Report on A Scheme for Offline Electronic Payment System

Abdullah Kutubi
mkutubi1@student.gsu.edu

Prashanth Sai Konda
pkonda1@student.gsu.edu

Tony Derado
tderado1@student.gsu.edu

Dec 10, 2023

1 Abstract

This research presents an innovative offline electronic payment system using Schnorr's untraceable blind signature (BS) to address the essential security requirements of e-payment systems. This scheme stands out for its practicality due to simpler computations compared to many existing systems, and it significantly contributes to the field of cryptographic protocols for anonymous and untraceable e-payment systems.

2 Introduction

The evolution of computer technology has given rise to the widespread embrace of digital payment methods, including e-cash and e-wallets, as alternatives to conventional paper currency and physical wallets [4]. These e-payment systems rely on cryptographic protocols and distributed computing systems to facilitate secure transactions between customers and merchants, eliminating the necessity for direct face-to-face interactions and offering efficiency gains. In recent years, computer applications have emerged, enabling rapid and convenient fund transfers through near field communication (NFC) techniques, thus reducing costs and reducing the reliance on paper checks within the banking sector [29]. Nonetheless, concerns regarding the security and privacy of these applications have been raised due to vulnerabilities such as counterfeit currency, unauthorized access, mishandling, and multiple spending attempts. Consequently, there has been a growing emphasis on research into e-payment systems based on cryptography that prioritize privacy and confidentiality.

E-payment systems can be broadly categorized based on the mode of connectivity of the third party involved (e.g., the Bank, the trusted central authority (CA), etc.) within the system, resulting in two primary categories: online and offline systems. In an online system [19, 9, 14, 24, 16] , the Bank typically has real-time access to transactions between customers and merchants, allowing for immediate detection of any illicit activities and straightforward control. However, this continuous monitoring creates a bottleneck situation and compromises customer anonymity to some extent. In contrast, the offline system [2, 31, 10, 30] operates without the simultaneous presence of any third party during transactions. This system enables merchants to securely deposit their earned e-coins into their bank accounts at a convenient time, relieving congestion. Offline systems come in two varieties: identified and anonymous.

The anonymous system safeguards the privacy of honest customers by rendering them anonymous to merchants, banks, certification authorities, or any other third parties, as long as they engage in legitimate transactions and avoid fraudulent activities. Consequently, research on anonymous offline systems has garnered significant attention. A practical e-payment scheme must meet several security requirements, including anonymity, unlinkability, unforgeability, multiple spending prevention, fraud deterrence, impersonation prevention, framing prevention, fraudster control, conditional traceability, offline payment capability, and date attachability, among others [2, 12, 15, 17, 21]

This paper introduces an innovative offline electronic payment scheme that leverages Schnorr's blind signature (BS). The proposed scheme surpasses many existing solutions in terms of both efficiency and security, particularly outperforming Hwang et al.'s RSA BS-based offline scheme presented in [17]. Schnorr's BS [23] offers an advantage through its streamlined computations, ensuring untraceability and resulting in reduced computational overhead, ultimately achieving efficient customer anonymity. Additionally, the proposed scheme guarantees unlinkability between the e-coin and its owner, ensuring that only the e-coin owner has access to this linkage. While various BS protocols exist in cryptography, most face challenges in maintaining untraceability in a cost-effective manner. However, Schnorr's BS protocol is recognized for its provable security, efficiency, and adherence to standard BS protocol characteristics, relying on the complexity of the discrete logarithm problem (DLP). The developed scheme also incorporates the RSA algorithm to ensure secure communication.

Furthermore, each e-coin stores three distinct pieces of information: the expiration date, the deposit date, and the transaction date. The expiration date serves to minimize the size of the bank's database, the deposit date aids in accurately estimating e-coin interest, and the transaction date helps detect any instances of multiple spending.

2.1 Evolution of Digital Payment Systems

The shift from traditional cash and physical wallets to digital payment methods has been catalyzed by the rapid development of computer technologies. E-cash and e-wallets have become predominant, facilitating transactions without the need for physical interactions, thereby saving time and resources.

2.2 Challenges in Digital Payment Systems

Despite the convenience offered by digital payment systems, they have raised significant privacy and security concerns. Vulnerabilities such as cash forgery, unauthorized access, and multiple spending attempts have necessitated a focus on cryptography-based e-payment systems.

2.3 Online vs. Offline E-Payment Systems

E-payment systems are broadly categorized into online and offline systems based on the connectivity of third parties like banks and central authorities. Online systems offer real-time transaction checking but at the cost of customer anonymity and system efficiency. Offline systems, conversely, provide more privacy and less congestion but require robust security mechanisms to handle transactions securely.

2.4 Proposed Scheme

2.4.1 Efficiency and Security

The scheme leverages Schnorr's BS for its efficiency and security. The BS protocol uses simpler computations to ensure untraceability and anonymity, thus addressing the critical security requirements of e-payment systems with reduced computational overhead.

2.4.2 Information Embedded in E-Coins

Each e-coin in the system carries three distinct pieces of information: expiration, deposit, and transaction dates. This design aids in managing the bank's database size, calculating interest accurately, and resolving disputes related to multiple spending.

3 Related Works

In the realm of e-commerce research, a multitude of e-payment schemes currently exist, each subject to cryptanalysis scrutiny. The initial foray into offline e-payment systems, as proposed in [8], relied on a blind signature (BS) concept coupled with the cut-and-choose protocol. However, subsequent cryptanalysis [25] revealed that this scheme necessitated extensive data exchange among the involved parties, including the Bank, the customer, and the merchant, leading to efficiency degradation. Although subsequent schemes such as [5, 28, 27] sought to address the shortcomings of the cut-and-choose protocol, as well as other proposals like [2, 31, 10, 30, 11] aimed at meeting the criteria of an ideal offline e-payment system, cryptanalysis brought to light their inefficiencies and susceptibility to significant security vulnerabilities. For example, schemes introducing divisibility and transferability of e-coins like [2, 10, 1] became impractical due to the expansion of e-coin size caused by the transaction history appended to detect fraudulent payments [17].

