A Survey on the Security Issues of QUIC

Conference Paper · October 2022

DOI: 10.1109/CSNet56116.2022.9955622

2 authors, including:

Y A Joarder
Concordia University Montreal
16 PUBLICATIONS **24** CITATIONS

SEE PROFILE

A Survey on the Security Issues of QUIC

Y A Joarder
Concordia Institute for Information Systems Engineering (CIISE)
Concordia University
Montreal, Canada
y_joarde@encs.concordia.ca

Carol Fung
Concordia Institute for Information Systems Engineering (CIISE)
Concordia University
Montreal, Canada
carol.fung@concordia.ca

2022 6th Cyber Security in Networking Conference (CSNet) | 979-8-3503-9722-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CSNET56116.2022.9955622

***Abstract*—A newly established multiplexed network protocol – QUIC, which is based on User Datagram Protocol (UDP), has emerged in recent years and gained a large share of Internet traffic quickly. Initially proposed by Google, the goal of QUIC is to achieve a higher Internet communication performance and eventually replace the Transmission Control Protocol (TCP) + Transport Layer Security (TLS) + HTTP/2 architecture. In particular, the 3^{rd} version of the Hypertext Transfer Protocol – HTTP/3.0 is built on top of QUIC. A good number of research papers have been published recently to evaluate the performance and security of the QUIC protocol. In this paper, we conduct a comprehensive survey on the QUIC security issues and analyze its future research directions regarding security prospective. We investigate several topics including the QUIC protocol structure, QUIC security model, security issues related to QUIC protocol, and future research directions on QUIC Security. To the best of our knowledge, it is the one of first surveys that focus on the security of the QUIC protocol.**

***Index Terms*—QUIC, Survey, TLS, Network Security, HTTP/3, Network Protocol, Transport Layer Protocol, TCP, UDP, Vulnerabilities**

I. Introduction

In recent years, a contemporary general-purpose, reliable, latency reducing, connection-oriented and secure transport layer network protocol: QUIC [1] has emerged and has gained popularity quickly. It is now the default transport layer encrypted protocol for the majority of well-known applications, including Facebook, Gmail, Instagram, Google Chrome, and YouTube [2]. Interestingly, from the operating system perspective, it looks like an application layer protocol that behaves like a transport layer network protocol. The QUIC protocol's primary goals are to increase Internet traffic's speed and reduce latency by decreasing connection establishment duration [3], multiplex without Head-of-Line (HOL) blocking [4], and provide invariably-encrypted edge-to-edge security [5]. In 2012, Google first introduced a new transport layer network protocol built on User Datagram Protocol (UDP) titled "gQUIC", to address the web traffic performance issues at that time [6], [7]. In 2016, the Internet Engineering Task Force (IETF) formed a research group to enlarge and standardize QUIC. The effort led to the standardized QUIC protocol as RFC 9000 in May 2021 [8]. At around the same time, RFC 9001 [9] was released that standardizes how TLS 1.3 functions as a security component of QUIC protocol. It is worth noting that HTTP/3 [10] connections can only be established using QUIC. It is developed as a better substitute for Transmission Control Protocol (TCP) [8]. It has multiple unique or pioneer characteristics that surpass TCP in various areas theoretically. For instance, it offers a 0-Round Trip Time (0-RTT) handshake mechanism to reduce handshake latency [11]. Although the same feature is possible in TCP by combining the use of TCP Fast Option (TFO) and 0-RTT (early data) in TLS 1.3, recent version of QUIC is superior to TFO regarding security aspect of 0-RTT handshake mechanism [12]. Since the 0-RTT feature was initiated by QUIC and it is performing better in QUIC architecture compared to the TFO, we can consider 0-RTT to be a pioneer feature of QUIC. By using multiplexing approach, it also overcomes HOL blocking issue, which is one of the major problems of TCP. For being mobility-friendly and responsive, it has connection migration feature as well [13]. Note that connection migration feature stands for switching from one type of network to another type of network. For example, switching from Local Area Network (LAN) connection to Wide Area Network (WAN) connection. There are already a number of QUIC implementations in use, some of these use gQUIC and others QUIC. gQUIC is still used by about 8% of the top 10 million leading websites, according to latest assessments on those sites [14]. In contract, approximately 25% of websites worldwide presently use HTTP/3 over QUIC [15]. More than 75% of internet traffic of Facebook uses QUIC and HTTP/3 by October 2020 [16].

In spite of a de novo design brought higher performance to QUIC, security loopholes still exist. QUIC cannot perform according to its potentiality in real world. As a result, every now and then, cyber attackers contravene QUIC protocol's security [5]. Large Technological Organizations, Internet Content Providers (ICPs), and other businesses are increasingly embracing

QUIC, making it a desirable target for malicious attackers. Thus, analyzing and enumerating security issues and threats of QUIC on existing network services is completely vital. A lot of research works have been conducted and published in the literature regarding the security issues around QUIC [5], [17], [18], [19], [20], [21]. Unfortunately, there is still a lack of a pervasive inspection in the pertinent literature on QUIC security. In this paper, we aim at filling the gap and conduct a comprehensive survey on the QUIC protocol's security related issues that have been published so far.

