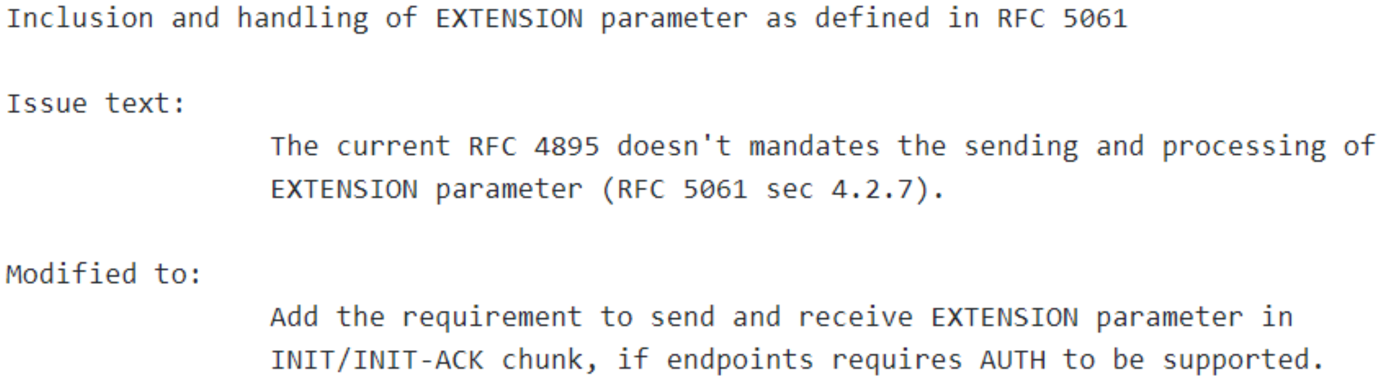
Changes to SCTP RFC 4895 - bis

AUTH Negotiation Procedure

- RFC 4895 has no section explaining the negotiation procedure.
- Upper 2 bits of new chunks (RANDOM, CHUNKS & HMAC-ALGO) is 10, so the receiver if doesn’t support, is free to ignore AUTH procedure and establish the SCTP association.
- If client supports AUTH and server doesn’t, then client end assumes the AUTH is negotiated successfully. Moreover if client has added DATA and SACK chunk to be sent in authenticated way, then all the data sent by server would be dropped by the client.
- Eventually connection closes without any meaning application data exchanged between the peers.
- Socket API extensions as defined in RFC 6458, enables client to know if AUTH is negotiated successfully. Not all implementations would implement this extension.
- Negotiation can be achieved without additional chunks/parameters and both the endpoints can detect it by the time INIT/INIT-ACK is successfully negotiated among them.

Adding Supported EXTENSION parameter (RFC 5061)



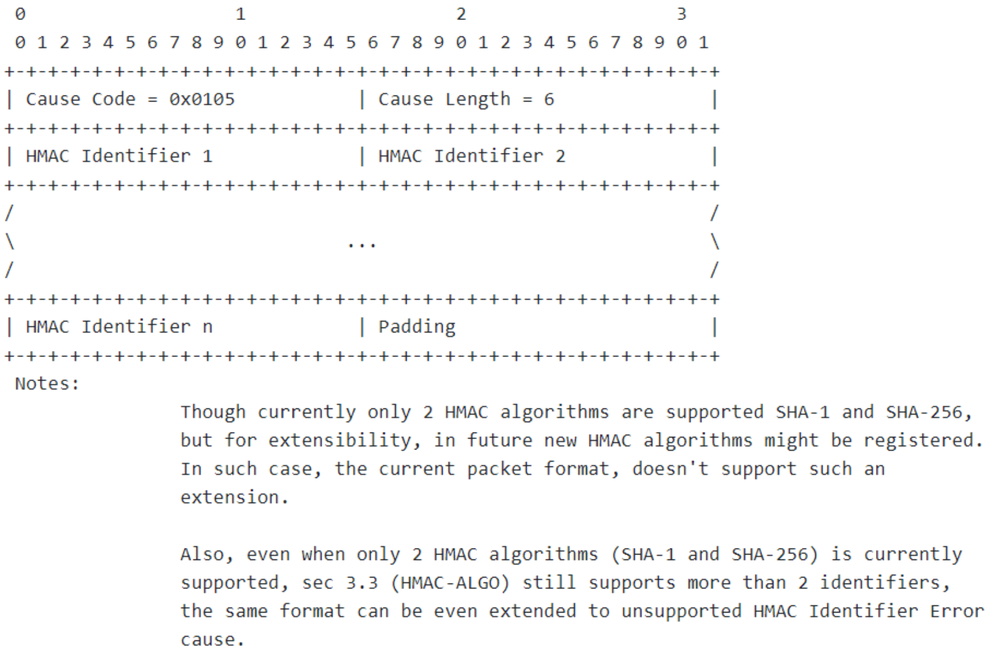
```
Inclusion and handling of EXTENSION parameter as defined in RFC 5061

Issue text:
                The current RFC 4895 doesn't mandates the sending and processing of
                EXTENSION parameter (RFC 5061 sec 4.2.7).

Modified to:
                Add the requirement to send and receive EXTENSION parameter in
                INIT/INIT-ACK chunk, if endpoints requires AUTH to be supported.
```