The e-cash system known as Mondex, presented in [27], harnessed a public key cryptosystem for microchips on payment cards, which is still in use by MasterCard Inc. However, this system required customers to reveal their identity when obtaining a Mondex card, thereby compromising anonymity. In contrast, the approach advocated in [5] introduced a trusted third party to mitigate continuous Bank involvement, reducing congestion overhead during payments. Nevertheless, cryptanalysis exposed weaknesses, such as inaccuracies in representing customer identities. Successive schemes, as discussed in [11, 7], and others, aimed to address this flaw. An alternate method, as presented in [13], involved the Bank in revealing the identity of a dishonest customer who attempted double spending of their e-coin, rather than relying on a trusted third party. However, this approach came with its own security issues, including the possibility of an issuer Bank or attacker identifying the link between the e-cash and its owner.

A paper proposed in [32] achieved optimal anonymity through the use of automorphic BS, group BS, and Groth-Sahai zero-knowledge proof (ZKP). This innovative scheme introduced a distinct structure for e-coins, dividing transferred e-coins into two parts, thereby addressing limitations in existing transferable e-payment systems. The proposal in [6] marked a breakthrough as the first efficient divisible e-coin system, with predetermined withdrawal and payment protocols, and it assigned the Bank the role of quickly detecting double spending. Similarly, the solution put forward in [25] aimed to fulfill security requirements such as transferability and divisibility. It was designed for mobile devices, utilizing ElGamal DS and Schnorr's identification protocol. In contrast, the scheme outlined in [3] tackled a novel security requirement, the change-giving problem, by allowing online shops to provide change in e-cash to customers. It resolved this issue through an existing group BS protocol.

The introduction of the concept of Bitcoin and blockchain technology in [22] marked the emergence of the first decentralized digital currency, enabling online transactions without the need for a bank or trusted third party, reducing processing fees [20]. These transactions are publicly transmitted and recorded in a publicly remiers2013zerocoinadable blockchain ledger [18]. However, blockchain systems have a limitation of supporting only about seven transactions per second, rendering them unsuitable for high-capacity and high-frequency payment systems [33]. Consequently, researchers have developed off-chain and offline e-payment schemes to overcome this limitation. In [35], user privacy in blockchain is maintained, while [18] utilizes blockchain to monitor transaction constraints and reduce fraud risk.

However, it lacks real-time protection against multiple spending, potentially resulting in substantial fraud before detection. A secure and versatile light payment system based on blockchain was presented in [33].

Existing offline e-payment schemes based on blind signatures have undergone analysis, revealing that many compromise untraceability to maintain control over multiple spending and conditional traceability. While the scheme outlined in [17] meets the fundamental security requirements of e-payment systems, it is not particularly efficient as it relies on Hwang et al.'s blind signature scheme [26], which entails computationally expensive underlying operations. To address this inefficiency, this paper proposes a streamlined e-payment scheme employing a customized version of Schnorr's blind signature scheme [23]. This approach maintains customer anonymity and ensures unlinkability between the e-coin and its owner through simplified computations. It guarantees customer anonymity under legal spending conditions, with the option of disclosing the identity of a fraudulent customer only with the assistance of the Certification Authority (CA).

3.1 Early Developments and Limitations

The first offline e-payment system based on BS and the cut-and-choose protocol marked the beginning of this domain. Subsequent systems tried to address its inefficiencies but faced significant security flaws, particularly regarding transaction size and customer anonymity.

3.2 Emergence of Decentralized Systems

The advent of Bitcoin and blockchain technology introduced a new era of decentralized digital currencies. However, these systems faced their own set of limitations, like transaction capacity, which led to the exploration of efficient offline e-payment systems.

4 Security Components

This section explains the primitive security components used by authors for the developed e-payment system. These are the RSA algorithm and Schnorr's BS protocol.

4.1 RSA Algorithm

The RSA used in the model is from pycrytodome library, we use the methods from Crypto, including PublicKey's RSA, Signature's pkcs1_15, Hash's SHA256, Cipher's PKCS1_OAEP. RSA is used to generate the public and private keys for all entities. For each the key is generated with 2048 bits, then the private key is exported, then another pair is generated the same way for the signing and verifying for each entity. The RSA is also used to import keys when needed. SHA256 is used as the hash function for the scheme, when hashing it can be assumed that SHA256 was used. Finally, pkcs_15 is used to handle signing and verifying while PKCS1_OAEP handles the encryption and decryption.

4.2 Schnorr's Blind Signature

The Schnorr's blind signature is implemented throughout the protocols mentioned later, the schnorr method gets a large prime p and picks a candidate q such that q divides (p-1), here is done by getting (p-1) /2, if the number is not a prime number it will start from the top, selecting a new p. Then the generates a g value, making sure it is between 1 and p-1. The signing keys use g in their create as key = pow(g, -s, p) (gs mod p). The rest is within the protocols, the signing keys are used along with random secret numbers for each entities a part of the interaction. For example in withdraw the bank, customer, and CA generate a random secret number, bank blinds a challenge to the customer which is the customer in return blinds using their secret number, the CA blinds the customer's public ID number to create Yr and encrypts the secret key twice, once with its encryption key and once with the customer's, the values are appended to C or coin amount to create m which is then hashed with the blinded challenge from customer to create e. The validity of the coin can be verified with e = H(m, ghe mod p) to check the banks signature, and for merchants the customer's signature can be verified with e = H(m, ghe mod p).

4.3 RSA Algorithm

The RSA algorithm forms the basis of the RSA DS and RSA public key cryptosystems. The distinction between the private and public keys for signing/verification and encryption/decryption is crucial in maintaining the security integrity of the system.

4.4 Schnorr's Blind Signature

Schnorr's BS protocol, a variant of the DS protocol, plays a pivotal role in preserving customer anonymity. The protocol involves several phases, such as initialization, blinding and signing, verification, and revelation, to ensure secure and anonymous transactions.