Contributions: In this paper, we introduce the essential features of QUIC and its development history. We also provide security analysis on QUIC, including the security model, security issues, and types of security threats. Finally, we discusses probable future research directions for QUIC security related challenges. The contributions of this paper can be summarized as follows:

- To fully comprehend the QUIC protocol's operational principles
- To analysis the QUIC protocol's security model
- To identify all the existing security issues and threats of QUIC protocol
- To vision the directions of future research on problems related to QUIC security

The rest of the paper is structured as follows. Section II describes synopsis of QUIC protocol. Section III presents security analysis of QUIC. Section IV provides the discussion and our vision of future research scopes on QUIC security. Finally, Section V concludes the paper.

II. Synopsis of QUIC Protocol

In this section, we will succinctly introduce overall operational principles (synopsis) of QUIC protocol, including the architecture, protocol characteristics, connection establishment process, and packet structure.

A. Architecture of QUIC Protocol

QUIC exploits the operating system's UDP (User Datagram Protocol) socket downstream to give application layer protocols (e.g., HTTP/3) a dependable and secure transmission channel. Although the implementation is based on UDP, which is a transport layer protocol, QUIC does not rely on UDP's features in its protocol design and does not use UDP ports to indicate a transport layer connection. Since QUIC builds upon UDP at the transport layer, which is the reason that it has been often described as a transport layer protocol [1], [22].

The TCP/IP network uses five tuples: source port, destination port, protocol, source IP addresses and destination IP addresses, to uniquely identify a connection. In QUIC protocol, a globally unique randomly generated Connection ID is used identify a connection. The QUIC connection created in the one (original) network can be easily moved to a new network, so that the network service won't be disrupted when the user switches their network. It is particularly useful for handover process in cellular networks or WiFi networks. Figure 1 illustrates the architectural view of the QUIC protocol.

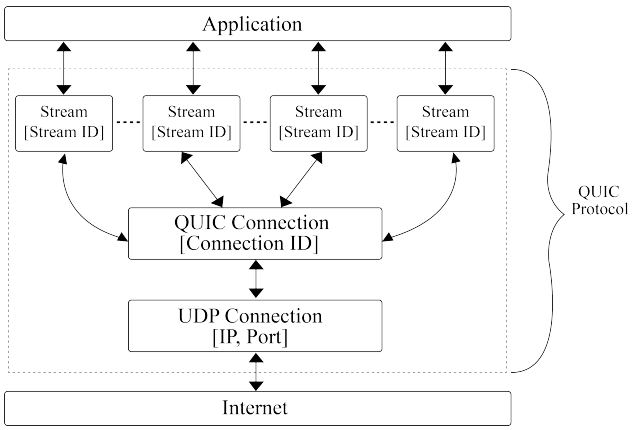

Fig. 1: Architecture of the QUIC protocol

B. Protocol Characteristics of QUIC

The two most significant new features of QUIC, compared to TCP, are capability for multiplexing on a single connection and reduced handshake latency.

1) Multiplexing Overview: Multiplexing competency allows QUIC to circumvent TCP's Head-of-Line (HOL) blocking issue [23]. As illustrated in Figure 2, TCP connection maintains a First In First Out (FIFO) channel, which requires the receiver to strictly follow the order of the sender while processing the received data. As shown in the example (in Figure 2), the client transmits to the server packets 3 and 4 . If packet 4 comes before packet 3, the Upper Layer Application will hold on the processing of packet 4 until packet 3 is received. This delayed is called HOL blocking [24], [25] in TCP.

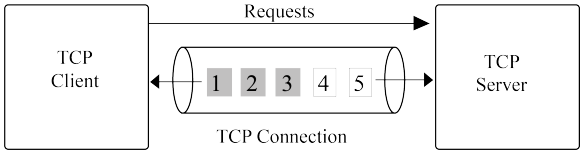

Fig. 2: Singleplexing in TCP

QUIC can fix the HOL blocking issue by adding capability for multiplexing on a transport layer connection. Figure 3 shows the multiplexing mechanism in QUIC. Under the transport layer connection, the idea of stream is introduced. Multiple streams can be present over one QUIC connection; however, these streams are independent of one another and each assures the FIFO

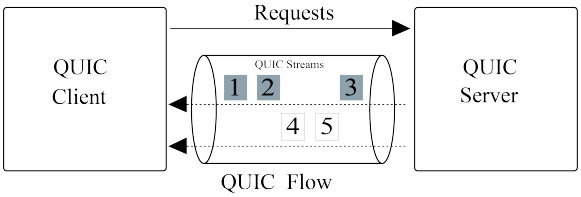

Fig. 3: Multiplexing in QUIC

order. From this perspective, "Flow" in QUIC is similar to "connection" in TCP because both terms are used for FIFO communication channel. In QUIC, however, many flows exchange connection information. By using a single QUIC connection for numerous streams instead of many TCP connections, multiplexing is accomplished while also conserving system compatibility.

2) Handshaking Mechanism: QUIC developed its own handshake protocol and outperformed TCP+TLS in terms of handshake latency. Transport Layer Security is referred to as TLS [26], [27]. To guarantee data security, it employs both Asymmetric and Symmetric Encryption.

