IPSeQ: A Security-Enhanced IPSec Protocol Integrated with Quantum Key Distribution

Xumin Gao, Kaiping Xue, Jian Li, Zhonghui Li, Jiaqi Wu, Nenghai Yu, Qibin Sun, and Jun Lu

Abstract

IPSec is a widely used network security protocol that plays a crucial role in providing secure transmission channels in the current Internet. However, the advent of quantum computing poses unprecedented challenges to the security of traditional cryptographic methods, including those used in IPSec. Fortunately, quantum key distribution (QKD) offers a theoretically unbreakable method for exchanging keys between two communicating parties. To address the security threats posed by quantum computing, we propose IPSeQ, a security-enhanced IPSec protocol that integrates QKD into its design. Specifically, IPSeQ leverages quantum keys to strengthen key negotiation, authentication, and data encryption processes. To achieve rapid key updates while ensuring transmission efficiency and key synchronization, IPSeQ introduces a sliding window-based dynamic key updating mechanism. Experiments conducted with real QKD devices demonstrate that our proposed mechanism can improve throughput by more than 50 percent compared to traditional schemes, particularly at higher quantum key generation rates. Additionally, IPSeQ effectively maintains robust data transmission in scenarios where quantum keys are scarce (e.g., when the key generation rate is less than 10 kb/s).

Introduction

Internet Protocol Security (IPSec) has long been a cornerstone in establishing secure communication channels over public Internet infrastructure, providing authentication, confidentiality, and data integrity at the network layer to support secure communication. However, the advent of the second quantum revolution has significantly accelerated the development of quantum computing technology, which poses unprecedented challenges to the security of cryptographic methods and potentially threatens the security of IPSec. In particular, Shor's algorithm [1] can efficiently factor large numbers and compute discrete logarithms, endangering key cryptographic algorithms used in IPSec, such as RSA and Elliptic-Curve Diffie-Hellman (ECDH) or Diffie-Hellman (DH). Additionally, Grover's algorithm [2] can reduce the effective security of symmetric encryption key lengths by half, thereby compromising the security of encryption algorithms utilized by the IPSec protocol. Significant advancements in quantum computing technology have already been made worldwide, and modern encryption systems are expected to become vulnerable to quantum computing attacks in the coming decades. Given the widespread use of IPSec, it is urgent to design a quantum-resistant IPSec protocol.

Many solutions have been proposed to address the security threats introduced by quantum computing technology. For example, some schemes employ Post-Quantum Cryptography (PQC) algorithms [3], which typically require larger key sizes. While these schemes can enhance security and integrate seamlessly with existing infrastructure, they often demand more computing resources and cannot fully guarantee protection against future quantum threats. Another effective solution is to use Quantum Key Distribution (QKD) technology [4]. By leveraging the fundamental principles of quantum mechanics, QKD can distribute symmetric secret keys (hereafter referred to as quantum keys) with information-theoretic security between two communicating parties. This ensures that key exchanges remain secure even in the quantum era, thereby enhancing the overall security of cryptographic applications. Therefore, incorporating QKD technology into the IPSec protocol represents a promising long-term solution for achieving quantum-resistant security.

There are two main types of solutions for integrating IPSec and QKD. The first category of approaches complements the Internet Key Exchange (IKE) protocol by combining quantum and classical keys in various ways. For example, the DARPA network introduced a QKD-based extension scheme [5] with two implementations: deriving IKE keys from quantum keys or employing One-Time Pad (OTP) encryption. However, the IKE Security Association (SA) remains unchanged and is established using insecure classical public key algorithms. The SeQKEIP protocol [6] introduces a phase for on-demand quantum key generation for authentication and encryption, but its data throughput is limited due to scarce quantum key resources. The QIKE protocol [7] focuses on key management for quantum keys, enabling fast key updates without renegotiation. Nonetheless, its fixed request rate fails to adapt to dynamic transmission requirements and key generation. The second category of approaches replaces IKE's key exchange function with QKD. For example, the fast rekeying protocol introduced in [8] eliminates the need for classical key exchange and allows for fast key updates using quantum keys, it requires the pre-configuration of SA parameters at both ends, resulting in poor

Digital Object Identifier: 10.1109/MCOM.004.2400475

The authors are with University of Science and Technology of China, China; Kaiping Xue and Jian Li are corresponding authors.

0163-6804/25/$25.00 © 2025 IEEE

flexibility and suitability. In summary, while existing approaches have made advances in some aspects, limitations remain in terms of security, efficiency, and flexibility.

To overcome the limitations of existing approaches, this article proposes a security-enhanced IPSec protocol integrated with QKD, named IPSeQ. To mitigate the threat posed by insecure asymmetric encryption algorithms, IPSeQ incorporates quantum keys into the key exchange, authentication, and encryption processes, aiming to safeguard the entire IPSec protocol. Designed concerning the first category, we retain the framework of the standard IKE protocol, which not only ensures flexible negotiation capabilities but also allows for seamless integration with traditional IPSec. Specifically, IPSeQ introduces a dynamic key updating mechanism that adapts the frequency of key updates in response to evolving quantum key generation rates and encryption key demand. By fully utilizing quantum keys, this dynamic key updating mechanism not only ensures secure data transmission but also avoids the degradation of transmission efficiency due to the exhaustion of quantum key resources. We analyze the adjustment of the key update frequency and prove that IPSeQ exhibits strong security properties. Finally, we implement a prototype system with real QKD devices and conduct experiments to validate its efficiency and robustness. The contributions of this article can be summarized as follows:

- We propose a security-enhanced IPSec protocol named IPSeQ, using QKD technology to mitigate quantum computing threats. It ensures comprehensive security for IPSec without disrupting the original IKE architecture. We also provide an efficient quantum key management design for IPSec.
- A dynamic key updating mechanism is proposed to enhance communication security by rapidly updating quantum keys and preventing key resource exhaustion, thus ensuring transmission efficiency. In addition, we adopt a sliding window mechanism to implement key synchronization during the rapid key update process.
- We build an IPSeQ prototype system with real QKD devices and conduct experimental validation. Experimental results demonstrate that IPSeQ notably enhances transmission efficiency and maintains stability even in scenarios where quantum keys are scarce.

