Applications of MASQUE-proxies in TEE Environments

Martin Halfen, Lion Steger*, Daniel Petri*
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: martin.halfen@tum.de, stegerl@net.in.tum.de, petriroc@net.in.tum.de

***Abstract*—This paper will talk about two privacy and anonymity enhancing technologies: MASQUE-Proxies and Trusted Execution Environments (TEEs). TEEs offer hardware-based confidentiality and integrity for sensitive computations, whereas MASQUE enables encrypted and obfuscated traffic tunneling over QUIC and HTTP/3, enhancing resistance to traffic analysis. This paper investigates the possibilities of merging these two technologies to create secure, privacy-preserving proxy infrastructures. We investigate the technical foundation of TEEs and MASQUE, assess practical implementations like Apple's iCloud Private Relay, and discuss the role of TEEs in enhancing trust through verifiable execution. We assess performance trade-offs using related TEE-based proxy implementations.**

***Index Terms*—confidential computing, cloud computing, MASQUE-Proxy, TEE**

1. Introduction

Today the demand for privacy and anonymity in the online realm is higher than ever, New technologies such as Trusted Execution Environments (TEEs) and MASQUE proxies are promising ways to address these concerns. TEEs leverage hardware-based security to ensure that sensitive data is processed securely, even if the surrounding software stack is compromised by an adversary.

MASQUE (Multiplexed Application Substrate over QUIC Encryption) is a new proxying protocol built upon the modern foundations of HTTP/3 and QUIC. The protocol allows the user to tunnel arbitrary data through connections that appear to external observers as standard encrypted HTTP traffic [1]. This obfuscation method helps protect metadata and improves resistance against traffic analysis. MASQUE is already deployed in services like Apple's iCloud Private Relay, which aims to hide users' network traffic from adversaries [2].

The combination of TEEs and MASQUE proxies creates a powerful framework for both secure computation and private communication. This synergy opens up promising opportunities for privacy-preserving applications in this area.

This paper explores the technical foundations, potential applications, and challenges of combining these technologies. In Section 2.1, we provide an overview of TEEs and their specification. In Section 2.2 we cover the MASQUE protocol and its technical foundations. With this background we discuss in Section 4 the potential applications of combining these technologies, as well as the technical challenges that can arise with their integration.

2. Background

This section provides the essential background needed to understand how MASQUE proxies are integrated with Trusted Execution Environments. Our starting point will be a description of the fundamental principles and security guarantees of TEEs, with an emphasis on the mechanisms that render them especially appropriate for safeguarding sensitive computations on untrusted infrastructure. Subsequently, we investigate the MASQUE protocol and its distinctive proxying features which uses QUIC and HTTP/3, emphasizing how it facilitates efficient and private network tunneling. These technologies can be combined to establish the foundation for proxy architectures that are secure, high-performance, and privacy-preserving.

2.1. Trusted Execution Environments (TEEs)

TEEs leverage hardware-based security mechanisms to protect the integrity and confidentiality of software running on potentially untrusted platforms, such as public cloud infrastructure, third-party data centers, or infrastructure with uncontrolled access. In this paper, we assume an adversarial model where the adversary has full control over the entire software stack and can execute any arbitrary privileged program, and has also full control over all OS duties like CPU scheduling and IO operations [3]. TEEs offer two critical functionalities: **remote attestation** and **runtime protection**.

Remote attestation enables an external verifier to determine the integrity of a TEE instance before interacting with it as Li et. al show in their work [3]. This process involves the TEE instance requesting an attestation from a trusted attestation authority, typically a secure manufacturer TEE instance or a secure hardware component on the chip [3]. The authority generates a signed attestation report that includes a cryptographic measurement of the TEE's initial state, covering both the deployed software and critical system components, including CPU features, memory layout, and operating system parameters [3]. The verifier can then use this report to determine whether the TEE instance is trustworthy before providing sensitive data or workloads.

After verifying the initial state of the TEE through remote attestation, the system must also ensure its integrity during runtime. This phase, referred to as *Runtime Protection* (RP), is responsible for managing system resources such as CPU, memory, and I/O—while protecting the TEE from interference by the untrusted components of the OS [3].

