Received: 13 April 2024 | Revised: 15 June 2024 | Accepted: 18 July 2024

IET Quantum Communication

DOI: 10.1049/qtc2.12109

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

An efficient and secure quantum blind signature-based electronic cash transaction scheme

Aman Gupta[1] | Gunja Venkat Chandra[2] | Nayana Das[3] | Goutam Paul[4]

[1]Department of Physics, Indian Institute of Technology, Roorkee, India

[2]Qulabs, Hyderabad, India

[3]Applied Statistics Unit, Indian Statistical Institute, Kolkata, India

[4]Cryptology and Security Research Unit, Indian Statistical Institute, Kolkata, India

Correspondence

Goutam Paul.
Email: goutam.paul@isical.ac.in

Abstract

The authors present a novel token exchange scheme with an example of an electronic cash (eCash) transaction scheme that ensures quantum security, addressing the vulnerabilities of existing models in the face of quantum computing threats. The authors' comprehensive analysis of various quantum blind signature mechanisms revealed significant shortcomings in their applicability to eCash transactions and their resilience against quantum adversaries. In response, the authors drew inspiration from D. Chaum's original classical eCash scheme and innovated a quantum-secure transaction framework. The authors detail the developed protocol and rigorously evaluate its security aspects. The protocol's adherence to critical security requirements such as blindness, non-forgeability, non-deniability, and prevention of double spending is analysed. Moreover, the scheme against Intercept and Resend, Denial of Service, Man-in-the-Middle, and Entangle-and-Measure attacks is rigorously tested. The authors' findings indicate a robust eCash transaction model capable of withstanding the challenges posed by quantum computing advancements.

KEYWORDS

quantum communication, quantum computing, quantum computing techniques, quantum cryptography, quantum information, telecommunication security

1 | INTRODUCTION

Quantum cryptography represents a major leap forward in securing digital communication. It uses principles from quantum mechanics to create secure ways of transferring information. Unlike traditional cryptography, which is based on hard math problems, quantum cryptography provides unconditional security based on the laws of quantum physics, such as the Heisenberg uncertainty principle [1] and quantum no-cloning theory [2]. One of the key developments in this field is Quantum Key Distribution (QKD) [3–7], which allows two or more people to share a secret key to encrypt and decrypt secret messages, making it nearly impossible for someone to eavesdrop without being detected. Quantum Secure Direct Communication (QSDC) [8–13] takes things a step further by letting people send secret messages directly, without even needing a key. This adds an extra layer of security in the world of quantum networks. Another key concept in this field is Blind Quantum Computing [14–19]. It lets users do computations on a quantum computer while keeping data and the nature of the computation hidden. This means one can use a quantum computer while ensuring the information stays secure.

In the digital transaction landscape, the proliferation and integration of electronic cash (eCash) systems and E-payment systems have been a pivotal advancement. These systems, designed to facilitate and secure monetary exchanges in the digital era, have gained significant traction. However, their security mechanisms are currently being challenged by the advent of quantum computing, especially with the development of Shor's algorithm [20], which threatens the cryptographic bedrock of traditional eCash schemes. This situation demands a rethinking of eCash protocol design to ensure resilience against these advanced computational threats.

In this paper, we introduce a novel method for creating an eCash system using quantum cryptography. Our protocol does

not use entanglement and thus makes it more practical for real-world applications. Inspired by Chaum's [21] earlier system using 'blind signatures', we have developed a quantum version of this technique for a secure eCash system.

Our system checks all the essential boxes for eCash—it is private, secure, and prevents the same money from being used twice. In Section 3, we detail how our system meets these key criteria. We have also tested our system against different types of common attacks and confirmed its robustness.

Beyond eCash, our method has potential applications in other areas such as online voting systems [22–26] or in creating secure quantum signatures [27–31]. This demonstrates the versatility and importance of our approach in the growing field of safe digital transactions. Overall, our research presents a novel and exciting perspective on digital security, particularly in financial transactions.

The paper is structured as follows: In the immediate subsections, we cover the history of research in blind signatures, quantum signatures, and eCash-related protocols1.1, and we describe the notations used in this paper 1.2 and we give the definitions used in the paper 1.3. In Section 2, we lay down the proposed transaction protocol, and the algorithms and control flow diagrams are also presented in this section. In Section 3, we cover the security aspects of our protocol, proving all the features, security parameters, and prevention/detection against several common attacks. In Section 4, we compare our protocol with several other similar proposed protocols. Finally, we conclude with the possible future scopes of this protocol (Section 5). Some of the standard concepts in classical and quantum computation used in this paper have been described in detail in Appendix.

1.1 | Literature survey

Untraceable payments that can be implemented using classical blind signatures were first introduced and implemented by Chaum [21]. In the early 1980s, Chaum worked on a system for secure, private electronic payments called blind signatures. The basic idea behind blind signatures is that a user can sign a message without seeing its contents, and then the signature can be verified by anyone who does not know the contents of the message. This allows for secure, private transactions without revealing any sensitive information.

In 1988, Chaum aimed to commercialise 'eCash', which used Chaum's blind signature protocol to ensure the privacy and security of transactions. Users could store eCash on their computers and use it to make payments to other eCash users without revealing any identifying information. DigiCash and Chaum's work on blind signatures and secure electronic payments laid the foundation for the development of other digital currencies, such as Bitcoin [32], which use similar cryptographic techniques to ensure the privacy and security of transactions. In the quantum era, using Shor's factoring algorithm, RSA can be easily broken [20, 33], and since the above-stated protocol is completely based on the RSA algorithm, such schemes are at a significant security risk.

Quantum digital signatures [34], initially proposed by D. Gottesman and L. Chuang in 2001, was one of the first quantum signature schemes. However, it has some notable drawbacks; first, the scheme requires the sender and receiver to share a private key in advance, which can be vulnerable to attacks such as eavesdropping. Second, the scheme requires a significant amount of classical communication to be exchanged between the sender and the receiver, which can limit its practicality in certain scenarios. Finally, the security of the scheme relies on the assumption that the sender and receiver are honest, and there is no guarantee of security if either party behaves maliciously.

Since 2010, many quantum blind signature-based E-payment protocols have been introduced in both centralised schemes [31, 35–41] and decentralised schemes using blockchain technology [42–45], and all these protocols, including ours, emerge as practical solutions to the problem of secure payments in the era of quantum computation while ensuring the requirements posed by a blind transaction scheme, namely, blindness, non-deniability, non-forgeability, no double spending and security against attacks. While most of the protocols suffice these requirements, they each suffer from some drawbacks or limitations that we detail in Sec. 4 while comparing these with our proposed protocol.

In 2021, G. Alagic et al. [46] presented evidence to support the notion that quantum signatures that are publicly verifiable are not possible. However, they put forward a possible solution known as 'signcryption'. We have used this idea of signcryption in our protocol as well.

Quantum signcryption is a cryptographic technique that is designed to provide both confidentiality and authenticity to a message sent between parties. It is a combination of two fundamental cryptographic primitives, namely signature and encryption. Signcryption enables a message to be signed and encrypted simultaneously, which provides both confidentiality and authenticity to the message. Unlike classical signcryption schemes, quantum signcryption schemes are not publicly verifiable, and quantum states can only be verified once. The sender of the message can verify that the message has not been tampered with during transmission, and the recipient can verify the authenticity of the sender. In summary, quantum signcryption is a powerful cryptographic tool that provides a secure means of sending messages between parties, ensuring that the message is authentic, confidential, and non-repudiable.

1.2 | Notations

Here, we describe the notations and symbols used throughout this paper.

- Z basis $\equiv \{|0\rangle, |1\rangle\}$
- $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.
- X basis $\equiv \{|+\rangle, |-\rangle\}$
- M: A random classical message, $M = \{m_n \cdots m_0 | m_i \in \mathbb{Z}^+\}$
- H: The hash value of a classical message.
- (c,d): A request packed where c is the credential and d is the denomination of the money requested.