The remainder of this article is organized as follows: In the next section, we briefly introduce the IPSec protocol and discuss the potential security threats to it in the quantum computing era. Then we present the detailed design of IPSeQ and perform a security analysis. Following that, we build an IPSeQ prototype system and demonstrate its superiority. Finally, we conclude the article.

Background

IPSec Protocol Suite

The IPSec protocol suite [9] is designed to ensure secure communication at the network layer. It provides confidentiality, integrity, and authenticated data transmission between two computers over an Internet Protocol network. It is commonly used in Virtual Private Networks (VPNs). IPSec can also protect the flow of data between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The IPSec protocol suite consists of two main protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). AH primarily ensures integrity protection, while ESP extends this by also providing confidentiality. The fundamental concept of IPSec is the SA, which defines the parameters necessary for securing IP traffic. An SA is identified by a unique Security Parameter Index (SPI) value. The IKE protocol plays a critical role in IPSec, as it is responsible for dynamically creating and maintaining SAs. IKE first generates a shared key using the DH key exchange algorithm (including both DH and ECDH for simplicity) to establish a secure communication channel. IKE then performs authentication using a variety of methods, including Pre-Shared Key (PSK), RSA signature, and digital certificates. Finally, the IKE peers negotiate IPSec SAs using the established secure channel. After the SA negotiation, IPSec employs encryption algorithms such as 3DES or AES to protect data transmission.

IPSec Procedures and Security Risks

As a widely adopted and more recent iteration of IPSec, IKEv2 provides a valuable point of comparison for our forthcoming analysis. The IKEv2 negotiation process mainly comprises three phases: the initiation phase, the authentication phase, and the IPSec SA establishment phase. We now analyze and evaluate the security risks at each phase of the IKEv2 process in the event of a quantum computing attack.

The First and Primary Risk Is the Key Exchange Aspect: In the initiation phase, the two parties begin communicating by sending `IKE_SA_INIT` messages to complete the key exchange, resulting in the generation of a shared seed key, `SKEYSEED`. The `SKEYSEED` is used to derive the encryption and authentication keys for the IKE SA, which is the foundation of protocol security. However, current key exchange methods generally utilize the DH algorithm, which is vulnerable to Shor's algorithm.

The Second Risk Is Authentication: In the second phase, the two parties authenticate each other by sending `IKE_AUTH` messages to ensure that they are legitimate. However, the RSA signature and digital certificate methods are susceptible to Shor's algorithm. In addition, although the PSK-based authentication approach is not vulnerable to quantum computing attacks, it is difficult to configure and manage the PSKs, and the security of these keys tends to deteriorate over time.

The Last Risk Is the Update of the Session Keys: After authentication is complete, the parties negotiate the IPSec SA for encrypted data transmission. It is necessary to renegotiate or update the session key after a specific lifetime or after a specific data volume to provide Perfect Forward Secrecy (PFS). The renegotiation of an expired IPSec SA is accomplished through the `CREATE_CHILD_SA` messages. However, the new DH key exchanges conducted during this process are susceptible to quantum computing technology.

Key Issues for Quantum-Resistant IPSec Design

According to the security analysis of IKEv2, to achieve a quantum-resistant IPSec protocol, we need to utilize quantum keys to eliminate security threats in the key exchange and authentication

To mitigate the threat posed by insecure asymmetric encryption algorithms, IPSeQ incorporates quantum keys into the key exchange, authentication, and encryption processes, aiming to safeguard the entire IPSec protocol.

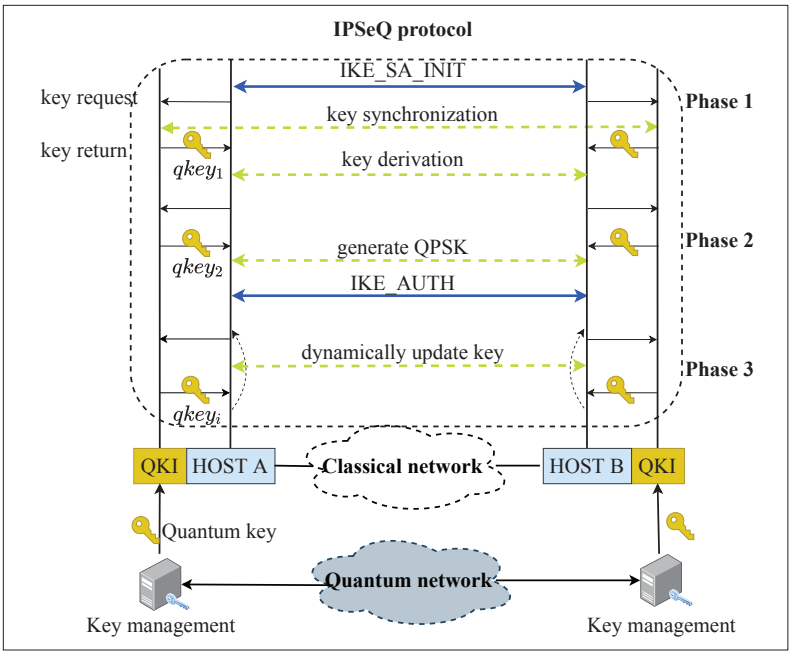

FIGURE 1. Overview of IPSeQ. Inside the dotted lines are the interactions performed in each phase. Original IKE messages are represented by solid blue lines, and yellow dashed lines represent interactions added by IPSeQ.

