electronics

MDPI

Review

Challenges and Advances in Analyzing TLS 1.3-Encrypted Traffic: A Comprehensive Survey

Jiuxing Zhou, Wei Fu, Wei Hu, Zhihong Sun *, Tao He and Zhihong Zhang

Department of Information Security, Naval University of Engineering, Wuhan 430030, China
* Correspondence: zhihong.sun@whu.edu.cn

Abstract: The widespread adoption of encrypted communication protocols has significantly enhanced network security and user privacy, simultaneously elevating the importance of encrypted traffic analysis across various domains, including network anomaly detection. The Transport Layer Security (TLS) 1.3 protocol, introduced in 2018, has gained rapid popularity due to its enhanced security features and improved performance. However, TLS 1.3's security enhancements, such as encrypting more of the handshake process, present unprecedented challenges for encrypted traffic analysis, rendering traditional methods designed for TLS 1.2 and earlier versions ineffective and necessitating the development of novel analytical techniques. This comprehensive survey provides a thorough review of the latest advancements in TLS 1.3 traffic analysis. First, we examine the impact of TLS 1.3's new features, including Encrypted ClientHello (ECH), 0-RTT session resumption, and Perfect Forward Secrecy (PFS), on existing traffic analysis techniques. We then present a systematic overview of state-of-the-art methods for analyzing TLS 1.3 traffic, encompassing middlebox-based interception, searchable encryption, and machine learning-based approaches. For each method, we provide a critical analysis of its advantages, limitations, and applicable scenarios. Furthermore, we compile and review key datasets utilized in machine learning-based TLS 1.3 traffic analysis research. Finally, we discuss the main challenges and potential future research directions for TLS 1.3 traffic analysis. Given that TLS 1.3 is still in the early stages of widespread deployment, research in this field remains nascent. This survey aims to provide researchers and practitioners with a comprehensive reference, facilitating the development of more effective TLS 1.3 traffic analysis techniques that balance network security requirements with user privacy protection.

Keywords: TLS 1.3; encrypted traffic analysis; machine learning; interception techniques; searchable encryption

check for updates

Citation: Zhou, J.; Fu, W.; Hu, W.; Sun, Z.; He, T.; Zhang, Z. Challenges and Advances in Analyzing TLS 1.3-Encrypted Traffic: A Comprehensive Survey. *Electronics* **2024**, *13*, 4000. https://doi.org/10.3390/electronics13204000

Academic Editors: Yawen Chen and Fei (Travis) Dai

Received: 5 September 2024
Revised: 6 October 2024
Accepted: 9 October 2024
Published: 11 October 2024

1. Introduction

With the rapid development of the Internet, encrypted communication protocols have become crucial in safeguarding network communication security. According to Google's Transparency Report from April 2024 [1], 94% of traffic across all Google products and services, as well as web pages loaded via HTTPS in the Chrome browser (Windows system), is encrypted. Since its inception in 1999, the Transport Layer Security (TLS) protocol [2], the most widely used secure transmission protocol, has undergone multiple iterations and improvements. Notably, TLS 1.3, released in August 2018 [3], with enhanced security, higher performance, and greater flexibility, has garnered widespread attention and rapid adoption in both academia and the industry. According to the latest statistics from Qualys SSL Labs [4], as of May 2024, 70.1% of websites surveyed by SSL Labs support TLS 1.3, and this proportion is continuing to rise. It is foreseeable that TLS 1.3 will become the de facto standard for Internet communication encryption in the coming years.

However, the widespread adoption of TLS 1.3 presents new challenges for network traffic analysis. Compared to its predecessor, TLS 1.2 [5], TLS 1.3 introduces significant protocol design improvements and new features, making traditional traffic analysis methods based on TLS 1.2 and earlier versions difficult to apply directly. For instance, existing machine learning-based TLS 1.2 traffic classification methods primarily rely on plaintext features from the handshake phase, such as packet length and sequence patterns. Due to TLS 1.3's reduction in handshake round trips, the observed packet length sequences and traffic characteristics during the handshake phase have changed. This change leads to a decrease in the effectiveness of classification methods based on packet length sequences when distinguishing between different sessions or applications, necessitating adjustments to accommodate new patterns in TLS 1.3 traffic, as demonstrated by recent studies [6]. Additionally, TLS 1.3 enhances privacy protection by strengthening encryption during the handshake process, significantly reducing the plaintext information available during this phase. The Encrypted Server Name Indication (Encrypted SNI) extension further limits traffic analysis visibility, objectively restricting the effectiveness of traditional traffic analysis methods. Recent research [7] indicates that, to maintain high classification performance in TLS 1.3, existing models require corresponding adjustments. This includes re-selecting and extracting new features applicable to TLS 1.3, such as changes in traffic patterns caused by 0-RTT session resumption, and enhancing the feature extraction capabilities of models using time series analysis and deep learning techniques. Moreover, retraining models to encompass TLS 1.3-specific traffic characteristics is also a crucial step in improving classification performance.

In light of the pervasive adoption of TLS 1.3 and its significant impact on traffic characteristics, it is crucial to assess and improve the applicability of existing TLS 1.2 traffic analysis methods in the TLS 1.3 environment. On the one hand, security threats such as malware and botnets are actively utilizing TLS 1.3 to evade detection, making the analysis and identification of their traffic behavior urgent. On the other hand, network operators and service providers still need to legally and compliantly identify and manage TLS 1.3 traffic to achieve network optimization and anomaly diagnosis. Additionally, in specific fields such as cybercrime investigation and national security, there remains a need for the compliant analysis of TLS 1.3 traffic. Therefore, researching new traffic analysis methods and techniques for TLS 1.3 is not only a frontier topic in academic research but also has broad application prospects and practical relevance.

In response to the challenges posed by TLS 1.3, the academic community has conducted extensive and in-depth research. This paper not only deeply analyzes the new features introduced by TLS 1.3 and their impact on traffic analysis but also systematically reviews the latest achievements in TLS 1.3 traffic analysis to support future research.

1.1. Data Sources and Methodology of Study

In this survey, we systematically reviewed literature relevant to TLS 1.3 traffic analysis. We conducted comprehensive searches across major academic databases such as IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and Google Scholar. Our search employed a range of keywords including "TLS", "TLS 1.3", "TLS-1.3", "network traffic", "encrypted traffic", "traffic analysis", "traffic classification", "machine learning", "middleboxes", and "searchable encryption". Although our primary focus was on publications from 2018 onwards—postdating the release of TLS 1.3—we also included seminal works from earlier years that remain relevant. Recognizing that some papers do not explicitly mention "TLS 1.3" but offer applicable methodologies (e.g., searchable encryption), we included these studies in our review. Additionally, we explored Internet standards repositories for relevant technical standards related to TLS.

In total, we collected 223 documents comprising papers, websites, technical standards, and other resources. After filtering out those not directly relevant to our study's focus, approximately 120 references were incorporated into our paper. For a detailed evaluation of TLS 1.3 traffic analysis techniques' progress, we prioritized high-impact papers published in leading conferences and journals, ultimately selecting 37 papers for in-depth assessment.

By detailing our data sources and methodology, we aim to provide transparency in our research process and facilitate reproducibility for future studies.

1.2. Differences from Existing Surveys

As early as 2015, Velan et al. [8] conducted a systematic review of methods for encrypted traffic analysis. Since then, extensive research has been conducted in this field by scholars. In this survey, we present existing research on encrypted traffic analysis, with a primary focus on TLS 1.3 traffic analysis techniques.

Table 1 compares this study with other surveys on encrypted traffic analysis. Recent studies have provided comprehensive surveys on encrypted traffic analysis [9–12], covering techniques and applications such as machine learning and deep learning. Some researchers have concentrated on the application of deep learning in anomaly detection [13] and network attack detection [14]. With the emergence of new network architectures and application scenarios, researchers have started investigating encrypted traffic analysis in fields such as mobile applications [15–17], the Internet of Things (IoT) [18], and Software-Defined Networking (SDN) [19]. Poh et al. [20] focused on privacy protection issues and surveyed techniques for privacy-preserving encrypted traffic inspection. In the field of TLS-encrypted traffic analysis, Oh et al. [21] investigated various techniques for detecting TLS 1.2-encrypted malicious traffic in Security Operations Center (SOC) scenarios. Although de Carnavalet et al. [22] studied TLS 1.3 and earlier versions, their work primarily focused on surveying and analyzing interception mechanisms and motivations, lacking investigation into other traffic analysis techniques. This paper aims to comprehensively review the research progress in TLS 1.3 traffic analysis techniques, discuss the applicability and limitations of existing methods, and explore future research directions.

Table 1. Differences between existing encrypted traffic analysis surveys and this survey.

Survey	Year	Encryption Protocol	Domain	Contributions
[8]	2015	Various	ETC	Summarized methods for ETC and analysis
[9]	2019	Various	ETC	Overviewed the application of DL in ETC tasks
[10]	2019	Various	NTC	Systematically reviewed the process of using ML techniques for TC
[13]	2019	Various	NAD1	Summarized various methods for NAD1 using DL techniques
[15]	2019	Various	Mobile	Extensively surveyed the application of DL techniques in Mobile
[16]	2019	Various	Mobile ETC	Evaluated the performance of DL in Mobile ETC tasks through experiments
[19]	2019	Various	SDN network intrusion detection	Investigated the current state of research on intrusion detection using ML methods in SDNs
[17]	2020	Various	Mobile ETC	Proposed a general framework for evaluating the effectiveness of DL in mobile ETC
[18]	2020	Various	IoT-TC	Reviewed various techniques and methods for IoT-TC
[11]	2021	Various	ETA	Comprehensive review of ETA research progress from application, technology, and countermeasure perspectives
[20]	2021	Various	Privacy protection	Investigated various privacy-preserving techniques for ETA over network middleboxes
[12]	2022	Various	ETA	Thoroughly reviewed various methods for ETA using ML techniques

Table 1. *Cont.*

Survey	Year	Encryption Protocol	Domain	Contributions
[21]	2022	TLS 1.2	Malicious ETA	Specifically surveyed various analysis techniques for detecting TLS malicious traffic in SOC scenarios
[14]	2023	Various	NAD2	Reviewed and analyzed DL-based NAD2 techniques
[22]	2023	TLS	Interception Technology	Discusses the implementation methods and underlying motivations of various TLS interception mechanisms
Ours	2024	TLS 1.3	ETA	Surveyed the latest advancements in ML-based TLS 1.3 traffic analysis techniques

Note 1: NTC denotes Network Traffic Classification. ETA denotes encrypted traffic analysis. ETC denotes encrypted traffic classification. NAD1 denotes network anomaly detection. NAD2 denotes network attack detection. Mobile denotes Mobile and Wireless Networks. TC represents traffic classification. ML represents machine learning and DL represents deep learning. Note 2: In this survey, we distinguish between ETA and ETC. ETA encompasses a wide range of techniques and applications for analyzing encrypted traffic, including traffic characterization, anomaly detection, and performance analysis. ETC, on the other hand, is a subset of ETA that specifically deals with categorizing encrypted traffic into predefined classes, such as application types or protocols.

1.3. Research Objectives and Contributions of This Paper

This paper aims to address the significant challenges posed by the adoption of TLS 1.3, focusing on several key research questions.

- What improvements does TLS 1.3 have compared to previous versions? What challenges do these changes pose to traditional traffic analysis methods?
  This question aims to identify specific features of TLS 1.3 that complicate existing analysis techniques and require new approaches.
- What are the main categories of current TLS 1.3 traffic analysis techniques? What are the latest advancements in these methods? How applicable and limited are these methods?
  By exploring this question, we aim to provide a comprehensive overview of current methodologies and their effectiveness in handling TLS 1.3 traffic.
- In studies using machine learning methods to analyze TLS 1.3 traffic, what TLS 1.3 datasets are currently available? What is the quality of these datasets?
  This question aims to summarize the main datasets used in the field of TLS 1.3 traffic analysis, analyze their importance in research, and point out the deficiencies of current datasets.
- What challenges do current TLS 1.3 traffic analysis techniques face? What are the future research directions?
  This question seeks to highlight gaps in current research and propose potential areas for future investigation.

By addressing these questions, this paper not only reviews existing literature but also provides a roadmap for future research in analyzing TLS 1.3-encrypted traffic.

The main contributions of this paper include the following:

- Firstly, this paper conducts a comprehensive survey of recent major techniques for TLS 1.3 traffic analysis, including middlebox-based interception techniques, searchable encryption techniques, machine learning-based traffic analysis methods, analyzing their advantages, disadvantages, and applicable scenarios. To our knowledge, this paper is the first study which specifically focuses on TLS 1.3 traffic analysis techniques.
- Secondly, this paper analyzes the impact of TLS 1.3 protocol changes on traffic analysis. We delve into the impact of new features in the TLS 1.3 protocol on traditional traffic analysis methods, such as encrypted ClientHello, 0-RTT session resumption, and PFS, highlighting the challenges posed by these changes.

- Moreover, we summarize the main datasets used in the field of TLS 1.3 traffic analysis, emphasize the importance of datasets in research and point out the current deficiencies in datasets.
- Finally, the existing issues and future directions of TLS 1.3 traffic analysis are analyzed in this survey.

1.4. Survey Organization

The organization of this paper is as follows. Section 1 outlines the research background, differences from existing surveys, and the main contributions of this study. Section 2 introduces the main application areas of TLS traffic analysis, including network security threat detection, network management and service quality assurance, user behavior analysis and privacy protection, and network censorship and forensics. Section 3 focuses on discussing the new features of TLS 1.3 and their impact on traffic analysis. We present the core part of this paper in Section 4, which comprehensively reviews the latest progress in TLS 1.3 traffic analysis techniques, including middlebox-based interception techniques, searchable encryption techniques, and machine learning-based analysis techniques. The key challenges in TLS 1.3 traffic analysis and explores future research directions are summarized in Section 5. Finally, a conclusion is drawn in Section 6. Through this organization, this paper aims to comprehensively and systematically introduce the latest research progress in the field of TLS 1.3 traffic analysis, analyze the advantages and disadvantages of existing methods, and identify directions for future research.