- $(|I\rangle,p)$: SecurityCheck packet, where $|I\rangle$ is the original qubit $|Q\rangle$ interleaved with trap qubits $|T\rangle$ according to permutation p.
- λ: Security parameter.
- (L,b): A packet of random binary bits L and choice of a random basis b in the set, $\{|0\rangle,|1\rangle,|+\rangle,|-\rangle\}$
- G: Measurement outcome after measuring trap qubits $|T\rangle$ in random basis b
- $len(M)$: the length of a message M.
- $K_j^d = \left(K_x^d, K_z^d\right)$: The classical key corresponding to denomination d for both X and Z basis, respectively.
- $|E\rangle = |e_n \cdots e_j \cdots e_1\rangle$, $j \in \{1,\ldots,n\}$: Encoding of classical messages in a quantum state,
- where $|e_j\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\theta_j}|1\rangle\right) : \theta_j \in \{0, \pi/8, 2\pi/8, \ldots, 2\pi\}$
- $\left\{|S_E^i\rangle\right\}_{i=A,B}$: A quantum state representing the quantum blind signature (eCash) of an encoded message, which is divided into two parts $|S_E^A\rangle$ and $|S_E^B\rangle$.
- $|N\rangle$: Ancillary qubits in measure-entangle attack.
- D_A: Decoy photons where d_i represents the i-th qubit of D_A.
- $\mathcal{B}_i' \in \{X,Z\}$ basis
- $\Pr\left(d_i'' = d_i\right)$: The probability that the i-th qubit of the decoy photon $d_i = d_i''$.
- $\Pr\left(r_j' = d_i\right)$: The probability that the j-th qubit r_j' of R is equal to the i-th qubit d_i of decoy D.

1.3 | Definitions

Here, we give the relevant technical terminologies that are used throughout this paper.

Definition 1.1 (Digital Signature). A digital signature is a mathematical method used to authenticate digital messages or documents. When a message bears a valid digital signature, the recipient can trust that the message originated from a sender is recognised by them.

Definition 1.2 (Blind Signatures). Blind signatures, an extension of digital signatures, enhance privacy by enabling a user to obtain a signature on a message without the signer seeing the message's contents. The signer, when presented with the signed document later, cannot link it to the signing session or the user on whose behalf the message was signed. A blind signature scheme comprises three polynomial-time algorithms, denoted as $\{KeyGen, Sign, Verify\}$:

- **KeyGen:** This algorithm, given a security parameter, produces a key pair ($public\ key$,$secret\ key$).
- **Sign:** An interactive protocol between a signer S and a user U. S takes a secret key sk, while U takes a public key pk and a blind message $b \in B$ from a blinded message space B. The output of S is a view ν, and U obtains a signature σ.
- **QVerify:** This verification algorithm outputs 1 if σ is a valid signature and 0 otherwise.

Definition 1.3 (Quantum Blind Signature). Quantum blind signature is a scheme where the signer puts a signature on a quantum state without the knowledge of the state being signed and a verifier verifies the signature. In our case, the signer of the signature itself becomes the verifier at a later stage and it will become evident from our protocol (see Sec.2) why such a requirement is necessary.

Definition 1.4 (SecurityCheck). SecurityCheck is a protocol that allows one to check for intervention/ attack in the channel, while qubits are being sent from the sender to the receiver. The aim is to send some quantum information through a quantum channel where the receiver should be able to detect whether the received qubits were tampered with by some Eve in the channel. The sender interleaves trap qubits at random positions with the true qubits and sends it. The receiver then checks the trap qubits for correctness. The details of the protocol are given in Sec. 2.4.

Definition 1.5 (MessageGen). $MessageGen() \rightarrow M$ is a function that produces a random classical message string, M, when triggered. When the eCash request is initiated by a client, the function, $MessageGen()$ is automatically called.

Note: In a paper for an improved quantum E-Payment System [41], they used the purchase information as the message M. Such a format of message and its corresponding hash are useful when in dispute; in the court of law, it can be proven that only this purchase information can be used to generate the same hash, proving the purchase.

Definition 1.6 (Hashing). $Hashing(M) \rightarrow H$ produces a hash value H of the message M. A detailed description of hashing can be found in Appendix 8.

Secure and standard hash functions such as SHA-2 can be used in our protocol, since they are practically collision-resistant in classical computation environments and there are not any quantum algorithms for collision attacks [47], as of this work. Additionally, in our scenario, second pre-image resistance, which is a weaker property than collision resistance, is a much more significant property in terms of the security of our protocol. Therefore, these hash functions are fairly secure to use here.

Definition 1.7 (Encode). $Encode(M) \rightarrow |E\rangle$ takes in a message or hash and outputs the encoding of the classical message into a quantum state $|E\rangle$ in X basis. One suggested encoding method that can be used is phase encoding (detailed, with an example in Appendix 8).

Definition 1.8 (DenomKeyGen). $DenomKeyGen(len(H),d) \rightarrow K$ function is triggered by the bank to generate a pair of keys. $DenomKeyGen$ takes in the length of hash H of message and the denomination d of the requested money. These keys

are not random, and for each denomination, there exists a unique pair of keys, K^d, and if the $len(H) > len(K)$, then repeat the keys as follows:

$$K_d = (K_x, K_z)^{\otimes \alpha}, \quad \text{where} \quad \alpha = \left\lceil \frac{len(H)}{len(K)} \right\rceil$$

Definition 1.9 (Req_packet). $Req\_packet(c,d)$ is a request packet container that stored (c,d). When a request to either deposit or issue an eCash is made, then the request packet is sent along with the quantum blind signature to the bank.

Definition 1.10 (SignCryption). $SignCryption(|E\rangle, K) \rightarrow |S_E\rangle$ function inputs an encoded quantum state (generated from 'Encode') and a key K associated to a denomination(d) (generated from 'DenomKeyGen') to produce a quantum blind signature S_E. Here, the SignCryption is initiated by the bank to put a signature on a received quantum message, $|E\rangle$.

Since the major security of our protocol depends on the structure of the eCash token and techniques used during the exchange of the tokens, and the security of hashes and SignCryption are standard structures used to enhance the security, therefore, any SignCryption scheme that is not trivial can be used.

However, a scheme that maps input to a maximally mixed state will be the most suited one. One such very suitable and secure SignCryption scheme for our protocol can be quantum one-time padding [14] of the encoded quantum state repeated α times (α has been defined in Def.1.8). An example of SignCryption using this scheme has been given in Appendix

Definition 1.11 (DivideSign). $DivideSign(|S_E\rangle) \rightarrow \left(|S_E^A\rangle, |S_E^B\rangle\right)$ inputs the signature $|S_E\rangle$ and produces a packet $\left(|S_E^A\rangle, |S_E^B\rangle\right)$, where the packet has two parts of the signature. It is initiated by the bank which divides the complete signature into two parts such that both the parts have a similar information content.

Definition 1.12 (Credeb_Mem). Credeb_Mem is a hybrid classical-quantum memory that the bank uses to store the quantum signatures and the corresponding classical req_packet info.

Definition 1.13 (Rand_bit_basis). $Rand\_bit\_basis()$ function, when initiated, produces some random bits L and corresponding random basis b, where the $\delta > len(b) = len(L) > 2$. The choice of δ depends on the extent of security requirement and computational power. We chose $\delta = 2len(|Q\rangle)$.

Definition 1.14 (Encode_in_basis). $Encode\_in\_basis(L,b) \rightarrow |T\rangle$, the function is much similar to the $Encode()$ function defined in Def. 1.7, but instead of encoding all classical messages in X basis, it encodes in the input bases b to output trap qubits $|T\rangle$.

Definition 1.15 (QPermute). $QPermute(|Q\rangle, |T\rangle) \rightarrow (|I\rangle, p)$ takes a quantum state $|Q\rangle$ and interleaves it with trap qubits $|T\rangle$ in random positions, and the final interleaved quantum state $|I\rangle$ as well as the information of these random positions p are returned.

Definition 1.16 (inv_QPermute). $inv\_QPermute(|I\rangle, p) \rightarrow (|Q\rangle, |T\rangle)$, as the name suggests, is the inverse of the $QPermute()$ operation given in Def. 1.15, and it takes in the interleaved qubits $|I\rangle$ and separates the quantum state $|Q\rangle$ from the trap qubits $|T\rangle$ based on p.

