derived from the BBS+ signature. The user commits the secret key u and obtains a signature (A, e, s) on u from the payment server, while the latter learns nothing about u. Upon successful execution of this protocol, the payment server stores the user's identifier U and the user who possesses the account (A, e, s, u) can prove himself as a legitimate user.
- Withdrawing a Coin. A user withdraws a coin from the payment server executing the ZKPoK protocol derived from the BBS+ signature. The user makes a commitment and authenticates the identifier U. Upon successful execution of this protocol, the user obtains a signature (B, f, t, v) from the payment server, a coin that can be used to transact with others.
- Spending a Coin. A user with account (A, e, s, u) who acts as a payer interacts with another user with account (A', e', s', u') that acts as a payee to spend a coin (B, f, t, v) in a dual-anonymous manner. The payee proves himself as a valid user without exposing the actual identifier via zero-knowledge proof. The payer validates the proof and thereby generates a unique transaction identifier R and calculates the traceable tag (S, T), and then proves the following facts anonymously.

- The payer possesses a coin (B, f, t, v), which is a valid signature of the payment server.
- S, T are computed correctly with respect to v, which is deemed as the coin series number.

If the facts above are approved, the payee stores $(\mathcal{SPK}, S, T, INFO, N, M)$, where $INFO$ is the transaction information, N is a chosen random value for the transaction and the payee's transaction identifier $M = \mathcal{G}(INFO||N)^{u'}$.

- Randomising a Coin. The payee with account (A', e', s', u') can submit a stored transcript $(\mathcal{SPK}, S, T, INFO, N, M)$ to the payment server and prove himself as a valid user without disclosing the identifier. The payment server validates the following facts.
  - $\mathcal{SPK}$ is a valid zero-knowledge proof of a coin (B, f, t, v).
  - R and N are fresh.
  - The payee has a valid account (A', e', s', u').

If the facts above are approved, the payment server generates a fresh coin $(\hat{B}, \hat{f}, \hat{t}, \hat{v})$ for the payee which can be paid to himself for deposit or to any other users. At last, the payment server checks whether the payer has double-spent the coin. If yes, it can recover the identifier of the payer by utilizing the Cheater Detection algorithm.

- Finalising a Coin. A payee with account (A', e', s', u') submits a stored transcript $(\mathcal{SPK}, S, T, INFO, N, M)$, which is generated by paying a coin $(\hat{B}, \hat{f}, \hat{t}, \hat{v})$ to himself, to the payment server to finalise the coin, along with the zero-knowledge proof of the identifier. The payment server validates the proof and accepts the deposit and then checks whether the payer has double-spent the coin. If yes, it can recover the identifier of the payer by utilizing the Cheater Detection algorithm.
- Dealing with Double Spending. If there is another $(\mathcal{SPK}', S, T', INFO', N', M')$ in the database of the payment server, the identifier of the double-spender can be computed as $(\frac{T^{R'}}{T'^{R}})^{\frac{1}{R'-R}}$.

5.2 Construction Details

We then present the construction of the dual-anonymous off-line e-cash scheme.

- Setup. Let λ be a sufficiently large security parameter. Let $(\mathbb{G}, \mathbb{G}_T)$ be the bilinear group pair with $|\mathbb{G}| = |\mathbb{G}_T| = p$, where p is a prime of λ bits. The payment server chooses generators $g, g_0, g_1, h, h_0, h_1, h_2 \in_R \mathbb{G}$ and $H_1 \in \mathbb{G}_T$. For notional convenience, we use G and H to demote $\hat{e}(g, g)$ and $\hat{e}(h, h)$ respectively. The payment server randomly generates $\alpha, \beta \in_R \mathbb{Z}_p$ and computes $W = g^{\alpha}$, $X = h^{\beta}$. The server specifies two collision-resistant hash functions $\mathcal{G} : \{0,1\}^* \rightarrow \mathbb{G}_T$ and $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p$. The public parameters $param$ are

$$(\mathbb{G}, \mathbb{G}_T, p, g, g_0, g_1, h, h_0, h_1, h_2, G, H, H_1, \mathcal{G}, \mathcal{H}).$$

The public key of the payment server is (W, X) and its secret key is (α, β).

- Account Establishment. A user establishes an account with the payment server in the following protocol.
  - The user picks random values $s''', u \in \mathbb{Z}_p$, computes the commitment $C = g_0^{s'''} g_1^{u}$ and the user's identifier $U = G^{u}$, and then sends (C, U) to the server, along with the following zero-knowledge proof expressed in Camenisch-Stalder notation [53].

$$\mathcal{PK}_1\{(s''', u) : C = g_0^{s'''} g_1^{u} \wedge U = G^{u}\}.$$

  - The server returns failure if the verification of $\mathcal{PK}_1$ returns invalid. Otherwise, the server picks $s'', e \in_R \mathbb{Z}_p$ to compute $A = (gCg_0^{s''})^{\frac{1}{\alpha+e}}$ and returns a tuple (A, e, s'') to the user. The server stores U as the identifier of the user's account.
  - The user calculates $s = s''' + s''$ and returns failure if $\hat{e}(A, Wg^{e}) \neq \hat{e}(gg_0^{s} g_1^{u}, g)$. Otherwise, the user stores the BBS+ signature (A, e, s, u) as the account.
- Withdrawal. To obtain a coin that can be spent to other users, a user with account (A, e, s, u) engages the payment server in the following protocol.
  - The user chooses random values $t', v' \in \mathbb{Z}_p$, computes the commitment $C' = h_0^{t'} h_1^{u} h_2^{v'}$. Then the user sends C' to the payment server, along with the following zero-knowledge proof.

$$\mathcal{PK}_2\{(t', u, v') : C' = h_0^{t'} h_1^{u} h_2^{v'} \wedge U = G^{u}\}.$$

  - The server returns failure if the verification of $\mathcal{PK}_2$ outputs invalid or U is not the identifier of a valid user. Otherwise, the server picks random values $t'', f, v'' \in_R \mathbb{Z}_p$ to compute $B = (hC' h_0^{t''} h_2^{v''})^{\frac{1}{\beta+f}}$, and then returns a tuple (B, f, t'', v'') to the user.
  - The user computes $t = t' + t''$, $v = v' + v''$ and returns failure if $\hat{e}(B, Xh^{f}) \neq \hat{e}(hh_0^{t} h_1^{u} h_2^{v}, h)$. Otherwise, the user stores the BBS+ signature (B, f, t, v) as a coin.
- Payment. Suppose a user with account (A, e, s, u) would like to transfer a coin (B, f, t, v) to another user with account (A', e', s', u') in a privacy-preserving manner. Let $INFO$ be a fixed-length string containing the information of the transaction.
  - The payee chooses a random value N, calculates its transaction identifier M from the transaction identifier $INFO$, the random N, and the private key u' as $M = \mathcal{G}(INFO||N)^{u'}$. He sends M and N to the payer, along with the following zero-knowledge proof.