phases of the IKE protocol. Furthermore, to fully leverage the advantages of quantum keys, it is essential to facilitate rapid key updates between communicating parties, thereby enhancing the level of PFS. Notably, the quantum keys distributed between the two communicating parties are symmetric random keys, and the rate of quantum key generation decays exponentially with distance. Consequently, it is necessary to keep session keys synchronized and address the scarcity of quantum keys. In summary, the secure and efficient integration of QKD and IPSec needs to address the following three key challenges.

Key Management: The standard IPSec protocol does not provide interfaces for QKD. To distribute quantum keys to the IPSec process, a key management process is required. Furthermore, to achieve efficient and fast key updates, it is imperative to implement an effective key management approach that will provide keys on time and minimize communication overhead [7].

Key Synchronization: Inconsistencies in encryption and decryption keys may arise due to packet loss or out-of-order arrival of network packets in suboptimal network environments. To ensure accurate encryption and decryption, it is essential to design an efficient and reliable key synchronization mechanism.

Transmission Efficiency: The current state-of-the-art QKD systems can achieve key rates of a few hundred kb/s over limited distances. To increase the key rate and extend the transmission range, the key generated by QKD at one end node is relayed hop-by-hop to the other [10]. Many researches have explored the potential and implementation methods of key relaying to improve the rate in QKD networks [11], but the end-to-end key rate still cannot meet the high bandwidth demand due to the low link-level key rate. To maintain transmission stability, quantum key resources must be effectively utilized to avoid overconsumption.

A Security-Enhanced IPSec Protocol

Overview

In response to the security risks and key issues mentioned above, we developed an integrated approach called IPSeQ, which combines QKD technology with IPSec. For users who have deployed IPSec VPN tunnels, they can obtain quantum keys by accessing the QKD network and obtaining quantum keys from the key management system; they can also obtain quantum keys by directly connecting to the QKD device. In short, IPSeQ itself does not limit the access scenarios but only requires the node to have the ability to obtain quantum keys. It is an improvement of the IPSec protocol, which belongs to the application layer of the QKD network and does not require additional hardware. In this article, we introduce Quantum Key Interface (QKI) as a middleware and address important key management issues.

The workflow of IPSeQ is shown in Fig. 1. IPSeQ consists of three main phases. In Phase 1, IPSeQ introduces a QKD-based key exchange mechanism, providing quantum resistance to key material generated through DH exchange, thereby protecting the IPSec SA and IKE SA. In Phase 2, IPSeQ combines PSK and quantum keys to enhance authentication security and uses the quantum keys in the form of OTP, effectively mitigating the risk of PSK security degradation. In Phase 3, IPSeQ uses quantum keys to rapidly update session keys, further improving PFS, and achieves key synchronization through a sliding window mechanism. Additionally, IPSeQ dynamically adjusts the key reuse parameters to fully utilize quantum key resources and maintain transmission efficiency.

QKI Design

The QKI process retrieves the key from the key management process via the API, following our streamlined design based on the ETSI standard protocol to ensure interoperability and security. Meanwhile, the IPSec process requests encryption and decryption keys from the QKI using the internal key acquisition protocol. The IPSec process initiates registration by submitting a request to the QKI, identified via SPI. The QKI then allocates a key buffer for the request. Since each SA is unidirectional, a bidirectional encryption and decryption key pool must be divided between pairs of SAs. Multiple pairs of SAs may exist between IPSec VPNs, enabling the implementation of various security policies. Therefore, it is necessary for the QKI to manage the division of the quantum key pool effectively. Neppach *et al.* [7] proposed that the application obtains the key buffer with a fixed-rate request and refills it by setting a threshold value. However, since the traffic encrypted by each SA is uncertain and dynamically changing, the fixed-rate request can lead to an unfair distribution of quantum key resources and cause starvation problems. To address this, we utilize a fair division key management model that allocates equal-sized key buffers for each request. By monitoring the key buffer size and replenishing keys for requests that consume keys more quickly, we ensure the robustness of the service, even during traffic bursts on any SA.

IPSeQ Design

Phase 1: To protect IKE SA and IPSec SA from quantum computing threats, we propose a QKD-based key exchange mechanism that uses quantum key material to augment the insecure DH key exchange material. The combination of quantum and DH keys can be employed in various operating modes, including concatenation and Exclusive-OR (XOR). Here, we refer to the implementation in [12] and combine the first quantum key $qkey_1$ in the following way: The communicating parties add $qkey_1$ to derived key SK_d to provide quantum-resistant security to the key material used to generate IPSec SA and subsequent IKE SA. Then, they add $qkey_1$ to the initiator's integrity key SK_{pi} and the responder's integrity key SK_{pr} for computing the signature object, enabling both parties to detect any quantum key mismatch. To exploit the unconditional security of quantum keys, we use quantum keys as input for the key derivation function. The security relies on $qkey_1$ having sufficient entropy and the pseudo-random function is a secure one-way function that guarantees that the output key is statistically indistinguishable and therefore does not compromise unconditional security.