Definition 1.17 (QMeasure). $QMeasure(|T\rangle, b)$ function measures the quantum state $|T\rangle$ in basis b to generate classical string G as output.

Definition 1.18 (QVerify). $QVerify(|Q_1\rangle, |Q_2\rangle) \rightarrow bool$ function takes in any general two quantum states. It compares the two quantum states using the CSWAP test (Appendix 8), returns false if they do not match, and initiates the 'Abort_Transaction' protocol; while if they match, it returns true and follows the instruction to complete the transaction.

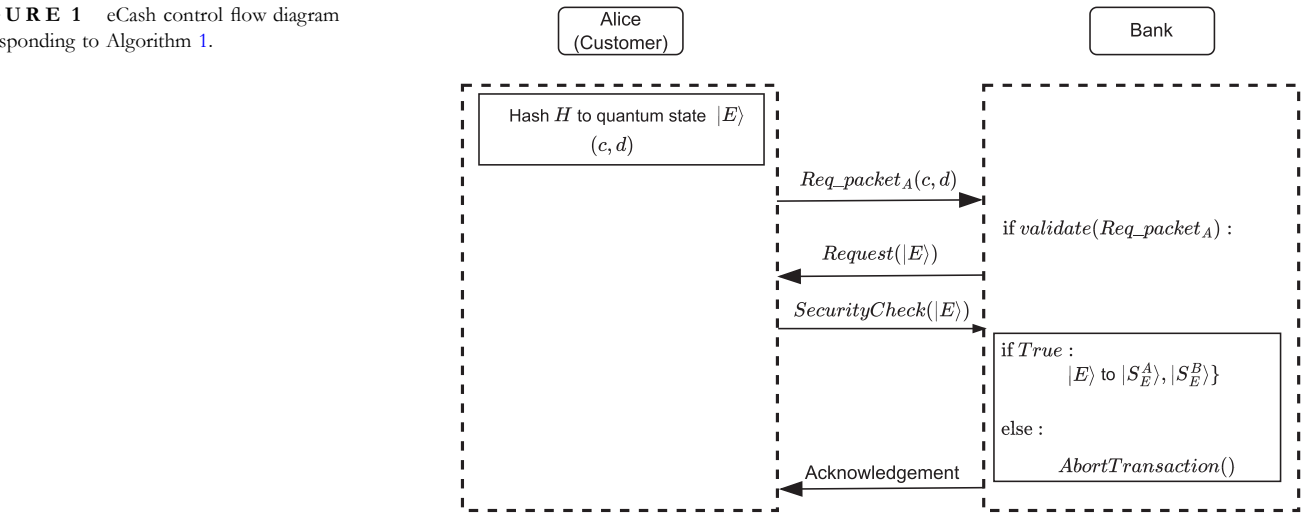
Definition 1.19 (Abort_Transaction). Abort_Transaction protocol is initiated at any stage when either the 'security check' or 'verify' process returns a failure result. The details of the protocol is mentioned in Sec. 2.3.

2 | OUR eCASH PROTOCOL

In this section, we provide the algorithm for the complete eCash transaction protocol and the corresponding control flow diagrams (see Figures 1 and 2). The protocol is divided into two stages: in the first stage, 'Issue eCash' (Sec. 2.1), the first half of the eCash/signature (in this work, we have used eCash and signature interchangeably) token is issued to the requester (Alice) after verifying the satisfaction of issue criteria by the issuer (Bank), and this token can now be sent to a merchant (Bob). At this stage, the transaction is partially completed. The next stage, 'Spend eCash' (Sec. 2.2), is initiated when Alice wishes to pay the eCash to Bob. This stage completes the transaction between Alice and Bob, while the bank is blind to the connection between Alice and Bob. Sec. 2.3 details the protocol to cancel a transaction at any stage.

2.1 | Issue eCash

The issue eCash is an interactive communication between the bank and Alice; much like the issuing of real cash, the bank issues the eCash to Alice after verifying her credentials for the existence



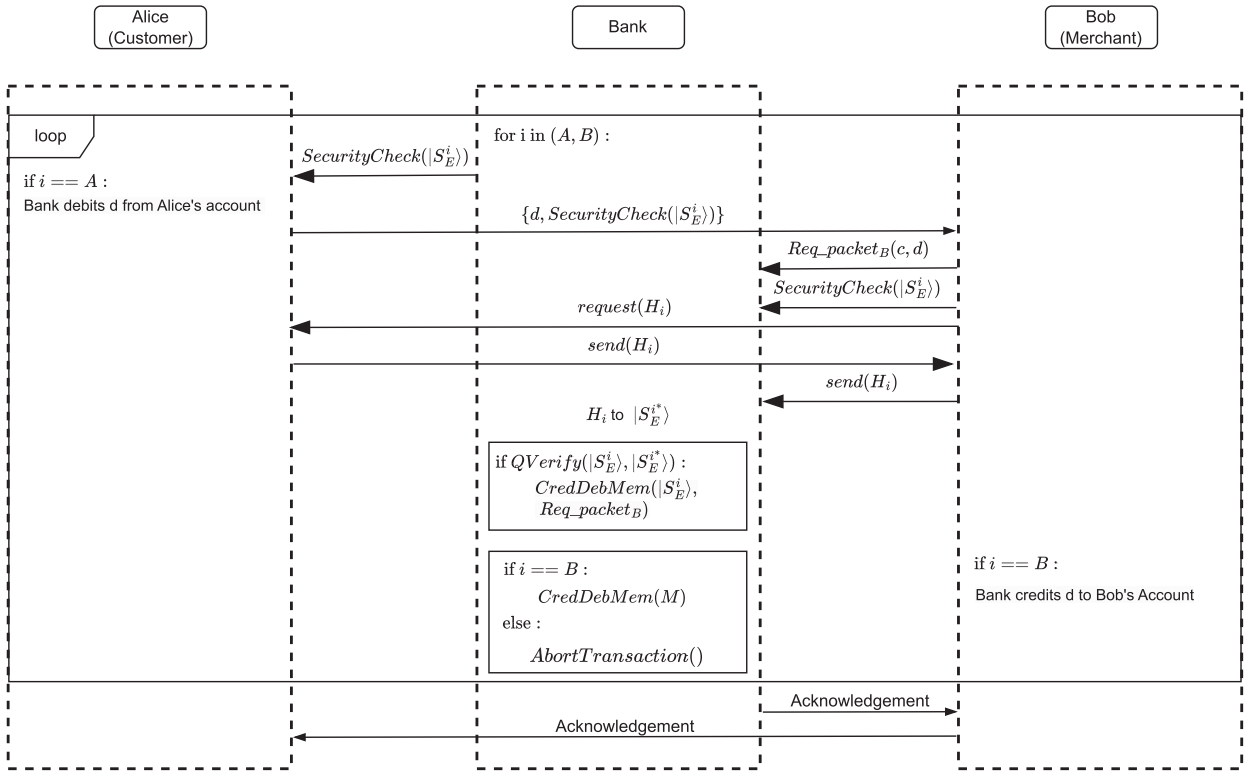


FIGURE 1 eCash control flow diagram corresponding to Algorithm 1.

FIGURE 2 eCash control flow diagram corresponding to Algorithm 2.

of the account and a sufficient balance for a successful transaction. Algorithm 1 details the process of issuing eCash.

2.2 | Spend eCash

Once the first half part of eCash has been successfully issued, Alice is free to spend it with any merchant (Bob). The spend eCash, detailed in Algorithm 2, process is executed by Alice via sending the half part of eCash which is verified by the bank and again the other half is sent which again is satisfied by the bank. In this process the actual debit and credit of money takes place and here the interaction takes place between all parties, that is, Alice, the bank, and Bob.

2.3 | Abort transaction protocol

During the transaction process, if any SecurityCheck or verification process fails, then this protocol is automatically initiated to revert back to the original state of the system described in Algorithm 3.

