The ISC Int'l Journal of Information Security
ISeCure

October 2025, Volume 17, Number 3 (pp. 1–13)
http://www.isecure-journal.org

Presented at the ISCISC'2025 in Tehran, Iran.

QuMixnet: A Quantum-Safe Mixnet Protocol **

Seyed Mohammad Dibaji [1], Taraneh Eghlidos [2,*], and Hossein Pilaram [2]
[1] *Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran.*
[2] *Electronics Research Institute, Sharif University of Technology, Tehran, Iran.*

ARTICLE INFO.

Keywords:
Mix networks, Post-quantum cryptography, Anonymous communication, Peer-to-peer networks

Type:

doi:

ABSTRACT

The emergence of quantum computing threatens the security of traditional cryptographic primitives underpinning anonymous communication protocols like mix networks (mixnets), necessitating quantum-resistant alternatives. This paper introduces QuMixnet, a mixnet protocol designed to withstand quantum attacks while ensuring robust anonymity and privacy. QuMixnet employs post-quantum cryptographic primitives, utilizing CRYSTALS-Dilithium for digital signatures to guarantee authenticity and CRYSTALS-Kyber for key encapsulation to secure message encryption with symmetric ciphers (e.g., AES-GCM). Operating on a peer-to-peer (P2P) architecture, every node can serve as a sender, receiver, or mix node, enhancing anonymity by obscuring participant roles. Sender-determined routing ensures that only the sender knows the full message path, with onion routing layered encryption across nodes. To counter traffic analysis, QuMixnet implements message padding to a fixed size, dummy messages for traffic covering, and batch processing with shuffling. A security model, evaluated through formal security games, confirms resilience of QuMixnet against adversaries with quantum capabilities, achieving strong sender and receiver anonymity, communication anonymity, confidentiality, and integrity. QuMixnet advances anonymous communication by offering a scalable, quantum-safe solution that fortifies privacy against evolving threats.

1 Introduction

Mix networks, or mixnets, are cryptographic protocols designed to provide anonymity and privacy for communication over the internet. First introduced by David Chaum in 1981 [1], mixnets route messages through proxy servers called mixes or mix nodes, which shuffle and relay messages to obscure the link between sender and recipient. This mechanism makes it challenging for adversaries to perform traffic analysis and trace communications, thereby protecting user privacy in applications such as electronic voting and anonymous messaging. Despite their potential, mixnets face significant challenges, including vulnerability to powerful adversaries capable of monitoring entire networks and the computational overhead associated with cryptographic operations, particularly in

* Corresponding author.
**The ISCISC'2025 program committee effort is highly acknowledged for reviewing this paper.
Email addresses: dibaji.smohammad@ee.sharif.edu, teghlidos@sharif.edu, pilaram@sharif.edu
ISSN: 2008-2045

real-time applications. Moreover, the advent of quantum computing poses a threat to traditional cryptographic primitives, such as RSA and Elliptic Curve Cryptography (ECC), which are susceptible to quantum attacks using algorithms like Shor's [2]. This necessitates the development of quantum-resistant cryptographic solutions to ensure long-term security.

Several mixnet protocols have been developed over the years, each addressing specific aspects of anonymity and security. Notable examples include Mixmaster [3], a widely used anonymous remailer system, and Mixminion [4], which introduced enhanced security features and flexibility. The Invisible Internet Project (I2P) [5], launched in 2003, is a P2P anonymity network that employs a mixnet architecture for secure communication. The Loopix anonymity system [6] has been proposed, offering a decentralized mixnet infrastructure with robust privacy guarantees. Building on Loopix, the Nym mixnet [7], launched in 2022, offers scalable privacy solutions with an incentive-driven model, as detailed in research on reward-sharing schemes [8]. Other post-2020 developments include HOPR [9], which provides an incentivized P2P mixnet for metadata privacy, and the Zero Knowledge Network (0KN) [10], which leverages the Trellis protocol [11] for anonymous broadcast communication. Notably, Katzenpost [12] and Elixxir's cMix (now part of xx.network) [13] incorporate post-quantum cryptographic techniques to ensure resilience against quantum adversaries. Katzenpost employs hybrid post-quantum cryptography, such as Kyber768, in its link layer [12], while xx.network uses quantum-resistant cryptography during its precomputation phase [13]. Despite these advancements, many protocols rely on classical cryptography or are in the process of transitioning to quantum-safe designs, as seen in Nym's 2025 roadmap to adopt the Outfox format [14].

The authors' contributions

In this paper, we present QuMixnet, a mixnet protocol designed to address these challenges. QuMixnet incorporates post-quantum cryptography to ensure long-term security against quantum computing threats. Specifically, it employs CRYSTALS-Dilithium [15] for digital signatures and CRYSTALS-Kyber [16] for key encapsulation, both of which are lattice-based cryptographic schemes standardized by the National Institute of Standards and Technology (NIST) as quantum-resistant algorithms [17]. Additionally, QuMixnet utilizes a P2P architecture, where every node can function as a sender, receiver, or mix node, enhancing scalability and resilience. The protocol also implements traffic obfuscation techniques, including message padding, dummy message injection, and batch processing, to thwart traffic analysis.

Compared to well-known mixnets, QuMixnet offers distinct advantages. Unlike Loopix and Nym, which rely on classical cryptography such as the Sphinx packet format [18], QuMixnet provides quantum-resistant security, making it future-proof against quantum adversaries. Its P2P architecture offers superior scalability and flexibility compared to non-P2P methods, where static roles and rigid structures often limit adaptability and expose vulnerabilities to targeted attacks. The decentralized design also strengthens anonymity by eliminating single points of failure and making traffic patterns harder to analyze, as the dynamic participation of nodes obscures communication flows. In contrast, non-P2P systems can be more easily mapped and exploited by adversaries due to their predictable topologies. Additionally, QuMixnet employs sophisticated traffic obfuscation techniques that provide a significant edge over systems with less robust countermeasures.

The remainder of this paper is organized as follows: Section 2 provides an overview of the QuMixnet protocol, including its architecture, cryptographic primitives, key features, and a 5-Node example. Section 3 presents a detailed step-by-step description of the protocol's operation. Section 4 presents a security analysis, defining the adversary model and demonstrating the protocol's security properties through formal security games. Finally, Section 5 concludes the paper, highlighting the main contributions.

2 Related Work

QuMixnet distinguishes itself through its fully post-quantum cryptographic framework and P2P architecture, offering enhanced scalability and robust traffic obfuscation techniques that address evolving security threats. Table 1 provides a comparative analysis of QuMixnet against some well-known mixnets.

In the context of post-quantum mixnets tailored for electronic voting, recent research has focused on lattice-based constructions to ensure long-term security. Aranha *et al.* [19] propose a verifiable mixnet with distributed decryption, utilizing BGV ciphertexts and amortized zero-knowledge proofs to ensure shuffle correctness. Boyen *et al.* [20] introduce a practical decryption mixnet with external auditing, employing lattice-based primitives and a trip wire technique for accountability. Farzaliyev *et al.* [21] enhance lattice-based mixnets for e-voting by optimizing zero-knowledge proofs, achieving scalability up to 100,000 votes. These works prioritize verifiability and accountability in sequential architectures, contrasting with QuMixnet's broader application scope.