 doi: 10.2313/NET-2025-11-3_14

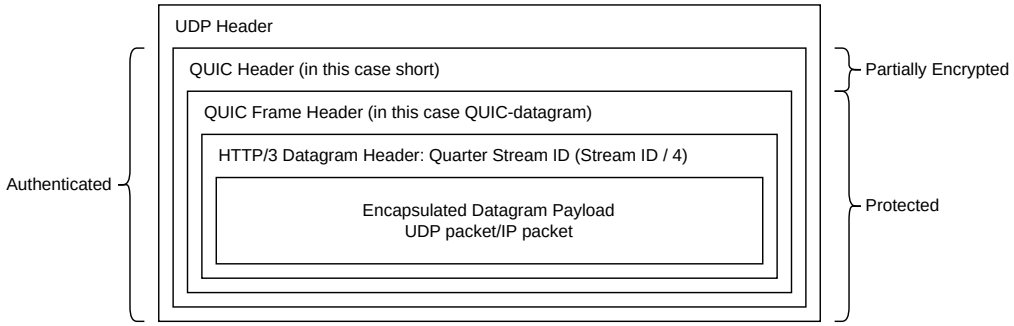

Figure 1: UDP/IP over MASQUE-Proxy taken from [1]

Several runtime protection strategies have been proposed by Li et. all [3] in thier paper:

1) **Unprotected Mode**: The TEE relies fully on the untrusted OS for resource management, especially for I/O operations like networking or disk access [3]. This simplicity comes at the cost of exposure to side-channel attacks and manipulation.
2) **Isolated Runtime Management**: TEE resources are managed separately by the runtime layer, while the OS handles non-TEE workloads. This is typical for context switching scenarios [3].
3) **Guarded Runtime**: A mediation layer observes and verifies OS actions, such as memory allocation, to maintain integrity without full detachment from the OS [3].
4) **TEE-Managed Mode**: The TEE manages its hardware resources independently (e.g., virtual memory). While highly isolated, this mode can conflict with system schedulers and degrade performance [3].

Each of these modes represents a trade-off between security, performance and trust assumptions. Selecting the appropriate runtime protection strategy depends on the threat model and the level of trust placed in the underlying OS.

2.2. MASQUE Proxy

The core idea behind the *Multiplexed Application Substrate over QUIC Encryption* (MASQUE) protocol is to enable the tunneling of arbitrary data streams over HTTP/3 using the QUIC transport protocol [1]. MASQUE supports two main tunneling modes: `CONNECT-UDP` and `CONNECT-IP`, both of which allow traffic to be obfuscated and multiplexed in a way that resembles standard web traffic, thereby enhancing privacy [1].

In `CONNECT-UDP` mode, the client must first specify the target IP address and port to which UDP packets should be forwarded. After the QUIC connection with the MASQUE-proxy is established, raw UDP packets are wrapped within HTTP/3 datagrams and sent to the MASQUE-proxy. The proxy then unwraps the UDP-packages from the QUIC-header and forwards the UDP payloads to the specified endpoint and relays the responses back to the client. This mode is particularly suitable for use cases such as Voice-over-IP, where UDP is already the underlying transport protocol [1].

In `CONNECT-IP` mode, the client encapsulates raw IPv4 or IPv6 packets into HTTP/3 datagrams, which are then transmitted over a QUIC connection to the MASQUE proxy. The proxy unwraps the packets and forwards them to their intended destination, enabling the tunneling of arbitrary IP traffic. This mechanism allows MASQUE to emulate a full-tunnel VPN, where the client's entire IP-layer traffic is routed through the proxy in a way that is indistinguishable from standard encrypted web traffic [1].

In both tunneling modes, MASQUE leverages the encryption of QUIC and commonness of HTTP/3 traffic to hide the nature of tunneled traffic, making it more difficult for an adversary to perform traffic classification, surveillance or correlation. This makes MASQUE a compelling candidate for privacy-preserving proxy deployments in trusted or semi-trusted network environments.

3. Related Work

TEEs have been proposed as a foundation for secure computing in an untrusted environment such as data centers. Li et al. [3] provide a comprehensive systematization of the design choices and pitfalls in modern TEE architectures, including remote attestation and runtime protection models [3]. Practical deployments methods such as SCONE [4] have demonstrated how TEEs can be used to secure containerized applications.

In the context of encryption proxies, Bouhairi et al. [5] evaluate a SCONE-based implementation of the Eperi Gateway, an encryption proxy, and demonstrate how TEEs can ensure data confidentiality. However, their findings also reveal a measurable performance overhead, with latency increasing from 423 ms to 912 ms, effectively more than doubling [5]. Their findings provide insights into the practical trade-offs when deploying proxy services within secure enclaves.

The MASQUE protocol is a relatively recent development, offering encrypted tunneling over HTTP/3 using QUIC. It has been implemented in Apple's iCloud Private Relay, which uses a two-hop architecture to decouple user identity from destination servers [2]. Probst [1] presents a MASQUE-based proxy prototype for lower OSI-layer traffic, illustrating the flexibility of MASQUE for privacy-preserving proxying.

Kühlewind et al. [6] published a paper about the performance of MASQUE without a TEE environment.