2. Applications of TLS Traffic Analysis

A significant portion of Internet traffic is now encrypted using the TLS protocol. Encrypted traffic analysis has broad applications in network security threat detection, network management and Quality of Service (QoS) assurance, user behavior analysis and privacy protection, and network censorship and forensics.

2.1. Network Security Threat Detection

Encrypted traffic analysis is widely used in network security threat detection. Many network attacks and malware utilize the TLS protocol to protect their communications, thereby concealing their malicious activities. However, detecting such attack traffic remains feasible, and machine learning methods have proven highly effective in identifying malicious TLS traffic. Researchers have proposed various methods for detecting malicious encrypted traffic [23–33], employing machine learning algorithms to analyze the behavior patterns of TLS-encrypted traffic and detect various security threats such as malware, botnets, and DDoS attacks. This is crucial for safeguarding critical information infrastructures in government, enterprises, and the financial sector. Additionally, using specific TLS configuration parameters to detect malware is a common method. For example, comparing the hash value of the ClientHello parameters during the TLS handshake (also known as TLS fingerprinting) with an internal database can detect malware [34].

2.2. Network Management and Quality of Service Assurance

With the widespread adoption of encryption protocols like TLS, the prevalence of encrypted traffic is increasing. Many studies analyze TLS-encrypted traffic to identify the applications or services to which the traffic belongs, such as distinguishing between video, gaming, and instant messaging applications. This helps Internet Service Providers (ISPs) or network managers understand the composition of traffic from different applications and services, facilitating tasks such as traffic engineering and bandwidth allocation to ensure the quality of critical services and improve user experience [35–39]. For instance, classification methods based on Server Name Indication (SNI) [40,41] are frequently used to provide QoS. Although encryption technologies like Encrypted Client Hello (ECH) may hinder SNI-based classification, research indicates that effective QoS-aware classification can still be achieved by analyzing the payload of the TLS handshake, even when SNI is hidden [42,43].

2.3. User Behavior Analysis and Privacy Protection

Although the TLS protocol aims to secure the content of data packets, encrypted traffic generated by different websites or applications still exhibits identifiable differences. These distinguishing features, such as packet length, direction, and sequence, can be used to infer the websites or applications a user is accessing, and even specific actions within an application. This analysis of user behavior and preferences is valuable for fields such as advertising and recommendation systems. However, excessive user behavior analysis can infringe on user privacy. Researchers have proposed privacy-preserving traffic analysis methods that focus on identifying potential information leaks from encrypted traffic, such as website fingerprinting [44–48], application fingerprinting [49–52], and user behavior identification [53–57]. These methods can detect and prevent behaviors that may infringe on user privacy, which is crucial for protecting user privacy. With the increasing number of privacy protection regulations, such as GDPR and CCPA, encrypted traffic analysis is also employed to ensure compliance with data processing activities.

2.4. Network Censorship and Forensics

As more Internet traffic is encrypted using protocols like TLS, various techniques are employed to censor TLS and extended HTTPS traffic, enabling censorship authorities to identify and filter illegal or inappropriate content in encrypted traffic, such as malware, spam, hate speech, terrorism-related content, and corporate policy compliance [58]. Since encryption limits the effectiveness of content-based filtering, many studies focus on identifying and intercepting traffic based on TLS handshake metadata [21,59–61]. For example, the SNI field can be used for censorship [62–66], and server certificates can be used to review HTTPS content [67]. Research on TLS interception techniques using middlebox network devices demonstrates that these devices can be used for detailed traffic analysis and monitoring within the network [68,69]. Machine learning algorithms are also commonly used in encrypted traffic censorship. By analyzing the TLS handshake process and traffic patterns, censorship authorities can identify specific services and applications and detect anomalies that may indicate security threats or prohibited content [11,21]. In the field of network forensics, censorship techniques can be used to extract evidence from TLS-encrypted traffic [70,71], investigate security incidents, data breaches, and cybercrimes, providing critical information for law enforcement.

In summary, these application areas illustrate the significant and multifaceted importance of TLS-encrypted traffic analysis in the contemporary network environment. Future research should prioritize the optimization of TLS-encrypted traffic while maintaining the confidentiality and security of the data.

3. New Features of TLS 1.3 and Their Impact on Traffic Analysis

3.1. Changes in TLS Protocol Versions

The Transport Layer Security (TLS) protocol is the cornerstone of secure Internet communication, evolving from the Secure Sockets Layer (SSL) protocol initially developed by Netscape. Its primary goal is to encrypt data transmitted over the Internet to prevent eavesdropping and tampering, thereby ensuring secure communication between web browsers and servers.

TLS 1.0, introduced in 1999 (RFC 2246 [2]), was an upgrade from SSL 3.0, addressing multiple security issues present in SSL 3.0. TLS 1.1, released in 2006, is documented in RFC 4346 [72]. TLS 1.2, published in 2008 (RFC 5246 [5]), introduced significant updates compared to TLS 1.1. As of today, TLS 1.2 remains the most widely used version, with 99.9% of surveyed websites supporting it [4]. Earlier versions, namely TLS 1.0 and TLS 1.1, were deprecated in March 2021 [73].

The latest version, TLS 1.3, released in 2018, is defined in RFC 8446 [3]. It represents a major overhaul of the TLS protocol, removing outdated encryption algorithms, shortening the handshake process to speed up connections, and implementing forward secrecy by default, thereby providing stronger security guarantees and better privacy. Unlike the slow adoption of previous TLS versions, TLS 1.3 has been rapidly deployed. Within 15 months of its standardization, approximately 20% of connections used TLS 1.3, and about 30% of popular domains had adopted it [74].

3.2. Impact of TLS 1.3 on Traffic Analysis

As the latest version of the secure transmission protocol, TLS 1.3 introduces several innovative features that enhance communication security and efficiency while presenting new challenges for traffic analysis. This section examines the main new features and their impacts.

(1) Simplified Handshake Process: In TLS 1.2 and earlier versions, completing a full handshake required two round-trip times (RTTs) (as shown in Figure 1). The first RTT involved negotiating encryption parameters through “ClientHello” and “ServerHello” messages, while the second RTT completed the key exchange. TLS 1.3 compresses the previous Hello negotiation process, reducing the handshake time to one RTT (as shown in Figure 2). The client includes all necessary key-sharing information (such as pre-shared keys or Elliptic Curve Diffie–Hellman key shares) along with a list of supported cipher suites in its initial ClientHello message. Upon receiving the ClientHello, the server selects a cipher suite and immediately sends the ServerHello, certificate, key exchange parameters, and all other messages in a single transmission. This design significantly reduces the number of round trips required for the handshake, thereby enhancing connection establishment speed [3,75]. This means that communication between the client and server is faster, especially when establishing new connections, thereby significantly enhancing performance and user experience. However, this simplified process also changes the observable characteristics of the TLS handshake, potentially complicating traffic classification methods that rely on analyzing handshake patterns.

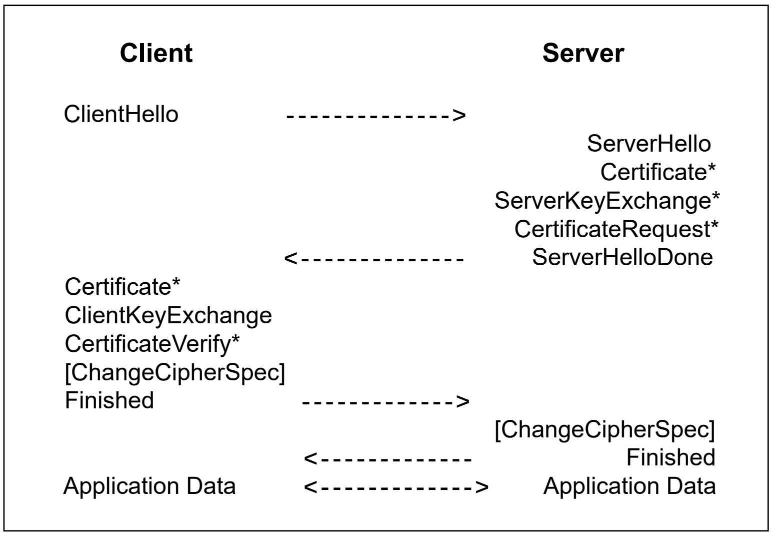

Figure 1. TLS 1.2 full handshake framework (from RFC 5246, * indicates a message that is optional).

Additionally, TLS 1.3 introduces the early_data extension, allowing for zero round-trip time (0-RTT) session resumption with previously visited websites. In TLS 1.3, if the client and server have previously established a connection and the server permits 0-RTT data, the client can send application data with the first handshake message. In contrast, TLS 1.2 and earlier versions required at least one full round-trip communication (1-RTT) to send application data. This necessitates redesigning session-based traffic classification methods, as the traffic characteristics under 0-RTT mode differ from those of traditional TLS handshake processes. Moreover, the 0-RTT mode introduces new scenarios for malicious traffic analysis, as attackers might exploit 0-RTT to conduct replay attacks.

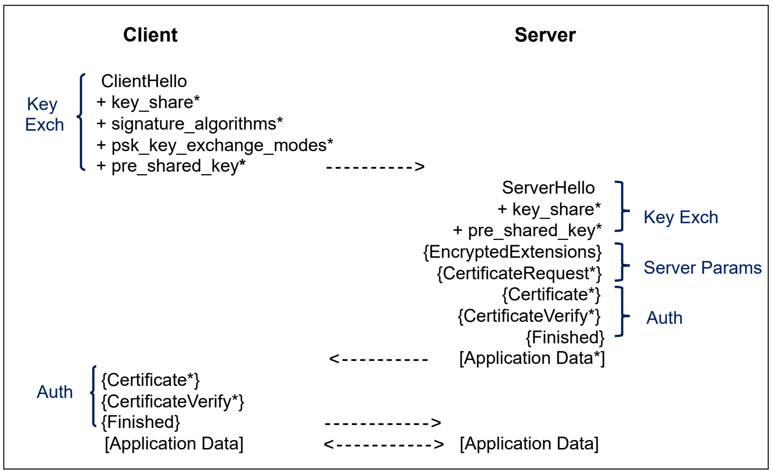

Figure 2. TLS 1.3 full handshake framework (from RFC 8446, + indicates noteworthy extensions sent in the previously noted message, * indicates a message that is optional).

(2) Enhanced Security: In TLS 1.2 and earlier versions, the RSA key exchange was a common method. In this mode, the client generates a pre-master secret (PMS), encrypts it with the server's RSA public key, and transmits it to the server. The server decrypts the PMS with its private key, and both parties derive session keys from the PMS for subsequent encryption of communication. Traditional TLS interception techniques could passively decrypt traffic if the middlebox possessed the server's RSA private key, thus enabling intrusion detection, content filtering, and other traffic analysis tasks [22]. However, TLS 1.3 makes significant changes by removing support for outdated and insecure encryption algorithms and mandating stronger encryption methods. It eliminates RSA-based key exchange, requiring all key exchanges to use forward-secure key agreement mechanisms such as Diffie–Hellman Ephemeral (DHE) or Elliptic Curve Diffie–Hellman Ephemeral (ECDHE). Perfect Forward Secrecy (PFS) ensures that, even if the server's key is compromised in the future, past communications cannot be decrypted, as each session utilizes independently generated ephemeral keys. These improvements in TLS 1.3 enhance security, particularly against key compromise and replay attacks. Even if a third party possesses the server's RSA private key or static DH private key, they are unable to decrypt TLS 1.3 communications. This means that traditional traffic analysis methods relying on passive decryption with the server's private key are no longer effective, posing challenges for middleboxes that need to perform specific network operations. Network operators, content providers, and security solution providers must reconsider their strategies and explore new methods to adapt to these changes.

(3) Encrypted ClientHello (ECH): TLS 1.3 introduces an extension called Encrypted ClientHello (ECH) [76], previously known as Encrypted Server Name Indication (ESNI). ECH is an extension to TLS 1.3 that is currently being standardized by the IETF TLS working group [77]. While not a standard component of TLS 1.3, and with a limited practical application in real-world networks at present, ECH represents a significant privacy-enhancing technology that warrants discussion due to its potential impact on traffic analysis. In traditional TLS handshakes, the client sends a ClientHello message to the server, which includes privacy-sensitive information such as Server Name Indication (SNI) and Application-Layer Protocol Negotiation (ALPN) in plaintext. Internet Service Providers (ISPs) often use SNI to identify specific applications and monitor network operations. ECH aims to encrypt most of the ClientHello content, including SNI, leaving only a minimal amount of necessary information in plaintext to facilitate the handshake. This enhancement, if widely adopted,

could significantly improve user privacy by preventing third parties from accessing the actual SNI information. However, it simultaneously poses challenges for traffic analysis based on service quality (QoS) classification and network monitoring. Many existing traffic classification methods, such as SNI-based classification [40,41], rely on unencrypted handshake metadata. The introduction of ECH represents a significant challenge in the field of network traffic analysis and management, requiring innovative approaches to maintain effective network monitoring and security in an increasingly privacy-focused Internet landscape.

In summary, TLS 1.3 introduces several new features, such as ECH, 0-RTT, and PFS, which enhance security and performance but present new challenges for traffic analysis. To address these challenges, researchers need to thoroughly analyze the TLS 1.3 protocol mechanisms, identify new traffic characteristics, and study new traffic analysis techniques.