$$\mathcal{PK}_3\left\{\begin{array}{c}(A', e', s', u') : \\ \hat{e}(A', Wg^{e'}) = \hat{e}(gg_0^{s'} g_1^{u'}, g) \wedge \\ M = \mathcal{G}(INFO||N)^{u'}\end{array}\right\}.$$

  - The payer returns failure if the verification of $\mathcal{PK}_3$ outputs invalid. Otherwise, the payer computes a unique transaction identifier R from the transaction identifier $INFO$, the random N, and the payee's transaction identifier M as $R = \mathcal{H}(INFO||N||M)$, and the traceable tag (S, T) as $S = H^{v}$ and $T = G^{u} H_1^{Rv}$, which is linked

to the payer's private key u, the unique transaction identifier R, and the coin series number v. He sends a tuple (R, S, T) to the payee, along with the following non-interactive zero-knowledge proof, again, expressed in Camenisch-Stalder notation [53].

$$\mathcal{SPK}\left\{\begin{array}{c}(B, f, t, v, u):\\ \hat{e}(B, Xh^{\hat{f}}) = \hat{e}(hh_0^t h_1^u h_2^v, h) \wedge\\ S = H^v \wedge\\ T = G^u H_1^{Rv}\end{array}\right\}(R).$$

– The payee returns failure if the verification of $\mathcal{SPK}$ returns invalid. Otherwise, the payee stores $(\mathcal{SPK}, S, T, INFO, N, M)$.

- Randomise. A payee with account (A', e', s', u') and a stored transcript $(\mathcal{SPK}, S, T, INFO, N, M)$ engages the payment server in a privacy-preserving manner to obtain a fresh coin.
  – The payee sends the transcript $(\mathcal{SPK}, S, T, INFO, N, M)$ to the payment server, along with the following zero-knowledge proof.

$$\mathcal{PK}_3\left\{\begin{array}{c}(A', e', s', u'):\\ \hat{e}(A', Wg^{e'}) = \hat{e}(gg_0^{s'} g_1^{u'}, g) \wedge\\ M = \mathcal{G}(INFO||N)^{u'}\end{array}\right\}.$$

  – The payment server returns failure if the verification of $\mathcal{PK}_3$ returns invalid. Otherwise, the server calculates the unique transaction identifier $R = \mathcal{H}(INFO||N||M)$ and validates $\mathcal{SPK}$. If the verification of $\mathcal{SPK}$ outputs invalid, the server returns failure. Otherwise, the server continues to check if R and N are fresh, if not, it rejects the request. Otherwise, it issues a new coin to the payee as follows.
    * The payee picks random values $\hat{t}', \hat{v}' \in \mathbb{Z}_p$, calculates a commitment $\hat{C}' = h_0^{\hat{t}'} h_1^{u'} h_2^{\hat{v}'}$. The payee sends $\hat{C}'$ to the server, along with the following zero-knowledge proof.

$$\mathcal{PK}_4\left\{(\hat{t}', u', \hat{v}'): \begin{array}{c}\hat{C}' = h_0^{\hat{t}'} h_1^{u'} h_2^{\hat{v}'} \wedge\\ M = \mathcal{G}(INFO||N)^{u'}\end{array}\right\}.$$

    * The payment server returns failure if the verification of $\mathcal{PK}_4$ returns invalid. Otherwise, it picks $\hat{t}'', \hat{f}, \hat{v}'' \in_R \mathbb{Z}_p$, computes $\hat{B} = (h\hat{C}' h_0^{\hat{t}''} h_2^{\hat{v}''})^{\frac{1}{\beta+\hat{f}}}$ and returns a tuple $(\hat{B}, \hat{f}, \hat{t}'', \hat{v}'')$ to the payee.
    * The payee computes $\hat{t} = \hat{t}' + \hat{t}''$, $\hat{v} = \hat{v}' + \hat{v}''$ and returns failure if $\hat{e}(\hat{B}, Xh^{\hat{f}}) \neq \hat{e}(hh_0^{\hat{t}} h_1^{u'} h_2^{\hat{v}}, h)$. Otherwise, the payee stores $(\hat{B}, \hat{f}, \hat{t}, \hat{v})$ as a randomized coin. The coin series number is updated to be $\hat{v}$ so as to break the linkage with the old series number v. The payee can pay the coin to himself for deposit or to any other users.
  – Finally, the payment server further checks if S is fresh. If there exists another transcript in the database $(\mathcal{SPK}', S, T', INFO', N', M')$, it invokes the Cheater Detection algorithm described below. Otherwise, it stores the transcript $(\mathcal{SPK}, S, T, INFO, N, M)$ in its database.

- Finalise. A payee with account (A', e', s', u') and a stored transcript $(\mathcal{SPK}, S, T, INFO, N, M)$ engages the payment server to compute the transaction.
  – The payee sends the transcript $(\mathcal{SPK}, S, T, INFO, N, M)$ to the payment server, along with the following zero-knowledge proof.

$$\mathcal{PK}_5\{(u'): M = \mathcal{G}(INFO||N)^{u'} \wedge U' = G^{u'}\}.$$

  – The payment server returns failure if the verification of $\mathcal{PK}_5$ returns invalid. Otherwise, the server calculates $R = \mathcal{H}(INFO||N||M)$ and validates $\mathcal{SPK}$. If the verification of $\mathcal{SPK}$ outputs invalid, the server returns failure. Otherwise, the server continues to check if R and N are fresh, if yes, it accepts the deposit and credits for payee. Otherwise, it rejects the request.
  – The payment server further checks if S is fresh. If there exists another transcript in the database $(\mathcal{SPK}', S, T', INFO', N', M')$, it invokes the Cheater Detection algorithm described below. Otherwise, it stores the transcript $(\mathcal{SPK}, S, T, INFO, N, M)$ in its database.

- Cheater Detection. Suppose there exists two transcripts $(\mathcal{SPK}', S', T', INFO', N', M')$ and $(\mathcal{SPK}, S, T, INFO, N, M)$ such that $S = S'$, the payment server invokes the following procedures.
  – Compute $R = \mathcal{H}(INFO||N||M)$ and $R' = \mathcal{H}(INFO'||N'||M')$. Note that due to $\mathcal{H}$ being collision-resistant, $R \neq R'$ (unless it is the same coin which will be rejected straight away).
  – Compute U^* as $(\frac{T^{R'}}{T'^R})^{\frac{1}{R'-R}}$.
  – Locate the user with U^* as identifier. This user is responsible for spending a coin twice.