To date, no published work has examined the integration of MASQUE proxies with TEEs. This paper proposes such a combination, outlines the benefits for

 doi: 10.2313/NET-2025-11-3_14

privacy-preserving analytics, and identifies future research directions to realize this architecture in practice.

4. Applications of MASQUE-Proxies in TEE Environments

The integration of MASQUE proxies with TEEs presents a promising pathway for achieving secure and more private communication over the internet. TEEs, such as Intel SGX, can help eliminate the need to trust proxy providers blindly by providing a mechanism to verify the running TEE instance. The combination of MASQUE's effective multiplexed proxying via the QUIC protocol with the confidentiality assurances of TEEs enables a new way of secure proxy-based architectures.

This section investigates three particular domains of application in which TEE integration can be advantageous for MASQUE proxies.

First, we discuss in Section 4.1 how TEE can be used in way that traffic obfuscation techniques are used to improve user anonymity. Next, we examine in Section 4.2 how TEEs can enhance trust in systems like *iCloud Private Relay*. Finally, we analyze in Section 4.3 the performance trade-offs observed in TEE-based proxy implementations to understand their practical implications.

4.1. Confidential Proxying with Traffic Obfuscation

When browsing the internet, an adversary observing the network traffic can often determine which website a user is accessing. The primary purpose of a proxy is to conceal the user's intended destination, thereby enhancing privacy by preventing direct association between the user and the target website [1].

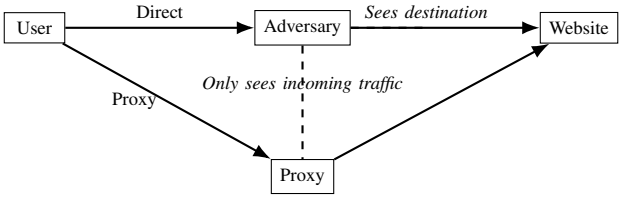

Figure 2: Using a proxy to hide the destination website from network observers

Ensuring user anonymity when using proxies is a critical requirement when using a proxy. One effective method to achieve this is through a technique known as *mixing* [7]. In this approach, data packets originating from multiple users are collected at an intermediate node. These packets are then shuffled and subjected to an intentional delay before being forwarded to the destination [7]. This randomized reordering and timing obfuscation significantly complicate the task of correlating input and output flows, both for the proxy provider and for a global passive adversary, thereby enhancing traffic-level privacy, as depicted in Figure 4.1.

A fundamental challenge in proxy based anonymization is the need to trust the proxy provider not to log incoming and outcoming traffic. As such logs would enable the correlation of network activity with a specific

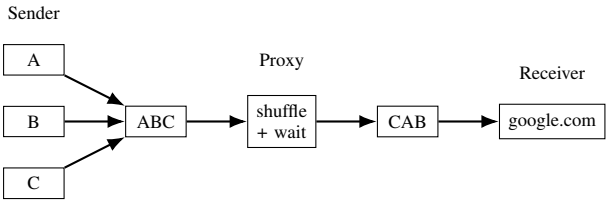

Figure 3: The mixing Workflow

users. This trust requirement can be mitigated through the use of TEEs combined with remote attestation. If the cloud provider supplies both the proxy source code and the corresponding TEE measurement, remote attestation can be employed to verify that the proxy instance is indeed running within a genuine, untampered TEE. An additional advantage of this approach is the strong isolation provided by TEEs. It becomes significantly more difficult for an adversary, even one with access to the same physical hardware, to escape their execution context and observe traffic processed by the TEE instance [3]. This strengthens the anonymity of the traffic obfuscation mechanism.

4.2. Enhancing Trust in iCloud Private Relay through TEEs

Another approach to enhancing privacy is known as *re-routing*, where traffic is relayed through one or more intermediary nodes [7]. Apple implements this concept in its proprietary service *iCloud Private Relay* [2], which uses the MASQUE protocol as the underlying proxying mechanism.

The core idea behind this method is to separate the knowledge of the data source and its destination. Apple receives the user's incoming traffic but does not see the final destination. Instead, it forwards the encrypted request to a third-party relay, which then delivers the data to the intended receiver [2]. This two-hop architecture ensures that no single entity has full visibility of both the sender and the receiver [1], [2].

However, this model relies on a degree of trust in both Apple and the third-party relay provider. Users must assume that neither party colludes or logs identifying information [2]. This is where *TEEs* can strengthen the trust model. If both relay providers executed their services inside a TEE environment, published their code and attestation measurements, users could independently verify that their data is handled as promised.

Unfortunately, this level of transparency is currently unrealistic. Apple does not typically open source its codebase, and Apple did not state anything about public attestation in their white paper [2]. Nonetheless, this idea highlights the potential role of TEEs in increasing accountability and verifiability in privacy-preserving relay systems.