4. TLS 1.3 Traffic Analysis Techniques

The new features of the TLS 1.3 protocol, such as the simplified handshake process and encrypted SNI, enhance communication security and efficiency, but they also present unprecedented challenges for traffic analysis. To address these challenges, researchers have devised various traffic analysis techniques. Generally, existing TLS 1.3 traffic analysis techniques can be categorized into two types: active detection and passive detection.

(1) Active Detection Techniques: These techniques primarily utilize middlebox devices to perform decryption or partial decryption of the original encrypted traffic, thereby making the payload information or keywords visible. This enables the use of deep packet inspection (DPI) or rule-based matching to identify malicious traffic. Common solutions encompass middlebox-based interception techniques, searchable encryption, and trusted execution environment (TEE) (e.g., SGX [78]).

(2) Passive Detection Techniques: These techniques do not alter the underlying protocol (i.e., TLS 1.3) nor decrypt the encrypted traffic for inspection. They primarily include machine learning-based analysis techniques and statistical analysis based on traffic characteristics. Passive detection techniques can analyze traffic without compromising the security of TLS 1.3, though their detection accuracy and granularity may not be as high as those of active detection techniques.

We introduce three primary types of TLS 1.3 traffic analysis techniques below: middlebox-based interception, searchable encryption-based analysis, and machine learning-based analysis. We discuss their fundamental principles, advantages, disadvantages, and challenges, with an emphasis on recent research progress. Middlebox-based interception techniques, despite facing certain challenges, remain the most widely used active detection methods. Searchable encryption-based analysis techniques achieve essential traffic detection functions while protecting privacy, thus becoming a new research direction. Machine learning-based analysis techniques have made significant progress in accuracy and efficiency with the development of deep learning, thereby becoming a research hotspot in passive detection techniques.

4.1. Middlebox-Based Interception Techniques

Middlebox-based interception techniques, such as SSL/TLS proxy servers, involve inserting an intermediary entity between the sender and receiver of encrypted traffic. This entity can decrypt and inspect the traffic, as shown in Figure 3. These techniques are mainly used for network security detection and censorship. Table 2 summarizes existing middlebox interception schemes that support TLS 1.3 traffic.

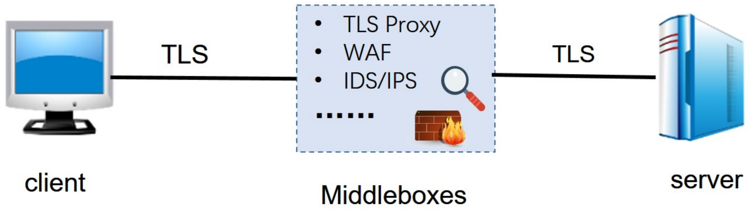

Figure 3. TLS traffic interception model based on middleboxes.

Table 2. Summary of middlebox solutions that support TLS 1.3.

Work	Year	Changes to TLS 1.3	Forward Secrecy	Privacy Protection	Performance HL	Performance CO	Performance BO	Deployment Complexity	Application Scenarios
TLS_visibility [79]	2018	Yes	Partial	M	M	M	M	H	Enterprise server TLS inspection
ETS [80]	2022	No	No	M	M	L	L	M	Passive decryption of internal enterprise traffic
LURK [81]	2022	No	Yes	H	M	M	L	M	Centralized management of TLS certificates and keys in enterprise intranets, securityaudit systems for monitoring TLS traffic
RFC9345 [82]	2023	No	Yes	M	L	L	L	M	Content Delivery Networks, remote data centers
ACCE-AP [83]	2018	No	Yes	H	M	M	M	M	Content Delivery Networks, enterprise firewalls, and content filtering
MaTLS [84]	2019	Yes	Yes	H	M	M	L	H	Enterprise networks, Content Delivery Networks, Middlebox in cloud services
ME-TLS [85]	2019	No	Yes	M	M	L	L	M	Industrial IoT, smart homes
ZKMB [86]	2022	No	Yes	H	H	H	L	M	Encrypted DNS filtering, HTTP firewalls
Zombie [87]	2024	No	Yes	H	M	M	M	M	DNS filtering, keyword filtering for search engine queries, Data Loss Prevention (DLP)

Note: HL denotes the handshake latency. CO denotes the computational overhead. BO denotes the bandwidth overhead. H denotes high. M denotes medium. L denotes low.

4.1.1. Session Splitting and Key-Sharing Techniques

To inspect TLS-encrypted traffic, TLS sessions are typically split, or TLS keys/secrets are shared after session establishment to circumvent TLS. The primary method of session splitting involves inserting a middlebox into the TLS connection and dividing it into two or more sessions. The client establishes an end-to-end TLS session with a middlebox (usually a TLS proxy), which, in turn, acts as a client and communicates with the target server. At this point, the middlebox acts as both the client and server. This model requires proper configuration of the client to trust certificates issued by the middlebox.

When clients cannot accept split TLS sessions, some techniques inspect traffic by sharing TLS secrets provided by the client. Browsers like Firefox can log TLS encryption information [88] (such as handshake/application traffic encryption information for TLS 1.3) to passively decrypt corresponding TLS traffic, but support for this feature is limited to a few applications and is subject to restrictions, and it may be abused.

Sharing the server's certificate private key is another effective method for passively monitoring TLS traffic [79,89]. When systems like intrusion detection obtain the server's certificate private key, they can decrypt the TLS traffic. Although these practices compromise the end-to-end security of TLS and pose risks of man-in-the-middle attacks and privacy issues, they are still widely deployed in antivirus software and enterprise network devices [69,90,91]. Due to the enhanced security of TLS 1.3, particularly through the mandatory use of PFS and the removal of some outdated and insecure encryption suites, including RSA-based key exchange mechanisms, its suitability for passive monitoring has decreased. Possessing the server's private key is no longer sufficient to inspect TLS 1.3 traffic. These changes make traditional interception methods, such as man-in-the-middle attacks (MITM) and key sharing, more challenging or impossible.

To facilitate server-side traffic monitoring, a simple solution is to make the server's DH key share static [22] and then share the private part with the middlebox, which reconstructs the TLS session secrets similarly to the server. However, the resulting semi-static DH key exchange no longer provides forward secrecy, contradicting the original intent of TLS 1.3. ETS [80], standardized by ETSI (European Telecommunications Standards Institute), is a variant of TLS 1.3 with semi-static DH keys, primarily used for the passive decryption of internal enterprise traffic. In the scenarios described in the ETS standard, connections entering the public Internet are protected by regular TLS, and, once inside the enterprise or data center, the traffic is forwarded to the final server using ETS. ETS provides keys to enterprise servers and middleboxes through a key management server, which significantly increases deployment complexity and attack risk. Due to the lack of forward secrecy, ETS has been criticized by various organizations.

An Internet draft [79] suggests establishing a tls_visibility extension in TLS 1.3, allowing pre-approved third parties to inspect connections. Clients opt in by indicating their willingness in the ClientHello message. This proposal is suitable for inspecting TLS connections to internal enterprise servers. However, if malicious users control the client and choose not to disclose TLS keys and secrets, they can circumvent the IPS inspection of encrypted traffic.

4.1.2. Delegation Credential Mechanisms

Although session splitting and key-sharing techniques can enable the inspection of TLS traffic, these methods often reduce TLS security. To maintain TLS security while allowing limited access to encrypted traffic, some new mechanisms have been proposed that delegate certain operations to trusted third parties.

The Limited Usage of Remote Key (LURK) protocol [92,93] is an extension to the TLS protocol that outsources sensitive key material to a remote LURK server, allowing LURK clients to interact with it for cryptographic operations. This design allows TLS handshakes without directly exposing private keys, thereby increasing security, especially in distributed and multi-tenant environments. LURK has defined a specific extension for TLS 1.3 [81] to ensure secure and effective TLS delegation and key management when using TLS 1.3, but its security guarantees have yet to be formally verified.

Internet standard RFC 9345 [82] introduces the concept of delegation credentials to address some limitations between TLS endpoint operators and certificate authorities, such as certificate validity periods, usage modes, and supported algorithms. It designs a mechanism for using "delegation credentials" in the TLS 1.3 handshake protocol, compatible with TLS 1.3, but may face challenges in practical deployment and operation.

4.1.3. Multi-Party TLS Protocol Variants

While delegation credential mechanisms offer ways to securely delegate certain TLS operations to trusted third parties, they still maintain a relatively simple protocol structure. However, as TLS 1.3 becomes more widely adopted and security requirements continue to evolve, researchers have begun exploring more complex approaches that allow trusted middleboxes to participate in TLS sessions while maintaining end-to-end encryption. These approaches, often referred to as multi-party TLS protocol variants, attempt to strike a balance between security, functionality, and compatibility. Such protocols typically involve modifying the TLS handshake process, introducing new message types or extensions to enable middlebox awareness and control.

When simple TLS session splitting and key-sharing solutions are unsustainable, such as incompatibility with TLS 1.3, another approach is to involve middleboxes in the TLS handshake and assign them different permissions or selectively expose parts of the traffic. Naylor et al. [94] pioneered proxy accountability, proposing the mcTLS protocol, which provides fine-grained access control for all proxy TLS connections between clients and servers. However, mcTLS is designed based on TLS 1.2 and does not natively support TLS 1.3. Bhargavan et al. [83] proposed a provably secure alternative to mcTLS: a general TLS proxy protocol design whose security can be modularly proven based on the underlying TLS security. The study implemented a prototype system based on the miTLS library, demonstrating how to instantiate the proposed design with TLS 1.3. Due to the end-to-end encryption characteristics of TLS limiting the functionality of middleboxes, Lee et al. [84] proposed a modified TLS protocol called maTLS, designed to be middlebox-aware, allowing middleboxes to participate in TLS sessions in a controllable and auditable manner. In terms of TLS 1.3 compatibility, maTLS was specifically designed to support TLS 1.3 by adding an ExtendedFinished message after the server's Finished message in pure server authentication mode. Since maTLS requires additional processing to support middlebox functionality, it may impact some performance advantages of TLS 1.3.

Li et al. [85] designed and implemented the ME-TLS protocol based on TLS 1.3 to address some limitations of traditional TLS protocols in IoT environments, allowing endpoints to introduce middleboxes into sessions with mutual consent. In ME-TLS, middleboxes can participate in TLS sessions while maintaining communication security without modifying the TLS 1.3 handshake structure, ensuring compatibility with existing systems. In actual deployment and application, careful consideration of middlebox management, protocol implementation complexity, and security challenges is essential.

4.1.4. Zero-Knowledge Proof Schemes

Traditionally, middleboxes decrypt traffic for inspection to enforce network policies, which compromises user privacy. An alternative approach seeks to balance network policy enforcement with privacy preservation. Zero-knowledge proof schemes offer a promising direction by allowing middleboxes to verify compliance with network policies without decrypting the traffic.

Traditionally, middleboxes decrypt traffic for inspection to enforce network policies, but this compromises user privacy. Paul Grubbs et al. [86] proposed the concept of Zero-Knowledge Middleboxes (ZKMBs), which can verify whether traffic complies with specific network management policies without decrypting it. The paper demonstrated how to combine a ZKMB with unmodified TLS 1.3, designing optimized zero-knowledge proofs for TLS 1.3 session keys. However, its computational overhead is significant, adding several seconds of traffic delay even under optimistic assumptions and relatively simple policies. Subsequently, the authors proposed a novel system called Zombie [87] based on ZKMB, adopting three techniques to reduce end-to-end delay, capable of handling policies based on regular expression matching, specifically designed and implemented for TLS 1.3 to ensure policy enforcement on network traffic while maintaining TLS 1.3 security and privacy. Compared to previous work, Zombie can reduce client and middlebox overhead by about 3.5 times, but it is still far from deployment in real-world network environments.

In summary, TLS 1.3 presents new requirements and challenges for interception methods. While new interception schemes attempt to achieve lawful interception and analysis of traffic without compromising the security of TLS 1.3, the effectiveness, security, and feasibility of deployment of these schemes remain key issues to be addressed in both research and practice. For example, general solutions like maTLS require support from both parties, and, unless these new schemes are widely deployed, users can easily choose to use standard TLS. Additionally, formal verification and security analysis of TLS 1.3 interception schemes have not yet met the required standards and require further research. It is important to note that these interception techniques violate the end-to-end security characteristics, raising privacy and potential compliance issues, especially as user privacy awareness increases, often requiring a balance between security, privacy protection, and practicality.

4.2. Searchable Encryption Techniques

With the rapid development of the Internet and cloud computing, network security concerns have become increasingly prominent. Deep packet inspection (DPI) is an effective network security technology that can thoroughly analyze network traffic to detect and prevent malicious activities. However, traditional DPI techniques require decrypting traffic, which can lead to privacy violations and security risks. To address this issue, researchers have proposed searchable encryption (SE) [95], which can detect malicious traffic through token matching without decryption. The core idea of SE is to establish a mappable relationship between encrypted keywords (tokens) and encrypted rule sets via a searchable encryption scheme. This technology aims to protect data privacy while supporting the efficient search and processing of encrypted data.

Middlebox systems based on searchable encryption (as shown in Figure 4) typically cannot directly decrypt TLS session data. Instead, they deploy proxy modules on the client and server sides, maintaining a dedicated encrypted token transmission link for traffic detection. Specifically, endpoint proxies tokenize the original network traffic using TLS session keys, generating encrypted tokens. These tokens are then encrypted with keys derived from the TLS session key and transmitted through a dedicated link to the middlebox. The middlebox maintains a rule set encrypted with the same derived key, such as rules for malicious traffic, and performs rule matching on the received encrypted token data to identify potential threats. After detection, the endpoint proxy verifies the consistency between the TLS session data and the token transmission link data to prevent attackers from bypassing detection through forged token data.

