Internet-Draft
Intended Status: Experimental
Expires: June 2026

Z. Eli

December 2025

StealthFlow Protocol (SFP)
draft-eli-stealthflow-protocol-00

Abstract

The StealthFlow Protocol (SFP) is a low-observability, stateless pre-authentication and transport armor mechanism designed to precede existing TLS or QUIC sessions. SFP aims to reduce handshake fingerprinting, limit asymmetric denial-of-service amplification, and minimize early plaintext metadata exposure, while preserving compatibility with the existing PKI and TLS ecosystems.

SFP does not replace TLS or QUIC. Instead, it provides an optional single-round pre-filtering and authentication layer that enables servers to reject illegitimate traffic with minimal computational cost and minimal network observability.

This document supersedes the prior informal specification known as draft-eli-z-protocol-00.

Table of Contents

1. Introduction
2. Design Goals
3. Protocol Overview
4. Packet Structure
5. Processing Model
6. Security Considerations
7. Privacy Considerations
8. IANA Considerations
9. Future Work
10. References
Author’s Address

1. Introduction

Modern TLS and QUIC deployments expose observable handshake patterns early in the connection lifecycle. Even with encrypted handshakes, initial packets often reveal protocol versioning, structural fingerprints, and asymmetric cost characteristics exploitable for traffic analysis and denial-of-service attacks.

In particular, attackers may generate large volumes of syntactically valid but unauthenticated initial packets at negligible cost, while forcing servers to perform comparatively expensive cryptographic or stateful operations.

The StealthFlow Protocol (SFP) introduces a stateless, single-round pre-authentication mechanism that allows servers to cheaply discard unauthenticated traffic while minimizing protocol observability. SFP operates strictly before and outside of TLS or QUIC, and does not alter their cryptographic or transport semantics.

2. Design Goals

SFP is designed with the following goals:

* Minimize early observable protocol fingerprints.
* Enable fail-silent rejection of unauthenticated traffic.
* Impose symmetric or unfavorable economics on large-scale abuse.
* Avoid introducing new trust anchors or modifying PKI behavior.
* Remain compatible with existing TLS and QUIC implementations.
* Allow stateless server-side processing at line rate.

3. Protocol Overview

SFP consists of a single client-to-server packet, optionally followed by a single server response.

The client sends an armored initial packet containing:

* A fixed guide code identifying SFP traffic
* A client-generated ephemeral public key
* A proof-of-work (PoW) commitment bound to a freshness value
* A whole-packet integrity hash
* A digital signature covering all prior fields
* Variable-length random padding

Upon successful verification, the server MAY respond with a single packet containing encrypted session bootstrap information. Upon failure, the server silently discards the packet.

No protocol negotiation, retransmission signaling, or persistent state is required.

4. Packet Structure

4.1. Client Initial Packet

The client initial packet contains the following fields, in order:

[Guide Code]
[Client Ephemeral Public Key]
[PoW Commitment]
[Freshness Value]
[Client Nonce]
[Packet Hash]
[Client Signature]
[Random Padding]

The Packet Hash MUST cover all preceding fields except padding.

The Client Signature MUST cover the Packet Hash.

4.2. Server Response Packet

If the server elects to respond, it sends:

```
 [Server Ephemeral Public Key]
     [Encrypted Payload]
     [Response Hash]
     [Server Signature]
```

The Encrypted Payload is encrypted using the client ephemeral public key and contains the server certificate chain and session parameters necessary to bootstrap a subsequent TLS or QUIC connection.

5. Processing Model

Server processing MUST prioritize inexpensive validation steps:

1. Verify the guide code.
2. Validate packet structure and length bounds.
3. Verify the packet hash and client signature.
4. Validate the PoW commitment.
5. Validate freshness within a configurable acceptance window.

Packets failing any step MUST be silently discarded.

Freshness validation SHOULD allow for network latency and clock skew. Typical deployments are expected to permit windows on the order of seconds rather than milliseconds.

After successful validation, servers MAY proceed with TLS or QUIC session establishment using conventional mechanisms.

6. Security Considerations

SFP reduces exposure to fingerprinting by avoiding explicit negotiation fields and minimizing deterministic structure in initial packets.

Proof-of-work commitments increase the computational cost of large-scale abuse while remaining negligible for legitimate clients.

Whole-packet signatures and hashes provide integrity protection against active manipulation.

SFP does not protect against compromised endpoints, compromised PKI, or adversaries capable of global traffic correlation.

7. Privacy Considerations

SFP is designed to minimize linkability across connections by using ephemeral keys and nonces. No stable client identifiers are exposed on the wire.

Freshness values and PoW commitments MAY introduce coarse temporal correlation. Implementations SHOULD avoid excessively narrow acceptance windows to reduce unintended fingerprinting.

Silent packet drops MAY still be observable through side channels such as timing or path MTU behavior.

8. IANA Considerations

This document has no IANA actions.

9. Future Work

Future revisions may include:

* Formal wire encodings (e.g., CBOR)
* Mandatory-to-implement cryptographic algorithms
* Performance measurements and test vectors
* Integration guidance for common TLS and QUIC stacks

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

10.2. Informative References

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.

[RFC9000] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, May 2021.

Author’s Address

Z. Eli
Email: li.xiaoming@tutamail.com