Algorithm 1 Issue eCash protocol

Input: None
Output: $|S_E^A\rangle, |S_E^B\rangle$
// Step 1: Alice generates a random message, hashes it, and encodes it into a quantum state
$M \leftarrow$ MessageGen();
$H \leftarrow$ Hashing(M);
$|E\rangle \leftarrow$ Encode(H);
; /* An example encoding scheme is given in Appendix 8 */
// Step 2: Alice sends a request packet to the bank
$Req\_packet_A \leftarrow \{c, d\}$;
; /* Here c is Alice's credentials like bank A/C No., etc., and d is the denomination */
send(Req_packet_A);
// Step 3: Bank validates the request packet and requests Alice to send her encoded qubits
if validate(Req_packet_A) **then**
  send($SecurityCheck(|E\rangle)$);
else
  initiate AbortTransaction();
end
// Step 4: Alice sends her encoded qubits using a SecurityCheck protocol
$SecurityCheck(|E\rangle) \leftarrow$ receive();
if SecurityCheck($|E\rangle$) **then**
  goto Step 5;
else
  initiate AbortTransaction();
end
// Step 5: Bank generates keys and performs SignCryption on the encoded qubits
$K \leftarrow$ DenomKeyGen($len(|E\rangle), d$);
$|S_E\rangle \leftarrow$ SignCryption($|E\rangle, K$);
; /* An example $SignCryption$ scheme is given in Appendix 8 */
// Step 6: Bank divides the signature and stores one part with the request packet in Alice's $Credeb\_Mem$ and sends the other to Alice
$\{|S_E^A\rangle, |S_E^B\rangle\} \leftarrow$ DivideSign($|S_E\rangle$);
storein Credeb_Mem($|S_E^B\rangle, Req\_packet_A$);
; /* $Credeb\_Mem$ here is associated with Alice */
// Step 7: Send the half SignCryption to Alice using SecurityCheck
send($SecurityCheck|S_E^A\rangle$);
// Step 8: Alice receives the signature and accepts it if the security check passes
$SecurityCheck(|S_E^A\rangle) \leftarrow$ receive();
if SecurityCheck($|S_E^A\rangle$) **then**
  accept($|\mathrm{S_E^A}\rangle$);
else
end

2.4 | SecurityCheck protocol

At any stage of transmission in the transaction process, the SecurityCheck protocol is performed in order to detect any eavesdropping. The process is defined in Algorithm 4.

3 | SECURITY ANALYSIS

In this section, we show and prove that our proposed protocol satisfies non-forgeability, non-deniability, blindness, and no double spending, and it is also secure against any attacks by an eavesdropper, *Eve*.

3.1 | Non-forgeability

Here, we show that Alice, Bob, or any third party cannot forge the signature produced by the bank. This is achieved through a signature protocol that employs quantum operations that are dependent on a private key (see 1.8). The process of generating a quantum signature involves manipulating a quantum state using the private key. This key is unique to the bank and is not shared with any unauthorised entities. Therefore, any attempts to generate a signature qubit by a forger who does not have access to the private key will be unsuccessful. Additionally, even if the forger tries to copy the quantum state, it is not possible due to the 'No cloning theorem [2]', which states that it is impossible to create an identical copy of an unknown quantum state. Therefore, the forger cannot reproduce the quantum signature using the copied quantum state. In other words, the use of quantum operations and a private key ensures that the signature produced by the bank is secure and cannot be forged by any unauthorised entities. Attempts to copy the quantum state or obtain the private key will be unsuccessful, thus providing an additional layer of security to the signature protocol.

Algorithm 2 Spend eCash protocol

Input: $|S_E\rangle$,d
Output: Bool
for *i* **in** *[A,B]* **do**
  // Step 1: Alice sends $|S_E^i\rangle$ and denomination d to Bob
  send($SecurityCheck(|S_E^i\rangle), d$);
  // Step 2: Bob Completes SecurityCheck, adds his info, and sends to the bank. **if** SecurityCheck($|S_E^i\rangle$) **then**
    if *i* == *A* **then**
      $Req\_packet_B \leftarrow \{c, d\}$;
      ; /* Here c is Bob's credentials, like bank A/C No., etc. */
    else
      pass
    end
    send($Req\_packet_B, SecurityCheck(|S_E^i\rangle)$)
  else
    initiate AbortTransaction();
  end
  // Step 3: Bank initiates SecurityCheck and validates the request packet
  if validate(Req_packet_B) **and** $SecurityCheck(|S_E^i\rangle)$ **then**
    if *i* == *A* **then**
      Debit Alice's balance with d
    else
      pass
    end
    request(H_i); ; /* H_i is the classical hash of message corresponding to $|S_E^i\rangle$ see Def. 1.3.11 */
    // Step 4: Alice sends M_i to Bob which he sends to bank
    send(H_i); /* Alice sends to Bob who sends it to bank */
    // Step 5: With key K (corresponding to d) and H_i bank generates a new signature
    $|E_i^*\rangle \leftarrow$ Encode(H_i);
    $|S_E^{i^*}\rangle \leftarrow$ SignCryption($|E_i^*\rangle, K_i$);
    //Step 6: Bank now uses the verify method to check if the two signatures are the same and requests the next part
    if QVerify($|S_E^i\rangle, |S_E^{i^*}\rangle$) **then**
      storein Credeb_Mem($|S_E^i\rangle, Req\_packet_B$);
      if *i*==*B* **then**
        Credit Bob's Account with d;
        storein Credeb_Mem(M)
      else
        pass
      end
    else
      initiate AbortTransaction();
    end
  else
    initiate AbortTransaction();
  end
end

3.2 | Non-deniability

Neither Alice nor Bob can deny their possession of the signature from the bank in a two-step transaction process. This is because the transaction involves a message that is sent from Alice to Bob and then from Bob to the bank. In a successful transaction, Alice receives the signed message from the bank, and Bob must receive the original message along with the signature from Alice, and then Bob passes it to the bank for verification. If a dispute arises, Alice cannot deny the purchase because the message that was sent from her to Bob and then to the bank has a specific hash value that can only be reproduced with the same message. This hash value acts as a unique identifier for the message and can be verified by the bank. Therefore, the bank can confirm that Alice did indeed authorise the transaction, and the dispute can be resolved accordingly. This process provides a secure and transparent way to conduct transactions, as it prevents both Alice and Bob from denying their involvement in the transaction. It also ensures that the transaction cannot be tampered with or altered, providing an additional layer of security to the system.

3.3 | Blindness

In the proposed protocol, Alice encodes a hash value of her message, which blinds the message in two steps. This is done by sending the encoded quantum state of a classical hash,

Algorithm 3 Abort transaction protocol

Input: λ
; /* Security parameter, $\lambda \in \{0, 1\}$ */
Output: Bool
; /* 0 means we have to initiate abort transaction protocol */
if $\lambda == 0$ **then**
  delete($|S_E\rangle, |E\rangle$);
  reset(Credeb_Mem)
else
  pass
end

Algorithm 4 SecurityCheck protocol

Input: $|Q\rangle$
Output: Bool
// Step 1: Sender has some qubits $|Q\rangle$ to send. It generates some trap qubits $|T\rangle$ in random basis b
$(L, b) \leftarrow$ Rand_bit_basis();
$|T\rangle \leftarrow$ Encode_in_basis(K, b);
// Step 2: Sender now interleaves the original qubits $|Q\rangle$ with trap qubits $|T\rangle$
$(|I\rangle, p) \leftarrow$ QPermute($|Q\rangle, |T\rangle$);
send($|I\rangle, p$);
// Step 3: Receiver receives the qubits and separates the Trap from original qubits
$(|I\rangle, p) \leftarrow$ receive();
$(|Q\rangle, |T\rangle) \leftarrow inv\_QPermute(|I\rangle, p)$;
send Acknowledgement;
// Step 4: Sender sends the basis and random bits
send(L, b);
// Step 5: Receiver measures the trap qubits in b bases and compares the measurement with L
$G \leftarrow$ QMeasure($|T\rangle, b$);
if $G == L$ **then**
  accept$|Q\rangle$;
  return $True$
else
  return $False$
end

which means that any party attempting to obtain the message will have to measure the qubits. This measurement process destroys the information contained in the qubits, making it impossible to obtain the original message. Furthermore, even if the qubits are measured, the information obtained is most likely incorrect because the basis of the measurement is unknown. In other words, measuring the qubits without the knowledge of the basis results in random values, which do not reveal any information about the original message. Additionally, the hash value of the message is being used and not the message itself, which means that it cannot be inverted to obtain the original message. This hash value acts as a unique identifier for the message and can be used to verify its authenticity without revealing the original message.