5.3 Transaction Fairness

Due to the dual anonymity, it is imperative to guarantee fair transactions between payers and payees during payment. The main feature of fair e-cash is the existence of a qualified third party (a "trustee") to revoke the anonymity of any given coin. In doing so, traditional fair e-cash provides the additional properties of "payer tracing" and "coin tracing". Specifically, the payer tracing enables the payment server to identify the payer (i.e., the coin owner) based on the database of user accounts and the specific information extracted from the views of deposit protocol by the trustee; and the coin tracing allows the payment server to find the coin in the deposit phase having the specific information learnt from the views of withdrawal protocol by the trustee. Hence, both features support the tracing of suspicious payments. To achieve the payee anonymity, the proposed scheme introduces the Randomize protocol to randomize the received coin. Thus, the payee tracing can be supported by enabling the payment server to identify the payee based on the database of user accounts and the information obtained by the trustee from the views of Randomise or Finalize protocols. The extension for achieving fair transaction from our proposed scheme is efficient and straightforward following

the approach due to Frankel *et al.* [54]. Suppose the secret-public key pair of the trustee is (sk_t, pk_t). Each user is required to register at the trustee and obtain a signature of the trustee σ on the user account. To ensure the payer tracing, in Payment phase, the payer generates a ciphertext of σ for the trustee using Cramer-Shoup cryptosystem [55] and prove the items in the zero-knowledge proof $\mathcal{SPK}$. Thus, the payment server can identify the payer based on the user account with the decryption of the trustee. Similarly, to support payee tracing, the payee is needed to compute the ciphertext of σ for the trustee using Cramer-Shoup cryptosystem [55] and prove the items in $\mathcal{PK}_3$. Therefore, the payee can be traced for the payment server with the assistance of the trustee. In terms of coin tracing, the user is required to append the ciphertext of $C''' = h_0^{t'} h_1^u h_2^v$ under pk_t using Cramer-Shoup cryptosystem [55] and integrate the zero-knowledge proof of (t', u, v) to $\mathcal{PK}_2$ in Payment phase. The trustee is able to decrypt the ciphertext to acquire C''' and the payment server can check whether $C' = C''' h_0^{t''}$ holds, until a couple (C', B, f, t'', v'') maintained on withdrawal database succeeds it.

6 Security Proofs

The proposed dual-anonymous off-line e-cash scheme is secure if the security requirements stated in Section 3 are satisfied, including balance, payer anonymity, payee anonymity, and exculpability. The balance ensures that the number of the coins circulating on the market is equal to the quantity of the coins withdrew minus deposited. No adversary can forge or double-spend the coins. Payer anonymity and payee anonymity provide the privacy protection of payers and payees, i.e., dual anonymity. The exculpability offers the properties that an adversary cannot slander an honest user and the double-spending user will be identified.

Theorem 1. *Our construction is secure under DDH assumption and q-SDH assumption in the random oracle model.*

Balance. Assume that there is an adversary $\mathcal{A}$ that makes q_{AE} Acco-Estab queries, q_W Withdrawal queries, q_{P_1} Payment$_1$ queries, q_{P_2} Payment$_2$ queries, q_R Randomize queries, q_F Finalise queries, we show how to construct a forgery attack against the underlying BBS+ signature. Since the BBS+ signature is proved secure under the q-SDH assumption, where $q = q_{AE} + q_W + q_R$, there is no probabilistic polynomial time (PPT) adversary can win the Balance game with non-negligible probability.

Proof. We assume that the zero-knowledge proofs $\mathcal{PK}_1$, $\mathcal{PK}_2$, $\mathcal{PK}_3$, $\mathcal{PK}_4$, $\mathcal{PK}_5$ and $\mathcal{SPK}$ are sound. That is, there exist extract algorithms $\mathcal{EX}_1$, $\mathcal{EX}_2$, $\mathcal{EX}_3$, $\mathcal{EX}_4$, $\mathcal{EX}_5$ and $\mathcal{EX}_{\mathcal{SPK}}$ to capture the witnesses used by the provers, respectively. If the proofs are constructed non-interactively, the protocols described in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2021.3135301 are sound in random oracle model.

We describe an algorithm called simulator $\mathcal{S}$ that interacts with the adversary $\mathcal{A}$. $\mathcal{S}$ is given the public parameters $param$, the public key (W, X) and the secret key (α, β), and is allowed to access the signature oracle $\mathcal{SO}$ that can output BBS+ signature on block messages.

$\mathcal{S}$ gives the $param$ to $\mathcal{A}$ and interacts with $\mathcal{A}$ in each possible interaction with $\mathcal{SO}$.

1) Acco-Estab. $\mathcal{A}$ randomly chooses an identifier $U = G^u$, generates a proof $\mathcal{PK}_1$ and queries the Acco-Entab oracle. $\mathcal{S}$ extracts the witness (s''', u) from $\mathcal{PK}_1$ using $\mathcal{EX}_1$. Then, $\mathcal{S}$ issues a signature query to $\mathcal{SO}$ and receives (A, e, s). It computes $s'' = s - s'''$. At last, $\mathcal{S}$ returns (A, e, s'') to $\mathcal{A}$.
2) Withdrawal. $\mathcal{A}$ generates a proof $\mathcal{PK}_2$ and queries the Withdrawal oracle. $\mathcal{S}$ extracts the witness (t', u, v') from $\mathcal{PK}_2$ using $\mathcal{EX}_2$. Then, $\mathcal{S}$ issues a signature query to $\mathcal{SO}$ and receives (B, f, t, v). It computes $t'' = t - t'$, $v'' = v - v'$. At last, $\mathcal{S}$ returns (B, f, t'', v'') to $\mathcal{A}$.
3) Payment$_1$. $\mathcal{A}$ generates a proof $\mathcal{PK}_3$ and queries the Payment$_1$ oracle. $\mathcal{S}$ executes the Withdrawal queries to obtain a coin (B, f, t, v). Then, it transacts with $\mathcal{A}$.
4) Payment$_2$. $\mathcal{A}$ spends a coin to $\mathcal{S}$. At last, $\mathcal{S}$ validates the transcript of the transaction, and then checks whether S is fresh, if not, executes the Cheater Detection algorithm to recover the identifier U of the payer; Otherwise, stores the transcript in the database.
5) Randomise. $\mathcal{A}$ sends the transcript along with a proof of the account $\mathcal{PK}_3$ and a proof $\mathcal{PK}_4$ to $\mathcal{S}$. $\mathcal{S}$ validates the proofs and issues a new coin to $\mathcal{A}$ as follows.
   - $\mathcal{S}$ extracts the witness $(\hat{t}', u', \hat{v}')$ from $\mathcal{PK}_4$ using $\mathcal{EX}_4$, and issues a signature query to $\mathcal{SO}$. $\mathcal{S}$ receives $(\hat{B}, \hat{f}, \hat{t}, \hat{v})$ and computes $\hat{t}'' = \hat{t} - \hat{t}'$, $\hat{v}'' = \hat{v} - \hat{v}'$.
   - $\mathcal{S}$ returns $(\hat{B}, \hat{f}, \hat{t}'', \hat{v}'')$ to $\mathcal{A}$.

Finally, $\mathcal{S}$ checks whether S is fresh, if not, executes Cheater Detection algorithm to reveal the payer's identifier; Otherwise, stores the transcript in the database.

6) Finalise. $\mathcal{A}$ sends the transcript along with a proof of u' to $\mathcal{S}$. $\mathcal{S}$ verifies the transcript and deposits to the account. At last, $\mathcal{S}$ checks whether S is fresh, if not, executes the Cheater Detection algorithm to recover the payer's identifier; Otherwise, stores the transcript in the database.