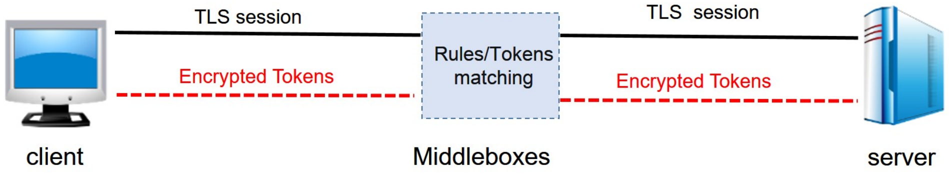

Figure 4. Middlebox model based on searchable encryption.

According to our literature search, no specific research on searchable encryption techniques for TLS 1.3 traffic exists. Theoretically, SE can handle any form of encrypted traffic without decrypting the content, as long as the traffic can be processed by the encryption/decryption and rule-matching mechanisms in the system. Since TLS 1.3 primarily enhances security and performance without changing the fundamental nature of encrypted traffic, SE techniques are applicable to both TLS 1.3 and other encrypted traffic types. We have reviewed relevant studies on encrypted traffic analysis using SE since 2018 (after the release of TLS 1.3), summarized in Table 3, and introduce them below.

Table 3. Summary of TLS traffic analysis techniques based on searchable encryption.

Work	Year	Changes to	Cryptographic	Detection	Privacy Protection			Matching and	Deployment Complexity					Application Scenarios
		TLS 1.3	Techniques	Functionality	S	C	MB	Inspection	S	C	MB	TTP	GW	
[96]	2015	Yes	AES	Full	H	H	M	Exact Matching, Regular Expression	✓	✓	-	-	-	DED, IDS, Parental filtering
[97]	2019	No	AES,DLP	Partial	H	H	H	Exact Matching	✓	✓	✓	-	-	ENS,ISP
[98]	2020	No	AES,PRF,DLP	Full	M	M	H	Exact Matching	✓	✓	✓	–	✓	ENS, CSSec
[99]	2020	No	AES, PRF, BF, Cuckoo Hashing	Partial	H	H	H	BF, Exact Matching	✓	–	✓	–	✓	Cloud-Assisted Middlebox
[100]	2020	No	Group Key Agreement, SymEnc	Full	H	H	M	Exact Matching	✓	✓	✓	✓	–	Encrypted Traffic Detection in IoT Scenarios
[101]	2021	No	AES, KH-PRF	Partial	H	H	H	Exact Matching	✓	✓	✓	–	–	IDS, IPS, DED
[102]	2021	No	PRF	Partial	H	H	H	Exact Matching	✓	✓	✓	–	–	ENS, CSSec
[103]	2022	No	SymEnc, Hash, PRF	Full	H	H	H	Exact Matching	✓	✓	✓	✓	–	Encrypted Traffic Detection in IoT Scenarios
[104]	2022	No	SMPC, SymEnc	Full	H	H	M	Exact Matching	✓	✓	✓	✓	–	IDS, DED
[105]	2023	No	AES, PRF, PRP, BF	Full	H	H	H	Exact Matching, TCK	–	–	✓	–	✓	ENS, IDS in Cloud Services
[106]	2023	No	HMAC ,CA	Full	H	H	H	Exact Matching	✓	–	–	✓	✓	IDS, WAF, DPI Services in the Cloud
[107]	2023	No	PRF, ECC, BF	Full	H	H	H	BF, Exact Matching	–	–	✓	✓	✓	Blockchain-based IIoT Environment

Note: S, C, and MB denote server, client, and middlebox. TTP and GW denote trusted third party and gateway. SymEnc denotes symmetric encryption, DLP denotes Discrete Logarithm Problem, PRF denotes Pseudorandom Function, KH-PRF denotes Key-Homomorphic Pseudo-Random Function, SMPC denotes secure multi-party computation, PRP denotes Pseudo-Random Permutation, HMAC denotes Hash-based Message Authentication Code, CA denotes Cryptographic Accumulator, BF denotes Bloom Filter, ECC denotes Elliptic Curve Cryptography, TCK denotes Token Continuity Check. Detection Functionality: Full means that all functions similar to plaintext detection can be realized, and Partial means restricted. Privacy Protection: indicates the degree of privacy protection of the client, server, and middleware. H means high, M means medium. Deployment Complexity: Whether or not clients, servers, middleware, trusted third parties, gateways, etc., need to be installed or configured. ✓ indicates that it is required and - indicates that it is not required. IDSs denotes intrusion detection systems, ISP denotes Internet Service Provider, ENS denotes Enterprise Network Security, CSSec denotes Cloud Service Security, DED denotes Data Exfiltration Detection, IPS denotes Intrusion Prevention System, IIoT denotes Industrial Internet of Things, WAF denotes Web Application Firewall.

4.2.1. Foundational Searchable Encryption Techniques

Sherry et al. [96] proposed BlindBox, the first DPI scheme based on searchable encryption, allowing middleboxes to match encrypted detection rules within encrypted packets without decryption, thus providing good user privacy protection. BlindBox establishes two connections between the client and server, one for the TLS session and another for token matching, thereby enabling the detection of encrypted traffic. Subsequently, several studies have utilized searchable encryption for encrypted traffic analysis (e.g., Embark [108], BlindIDS [109], SPABox [110], and Yuan et al. [111]). Generally, these studies incur high computational and communication overhead due to the use of encryption techniques.

4.2.2. Performance-Optimized Searchable Encryption Techniques

Building upon the foundational work in searchable encryption, researchers have focused on optimizing performance to address the high computational and communication overhead associated with earlier systems. These performance-oriented approaches aim to enhance encryption and detection speeds, improve rule processing efficiency, and reduce overall system latency.

Ning et al. [97] proposed the PrivDPI system, which significantly improved BlindBox's performance by introducing a reusable obfuscation-rule generation technique. Then, they further improved the performance of PrivDPI through the Pine scheme [98], which supports rule addition operations. Kim et al. [101] proposed the P2DPI system, which uses Key-Homomorphic Pseudo-Random Functions (KH-PRFs) to exchange encrypted detection rules, achieving the effective inspection of TLS-encrypted traffic without compromising user privacy. P2DPI shows significant improvements in connection setup and encryption speed, surpassing PrivDPI.

Canard et al. [102] proposed a symmetric encryption-based intrusion detection system, which demonstrates significant improvements in encryption and detection speed compared to BlindBox [96] and BlindIDS [109], showcasing better practicality.

These performance-optimized systems represent significant progress in making searchable encryption more viable for real-world applications. By addressing the challenges of speed and efficiency, these approaches have paved the way for more practical implementations of privacy-preserving deep packet inspection in encrypted networks.

4.2.3. Advanced Privacy and Security Mechanism

While performance optimization remains crucial, the evolving landscape of cyber threats and increasing privacy concerns have driven researchers to develop more sophisticated privacy and security mechanisms for searchable encryption. These advanced approaches aim to enhance the protection of both user data and detection rules, while maintaining the ability to effectively analyze encrypted traffic.

Ren et al. [99] proposed the EV-DPI system with a two-layer filtering architecture, ensuring the privacy of packet payloads and inspection rules while improving processing efficiency using fast symmetric encryption systems such as hash functions. EV-DPI supports deep packet inspection (DPI) in cloud environments, ensuring both privacy and efficiency. However, EV-DPI cannot dynamically add new rules and may fail when handling discontinuous tokens.

Jia et al. [104] proposed the OTEPI system, which uses Oblivious Transfer (OT) technology for rule encryption and natural language processing (NLP)-based tokenization to optimize text payload processing, thereby reducing the number of generated tokens to lower computational load. Results show that OTEPI can accurately detect rule matches in encrypted traffic while maintaining both traffic and rule privacy. Compared to previous methods, it achieves a better balance between communication traffic consumption and computational resource usage.

Deng et al. [105] proposed the DCDPI system for encrypted traffic detection, introducing the VBTree+ (an improved virtual binary tree structure) and token continuity check mechanism, supporting dynamic rule addition and efficient token continuity checks, enhancing system flexibility and robustness while ensuring forward privacy, i.e., not leaking the relationship between historical tokens and rules when updating rules.

Recently, Zhang et al. [106] proposed a privacy-preserving lightweight cloud-based DPI execution verification scheme, which protects the privacy of DPI rule sets. This scheme introduces a commitment-based delayed sampling mechanism and a trusted third party based on Intel SGX, thereby achieving efficient and secure verification.

These studies showcase innovative techniques that push the boundaries of privacy-preserving searchable encryption, introducing novel cryptographic methods, improved rule management, and robust verification mechanisms. These advancements not only address the limitations of earlier systems but also pave the way for more secure and flexible encrypted traffic analysis solutions.

4.2.4. Domain-Specific and Emerging Technology Solutions

As searchable encryption techniques continue to evolve, researchers are increasingly focusing on tailoring these solutions to specific domains and leveraging emerging technologies. This trend reflects the growing need for specialized approaches that can address the unique challenges posed by different environments, such as IoT ecosystems and industrial settings. Additionally, the integration of cutting-edge technologies like blockchain and trusted execution environments is opening up new possibilities for enhancing the security, efficiency, and functionality of searchable encryption systems.

In the cloud computing domain, the previously mentioned EV-DPI system by Ren et al. [99] stands out for its optimization for cloud environments. While its privacy-preserving aspects were discussed earlier, it is worth noting that EV-DPI's two-layer filtering architecture is particularly well suited for the distributed nature of cloud computing, offering a balance between privacy protection and efficient processing in large-scale cloud infrastructures.

Fan et al. [100] present GKA_DPI, an encrypted traffic detection scheme designed for IoT environments. This scheme addresses the limitations of existing schemes, such as PrivDPI, in the one/many-to-many communication model of IoT by using a dynamic group key agreement protocol to reduce power consumption. The scheme is particularly suitable for IoT environments using the MQTT protocol. Although GKA_DPI offers enhanced efficiency and security, it is more complex to deploy.

Chen et al. [103] proposed a novel scheme using symmetric encryption to extract tokens from traffic, combining hash functions and pseudo-random functions to obfuscate rules. This scheme is used to detect encrypted network traffic in IoT environments while protecting the privacy of transmitted data.

Zhang et al. [107] also proposed the BTDPI system, a blockchain-based privacy-preserving traceable encrypted traffic detection system designed for industrial IoT scenarios. BTDPI uses a two-layer filtering architecture and an online–offline certificate-free aggregate signature mechanism, enabling efficient detection and anomaly source tracing of encrypted traffic. Experimental results show that BTDPI is 26.7 times faster than existing systems when processing 3000 tokens and 3000 rules.

These domain-specific solutions demonstrate how searchable encryption techniques are being adapted to meet the diverse needs of modern network ecosystems. From IoT-specific protocols to cloud-optimized architectures and blockchain integration, these approaches showcase the versatility and potential of searchable encryption in various technological contexts.

Despite the progress made in using SE for encrypted traffic analysis, several challenges persist. First, these schemes typically require additional computational resources, which may increase performance overhead. Second, the limitations of regular expression detection prevent these schemes from fully replacing traditional DPI techniques. Although SE-based traffic analysis methods are theoretically applicable to TLS 1.3, practical deployment and application necessitate adapting and optimizing related systems to address the new security features and performance requirements of TLS 1.3. Future research should focus on proposing efficient and compatible SE schemes to support secure traffic analysis based on TLS 1.3, necessitating further exploration by academia and the industry.

4.3. Machine Learning Methods for TLS 1.3 Traffic Analysis

Machine learning has become a powerful tool for extracting information features without decrypting traffic, making it widely applicable to TLS-encrypted traffic analysis. Since the payload of TLS is encrypted, most machine learning methods utilize statistical or behavioral features of TLS-encrypted traffic, such as packet size, direction, and timing data, to train corresponding machine learning models.

The release of TLS 1.3 in 2018 brought significant security enhancements, including reduced handshake round trips and the removal of insecure encryption algorithms, altering the characteristics of TLS 1.3 traffic. For example, the TLS 1.2 handshake typically required the exchange of 5 to 7 packets, whereas the TLS 1.3 handshake can be completed with 0 to 3 packets. Therefore, the effectiveness of machine learning-based traffic analysis methods suitable for TLS 1.2 and earlier versions in the context of TLS 1.3 remains to be explored. In this section, we provide a comprehensive overview of the latest advancements in machine learning-based TLS 1.3-encrypted traffic analysis. As shown in Table 4, we summarize the relevant research on machine learning-based TLS 1.3 traffic analysis.

4.3.1. Feature Extraction and Representation Learning

The advent of TLS 1.3 has necessitated the development of new feature extraction and representation learning techniques. This is primarily because the TLS 1.3 protocol encrypts all information after the handshake, thereby significantly reducing plaintext information. This change renders many encrypted traffic analysis methods based on TLS 1.2 plaintext features ineffectual. Consequently, researchers have proposed various innovative approaches to address this challenge.

Akbari et al. [112] extracted features using flow statistics, traffic shape (size, time, and direction), and raw bytes of the TLS handshake, constructing a neural network architecture based on convolutional neural networks (CNNs) and stacked LSTM layers for encrypted web traffic classification. Although not specifically analyzed for TLS 1.3, this method reduces reliance on server identity exposure fields (such as SNI), making it adaptable to the TLS 1.3 environment.

Fu et al. [31] proposed an encrypted malicious traffic detection system called ST-Graph, which uses graph representation learning algorithms to analyze the spatial and temporal characteristics of network behavior by constructing interaction graphs between hosts and servers to reveal patterns of malicious activities. Experimental results show that even after removing features related to the TLS protocol, ST-Graph can still achieve 99.85% accuracy, demonstrating its potential for effective analysis and detection of malicious activities within the TLS 1.3 environment. Taking a novel approach to traffic representation, Chen et al. [123] proposed a novel traffic graph representation model called Weaved Flow Fragment (WFF) by deeply studying the transmission patterns and interaction characteristics of encrypted traffic, converting the packet sequence of encrypted traffic into a graph to better capture the intrinsic relationships between packet sequences. Based on this model, the authors proposed an integrated graph neural network (EGNN) architecture, significantly improving the accuracy of encrypted traffic classification through voting and stacking integration mechanisms. The dataset includes new protocols (TLS 1.3, QUIC, and DTLS) and applications, demonstrating EGNN's better applicability in open-world environments.