5 Offline e-payment system

This section presents an offline e-payment system, detailing the roles of the Central Authority (CA) and the Bank, their responsibilities, and the various databases they control.

5.1 Central Authority (CA)

- The CA keeps two separate key pairs using the RSA algorithm. One pair is for public encryption and private decryption {(e, n), (d, n)}, and the other is for private signing and public verification {(d', n'), (e', n')}.

- The CA is involved in creating valid e-coin in the Withdrawal protocol and assists the Bank in identifying the identity of multiple spenders.

5.2 Bank

- The Bank authenticates customers and merchants using their own Bank account numbers. It controls several databases, namely the account database, withdrawal database, and deposit database.

- To manage the information of customers and merchants, it maintains the account info database. The Bank also attaches an expiration date to the e-coin requested for withdrawal, and the withdrawn e-coin is stored in the withdrawal database for further usage.

- The Bank verifies the validity of an e-coin requested to be deposited by the merchant before depositing it in the deposit database. At the time of deposit, the Bank also detects and restrains multiple spenders, disclosing the liable one's identity with the help of the CA.

5.3 Merchant

- Verifies the validity and ownership of e-coins received from customers before exchanging commodities for e-coins. Attaches the transaction date (TDT) to valid e-coins and deposits them to the Bank.

- Generates and maintains two separate RSA key pairs using the RSA algorithm: one for public encryption and private decryption $(\mathrm{E}_M, N_M), (D_M, p_M, q_M)$ and the other for private signing and public verification $(\mathrm{S}_M, p_{MS}, q_{MS}), (V_M, N_{MS})$.

5.3.1 Description of Databases

The 4 databases used were the CA's registration database, containing the customer and merchant IDs, along with information about the individual. The other 3 databases are held by the banking entities. One for account information including an account number, hashed pin code, and balance. The second is the withdraw database, holding the information about E coin, the coins unique id, and additional information. The final database for the bank is the deposit table, holding the e-coin information along with the deposit date time and the transactions date time. To implement these systems the model used pandas' DataFrame method to create spreadsheets to act as the databases for testing the algorithms.

Table 1: THE CA'S REGISTRATION DATABASE

IDs of customers and merchants	**Information of customers and merchants**
IDc1	Name, address etc.
IDm1	Name, address etc.
IDc2	Name, address etc.
IDm2	Name, address etc.
IDcn	Name, address etc.
IDmn	Name, address etc.

Table 2: THE BANK'S ACCOUNT INFO DATABASE

Account number (customers and merchants)	**PIN Hash Code**	**Account's balance ($)**
AIDc1	PHCc1	2000
AIDm1	PHCc2	5000
AIDc2	PHCc3	4000
AIDm2	PHCc4	3000
...	...	...
AIDcn	PHCcn	2000
AIDmn	PHCmn	1000

Table 3: THE BANK'S WITHDRAWAL INFO DATABASE

E-coin	**E-coin unique id**	**Additional info**
m1, R1, e1	Cid1	α_1
m2, R2, e2	Cid2	α_2
...	...	...
mn, Rn, en	Cidn	α_n

Table 4: THE BANK'S DEPOSIT INFO DATABASE

E-coin	**Date time of deposit**	**Date time of transaction**
$(m, \rho, e, \sigma, w, T)_1$	DDT1	TDT1
$(m, \rho, e, \sigma, w, T)_2$	DDT2	TDT2
...	...	...
$(m, \rho, e, \sigma, w, T)_n$	DDTn	TDTn

6 System Configuration

6.1 Entities and Their Roles

The system includes several key entities: Central Authority (CA), Bank, Customer, and Merchant. Each entity has specific roles, like the CA acting as a trusted arbitrator, the Bank managing various databases, and Customers and Merchants handling transactions with specific RSA algorithm-based keys.