- lksctp expects the client to include this parameter even when only RFC 4895 is supported without the support for RFC 5061. FreeBSD doesn’t have any such restriction.
- RFC 4895 was standardized before RFC 5061, which introduced supported EXTENSION parameter. Adding the same to this would make this RFC consistent with others.
- This parameter MAY be added to enable interworking with all the implementations available.

Unsupported HMAC Identifier Error Cause

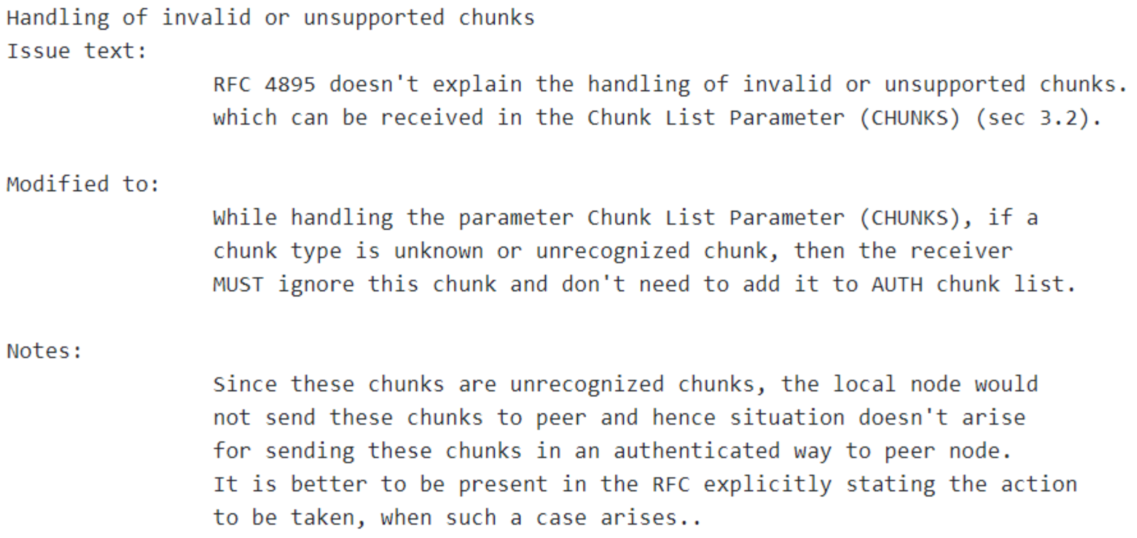


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cause Code = 0x0105           | Cause Length = 6              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| HMAC Identifier 1             | HMAC Identifier 2             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               /
\                              ...                              \
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| HMAC Identifier n             | Padding                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 Notes:
               Though currently only 2 HMAC algorithms are supported SHA-1 and SHA-256,
               but for extensibility, in future new HMAC algorithms might be registered.
               In such case, the current packet format, doesn't support such an
               extension.