4.3. Performance Implications of TEE-Based Proxy Implementations

Bouhairi et al. [5] investigate how the Eperi Proxy can be implemented using SCONE, a secure Docker container

 doi: 10.2313/NET-2025-11-3_14

framework based on Intel SGX technology. SCONE enables the creation of secure Dockerfiles from any Docker image and offers additional features such as filesystem shielding, network shielding and secure system calls [4]. Through this implementation, encryption and decryption operations are executed entirely within the secure enclave, ensuring both data confidentiality and integrity.

The Eperi Gateway shares similarities with MASQUE, which also emphasizes secure and confidential data transmission, but uses TCP/TLS over the QUIC protocol [5]. Therefore, several observations from the SCONE-based implementation may provide valuable insights for a potential MASQUE-based TEE deployment.

One of the main findings from their evaluation is a significant increase in latency, rising from 423 ms to 912 ms, more than double [5]. This latency loss is critical, especially for latency-sensitive applications such as *iCloud Private Relay*. Several factors contribute to this overhead, including cache flushes, the cost of integrity checks performed at each context switch, and the memory encryption overhead introduced by Intels SGX architecture [3].

Another key metric analyzed is throughput. While the TEE-based implementation achieves similar throughput levels to the non TEE implementation, it demands substantially more CPU resources, 910 % instead of 790 % at maximum throughput [5]. As a result, the SCONE system reaches a throughput ceiling at approximately 100req/s, due to near saturation of the available virtual CPU cores which will result in a latency jump [5].

5. Conclusion and Future Work

In conclusion, TEEs offer promising new opportunities to enhance the privacy guarantees of MASQUE-based proxy architectures. By isolating sensitive operations, like encryption and decryption, and enabling verifiable execution, TEEs could mitigate trust assumptions inherent in current systems. However, challenges remain, including susceptibility to denial-of-service (DoS) attacks and enclave resource exhaustion, which must be systematically addressed to ensure robust and practical deployments [3].

However, given that MASQUE is a relatively new protocol, there is currently little research on the integration of TEEs in practical MASQUE proxy deployments. Most available comparisons rely on alternative systems, such as the Eperi Gateway.

For future work, a prototype implementation of a MASQUE proxy within a TEE instance, using frameworks such as SCONE, could provide valuable insights into performance overheads and practical feasibility. Additionally, further investigation is needed into mechanisms that allow end users to verify that a MASQUE proxy is genuinely running inside a trusted TEE instance on a cloud provider, possibly through remote attestation.

Such developments could significantly increase the trustworthiness and transparency of next-generation privacy infrastructures.

References

[1] C. Probst, "Rust-based MASQUE-Proxying for Lower OSI Layer Traffic," Jul 2023, accessed: 2025-06-01. [Online]. Available: https://oc.net.in.tum.de/s/MsdcpDFrJ2ikHsa/download/thesis_probst.pdf

[2] Apple Inc., "iCloud Private Relay Overview," Whitepaper, Dec 2021, accessed: 2025-06-01. [Online]. Available: https://www.apple.com/icloud/docs/iCloud_Private_Relay_Overview_Dec2021.pdf

[3] M. Li, Y. Yang, G. Chen, M. Yan, and Y. Zhang, "Sok: Understanding design choices and pitfalls of trusted execution environments," in *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1600–1616. [Online]. Available: https://doi.org/10.1145/3634737.3644993

[4] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'Keeffe, M. L. Stillwell, D. Goltzsche, D. Eyers, R. Kapitza, P. Pietzuch, and C. Fetzer, "Scone: secure linux containers with intel sgx," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. USA: USENIX Association, 2016, p. 689–703.

[5] M. J. A. Bouhairi, M. Mullick, M. Wolf, I. Gudymenko, and S. Clauß, "Encryption Proxies in a Confidential ComputingEnvironment," feb 2023, accessed: 2025-06-08. [Online]. Available: https://wwwpub.zih.tu-dresden.de/~s0278016/publications/Encryption_Proxies_in_Conf_Comp_Environments.pdf

[6] M. Kühlewind, M. Carlander-Reuterfelt, M. Ihlar, and M. Westerlund, "Evaluation of quic-based masque proxying," in *Proceedings of the 2021 Workshop on Evolution, Performance and Interoperability of QUIC*, ser. EPIQ '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 29–34. [Online]. Available: https://doi.org/10.1145/3488660.3493806

[7] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes, and C. Diaz, "A survey on routing in anonymous communication protocols," *ACM Comput. Surv.*, vol. 51, no. 3, Jun. 2018. [Online]. Available: https://doi.org/10.1145/3182658