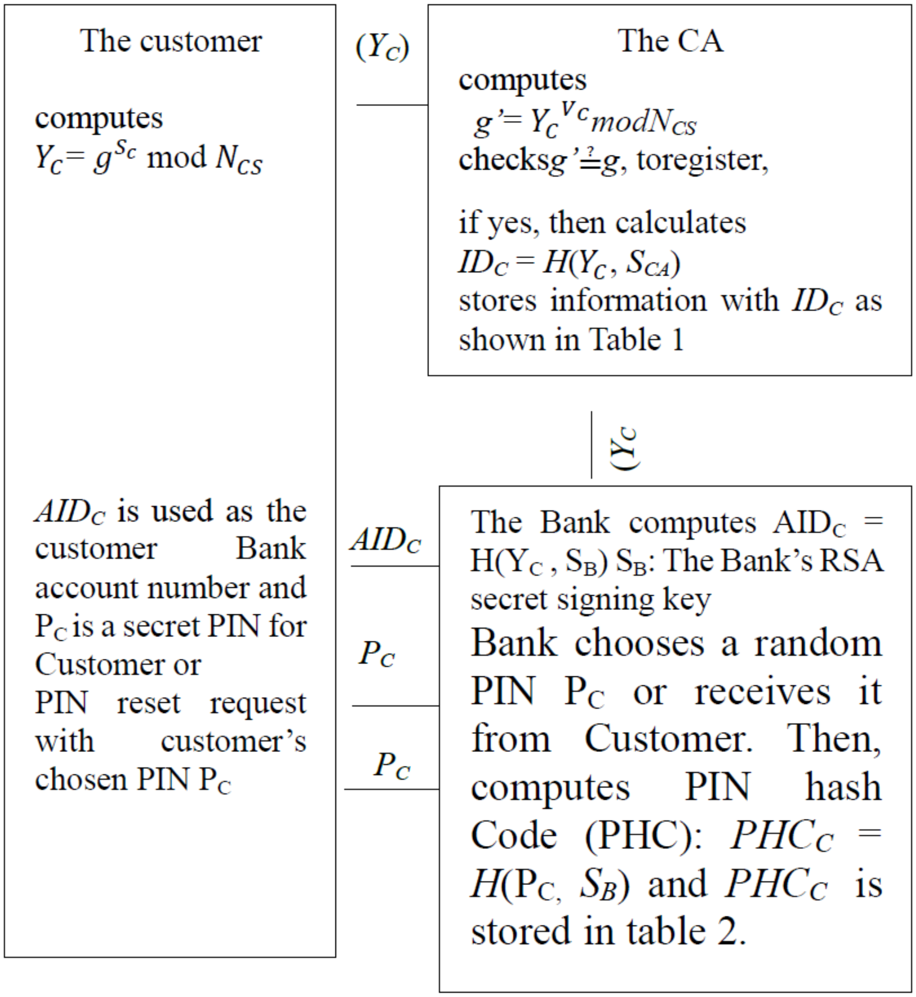
7 Protocols Overview

The six protocols - Registration, Withdrawal, Payment, Deposit, Renewal, and Tracing - collectively ensure the system's security and efficiency. Each protocol addresses specific aspects of the transaction process, from user authentication to handling and renewing e-coins.

7.1 Protocols Implementation:

The implementation of the offline E-payment system contains 6 protocols to facilitate the working of the scheme. These include the registration, withdraw, payment, deposit, renewal, and the Tracing protocol.

7.1.1 Registration



The customer

computes
$Y_C = g^{S_c} \bmod N_{CS}$

AID_C is used as the customer Bank account number and P_C is a secret PIN for Customer or PIN reset request with customer's chosen PIN P_C

The CA

computes
$g' = Y_C^{V_C} mod N_{CS}$
checks $g' \overset{?}{=} g$, toregister,

if yes, then calculates
$ID_C = H(Y_C, S_{CA})$
stores information with ID_C as shown in Table 1

The Bank computes $AID_C = H(Y_C, S_B)$ S_B: The Bank's RSA secret signing key
Bank chooses a random PIN P_C or receives it from Customer. Then, computes PIN hash Code (PHC): $PHC_C = H(P_C, S_B)$ and PHC_C is stored in table 2.

For the registration, the protocol is split into two methods, the Registration_customer and Registration_mer Both running the same except for the labeling, so only one will be explained. The method is called with the customer-generated secret signing key and verification key, S_C and V_C, along with the user information. The customer's S_C is blinded by using the public generator g, using $\mathrm{pow}(g, S_C, N_{CS})$ (where pow uses values as such: $\mathrm{pow}(\mathrm{base}, \mathrm{exp}, \mathrm{mod})$). The value given is Y_C, which is used as the customer's public ID number ($\mathrm{CPID_N}$). CA verify is implemented as verify, which uses g, Y_C, and V_C to verify the message as from the customer. Then, if true, the CA uses its secret signing key (S_{CA}) to create the customer's IDC by using SHA256 to hash both Y_C and S_{CA}. The information is then placed into the database before the bank creates the AIDC or bank account number for the customer using the same SHA256 hash with Y_C and the bank's secret signing key S_B. Then, the data is added to the bank's account table. Finally, the method returns the customer's $\mathrm{AID_C}$ and $\mathrm{PC_{pin}}$ for the user.