To ensure compatibility with the standard IPSec protocol, we retain the original DH exchange mode and include a quantum key exchange notification payload as an optional key exchange mode in the `IKE_SA_INIT` message. The advantages of this approach are as follows: First, we can seamlessly integrate IPSeQ with the standard IPSec protocol because no modifications to the standard IKE framework are necessary; only an extension field is added. Second, if the counterparty does not support QKD or has insufficient quantum key resources, it can revert to the traditional operation to ensure the protocol's robustness. Finally, if the quantum key exchange mode is accepted, both parties perform the following actions: the IPSec processes on both sides request a 32-byte quantum key $qkey_1$ from their respective QKIs. The QKIs then allocate key buffers for the IPSec processes, perform initial key synchronization, and return $qkey_1$. Both parties combine $qkey_1$ with DH key material to generate quantum-resistant key material to protect IKE SA and IPSec SA.

Phase 2: To avoid the use of insecure public key authentication methods and ensure the long-term security of PSK-based authentication, we use a combination of PSK and quantum keys for authentication in IPSeQ. Specifically, IPSeQ requires the two parties involved in a VPN tunnel to share a long-term key (denoted as *psk*), which serves as the authentication base during the handshake. Before the `IKE_AUTH` phase, both parties obtain the second quantum key $qkey_2$. They then combine *psk* and $qkey_2$ to compute the Quantum-PSK (QPSK) as follows: QPSK = **HMAC**($qkey_2$, *psk*), and use QPSK to generate the AUTH payload. Both parties compare whether the computed AUTH values are consistent. If *psk* or $qkey_2$ are inconsistent, both parties will detect an AUTH mismatch, causing the authentication to fail. The $qkey_2$ is used in OTP form, preventing an adversary from predicting the subsequent QPSK.

In most cases, using PSK instead of asymmetric encryption may introduce some limitations in key management. However, IPSec VPN tunnels between communicating parties are unlikely to be dynamic or involve a large number of participants due to deployment cost constraints. Nodes only need to pre-configure the PSK for the first connection, for example, by using a human messenger to deliver the PSK. The long-term security of the PSK can be guaranteed by QKD without the need for regular manual updates. In addition, the key generated by QKD can be used as the PSK to provide stronger security.

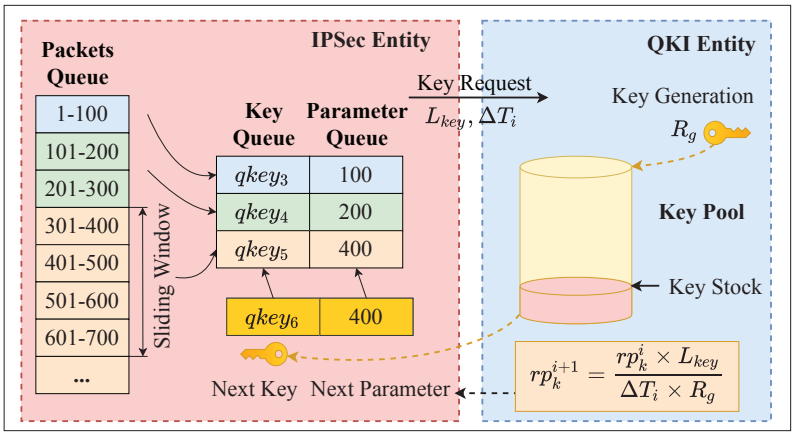

FIGURE 2. Dynamic adjustment and key synchronization mechanisms: Schematic Overview. The left part illustrates the sliding window and key queue within the IPSec process, while the right part depicts the QKI process and adjustments to key reuse parameters.

Phase 3: To address the security degradation issues associated with IPSec SA keys and to achieve secure and efficient key updates, we propose a fast key update scheme. This scheme enhances PFS by continuously updating session keys using quantum keys. It obtains new quantum keys directly from the local QKI, eliminating the need for DH key exchanges or frequent SA replacements, thereby reducing communication overhead. In the following section, we describe the key update scheme in detail and explain how it addresses the synchronization and efficiency challenges mentioned previously.

Key Synchronization: Since both communicating parties obtain quantum keys independently from the QKI device, it is necessary to first synchronize the starting position of the quantum key streams. To ensure consistency between the encryption and decryption keys during rapid session key updates, it is important to note that the ESP packet header contains a 4-byte sequence number field. This number is incremented by one for each packet sent, allowing it to be used to associate packets with their corresponding quantum keys. Specifically, a sliding window is maintained, with each window corresponding to the protection range of a quantum key. During the encryption and decryption process, the sliding window adjusts to ensure seamless alignment between packets and their corresponding quantum keys. As illustrated in Fig. 2, the $qkey_3$ protects packets with sequence numbers ranging from 1 to 100. At this point, the sliding window covers sequence numbers 1 to 100. With each change in the quantum key, the sliding window shifts to cover sequence numbers 101 to 300, and subsequently, 301 to 700. We depend on the reliable TCP communication mechanism between

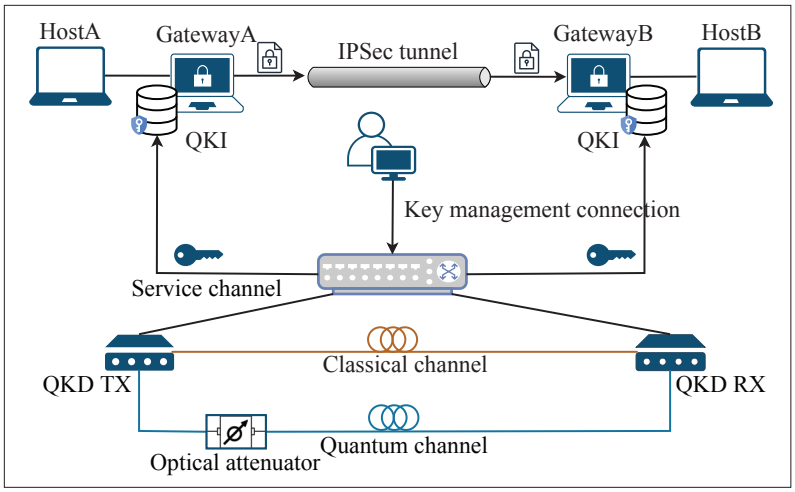

FIGURE 3. Experimental setup: device framework and network topology. Two QKD devices run the QKD protocol, with key management handling key processing and acquisition. The QKI obtains quantum keys via APIs for storage in the key pool. Four PCs are arranged in a linear topology.