Finally, assume that $\mathcal{A}$ runs q_{AE} Acco-Estab queries, q_W Withdrawal queries, q_{P_1} Payment$_1$ queries, q_{P_2} Payment$_2$ queries, q_R Randomize queries and q_F Finalise queries. $\mathcal{A}$ wins the game if $q_F > q_W + q_{P_1} - q_{P_2}$ and the Cheater Detection algorithm does not point to any of identifiers that have presented in the Withdrawal queries.

However, since $\mathcal{A}$ withdrew q_W coins in Withdrawal queries, obtained q_{P_1} transcripts in Payment$_1$ queries and spent q_{P_2} coins in Payment$_2$ queries, $\mathcal{A}$ only can deposit at most $q_W + q_{P_1} - q_{P_2}$ coins. If $q_F > q_W + q_{P_1} - q_{P_2}$, $\mathcal{A}$ must (1) have conducted forged coins (or transcripts) or (2) have double-spent the coins without being detected in Cheater Detection algorithm.

In case (1), if $\mathcal{A}$ can forge a valid coin or a transcript, which includes the signature of knowledge of a coin, it must have possessed the BBS+ signature on block of messages (t, u, v). While the BBS+ signature is secure under the q-SDH assumption. Therefore, the probability of forging a

coin is negligible under the q-SDH assumption, where $q = q_{AE} + q_W + q_R$.

Case (1) guarantees that all coins deposited are valid. In case (2), we suppose duplicated coins are accepted. Due to the zero-knowledge property of the Payment protocol, $\mathcal{A}$ learns nothing about the coin (B, f, t, v) presented during Payment$_1$ queries. Thus, $\mathcal{A}$ cannot produce valid deposit using the same (B, f, t, v) more than once. In Payment$_2$ queries, the transcripts from $\mathcal{A}$ are maintained in the database. Therefore, $\mathcal{A}$ cannot deposit successfully using the paid coins in Payment$_2$ queries.

It remains to show that the identifier of a double-spender must be recovered due to the correctness of the Cheater Detection algorithm. Due to the soundness of the zero-knowledge proof protocol, $T = G^u H_1^{Rv}$ is the only valid T to accompany specific transaction identified by the $R = \mathcal{H}(INFO||N||M)$ and $S = H^v$. Since R should be different in two transactions, G^u would be obtained as long as the proof during the Payment protocol is not faked. We already assume that the proof $\mathcal{SPK}$ is sound and all deposited coins are valid. Thus, the success probability of $\mathcal{A}$ is negligible. ■

Payer Anonymity. The adversary $\mathcal{A}$ cannot distinguish that a specific transaction is from one of two possible honest payers, if the DDH assumption in $\mathbb{G}_T$ holds.

Proof. Our security proof reduces the anonymity of payers to the DDH assumption in $\mathbb{G}_T$. That is, if $\mathcal{A}$ can distinguish the action of two honest payers, we show how to construct a simulator $\mathcal{S}$, which can solve the DDH problem. That is, given a 4-tuple $(D, D_1, D_2, D_3) \in \mathbb{G}_T^4$, $\mathcal{S}$ can tell if there exists (a, b), such that $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$. The view of $\mathcal{A}$ is provided by $\mathcal{S}$ with the random oracle. $\mathcal{S}$ generates $param$, sets $H = D$, $H_1 = D_1$, and sends them to $\mathcal{A}$. $\mathcal{A}$ acts as the payment server and gives bpk to $\mathcal{S}$. $\mathcal{S}$ picks two challenge identities $U_0 = G^{u_0}$ and $U_1 = G^{u_1}$, where $u_0, u_1 \in_R \mathbb{Z}_p$ and sends U_0 and U_1 to $\mathcal{A}$. Then, $\mathcal{S}$ interacts with $\mathcal{A}$ in each of possible interactions.

1) Acco-Estab. $\mathcal{S}$ acts on behalf of the user U_0 or U_1 honestly to establish an account A_0 or A_1.
2) Withdrawal. $\mathcal{S}$ randomly picks $C' \in \mathbb{G}$ and simulates the zero-knowledge proof $\mathcal{PK}_2$ to withdraw coins $\{C_{00}, \ldots, C_{0n}\}$ for U_0 ($\{C_{10}, \ldots, C_{1n}\}$ for U_1) interacting with $\mathcal{A}$.
3) Payment$_1$. $\mathcal{S}$ randomly chooses $v_j \in \mathbb{Z}_p$ and $R_j \in \mathbb{Z}_p$ to compute $S_j = H^{v_j}$, $T_j = U_0 H_1^{R_j v_j}$ ($T_j = U_1 H_1^{R_j v_j}$) for the jth query of U_0 (U_1). $\mathcal{S}$ also simulates the zero-knowledge proof $\mathcal{SPK}$.

Finally, $\mathcal{S}$ slips a fair coin $\delta \in \{0, 1\}$. $\mathcal{S}$ randomly chooses $R^* \in \mathbb{Z}_p$ and sets $S = D_2$ and $T = U_\delta D_3^{R^*}$, then simulates the zero-knowledge proof $\mathcal{SPK}$. Obviously, if $log_D D_3 = log_D D_1 \cdot log_D D_2$, the simulation is perfect; Otherwise, it contains no information about U_0 or U_1.

$\mathcal{A}$ outputs a bit $\hat{\delta}$. If $\hat{\delta} = \delta$, $\mathcal{A}$ wins the game and $\mathcal{S}$ is able to confirm that there exists (a, b) such that $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$; Otherwise, $\mathcal{S}$ confirms that there is no such item of (a, b) exists.

If there is such (a, b) that makes $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$ hold, the probability that $\mathcal{S}$ answers correctly is equal to that of $\mathcal{A}$ wins the game, that is, $\Pr[\mathcal{A}\ wins|\hat{\delta} = \delta] = \frac{1}{2} + \epsilon$, where ϵ is the probability of $\mathcal{A}$ wins the game better than random guessing. If there is no (a, b) such that $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$, the probability that $\mathcal{S}$ answers correctly is equal to that of $\mathcal{A}$ loses the game, that is, $\Pr[\mathcal{A}\ loses|\hat{\delta} \neq \delta] = \frac{1}{2}$.

In summary, the probability that $\mathcal{S}$ solves the DDH problem is $\frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} + \frac{\epsilon}{2}$, if we assume each instance of DDH problem comes from the distribution with probability $\frac{1}{2}$. Therefore, the DDH problem can be solved with probability $\frac{\epsilon}{2}$ better than random guessing. ■

Payee Anonymity. The adversary $\mathcal{A}$ cannot distinguish that a specific transaction is received by one of two possible honest payees, if the DDH assumption in $\mathbb{G}_T$ holds.