Table 4. Summary of machine learning-based TLS 1.3 traffic analysis research.

Work	Application Domain	Method	Granularity	Feature	Metrics	TLS 1.3	
						Theo	Data
[6]	Service Classification	LS-CapsNet	Packet-level	PDU length	P, R, F1	✓	✗
[112]	Service and Application Classification	CNN,LSTM	Flow-level	Flow statistics,TLS handshake packets	P, R, F1, Acc,Time	✓	—
[113]	Service Classification	LS-CapsNet, LSTM	Packet-level, Flow-level	multiPDU length sequence	P, R, F1	✓	✓
[114]	Cloud Platform Application Classification	NeuTic	Packet-level	Packet length,Packet window size, TCP flag	TP, FP, FN, P, R, F-m, Acc	✓	✓
[115]	Service Classification	AB-RF, RB-RF	Flow-level	TLS handshake packets	Acc, P, F1, Error Rate	✓	✓
[7]	Encrypted Application Classification	ET-BERT	Packet-level, Flow-level	Convert raw traffic data into a sequence of tokens	Acc,P, R, F1	✓	✓
[31]	Malicious Traffic Detection	ST-Graph	Packet-level, Flow-level	Flow statistics, TLS handshake packets	P, R, FP	✓	—
[116]	Application Layer Protocol Identification	GGFAS	Packet-level, Flow-level	Packet size, direction, and order	Acc, F1, Confusion Matrix	✓	✓
[117]	Malware Traffic Classification	R1DIT	Raw-level	The relative position of the original byte sequence of each packet	R, P, F1, TP, FAR	✓	✓
[118]	Webpage Fingerprinting	KNN	Trace-level	Distance between samples	Top-N Acc	✓	✓
[119]	Service Classification	LightGBM, CNN	Packet-level, Flow-level	Packet and flow statistics	Acc, F1, TP, FP, AUROC	✓	✓
[120]	Application Identification	PASS	Packet-level Raw-level	Packet length sequence Raw payload sequence	Acc, P, R, F1	✓	✓
[121]	Phishing Detection	LR, SVM, RF, XGBoost, LightGBM	Packet-level	12 characteristics such as SNI, selected Cipher Suite (SCS)	TP, FP, TN, FN, P,ACC	✓	✓
[122]	Traffic Classification	Rosetta	Flow-level	Packet length sequence	Acc, R, F1, FP	✓	—
[123]	Service Classification	WFF-EGNN	Flow-level	Packet length sequence	P, R, F1, Time	✓	✓
[124]	Application Classification	MISS	Packet-, Flow-, and Raw-level	Multiview sequence features such as packet length sequence, TLS header, and payload byte sequence	Acc, F1	✓	—

Note: Theo and Data represent theoretical and dataset, respectively. P and R denote Precision and Recall, respectively. F1 represents F1-Score. TP, FP, and FN denote True Positive, False Positive, and False Negative, respectively. Acc represents Accuracy. FAR denotes the false alarm rate. Top-N Acc denotes the attack success rate. ✓and ✗represent theoretical and dataset support, respectively. — denotes that the dataset does not specify TLS 1.3.

Yun et al. [114] proposed a new method called NeuTic, which is based on the packet sequence of each TLS flow. This method is designed to effectively classify raw TLS flows generated by multiple "cloud" applications. This method can automatically capture long-distance dependencies between elements in the packet sequence, achieving robust and accurate classification of encrypted TLS traffic. It should be noted that the dataset used in this study includes both TLS 1.2 and TLS 1.3 traffic, but the authors did not distinguish between these versions in subsequent experiments and analyses, treating them collectively.

Focusing on specific security threats, Kumar et al. [121] proposed a new machine learning model for detecting phishing URLs within encrypted traffic. For TLS 1.3 traffic, the authors extracted features from ClientHello and ServerHello messages for phishing detection, yielding significant results. However, these features mainly focus on visible information exchanged during the TLS handshake. Given the enhanced privacy protection in TLS 1.3, exploring deeper and more distinctive features from TLS 1.3 traffic remains a promising direction.

Addressing the challenge of web fingerprinting, Mavroudis et al. [118] proposed an adaptive method that learns low-dimensional representations (embedding models) of input data to achieve high-accuracy recognition of web pages in TLS 1.2- and TLS 1.3-encrypted traffic. To facilitate further research, the authors released the TLS 1.3 dataset, Github500. However, the single-source nature of this dataset and the lack of testing for scenarios with significant dynamic changes in web pages suggest that further validation and improvement of the model's adaptability is necessary.

Addressing the challenges of analyzing encrypted traffic across different TLS versions, Piet et al. [116] proposed an automated network traffic analysis framework called GGFAST, which searches for characteristic patterns in information length and provides corresponding packet length conversion functions for different encryption methods such as stream ciphers, block ciphers, and AEAD under different TLS versions. On a dataset collected from a large enterprise network, GGFAST was able to identify 95.1% of DNS-over-HTTPS (DoH) traffic within TLS traffic, partially overcoming the challenges of analyzing TLS 1.3-encrypted traffic.

Recently, Xie et al. [122] proposed a method called Rosetta, which uses TCP-aware traffic enhancement mechanisms and self-supervised learning to understand implicit TCP semantics, thereby extracting robust features of TLS traffic. Experiments show that Rosetta significantly improves the performance of existing deep learning models in classifying TLS traffic in diverse network environments. Although this study did not specifically analyze TLS 1.3 traffic, Rosetta's method is protocol-agnostic and applicable to TLS 1.3, aiding researchers and network administrators in better understanding and analyzing TLS 1.3-encrypted traffic. Li et al. [124] proposed an incremental learning framework called Multi-view Sequences Fusion (MISS), which improves the classification accuracy of encrypted traffic by extracting multi-view sequence features, generating cross-view information, and adaptively fusing multi-view data. The framework also mitigates the problem of knowledge forgetting during model evolution by designing plasticity and stability branches. Although the study did not specifically discuss the TLS 1.3 protocol, the methodology and techniques of the MISS framework provide a possible solution for classifying TLS 1.3-encrypted traffic.

These diverse approaches to feature extraction and representation learning demonstrate the ongoing efforts to adapt to the challenges posed by TLS 1.3, paving the way for more effective and robust encrypted traffic analysis techniques.

4.3.2. Deep Learning Models and Algorithms

While some researchers have focused on advancing feature extraction and representation learning techniques, others have concentrated on developing sophisticated deep learning models and algorithms to address the challenges posed by TLS 1.3-encrypted traffic analysis. The deep learning models and algorithms discussed in this section aim to improve classification accuracy, enhance generalization capabilities, and increase adaptability to new protocols and emerging threats.

Chen et al. [6] proposed a service classification method for encrypted traffic based on the length of multiple protocol data units (PDUs), utilizing the Markov properties between PDU length sequences and employing a length-sensitive capsule neural network (LS-CapsNet) model for traffic classification. Although this method is suitable for the TLS 1.3 environment, it has not been validated using a TLS 1.3 dataset. Subsequently, the authors proposed a length-sensitive composite deep learning (LSCDL) model to achieve encrypted traffic service classification [113], which showed good classification results on a proprietary dataset containing a significant amount of TLS 1.3 traffic.

Although ECH encrypts sensitive information (such as SNI) in the ClientHello message, posing challenges to existing algorithms that rely on this information for traffic classification, new algorithms can still effectively classify TLS-encrypted traffic. Shamsimukhametov et al. [115] proposed two new traffic classification algorithms (AB-RF and RB-RF) to address the challenges introduced by ECH. Even with ECH encryption in TLS 1.3, traffic classification based on the TLS handshake remains feasible.

Existing solutions heavily rely on deep features such as data size, making it difficult to generalize for unseen data. How to use open-domain unlabeled traffic data to learn representations with strong generalization capabilities remains a significant challenge.

Lin et al. [7] proposed a model called ET-BERT, which pre-trains deep datagram-level representations on large-scale unlabeled data and then fine-tunes on small-scale labeled data for specific tasks to achieve accurate classification of encrypted traffic. The authors collected a new dataset named CSTNET-TLS 1.3. ET-BERT significantly improved the F1 score of the encrypted application classification task by 10.0%, reaching an F1 score of 97.4%, significantly surpassing existing methods.

Barut et al. [117] proposed a residual one-dimensional image transformation model (R1DIT) based on raw data, using meta-learning methods (transfer learning and few-shot learning) to extend its classification accuracy to unseen malware categories, capable of identifying newly emerging TLS 1.3 malware traffic. Due to the very small size and highly imbalanced category distribution of the target dataset, further validation on larger and more comprehensive datasets is required to verify the proposed method's efficacy.

Luxemburk et al. [119] provided a robust framework and method set for analyzing TLS-encrypted traffic, creating and evaluating a large TLS dataset, and proposing effective methods for detecting unknown services. Although the dataset covers TLS 1.3 protocol traffic, the study did not specifically discuss TLS 1.3 traffic analysis separately.

Li et al. [120] proposed a new semi-supervised encrypted traffic classification framework called PASS, which improves the model's ability to learn features of minority class applications through contrastive pre-training and semi-supervised learning while reducing dependence on labeled training data. The study analyzed TLS 1.3 protocol traffic and validated the effectiveness of the PASS framework in handling the latest encrypted protocol traffic through experiments on the CSTNET-TLS 1.3 dataset.

4.3.3. Datasets

Table 5 lists the datasets used in the field of TLS 1.3 traffic analysis. It is important to note that many TLS datasets do not explicitly indicate whether they contain TLS 1.3 traffic, which can complicate the evaluation of machine learning models designed for this protocol version. Therefore, Table 5 only includes datasets that explicitly state that they contain TLS 1.3 traffic.

In the area of service classification, the study [115] used the WNL TLS dataset, which includes network traffic from 100 popular websites and background network traffic, covering 12 categories and 3547 flows, including TLS 1.3 traffic. The CSTNET-TLS 1.3 [7] is the first dataset for TLS 1.3, which was collected in 2021 and contains traffic from 120 applications deployed on Alexa Top-5000 websites using TLS 1.3.

We observed that, due to the scarcity of public datasets, most works in this section did not use public datasets to train models but instead used self-created datasets. For example, the dataset [113] collected in the CERNET environment includes traffic from seven services, with a significant amount of TLS 1.3-encrypted traffic. Another dataset [114] includes TLS traffic from 15 mobile applications from companies like Alibaba, Baidu, and ByteDance, as well as six video/short video mobile applications, covering both TLS 1.2 and TLS 1.3. The dataset used for encrypted application classification [116] consists of seven datasets, comprising 15 categories, including POP3-over-TLS, IMAP-over-TLS, SMTP-over-TLS, and HTTP-over-TLS.

The dataset [117] focuses on malware traffic classification, extracting TLS 1.3 traffic from the CICDDoS2019 dataset, covering five DDoS variants and 209 TLS 1.3 malware flows.

In addition to the aforementioned datasets, numerous other datasets have been released. However, a comprehensive statistical analysis of these datasets has not been conducted in this study, primarily because the majority of newly proposed datasets do not explicitly specify whether they include TLS 1.3 traffic, even when released during the period of widespread TLS 1.3 adoption. For instance, the Malicious_TLS dataset [125] serves as a valuable resource for studying encrypted malicious traffic. This dataset encompasses 22 categories of real-world encrypted malicious traffic generated by malware families active between 2018 and 2021, along with benign TLS traffic for comparative purposes. All samples were collected from actual networks and encrypted using TLS. However, the paper does not explicitly state whether the dataset contains TLS 1.3 traffic or specify the exact proportion of TLS 1.3 traffic within the dataset.

Although some public datasets are available for TLS 1.3 traffic analysis, the lack of publicly available, labeled datasets means that most research still relies on private datasets. This highlights the need for more comprehensive public datasets to support research and development in this domain. In addition, the lack of clear information on the distribution of TLS versions is also the reason for the current lack of TLS 1.3 datasets.

Machine learning techniques can be directly applied to existing TLS 1.3 traffic without any modifications to the protocol itself. This makes machine learning methods easy to deploy while preserving the end-to-end encryption security provided by the TLS 1.3 protocol. This contrasts with other technologies that require protocol modifications (such as access control technologies) or changes to client/server settings (such as middlebox, searchable encryption, and trusted hardware). However, machine learning struggles to achieve the fine-grained, rule-based detection capabilities of deep packet inspection. Current research mainly focuses on tasks such as traffic classification and anomaly detection, with limited capabilities for executing more complex security policies. Additionally, due to privacy and security concerns, publicly available TLS 1.3 traffic datasets are small and lack diversity. Most studies use privately collected datasets, which poses challenges for fair evaluation and comparison of algorithms and limits the reproducibility and generalization of research findings.

Table 5. Datasets used by works in TLS 1.3 traffic analysis.