QKIs to keep the sliding windows of the encryption and decryption parties synchronized, ensuring robustness even in environments with high latency and high packet loss.

Dynamic Key Updating: The rate of quantum key generation between distant communicating parties is low and often fluctuates. This inevitably leads to a gap between the insufficient supply of quantum keys and the high demand for data transmission over the Internet. A fixed-rate key update mechanism will either exhaust its supply of keys or accumulate a large number of unused keys. To tackle this issue, we introduce the key reuse parameter rp_k to regulate the number of times each quantum key is utilized. Given that both quantum key generation and transmission requirements fluctuate dynamically, the rp_k can be adjusted in real-time to maintain a balance between quantum key generation and consumption. The specific adjustment mechanism is as follows: To maintain the sustainable service of the key, we hope that the key consumption rate R_c does not exceed the key generation rate R_g. Given the current transmission demand R_t and the key reuse parameter rp_k, along with each packet having a length of MTU and a key length of L_{key}, to meet the condition $R_c \le R_g$, it is necessary that:

$$rp_k \ge \frac{R_t \times L_{key}}{R_g \times MTU}.$$

While R_t cannot be given directly, it can be estimated from the previous round's key reuse parameter rp_k^i (round i) and time interval of request ΔT_i using the relation:

$$R_t^i = \frac{rp_k^i \times \mathrm{MTU}}{\Delta T_i}$$

Substituting this into the inequality and taking the equal sign directly, we lead to the adjustment equation:

$$rp_k^{i+1} = \frac{rp_k^i \times L_{key}}{\Delta T_i \times R_g}.$$

Through this regulation, on the one hand, when the transmission demand increases or the key generation is insufficient, we can increase rp_k in time to ensure the sustainability of the service; on the other hand, when the transmission demand decreases or the key is sufficient, we can decrease rp_k to improve the security. As shown in the QKI Entity calculation in Fig. 2, The current transmission process is stable, R_c and R_g are equal, and the next rp_k value is unchanged.

Key Reuse Parameter Analysis

Regarding key length, a minimum of 128 bits is required due to the Grover search algorithm. A key length of 256 bits has been demonstrated to be effective in defending against quantum computing attacks. The current generation of commercial QKD solutions is capable of generating quantum keys at a rate of 10 kb/s over a distance. The application of AES-128 encryption, with a key generation rate of up to 12,800 bps, allows for approximately 100 key changes per second. For unidirectional IPSec channels, this results in a key cycle of approximately 20 milliseconds. Considering a data transmission rate of 100 Mb/s, this implies that the key reuse parameter for sustained service is approximately 170. As a conservative upper bound, if a high-speed connection of 10 Gb/s were employed, each key would need to protect 17,000 packets. In general, key reuse parameters should be adjusted based on the key generation rate and data transmission requirements. Reducing the amount of data protected by each key is an effective method of reducing the risk of compromise.

Formal Verification

We use the formal security protocol verification tool Tamarin [13] to model the security of IPSeQ. We consider an extension of the Dolev-Yao (DY) attacker as our threat model. The DY-attacker has complete control over the network and can intercept, send, replay, and delete any message. We assume that the attacker has quantum computing power to break the DH key. We first model the quantum key and PSK as independent symmetric keys. Then, we generate Tamarin rules and lemmas to establish the security properties of the protocol. We prove most of the specified security requirements for IPSeQ, including peer-to-peer authentication, session key confidentiality, and PFS of session keys. Here, we explain how we achieve these security properties.

Authentication: Both parties can identify the other party during the handshake. If either party does not have access to the *psk* or $qkey_2$, the authentication process will fail. Even if the *psk* is compromised, an adversary cannot impersonate the parties because the handshake is secured by the quantum key.

Consistency: When two honest parties agree on a matching session, they also agree on the session key. This is because sessions are uniquely identified based on SA pairs. Consequently, when two peers observe sessions with the same SA pairs, they both agree on the key material to be used for the session.

Key Secrecy: Once the security-enhance IPSec protocol has been successfully completed, only the initiator and the responder know the key. This confidentiality is ensured by QKD technology. Due to the no-cloning theorem, any eavesdropping attempt by an adversary to acquire the quantum key would be detected. Even if the DH shared key

is compromised, a successful handshake remains unattainable without the correct quantum key.

Perfect Forward Security: PFS ensures that the exposure of a long-term secret, such as the *psk*, has no direct impact on the IPSec session. This is achieved through the PFS property of the quantum key and the frequent updates of encryption and decryption keys.

We then extend the security analysis to potential side-channel vulnerabilities and resilience to Denial of Service (DoS) attacks, and describe how IPSeQ can prevent or mitigate these attacks in real-world deployment scenarios.

Side-Channel Attack: Side-channel attacks exploit leaks in physical implementations, like timing and power use, rather than attacking the encryption directly. IPSeQ boosts security by minimizing data exposure through frequent key updates, preventing attackers from recovering key information.