3.4 | No double spending

The signature produced by the bank using a quantum state is a unique quantum object that cannot be copied due to the 'No cloning theorem' [2]. Therefore, any attempts to create a duplicate of the signature will destroy the original signature. This property of the quantum signature ensures that it cannot be double-spent. Double spending occurs when a user spends the same funds more than once. In traditional digital transactions, this can be achieved by creating a duplicate of the digital signature, but this is not possible with a quantum signature. Therefore, the use of a quantum state to generate a signature provides security to the transaction, preventing the possibility of double spending. This ensures that the transaction is conducted securely, and the funds are transferred only once.

3.5 | Security against common attacks by *Eve*

We now show that our protocol is secure against some common attacks. Here, we discuss the intercept-and-resend attack, Denial-of-Service attack, man-in-the-middle attack, and

entangle-measure attack. For each case, our proposed protocol is secure.

3.5.1 | Intercept-and-resend attack

Let us consider *Eve* intercepting the qubit sequence S from the quantum channel. As the qubits in S are prepared in a random basis, *Eve* cannot gain any information by measuring those qubits. The most she can do is measure the qubits of S in either the Z or X basis and resend those measured qubits to Bob. However, in this case, *Eve* does not obtain any useful information about the secret message, and the legitimate parties detect her and terminate the protocol during security checking. Let the set of decoy photons D_A contain l qubits.

We will now calculate the probability that the sender and receiver can detect *Eve*. Let the i-th qubit of D_A be d_i prepared in basis $\mathcal{B}_i \in Z,X$, and let *Eve* choose the basis $\mathcal{B}_i'$ to measure d_i and obtain d_i'. During security checking, Bob measures d_i' in $\mathcal{B}_i$ and obtains the result d_i''. The winning probability of *Eve* for the i-th decoy qubit is given by

$$\begin{aligned}
&\Pr\left(d_i'' = d_i\right)\\
&= \Pr\left(d_i'' = d_i|\ \mathcal{B}_i = \mathcal{B}_i'\right)\Pr\left(\mathcal{B}i = \mathcal{B}_i'\right)\ +\\
&\qquad \Pr\left(d_i'' = d_i|\ \mathcal{B}_i \neq \mathcal{B}_i'\right)\Pr\left(\mathcal{B}_i \neq \mathcal{B}_i'\right)\\
&= \frac{1}{2}\left\{\Pr\left(d_i'' = d_i|\ \mathcal{B}_i = \mathcal{B}_i'\right) + \Pr\left(d_i'' = d_i|\ \mathcal{B}_i \neq \mathcal{B}_i'\right)\right\}\\
&= \frac{1}{2}\left(1 + \frac{1}{2}\right) = \frac{3}{4}.
\end{aligned}$$

Thus, the probability that the sender and the receiver can detect the existence of *Eve* is $1 - \left(\frac{3}{4}\right)^l > 0$.

3.5.2 | Denial-of-service (DoS) attack

Eve aims to tamper with the secret message by employing a DoS attack. To achieve this, she intercepts the sequence S and applies a certain unitary operation $\mathcal{U}$ to each qubit of S. However, the legitimate parties can detect this action during the security checking procedure and terminate the protocol. Since the Pauli matrices I, σ_x, $i\sigma_y$, and σ_z form a basis for the space of all 2×2 Hermitian matrices [48], $\mathcal{U}$ can be expressed as a linear combination of these basis vectors. Let $\mathcal{U} = w_1 I + w_2\sigma_x + w_3 i\sigma_y + w_4\sigma_z$ where $\sum_{j=1}^{4} w_j^2 = 1$ as $\mathcal{U}$ is unitary.

Next, we calculate *Eve*'s winning probability for each decoy qubit $d \in D_A$ (or $d \in D_A'$). Firstly, we compute the winning probabilities p_1, p_2, p_3, and p_4 of *Eve* when applying the Pauli matrices I, σ_x, $i\sigma_y$, and σ_z, respectively. We find that $p_1 = 1$, as applying I on d does not change its state; $p_2 = 1/2$, as σ_x changes the state of a decoy qubit d only if $d \in |0\rangle, |1\rangle$; $p_3 = 0$, as $i\sigma_y$ always changes the state of a decoy qubit; and $p_4 = 1/2$, as σ_z changes the states in the X basis. Therefore, *Eve*'s winning probability is $p = \sum_{j=1}^{4} p_j w_j^2 < 1$ unless $\mathcal{U} = \mathcal{I}$ (which is equivalent to no attack by *Eve*). As a result, during the security check processes, the legitimate parties can detect this eavesdropping with a probability of $1 - p^l > 0$.

3.5.3 | Man-in-the-middle attack

When the sender sends the sequence S to the receiver, *Eve* intercepts S and keeps this with her. She prepares another set of qubits R and sends it to the receiver. In this case, the legitimate parties can also realise the existence of *Eve* and abort the protocol and terminate the protocol. We now calculate the detection probability of *Eve* when she intercepts S. Let the i-th decoy qubit of D_A be d_i and suppose it is the j-th qubit of S. Also, let *Eve* prepare r_j as the j-th qubit of R. Let the preparation bases of d_i and R_j be $\mathcal{B}_1$ and $\mathcal{B}_2$ respectively. In the security check process, Bob measures r_j in basis $\mathcal{B}_1$ and gets r_j'. Thus, the winning probability of *Eve* for the i-th decoy qubit is as follows:

$$\begin{aligned}
&\Pr\left(r_j' = d_i\right)\\
&= \Pr\left(r_j' = d_i|\ \mathcal{B}_1 = \mathcal{B}_2\right)\Pr(\mathcal{B}_1 = \mathcal{B}_2)\ +\\
&\qquad \Pr\left(r_j' = d_i|\ \mathcal{B}_1 \neq \mathcal{B}_2\right)\Pr(\mathcal{B}_1 \neq \mathcal{B}_2)\\
&= \frac{1}{2}\left[\Pr\left(r_j' = d_i|\ \mathcal{B}_1 = \mathcal{B}_2,\ r_j = d_i\right)\Pr\left(r_j = d_i\right) +\right.\\
&\left.\Pr\left(r_j' = d_i|\ \mathcal{B}_1 = \mathcal{B}_2,\ r_j \neq d_i\right)\Pr\left(r_j \neq d_i\right) + \frac{1}{2}\right] = \frac{1}{2}
\end{aligned}$$

Hence, they detect *Eve* with probability $1 - (1/2)^l > 0$.

