The OSTP Protocol

Ian M. Moffett and Quinn Stephens

February 26, 2025

1 Introduction

The OSMORA Secure Tunnel Protocol (OSTP) is a secure channel designed for reliable and confidential message exchanges among members of the OSMORA Project. Additionally, OSTP enables OSMORA Central Server (OCS) administrators to reliably notify the OCS of various states such as updates, alerts or panic conditions.

2 Packet Structures

2.1 Session Request

- **options**: Flag bits (U, R).
- **pubkey**: Ephemeral public key.
- **pad**: Random padding for obscuring length.

2.2 Field Descriptions

- **options (uint8_t)**: A single byte used to represent various options and flags for the Session Request. The bits are defined as follows:
  - Bit 0 (U): User authentication required (may be enforced).
  - Bits 1-7 (R): Reserved for future use, must be zero.
- **pubkey[32]**: The ephemeral public key generated by the client for the session. This key is used in the Elliptic-curve Diffie–Hellman (ECDH) exchange for creating a session key, enabling forward secrecy.
- **pad[8]**: Optional random padding added to obscure the length of the message. This padding can range from 8 to 32 bytes.

2.3 Parameter Arbitration Packet

- **spw (uint16_t)**: Session Parameter Word (see **Section** 3)
- **code (uint8_t)**: Status/error code (see subsection 3.1)

2.4 Message Frame

- **iv[AES_IV_SIZE: 16]**: 128-bit AES-256-GCM IV
- **tag[AES_GCM_TAG_SIZE: 16]**: 128-bit AES-256-GCM tag
- **len (uint16_t)**: Length of payload
- **payload[4096]**: Payload containing encrypted data

2.5 Management Frame

- **type (uint8_t)**: Management frame type
- **word (uint16_t)**: Frame control word (payload)

Table 1: PAP Codes

Code	Value	Description
PAP_SUCCESS	0x00	Indicates that the request was successful and processed correctly.
PAP_BAD_SPW	0x01	Indicates that the provided Session Parameter Word (SPW) is not supported by the server.
PAP_BAD_PERMS	0x02	Indicates that the SPW provided requires additional permissions that the client does not possess.
PAP_RESOURCE	0x03	Indicates that the server has insufficient resources to process the request.

2.6 Management Frame Types

Table 2: Types

Code	Value	Description
MFRAME_ACK	0x00	Acknowledgement
MFRAME_TRUNC	0x01	Truncate blocks

2.7 Operation

When a client connects to the OCS, it must first generate an X25519 ephemeral key-pair. Once the key-pair is generated, the client constructs a Session Request containing necessary options/flags and the ephemeral public key. Upon receiving the Session Request, the OCS generates its own X25519 ephemeral key-pair and responds by sending its ephemeral public key to the client. Both the client and the OCS compute a shared session key through an Elliptic-curve Diffie–Hellman key exchange. **All packets and data sent after this process are then encrypted with the session key and bundled within a message frame.**

The options field in the Session Request packet contains basic options to be used once the secure connection is established. Additional options known as Session Parameters may be negotiated with the OCS during the Session Parameter Negotiation stage.

2.8 User Authentication

If the U bit is set in the Session Request (refer to **Section** 2), the client must send a Session Authentication Packet (SAP) containing a username and password. The username can be up to 64 bytes while the password can be up to 256 bytes. The OCS responds by echoing the SAP with its code field set (see **Section** 5). If the code field indicates success, the client may proceed with Session Parameter Negotiation (see **Section** 3).

2.9 Message Frame

The message frame is a crucial structure in the OSTP protocol. Before any encrypted data (e.g., PAP payload, text, etc) can be sent to the server, it must be bundled within a Message Frame which contains various fields such as the AES-256-GCM IV, the AES-256-GCM tag and the payload itself. The maximum size of each payload can be up to 2^{12} bytes which is usually sufficient for most use cases. However, if an implementation requires larger payloads, message chaining can be used, in which each message is broken up into several frames. This would allow up to $2^{12} \cdot n$ bytes of data to be sent, where n is the number of frames used in a chain.

3 Session Parameter Negotiation

The client may need to apply server specific options/flags that are not available in the options/flags field within the Session Request (see subsection 2.1). This can be achieved by performing Session Parameter Negotiation which may involve Parameter Arbitration if necessary (see subsection 3.1). During the typical operation of the Session Parameter Negotiation phase, the client sends the server a Parameter Arbitration Packet (PAP) (see subsection 2.3) containing a Session Parameter Word (SPW). Within the SPW are 15 bits of options/flags with bit 15 being reserved for Quick Session Requests (SRQs) which allow the client to bypass the Session Parameter Negotiation phase all together.