Work	Analysis Objective	Name	Year	Available	Description
[113]	Service Classification	–	2020	Private	Including traffic from seven services, with a significant amount of TLS 1.3-encrypted traffic
[114]	Cloud Application Classification	–	–	Private	Generated by 15 mobile applications from three companies and six video mobile applications
[115]	Service Classification	WNL TLS	2021	Public	Includes web traffic from 100 popular websites and background web traffic
[7,120]	Application Classification	CSTNET-TLS 1.3	2021	Public	Including traffic from 120 applications deployed on Alexa Top-5000 websites using TLS 1.3
[116]	Protocol Identification	–	2022	Private	Composed of seven datasets, including 15 categories such as SMTP-over-TLS and HTTP-over-TLS
[117]	Malware Traffic Classification	–	2019	Private	Extracted from the CICDDoS2019 dataset, covering 5 DDoS variants and 209 TLS 1.3 malware flows
[118]	Web Fingerprinting	Github500	–	Private	Includes 500 TLS 1.3 categories generated by accessing the top 500 GitHub project pages
[119]	Service Classification	CESNET-TLS22	2021	Public	The dataset contains a total of 140 million flow records, covering 191 network services
[121]	Phishing Detection	–	–	Private	URLs collected from the Alexa database, phishing websites collected from Phishtank
[123]	Service Classification	CESNET-2022Service	2022	Public	Covering seven applications with TLS protocols, including TLS 1.2 and TLS 1.3

5. Challenges and Future Research Directions in TLS 1.3 Traffic Analysis

5.1. Challenges

(1) Technical Challenges Introduced by TLS 1.3: TLS 1.3 enhances security and efficiency through measures such as PFS, 0-RTT session resumption, and encryption of all handshake messages. While some new interception schemes for TLS 1.3 have been proposed, they often fall short in terms of security, performance, and deployment costs. For example, maTLS [84] and tls_visibility [79] lack sufficient security analysis and validation, and ETS [80] has been criticized for its lack of forward secrecy. Designing secure, efficient, and easily deployable interception schemes remains a significant challenge.

(2) Balancing Privacy and Security: TLS 1.3 further strengthens user privacy protection, but middlebox-based interception and searchable encryption technologies may pose information leakage risks. Users and privacy advocates typically aim to maximize the protection of communication content, while network operators need to inspect traffic for compliance or security reasons. Balancing user privacy protection with network security is a critical challenge in TLS traffic analysis.

(3) Comprehensive Detection Capabilities: Current technologies like searchable encryption and machine learning methods have limited detection capabilities compared to plaintext inspection. Searchable encryption primarily supports keyword matching and struggles with regular expression matching. While machine learning can perform traffic classification and anomaly detection, it falls short in executing complex security policies and rules. Expanding these technologies to cover more detection needs is an urgent challenge. Although trusted hardware solutions can achieve full functionality, their deployment flexibility is limited.

(4) Challenges for Machine Learning Techniques: Machine learning techniques can analyze traffic without changing existing network settings or decrypting network traffic, providing an optimal solution for passive detection. However, the standardization and widespread use of ECH pose significant challenges to these methods. With valuable feature information such as SNI becoming invisible, the statistical model of encrypted traffic changes dramatically. This shift increases the difficulty of traffic classification, demanding machine learning models with enhanced feature extraction and generalization capabilities to uncover patterns from limited metadata and temporal features of encrypted traffic. The impact of ECH on machine learning techniques extends beyond feature availability. It necessitates a fundamental rethinking of model architectures and training strategies to adapt to this new, more opaque traffic environment. Furthermore, the scarcity of publicly available TLS 1.3 traffic datasets hinders related research. Since TLS 1.3 is not yet widely deployed, most research is in the exploratory stage. To achieve the large-scale application of machine learning methods in TLS 1.3 traffic analysis, more work is needed in dataset construction, feature engineering, model performance optimization, and knowledge transfer.

(5) Incomplete Adoption and Complex Environment: The current limited adoption of ECH and the continued use of lower TLS versions for side channels create a complex environment [126] for traffic analysis. Researcher must now contend with a mix of fully encrypted, partially encrypted, and legacy traffic patterns. This diversity poses significant challenges for developing consistent and effective traffic analysis techniques. It requires adaptive methods capable of handling multiple protocol versions and encryption levels simultaneously, complicating both the design of analysis algorithms and their real-world deployment. Moreover, this transitional phase may persist for an extended period, necessitating flexible and scalable solutions that can evolve with the changing landscape of encrypted traffic.

5.2. Future Research Directions

Future research directions in TLS traffic analysis should include the following:

(1) Privacy-Preserving Traffic Detection Technologies: Research methods that combine various privacy-preserving technologies to allow authorized parties to perform limited detection on encrypted traffic without compromising forward secrecy. Implementing privacy-preserving techniques can introduce significant computational overhead and com-

plexity. Research should focus on optimizing cryptographic protocols to reduce overhead and improve efficiency, developing hybrid approaches that combine multiple privacy-preserving techniques. For example, exploring privacy-preserving traffic detection technologies based on searchable encryption, homomorphic encryption, secure multi-party computation, and trusted execution environments can meet network security and management needs while protecting user privacy. environments can meet network security and management needs while protecting user privacy.

(2) Machine Learning-Based Traffic Analysis Technologies: With the development of deep learning, graph neural networks, and multimodal learning, researchers are continuously exploring new methods to improve the accuracy and efficiency of encrypted traffic analysis. For example, using graph neural networks to model the spatiotemporal features of encrypted traffic and multimodal learning to integrate traffic metadata, external threat intelligence, and other heterogeneous information. The scarcity of labeled datasets and the need for robust feature extraction in the face of limited metadata visibility are significant hurdles. Active learning and federated learning can help address data scarcity by leveraging unlabeled data. Transfer learning and meta-learning can enhance model generalization, while graph neural networks and multimodal learning can improve feature extraction from encrypted traffic.

(3) System Verifiability and Transparency: Given the complexity of the TLS 1.3 protocol and the black-box nature of encryption, users find it difficult to verify whether the behavior of traffic analysis systems aligns with their claims. Some studies mention using trusted hardware and cryptographic proofs to provide partial transparency, but these face many challenges in actual deployment. Interdisciplinary research involving cryptography, formal verification, and secure hardware is needed to develop frameworks that ensure system trustworthiness. Combining formal verification with machine learning could automate verification processes, enhancing efficiency and coverage.

(4) Customized Interception Schemes: Security requirements, privacy protection needs, and performance optimization goals can vary significantly across different scenarios, complicating the design of interception schemes. This diversity necessitates the development of tailored interception approaches based on specific characteristics and requirements of each environment. To address this challenge, it is crucial to develop modular interception architectures customized for specific contexts, such as enterprise networks, cloud services, and IoT ecosystems. This approach will ensure robust security while optimizing performance for each unique setting. For example, enterprise networks may prioritize data leakage prevention and access control, cloud services may focus on multi-tenant isolation and virtualization security, and IoT may emphasize device authentication and lightweight encryption. Successful implementation of this strategy requires close collaboration with industry stakeholders to accurately identify and address their specific needs and concerns.

(5) Detection Function Expansion: Research technologies that can support multiple detection needs simultaneously to meet the requirements of various middlebox services, for example, developing technologies capable of regular expression matching and complex rule detection [20] to provide detection capabilities equivalent to plaintext analysis. Further exploration is needed of the application of machine learning in encrypted traffic analysis, expanding machine learning-based passive analysis techniques to cover more detection needs, for instance, using deep learning to automatically extract advanced features, unsupervised learning to discover unknown threats, and reinforcement learning to optimize detection strategies. Additionally, methods such as active learning and data synthesis can help generate high-fidelity labeled data for training models.

6. Discussion

Through a comprehensive review of the state-of-the-art research efforts in analyzing TLS 1.3-encrypted traffic, we have identified several key findings and lessons learned.

Middlebox interception techniques, such as TLS proxies and session splitting, have been widely used for network security and management. However, the enhanced security features of TLS 1.3, such as mandatory forward secrecy and the removal of insecure cryptographic algorithms, have rendered traditional interception methods less effective or even infeasible. While researchers have proposed various TLS 1.3-aware interception schemes, their security, performance, and deployability still require further investigation and improvement. This highlights the need for developing more secure, efficient, and easily deployable interception schemes that can adapt to the evolving TLS protocol.

Searchable encryption techniques enable encrypted traffic inspection without decrypting the content, providing a promising approach for privacy-preserving analysis. Although existing searchable encryption schemes are theoretically applicable to TLS 1.3 traffic, their practical deployment and adaptation to the new protocol features remain an open challenge. The limitations of existing schemes in terms of efficiency, functionality extension, and formal security analysis necessitate further research to make searchable encryption a viable solution for TLS 1.3 traffic analysis.

Machine learning has emerged as a powerful tool for encrypted traffic analysis, capable of extracting valuable information from encrypted traffic without decryption. Despite the challenges posed by TLS 1.3's enhanced encryption and reduced metadata visibility, researchers have developed innovative machine learning models that leverage various traffic features, such as packet lengths, timing, and flow patterns. However, the scarcity of publicly available TLS 1.3 datasets and the need for robust feature engineering and model optimization remain significant obstacles to the widespread adoption of machine learning techniques in this domain. Collaborative efforts in dataset creation and sharing, as well as advancements in feature extraction and model generalization, are crucial for the progress of machine learning-based TLS 1.3 traffic analysis.

Balancing privacy and security is a critical challenge in TLS traffic analysis. While TLS 1.3 strengthens user privacy protection, middlebox-based interception and searchable encryption technologies may pose information leakage risks. Finding solutions that protect user privacy while allowing necessary traffic inspection for security purposes requires a delicate balance and innovative approaches. Privacy-preserving traffic detection technologies that combine various techniques, such as searchable encryption, homomorphic encryption, secure multi-party computation, and trusted execution environments, hold promise for achieving this balance.

These findings underscore the importance of continued research and innovation in TLS 1.3 traffic analysis techniques. By addressing the identified challenges and exploring the proposed future research directions, we can develop more effective, secure, and privacy-preserving solutions for analyzing TLS 1.3-encrypted traffic.

7. Conclusions

The widespread adoption of TLS 1.3 has brought significant improvements in security and privacy for encrypted communication. However, this has also posed unprecedented challenges for encrypted traffic analysis. This survey provides a comprehensive overview of the state-of-the-art research efforts in analyzing TLS 1.3-encrypted traffic. We have systematically reviewed three major analysis approaches: middlebox interception techniques, searchable encryption methods, and machine learning-based techniques. For each approach, we have conducted a critical examination of its applicability, strengths, and limitations in the context of TLS 1.3.

TLS 1.3 traffic analysis is a dynamic and challenging field with significant implications for network security, management, and privacy. While existing techniques have made notable progress, there remain open problems and opportunities for future research. Several key research directions deserve further exploration, including the development of privacy-preserving traffic inspection techniques, advancements in machine learning techniques for encrypted traffic analysis, and the establishment of formal verification methods and security transparency frameworks. By addressing these challenges and exploring innovative solutions, we can fully realize the potential of encrypted traffic analysis in the context of TLS 1.3. This will enable effective network monitoring and protection while preserving user privacy.

Author Contributions: Conceptualization, J.Z. and Z.S.; methodology, J.Z.; formal analysis, J.Z. and Z.S.; investigation, J.Z.; resources, J.Z.; data curation, J.Z.; writing—original draft preparation, J.Z and Z.S.; writing—review and editing, W.F., W.H., T.H. and Z.Z.; visualization, J.Z. and Z.S.; supervision, Z.S.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the University's Independent Research Project (Grant No. 2022508020).

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Google. HTTPS Encryption on the Web. Available online: https://transparencyreport.google.com/https/overview (accessed on 18 April 2024).
2. Allen, C.; Dierks, T. The TLS Protocol Version 1.0. RFC 2246. 1999. Available online: https://www.rfc-editor.org/info/rfc2246 (accessed on 19 April 2024).
3. Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. 2018. Available online: https://www.rfc-editor.org/info/rfc8446 (accessed on 19 April 2024).
4. Qualys. Qualys SSL Labs—SSL Pulse. Available online: https://www.ssllabs.com/ssl-pulse/ (accessed on 19 April 2024).
5. Rescorla, E.; Dierks, T. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246. 2008. Available online: https://www.rfc-editor.org/info/rfc5246 (accessed on 19 April 2024).
6. Chen, Z.; Cheng, G.; Jiang, B.; Tang, S.; Guo, S.; Zhou, Y. Length matters: Fast internet encrypted traffic service classification based on multi-PDU lengths. In Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN), Tokyo, Japan, 17–19 December 2020; pp. 531–538.
7. Lin, X.; Xiong, G.; Gou, G.; Li, Z.; Shi, J.; Yu, J. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 633–642.
8. Velan, P.; Čermák, M.; Čeleda, P.; Drašar, M. A survey of methods for encrypted traffic classification and analysis. *Int. J. Netw. Manag.* **2015**, *25*, 355–374. [CrossRef]
9. Rezaei, S.; Liu, X. Deep learning for encrypted traffic classification: An overview. *IEEE Commun. Mag.* **2019**, *57*, 76–81. [CrossRef]
10. Pacheco, F.; Exposito, E.; Gineste, M.; Baudoin, C.; Aguilar, J. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Commun. Surv. Tutorials* **2018**, *21*, 1988–2014. [CrossRef]
11. Papadogiannaki, E.; Ioannidis, S. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [CrossRef]
12. Shen, M.; Ye, K.; Liu, X.; Zhu, L.; Kang, J.; Yu, S.; Li, Q.; Xu, K. Machine learning-powered encrypted network traffic analysis: A comprehensive survey. *IEEE Commun. Surv. Tutorials* **2022**, *25*, 791–824. [CrossRef]
13. Kwon, D.; Kim, H.; Kim, J.; Suh, S.C.; Kim, I.; Kim, K.J. A survey of deep learning-based network anomaly detection. *Clust. Comput.* **2019**, *22*, 949–961. [CrossRef]
14. Yi, T.; Chen, X.; Zhu, Y.; Ge, W.; Han, Z. Review on the application of deep learning in network attack detection. *J. Netw. Comput. Appl.* **2023**, *212*, 103580. [CrossRef]
15. Aceto, G.; Ciuonzo, D.; Montieri, A.; Pescapé, A. Toward effective mobile encrypted traffic classification through deep learning. *Neurocomputing* **2020**, *409*, 306–315. [CrossRef]
16. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2224–2287. [CrossRef]
17. Aceto, G.; Ciuonzo, D.; Montieri, A.; Pescapé, A. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 445–458. [CrossRef]
18. Tahaei, H.; Afifi, F.; Asemi, A.; Zaki, F.; Anuar, N.B. The rise of traffic classification in IoT networks: A survey. *J. Netw. Comput. Appl.* **2020**, *154*, 102538. [CrossRef]
19. Sultana, N.; Chilamkurti, N.; Peng, W.; Alhadad, R. Survey on SDN based network intrusion detection system using machine learning approaches. *Peer Netw. Appl.* **2019**, *12*, 493–501. [CrossRef]
20. Poh, G.S.; Divakaran, D.M.; Lim, H.W.; Ning, J.; Desai, A. A survey of privacy-preserving techniques for encrypted traffic inspection over network middleboxes. *arXiv* **2021**, arXiv:2101.04338.
21. Oh, C.; Ha, J.; Roh, H. A survey on TLS-encrypted malware network traffic analysis applicable to security operations centers. *Appl. Sci.* **2021**, *12*, 155. [CrossRef]
22. de Carné de Carnavalet, X.; van Oorschot, P.C. A Survey and Analysis of TLS Interception Mechanisms and Motivations: Exploring how end-to-end TLS is made "end-to-me" for web traffic. *ACM Comput. Surv.* **2023**, *55*, 1–40. [CrossRef]