Table 2 compares QuMixnet with these lattice-

Table 1. Comparison of QuMixnet with Well-Known Mixnets

Protocol	Architecture	Security	Routing	Traffic Obfuscation	Quantum Resistance
Mixminion [4]	Free-route decentralized mixnet	Classical (enhanced remailer security)	Free-route with forward anonymity	Shuffling, delays, batching	No
Loopix [6]	Decentralized stratified topology with providers and mixes	Classical (Sphinx packet format, layered encryption)	Poisson mixing with brief delays	Cover/loop traffic, Poisson-process randomness, exponential delays, shuffling	No
HOPR [9]	Incentivized P2P mixnet	Not specified (focus on metadata privacy)	Multi-hop relay	Packet splitting, mixing, cover traffic	No
cMix [13]	Decentralized fixed cascade of mix nodes	Quantum-resistant in precomputation phase	Cascade with precomputation	Mixing via precomputed shared values, minimal real-time ops	Yes (xx.network)
Nym [7]	Decentralized layered (5 hops: entry/exit gateways + 3 mix layers)	Classical (Sphinx format), post-quantum planned (Outfox format)	Source-routed decryption	Uniform packet sizes, cover traffic, exponential delays, timing obfuscation	Planned (2025 roadmap)
Katzenpost [12]	Decentralized mixnet with providers and layered mixes	Hybrid post-quantum (Kyber variants, Sphinx format)	Onion-like with Sphinx routing	Delays, shuffling, batching	Yes (hybrid PQ)
QuMixnet	P2P (every node as sender/receiver/mix)	Post-quantum (Dilithium signatures, Kyber KEM, AES-GCM symmetric cipher)	Sender-determined onion routing	Fixed-size padding, dummy messages, batch processing with shuffling, timestamps	Yes (lattice-based)

Table 2. Comparison of QuMixnet with Lattice-Based Voting Mixnets

Protocol	Architecture	Security	Routing	Traffic Obfuscation	Quantum Resistance
Boyen 2020 [20]	Sequential mix servers with external auditors	Lattice-based (IND-CCA2 PKE from LWE, digital signatures)	Iterative decryption and shuffling by servers	Shuffling with trip wires for verifiability (no explicit obfuscation beyond mixing)	Yes (lattice-based)
Farzaliyev 2023 [21]	Sequential mix nodes	Lattice-based (Ring-LWE, Module-SIS/LWE, amortized zero-knowledge proofs for shuffle)	Re-encryption shuffling by mix nodes	Shuffling with zero-knowledge proofs (no explicit padding or dummies)	Yes (lattice-based)
Aranha 2023 [19]	Sequential mix nodes with distributed decryption	Lattice-based (BGV ciphertexts, BDLOP commitments, amortized zero-knowledge proofs)	Sequential shuffling by mix nodes	Shuffling for permutation hiding (no explicit padding or dummies)	Yes (lattice-based)
QuMixnet	P2P (every node as sender/receiver/mix)	Post-quantum (Dilithium signatures, Kyber KEM, AES-GCM symmetric cipher)	Sender-determined onion routing	Fixed-size padding, dummy messages, batch processing with shuffling, timestamps	Yes (lattice-based)

based voting mixnets. While QuMixnet leverages its P2P design and extensive traffic obfuscation for general anonymous communication, the compared protocols focus on e-voting-specific requirements, emphasizing verifiable shuffles and decryption.

3 Overview of the Proposed Protocol

3.1 Objective

The ultimate goal of this protocol is to securely transmit a secret from a sender to a receiver with a high degree of confidentiality, integrity, and anonymity. Specifically, the protocol aims to provide relationship anonymity, ensuring that while the sender and receiver are aware of each other's identities, their communication relationship remains hidden from intermediate nodes and external observers. This is achieved through a combination of onion routing [22], traffic obfuscation, and the P2P architecture. The design leverages a mixnet architecture, combined with advanced cryptographic techniques and traffic obfuscation methods, to ensure that even a powerful adversary, monitoring the network, cannot reliably associate a specific message with its true sender or receiver, nor determine the communication patterns between nodes. Key enhancements, such as padding messages to a fixed size and injecting dummy messages, further obscure traffic patterns, while timestamps prevent replay attacks, collectively strengthening the security guarantees of the protocol.

3.2 Key Components and Cryptographic Primitives

3.2.1 CRYSTALS-Dilithium for Digital Signatures

The sender (Alice) uses Dilithium to sign the message, which includes not only the secret, but also the identities of both the sender and the receiver. This signature ensures the authenticity and integrity of the transmitted data, binding the secret to both identities and preventing forgery.

3.2.2 CRYSTALS-Kyber for Key Encapsulation

At the core of this protocol is CRYSTALS-Kyber, which serves as the foundation for the encryption

scheme. Instead of directly employing public-key encryption, the protocol leverages Kyber's key encapsulation mechanism (KEM) to encapsulate symmetric keys, which are then used with state-of-the-art symmetric ciphers, such as AES-GCM (Advanced Encryption Standard with the Galois/Counter Mode) [23], to encrypt the payload. This cryptographic approach ensures both post-quantum security and high performance, combining the robustness of Kyber against quantum attacks with the efficiency of symmetric encryption for larger data.

3.2.3 Onion Routing

Once the message is signed and encrypted for the receiver, it is further wrapped in multiple layers of encryption, one for each node, in the selected route. Each layer contains the address of the next node and is encrypted with the public key of that node. This layered approach means that every mix node can only decrypt its specific layer, revealing just enough information to forward the message without uncovering its entire content.

3.3 P2P Mixnet and Route Selection

3.3.1 P2P Mixnet Architecture

In this protocol, the network is structured as a P2P mixnet. Each node within the network is capable of simultaneously functioning as a sender, receiver, and mix node. This design is essential because it ensures that the node through which a message enters or exits is not necessarily the original sender or the ultimate recipient. Such a configuration creates an environment where message flows are indistinguishable from one another, greatly complicating efforts by an adversary to correlate traffic patterns. The P2P architecture therefore plays a crucial role in enhancing anonymity. Because any node may originate, terminate, or merely relay a message, an adversary cannot reliably determine whether traffic entering or leaving a node began or ended there. That ambiguity significantly hinders traffic analysis efforts.

3.3.2 Sender-Determined Routing

An important feature of the protocol is that the sender selects the entire route that the message will traverse through the network. By choosing the complete sequence of mix nodes in advance, only the sender has knowledge of the final recipient's address. This contrasts with other designs where each mix node might determine the next hop. In those cases, intermediate nodes would eventually need to know the final recipient's identity, potentially aiding an adversary if any of those nodes are compromised. By keeping the recipient's address hidden from all mix nodes except for the last one, the protocol minimizes the risk of targeted surveillance or collusion among mix nodes. It should be noted that the last mix node knows the address of the final recipient but does not know that it is the final recipient due to padding and encryption. The sender selects the route based on trust and reputation, leveraging a Distributed Hash Table (DHT) [24] and real-time updates via gossip protocols [25] to choose reliable and diverse nodes, further enhancing security against collusion and targeted attacks.