7.1.2 Withdraw

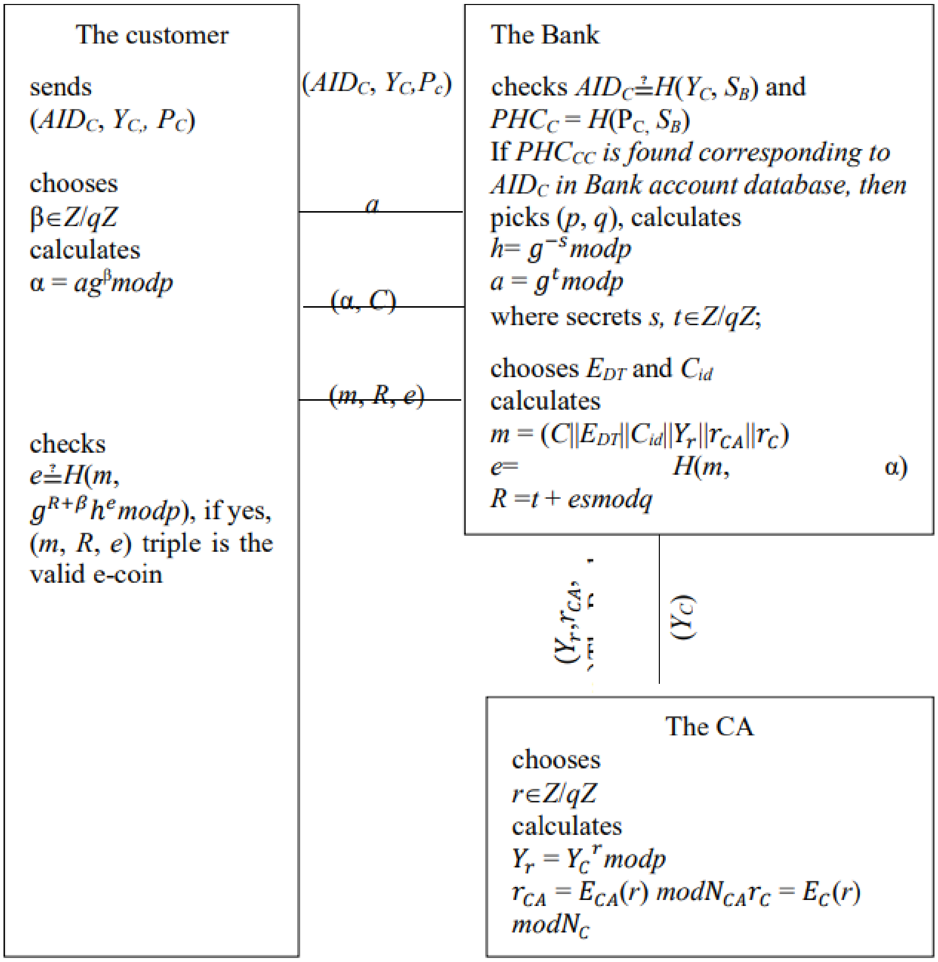

The customer
sends
$(AID_C, Y_{C,} P_C)$
chooses
$\beta \in Z/qZ$
calculates
$\alpha = ag^{\beta} modp$
checks
$e \overset{?}{=} H(m,$
$g^{R+\beta} h^e modp)$, if yes,
(m, R, e) triple is the
valid e-coin
(AID_C, Y_C, P_c)
a
(α, C)
(m, R, e)
The Bank
checks $AID_C \overset{?}{=} H(Y_C, S_B)$ and
$PHC_C = H(P_{C,} S_B)$
If PHC_{CC} is found corresponding to
AID_C in Bank account database, then
picks (p, q), calculates
$h = g^{-s} modp$
$a = g^t modp$
where secrets $s, t \in Z/qZ$;
chooses E_{DT} and C_{id}
calculates
$m = (C \| E_{DT} \| C_{id} \| Y_r \| r_{CA} \| r_C)$
$e = H(m, \alpha)$
$R = t + esmodq$
$(Y_r, r_{CA},$
(Y_C)
The CA
chooses
$r \in Z/qZ$
calculates
$Y_r = Y_C{}^r modp$
$r_{CA} = E_{CA}(r)\ modN_{CA} r_C = E_C(r)$
$modN_C$

Figure 2: The dataflow diagram of the Withdrawal protocol.

This method is implemented using input AIDC, $\mathrm{PC}_{\mathrm{pin}}$, and Y_{C}. The bank uses its secret key S_{B} to check the AIDC by hashing Y_{C} with S_{B} and comparing it to $\mathrm{AID}_{\mathrm{C}}$. If they match, the pin $\mathrm{PC}_{\mathrm{pin}}$ is hashed with S_{B} to compare with the table. Then the bank generates a secret number using q, where a number t is picked from 1 to $q-1$. A is calculated from $\mathrm{pow}(g, t, p)$. Then the bank blinds A by multiplying it with $\mathrm{pow}(g, \beta_{\mathrm{subscript}}, p)$ and taking the modulo p operation, creating α. The method is called `Bank_interact_with_CA_withdrawal`, passing on Y_{C} which acts as the CA portion, generating its own secret number r from q, then encrypting r into both R_{CA} and R_{C} using the CA's public key and the customer's public key. Y_r is also generated using $\mathrm{pow}(Y_{\mathrm{C}}, r, q)$. All three are sent back to the withdrawal method. The bank then creates a C value for the e-coin amount, an expiration date, and a coin id. The values are appended together as $m = (C||EDT||Cid||Y_r||R_{\mathrm{CA}}||R_{\mathrm{C}})$. Two values e and R are calculated, the value of e is calculated from hashing m with α and the value of R is $(t + e \cdot s) \mod q$. The withdrawal is then added to the database before e is checked by the customer using $e \overset{?}{=} \mathrm{H}(m, \mathrm{pow}(g^R + h^e \mod p))$.

7.1.3 Payment

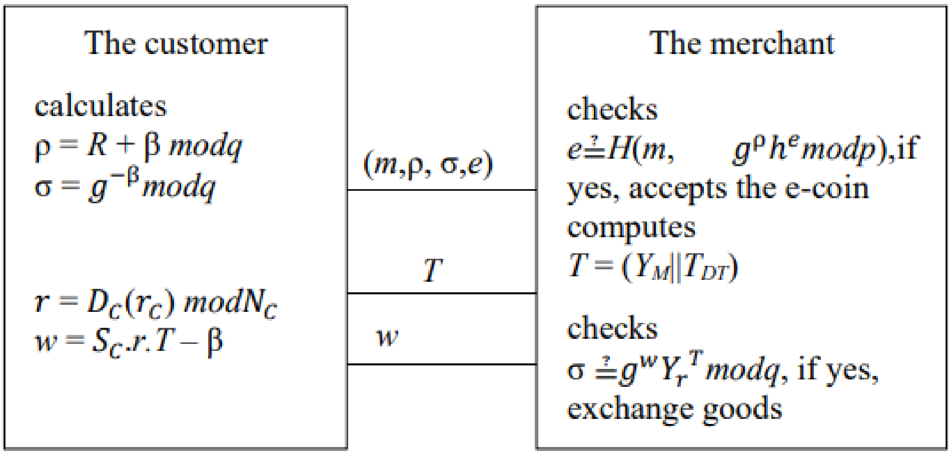


Figure 3 : The dataflow diagram of the Payment protocol.

The payment method is called using m, R, e, Y_M, T_{DT} (Y_M being the merchant's Y_C). The ρ value is calculated as $(R+\beta) \mod q$. σ is calculated as pow$(g, -\beta, q)$. The merchant verifies e with $e \overset{?}{=} \mathrm{H}(m, g^{\rho} h^{e} \mod p)$. If e matches the calculated value, then the method proceeds, creating T as Y_M appended with T_{DT}. The m is split so the customer can get r_{C} to decrypt using the customer's decryption key. Then the value is used to calculate $w = S_{\mathrm{C}} \cdot r \cdot T - \beta_{\mathrm{subscript}}$. The method then runs the merchant verification using Y_r from m. The merchant checks if σ is equal to $g^{\omega} Y_r T \mod q$. If true, the merchant verifies the transaction, storing m, ρ, e, σ, w, and T for use in the deposit method later.