The Diffie-Hellman algorithm or the RSA algorithm completes the handshake protocol in TLS1.2. In RSA algorithm, the handshake requires 2 Round Trip Times (2-RTTs); because, at first the two parties (client and server) exchange their respective RSA public keys before sending a freshly created shared key using RSA Encryption. On the other hand, TLS1.3 uses only Diffie-Hellman Algorithm for key exchange.

The RTT distinction between TCP (TCP+TLS 1.2 and TCP+TLS 1.3) and QUIC Handshake RTT is depicted in Figure 4a. For TCP+TLS 1.2, a minimum of 3-RTTs must pass during the handshake between the client and the server, including 1-RTT for the TCP handshake and 2-RTTs for the TLS handshake. The RTT handshake for TCP+TLS1.3 protocol is reduced to 2-RTT. QUIC has further optimized RTT handshake . It allows TLS Handshakes to take place in parallel with a transport layer handshake, which reduced the delay to 1-RTT. Therefore, This handshake of QUIC is referred to as a 1-RTT Handshake. If the client has previously connected to the server, the shared key will be reused and the client can immediately connect. In other words, data transfer does not need to wait for the handshake to be finished. This procedure of QUIC is called as 0-RTT. It dramatically minimizes the client's handshake latency, which can greatly enhance the communication efficiency.

C. Connection Establishment Process in QUIC

In QUIC, the handshake procedure is used to establish the first connection. Figure 4b represents the QUIC's handshake process in details. The client sends the server an initial packet at the beginning of the connection, which contains a TLS 1.3 "Client Hello" message. The initial packet, which includes TLS "Server Hello", is then returned by the server. A handshake packet made up of certificates, encrypted extensions, and other TLS server communications comes next. A message from the client is delivered after the handshake. Using 1-RTT packets, application data can now be sent.

D. Packet Structure of QUIC

Each QUIC packet contains two main sections: "Header" and "Payload". The header portion of a packet contains the data necessary to make sure the data reaches the desired destination. The payload portion of a packet includes the data that the packet is intended to deliver. The Header section of QUIC can be long or short depending on the scenario, which is one of the major differences between QUIC and TCP. Until both 1-RTT packet protection and version negotiation are finished, long header packets are utilised for the first exchange. The majority of data is carried via short header packets. Figure 5 represent Long and Short Header Packets successively. A 1-RTT "Protected Payload" is always included in packets with a short header.

III. Security Analysis of QUIC

As a novel transport layer network protocol, QUIC has attracted much attention in research recently, including the evaluations on its security features. We can divide them into 3 major categories: 1) 0-RTT Handshake and Forward Security 2) QUIC's Security Model 3) Security Threats of QUIC. Each category is elaborated in the following subsections.

A. 0-RTT Handshake and Forward Security

An important security feature of communication protocols is forward security, which prevents the disclosure of earlier session keys in the event where the long-term master key is compromised [28]. Both TLS1.3 of QUIC and TLS1.2's 1-RTT handshake effectively provide Forward Security feature. However, the QUIC TLS1.3 0-RTT handshake, is unable to offer forward security from both communicating parties [5]. This is due to the fact that the initial session key, which is utilized in the 0-RTT communication process, is produced in accordance with the server's static configuration. The key leading to 0-RTT will also be exposed if the configuration is compromised in the future. It is a great security concern of QUIC protocol. To address this issue, Günther et al. [29] modified the server side of QUIC's 0-RTT handshake to provide forward security. The authors employed a unique key design. The current key is changed each time whenever the server decrypts the ciphertext, and the new key can be interpreted as the original key. This way forward security is guaranteed

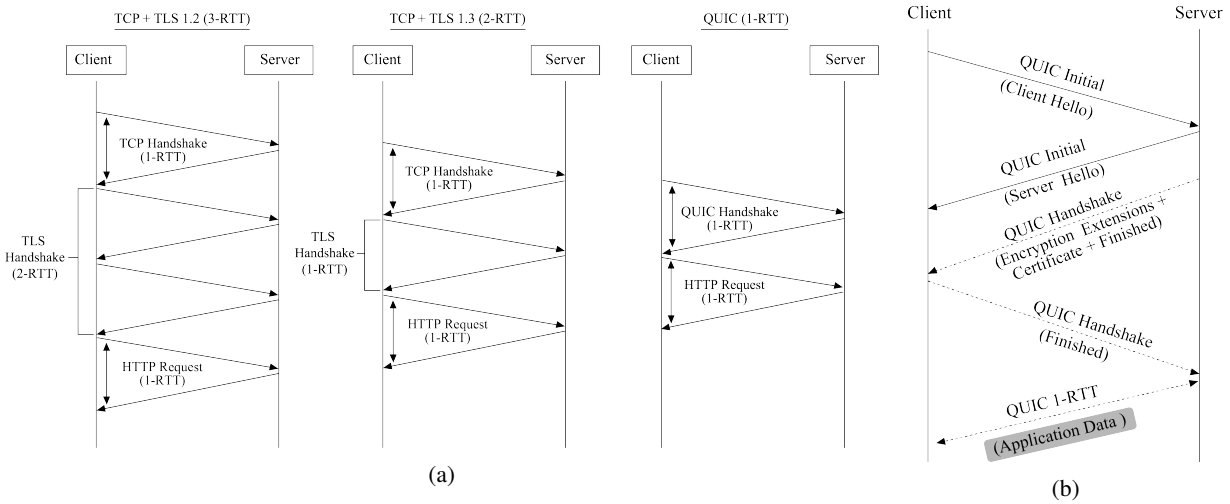

Fig. 4: (a) TCP and QUIC Handshake Latency illustration (b) QUIC's Handshake Procedure