3.3.3 Enhanced Security through Layered Anonymity

Although the receiver and sender are aware of each other's identities in end-to-end communication, the use of a P2P mixnet introduces multiple layers of anonymity during the message's transit. This implies that even if an eavesdropper monitors the network's entry or exit points, it remains unlikely they can associate a message with its actual sender or receiver, as the message is obscured by being mixed with traffic from other nodes. The additional encryption layers not only protect the message content, but also obscure the communication endpoints, providing robust defense against traffic analysis.

3.4 Security and Practical Considerations

3.4.1 Confidentiality and Integrity

By combining the capabilities of Kyber with Dilithium, the protocol provides end-to-end security. The receiver can verify the authenticity of the message and be confident that the secret has not been tampered with, while any intercepted message remains indecipherable without the appropriate decryption keys.

3.4.2 Resistance to Collusion and Traffic Analysis

The sender-determined routing reduces the risk of collusion among mix nodes because no single node or group of nodes can piece together the full communication path, unless they control the entire route. Additionally, the P2P structure makes it challenging for an adversary to perform effective traffic analysis, even when monitoring final links. To further resist traffic analysis, the protocol incorporates a sophisticated padding mechanism. All messages transmitted over the network are padded to a fixed size denoted by MSG_SIZE, preventing adversaries, specifically external observers from correlating messages based on their lengths. Additionally, a specific padding strategy is applied to the message payload intended for

Bob to ensure that the last mix node cannot deduce whether it is forwarding the message to another mix node or to the final recipient, thereby protecting the recipient's identity. This strategy adjusts padding based on the secret size relative to a typical mix node layer denoted by L_{mix} which is explained in more detail in Section 4.1.2.

Furthermore, timestamps are included in both the message content and the padding layers to prevent replay attacks, ensuring that each message is fresh and cannot be reused by an adversary. Moreover, the protocol utilizes dummy messages, encrypted and padded similarly to real messages, to create additional cover traffic. These dummies are injected by mix nodes and sent to random nodes, making it harder for an adversary to distinguish real communication patterns from noise. Mix nodes process messages in batches by collecting them over a defined time interval and shuffling them before forwarding. This batching and shuffling disrupt temporal correlations, thereby hindering an adversary's ability to associate incoming and outgoing messages based on their sequence or timing characteristics.

3.4.3 Scalability and Resource Management

Although the proposed design introduces increased computational and bandwidth overhead, it remains an attractive solution due to the security and privacy benefits, since nodes handle both their own traffic and relayed messages. Effective load balancing and efficient cryptographic implementations are essential to maintain network performance, especially under heavy traffic conditions.

3.4.4 Robustness Against Powerful Adversaries

The protocol is designed to withstand attacks from powerful adversaries, including those with global visibility of the network and control over a subset of nodes. Through the combination of cryptographic protections, traffic obfuscation, and careful route selection, the protocol ensures that even such adversaries cannot reliably trace messages or identify communicating parties.

3.5 5-Node Setup Example

Consider a P2P mixnet with numerous nodes, among which we focus on five: Alice, MixNode 1 (M1), MixNode 2 (M2), MixNode 3 (M3), and Bob. Alice selects a route for her message: Alice → M1 → M2 → M3 → Bob, as shown in Figure 1. Figure 2 illustrates onion routing mechanism of the QuMixnet protocol. Alice's message to Bob is wrapped in multiple encryption layers: Orange for M1, green for M2, blue for M3, red for Bob, and a yellow padding layer for standardization of the message size to MSG_SIZE. As the message travels through M1, M2, and M3, each node peels off its outer layer and adds new padding, ensuring anonymity and security by fixing the message size through the network. When it reaches Bob, only the core message remains, which is then decrypted by Bob.

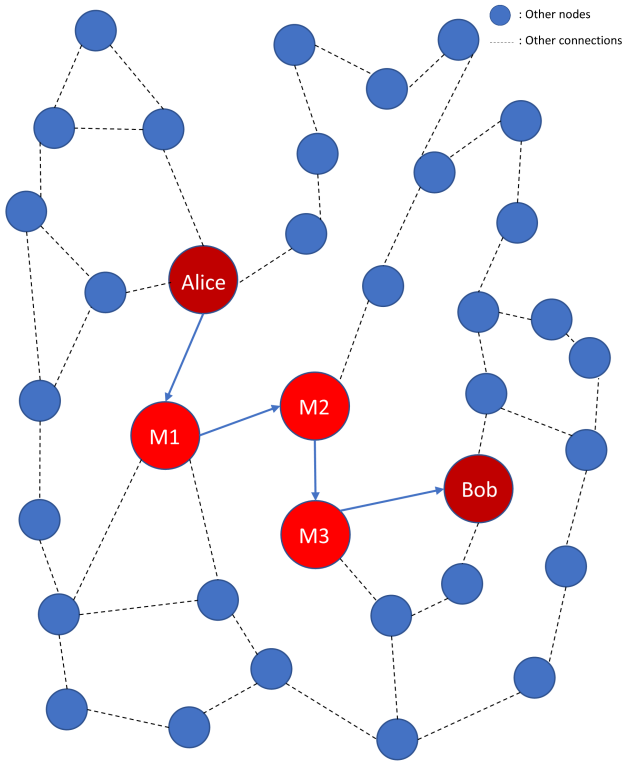

Figure 1. 5-Node Setup Example

4 Protocol Workflow

This section describes the QuMixnet protocol, detailing the steps for Alice to send a secret to Bob over a P2P mixnet with post-quantum security. The protocol uses CRYSTALS-Dilithium for signatures, CRYSTALS-Kyber for key encapsulation, onion routing, message padding, dummy traffic, and batch processing to ensure anonymity, confidentiality, and resistance to traffic analysis.

4.1 Alice's Operations

4.1.1 Message Preparation and Signing

Alice initiates the process by preparing a message that contains the secret she wants to send to Bob. This message includes the secret itself, her unique identifier such as hash of a public key or network address, Bob's unique identifier, and a timestamp to ensure the message's freshness and prevent replay attacks.