7.1.4 Deposit

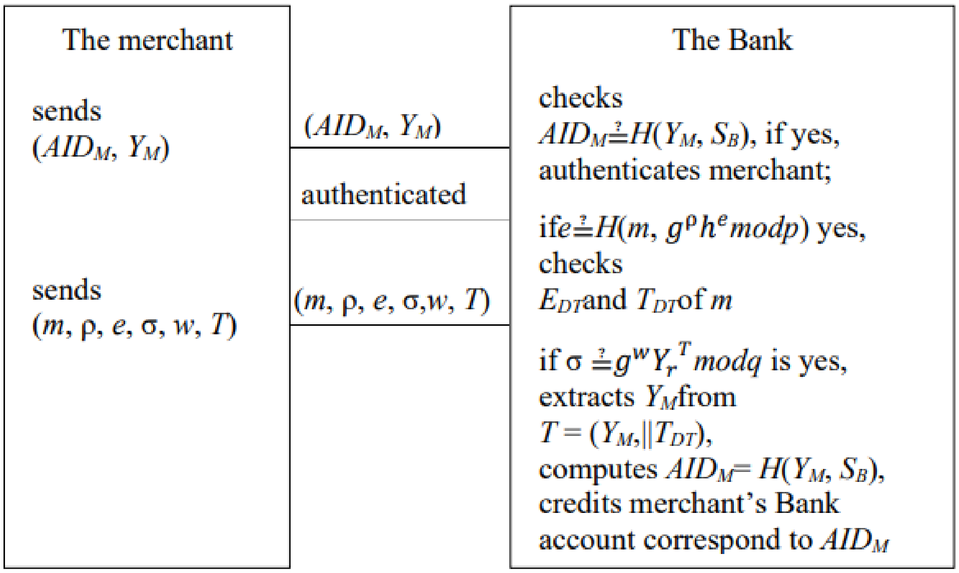

The merchant
sends
(AID_M, Y_M)
sends
$(m, \rho, e, \sigma, w, T)$
(AID_M, Y_M)
authenticated
$(m, \rho, e, \sigma,w, T)$
The Bank
checks
$AID_M \overset{?}{=} H(Y_M, S_B)$, if yes,
authenticates merchant;
if$e \overset{?}{=} H(m, g^{\rho} h^{e} modp)$ yes,
checks
$E_{DT}$and $T_{DT}$of m
if $\sigma \overset{?}{=} g^{w} Y_r^{\;T} modq$ is yes,
extracts $Y_M$from
$T = (Y_{M,} \| T_{DT})$,
computes $AID_M = H(Y_M, S_B)$,
credits merchant's Bank
account correspond to AID_M

Figure 4 :The dataflow diagram of theDeposit protocol.

The deposit method takes the previous values along with h and Y_M. h is calculated from pow$(g, -s, p)$. The merchant passes the AID_M, Y_M, and P_M for the bank to check if the coin is in the database. Then, e is verified with $e \overset{?}{=} \mathrm{H}(m, g^{\rho} h^{e} \mod p)$. The m is split to get the Y_r value and checks σ with $\sigma \overset{?}{=} g^{\omega} Y_r T \mod q$, if equal, the bank proceeds with the deposit. The Y_M is retrieved from splitting T, and AID_M is calculated from Y_M hashed with S_B or the secret signing key of the bank. The datetime is retrieved, and the e-coin is added to the deposit table, with the datetime serving as the current time and Y_M as the merchant's ID, along with the e-coin and credentials.

7.1.5 Renewal

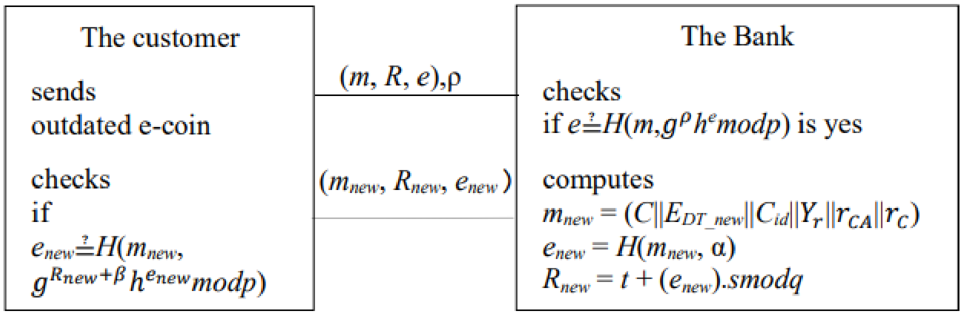

The customer
sends
outdated e-coin
checks
if
$e_{new} \overset{?}{=} H(m_{new},$
$g^{R_{new}+\beta} h^{e_{new}} modp)$
(m, R, e),ρ
$(m_{new}, R_{new}, e_{new})$
The Bank
checks
if $e \overset{?}{=} H(m, g^{\rho} h^{e} modp)$ is yes
computes
$m_{new} = (C||E_{DT\_new}||C_{id}||Y_r||r_{CA}||r_C)$
$e_{new} = H(m_{new}, \alpha)$
$R_{new} = t + (e_{new}).smodq$

Figure 5 : The dataflow diagramofthe Renewal protocol.

The renewal has two separate methods: `EcoinVerify` and `EcoinRenewal`. The first method runs as the customer, calculating $\rho = R + \beta_{\text{subscript}} \mod q$, and the bank verifies the e value of the coin with $H(m, g^{\rho}h^{e} \mod p)$. If the first method determines the e-coin is valid, then the second method starts, calculating the new m, e, and R values. M has its EDT value updated. The new values are recalculated as done in the withdraw method. Then the new e is checked if it is equal to the calculated e. All values are then returned to the caller, which is the customer in this case.