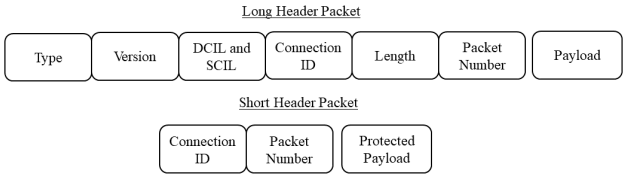

Fig. 5: Long and Short Header Packet Structure of QUIC

since the parsed ciphertext cannot be reverse-encrypted or decrypted. QUIC protocol will also no longer be susceptible to Replay Attacks as a result of this design.

B. QUIC's Security Model

Fischlin et. al. [30] used a multi-stage key exchange model to demonstrate the security flaws in QUIC's handshake, where the security of the QUIC protocol cannot be guaranteed even if both communicating parties employ a safe encryption protocol that includes an authentication mechanism for Data Encryption. To address this issue, the authors suggested QUIC*i*, a key-independent version, which is capable of conforming to their security paradigm. QUIC*i* implements a more sophisticated key generation technique to improve security. Afterwards, Lychev et. al. [5] performed an extensive investigation on the security of QUIC. They introduced the Quick Communication (QC) protocol to define the utilization of the initial session key before the final session key is created for the 0-RTT handshake. The QACCE (Quick Authenticated And Confidential Channel Establishment) model is used to demonstrate the security of the QUIC connection formation procedure and the Data Encryption Transmission mechanism. To address the forward security issue of QUIC's 0-RTT Handshake, Jager et. al. [31] proposed the Bleichenbacher Attack to quickly guess the server's secret key in TLS1.2, compromising the security of the protocol [32]. By adopting PKCS#1 v1.5, QUIC avoided the security flaws in TLS1.2 and is no longer vulnerable to similar attacks.

C. Security Threats of QUIC

Although QUIC assures the data communication security, attackers can still obstruct regular communication between two parties. This subsection goes over different types of security threats of QUIC protocol.

1) QUIC Reflection DDoS attack: an attacker can launch a QUIC flood Distributed Denial-of-Service (DDoS) attack [17] to overwhelm the targeted server via QUIC. When the attack traffic volume is high, the victim service slows down and impacts authorized users. As QUIC is based on UDP, which provides little sender's information to the receiver. As a result, DDoS attacks through QUIC are challenging issues. QUIC protocol is especially vulnerable to Reflection-based DDoS attacks. A QUIC Reflection DDoS attack involves spoofing the victim's IP address and send queries to many servers. Responses by the servers go to the victim rather than the offender. As QUIC was created in conjunction with TCP and TLS encryption, the first reply message contains its TLS certificate and is significantly larger than the query message from the client, which makes it possible for attackers to utilize third-party servers to send a huge amount of unwanted traffic to an victim.

2) Handshake Denial of Service: QUIC offers authenticated and Encrypted transport [13] to remove spoofed traffic. The majority of unauthenticated packets are often discarded by QUIC endpoints through handshakes, which prevents attackers from tampering active connections. QUIC endpoints may accept some unauthenticated ICMP packets after a connection has been made. However, their impact is limited. An alternative packet type that an endpoint may acknowledge is a stateless reset, which depends on the token's confidentiality. QUIC only offers defense

against attacks coming from outside the network where a connection is established. There is a proof that the recipient saw a previous packet from its peer is included in every QUIC packet. However, The available defenses aren't meant to be useful against an attacker who can catch QUIC packets before the connection is made.

QUIC is susceptible to a number of security threats, according to [5]. These attacks are separated into two categories: online attacks and offline attacks depending on whether the attacker is on the network path connecting the client and the server.

3) Online Attacks: As mentioned by Li, et al. [11], an attacker can make both parties (client and server) in the communication believe that the connection has been successful for a long period of time by tampering with the connection ID used by the client during the handshake process. However, they are unable to analyze the received data normally. As a result, the connection drop is taken place. The attacker can also tamper with the source-address token [11] in a manner similar to the connection ID tampering attack, preventing both parties from parsing the data packets that the other side has received. The connection is seen as successful within the first a few seconds before being aggressively cut off.

4) Offline Attacks: Li, et al. [11] described a server configuration repeated attack, which is similar to TCP reset injection [33]. In this attack, the attacker sniffs the server's configuration information and use that knowledge together with a cloak of the server's IP address to transmit reset packet to the client, which resets the QUIC connection. Iyengar, et al. [13] also described that a stateless resets can lead to a DoS attack. This attack is available if an attacker can make a connection with a certain connection ID with a stateless reset token. An attacker who produces this token can reset a open connection with the same connection ID.

QUIC is unable to offer an efficient way to stop both online and offline attackers from disrupting a QUIC connection. It can cause both parties' connections to go inactive for a while, which can be used by online attackers to delay detection. Both communication parties are able to identify it. It takes little connection-related knowledge for an attacker to break a QUIC connection.