DoS Attack: IPSeQ is not inherently resistant to DoS attacks, and attackers may exhaust quantum keys by forging a large number of IKE requests. We can mitigate these attacks to some extent by using strategies such as flow control, deployment of firewalls, and intrusion detection systems.

Testbed Implementation

Experimental Methods

To evaluate the performance of IPSeQ, we conduct tests using actual QKD devices and network platforms. The experimental framework, depicted in Fig. 3, consists of three main components.

QKI Process: In the QKI, we configure specific parameters including an initial key reuse parameter of 100, an upper bound on the key reuse parameter of 10000, and a lower bound of 64.

IPSec Process: We adopt the open-source software strongSwan for the IPSec process, with AES128GCM16 set as the authentication encryption scheme. Additionally, we employ user-mode IPSec to implement our scheme to facilitate the obtaining and changing of keys.

Test Platforms: We set up a test network using four Ubuntu mini-PC hosts. Two serve as gateways to protect the subnets, while the other two are within each subnet. Two QKD devices provide quantum keys for the gateways. The equipment used is a pair of QKD-PHA1250-S high-speed time-phase encoding QKD systems from QuantumCTek. The system's performance range extends from 100 kb/s at approximately 0 dB to 1 kb/s at less than 30 dB. A quantum channel is used for the QKD process, and a classical channel is used for post-processing. Additionally, we use an optical attenuator to simulate the loss of optical fiber over distance.

In our experiments, we first compare IPSeQ with the standard IPSec scheme in terms of processing delays, the latency of each packet using the ping command. Subsequently, we quantify the overhead of SA key replacement and the time required to establish an IPSec connection. Additionally, to demonstrate that IPSeQ can achieve fast key updates and maintain key synchronization, we evaluate the transmission performance between two IPSec gateways using iPerf, a tool for actively measuring the maximum achievable bandwidth on IP networks. Finally, to verify the effectiveness of the dynamic key updating mechanism and its feasibility in practical noisy quantum channels, we evaluate the impact of optical distance on the key generation rate and key reuse parameters.

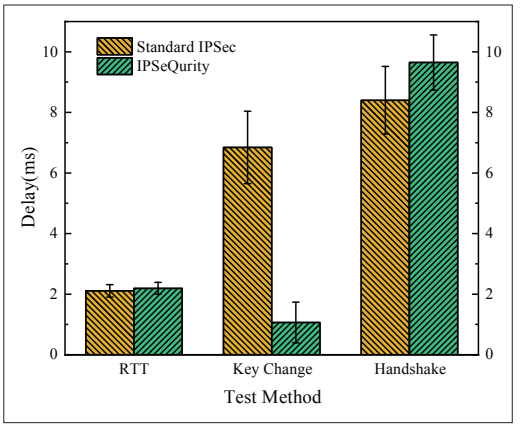

FIGURE 4. Latency impact evaluation: RTT, key change delay, and handshake delay measurements. RTT measures the delay between two hosts, Key Change measures the delay in performing a key update, and Handshake measures the delay in completing the IKE negotiation.

Experimental Results

Figure 4 shows that the performance of IPSeQ is comparable to the standard scheme in terms of RTT, with an RTT of about 2 ms between Host A and Host B. We also observe that the standard scheme incurs a key change delay of about 7 ms due to the `CREATE_CHILD_SA` exchange. In contrast, IPSeQ obtains the key directly from the key pool, resulting in a communication delay of about 1 ms, thereby significantly reducing the overhead. Additionally, the standard handshake delay is more than 8 ms, while IPSeQ introduces two quantum key enhancement processes with a combined delay of less than 10 ms. The differences in latency between IPSeQ and the standard method are small and acceptable, while IPSeQ offers significantly enhanced security.

Figure 5 shows the throughput performance of IPSeQ compared to the standard approach at different key update frequencies. The standard approach relies on the `CREATE_CHILD_SA` exchange, updating the key when the application sends a certain number of packets. This exchange involves the interaction of multiple packets, leading to excessive overhead. Therefore, when the key update frequency is high, the throughput drops dramatically. In contrast, IPSeQ uses the key reuse parameter to control key updates and obtains quantum keys directly from the QKI, minimizing interaction overhead. Even as the update frequency increases, the throughput remains relatively unaffected and stays at a high level. As for the scenarios without key updates, IPSeQ experiences only about 7 percent performance degradation caused by the overhead associated with QKI communication, key synchronization, and other related processes. Additionally, in cases of rapid key updates, Fig. 5 demonstrates that IPSeQ outperforms the standard scheme in terms of packet loss. This is because the standard scheme may cause inconsistencies in packet decryption during the key renegotiation process of an IPSec SA. In contrast, IPSeQ maintains good synchronization between packets and keys through the sliding win-