23. Anderson, B.; McGrew, D. Identifying encrypted malware traffic with contextual flow data. In Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, Vienna, Austria, 28 October 2016; pp. 35–46.
24. Anderson, B.; McGrew, D. Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 1723–1732.
25. Wang, S.; Chen, Z.; Zhang, L.; Yan, Q.; Yang, B.; Peng, L.; Jia, Z. Trafficav: An effective and explainable detection of mobile malware behavior using network traffic. In Proceedings of the 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), Beijing, China, 20–21 June 2016; pp. 1–6.
26. Liu, C.; Cao, Z.; Xiong, G.; Gou, G.; Yiu, S.M.; He, L. Mampf: Encrypted traffic classification based on multi-attribute markov probability fingerprints. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–10.
27. Liu, C.; He, L.; Xiong, G.; Cao, Z.; Li, Z. Fs-net: A flow sequence network for encrypted traffic classification. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1171–1179.
28. Zhang, W.; Meng, Y.; Liu, Y.; Zhang, X.; Zhang, Y.; Zhu, H. Homonit: Monitoring smart home apps from encrypted traffic. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1074–1088.
29. Cisco. Cisco Encrypted Traffic Analytics White Paper. Available online: https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/nb-09-encrytd-traf-anlytcs-wp-cte-en.html (accessed on 18 April 2024).
30. Zheng, W.; Gou, C.; Yan, L.; Mo, S. Learning to classify: A flow-based relation network for encrypted traffic classification. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 13–22.
31. Fu, Z.; Liu, M.; Qin, Y.; Zhang, J.; Zou, Y.; Yin, Q.; Li, Q.; Duan, H. Encrypted malware traffic detection via graph-based network analysis. In Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses, Limassol, Cyprus, 26–28 October 2022; pp. 495–509.
32. Qing, Y.; Yin, Q.; Deng, X.; Chen, Y.; Liu, Z.; Sun, K.; Xu, K.; Zhang, J.; Li, Q. Low-Quality Training Data Only? A Robust Framework for Detecting Encrypted Malicious Network Traffic. *arXiv* **2023**, arXiv:2309.04798.
33. Fu, C.; Li, Q.; Xu, K. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. *arXiv* **2023**, arXiv:2301.13686.
34. Anderson, B.; McGrew, D. Tls beyond the browser: Combining end host and network data to understand application behavior. In Proceedings of the Internet Measurement Conference, Amsterdam, The Netherlands, 21–23 October 2019; pp. 379–392.
35. Dimopoulos, G.; Leontiadis, I.; Barlet-Ros, P.; Papagiannaki, K. Measuring video QoE from encrypted traffic. In Proceedings of the 2016 Internet Measurement Conference, Santa Monica, CA, USA, 14–16 November 2016; pp. 513–526.
36. Pan, W.; Cheng, G.; Wu, H.; Tang, Y. Towards QoE assessment of encrypted YouTube adaptive video streaming in mobile networks. In Proceedings of the 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), Beijing, China, 20–21 June 2016; pp. 1–6.
37. Oche, M.; Noor, R.M.; Chembe, C. Multivariate statistical approach for estimating QoE of real-time multimedia applications in vehicular ITS network. *Comput. Commun.* **2017**, *104*, 88–107. [CrossRef]
38. Shen, M.; Zhang, J.; Xu, K.; Zhu, L.; Liu, J.; Du, X. Deepqoe: Real-time measurement of video qoe from encrypted traffic with deep learning. In Proceedings of the 2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS), Hangzhou, China, 15–17 June 2020; pp. 1–10.
39. Wu, H.; Li, X.; Cheng, G.; Hu, X. Monitoring video resolution of adaptive encrypted video traffic based on HTTP/2 features. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada, 10–13 May 2021; pp. 1–6.
40. Shbair, W.M.; Cholez, T.; Francois, J.; Chrisment, I. A multi-level framework to identify HTTPS services. In Proceedings of the NOMS 2016—2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; pp. 240–248.
41. Yamauchi, H.; Nakao, A.; Oguchi, M.; Yamamoto, S.; Yamaguchi, S. A study on service identification based on server name indication analysis. In Proceedings of the 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW), Nagasaki, Japan, 26–29 November 2019; pp. 470–474.
42. Liu, X.; You, J.; Wu, Y.; Li, T.; Li, L.; Zhang, Z.; Ge, J. Attention-based bidirectional GRU networks for efficient HTTPS traffic classification. *Inf. Sci.* **2020**, *541*, 297–315. [CrossRef]
43. Cheng, J.; Wu, Y.; Yuepeng, E.; You, J.; Li, T.; Li, H.; Ge, J. MATEC: A lightweight neural network for online encrypted traffic classification. *Comput. Netw.* **2021**, *199*, 108472. [CrossRef]
44. Panchenko, A.; Lanze, F.; Pennekamp, J.; Engel, T.; Zinnen, A.; Henze, M.; Wehrle, K. Website Fingerprinting at Internet Scale. In NDSS. 2016. Available online: https://nymity.ch/tor-dns/pdf/Panchenko2016a.pdf (accessed on 26 April 2024).
45. Li, S.; Guo, H.; Hopper, N. Measuring information leakage in website fingerprinting attacks and defenses. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1977–1992.

46. Rimmer, V.; Preuveneers, D.; Juarez, M.; Goethem, T.V.; Joosen, W. Automated Website Fingerprinting through Deep Learning. In Proceedings of the Proceedings 2018 Network and Distributed System Security Symposium, San Diego, CA, USA, 18–21 February 2018. [CrossRef]
47. Sirinam, P.; Mathews, N.; Rahman, M.S.; Wright, M. Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 1131–1148.
48. Mathews, N.; Holland, J.K.; Oh, S.E.; Rahman, M.S.; Hopper, N.; Wright, M. SoK: A critical evaluation of efficient website fingerprinting defenses. In Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–25 May 2023; pp. 969–986.
49. Rezaei, S.; Kroencke, B.; Liu, X. Large-scale mobile app identification using deep learning. *IEEE Access* **2019**, *8*, 348–362. [CrossRef]
50. Jiang, M.; Li, Z.; Fu, P.; Cai, W.; Cui, M.; Xiong, G.; Gou, G. Accurate mobile-app fingerprinting using flow-level relationship with graph neural networks. *Comput. Netw.* **2022**, *217*, 109309. [CrossRef]
51. Van Ede, T.; Bortolameotti, R.; Continella, A.; Ren, J.; Dubois, D.J.; Lindorfer, M.; Choffnes, D.; Van Steen, M.; Peter, A. Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 23–26 February 2020; Volume 27.
52. Xu, G.; Xu, M.; Chen, Y.; Zhao, J. A Mobile Application-Classifying Method Based on a Graph Attention Network from Encrypted Network Traffic. *Electronics* **2023**, *12*, 2313. [CrossRef]
53. Conti, M.; Mancini, L.V.; Spolaor, R.; Verde, N.V. Ca not you hear me knocking: Identification of user actions on android apps via traffic analysis. In Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, San Antonio, TX, USA, 2–4 March 2015; pp. 297–304.
54. Saltaformaggio, B.; Choi, H.; Johnson, K.; Kwon, Y.; Zhang, Q.; Zhang, X.; Xu, D.; Qian, J. Eavesdropping on {Fine-Grained} user activities within smartphone apps over encrypted network traffic. In Proceedings of the 10th USENIX Workshop on Offensive Technologies (WOOT 16), Austin, TX, USA, 8–9 August 2016.
55. Dubin, R.; Dvir, A.; Pele, O.; Hadar, O. I know what you saw last minute—encrypted http adaptive video streaming title classification. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 3039–3049. [CrossRef]
56. Li, Y.; Huang, Y.; Xu, R.; Seneviratne, S.; Thilakarathna, K.; Cheng, A.; Webb, D.; Jourjon, G. Deep content: Unveiling video streaming content from encrypted wifi traffic. In Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 1–3 November 2018; pp. 1–8.
57. Wu, H.; Wu, Q.; Cheng, G.; Guo, S.; Hu, X.; Yan, S. SFIM: Identify user behavior based on stable features. *Peer Netw. Appl.* **2021**, *14*, 3674–3687. [CrossRef]
58. Scheffler, S.; Mayer, J. Sok: Content moderation for end-to-end encryption. *arXiv* **2023**, arXiv:2303.03979. [CrossRef]
59. Hall, J.L.; Aaron, M.D.; Andersdotter, A.; Jones, B.; Feamster, N.; Knodel, M. A Survey of Worldwide Censorship Techniques. RFC 9505. 2023. Available online: https://www.rfc-editor.org/info/rfc9505 (accessed on 26 April 2024).
60. Wu, M.; Sippe, J.; Sivakumar, D.; Burg, J.; Anderson, P.; Wang, X.; Bock, K.; Houmansadr, A.; Levin, D.; Wustrow, E. How the Great Firewall of China detects and blocks fully encrypted traffic. In Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23), Anaheim, CA, USA, 9–11 August 2023; pp. 2653–2670.
61. Frolov, S.; Wustrow, E. The use of TLS in Censorship Circumvention. In NDSS. 2019. Available online: https://www.freehaven.net/anonbib/papers/ndss2019_03B-2-1_Frolov_paper.pdf (accessed on 26 April 2024).
62. Trustwave. Filter : SNI Extension Feature and HTTPS Blocking. 2015. Available online: https://www3.trustwave.com/software/8e6/hlp/r3000/files/1system_filter.html (accessed on 26 April 2024).
63. Sophos. Sophos Firewall: Web Filtering Basics. 2023. Available online: https://support.sophos.com/support/s/article/KB-000036518?language=en_US (accessed on 26 April 2024).
64. Shbair, W.M.; Cholez, T.; Goichot, A.; Chrisment, I. Efficiently bypassing SNI-based HTTPS filtering. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 990–995.
65. Morgus, R.; Sherman, J.; Nam, S. Analysis: South Korea's New Tool for Filtering Illegal Internet Content. 2019. Available online: https://www.newamerica.org/cybersecurity-initiative/c2b/c2b-log/analysis-south-koreas-sni-monitoring/ (accessed on 27 April 2024).
66. Bock, D.L.K.; Merino, L.; Fifield, D.; Housmansadr, A.; Levin, D. Exposing and Circumventing China's Censorship of ESNI. 2020. Available online: https://geneva.cs.umd.edu/posts/china-censors-esni/esni/ (accessed on 26 April 2024).
67. Satija, S.; Chatterjee, R. BlindTLS: Circumventing TLS-based HTTPS censorship. In Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet, Virtual, 27 August 2021; pp. 43–49.
68. Waked, L. Analyzing TLS Interception in Middleware Network Appliances. Ph.D. Thesis, Concordia University, Montreal, QC, Canada, 2018.
69. Waked, L.; Mannan, M.; Youssef, A. To intercept or not to intercept: Analyzing tls interception in network appliances. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security, Incheon, Republic of Korea, 4 June 2018; pp. 399–412.
70. Afzal, A.; Hussain, M.; Saleem, S.; Shahzad, M.K.; Ho, A.T.; Jung, K.H. Encrypted network traffic analysis of secure instant messaging application: A case study of signal messenger app. *Appl. Sci.* **2021**, *11*, 7789. [CrossRef]