7.1.6 Tracing

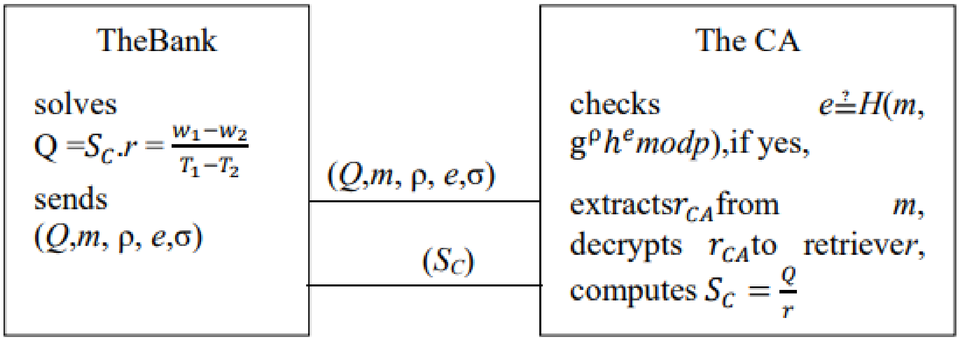

TheBank
solves
$Q = S_C.r = \frac{w_1 - w_2}{T_1 - T_2}$
sends
(Q, m, ρ, e, σ)
(Q, m, ρ, e, σ)
(S_C)
The CA
checks $e \overset{?}{=} H(m,$
$g^{\rho} h^{e} modp)$,if yes,
extracts$r_{CA}$from m,
decrypts $r_{CA}$to retriever,
computes $S_C = \frac{Q}{r}$

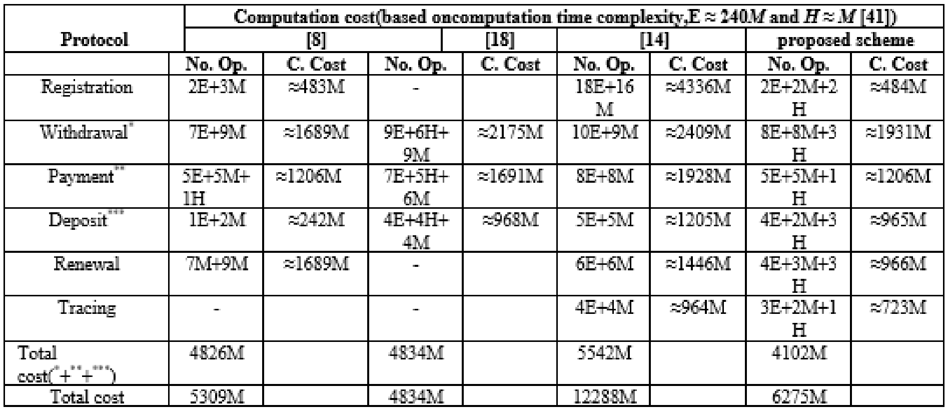
Figure 6 : The dataflow diagram of the Tracing protocol.

The tracing method takes an m, along with two w values (w_1 and w_2) and two T values (T_1 and T_2). Tracing runs during depositing; if the e-coin exists in both databases, then the tracing protocol runs using the w and T values retrieved from each table. The bank solves the equation $Q = S_C \cdot r = \frac{w_1 - w_2}{T_1 - T_2}$ and passes it to CA. Then the CA acts by getting R_{CA} from m, and decrypting using CA's private decryption key D_{CA}. The CA uses $\frac{S_C \cdot r}{\text{decrypted } r}$ value to get S_C, or the customer's secret signing key or ID.

8 Experimental Analyses

8.1 Prototype System Setup

TABLE 5:COMPARISION OF NUMBER OF OPERATIONS AND ITS EQUIVALENCE COMPUTATION COST REQUIRED



Protocol	Computation cost(based oncomputation time complexity,E ≈ 240M and H ≈ M [41])							
	[8]		[18]		[14]		proposed scheme	
	No. Op.	C. Cost	No. Op.	C. Cost	No. Op.	C. Cost	No. Op.	C. Cost
Registration	2E+3M	≈483M	-		18E+16M	≈4336M	2E+2M+2H	≈484M
Withdrawal*	7E+9M	≈1689M	9E+6H+9M	≈2175M	10E+9M	≈2409M	8E+8M+3H	≈1931M
Payment**	5E+5M+1H	≈1206M	7E+5H+6M	≈1691M	8E+8M	≈1928M	5E+5M+1H	≈1206M
Deposit***	1E+2M	≈242M	4E+4H+4M	≈968M	5E+5M	≈1205M	4E+2M+3H	≈965M
Renewal	7M+9M	≈1689M	-		6E+6M	≈1446M	4E+3M+3H	≈966M
Tracing	-		-		4E+4M	≈964M	3E+2M+1H	≈723M
Total cost(*+**+***)	4826M		4834M		5542M		4102M	
Total cost	5309M		4834M		12288M		6275M	

No. Op. = Number of Operations, and C. Cost = Computational cost

A detailed description of the experimental setup and implementation are provided in Colab Notebook which will be provided as supplementary materials. In table 5, computational costs are given interns of Modular and encryption operations for different protocols of the proposed scheme.

9 Strengths, Weaknesses, and Conclusions

9.1 Strengths

The main strength of this paper is the development of a robust mathematical framework for zero-knowledge proof, enhancing e-coin verification and anonymous payment capabilities.

9.2 Weaknesses

The system faces challenges related to computation costs and the practical implementation of divisible e-coins among client merchants.

9.3 Conclusions

The incorporation of Schnorr's BS protocol in the proposed offline e-payment scheme successfully addresses key security requirements. The scheme ensures customer anonymity and system efficiency, marking a significant advancement in offline electronic payment systems.

References

[1] Sattar J Aboud. Analysis of offline e-cash schemes. *International Journal*, 2(8):406–410, 2014. 3

[2] Yaser Baseri, Benyamin Takhtaei, and Javad Mohajeri. Secure untraceable off-line electronic cash system. *Scientia Iranica*, 20(3):637–646, 2013. 1, 2, 3

[3] Lynn Batten and Xun Yi. Off-line digital cash schemes providing untraceability, anonymity and change. *Electronic Commerce Research*, 19:81–110, 2019. 3

[4] Zlatko Bezovski. The future of the mobile payment as electronic payment system. *European Journal of Business and Management*, 8(8):127–132, 2016. 1

[5] Stefan Brands. Untraceable off-line cash in wallet with observers. In *Advances in Cryptology—CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings 13*, pages 302–318. Springer, 1994. 3