Proof. The anonymity of payees can be reduced to the DDH problem in $\mathbb{G}_T$. That is, if $\mathcal{A}$ is able to distinguish the action of two honest payees, we can construct a simulator $\mathcal{S}$ to solve the DDH problem. Specifically, given 4-tuple $(D, D_1, D_2, D_3) \in \mathbb{G}_T^4$, $\mathcal{S}$ can tell whether there is a pair (a, b), such that $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$. The view of $\mathcal{A}$ is offered by $\mathcal{S}$ with the random oracle. $\mathcal{S}$ generates the system parameter $param$, $\mathcal{G}$, $\mathcal{H}$ and sends it to $\mathcal{A}$. $\mathcal{A}$ acts on behalf of the payment server and provides $\mathcal{S}$ with bpk. $\mathcal{S}$ slips a fair coin $\delta \in \{0, 1\}$ and chooses two challenge identities $U_{1-\delta} = D^{x_{1-\delta}}$ and $U_\delta = D_1$, where $x_{1-\delta} \in_R \mathbb{Z}_p$, and sends $U_{1-\delta}$ and U_δ to $\mathcal{A}$. Then, $\mathcal{S}$ interacts with $\mathcal{A}$ in each of possible interactions.

1) Hash. $\mathcal{S}$ maintains a list of tuples $\langle INFO_j, N_j, x_j \rangle$ and responds queries from $\mathcal{A}$. For the jth hash query $(INFO_j, N_j)$, $\mathcal{S}$ sets $\mathcal{G}(INFO_j||N_j) = D^{x_j}$, where x_j is randomly chosen from $\mathbb{Z}_p$.
2) Acco-Estab. $\mathcal{S}$ acts on behalf of the user $U_{1-\delta}$ honestly to establish an account $A_{1-\delta}$. For U_δ, $\mathcal{S}$ randomly picks $C \in \mathbb{G}$ and then simulates the zero-knowledge proof $\mathcal{PK}_1$ to establish an account A_δ interacting with $\mathcal{A}$.
3) Payment$_2$. $\mathcal{S}$ honestly acts on behalf of the user $U_{1-\delta}$ to receive coins. For U_δ, M supplied to $\mathcal{A}$ is not formed correctly. For the jth query, $\mathcal{S}$ chooses random values $INFO_j$ and N_j, and sets the response of the Hash oracle as $\mathcal{G}(INFO_j||N_j) = D^{x_j}$. $\mathcal{S}$ computes $M_j = D_1^{x_j}$ and simulates the zero-knowledge proof $\mathcal{PK}_3$ interacting with $\mathcal{A}$.
4) Randomise. $\mathcal{S}$ honestly acts on behalf of the user $U_{1-\delta}$ to randomise the coins. For U_δ, $\mathcal{S}$ supplies $\mathcal{A}$ with transcripts and M that is not formed correctly. For the jth query, $\mathcal{S}$ chooses a transcript to randomise and finds x_j from the list according to $INFO_j$ and N_j. $\mathcal{S}$ also calculates $M_j = D_1^{x_j}$, randomly picks $C' \in \mathbb{G}$ and simulates the zero-knowledge proofs $\mathcal{PK}_3$, $\mathcal{PK}_4$ interacting with $\mathcal{A}$.

Finally, $\mathcal{S}$ chooses random values $INFO^*$ and N^*, and queries the random oracle to set $\mathcal{G}(INFO^*||N^*) = D_2^{x^*}$, where x^* is randomly chosen from $\mathbb{Z}_p$. $\mathcal{S}$ computes $M^* = D_3^{x^*}$. $\mathcal{S}$ also randomly picks $C' \in \mathbb{G}$ and simulates the zero-knowledge proofs $\mathcal{PK}_3$ and $\mathcal{PK}_4$. Note that the simulation is perfect if $log_D D_3 = log_D D_1 \cdot log_D D_2$; Otherwise, the whole interactions disclose no information about $U_{1-\delta}$ or U_δ.

$\mathcal{A}$ outputs a bit $\hat{\delta}$. If $\hat{\delta} = \delta$, $\mathcal{S}$ can confirm that there exists (a, b) such that $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$; Otherwise, $\mathcal{S}$ confirms that there is no such item of (a, b) exists.

TABLE 2
Comparison Results on Computational Overhead

Phases	Entities	Our Scheme	[28], [29]	[30]
Acco-Estab	User	$7T_{MP}$+$2T_{Exp}$+$2T_{BP}$	$21T_{MP}$+$10T_{Exp}$+$7T_{BP}$	T_{MP}
	Server	$5T_{MP}$+$2T_{Exp}$	$29T_{MP}$+$13T_{Exp}$+$10T_{BP}$	$2T_{BP}$
Withdrawal	User	$10T_{MP}$+T_{Exp}+$2T_{BP}$	$175T_{MP}$+$78T_{Exp}$+$57T_{BP}$	$18T_{MP}$+$35T_{Exp}$+$40T_{BP}$
	Server	$7T_{MP}$+$2T_{Exp}$	$124T_{MP}$+$129T_{Exp}$+$82T_{BP}$	$64T_{MP}$
Payment	Payer	$13T_{MP}$+$18T_{Exp}$+$3T_{BP}$	$143T_{MP}$+$53T_{Exp}$+$37T_{BP}$	$48T_{MP}$+$2T_{BP}$
	Payee	$13T_{MP}$+$17T_{Exp}$+$3T_{BP}$	$95T_{MP}$+$127T_{Exp}$+$96T_{BP}$	$2T_{MP}$+$35T_{Exp}$+$54T_{BP}$
Randomise	Payee	$17T_{MP}$+$6T_{Exp}$+$3T_{BP}$	/	/
	Server	$19T_{MP}$+$23T_{Exp}$+$4T_{BP}$	/	/
Finalise	Payee	$2T_{Exp}$	$75T_{MP}$+$31T_{Exp}$+$19T_{BP}$	$54T_{MP}$+$2T_{BP}$
	Server	$6T_{MP}$+$18T_{Exp}$+$2T_{BP}$	$54T_{MP}$+$72T_{Exp}$+$69T_{BP}$	$2T_{MP}$+$35T_{Exp}$+$70T_{BP}$

Similarly, if there exists (a, b) such that $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$, the probability that $\mathcal{S}$ answers correctly is $\Pr[\mathcal{A}\ wins|\hat{\delta} = \delta] = \frac{1}{2} + \epsilon$, where ϵ is the probability of $\mathcal{A}$ wins the game better than random guessing. While, if there is no (a, b) such that $D_1 = D^a$, $D_2 = D^b, D_3 = D^{ab}$, the probability that $\mathcal{S}$ answers correctly is $\Pr[\mathcal{A}\ loses|\hat{\delta} \neq \delta] = \frac{1}{2}$. The probability that $\mathcal{S}$ answers correctly is still $\frac{1}{2} + \frac{\epsilon}{2}$, if the instance of DDH problem is from the distribution with probability $\frac{1}{2}$. Therefore, the DDH problem can be solved with probability $\frac{\epsilon}{2}$ better than random guessing. ■

Exculpability. The adversary $\mathcal{A}$ cannot slander an honest user that he has double-spent coins, unless $\mathcal{A}$ can win the *Balance* game.