5) Reflective Amplification Attack and State Exhaustion Attack: In 2021, Nawrocki, et al. [17] described a security loophole in QUIC's Handshake protocol. During the first round-trip, the server responds to an unverified source. As a result, the attacker can easily establish Reflective Amplification Attack [17] and State Exhaustion Attack [34]. The responding to unverified source issue is a vital security weakness of QUIC protocol.

6) Spoofed ACK Attack: Iyengar, et al. [13] presented a severe security loophole of QUIC where an attacker may get an address validation token from the server and subsequently divulge the IP address used to get the token. The attacker may spoof this IP address to connect to a server using a 0-RTT connection disguised as the victim. The server will then transfer an excessive quantity of data to the IP address, which allows the attacker to spoof ACK frames to the server.

7) Stream Fragmentation and Reassembly Attacks: As Iyengar, et al. mentioned in [13], to generate excessive receive buffer memory commitment and/or the formation of a big, inefficient data structure, an adversarial sender may purposefully broadcast stream data fragments. In order to force the sender to hold the unacknowledged stream data for re-transmission, an adversarial receiver may purposefully fail to acknowledge packets carrying stream data.

8) Cache Poisoning Attacks: It is a cyber attack in which perpetrators inject false data into a web cache or the DNS cache with the intent of damaging users [35]. Cache poisoning attacks against HTTP-based implementations,like QUIC, are immensely troublesome [36].

9) Slowloris Attacks: Slowloris attacks [13] can be carried out against a QUIC endpoint by producing the bare minimum of activities required to prevent it from being shut down for inactivity. They aim to maintain as many connections open as possible to the target destination, by sending sparse quantities of data, progressively opening flow control windows to regulate the sender rate, or creating ACK frames that imitate a high loss rate.

10) Explicit Congestion Notification Attacks: Another major security threat for QUIC protocol is explicit congestion notification Attacks [13]. In order to affect the sender's rate, an on-path attacker can change the value of Explicit Congestion Notification (ECN) code points in the IP header. To alter the sender's rate, an on-the-side attacker can copy and transmit packets with altered ECN codepoints. An off-path attacker will need to race the duplicate packet against the original in order to succeed in this attack if a recipient discards duplicate packets.

11) Optimistic ACK Attack: In an optimistic ACK attack [13], a congestion controller could allow transmission at rates that are higher than what the network can handle because an endpoint recognizes packets it has not received. In order to identify this behavior, an endpoint can omit packet numbers while transmitting packets. Once this happens, an endpoint has the option to instantly terminate the connection with a PROTOCOL_VIOLATION connection error [37].

12) Firewall Negligence Issue: Firewalls often offer a variety of options when handling HTTP/HTTPS traffic [38]. When web traffic is discovered by a firewall, it often goes via a web protection module that runs extensive

checks using deep packet inspection and web filtering. Firewalls, these days, can provide a lot of information, including enhanced reporting and malware scanning. However, the majority firewalls do not recognize QUIC traffic as web traffic [2]. QUIC packets do not receive the same amount of inspection and logging. This raises serious security issues with consequences such as not being able to limit access to websites or turning on safe search on Google.

13) Recent Explored Attacks: Chatzoglou, et al. [43] categorized overall security related attacks on QUIC into five types: Cryptographic Attacks, Handshake Attacks, Privacy Attacks, Fuzzing Attacks, and Transport Layer Attacks. They found some new issues of QUIC protocol after deploying QUIC, including QUIC-downgrade, QUIC-out-of-joint, QUIC-fuzz, QUIC-loris, QUIC-flooding and QUIC-encapsulation. They mentioned a future potential research challenge of QUIC is "QUIC-focused fuzzer". To find setup errors in the several QUIC implementations, a stateful QUIC fuzzer can be helpful. Table I shows an overall taxonomy on QUIC attacks.

IV. Discussion and Future Work

Although some research work has been done on QUIC security, there are still room for advancement in the current body of scientific research regarding QUIC security. In this section, we discuss our vision on future work that can be done on QUIC security.

1) A Comprehensive Study on How Resistant QUIC is against IP Spoofing and Flooding Attacks: No research has yet focused on the QUIC's resistance to IP Spoofing and Flooding Attacks. Although address validation protection is implemented by QUIC, it should be further investigated to see if this protection is functional against all QUIC implementations or not. In addition, we could do comparison among available protection mechanisms of QUIC on a User Datagram Protocol (UDP) based Amplification Attack. In addition, We could propose feasible countermeasures that can be adopted by QUIC to improve its robustness.

2) Balancing Security with Performance: The forward-secure 0-RTT handshake [29] has a high performance cost, while 0-RTT handshake used by QUIC cannot ensure forward security. On the contrary, in order to obtain stronger security than the current security standards, QUIC uses TLS1.3. As a result, the processing demands of QUIC on the CPU have significantly increased due to the Encryption and Decryption burden imposed on by QUIC. Thus, it is important to investigate and explore how to balance security and computation overhead.