3.1 Session Parameter Arbitration

If the client has sent a Session Parameter Word (SPW) with bits that the server refuses to accept, the server will propose an alternative SPW bundled in a Parameter Arbitration Packet (PAP) along with the PAP code field set to indicate why it was rejected. The client has the option to either reply with the proposed SPW or send a new PAP containing a different SPW.

4 Trusted Users List

The Trusted Users List contains usernames of those who are permitted to create a secure tunnel. For security reasons, it is not recommended to have the root user listed as a trusted user. The Trusted Users List can be as follows:

```
//
// OSMORA Trust List
//
// Be careful with "root"!
//

ian
dominik
quinn
// root
```

5 Constants

Table 3: PAP Codes

Code	Value	Description
PAP_SUCCESS	0x00	Indicates that the request was successful and processed correctly.
PAP_BAD_SPW	0x01	Indicates that the provided Session Parameter Word (SPW) is not supported by the server.
PAP_BAD_PERMS	0x02	Indicates that the SPW provided requires additional permissions that the client does not possess.
PAP_RESOURCE	0x03	Indicates that the server has insufficient resources to process the request.

Table 4: Session Authentication Codes

Code	Value	Description
AUTH_SUCCESS	0x00	Indicates successful authentication of the user.
AUTH_BAD_PW	0x01	Indicates that the provided password is incorrect.

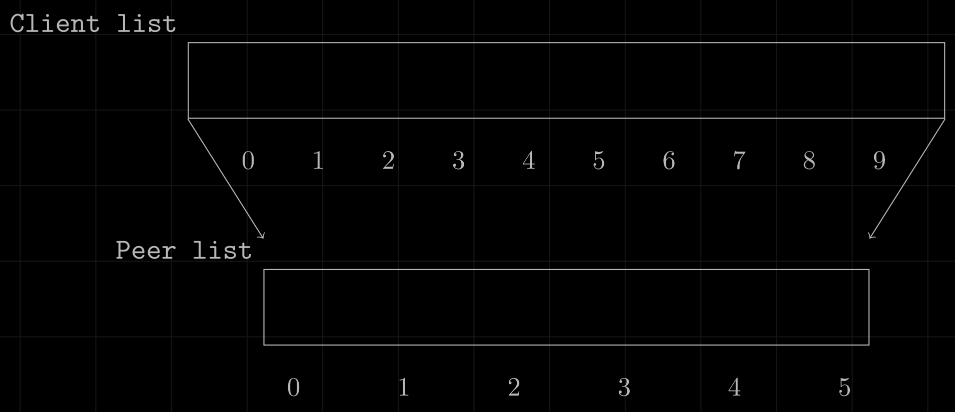
6 Peer-to-peer mode

The OSTP client list is a contiguous array of connected clients that includes the server and the current client. The OSTP peer list is a `PB_N_PEERS` sized array of available peers (where `PB_N_PEERS` is usually 16 peers per block). When a client transmits a session request with the 'P' (peer-to-peer) bit set within its `option` field (refer to subsection 2.1), the OCS responds by sending one or more peer blocks in a consecutive manner. Each block may list up to `PB_N_PEERS` peers.

6.1 Structure of a Peer

- **pad (uint8_t[1])**: Must be all ones, treat invalid otherwise.
- **port (uint16_t)**: Port the peer accepts connections on.
- **host (char[46])**: Peer IP address.

Figure 1: Peer block



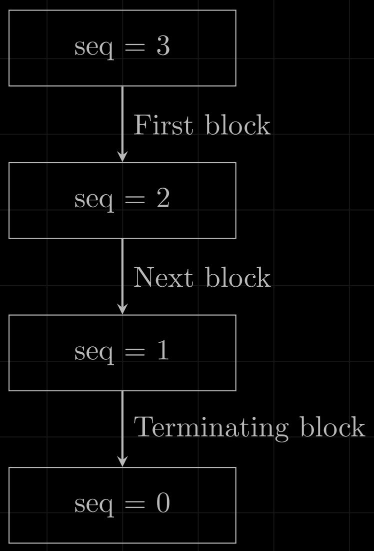
6.2 Structure of a Peer Block

- **peers (struct peer[PB_N_PEERS])**: List of available peers within this block. Where, PB_N_PEERS must be a power-of-two.
- **seq (uint8_t)**: Descending one-based block sequence number. A value of zero is invalid.

6.3 Peer Block Sequence

The `seq` field within the peer block structure represents the sequence. The sequence number being greater than one indicates that there are more peer blocks incoming from the OCS. The client may request the next peer block by responding with an ACK management frame. If the client no longer wants to receive peer blocks, it may respond with a TRUNC management frame (refer to subsection 2.5).

Figure 2: Sequence order



A sequence value of zero indicates the end of the block list