Proof. In the Cheater Detection algorithm, one is proved to be guilty if there are $T' = G^{u'} H_1^{R'v'}$ and $T = G^u H_1^{Rv}$ such that $u = u'$, $v = v'$ and $R' \neq R$. Thus, these T' and T must come from the same coin under the same account with two different transactions. Therefore, if the adversary $\mathcal{A}$ is successful at slandering an honest user, it is also successful forging the underlying proof of knowledge, which happens with negligible probability that has been shown in the proof of *Balance*. ■

7 Efficiency Analysis

We analyze the efficiency of our scheme by counting the number of the point multiplication operation in both $\mathbb{G}$, the exponentiation in $\mathbb{G}_T$ and bilinear pairing operation required in each phase. Other operations, such as point addition, integer multiplication, integer addition, hashing, etc. are insignificant compared with the exponentiation and pairing operations. Let T_{PM}, T_{Exp} and T_{BP} denote the execution time of each point multiplication operation in both $\mathbb{G}$, exponentiation in $\mathbb{G}_T$ and bilinear pairing operation, respectively. To show the computational efficiency of our proposed scheme, we compare the new scheme with Zhang *et al.*'s schemes [28], [29] and the instantiated construction due to Baldimtsi *et al.* [30] by structure-preserving signature and Elgamal encryption scheme. We restrict Zhang *et al.* [28], [29]'s and Baldimtsi *et al.* [30]'s schemes transfer once and give the comparison results on the computational overhead in terms of users and payment server in Table 2. Note that we pre-compute bilinear maps to reduce the computational cost as shown in Appendix A, available in the online supplemental material.

We also implement of our scheme on a notebook with Intel Core i5-4200U CPU and the clock rate is 2.29GHz and the memory is 4.00 GB. We used version 5.6.1 of MIRACL library to implement number-theoretic based methods of cryptography. The Weiling pairing is utilized to realize the pairing operation and the elliptic curve is chosen with base field size of 512 bits. The size of parameter p is 160 bits. We provide our simulation result in Table 3. To show the computational efficiency on a smartphone, we execute each time-consuming operation in our scheme on a HUAWEI MT2-L01 smartphone with Kirin 910 CPU 1.6GHz and 1250M memory. The operation system is Android 4.2.2 and the toolset is Android NDK r8d with MIRACL library [56]. The parameter p is still approximately 160 bits and the elliptic curve is defined as $y = x^3 + 3$ over $\mathbb{F}_q$, where q is 512 bits. The scalar multiplication operation and hash operation takes 3.609 ms and 7.560 ms, respectively. The executing time of the exponentiation operation in $\mathbb{G}_T$ and bilinear pairing operation is 0.001 ms and 56.201 ms. We illustrate the estimated executing time of each algorithm in our scheme and other related works [28], [29], [30] in Table 5. The results show that our scheme is highly efficient than the existing ones [28], [29], [30], such that it is suitable to be implemented on mobilephones for mobile payments.

We also show the communication overhead between two parties in our scheme in Table 3. If two users (a payer and a payee) take a transaction on two mobilephones through Bluetooth, whose bandwidth is 800Kps, time cost on transmission in Payment phase is about 2.5ms. In terms of the storage burden, apart from the system parameters that each user has to maintain, the user requires to store the account and the coin, which are both 124 bytes. The payment server needs to store the users' accounts and the transcripts, which are 192 bytes and 1144 bytes, respectively. However, in Zhang *et al.*'s schemes [28], [29], the binary length of a withdrew coin is nearly 20KB, and this length would increase around 10KB in each transaction. As for Baldimtsi *et al.*'s scheme [30], a withdrew coin is about 1KB and this coin would increase 0.2KB in each transaction. The comparison results of storage costs between our scheme and the related works [28], [29], [30] are shown in Table 3. In Table 3, we give the storage costs of each entity when they interact for account establishment, coin withdrawal, payment, coin randomization, and coin finalise. If the payment server provides payment services for k many users, the storage cost of the payment server is k times of the given cost in Table 3. It can be observed that the proposed scheme is highly efficient with respect to computational, communication and storage costs.

The above results are the computational, communication, and storage costs for the 80-bit security level. We also provide the computational, communication, and storage costs

TABLE 3
Runtime and Communication Overhead for 80-bit Security

Phase	Acco-Estab		Withdrawal		Payment		Randomise		Finalise	
	User	Server	User	Server	Payer	Payee	Payee	Server	Payee	Server
Runtime (ms)	70.43	75.24	60.32	74.36	282.46	205.64	187.41	380.69	24.84	308.74
Message length (bytes)	316	124	447	144	996	1144	2140	407	1332	20
Storage Cost (bytes)	336	192	480	192	336	1480	480	192	336	192

TABLE 4
Runtime and Communication Overhead for 128-bit Security

Phase	Acco-Estab		Withdrawal		Payment		Randomise		Finalise	
	User	Server	User	Server	Payer	Payee	Payee	Server	Payee	Server
Runtime (ms)	94.24	107.42	80.71	102.68	361.2	287.66	250.27	503.52	30.79	393.76
Message length (bytes)	480	224	603	256	1260	1460	2544	571	1596	32
Storage Cost (bytes)	384	232	544	232	384	1844	544	232	384	232

TABLE 5
Comparison Results on Runtime of Mobile Phones

Unit: sec

Phase	Acco-Estab		Withdrawal		Payment		Randomise		Finalise	
	User	Server	User	Server	Payer	Payee	Payee	Server	Payee	Server
Our Scheme (80-bit)	0.137	0.130	0.148	0.137	0.215	0.215	0.229	0.293	0.002	0.134
[28], [29] (80-bit)	0.469	0.667	3.836	5.056	2.596	5.739	/	/	1.335	4.073
[30] (80-bit)	0.003	0.112	3.313	0.231	0.286	3.042	/	/	0.307	3.942
Our Scheme (128-bit)	0.186	0.178	0.201	0.192	0.286	0.289	0.316	0.402	0.002	0.179
[28], [29] (128-bit)	0.621	0.935	5.256	6.924	3.491	7.854	/	/	1.757	5.638
[30] (128-bit)	0.004	0.154	4.581	0.307	0.398	4.137	/	/	0.425	5.428

for the 128-bit security level in Table 4. The 80-bit security level may not be sufficient for the mobile payment, and the 128-bit security level is more popular. The comparison of the executing time of each entity in our scheme and the related works [28], [29], [30] for the 128-bit security is shown in Table 5.

8 Conclusion

In this paper, we have introduced the property of dual anonymity for electronic cash schemes and presented the security model to accommodate payee anonymity. We have also proposed a dual-anonymous off-line electronic cash scheme from BBS+ signature. By utilizing the randomization technique, the payee can renew the received electronic cash with the payment server before deposit or new-round payment, such that it is impossible to identify the payee in a specific transaction for the payment server or other users. We have proved that the proposed scheme is secure under the proposed security model and demonstrated that the scheme is efficient to be implemented on mobile devices for mobile payment. Due to the property of off-line transaction, the proposed scheme is more suitable to be deployed in the scenario that the network connectivity is not in place, compared with the online payment methods, such as Bitcoin and e-wallets.