3.5.4 | Entangle-measure attack

Eve can use the following attack to steal partial information: Firstly, she intercepts the qubits of the sequence S' and prepares an ancillary state $|N\rangle$. Next, she applies a unitary operation U_N to the joint states of the qubits in S and $|N\rangle$ in such a way that the composite system becomes entangled. Let d_i denote the i-th decoy state in D_A. After applying U_N, let d_i' be the resulting decoy state. However, the effect of the unitary operation U_N on the second set of decoy photons is as follows:

$$\begin{aligned}
U_N|0\rangle|N\rangle &= \alpha_0|0\rangle|N_{00}\rangle + \beta_0|1\rangle|N_{01}\rangle,\\
U_N|1\rangle|N\rangle &= \alpha_1|0\rangle|N_{10}\rangle + \beta_1|1\rangle|N_{11}\rangle.
\end{aligned} \tag{1}$$

Since U_N is unitary, we must have

$$|\alpha_0|^2 + |\beta_0|^2 = 1,\ |\alpha_1|^2 + |\beta_1|^2 = 1,\ \alpha_0\alpha_1^* + \beta_0\beta_1^* = 0. \tag{2}$$

Thus, when the decoy state d_i is prepared in the Z basis, the error rate is $e = |\beta_0|^2 = |\alpha_1|^2$. Further, we get $U_N|\pm\rangle|N\rangle = \frac{1}{\sqrt{2}}(|+\rangle|N_{\pm+}\rangle + |-\rangle|N_{\pm-}\rangle)$, where

- $|N_{++}\rangle = \frac{1}{\sqrt{2}}\left(\alpha_0|N_{00}\rangle + \beta_0|N_{01}\rangle + \alpha_1|N_{10}\rangle + \beta_1|N_{11}\rangle\right)$,
- $|N_{+-}\rangle = \frac{1}{\sqrt{2}}\left(\alpha_0|N_{00}\rangle - \beta_0|N_{01}\rangle + \alpha_1|N_{10}\rangle - \beta_1|N_{11}\rangle\right)$,
- $|N_{-+}\rangle = \frac{1}{\sqrt{2}}\left(\alpha_0|N_{00}\rangle + \beta_0|N_{01}\rangle - \alpha_1|N_{10}\rangle - \beta_1|N_{11}\rangle\right)$,
- $|N_{--}\rangle = \frac{1}{\sqrt{2}}\left(\alpha_0|N_{00}\rangle - \beta_0|N_{01}\rangle - \alpha_1|N_{10}\rangle + \beta_1|N_{11}\rangle\right)$.

If the decoy state d_i is prepared in the X basis, the receiver will measure the first qubit d_i' of the entangled state $U_N|+\rangle|N\rangle$ or $U_N|-\rangle|N\rangle$ in the X basis. As a result, the correct outcome is obtained with a probability of $1/2$ (which is equal to the random guess probability and so *Eve* is not getting any advantage), leading to an error rate of $1/2$. Hence, the legitimate parties can detect eavesdropping by *Eve* in the security check processes based on the error rate introduced in the communication process.

4 | COMPARISON

The necessity for quantum-enhanced E-payment schemes to ensure information-theoretic security and transaction blindness in environments where classical encryption is vulnerable to quantum attacks has been recognised since 2010 [49]. Wen et al. [49] introduced the first quantum E-payment scheme based on quantum group signatures, employing QKD and a one-time pad for security. They [36] later proposed a similar scheme using quantum proxy blind signatures, but Cai et al. [50] demonstrated its vulnerability to malicious merchants. Zhang et al. [35] improved upon this by using a four-qubit entangled state and a trusted third party, while Guo et al. [37] offered a solution without entanglement to address the issue of malicious merchants. However, these protocols relied on a single-bank model. Tiliwalidi et al. [38] developed a multi-bank E-payment protocol using quantum proxy blind signatures and six-qubit entangled states. Our protocol, while also single-bank, ensures security through quantum blind signatures and one-time pads without requiring entanglement, QKD or a trusted third party, and it includes mechanisms for detecting malicious parties.

All the aforementioned protocols are based on centralised payment schemes, while there are others that are based on decentralised schemes (Blockchain [51] technology). In 2019, Zhang et al. [42] introduced a novel E-payment protocol combining blockchain and quantum signatures based on six-qubit entangled states. In 2021, Gou et al. [43] proposed a protocol using blockchain, QKD, and quantum proxy blind signatures, incorporating three-qubit entangled states for controlled quantum teleportation. These protocols, however, are limited by the Pirandola–Laurenza–Ottaviani–Banchi (PLOB) bound [52], which states that quantum capacity decreases with increasing distance between parties. To address this, in 2023, Li et al. [45] proposed a blockchain-based quantum E-payment scheme utilising twin-field quantum key distribution, which has been proven to overcome the PLOB bound constraints.

Our proposed quantum-enhanced eCash protocol follows a centralised payment model regulated by a central authority, such as banks, and ensures the detection and prevention of malicious behaviour even from the central authority. Blockchain technology, while ensuring transaction validity, incurs high computational costs due to redundant data processing. Although a detailed comparison of computational costs between centralised quantum systems and blockchain technology is lacking, exploring centralised systems in a quantum environment as scalable alternatives to blockchain is worthwhile. Our protocol avoids the computational overhead associated with blockchain technology by relying on quantum computation and hashing, ensuring security against malicious central authorities without QKD or entanglement. Additionally, to use quantum blind signatures and one-time pads, our protocol enhances transaction blindness with classical hashing, ensuring that both the transaction information and the connection between the sender (Alice) and receiver (Bob) remain confidential from third parties, including banks.

Table 1 provides a comparative analysis of our protocol against other protocols, highlighting key parameters. Unlike all the compared protocols that are E-payment methods similar to a digital transaction, our protocol resembles more of a real cash, where a quantum state is the token (hence eCash); therefore, even the connection between the sender (Alice) and the receiver (Bob) is also blinded to any other party. However, it is limited to a single-bank model and is subject to the PLOB bound, necessitating the use of quantum repeaters in quantum networks.

5 | CONCLUSION

In this work, we have given a simple protocol to implement an example eCash-based transaction scheme with quantum protocols. We also proved the security and feature claim of this protocol and we compared our protocol with several other similar proposed works. This protocol, unlike most, does not require entanglement between parties as a prerequisite, QKD, a trusted third party, or sharing of key, which is a much-desired feature for a private eCash scheme deployed in a real market. It also being a centralised transaction scheme that does not incur the additional computational overhead inherent to a decentralised blockchain technology.

Our proposed protocol can prove to be a potential transaction scheme for the future that can provide privacy of user data and security during an electronic transaction through the features of quantum mechanics. This protocol can ensure that the existing transaction method can be enhanced with absolute security against any malicious party, and since it is not a new currency, unlike cryto, it is just a new transaction scheme, and the existing system remains intact. As mentioned in the previous section, our protocol is a single-bank model that is subject to the PLOB bound and therefore it leaves the scope

TABLE 1 Comparison of different quantum E-payment protocols.

		Quantum resources		
Protocol	**Scheme type**	**Entanglement**	**QKD**	**Measurement**
Zhang et al. [35]	Centralised	4-ES	BB84	BM, SM
Wen et al. [36]	Centralised	GHZ	BB84	BM
Guo et al. [37]	Centralised	No	BB84	SM
Tiliwalidi el al. [38]	Centralised	6-ES	BB84	VMN, SM, TM, ThM
J. Zhang et al. [42]	Blockchain	6-ES	BB84	GHZM, BM, SM
Gou et al. [43]	Blockchain	3-ES	BB84	SM, BM
E. Li et al. [44]	Blockchain	BS	MDI-QKD	SM, BM
Niu et al. [40]	Centralised	4-CS	BB84	VNM, SM
Xie et al. [41]	Centralised	4-CS	BB84	BM, SM
Li et al. [45]	Blockchain	No	TF-QKD	SM
Ours	Centralised	No	No	SM

Abbreviations: 3-ES three-qubit entangled state; 4-CS, four qubit cluster state; 4-ES, four qubit entangled state; 6-ES, six qubit entangled state; BM, Bell measurement; BS, Bell State; GHZM, GHZ measurement; SM, single qubit measurement; ThM, three-qubit measurement; TM, two-qubit measurement; VNM, Von-Neumann measurement.

for further improvements that should address these limitations efficiently. This protocol can also find applications in many other quantum blind signatures or any e-commerce application such as e-voting etc.

AUTHOR CONTRIBUTIONS

Aman Gupta: Conceptualisation; methodology; writing – original draft; writing – review & editing. **Gunja Venkat Chandra**: Data curation; methodology; resources; visualisation; writing – original draft. **Nayana Das**: Formal analysis; investigation; methodology; supervision; writing – review & editing. **Goutam Paul**: Supervision; validation.

ACKNOWLEDGEMENTS

All four authors would like to express their sincere gratitude to Kaushik Nandi and Nixon Patel for their insightful advice and guidance in the early stages of this project. Additionally, the first three authors would like to thank their friends Heamanth, Rikteem Bhowmick, Devesh, Keshav, Nilesh, and Omkar for their technical comments.

CONFLICT OF INTEREST STATEMENT

The authors declares no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable—no new data generated, or the article describes entirely theoretical research.

