Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 21 June 2026

Z. Eli

draft-eli-z-protocol-00

December 2025

Z-Protocol: Zero-Handshake Adaptive Proof-of-Work Encrypted Transport Protocol

Status of This Memo

Copyright Notice

Abstract

This document describes Z-Protocol, an experimental transport-layer protocol that establishes secure communication without a visible handshake. Z-Protocol combines asymmetric cryptography, adaptive proof-of-work admission control, and stateless packet processing to mitigate denial-of-service attacks and reduce observable protocol metadata during connection establishment.

Table of Contents

1. Introduction

Traditional secure transport protocols rely on explicit handshake exchanges to negotiate cryptographic parameters. These handshakes introduce additional round trips and create protocol-visible metadata that may be observable by on-path entities.

Z-Protocol explores an alternative approach in which the initial client packet simultaneously performs authentication, key establishment, and data transmission without a prior negotiation phase.

2. Design Goals

Z-Protocol is designed with the following goals:

* Eliminate explicit handshake round trips
* Provide early application data (0-RTT)
* Mitigate denial-of-service attacks at minimal server cost
* Remain compatible with existing public-key infrastructures
* Support hardware-assisted packet filtering

3. Threat Model

Z-Protocol assumes the presence of passive and active on-path attackers, including traffic observers and packet injectors. Compromise of trusted certification authorities is considered out of scope, consistent with existing TLS threat assumptions.

4. Protocol Overview

A Z-Protocol connection begins with a single client packet containing encrypted session material and an admission token. The server processes the packet statelessly and replies with encrypted application data and optional credentials.

No cleartext handshake messages are exchanged.

5. Client Packet Construction

The client constructs an initial packet containing:

* A protocol identification marker
* A proof-of-work solution
* A compact root identifier hint
* An ephemeral session seed
* Optional application data

All sensitive fields are encrypted using the server’s public key.

6. Server Processing Model

Servers MAY deploy hardware or kernel-level filters to identify Z-Protocol packets based on the protocol marker.

Packets failing proof-of-work validation SHOULD be discarded prior to allocation of application-layer resources.

7. Adaptive Proof-of-Work

Z-Protocol employs an adaptive proof-of-work mechanism to regulate client access under load.

Servers MAY adjust acceptance thresholds dynamically. Clients experiencing non-response SHOULD increase proof-of-work effort on subsequent attempts.

8. Key Establishment

The session seed contained in the initial packet is used to derive symmetric encryption keys.

The seed MUST be unique per connection and MUST NOT be reused.

9. Replay Protection

Replay protection is achieved through a combination of ephemeral session seeds and server-side freshness checks.

Replayed packets MUST be rejected.

10. Transport Characteristics

Z-Protocol operates over an unreliable transport such as UDP. Reliability, if required, is provided by the application layer.

11. Deployment Considerations

Z-Protocol is intended for controlled deployments and experimental evaluation. It is not intended to replace TLS in general-purpose web traffic without further analysis.

12. Compatibility Considerations

Z-Protocol does not modify existing public-key infrastructures. Servers reuse existing certificate material for encryption.

13. Security Considerations

13.1. Forward Secrecy

Z-Protocol provides forward secrecy by deriving symmetric keys from ephemeral session seeds. Compromise of long-term keys does not expose past session data.

13.2. Denial-of-Service Resistance

Proof-of-work admission control limits asymmetric resource exhaustion and shifts computational cost to the initiating client.

13.3. Replay and Injection Attacks

Time-bound session material prevents successful replay of captured packets.

13.4. Post-Quantum Considerations

The protocol is compatible with post-quantum public-key algorithms, as the initial encryption mechanism can be replaced without

altering packet structure.

14. IANA Considerations

This document does not require any IANA actions.

15. References

15.1. Normative References

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

15.2. Informative References

[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