3) The Competition between QUIC and TCP as well as the Prediction of the Future of QUIC: The trend of increasing QUIC traffic has left TCP less and less bandwidth to use. However, will it completely take over TCP in bandwidth competition? We could answer this question by investigating the competition between QUIC and TCP when the bandwidth shares changes. Through studying the cost and benefit of adopting QUIC, we can make a prediction on the future of QUIC.

4) Cache Poisoning Attacks against QUIC: Cache poisoning attacks (DNS, web and so on) against HTTP-based implementations can be very problematic. However, no study has yet looked at similar attacks against QUIC. Future research can study cache poisoning threats in such infrastructures because QUIC is implemented in many proxies and load balancers.

5) Guarantee Mechanisms to QUIC Connections: As we described in the last section, QUIC connections are accessible by online or offline attackers. To better identify malicious attacker's actions and enhance connection security, we can investigate how to improve the protocol to provide additional guarantee mechanisms to QUIC connections.

V. Conclusion

QUIC is a new transport layer protocol appeared after 2012, which is built on top of UDP with several improvements over TCP to address its performance issues. It is designed to achieves lower latency and higher efficiency than TCP. In addition, QUIC offers improved privacy and higher performance in demanding network environments. In the past a few years, there have been much research done to address the security issues of QUIC protocol. However, there is a lack of comprehensive survey that focuses on QUIC security. In this paper, we aim to fulfil the gap and present a comprehensive survey on QUIC Security. We expect this effort to serve as a foundation and source of references for more research in the related field.

References

[1] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '17. New York, NY, USA: Association for Computing Machinery, Aug. 2017, pp. 183–196. [Online]. Available: https://doi.org/10.1145/3098822.3098842

[2] P. N. N. G, N. Dey, N. N, M. Hariprasad, S. S, M. Moharir, and M. Akram, "A Detail Survey on QUIC and its Impact on Network Data Transmission," in *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*. Tirunelveli, India: IEEE, Apr. 2022, pp. 378–385. [Online]. Available: https://ieeexplore.ieee.org/document/9777199/

[3] P. Kumar, "QUIC (Quick UDP Internet Connections) – A Quick Study," Oct. 2020, arXiv:2010.03059 [cs]. [Online]. Available: http://arxiv.org/abs/2010.03059

TABLE I: Categorization of Attacks on QUIC

Major Types of Attacks	Subtypes of Attacks and Research Works
Online	Connection ID Tampering [11], Source-Address Token Tampering [11]
Offline	Server Configuration Repeated [11] or State Reset Oracle [13], Crypto Stream Offset [11]
Handshake	Packet Manipulation [5], Downgrade [5], [39], Crypto Stream Offset [5], Replay [5], [40], QUIC RST [21], DoS [5], [21], [41], [17], Version forgery [21], [42], Packet Length Manipulation [42], Missing parameters [42], Frame Mangling [42], State Overflow [17], Reflective Amplification [17], QUIC-downgrade [43], QUIC-out-of-joint [43]
Cryptographic	Decryption [31], Drown [32], Client Impersonation [44], Selfie [45], Nonce Reuse/Misuse [46]
Fuzzing	Information Leak [47], Implementation Vulnerabilities [48], Impersonation [49], Enumeration [50], QUIC-fuzz [43], QUIC-flooding [43] QUIC-loris [43]
Transport Layer	UDP Hole Punching [51], QUIC-encapsulation [43]
Privacy	Traffic Analysis [52], [19], Website Fingerprinting [19], [20], [18], Session Linking [53]
Others	Spoofed ACK [13], Optimistic ACK [13], Slowloris [13], Steam Fragmentation and Reassembly [13], Stream Commitment [13], Explicit Congestion Notification [13], Cache Poisoning Attacks [35], [36]

[4] R. Marx, T. De Decker, P. Quax, and W. Lamotte, "Resource Multiplexing and Prioritization in HTTP/2 over TCP Versus HTTP/3 over QUIC," in *Web Information Systems and Technologies*, ser. Lecture Notes in Business Information Processing, A. Bozzon, F. J. Domínguez Mayo, and J. Filipe, Eds. Cham: Springer International Publishing, 2020, pp. 96–126.

[5] R. Lychev, S. Jero, A. Boldyreva, and C. Nita-Rotaru, "How Secure and Quick is QUIC? Provable Security and Performance Analyses," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 214–231, iSSN: 2375-1207.

[6] "Experimenting with QUIC." [Online]. Available: https://blog.chromium.org/2013/06/experimenting-with-quic.html

[7] "QUIC," Jul. 2022, page Version ID: 1097684976. [Online]. Available: https://en.wikipedia.org/w/index.php?title=QUIC&oldid=1097684976

[8] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," Internet Engineering Task Force, Request for Comments RFC 9000, May 2021. [Online]. Available: https://datatracker.ietf.org/doc/rfc9000/

[9] M. Thomson and S. Turner, "Using TLS to Secure QUIC," Internet Engineering Task Force, Request for Comments RFC 9001, May 2021. [Online]. Available: https://datatracker.ietf.org/doc/rfc9001/

[10] M. Bishop, "HTTP/3," Internet Engineering Task Force, Request for Comments RFC 9114, Jun. 2022. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-http/34/