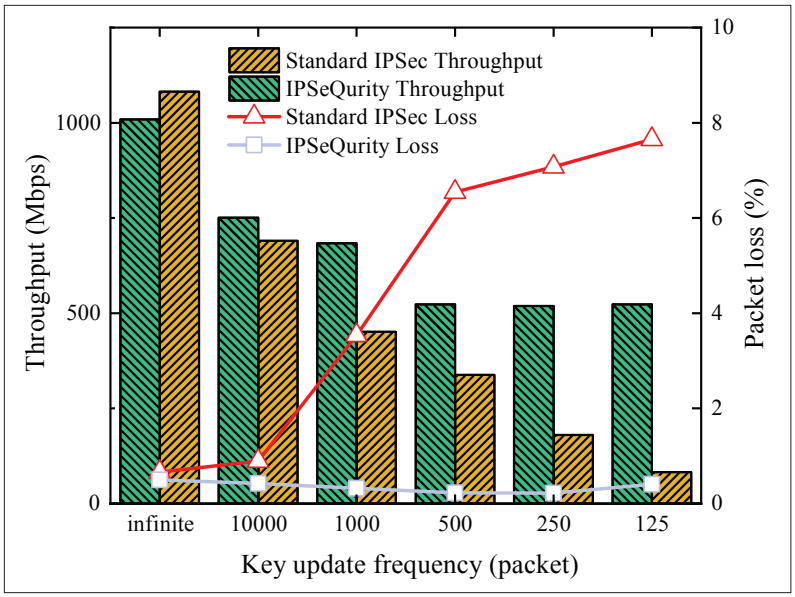

FIGURE 5. Impact on throughput and packet loss: Analysis at Various Update Frequencies. Assuming the key pool is adequately resourced, the iPerf tool runs for 60 seconds to test average throughput and packet loss rate. Results indicate that IPSeQ performs better at higher update frequencies.

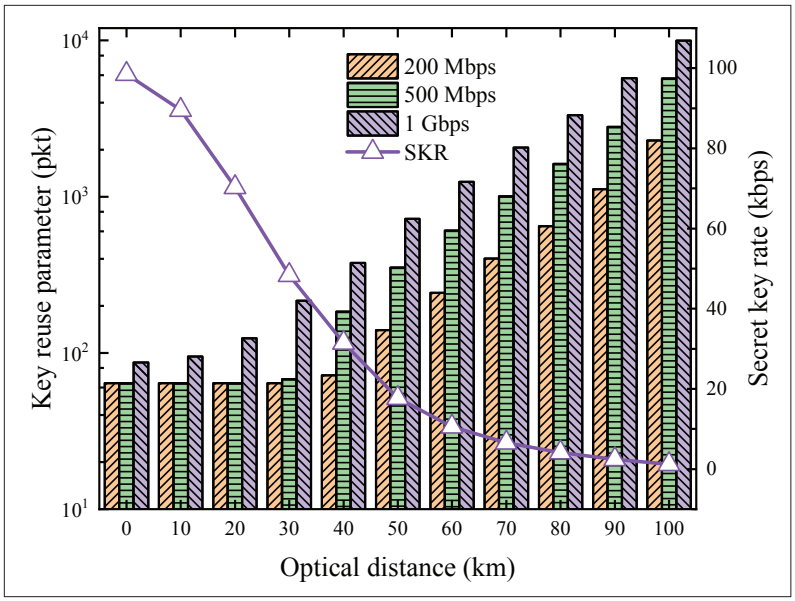

FIGURE 6. Analysis of dynamic key update mechanism over varying optical distances. The optical attenuator is adjusted to simulate a gradual increase in channel length from 0 to 100 km, with transmission bandwidths set at 200 Mb/s, 500 Mb/s, and 1 Gb/s.

dow mechanism, thus minimizing packet loss.

Figure 6 shows the variation in the average key generation rate and the average key reuse parameter as optical distance increases (corresponding to 0.2 dB/km fading) during the test. The results indicate that the key generation rate decays rapidly with increasing optical distance while the average key reuse parameter gradually rises. Compared to data transmission at 200 Mb/s, the average key reuse parameter is higher at 500 Mb/s bandwidth and the highest average key reuse parameter is found at 1 Gb/s. This is because, when the key generation rate is fixed, an increase in transmission rate also increases key consumption. Therefore, the key reuse parameter must be adjusted upward to maintain a balance between key generation and consumption rates. Experiments also demonstrate that our solution maintains high transmission efficiency even at low key generation rates.

Conclusion and Perspect

This article assesses the security threats posed by quantum computing technology to IPSec, emphasizing vulnerabilities in DH key exchange, authentication methods, and key updates. In response to these challenges, we proposed a comprehensive solution that seamlessly integrates quantum keys into the entire IKE process to ensure both security and flexibility. We introduced the QKI design and a dynamic key updating scheme, effectively addressing key management issues and facilitating secure, efficient data transmission. Additionally, a sliding window mechanism is proposed to resolve key synchronization issues during rapid key updates. Our comprehensive security analysis demonstrates that IPSeQ can eliminate security flaws, thereby enhancing the overall security of the IPSec protocol. The advantages of IPSeQ were verified through testing on a dedicated platform and with actual QKD devices. Experimental results show that IPSeQ can significantly improve transmission efficiency while maintaining stability in scenarios with key scarcity.

In future work, we plan to explore enhancements to the security of IPseQ and further improve its efficiency. This includes implementing the Q-CSKDF scheme [14], which can consistently generate high-rate derived keys while ensuring the desired level of security. Additionally, integrating QKD with PQC offers dual protection against quantum computer attacks, addressing current security needs and safeguarding against future quantum threats, thereby boosting IPSeQ's reliability.

The large-scale deployment of QKD systems is becoming increasingly feasible due to key advancements. Practical deployments, such as the space-to-ground networks in China and the OpenQKD project in Europe [11], have demonstrated the stability and versatility of QKD applications. In this emerging landscape, classical and quantum networks are anticipated to coexist, creating new opportunities for enhanced information security [15]. Quantum networks will supply secure quantum keys to classical networks, while IPSeQ can ensure secure classical communication, facilitating the functional realization of future quantum networks. We are optimistic that continued innovation and collaboration will enable IPSeQ and similar breakthroughs to play a pivotal role in developing secure, next-generation quantum networks.