During the payment, a payee may receive multiple coins from different payers and he/she has to separately randomize or finalize them interacting with the payment server once the mobile device has network connection, so as to suffer from heavy computational and communication overhead. A possible solution is to achieve batch randomization and batch finalization for e-cash, which is the open problem of practical universal signature aggregation. In the future work, we focus on the design of batch randomization and batch finalization to enhance the computational and communication efficiency of electronic cash.

References

[1] Z. Lu, S. Rallapalli, K. S. Chan, S. Pu, and T. La Porta, "Augur: Modeling the resource requirements of ConvNets on mobile devices," *IEEE Trans. Mob. Comput.*, vol. 20, no. 2, pp. 352–365, Feb. 2021.

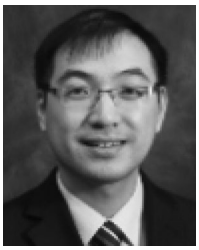
[2] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Trans. Mob. Comput.*, vol. 19, no. 6, pp. 1317–1331, Jun. 2020.

[3] X. Chen, J. Li, and W. Susilo, "Efficient fair conditional payments for outsourcing computations," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 6, pp. 1687–1694, Dec. 2012.

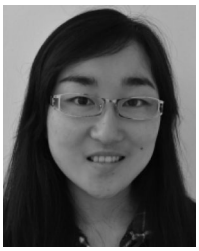
[4] Y. Ren, C. Wang, Y. Chen, M. C. Chuah, and J. Yang, "Signature verification using critical segments for securing mobile transactions," *IEEE Trans. Mob. Comput.*, vol. 19, no. 3, pp. 724–739, Mar. 2019.

[5] D. Chaum, "Blind signatures for untraceable payments," in *Proc. CRYPTO*, 1983, pp. 199–203.
[6] M. H. Au, W. Susilo, and Y. Mu, "Electronic cash with anonymous user suspension," in *Proc. ACISP*, 2011, pp. 172–188.
[7] S. Canard, D. Pointcheval, O. Sanders, and J. Traoré, "Divisible E-cash made practical," in *Proc. PKC*, 2015, pp. 77–100.
[8] S. Canard, D. Pointcheval, O. Sanders . and J. Traoré, "Scalable divisible E-cash," in *Proc. ACNS*, 2015, pp. 287–306.
[9] D. Pointcheval, O. Sanders, and J. Traoré, "Cut down the tree to achieve constant complexity in divisible E-cash," in *Proc. PKC*, 2017, pp. 61–90.
[10] F. Bourse, D. Pointcheval, and O. Sanders, "Divisible E-cash from constrained pseudo-random functions," in *Proc. ASIACRYPT*, 2019, pp. 679–708.
[11] B. Lian, G. Chen, J. Cui, and M. Ma, "Compact E-cash with efficient coin-tracing," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 220–234, Jan./Feb. 2021.
[12] M. H. Au, Q. Wu, W. Susilo, and Y. Mu, "Compact E-cash from bounded accumulator," in *Proc. CT-RSA*, 2007, pp. 178–195.
[13] S. Canard and A. Gouget, "Divisible E-cash systems can be truly anonymous," in *Proc. EUROCRYPT*, 2007, pp. 482–497.
[14] M. H. Au, W. Susilo, and Y. Mu, "Practical aonnymous divisible E-cash from bounded accumulators," in *Proc. FC*, 2008, pp. 287–301.
[15] M. Blantion, "Improved conditional E-payments," in *Proc. ACNS*, 2008, pp. 188–206.
[16] B. Yang, K. Yang, Z. Zhang, Y. Qin, and D. Feng, "AEP-M: Practical anonymous E-payment for mobile devices using ARM trustzone and divisible E-cash," in *Proc. ISC*, 2016, pp. 130–146.
[17] P. Märtens, "Practical compact E-cash with arbitrary wallet size," *Cryptology ePrint Archive: Report 2015/086*, 2015.
[18] J. Calhoun, C. Minwalla, C. Helmich, F. Saqib, W. Che, and J. Plusquellic, "Physical unclonable function (PUF)-based E-cash transaction protocol (PUF-Cash)," *Cryptography*, vol. 3, no. 3, 2019, Art. no. 18.
[19] G. Fragkos, C. Minwalla, E. E. Tsiropoulou, and J. Plusquellic, "Enhancing privacy in PUF-cash through multiple trusted third parties and reinforcement learning," *ACM J. Emerging Technol. Comput. Syst.*, vol. 18, no. 1, pp. 1–26, 2021.
[20] M. Scheir, J. Balasch, A. Rial, B. Preneel, and I. Verbauwhede, "Anonymous split E-cash toward mobile anonymous payments," *ACM Trans. Embedded Comput. Syst.*, vol. 13, no. 4, 2015, Art. no. 85.
[21] M. H. Au, W. Susilo, and Y. Mu, "Practical compact E-cash," in *Proc. ACISP*, 2007, pp. 431–445.
[22] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Compact E-cash," in *Proc. EUROCRYPT*, 2005, pp. 302–321.
[23] T. Okamoto, "An efficient divisible electronic cash scheme," in *Proc CRYPTO*, 1995, pp. 438–451.
[24] S. Canard and Gouget, "Multiple denominations in E-cash with compact transaction data," in *Proc. FC*, 2010, pp. 82–97.
[25] A. Deo, B. Libert, K. Nguyen, and O. Sanders, "Lattice-based E-cash, revisited," in *Proc. ASIACRYPT*, 2020, pp. 318–348.
[26] S. Canard and A. Gouget, "Anonymity in transferable E-cash," in *Proc. ACNS*, 2008, pp. 207–223.
[27] G. Fuchsbauer, D. Pointcheval, and D. Vergnaud, "Transferable constant-size fair E-cash," in *Proc. CANS*, 2009, pp. 226–247.
[28] J. Zhang, Z. Li, and H. Guo, "Anonymous transferable conditional E-cash," in *Proc. SecureComm*, 2013, pp. 45–60.
[29] J. Zhang, H. Guo, Z. Li, and C. Xu, "Transferable conditional E-cash with optimal anonymity in the standard model," *IET Inf. Secur.*, vol. 9, no. 1, pp. 59–72, 2015.
[30] F. Baldimtsi, M. Chase, G. Fuchsbauer, and M. Kohlweiss, "Anonymous transferable E-cash," in *Proc. PKC*, 2015, pp. 101–124.
[31] B. Bauer, G. Fuchsbauer, and C. Qian, "Transferable E-cash: A cleaner model and the first practical instantiation," in *Proc. PKC*, 2021, pp. 559–590.
[32] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed E-cash from Bitcoin," in *Proc. IEEE S&P*, 2013, pp. 397–411.
[33] E. Ben-Sasson *et al.*, "Zerocash: Decentralized anonymous payments from Bitcoin," in *Proc. of IEEE S&P*, 2014, pp. 459–474.
[34] C. Garman, M. Green, and I. Miers, "Accountable privacy for decentralized anonymous payments," in *Proc. FC*, 2016, pp. 81–98.
[35] A. Jivanyan, "Lelantus: Towards confidentiality and anonymity of blockchain transactions from standard assumptions," *IACR Cryptol. ePrint Arch. 373, Rep. 2019/373*, 2019.
[36] H. Tewari and A. Hughes, "Fully Anonymous Transferable Ecash," *Cryptology ePrint Archive: Rep. 2016/107*, 2016.
[37] D. Liu, A. Alahmadi, J. Ni, X. Lin, and X. Shen, "Anonymous reputation system for IIoT-enabled retail marketing atop PoS blockchain," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3527–3537, 2019.
[38] C. Lin, D. He, X. Huang, M. K. Khan, and K. K. R. Choo, "DCAP: A secure and efficient decentralized conditional anonymous payment system based on blockchain," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2440–2452, Jun. 2020.
[39] S. Noether, "Ring signature confidential transactions for monero," *IACR Cryptol. ePrint Arch. 1098*, 2015.
[40] Y. Li, G. Yang, W. Susilo, Y. Yu, M. H. Au, and D. Liu, "Traceable Monero: Anonymous Cryptocurrency with Enhanced Accountability," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 679–691, Mar./Apr. 2021.
[41] R. W. Lai, V. Ronge, T. Ruffing, D. Schröder, S. A. K. Thyagarajan, and J. Wang, "Omniring: Scaling private payments without trusted setup," in *Proc. ACM CCS*, 2019, pp. 31–48.
[42] G. Maxwell, "CoinJoin: Bitcoin privacy for the real world," 2013. [Online]. Available: https://bitcointalk.org/index.php?topic=279249.0
[43] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for Bitcoin with accountable mixes," in *Proc. FC*, 2014, pp. 486–504.
[44] G. Fuchsbauer, M. Orru, and Y. Seurin, "Aggregate cash systems: A cryptographic investigation of mimblewimble," in *Proc. EUROCRYPT*, 2019, pp. 657–689.
[45] T. E. Jedusor, "Mimblewimble," 2016. [Online]. Available: https://download.wpsoftware.net/ bitcoin/wizardry/mimblewimble.txt
[46] H. Tewari, "T-Cash: Transferable fiat backed coins," 2021, *arXiv: 2105.04485*.
[47] Atlantic Council, "The rise of central bank digital currencies," Accessed: Jun. 16, 2020. {Online]. Available: https://www.atlanticcouncil.org/blogs/econographics/the-rise-of-central-bank-digital-currencies/
[48] A. Dmitrienko, D. Noack, and M. Yung, "Secure Wallet-assisted offline bitcoin payments with double-spender revocation," in *Proc. AsiaCCS*, 2017, pp. 520–531.
[49] M. Bellare and O. Goldreich, "On defining proofs of knowledge," in *Proc. CRYPTO*, 1992, pp. 390–420.
[50] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems (Extended Abstract)," in *Proc. STOC*, 1985, pp. 291–304.
[51] I. Damgård, "Efficient concurrent zero-knowledge in the auxiliary string model," in *Proc. EUROCRYPT*, 2000, pp. 418–430.
[52] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," in *Proc. SCN*, 2006, pp. 111–125.
[53] J. Camenisch and M. Stadler, "Efficient group signature schemes for large group (extended abstract)," in *Proc. CRYPTO*, 1997, pp. 410–424.
[54] Y. Frankel, Y. Tsiounis, and M. Yung, "Fair off-line E-cash made easy," in *Proc. AsiaCrypt*, 1998, pp. 257–270.
[55] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for chosen ciphertext secure public key encryption," in *Proc. Eurocrypt*, 2002, pp. 45–64.
[56] Miracl, "About the MIRACL Crypto SDK," [Online]. Available: https://www.certivox.com/miracl