[11] L. Xuebing, C. Yang, Z. Mengying, and W. Xin, "Internet Data Transfer Protocol QUIC: A Survey," *Journal of Computer Research and Development*, vol. 57, no. 9, p. 1864, Sep. 2020. [Online]. Available: https://crad.ict.ac.cn/EN/10.7544/issn1000-1239.2020.20190693

[12] S. Chen, S. Jero, M. Jagielski, A. Boldyreva, and C. Nita-Rotaru, "Secure Communication Channel Establishment: TLS 1.3 (over TCP Fast Open) versus QUIC," *Journal of Cryptology*, vol. 34, no. 3, p. 26, May 2021. [Online]. Available: https://doi.org/10.1007/s00145-021-09389-w

[13] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," Jul. 2022. [Online]. Available: https://greenbytes.de/tech/webdav/draft-ietf-quic-transport-16.html#handshake-denial-of-service

[14] "Usage Statistics of QUIC for Websites, July 2022." [Online]. Available: https://w3techs.com/technologies/details/ce-quic

[15] "Usage Statistics of HTTP/3 for Websites, July 2022." [Online]. Available: https://w3techs.com/technologies/details/ce-http3

[16] "How Facebook is bringing QUIC to billions," Oct. 2020. [Online]. Available: https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/

[17] M. Nawrocki, R. Hiesgen, T. C. Schmidt, and M. Wählisch, "QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events," in *Proceedings of the 21st ACM Internet Measurement Conference*, Nov. 2021, pp. 283–291, arXiv:2109.01106 [cs]. [Online]. Available: http://arxiv.org/abs/2109.01106

[18] L. Barman, S. Siby, C. Wood, M. Fayed, N. Sullivan, and C. Troncoso, "This is not the padding you are looking for! On the ineffectiveness of QUIC PADDING against website fingerprinting," arXiv, Tech. Rep. arXiv:2203.07806, Mar. 2022, arXiv:2203.07806 [cs] type: article. [Online]. Available: http://arxiv.org/abs/2203.07806

[19] P. Zhan, L. Wang, and Y. Tang, "Website fingerprinting on early QUIC traffic," *Computer Networks*, vol. 200, p. 108538, Dec. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621004618

[20] Y. Govil, L. Wang, and J. Rexford, "{MIMIQ}: Masking {IPs} with Migration in {QUIC}," 2020. [Online]. Available: https://www.usenix.org/conference/foci20/presentation/govil

[21] X. Cao, S. Zhao, and Y. Zhang, "0-RTT Attack and Defense of QUIC Protocol," in *2019 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[22] "QUIC: Design Document and Specification Rationale." [Online]. Available: https://docs.google.com/document/d/1RNHkx_VvKWyWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34/edit?usp=embed_facebook

[23] M. Scharf and S. Kiesel, "NXG03-5: Head-of-line Blocking in TCP and SCTP: Analysis and Measurements," in *IEEE Globecom 2006*, Nov. 2006, pp. 1–5, iSSN: 1930-529X.

[24] F. Qian, V. Gopalakrishnan, E. Halepovic, S. Sen, and O. Spatscheck, "TM3: 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015," *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015*, Dec. 2015. [Online]. Available: http://www.scopus.com/inward/record.url?scp=84994161453&partnerID=8YFLogxK

[25] "SMig: Stream Migration Extension For HTTP/2," Jan. 2017. [Online]. Available: https://cse.buffalo.edu/faculty/xmi/publication/conext16_http2/

[26] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," Internet Engineering Task Force, Request for Comments RFC 8446, Aug. 2018. [Online]. Available: https://datatracker.ietf.org/doc/rfc8446/

[27] S. R. Das, "Evaluation of QUIC on web page performance," Thesis, Massachusetts Institute of Technology, 2014. [Online]. Available: https://dspace.mit.edu/handle/1721.1/91444

[28] M. Bellare and B. Yee, "Forward-Security in Private-Key Cryptography," in *Topics in Cryptology — CT-RSA 2003*, ser. Lecture Notes in Computer Science, M. Joye, Ed. Berlin, Heidelberg: Springer, 2003, pp. 1–18.

[29] F. Günther, B. Hale, T. Jager, and S. Lauer, "0-RTT Key Exchange with Full Forward Secrecy," in *Advances in Cryptology – EUROCRYPT 2017*, ser. Lecture Notes in Computer Science, J.-

S. Coron and J. B. Nielsen, Eds. Cham: Springer International Publishing, 2017, pp. 519–548.

[30] M. Fischlin and F. Günther, "Multi-Stage Key Exchange and the Case of Google's QUIC Protocol," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 1193–1204. [Online]. Available: https://doi.org/10.1145/2660267.2660308

[31] T. Jager, J. Schwenk, and J. Somorovsky, "On the Security of TLS 1.3 and QUIC Against Weaknesses in PKCS#1 v1.5 Encryption," *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, 2015. [Online]. Available: https://ris.uni-paderborn.de/record/3121

[32] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni, E. Käsper, S. Cohney, S. Engels, C. Paar, and Y. Shavitt, "{DROWN}: Breaking {TLS} Using {SSLv2}," 2016, pp. 689–706. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/aviram

[33] N. C. Weaver, "TCP Reset Injection," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 1282–1283. [Online]. Available: https://doi.org/10.1007/978-1-4419-5906-5_119