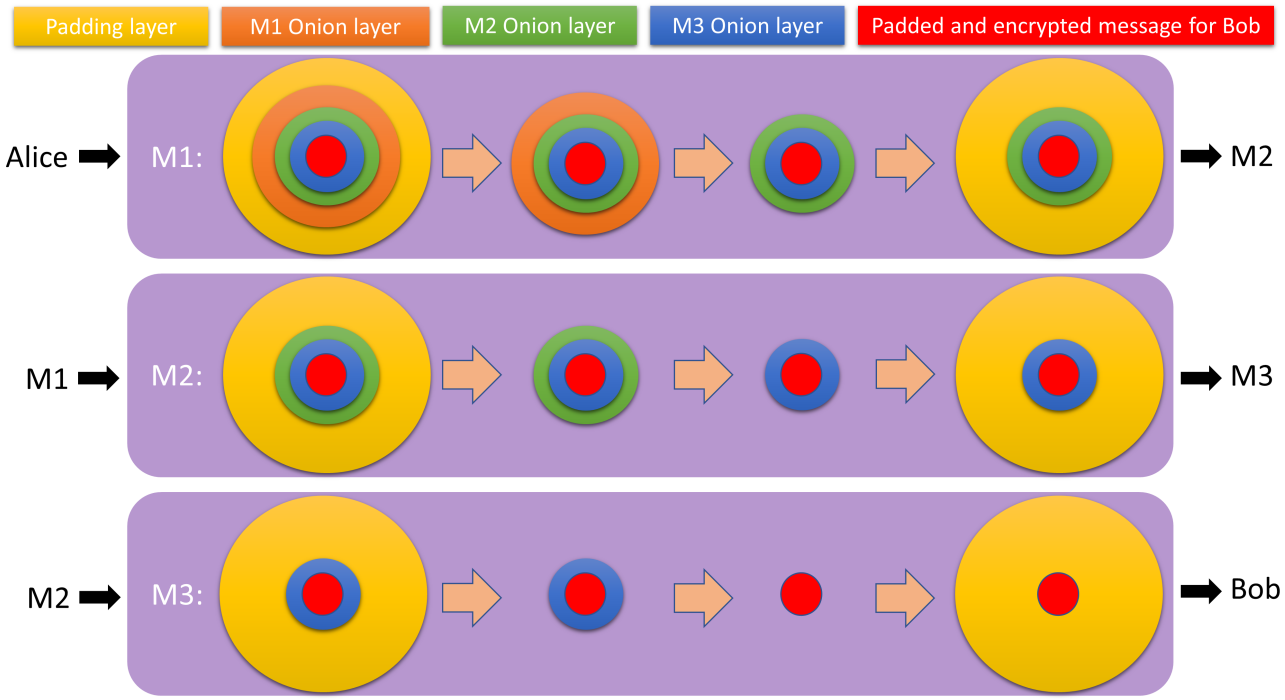

Figure 2. 5-Node onion routing mechanism

To guarantee that the message comes from her and has not been tampered with, Alice signs it using her CRYSTALS-Dilithium private key, producing a signature that Bob can later verify with her corresponding public key. The process is detailed in Pseudocode 1.

Pseudocode 1 Message Preparation and Signing

1: **function** PREPAREMESSAGE&SIGN(secret, ID_A, ID_B, $\mathrm{SK}_{\mathrm{Dil},A}$)
2: $T \leftarrow \mathrm{CurrentTime}()$
3: $M \leftarrow (\mathrm{secret}, \mathrm{ID}_A, \mathrm{ID}_B, T)$
4: $\sigma_A \leftarrow \mathrm{Sign}_{\mathrm{Dil}}(\mathrm{SK}_{\mathrm{Dil},A}, M)$
5: $M' \leftarrow (M, \sigma_A)$
6: **return** M'
7: **end function**

4.1.2 Encryption Process for Bob

The combination of the message and its signature forms the initial payload. To protect this payload from eavesdroppers, Alice encrypts it for Bob. She uses Bob's CRYSTALS-Kyber public key to generate a symmetric key and an encapsulation ciphertext through the Kyber encapsulation process. Before encrypting the payload $M' = (M, \sigma_A)$ for Bob, Alice applies a padding strategy to prevent adversaries from deducing information based on message length. The padding depends on the size of the secret within M. If the secret size is smaller than or equal to L_{mix}, where L_{mix} is the sum of the sizes of the nonce, encapsulation ciphertext, address, and symmetric encryption tag for a typical mix node layer, Alice randomly selects a size for padded M' within the range of L_{mix} and the maximum allowed by the fixed message size MSG_SIZE. If the secret size exceeds L_{mix}, the size for padded M' is randomly chosen between 0 and the maximum possible value constrained by MSG_SIZE. This approach prevents the last mix node from deducing whether it forwards to another mix node or the final recipient based on size, thus safeguarding the communication path's confidentiality. She then encrypts the padded payload with a symmetric cipher, such as AES-GCM, using the symmetric key and a random nonce, producing a ciphertext that only Bob can decrypt with his Kyber private key. Pseudocode 2 outlines the encryption procedure intended for Bob.

4.1.3 Onion Routing

Alice prepares the message for routing through the mixnet using onion encryption. She selects a sequence of mix nodes based on factors like trust and network topology. Starting with the last mix node, Alice encrypts the ciphertext intended for Bob, along with Bob's address, using the last mix node's Kyber public key to generate a symmetric key and encapsulation ciphertext. She then works backward, encrypting each subsequent layer with the public key of the preceding mix node, embedding the address of the next node in each layer. This creates a nested, onion-like structure where each mix node can only decrypt its own layer to reveal the next hop, ensuring that no single node

Pseudocode 2 Encryption for Bob

function EncryptForBob(M', $\text{PK}_{\text{Kyber},B}$, L_{mix}, MSG_SIZE)
  $\text{size}_{\text{secret}} \leftarrow \text{SizeOf}(\text{secret in } M')$
  if $\text{size}_{\text{secret}} \leq L_{\text{mix}}$ **then**
    $\text{pad}\_\text{size} \leftarrow \text{Random}(L_{\text{mix}} + \text{SizeOf}(M'), \text{max allowed by MSG\_SIZE})$
  else
    $\text{pad}\_\text{size} \leftarrow \text{Random}(\text{SizeOf}(M'), \text{max allowed by MSG\_SIZE})$
  end if
  padded $M' \leftarrow \text{Pad}(M', \text{pad}\_\text{size})$
  $(K_{\text{sym,msg},B}, C_{K,\text{msg},B}) \leftarrow \text{Kyber\_Encaps}(\text{PK}_{\text{Kyber},B})$
  $\text{nonce}_{\text{msg},B} \leftarrow \text{RandomNonce}()$
  $C_{\text{sym,msg},B} \leftarrow \text{SymEnc}^{\text{nonce}_{\text{msg},B}}_{K_{\text{sym,msg},B}}(\text{padded } M')$
  $C_B \leftarrow (C_{K,\text{msg},B}, \text{nonce}_{\text{msg},B}, C_{\text{sym,msg},B})$
  return C_B
end function

knows the full path from Alice to Bob.

To transmit the message through the network, Alice adds an additional padding layer for the first mix node. This layer includes a flag indicating that it is a real message as opposed to a dummy, a timestamp, and the onion-encrypted ciphertext. She pads this payload to a fixed size to maintain uniformity across all messages in the network, then encrypts it with the first mix node's Kyber public key using the same encryption approach. This padded and encrypted message is sent to the first mix node, marking the beginning of its journey through the network. The mechanism is algorithmically described in Pseudocode 3. Each mix node performs a similar padding and encryption operation, as detailed in Section 4.2.1.