Jianbing Ni (Member, IEEE) received the PhD degree in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2018. He is currently an assistant professor with the Department of Electrical and Computer Engineering and a member of the Ingenuity Labs Research Institute, Queen's University, Kingston, Canada. His research interests include applied cryptography and network security, with current focus on big data security, edge computing, mobile crowdsensing, Internet of Things, and Blockchain technology.



Man Ho Au (Member, IEEE) received the BEng and MPhil degrees from the Department of Information Engineering, Chinese University of Hong Kong, in 2003 and 2005 respectively, and the PhD degree from the School of Computer Science and Software Engineering, University of Wollongong, in 2009. He is currently an associate professor with the Department of Computer Science in the University of Hong Kong (HKU). Priori to joining HKU, he was an associate professor with the Department of Computing, Hong Kong Polytechnic University. He has authored or coauthored more than 160 refereed papers in top journals and conferences, including CRYPTO, ASIACRYPT, ACM CCS, ACM SIGMOD, NDSS, *IEEE Transactions on Information Forensics and Security*, *IEEE Transactions on Computers*, and *IEEE Transactions on Knowledge and Data Engineering*. His research interests include applied cryptography, information security, blockchain technology, and related industrial applications. He was the recipient of the 2009 PET runner-up award for outstanding research in privacy enhancing technologies. He is also an expert member of the China delegation of ISO/IEC JTC 1/SC 27 working group 2 - Cryptography and security mechanisms and a committee member of the Hong Kong Blockchain Society R&D division.



Wei Wu received the PhD degree from the University of Wollongong, Wollongong, Australia, in 2009. She is currently an associate professor with the School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, China. She has authored or coauthored more than 50 referred research papers at international conferences and journals. Her research interests include new public key cryptography systems and secure server-aided computation.

Xiapu Luo received the PhD degree in computer science from the Hong Kong Polytechnic University. He then spent two years with the Georgia Institute of Technology as a postdoctoral research fellow. He is currently an associate professor with the Department of Computing, Hong Kong Polytechnic University. His current research focuses on smartphone security, network security, and privacy.

Xiaodong Lin (Fellow, IEEE) received the PhD degree in information engineering from the Beijing University of Posts and Telecommunications, China, and the PhD degree (with Outstanding Achievement in Graduate Studies Award) in electrical and computer engineering from the University of Waterloo, Canada. He is currently a professor with the School of Computer Science, University of Guelph, Canada. His research interests include computer and network security, privacy protection, applied cryptography, computer forensics, and software security.

Xuemin (Sherman) Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a university professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include network resource management, wireless network security, social networks, 5G and beyond, and vehicular ad hoc and sensor networks. He is a registered professional engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society. He was the recipient of R.A. Fessenden Award in 2019 from IEEE, Canada, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society. He has also received the Excellent Graduate Supervision Award in 2006 and Outstanding Performance Award five times from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He was the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, Symposia Chair for the IEEE ICC'10, Tutorial Chair for the IEEE VTC'11 Spring, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He was the editor-in-chief of the *IEEE Internet Of Things Journal* and *IEEE Network*. He is the president of the IEEE Communications Society.