[34] X. Wang, "Memory and State Exhaustion Denial of Service," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 773–774. [Online]. Available: https://doi.org/10.1007/978-1-4419-5906-5_270

[35] "What is cache poisoning and how does it work?" [Online]. Available: https://www.techtarget.com/searchsecurity/definition/cache-poisoning

[36] K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan, "DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. Virtual Event USA: ACM, Oct. 2020, pp. 1337–1350. [Online]. Available: https://dl.acm.org/doi/10.1145/3372297.3417280

[37] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," Internet Engineering Task Force, Internet Draft draft-ietf-quic-transport-19. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/19/

[38] W. M. Shbair, T. Cholez, J. Francois, and I. Chrisment, "A Survey of HTTPS Traffic and Services Identification Approaches," Aug. 2020, arXiv:2008.08339 [cs]. [Online]. Available: http://arxiv.org/abs/2008.08339

[39] S. Lee, Y. Shin, and J. Hur, "Return of version downgrade attack in the era of TLS 1.3," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 157–168. [Online]. Available: https://doi.org/10.1145/3386367.3431310

[40] M. Fischlin and F. Günther, "Replay Attacks on Zero Round-Trip Time: The Case of the TLS 1.3 Handshake Candidates," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, Apr. 2017, pp. 60–75.

[41] A. Saverimoutou, B. Mathieu, and S. Vaton, "Which secure transport protocol for a reliable HTTP/2-based web service: TLS or QUIC?" in *2017 IEEE Symposium on Computers and Communications (ISCC)*, Jul. 2017, pp. 879–884.

[42] E. Gagliardi and O. Levillain, "Analysis of QUIC session establishment and its implementations," in *13th IFIP International Conference on Information Security Theory and Practice (WISTP)*, ser. Information Security Theory and Practice, M. Laurent and T. Giannetsos, Eds., vol. LNCS-12024. Paris, France: Springer International Publishing, Dec. 2019, pp. 169–184. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02468596

[43] E. Chatzoglou, V. Kouliaridis, G. Karopoulos, and G. Kambourakis, "Revisiting QUIC attacks: A comprehensive review on QUIC security and a hands-on study," In Review, preprint, Jul. 2022. [Online]. Available: https://www.researchsquare.com/article/rs-1676730/v1

[44] C. Cremers, M. Horvat, S. Scott, and T. van der Merwe, "Automated Analysis and Verification of TLS 1.3: 0-RTT, Resumption and Delayed Authentication," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 470–485, iSSN: 2375-1207.

[45] N. Drucker and S. Gueron, "Selfie: reflections on TLS 1.3 with PSK," *Journal of Cryptology*, vol. 34, no. 3, p. 27, May 2021. [Online]. Available: https://doi.org/10.1007/s00145-021-09387-y

[46] B. Arunkumar and G. Kousalya, "Nonce reuse/misuse resistance authentication encryption schemes for modern TLS cipher suites and QUIC based web servers," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 6483–6493, Jan. 2020. [Online]. Available: https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs179729

[47] K. L. McMillan and L. D. Zuck, "Formal specification and testing of QUIC," in *Proceedings of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '19. New York, NY, USA: Association for Computing Machinery, Aug. 2019, pp. 227–240. [Online]. Available: https://doi.org/10.1145/3341302.3342087

[48] G. S. Reen and C. Rossow, "DPIFuzz: A Differential Fuzzing Framework to Detect DPI Elusion Strategies for QUIC," in *Annual Computer Security Applications Conference*, ser. ACSAC '20. New York, NY, USA: Association for Computing Machinery, Dec. 2020, pp. 332–344. [Online]. Available: https://doi.org/10.1145/3427228.3427662

[49] J. Zhang, X. Gao, L. Yang, T. Feng, D. Li, and Q. Wang, "A Systematic Approach to Formal Analysis of QUIC Handshake Protocol Using Symbolic Model Checking," *Security and Communication Networks*, vol. 2021, p. e1630223, Aug. 2021. [Online]. Available: https://www.hindawi.com/journals/scn/2021/1630223/

[50] K. Thimmaraju and B. Scheuermann, "Count Me If You Can: Enumerating QUIC Servers Behind Load Balancers," *Electronic Communications of the EASST*, vol. 80, no. 0, Sep. 2021. [Online]. Available: https://journal.ub.tu-berlin.de/eceasst/article/view/1172

[51] K. Y. Gbur and F. Tschorsch, "A QUIC(K) Way Through Your Firewall?" arXiv, Tech. Rep. arXiv:2107.05939, Jul. 2021, arXiv:2107.05939 [cs] type: article. [Online]. Available: http://arxiv.org/abs/2107.05939

[52] V. Tong, H. A. Tran, S. Souihi, and A. Mellouk, "A Novel QUIC Traffic Classifier Based on Convolutional Neural Networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*. Abu Dhabi, United Arab Emirates: IEEE Press, Dec. 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/GLOCOM.2018.8647128

[53] G. Arfaoui, X. Bultel, P.-A. Fouque, A. Nedelcu, and C. Onete, "The privacy of the TLS 1.3 protocol," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, pp. 190 – 210, 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02482253