Pseudocode 3 Onion Routing

function OnionRouting(C_B, Addr_B, $\{N_1, \ldots, N_n\}$, $\{\text{PK}_{\text{Kyber},N_i}\}$, PAYLOAD_SIZE, MSG_SIZE)
  $C_{N_n} \leftarrow \text{EncryptOnionLayer}(C_B, \text{Addr}_B, \text{PK}_{\text{Kyber},N_n})$
  for $i = n-1$ **downto** 1 **do**
    $C_{N_i} \leftarrow \text{EncryptOnionLayer}(C_{N_{i+1}}, \text{Addr}_{N_{i+1}}, \text{PK}_{\text{Kyber},N_i})$
  end for
  flag $\leftarrow 0$
  $T_{\text{pad},N_1} \leftarrow \text{CurrentTime}()$
  $\text{Payload} \leftarrow (\text{flag}, T_{\text{pad},N_1}, C_{N_1})$
  $\text{Payload} \leftarrow \text{Pad}(\text{Payload}, \text{PAYLOAD\_SIZE})$
  $(K_{\text{sym,pad},N_1}, C_{K,\text{pad},N_1}) \leftarrow \text{Kyber\_Encaps}(\text{PK}_{\text{Kyber},N_1})$
  $\text{nonce}_{\text{pad},N_1} \leftarrow \text{RandomNonce}()$
  $C_{\text{sym,pad},N_1} \leftarrow \text{SymEnc}^{\text{nonce}_{\text{pad},N_1}}_{K_{\text{sym,pad},N_1}}(\text{Payload})$
  $M_{\text{pad},N_1} \leftarrow (C_{K,\text{pad},N_1}, \text{nonce}_{\text{pad},N_1}, C_{\text{sym,pad},N_1})$
  if $\text{SizeOf}(M_{\text{pad},N_1}) = \text{MSG\_SIZE}$ **then**
    $\text{Send}(M_{\text{pad},N_1}, \text{Addr}_{N_1})$
  else
    error "incorrect size"
  end if
end function
function EncryptOnionLayer(X, $\text{Addr}_{\text{next}}$, $\text{PK}_{\text{Kyber},N_i}$)
  $(K_{\text{sym,onion},N_i}, C_{K,\text{onion},N_i}) \leftarrow \text{Kyber\_Encaps}(\text{PK}_{\text{Kyber},N_i})$
  $\text{nonce}_{\text{onion},N_i} \leftarrow \text{RandomNonce}()$
  $C_{\text{sym,onion},N_i} \leftarrow \text{SymEnc}^{\text{nonce}_{\text{onion},N_i}}_{K_{\text{sym,onion},N_i}}(X, \text{Addr}_{\text{next}})$
  return $(C_{K,\text{onion},N_i}, \text{nonce}_{\text{onion},N_i}, C_{\text{sym,onion},N_i})$
end function

4.2 Mix Nodes' Operations

4.2.1 Processing at Each Mix Node

Each mix node in the sequence processes incoming messages in batches to further obscure their origins and destinations. Upon receiving a message, a mix node decrypts the padding layer using its Kyber private key, checks the timestamp to discard expired or replayed messages, and determines whether it is a real message or a dummy based on the flag. For real messages, the node decrypts the onion layer to uncover the ciphertext for the next hop and its address. The node then shuffles the batch of messages it has collected over a time interval to disrupt any timing correlations, re-encrypts each real message with a new padding layer for the next hop, and forwards them accordingly. To maintain consistent traffic patterns and confuse adversaries, the node also generates and sends dummy messages to random nodes or selected nodes based on a specific strategy in the network after shuffling. Note that these dummy messages are padded and encrypted payloads with a dummy flag. Pseudocode 4 provides a detailed algorithm for the operations performed by each mix node N_i.

Note: Among the real messages being sent to the next hop, some could originate from the mix node itself. However, since Pseudocode 4 focuses on the role of the mix node, this aspect has not been included.

4.2.2 Final Delivery to Bob

The last mix node processes its batch similarly and forwards the message to Bob. The process is detailed in Pseudocode 5. Note that this mix node functions similarly to the other mix nodes. However, the focus here is on its role as the last mix node.

Pseudocode 4 Processing at Mix Node N_i

function ProcessMixNode($\{M_{\text{pad}}\}$, $\text{SK}_{\text{Kyber},N_i}$, MIN_BATCH, PAYLOAD_SIZE, MSG_SIZE)
  messages $\leftarrow$ CollectMessagesOverTime($\{M_{\text{pad}}\}$)
  real˙msgs $\leftarrow [\,]$
  for each $M_{\text{pad}} = (C_{K,\text{pad}}, \text{nonce}_{\text{pad}}, C_{\text{sym,pad}})$ in messages **do**
    $K_{\text{sym,pad},N_i} \leftarrow \text{Kyber\_Decaps}(C_{K,\text{pad}}, \text{SK}_{\text{Kyber},N_i})$
    $(\text{flag}, T_{\text{pad},N_i}, \text{data}) \leftarrow \text{SymDec}^{\text{nonce}_{\text{pad}}}_{K_{\text{sym,pad},N_i}}(C_{\text{sym,pad}})$
    if T_{pad,N_i} is expired **then**
      discard M_{pad}
    end if
    if flag $= 1$ **then**
      discard M_{pad}
    else
      $C_{N_i} \leftarrow$ data
      real˙msgs.append(C_{N_i})
    end if
  end for
  out˙msgs $\leftarrow [\,]$
  for each $C_{N_i} = (C_{K,\text{onion},N_i}, \text{nonce}_{\text{onion},N_i}, C_{\text{sym,onion},N_i})$ in real˙msgs **do**
    $K_{\text{sym,onion},N_i} \leftarrow \text{Kyber\_Decaps}(C_{K,\text{onion},N_i}, \text{SK}_{\text{Kyber},N_i})$
    $(X, \text{Addr}_{\text{next}}) \leftarrow \text{SymDec}^{\text{nonce}_{\text{onion},N_i}}_{K_{\text{sym,onion},N_i}}(C_{\text{sym,onion},N_i})$
    Payload $\leftarrow (0, \text{CurrentTime}(), X)$
    Payload $\leftarrow \text{Pad}(\text{Payload}, \text{PAYLOAD\_SIZE})$
    $(K_{\text{sym,pad},N_{i+1}}, C_{K,\text{pad},N_{i+1}}) \leftarrow \text{Kyber\_Encaps}(\text{PK}_{\text{Kyber,next}})$
    $\text{nonce}_{\text{pad},N_{i+1}} \leftarrow \text{RandomNonce}()$
    $C_{\text{sym,pad},N_{i+1}} \leftarrow \text{SymEnc}^{\text{nonce}_{\text{pad},N_{i+1}}}_{K_{\text{sym,pad},N_{i+1}}}(\text{Payload})$
    $M_{\text{pad},N_{i+1}} \leftarrow (C_{K,\text{pad},N_{i+1}}, \text{nonce}_{\text{pad},N_{i+1}}, C_{\text{sym,pad},N_{i+1}})$
    out˙msgs.append($(M_{\text{pad},N_{i+1}}, \text{Addr}_{\text{next}})$)
  end for
  $R \leftarrow |\text{real˙msgs}|$
  for $j = 1$ **to** $\max(0, \text{MIN\_BATCH} - R)$ **do**
    $N_j \leftarrow$ SelectRandomNode()
    Payload $\leftarrow (1, \text{CurrentTime}(), \text{RandomBytes}(\text{typical onion size}))$
    Payload $\leftarrow \text{Pad}(\text{Payload}, \text{PAYLOAD\_SIZE})$
    $(K_{\text{sym,pad},N_j}, C_{K,\text{pad},N_j}) \leftarrow \text{Kyber\_Encaps}(\text{PK}_{\text{Kyber},N_j})$
    $\text{nonce}_{\text{pad},N_j} \leftarrow \text{RandomNonce}()$
    $C_{\text{sym,pad},N_j} \leftarrow \text{SymEnc}^{\text{nonce}_{\text{pad},N_j}}_{K_{\text{sym,pad},N_j}}(\text{Payload})$
    $M_{\text{pad,dummy}} \leftarrow (C_{K,\text{pad},N_j}, \text{nonce}_{\text{pad},N_j}, C_{\text{sym,pad},N_j})$
    out˙msgs.append($(M_{\text{pad,dummy}}, \text{Addr}_{N_j})$)
  end for
  out˙msgs $\leftarrow$ Shuffle(out˙msgs)
  for each $(M_{\text{pad}}, \text{Addr})$ in out˙msgs **do**
    if SizeOf(M_{pad}) = MSG_SIZE **then**
      Send(M_{pad}, Addr)
    else
      error "incorrect size"
    end if
  end for
end function