ORCID

Aman Gupta https://orcid.org/0000-0003-2112-1074

REFERENCES

1. Werner, H.: Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. In: Original Scientific Papers Wissenschaftliche Originalarbeiten, pp. 478–504. Springer (1985)
2. Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature 299.5886(5886), 802–803 (1982). https://doi.org/10.1038/299802a0
3. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: arXiv Preprint arXiv:2003 (2020).06557
4. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Phys. Rev. Lett. 67(6), 661–663 (1991). https://doi.org/10.1103/physrevlett.67.661
5. Bennett, C.H.: Quantum cryptography using any two nonorthogonal states. Phys. Rev. Lett. 68.21(21), 3121–3124 (1992). https://doi.org/10.1103/physrevlett.68.3121
6. Lo, H.-K., Curty, M., Qi, B.: Measurement-device-independent quantum key distribution. Phys. Rev. Lett. 108.13(13), 130503 (2012). https://doi.org/10.1103/physrevlett.108.130503
7. Zhong, X., et al.: Proof-of-Principle experimental demonstration of twin-field type quantum key distribution. Phys. Rev. Lett. 123(10 Sept), 100506 (2019). https://doi.org/10.1103/PhysRevLett.123.100506
8. Deng, F.G, Long, G.L, Liu, X.S: Two-step quantum direct communication protocol using the Einstein-Podolsky-Rosen pair block. Phys. Rev. 68(4), 042317 (2003). https://doi.org/10.1103/PhysRevA.68.042317
9. Das, N., Paul, G.: Improving the security of "measurement-device-independent quantum communication without encryption". Sci. Bull. 65.24(24), 2048–2049 (2020). https://doi.org/10.1016/j.scib.2020.09.015
10. Das, N and Paul, G: Cryptanalysis of quantum secure direct communication protocol with mutual authentication based on single photons and Bell states. EPL Europhys. Lett. 138(4), 48001 (2021). https://doi.org/10.1209/0295-5075/ac2246
11. Das, N., Paul, G., Majumdar, R.: Quantum secure direct communication with mutual authentication using a single basis. Int. J. Theor. Phys. 60(11-12), 4044–4065 (2021). https://doi.org/10.1007/s10773-021-04952-4
12. Werner, H.: Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. In: Original Scientific Papers Wissenschaftliche Originalarbeiten, pp. 478–504. Springer (1985)
13. Das, N., Paul, G.: Device-Independent quantum secure direct communication with user authentication. arXiv preprint arXiv:2304, 03201 (2023)
14. Childs, A.M.: Secure assisted quantum computation. Quant. Inf. Comput. 5(6), 456–466 (2005). https://doi.org/10.26421/qic5.6-4
15. Arrighi, P., Salvail, L.: Blind quantum computation. Int. J. Quant. Inf. 4.05(05), 883–898 (2006). https://doi.org/10.1142/s0219749906002171
16. Tomoyuki, M.: Continuous-variable blind quantum computation. Phys. Rev. Lett. 109.23, 230502 (2012)
17. Morimae, T., Fujii, K.: Blind quantum computation protocol in which Alice only makes measurements. Phys. Rev. 87(5), 050301 (2013). https://doi.org/10.1103/physreva.87.050301

18. Fisher, K.A.G., et al.: Quantum computing on encrypted data. Nat. Commun. 5(1), 3074 (2014). https://doi.org/10.1038/ncomms4074
19. Zhang, X.: Gate teleportation-based universal blind quantum computation. In: arXiv Preprint arXiv:1809 (2018).00185
20. Peter, W.S.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. 41.2, 303–332 (1999)
21. Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology: Proceedings of Crypto 82, pp. 199–203. Springer (1983)
22. Lou, X., Chen, Z., Guo, Y.: A weak quantum blind signature with entanglement permutation. Int. J. Theor. Phys. 54(9), 3283–3292 (2015). https://doi.org/10.1007/s10773-015-2568-4
23. Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature 299.5886(5886), 802–803 (1982). https://doi.org/10.1038/299802a0
24. Carcia, J.C.P., Benslimane, A., Boutalbi, S.: Blockchain-based system for e-voting using blind signature protocol. In: 2021 IEEE Global Communications Conference (GLOBECOM), pp. 01–06. IEEE (2021)
25. Alshammari, H., et al.: Group signature entanglement in e-voting system. In: IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014, pp. 1–4. IEEE (2014)
26. Lin, T.-S., et al.: Quantum blind signature based on quantum circuit. In: 14th IEEE International Conference on Nanotechnology, pp. 868–872. IEEE (2014)
27. Wang, M.M., Chen, X.Bo, Yang, Y.X.: A blind quantum signature protocol using the GHZ states. Sci. China Phys. Mech. Astron. 56(9), 1636–1641 (2013). https://doi.org/10.1007/s11433-013-5170-x
28. Guo, W., et al.: Multi-proxy strong blind quantum signature scheme. Int. J. Theor. Phys. 55(8), 3524–3536 (2016). https://doi.org/10.1007/s10773-016-2979-x
29. Wen, X., et al.: A weak blind signature scheme based on quantum cryptography. Opt Commun. 282(4), 666–669 (2009). https://doi.org/10.1016/j.optcom.2008.10.025
30. Li, X.-Y., et al.: Quantum blind signature scheme based on quantum walk. Int. J. Theor. Phys. 59(7), 2059–2073 (2020). https://doi.org/10.1007/s10773-020-04478-1
31. Li, W., Shi, R., Guo, Y.: Blind quantum signature with blind quantum computation. Int. J. Theor. Phys. 56(4), 1108–1115 (2017). https://doi.org/10.1007/s10773-016-3252-z
32. Rodríguez Ramos, A.: Bitcoin and Other Cryptocurrencies as Payment Methods (2018)
33. Bhatia, V., Ramkumar, K.R.: An Efficient Quantum Computing Technique for Cracking RSA Using Shor's Algorithm (2020)
34. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: arXiv Preprint arXiv:2003 (2020).06557
35. Zhang, J.-Z., Yang, Y.-Y., Xie, S.-C.: A third-party e-payment protocol based on quantum group blind signature. Int. J. Theor. Phys. 56(9), 2981–2989 (2017). https://doi.org/10.1007/s10773-017-3464-x
36. Wen, X., Chen, Y., Fang, J.: An inter-bank E-payment protocol based on quantum proxy blind signature. Quant. Inf. Process. 12(1), 549–558 (2013). https://doi.org/10.1007/s11128-012-0398-3
37. Guo, Xi, Zhang, J.-Z., Xie, S.-C.: A trusted third-party e-payment protocol based on quantum blind signature without entanglement. Int. J. Theor. Phys. 57(9), 2657–2664 (2018). https://doi.org/10.1007/s10773-018-3787-2
38. Tiliwalidi, K., Zhang, J.-Z., Xie, S.-C.: A multi-bank E-payment protocol based on quantum proxy blind signature. Int. J. Theor. Phys. 58(10), 3510–3520 (2019). https://doi.org/10.1007/s10773-019-04217-1
39. Chen, F.-L., Wang, Z.-H., Hu, Y.-Mo: A new quantum blind signature scheme with BB84-state. Entropy 21.4(4), 336 (2019). https://doi.org/10.3390/e21040336
40. Niu, Xu-F., et al.: A third-party E-payment protocol based on quantum multi-proxy blind signature. Int. J. Theor. Phys. 57(8), 2563–2573 (2018). https://doi.org/10.1007/s10773-018-3778-3
41. Xie, S.-C., Niu, Xu-F., Zhang, J.-Z.: An improved quantum E-payment system. Int. J. Theor. Phys. 59(2), 445–453 (2020). https://doi.org/10.1007/s10773-019-04338-7
42. Zhang, J.-L., et al.: A novel E-payment protocol implented by blockchain and quantum signature. Int. J. Theor. Phys. 58(4), 1315–1325 (2019). https://doi.org/10.1007/s10773-019-04024-8
43. Xiang-lin, G., et al.: A novel quantum E-payment protocol based on blockchain. Quant. Inf. Process. 20.5, 192 (2021)
44. Li, En, et al.: Measurement-device-independent quantum protocol for E-payment based on blockchain. Quant. Inf. Process. 22.1(1), 40 (2023). https://doi.org/10.1007/s11128-022-03770-9
45. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Phys. Rev. Lett. 67(6), 661–663 (1991). https://doi.org/10.1103/physrevlett.67.661
46. Alagic, G., Gagliardoni, T., Majenz, C.: Can you sign a quantum state? Quantum 5, 603 (2021). https://doi.org/10.22331/q-2021-12-16-603
47. Hosoyamada, A.: Security of hash functions against attacks using quantum computers. NTT Tech. Rev. 21.7(7), 43–47 (2023). https://doi.org/10.53829/ntr202307fa4
48. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge university press (2010)
49. Wen, X., Nie, Z.: An E-payment system based on quantum blind and group signature. In: Data, Privacy, and E-Commerce, International Symposium on, pp. 50–55. IEEE Computer Society (2010)
50. Cai, X.-Q., Wei, C.-Y.: Cryptanalysis of an inter-bank E-payment protocol based on quantum proxy blind signature. Quant. Inf. Process. 12(4), 1651–1657 (2013). https://doi.org/10.1007/s11128-012-0477-5
51. Shao, Qi-F., et al.: Blockchain: architecture and research progress. https://api.semanticscholar.org/CorpusID:208635655 (2017)
52. Pirandola, S., et al.: Fundamental limits of repeaterless quantum communications. Nat. Commun. 8(1), 1–15 (2017). https://doi.org/10.1038/ncomms15043