71. Sarhan, S.A.E.; Youness, H.A.; Bahaa-Eldin, A.M. A framework for digital forensics of encrypted real-time network traffic, instant messaging, and VoIP application case study. *Ain Shams Eng. J.* **2023**, *14*, 102069. [CrossRef]
72. Dierks, T.; Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346. 2006. Available online: https://www.rfc-editor.org/info/rfc4346 (accessed on 26 April 2024).
73. Moriarty, K.; Farrell, S. Deprecating TLS 1.0 and TLS 1.1. RFC 8996. 2021. Available online: https://www.rfc-editor.org/info/rfc8996 (accessed on 19 April 2024).
74. Holz, R.; Hiller, J.; Amann, J.; Razaghpanah, A.; Jost, T.; Vallina-Rodriguez, N.; Hohlfeld, O. Tracking the deployment of TLS 1.3 on the Web: A story of experimentation and centralization. *ACM SIGCOMM Comput. Commun. Rev.* **2020**, *50*, 3–15. [CrossRef]
75. Dowling, B.; Fischlin, M.; Günther, F.; Stebila, D. A cryptographic analysis of the TLS 1.3 handshake protocol. *J. Cryptol.* **2021**, *34*, 37. [CrossRef]
76. Rescorla, E.; Oku, K.; Sullivan, N.; Wood, C.A. TLS Encrypted Client Hello. 2024. Available online: https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-18 (accessed on 18 May 2024).
77. Bhargavan, K.; Cheval, V.; Wood, C. A symbolic analysis of privacy for tls 1.3 with encrypted client hello. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA, 7–11 November 2022; pp. 365–379.
78. Van Bulck, J.; Minkin, M.; Weisse, O.; Genkin, D.; Kasikci, B.; Piessens, F.; Silberstein, M.; Wenisch, T.F.; Yarom, Y.; Strackx, R. Foreshadow: Extracting the keys to the intel {SGX} kingdom with transient {Out-of-Order} execution. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 991–1008.
79. Housley, R.; Droms, R. TLS 1.3 Option for Negotiation of Visibility in the Datacenter. Internet-Draft draft-rhrd-tls-tls13-visibility-01, Internet Engineering Task Force. 2018. Available online: https://datatracker.ietf.org/doc/draft-rhrd-tls-tls13-visibility/01/ (accessed on 18 May 2024).
80. ETSI. Middlebox Security Protocol—Part 3: Enterprise Transport Security. 2019. Available online: https://www.etsi.org/deliver/etsi_ts/103500_103599/10352303/01.03.01_60/ts_10352303v010301p.pdf (accessed on 18 May 2024).
81. Migault, D. LURK Extension version 1 for (D)TLS 1.3 Authentication. Internet-Draft draft-mglt-lurk-tls13-06, Internet Engineering Task Force. 2022. Available online: https://datatracker.ietf.org/doc/draft-mglt-lurk-tls13/06/ (accessed on 18 May 2024).
82. Barnes, R.; Iyengar, S.; Sullivan, N.; Rescorla, E. Delegated Credentials for TLS and DTLS. RFC 9345. 2023. Available online: https://www.rfc-editor.org/info/rfc9345 (accessed on 18 May 2024).
83. Bhargavan, K.; Boureanu, I.; Delignat-Lavaud, A.; Fouque, P.A.; Onete, C. A formal treatment of accountable proxying over TLS. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 799–816.
84. Lee, H.; Smith, Z.; Lim, J.; Choi, G.; Chun, S.; Chung, T.; Kwon, T.T. maTLS: How to Make TLS Middlebox-Aware? In NDSS. 2019. Available online: https://hw5773.github.io/paper/matls.pdf (accessed on 18 May 2024).
85. Li, J.; Chen, R.; Su, J.; Huang, X.; Wang, X. ME-TLS: Middlebox-enhanced TLS for internet-of-things devices. *IEEE Internet Things J.* **2019**, *7*, 1216–1229. [CrossRef]
86. Grubbs, P.; Arun, A.; Zhang, Y.; Bonneau, J.; Walfish, M. {Zero-Knowledge} Middleboxes. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 4255–4272.
87. Zhang, C.; DeStefano, Z.; Arun, A.; Bonneau, J.; Grubbs, P.; Walfish, M. Zombie: Middleboxes that {Don't} Snoop. In Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24), Santa Clara, CA, USA, 16–18 April 2024; pp. 1917–1936.
88. Mozilla. NSS Key Log Format. 2019. Available online: https://nss-crypto.org/reference/security/nss/legacy/key_log_format/index.html (accessed on 20 June 2024).
89. Green, M.; Droms, R.; Housley, R.; Turner, P.; Fenter, S. Data Center Use of Static Diffie-Hellman in TLS 1.3. 2017. Available online: https://datatracker.ietf.org/doc/draft-green-tls-static-dh-in-tls13/ (accessed on 18 May 2024).
90. de Carnavalet, X.D.C.; Mannan, M. Killed by proxy: Analyzing client-end TLS interception software. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 21–24 February 2016.
91. Durumeric, Z.; Ma, Z.; Springall, D.; Barnes, R.; Sullivan, N.; Bursztein, E.; Bailey, M.D.; Halderman, J.A.; Paxson, V. The Security Impact of HTTPS Interception. In NDSS. 2017. Available online: https://git.safemobile.org/crimeflare/cloudflare-tor/raw/commit/020252c3748c37c4b0f2da47f46b3505f82435fa/pdf/2017-The_Security_Impact_of_HTTPS_Interception.pdf (accessed on 19 May 2024).
92. Migault, D. LURK Protocol Version 1. Internet-Draft draft-mglt-lurk-lurk-01, Internet Engineering Task Force. 2021. Available online: https://datatracker.ietf.org/doc/draft-mglt-lurk-lurk/01/ (accessed on 18 May 2024).
93. Migault, D.; Boureanu, I. LURK Extension Version 1 for (D)TLS 1.2 Authentication. Internet-Draft draft-mglt-lurk-tls12-05, Internet Engineering Task Force. 2021. Available online: https://datatracker.ietf.org/doc/draft-mglt-lurk-tls12/05/ (accessed on 18 May 2024).
94. Naylor, D.; Schomp, K.; Varvello, M.; Leontiadis, I.; Blackburn, J.; López, D.R.; Papagiannaki, K.; Rodriguez Rodriguez, P.; Steenkiste, P. Multi-context TLS (mcTLS) enabling secure in-network functionality in TLS. *ACM SIGCOMM Comput. Commun. Rev.* **2015**, *45*, 199–212. [CrossRef]
95. Song, D.X.; Wagner, D.; Perrig, A. Practical techniques for searches on encrypted data. In Proceedings of the 2000 IEEE Symposium on Security and Privacy. S&P 2000, Berkeley, CA, USA, 14–17 May 2000; pp. 44–55.

96. Sherry, J.; Lan, C.; Popa, R.A.; Ratnasamy, S. Blindbox: Deep packet inspection over encrypted traffic. *ACM SIGCOMM Comput. Commun. Rev.* **2015**, *45*, 213–226. [CrossRef]
97. Ning, J.; Poh, G.S.; Loh, J.C.; Chia, J.; Chang, E.C. PrivDPI: Privacy-preserving encrypted traffic inspection with reusable obfuscated rules. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 1657–1670.
98. Ning, J.; Huang, X.; Poh, G.S.; Xu, S.; Loh, J.C.; Weng, J.; Deng, R.H. Pine: Enabling privacy-preserving deep packet inspection on TLS with rule-hiding and fast connection establishment. In Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, 14–18 September 2020; Proceedings, Part I 25; Springer: Berlin/Heidelberg, Germany, 2020; pp. 3–22.
99. Ren, H.; Li, H.; Liu, D.; Xu, G.; Cheng, N.; Shen, X. Privacy-preserving efficient verifiable deep packet inspection for cloud-assisted middlebox. *IEEE Trans. Cloud Comput.* **2020**, *10*, 1052–1064. [CrossRef]
100. Fan, Z.; Zeng, Y.; Zhu, X.; Ma, J. A group key agreement based encrypted traffic detection scheme for Internet of Things. In Proceedings of the 1st ACM International Workshop on Security and Safety for Intelligent Cyber-Physical Systems, Virtual, 16–19 November 2020; pp. 19–26.
101. Kim, J.; Camtepe, S.; Baek, J.; Susilo, W.; Pieprzyk, J.; Nepal, S. P2DPI: Practical and privacy-preserving deep packet inspection. In Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, Virtual, 7–11 June 2021; pp. 135–146.
102. Canard, S.; Li, C. Towards practical intrusion detection system over encrypted traffic. *IET Inf. Secur.* **2021**, *15*, 231–246. [CrossRef]
103. Chen, D.; Wang, H.; Zhang, N.; Nie, X.; Dai, H.N.; Zhang, K.; Choo, K.K.R. Privacy-preserving encrypted traffic inspection with symmetric cryptographic techniques in IoT. *IEEE Internet Things J.* **2022**, *9*, 17265–17279. [CrossRef]
104. Jia, X.; Zhang, M. Encrypted Packet Inspection Based on Oblivious Transfer. *Secur. Commun. Networks* **2022**, *2022*, 4743078. [CrossRef]
105. Deng, M.; Zhang, K.; Wu, P.; Wen, M.; Ning, J. DCDPI: Dynamic and Continuous Deep Packet Inspection in Secure Outsourced Middleboxes. *IEEE Trans. Cloud Comput.* **2023**, *11*, 3510–3524. [CrossRef]
106. Zhang, X.; Geng, W.; Song, Y.; Cheng, H.; Xu, K.; Li, Q. Privacy-Preserving and Lightweight Verification of Deep Packet Inspection in Clouds. *IEEE/ACM Trans. Netw.* **2023**, *32*, 159–174. [CrossRef]
107. Zhang, K.; Deng, M.; Gong, B.; Miao, Y.; Ning, J. Privacy-Preserving Traceable Encrypted Traffic Inspection in Blockchain-based Industrial IoT. *IEEE Internet Things J.* **2023**, *11*, 3484–3496. [CrossRef]
108. Lan, C.; Sherry, J.; Popa, R.A.; Ratnasamy, S.; Liu, Z. Embark: Securely outsourcing middleboxes to the cloud. In Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), Santa Clara, CA, USA, 16–18 March 2016; pp. 255–273.
109. Canard, S.; Diop, A.; Kheir, N.; Paindavoine, M.; Sabt, M. BlindIDS: Market-compliant and privacy-friendly intrusion detection system over encrypted traffic. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 561–574.
110. Fan, J.; Guan, C.; Ren, K.; Cui, Y.; Qiao, C. Spabox: Safeguarding privacy during deep packet inspection at a middlebox. *IEEE/ACM Trans. Netw.* **2017**, *25*, 3753–3766. [CrossRef]
111. Yuan, X.; Wang, X.; Lin, J.; Wang, C. Privacy-preserving deep packet inspection in outsourced middleboxes. In Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
112. Akbari, I.; Salahuddin, M.A.; Ven, L.; Limam, N.; Boutaba, R.; Mathieu, B.; Moteau, S.; Tuffin, S. A look behind the curtain: Traffic classification in an increasingly encrypted web. *Proc. ACM Meas. Anal. Comput. Syst.* **2021**, *5*, 1–26. [CrossRef]
113. Chen, Z.; Cheng, G.; Xu, Z.; Guo, S.; Zhou, Y.; Zhao, Y. Length matters: Scalable fast encrypted internet traffic service classification based on multiple protocol data unit length sequence with composite deep learning. *Digit. Commun. Netw.* **2022**, *8*, 289–302. [CrossRef]
114. Yun, X.; Wang, Y.; Zhang, Y.; Zhao, C.; Zhao, Z. Encrypted tls traffic classification on cloud platforms. *IEEE/ACM Trans. Netw.* **2022**, *31*, 164–177. [CrossRef]
115. Shamsimukhametov, D.; Kurapov, A.; Liubogoshchev, M.; Khorov, E. Is encrypted clienthello a challenge for traffic classification? *IEEE Access* **2022**, *10*, 77883–77897. [CrossRef]
116. Piet, J.; Nwoji, D.; Paxson, V. Ggfast: Automating generation of flexible network traffic classifiers. In Proceedings of the ACM SIGCOMM 2023 Conference, New York, NY, USA, 10 September 2023; pp. 850–866.
117. Barut, O.; Luo, Y.; Li, P.; Zhang, T. R1DIT: Privacy-Preserving Malware Traffic Classification With Attention-Based Neural Networks. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 2071–2085. [CrossRef]
118. Mavroudis, V.; Hayes, J. Adaptive Webpage Fingerprinting from TLS Traces. In Proceedings of the 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Porto, Portugal, 27–30 June 2023; pp. 445–458.
119. Luxemburk, J.; Čejka, T. Fine-grained TLS services classification with reject option. *Comput. Netw.* **2023**, *220*, 109467. [CrossRef]
120. Li, X.; Guo, J.; Song, Q.; Xie, J.; Sang, Y.; Zhao, S.; Zhang, Y. Listen to Minority: Encrypted Traffic Classification for Class Imbalance with Contrastive Pre-Training. In Proceedings of the 2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Madrid, Spain, 11–14 September 2023; pp. 447–455.
121. Kumar, M.; Kondaiah, C.; Pais, A.R.; Rao, R.S. Machine learning models for phishing detection from TLS traffic. *Clust. Comput.* **2023**, *26*, 3263–3277. [CrossRef]

122. Xie, R.; Wang, Y.; Cao, J.; Dong, E.; Xu, M.; Sun, K.; Li, Q.; Shen, L.; Zhang, M. Rosetta: Enabling robust tls encrypted traffic classification in diverse network environments with tcp-aware traffic augmentation. In Proceedings of the ACM Turing Award Celebration Conference-China 2023, Wuhan, China, 28–30 July 2023; pp. 131–132.
123. Chen, Z.; Cheng, G.; Niu, D.; Qiu, X.; Zhao, Y.; Zhou, Y. WFF-EGNN: Encrypted Traffic Classification based on Weaved Flow Fragment via Ensemble Graph Neural Networks. *IEEE Trans. Mach. Learn. Commun. Netw.* **2023**, *1*, 389–411. [CrossRef]
124. Li, X.; Xie, J.; Song, Q.; Sang, Y.; Zhang, Y.; Li, S.; Zang, T. Let model keep evolving: Incremental learning for encrypted traffic classification. *Comput. Secur.* **2024**, *137*, 103624. [CrossRef]
125. Yuan, Q.; Liu, C.; Yu, W.; Zhu, Y.; Xiong, G.; Wang, Y.; Gou, G. BoAu: Malicious traffic detection with noise labels based on boundary augmentation. *Comput. Secur.* **2023**, *131*, 103300. [CrossRef]
126. Khandkar, V.S.; Hanawal, M.K.; Kulkarni, S.G. State of internet privacy and tales of ECH-TLS. In Proceedings of the 2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS), Bangalore, India, 3–8 January 2023; pp. 165–170.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.