Pseudocode 5 Final Delivery to Bob

function DeliverToBob(M_{pad,N_n}, $\text{SK}_{\text{Kyber},N_n}$, $\text{PK}_{\text{Kyber},B}$, PAYLOAD_SIZE, MSG_SIZE)
  $(C_{K,\text{pad}}, \text{nonce}_{\text{pad}}, C_{\text{sym,pad}}) \leftarrow M_{\text{pad},N_n}$
  $K_{\text{sym,pad},N_n} \leftarrow \text{Kyber\_Decaps}(C_{K,\text{pad}}, \text{SK}_{\text{Kyber},N_n})$
  $(\text{flag}, T_{\text{pad},N_n}, \text{data}) \leftarrow \text{SymDec}^{\text{nonce}_{\text{pad}}}_{K_{\text{sym,pad},N_n}}(C_{\text{sym,pad}})$
  if T_{pad,N_n} is expired **or** flag $= 1$ **then**
    discard M_{pad,N_n}
  end if
  $C_{N_n} \leftarrow$ data
  $(C_{K,\text{onion},N_n}, \text{nonce}_{\text{onion},N_n}, C_{\text{sym,onion},N_n}) \leftarrow C_{N_n}$
  $K_{\text{sym,onion},N_n} \leftarrow \text{Kyber\_Decaps}(C_{K,\text{onion},N_n}, \text{SK}_{\text{Kyber},N_n})$
  $(C_B, \text{Addr}_B) \leftarrow \text{SymDec}^{\text{nonce}_{\text{onion},N_n}}_{K_{\text{sym,onion},N_n}}(C_{\text{sym,onion},N_n})$
  Payload $\leftarrow (0, \text{CurrentTime}(), C_B)$
  Payload $\leftarrow \text{Pad}(\text{Payload}, \text{PAYLOAD\_SIZE})$
  $(K_{\text{sym,pad},B}, C_{K,\text{pad},B}) \leftarrow \text{Kyber\_Encaps}(\text{PK}_{\text{Kyber},B})$
  $\text{nonce}_{\text{pad},B} \leftarrow \text{RandomNonce}()$
  $C_{\text{sym,pad},B} \leftarrow \text{SymEnc}^{\text{nonce}_{\text{pad},B}}_{K_{\text{sym,pad},B}}(\text{Payload})$
  $M_{\text{pad},B} \leftarrow (C_{K,\text{pad},B}, \text{nonce}_{\text{pad},B}, C_{\text{sym,pad},B})$
  if SizeOf($M_{\text{pad},B}$) = MSG_SIZE **then**
    Send($M_{\text{pad},B}$, Addr_B)
  else
    error "incorrect size"
  end if
end function

4.3 Bob's Operations

4.3.1 Reception and Decryption by Bob

Bob receives the message, decrypts the padding layer with his Kyber private key, and checks the timestamp to ensure it is valid. He then decrypts the inner ciphertext using his Kyber private key to retrieve the symmetric key, which he uses to decrypt the padded payload and recover Alice's original message and signature. Bob verifies the signature with Alice's Dilithium public key, confirms the timestamp and identifiers match the expected values, and accepts the secret if all checks pass. Pseudocode 6 outlines the procedure Bob follows to decrypt and verify the received message.

Pseudocode 6 Reception and Decryption by Bob

function DECRYPTBYBOB($M_{\text{pad},B}$, $\text{SK}_{\text{Kyber},B}$, $\text{PK}_{\text{Dil},A}$, ID_B)
  $(C_{K,\text{pad},B}, \text{nonce}_{\text{pad},B}, C_{\text{sym,pad},B}) \leftarrow M_{\text{pad},B}$
  $K_{\text{sym,pad},B} \leftarrow \text{Kyber\_Decaps}(C_{K,\text{pad},B}, \text{SK}_{\text{Kyber},B})$
  $(\text{flag}, T_{\text{pad},B}, \text{data}) \leftarrow \text{SymDec}^{\text{nonce}_{\text{pad},B}}_{K_{\text{sym,pad},B}}(C_{\text{sym,pad},B})$
  if $T_{\text{pad},B}$ is expired **or** flag $= 1$ **then**
    discard $M_{\text{pad},B}$
  end if
  $C_B \leftarrow$ data
  $(C_{K,\text{msg},B}, \text{nonce}_{\text{msg},B}, C_{\text{sym,msg},B}) \leftarrow C_B$
  $K_{\text{sym,msg},B} \leftarrow \text{Kyber\_Decaps}(C_{K,\text{msg},B}, \text{SK}_{\text{Kyber},B})$
  padded $M' \leftarrow \text{SymDec}^{\text{nonce}_{\text{msg},B}}_{K_{\text{sym,msg},B}}(C_{\text{sym,msg},B})$
  $M' \leftarrow \text{RemovePadding}(\text{padded } M')$
  $(M, \sigma_A) \leftarrow M'$
  $(\text{secret}, \text{ID}_A, \text{ID}'_B, T) \leftarrow M$
  if T is expired **then**
    reject M
  end if
  if $\text{Verify}_{\text{Dil}}(\text{PK}_{\text{Dil},A}, M, \sigma_A) \neq \text{True}$ **then**
    reject M
  end if
  if $\text{ID}'_B \neq \text{ID}_B$ **then**
    reject M
  end if
  return $(\text{secret}, \text{ID}_A)$
end function

5 Security

In this section, we rigorously analyze the security properties of QuMixnet, establishing its resilience against quantum-capable adversaries through formal models and proofs. We begin by defining the adversary model, which assumes a powerful attacker with global network visibility, control over a fraction of nodes, and access to quantum computing resources. Subsequently, we introduce a series of security games that capture key anonymity and privacy guarantees, including sender and receiver anonymity, communication anonymity, confidentiality, and integrity. Our analysis demonstrates that QuMixnet achieves negligible adversary advantages under standard post-quantum cryptographic assumptions, leveraging the IND-CCA2 (Indistinguishability under Adaptive Chosen-Ciphertext Attack) security of CRYSTALS-Kyber and also the AEAD (Authenticated Encryption with Associated Data) security of symmetric cipher, plus the sEUF-CMA (Strongly Existential Unforgeability under Chosen Message Attack) security of CRYSTALS-Dilithium.