How to cite this article: Gupta, A., et al.: An efficient and secure quantum blind signature-based electronic cash transaction scheme. IET Quant. Comm. 1–13 (2024). https://doi.org/10.1049/qtc2.12109

APPENDIX A

A.1 | CSWAP test

The cswap (controlled swap) test is a quantum computing method that determines the extent of overlap between two quantum states. As shown in Figure A1, it involves two input states, $|\phi\rangle$ and $|\psi\rangle$ and outputs a Bernoulli random variable that is 1 with a probability of $\frac{1}{2}\left(1-|\langle\psi|\phi\rangle|^2\right)$.

- At $t=0: |0\rangle|\phi\rangle|\psi\rangle$
- At $t=1: \frac{1}{\sqrt{2}}(|0,\phi,\psi\rangle + |1,\phi,\psi\rangle)$
- At $t=2: \frac{1}{\sqrt{2}}(|0,\phi,\psi\rangle + |1,\psi,\phi\rangle)$
- At $t=3: \frac{1}{2}[|0\rangle(|\phi,\psi\rangle + |\psi,\phi\rangle) + |1\rangle(|\phi,\psi\rangle - |\psi,\phi\rangle)]$
- After measurement

◦ Case 1: Measurement result = 0,

$$\begin{aligned} P(0) &= \frac{1}{4}[\langle\phi,\psi|\phi,\psi\rangle + \langle\psi,\phi|\phi,\psi\rangle + \langle\phi,\psi|\psi,\phi\rangle \\ &\quad + \langle\phi,\psi|\phi,\psi\rangle] \\ &= \frac{1}{2}\left[1+|\langle\psi|\phi\rangle|^2\right] \end{aligned}$$

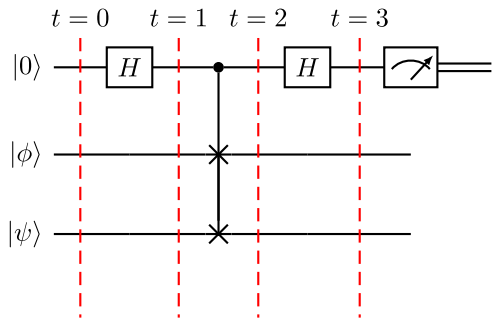


FIGURE A1 Quantum circuit to implement the cswap test between $|\psi\rangle$ and $|\phi\rangle$.

◦ Case 2: Measurement result = 1,

$$\begin{aligned} P(1) &= \frac{1}{4}[\langle\phi,\psi|\phi,\psi\rangle - \langle\psi,\phi|\phi,\psi\rangle + \langle\phi,\psi|\psi,\phi\rangle \\ &\quad - \langle\phi,\psi|\phi,\psi\rangle] \\ &= \frac{1}{2}\left[1 - |\langle\psi|\phi\rangle|^2\right] \end{aligned}$$

⇒ if $|\psi\rangle$ and $|\phi\rangle$ overlap then $P(0) = 1, P(1) = 0$ and if they are orthogonal $P(0) = P(1) = 1/2$.

A.2 | Hashing

Hashing is a process of converting an input string into a fixed-size output string called a hash, using a mathematical function. This mathematical function is called a hash function. It is used for data validation, security, and indexing.

A good hash function is

- Deterministic: Given the same input, the hash function should always produce the same output.
- Quick: The hash function should be computationally efficient.
- Uniform: A small change in the input should produce a significant change in the output.
- Collision-resistant: It should be computationally infeasible to find two different inputs that produce the same output hash value.

$$i \xrightarrow{f(i)} h$$

- i: input string
- $f(i)$: hash function
- h: output string or hash value.

A.3 | Phase Encoding

As mentioned in Def.1.7, phase encoding is a suitable candidate to convert the hash into a quantum state for the scheme proposed here. Phase encoding is a commonly used technique in quantum computing to encode classical information into qubits. This technique modifies the phase of a qubit by altering the relative phase of its two basis states to store classical information. By adjusting the relative phase, the state of the qubit becomes a linear superposition of the two basis states, each weighted by a complex coefficient. This change in phase can be used to represent classical information in the qubit, enabling classical information processing in a quantum computing system. Overall, phase encoding provides a way to represent classical data in a quantum form, facilitating the integration of quantum computing with classical information processing.

TABLE A1 Hex to phase angle mapping.

Hex	θ
0	0
1	$\pi/8$
2	$2\pi/8$
⋮	⋮
e	$14\pi/8$
f	$15\pi/8$

For example, consider the last 10 characters of a hash being

$$h = 8aeca37f3c$$

in base16. Then, we could divide the space of phase angles for a representative quantum state in the form $|+_\theta\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle + e^{i\theta}|1\rangle\right]$ into 16 parts as given in Table A1

Therefore, using the *Encode*() function gives a 10-qubit quantum state,

$$\begin{aligned} Encode(8aeca37f3c) = |&+_{\pi/8}+_{10\pi/8}+_{14\pi/8}+_{12\pi/8}+_{10\pi/8} \\ &+_{3\pi/8}+_{7\pi/8}+_{15\pi/8}+_{3\pi/8}+_{12\pi/8}\rangle = |\Psi\rangle. \end{aligned}$$

A.4 | An example Signcryption using quantum one-time pad

Taking the quantum state generated in the above example 8 and assuming the key generated using the *DenomKeyGen*() function defined in Def. 1.8 is

$$K = (K_x, K_z) = ((x_0, x_1, \ldots, x_{29}), (z_0, z_1, \ldots, z_{29})),$$

this means that $\alpha = 3$. Now the signcryption process is given by a unitary operator defined as follows:

$$\begin{aligned} \mathcal{U} = &\left(Z^{z_{20}} \otimes Z^{z_{21}} \otimes \cdots \otimes Z^{z_{29}}\right)\left(X^{x_{20}} \otimes X^{x_{21}} \otimes \cdots \otimes X^{x_{29}}\right) \\ &\left(Z^{z_{10}} \otimes Z^{z_{11}} \otimes \cdots \otimes Z^{z_{19}}\right)\left(X^{x_{10}} \otimes X^{x_{11}} \otimes \cdots \otimes X^{x_{19}}\right) \\ &\left(Z^{z_0} \otimes Z^{z_1} \otimes \cdots \otimes Z^{z_9}\right)\left(X^{x_0} \otimes X^{x_1} \otimes \cdots \otimes X^{x_9}\right) \end{aligned}$$

And the *SignCryption*() function gives

$$\mathcal{U}|\Psi\rangle = |S_\Psi\rangle.$$