[6] Sébastien Canard, David Pointcheval, Olivier Sanders, and Jacques Traoré. Divisible e-cash made practical. In *IACR International Workshop on Public Key Cryptography*, pages 77–100. Springer, 2015. 3

[7] Agnes Chan, Yair Frankel, Philip MacKenzie, and Yiannis Tsiounis. Mis-representation of identities in e-cash schemes and how to prevent it. In *Advances in Cryptology—ASIACRYPT'96: International Conference on the Theory and Applications of Cryptology and Information Security Kyongju, Korea, November 3–7, 1996 Proceedings*, pages 276–285. Springer, 1996. 3

[8] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*, pages 199–203. Springer, 1983. 3

[9] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Advances in Cryptology—CRYPTO'88: Proceedings 8*, pages 319–327. Springer, 1990. 1

[10] Yalin Chen and Jue-Sam Chou. Cryptanalysis on "secure untraceable off-line electronic cash system". *Cryptology ePrint Archive*, 2014. 1, 3

[11] George Davida, Yair Frankel, Yiannis Tsiounis, and Moti Yung. Anonymity control in e-cash systems. In *Financial Cryptography: First International Conference, FC'97 Anguilla, British West Indies February 24–28, 1997 Proceedings 1*, pages 1–16. Springer, 1997. 3

[12] Ziba Eslami and Mehdi Talebi. A new untraceable off-line electronic cash system. *Electronic Commerce Research and Applications*, 10(1):59–66, 2011. 2

[13] Chun-I Fan, Vincent Shi-Ming Huang, and Yao-Chun Yu. User efficient recoverable off-line e-cash scheme with fast anonymity revoking. *Mathematical and Computer Modelling*, 58(1-2):227–237, 2013. 3

[14] Chun-I Fan, Bo-Wei Lin, and Shi-Ming Huang. Customer efficient electronic cash protocols. *Journal of Organizational Computing and Electronic Commerce*, 17(3):259–281, 2007. 1

[15] Chun-I Fan, Wei-Zhe Sun, Hoi-Tung Hau, et al. Date attachable offline electronic cash scheme. *The Scientific World Journal*, 2014, 2014. 2

[16] Wen-Shenq Juang and Horng-Twu Liaw. A practical anonymous multi-authority e-cash scheme. *Applied Mathematics and Computation*, 147(3):699–711, 2004. 1

[17] Md Abdullah Al Rahat Kutubi, Kazi Md Rokibul Alam, Rafaf Tahsin, GG Ali, Peter Han Joo Chong, and Yasuhiko Morimoto. An offline electronic payment system based on an untraceable blind signature scheme. *KSII Transactions on Internet and Information Systems (TIIS)*, 11(5):2628–2645, 2017. 2, 3, 4

[18] Alex Lipton, Thomas Hardjono, and Alex Pentland. Digital trade coin: towards a more stable digital currency. *Royal Society open science*, 5(7):180155, 2018. 3

[19] Rafael Martínez-Peláez and Francisco J Rico-Novella. New electronic cash model: a script anonym. In *Proc. of the IADIS International Conference on E-Commerce,(e-commerce'06)*, pages 392–396, 2006. 1

[20] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *2013 IEEE Symposium on Security and Privacy*, pages 397–411. IEEE, 2013. 3

[21] Jonas Muleravicius, Inga Timofejeva, Aleksejus Mihalkovich, and Eligijus Sakalauskas. Security, trustworthiness and effectivity analysis of an offline e-cash system with observers. *Informatica*, 30(2):327–348, 2019. 2

[22] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 2008. 3

[23] David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 252–265. Springer, 1996. 2, 4

[24] R Sai Anand and CE Veni Madhavan. An online, transferable e-cash payment system. In *Progress in Cryptology—INDOCRYPT 2000: First International Conference in Cryptology in India Calcutta, India, December 10–13, 2000 Proceedings 1*, pages 93–103. Springer, 2000. 1

[25] Eligijus Sakalauskas, Inga Timofejeva, Aleksėjus Michalkovič, and Jonas Muleravičius. A simple off-line e-cash system with observers. *Information Technology and Control*, 47(1):107–117, 2018. 3

[26] David Schwartz, Noah Youngs, Arthur Britto, et al. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 5(8):151, 2014. 4

[27] Lara Srivastava and Robin Mansell. *Electronic Cash and the Innovation Process: A Use Paradigm*. University of Sussex, SPRU, 1998. 3

[28] Felix Stalder. Failures and successes: Notes on the development of electronic cash. *The Information Society*, 18(3):209–219, 2002. 3

[29] Hitesh Tewari and Arthur Hughes. Fully anonymous transferable ecash. *Cryptology ePrint Archive*, 2016. 1

[30] Chih-Hung Wang. Untraceable fair network payment protocols with off-line ttp. In *Advances in Cryptology-ASIACRYPT 2003: 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30–December 4, 2003. Proceedings 9*, pages 173–187. Springer, 2003. 1, 3

[31] Hua Wang and Yanchuan Zhang. Untraceable off-line electronic cash flow in e-commerce. In *Proceedings 24th Australian Computer Science Conference. ACSC 2001*, pages 191–198. IEEE, 2001. 1, 3

[32] Jiangxiao Zhang, Lina Huo, Xia Liu, Chunrong Sui, Zhoujun Li, and Jinxin Ma. Transferable optimal-size fair e-cash with optimal anonymity. In *2015 International Symposium on Theoretical Aspects of Software Engineering*, pages 139–142. IEEE, 2015. 3

[33] Lin Zhong, Qianhong Wu, Jan Xie, Jin Li, and Bo Qin. A secure versatile light payment system based on blockchain. *Future Generation Computer Systems*, 93:327–337, 2019. 3, 4

A Contribution

- All the members actively contributed to the exploration of topics related to privacy and security, engaging in comprehensive research. In instances of ambiguity, collaborative discussions were initiated, ensuring clarity in both presentation and report phases.
- All members provided equal contribution from presentation to report according to respected one's research interest, concurrently fostering a collaborative environment where mutual assistance and support were extended.