5.1 Adversary Model

The adversary $\mathcal{A}$ has:

- Full network visibility, observing all traffic.
- Control over a fraction $f < 1$ of $|N|$ nodes ($|C| = f \cdot |N|$).
- Quantum capabilities, bounded by the post-quantum security of Kyber, the symmetric cipher, and Dilithium.

The protocol uses onion routing, message padding (pad to `MSG_SIZE`), batching (`MIN_BATCH` $= B$), shuffling, and dummy messages. Security parameters are as follows:

- ϵ_{Kyber}: Advantage in breaking Kyber's IND-CCA2 security.
- ϵ_{sym}: Advantage in breaking the symmetric cipher's AEAD security.
- $\epsilon_{\text{Dilithium}}$: Advantage in forging a Dilithium signature.

All these advantages are negligible in the security parameter λ.

5.2 Security Games

We define five games, measuring the adversary's advantage relative to random guessing.

5.2.1 Game 1: Sender Anonymity

Goal: $\mathcal{A}$ identifies the sender of a message to a known receiver.

Setup:

(1) The challenger $\mathcal{C}$ initializes the P2P mixnet with $|N|$ nodes.
(2) $\mathcal{A}$ selects corrupted nodes $C \subset N$, $|C| = f \cdot |N|$.
(3) $\mathcal{C}$ picks two honest senders $S_0, S_1 \in N \setminus C$ and a receiver $R \in N \setminus C$, informing $\mathcal{A}$.
(4) $\mathcal{C}$ flips a bit $b \in \{0, 1\}$, and S_b sends a message to R via n mix nodes.

(5) $\mathcal{A}$ observes traffic and guesses $b' \in \{0,1\}$.

Adversary's Goal: Correctly guess b.

Advantage:

$$\text{Adv}_{\mathcal{A}}^{\text{sender-anon}} = \left|\Pr[b' = b] - \frac{1}{2}\right|$$

5.2.2 Game 2: Receiver Anonymity

Goal: $\mathcal{A}$ identifies the receiver of a message from a known sender.

Setup:

(1) $\mathcal{C}$ initializes the network.

(2) $\mathcal{A}$ selects $C \subset N$.

(3) $\mathcal{C}$ picks a sender $S \in N \setminus C$ and two honest receivers $R_0, R_1 \in N \setminus C$, informing $\mathcal{A}$.

(4) $\mathcal{C}$ flips a bit $b \in \{0,1\}$, and S sends a message to R_b via n mix nodes.

(5) $\mathcal{A}$ observes traffic and guesses $b' \in \{0,1\}$.

Adversary's Goal: Correctly guess b.

Advantage:

$$\text{Adv}_{\mathcal{A}}^{\text{receiver-anon}} = \left|\Pr[b' = b] - \frac{1}{2}\right|$$

5.2.3 Game 3: Communication Anonymity

Goal: $\mathcal{A}$ determines if a specific sender-receiver pair communicates.

Setup:

(1) $\mathcal{C}$ initializes the network.

(2) $\mathcal{A}$ selects $C \subset N$.

(3) $\mathcal{C}$ picks two honest pairs (S_0, R_0) and (S_1, R_1) in $N \setminus C$, informing $\mathcal{A}$.

(4) $\mathcal{C}$ flips a bit $b \in \{0,1\}$, and S_b sends a message to R_b via n mix nodes.

(5) $\mathcal{A}$ observes traffic and guesses $b' \in \{0,1\}$.

Adversary's Goal: Correctly guess b.

Advantage:

$$\text{Adv}_{\mathcal{A}}^{\text{com-anon}} = \left|\Pr[b' = b] - \frac{1}{2}\right|$$

5.2.4 Game 4: Confidentiality

Goal: $\mathcal{A}$ learns the content of a message.

Setup:

(1) $\mathcal{C}$ initializes the network.

(2) $\mathcal{A}$ selects $C \subset N$.

(3) $\mathcal{C}$ picks $S, R \in N \setminus C$.

(4) $\mathcal{A}$ submits two equal-length messages m_0, m_1 to $\mathcal{C}$.

(5) $\mathcal{C}$ flips a bit $b \in \{0,1\}$, and S sends m_b to R via n mix nodes.

(6) $\mathcal{A}$ observes traffic and guesses $b' \in \{0,1\}$.

Adversary's Goal: Correctly guess b.

Advantage:

$$\text{Adv}_{\mathcal{A}}^{\text{conf}} = \left|\Pr[b' = b] - \frac{1}{2}\right|$$

5.2.5 Game 5: Integrity

Goal: $\mathcal{A}$ tampers with a message undetected.

Setup:

(1) $\mathcal{C}$ initializes the network.

(2) $\mathcal{A}$ selects $C \subset N$.

(3) $\mathcal{C}$ picks $S, R \in N \setminus C$.

(4) S sends a message m to R via n mix nodes, signed with Dilithium.

(5) $\mathcal{A}$ may modify messages at corrupted nodes, producing m'.

(6) R outputs m' if the signature verifies, or $\perp$ if invalid.

Adversary's Goal: Produce $m' \neq m$ such that R accepts $m' \neq \perp$.

Advantage:

$$\text{Adv}_{\mathcal{A}}^{\text{int}} = \Pr[m' \neq m \wedge m' \neq \perp]$$

5.3 Security Analysis

We analyze QuMixnet's security assuming a sufficiently large network and honest node batching, we quantify adversary advantages in security games.

5.3.1 Assumptions

We make the following assumptions:

- Kyber is IND-CCA2 secure ($\epsilon_{\text{Kyber}} \leq 2^{-\lambda}$).

- The symmetric cipher is AEAD secure ($\epsilon_{\text{sym}} \leq 2^{-\lambda}$).
- Dilithium is sEUF-CMA secure ($\epsilon_{\text{Dilithium}} \leq 2^{-\lambda}$).
- $|N|$ is large, $f < 1$, n ensures $f^n \leq 2^{-\lambda}$.
- mix nodes process batches $\geq B$, with shuffling and dummies.

5.3.2 Game 1: Sender Anonymity

Analysis: $\mathcal{A}$ succeeds if they control all n mix nodes with probability $\Pr[E] = f^n$ or break Kyber or the symmetric cipher to infer the sender. Breaking either suffices for $\mathcal{A}$ to succeed, so the probability is $\frac{1}{2} + \epsilon_{\text{Kyber}} + \epsilon_{\text{sym}}$.

$$\Pr[b' = b] \leq f^n \cdot 1 + (1 - f^n) \cdot \left(\frac{1}{2} + \epsilon_{\text{Kyber}} + \epsilon_{\text{sym}}\right)$$

$$= \frac{1}{2} + \frac{1}{2} f^n + (1 - f^n)(\epsilon_{\text{Kyber}} + \epsilon_{\text{sym}})$$

$$\text{Adv}_{\mathcal{A}}^{\text{sender-anon}} \leq \frac{1}{2} f^n + (1 - f^n)(\epsilon_{\text{Kyber}} + \epsilon_{\text{sym}})$$

$$\text{Adv}_{\mathcal{A}}^{\text{sender-anon}} \leq \frac{1}{2} f^n + \epsilon_{\text{Kyber}} + \epsilon_{\text{sym}},$$

which is negligible since $f^n, \epsilon_{\text{Kyber}}, \epsilon_{\text{sym}} \leq 2^{-\lambda}$.