Acknowledgments

This work was supported in part by the Innovation Program for Quantum Science and Technology under Grant No. 2021ZD0301301, the Anhui Initiative in Quantum Information Technologies under Grant No. AHY150400, the National Natural Science Foundation of China under Grants No. 62402466 and No. 62201540, and the Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. Y202093.

References

[1] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," *Proc. 35th Annual Symposium on Foundations of Computer Science*, IEEE, 1994, pp. 124–34.
[2] L. K. Grover, "A Fast Quantum Mechanical Algorithm for

Database Search," *Proc. 28th Annual Symposium on Theory of Computing*, ACM, 1996, pp. 212–19.
[3] A. Pazienza *et al.*, "Analysis of Network-Level Key Exchange Protocols in the Post-Quantum Era," *Proc. 15th Workshop on Low Temperature Electronics*, IEEE, 2022, pp. 1–4.
[4] Z. Li *et al.*, "Entanglement-Assisted Quantum Networks: Mechanics, Enabling Technologies, Challenges, and Research Directions," *IEEE Commun. Surveys & Tutorials*, vol. 25, no. 4, 2023, pp. 2133–89.
[5] C. Elliott, D. Pearson, and G. Troxel, "Quantum Cryptography in Practice," *Proc. 2003 Conf. Applications, Technologies, Architectures, and Protocols for Computer Commun.*, ACM, 2003, pp. 227–38.
[6] M. Sfaxi *et al.*, "Using Quantum Key Distribution Within IPSec to Secure MAN Communications," *Proc. 2005 Conf. Metropolitan Area Networks*, 2005.
[7] A. Neppach *et al.*, "Key Management of Quantum Generated Keys in IPSec," *Proc. 2008 Int'l. Conf. Security and Cryptography*, SCITEPRESS, 2008, pp. 177–83.
[8] S. Marksteiner and O. Maurhart, "A Protocol for Synchronizing Quantumderived Keys in IPSec and Its Implementation," *Proc. 9th Int'l. Conf. Quantum, Nano/Bio, and Micro Technologies*, IEEE, 2015, pp. 35–40.
[9] P. Hoffman, "Cryptographic Suites for IPSec," 2005, RFC 4308, IETF; available: https://www.ietf.org/rfc/ rfc4308.txt; accessed Dec. 2024.
[10] P.-Y. Kong, "Challenges of Routing in Quantum Key Distribution Networks With Trusted Nodes for Key Relaying," *IEEE Commun. Mag.*, vol. 62, no. 7, 2024, pp. 124–30.
[11] M. Mehic *et al.*, "Quantum Key Distribution: A Networking Perspective," *ACM Computing Surveys*, vol. 53, no. 5, 2020, pp. 1–41.
[12] S. Fluhrer *et al.*, "Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-Quantum Security," 2020, RFC 8784, IETF; available: https://www.ietf.org/rfc/rfc8784.txt; accessed Dec. 2024.
[13] D. Basin *et al.*, "Tamarin: Verification of Large-Scale, Real-World, Cryptographic Protocols," *IEEE Security & Privacy*, vol. 20, no. 3, 2022, pp. 24–32.
[14] L. Chen *et al.*, "Q-CSKDF: A Continuous and Security Key Derivation Function for Quantum Key Distribution," *IEEE Network*, vol. 38, no. 5, 2024, pp. 123–30.
[15] L. Gyongyosi and S. Imre, "Advances in the Quantum Internet," *Communications ACM*, vol. 65, no. 8, 2022, pp. 52–63.

Biographies

XUMING GAO (dearlanxing@mail.ustc.edu.cn) received his bachelor's degree in information security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC), in 2019. Currently, he is a graduate student in the School of Cyber Science and Technology, USTC. His research interests include quantum networking and network security.

KAIPING XUE [M'09, SM'15] (kpxue@ustc.edu.cn) received his bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. Currently, he is a Professor in the School of Cyber Science and Technology, USTC. His research interests include future Internet architecture design, transmission optimization, and network security.

JIAN LI [M'20, SM'23] (lijian9@ustc.edu.cn) received his bachelor's degree from the Department of Electronics and Information Engineering, Anhui University, in 2015, and received his Ph.D degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2020. He is currently an associate researcher with the School of Cyber Science and Technology, USTC. His research interests include future Internet architecture design and quantum networking.

ZHONGHUI LI (leestone@ustc.edu.cn) received his bachelor's degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China, in 2018 and received his dcotor's degree in cyber security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC), in 2023. He is currently a Post-Doctoral researcher with the School of Cyber Science and Technology, USTC. His research interests include quantum networking and network security.

JIAQI WU (wujiaqi191383@mail.ustc.edu.cn) received her bachelor's degree in information security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC), in 2019. Currently, she is a graduated student in the School of Cyber Science and Technology, USTC. Her research interests include quantum networking and network security.

NENGHAI YU (ynh@ustc.edu.cn) received his bachelor's degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987, the M.E. degree from Tsinghua University, Beijing, China, in 1992, and his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), Hefei, China, in 2004. Currently, he is a Professor in the School of Cyber Science and Technology, USTC. His research interests include multimedia security and quantum networking.

QIBIN SUN [F'11] (qibinsun@ustc.edu.cn) received the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1997. He is currently a professor in the School of Cyber Science and Technology, USTC. His research interests include multimedia security, network intelligence and security and so on.

JUN LU (lujun2019@ustc.edu.cn) received his bachelor's degree from Southeast University in 1985 and his master's degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC) in 1988. He is currently a professor in the School of Cyber Security and Technology and the Department of EEIS, USTC. His research interests include theoretical research and system development in the field of integrated electronic information systems. He is an Academician of the Chinese Academy of Engineering (CAE).