5.3.3 Game 2: Receiver Anonymity

Analysis: Similar to Game 1, $\mathcal{A}$ needs to control all n nodes or break Kyber or the symmetric cipher.

$$\text{Adv}_{\mathcal{A}}^{\text{receiver-anon}} \leq \frac{1}{2} f^n + \epsilon_{\text{Kyber}} + \epsilon_{\text{sym}},$$

which is negligible.

5.3.4 Game 3: Communication Anonymity

Analysis: $\mathcal{A}$ succeeds if all n nodes are compromised or Kyber or the symmetric cipher is broken.

$$\text{Adv}_{\mathcal{A}}^{\text{com-anon}} \leq \frac{1}{2} f^n + \epsilon_{\text{Kyber}} + \epsilon_{\text{sym}},$$

which is negligible.

5.3.5 Game 4: Confidentiality

Analysis: $\mathcal{A}$ must break Kyber or the symmetric cipher to distinguish m_0 from m_1.

$$\text{Adv}_{\mathcal{A}}^{\text{conf}} \leq \epsilon_{\text{Kyber}} + \epsilon_{\text{sym}},$$

which is negligible.

5.3.6 Game 5: Integrity

Analysis: $\mathcal{A}$ must forge a Dilithium signature.

$$\text{Adv}_{\mathcal{A}}^{\text{int}} \leq \epsilon_{\text{Dilithium}},$$

which is negligible.

6 Conclusion

QuMixnet represents a significant advancement in the field of anonymous communication protocols by addressing the vulnerabilities posed by quantum algorithms. Through the integration of post-quantum cryptographic primitives, specifically CRYSTALS-Dilithium for digital signatures and CRYSTALS-Kyber for key encapsulation, QuMixnet ensures long-term security and resilience against quantum-based attacks. Its P2P architecture, where every node can function as a sender, receiver, or mix node, not only enhances scalability and flexibility, but also strengthens anonymity by obscuring the roles of participants and thwarting traffic analysis. Sophisticated traffic obfuscation techniques of the protocol, including message padding, dummy messages, and batch processing, further strengthen its security against adversarial attempts to trace message routing or identify communication patterns. As quantum computing capabilities continue to evolve, protocols like QuMixnet are essential for safeguarding privacy and security in digital communications, ensuring that anonymous systems remain robust and reliable in the face of emerging threats.

References

[1] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[2] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. IEEE, 1994.

[3] Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster protocol-version 2. *Online specification*, 2003.

[4] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *2003 Symposium on Security and Privacy, 2003.*, pages 2–15. IEEE, 2003.

[5] Bassam Zantout, Ramzi Haraty, et al. I2p data communication system. In *Proceedings of ICN*, pages 401–409. Citeseer, 2011.

[6] Ania M Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. The loopix anonymity system. In *26th usenix secu-*

rity symposium (usenix security 17), pages 1199–1216, 2017.

[7] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. The nym network. 2021.

[8] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. Reward sharing for mixnets. 2022.

[9] HOPR Association. Hopr protocol overview. 2021. Accessed: 2025-07-07.

[10] 0 Knowledge Network. 0kn overview. 2024. Accessed: 2025-07-07.

[11] Simon Langowski, Sacha Servan-Schreiber, and Srinivas Devadas. Trellis: Robust and scalable metadata-private anonymous broadcast. *Cryptology ePrint Archive*, 2022.

[12] Ewa J Infeld, David Stainton, Leif Ryge, and Threebit Hacker. Echomix: a strong anonymity system with messaging. *arXiv preprint arXiv:2501.02933*, 2025.

[13] David Chaum, Debajyoti Das, Farid Javani, Aniket Kate, Anna Krasnova, Joeri De Ruiter, and Alan T Sherman. cmix: Mixing with minimal real-time asymmetric cryptographic operations. In *Applied Cryptography and Network Security: 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings 15*, pages 557–578. Springer, 2017.

[14] Alfredo Rial and Ania M Piotrowska. Outfox: a packet format for a layered mixnet. *arXiv preprint arXiv:2412.19937*, 2024.

[15] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018.

[16] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[17] National Institute of Standards and Technology. Post-quantum cryptography standardization. 2024. Accessed: 2025-07-07.

[18] George Danezis and Ian Goldberg. Sphinx: A compact and provably secure mix format. In *2009 30th IEEE Symposium on Security and Privacy*, pages 269–282. IEEE, 2009.

[19] Diego F Aranha, Carsten Baum, Kristian Gjøsteen, and Tjerand Silde. Verifiable mix-nets and distributed decryption for voting from lattice-based assumptions. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1467–1481, 2023.

[20] Xavier Boyen, Thomas Haines, and Johannes Müller. A verifiable and practical lattice-based decryption mix net with external auditing. In *European Symposium on Research in Computer Security*, pages 336–356. Springer, 2020.

[21] Valeh Farzaliyev, Jan Willemson, and Jaan Kristjan Kaasik. Improved lattice-based mix-nets for electronic voting. *IET Information Security*, 17(1):18–34, 2023.

[22] Michael G Reed, Paul F Syverson, and David M Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected areas in Communications*, 16(4):482–494, 2002.

[23] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac, 2007. *NIST Special Publication (SP)*.

[24] Hao Zhang, Yonggang Wen, Haiyong Xie, Nenghai Yu, et al. *Distributed hash table: Theory, platforms and applications*. Springer, 2013.

[25] Alberto Montresor et al. Gossip and epidemic protocols. *Wiley encyclopedia of electrical and electronics engineering*, 1, 2017.

Seyed Mohammad Dibaji received his B.Sc. degree in Electrical Engineering from Isfahan University of Technology and his M.Sc. degree in Electrical Engineering with a specialization in Secure Communications and Cryptography from Sharif University of Technology. His research interests include quantum communications, quantum cryptography, and post-quantum cryptography.

Taraneh Eghlidos received the B.Sc. degree in Mathematics from the University of Shahid Beheshti, Tehran, Iran, in 1986, and the M.Sc. degree in Industrial Mathematics from the University of Kaiserslautern, Germany, in 1991 and the Ph.D. degree in Mathematics from the University of Giessen, Germany, in 2000. She joined Sharif University of Technology (SUT) in 2002 as the faculty member, and is currently an Associate Professor with the Electronics Research Institute at SUT. Her research interests include interdisciplinary research areas, such as symmetric and asymmetric cryptography, applications of coding theory in cryptography, and mathematical modeling for solving real world problems. Her current fields of research include Lattice-based and Code-based Cryptography.

Hossein Pilaram received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2010, 2012, and 2017, respectively. He is currently an Assistant Professor with the Electronics Research Institute, Sharif University of Technology. His research interests include cryptography, with a focus on post-quantum cryptographic schemes, secret sharing, and wireless network security.