               Also, even when only 2 HMAC algorithms (SHA-1 and SHA-256) is currently
               supported, sec 3.3 (HMAC-ALGO) still supports more than 2 identifiers,
               the same format can be even extended to unsupported HMAC Identifier Error
               cause.
```

- HMAC-ALGO chunk request supports n HMAC identifiers but the error supports only one, making the error format consistent with request would thus enable adding any new HMAC variants in future.
- SHA-1 which is currently marked as mandatory is no more recommended and used. By default, all implementations switches to SHA-256 as default. This can also be updated in the specification.

Explanation regarding handling of chunks which invalid/unknown to receiver



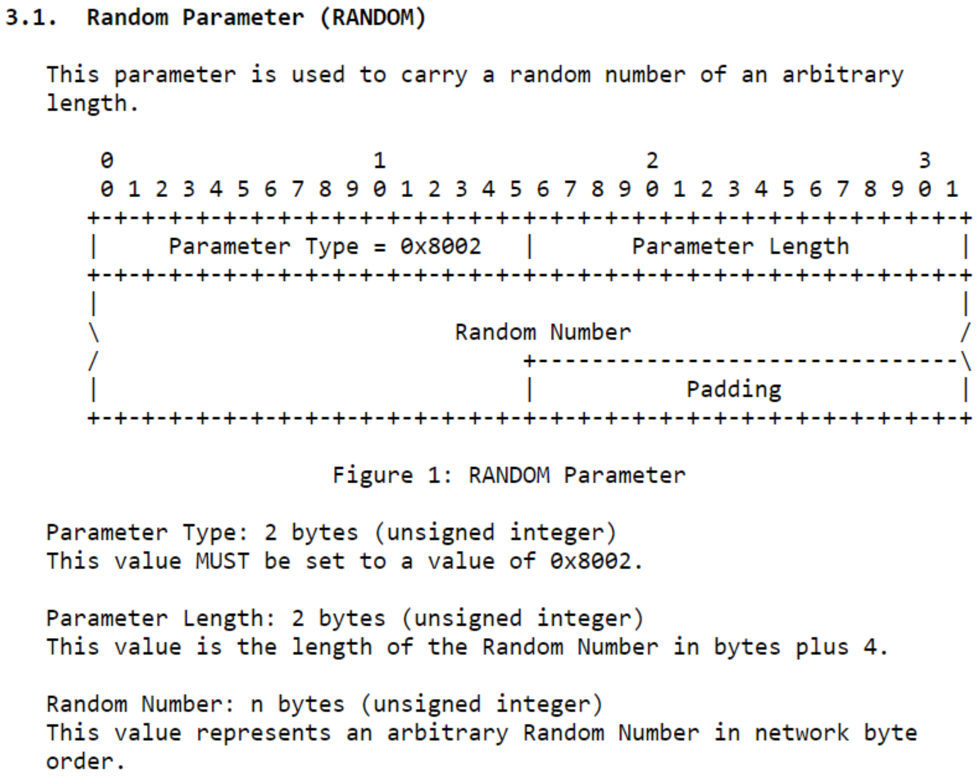
```
Handling of invalid or unsupported chunks
Issue text:
                RFC 4895 doesn't explain the handling of invalid or unsupported chunks.
                which can be received in the Chunk List Parameter (CHUNKS) (sec 3.2).

Modified to:
                While handling the parameter Chunk List Parameter (CHUNKS), if a
                chunk type is unknown or unrecognized chunk, then the receiver
                MUST ignore this chunk and don't need to add it to AUTH chunk list.

Notes:
                Since these chunks are unrecognized chunks, the local node would
                not send these chunks to peer and hence situation doesn't arise
                for sending these chunks in an authenticated way to peer node.
                It is better to be present in the RFC explicitly stating the action
                to be taken, when such a case arises..
```

- If the client implements RFC 6525 but the server hasn't supported it yet and client adds the parameter RE-CONFIG parameter in the CHUNKS list as defined in RFC 4895, then the client needs to ignore this parameter and need not add this parameter to authenticated chunk list.
- An explicit statement explaining the handling in such a scenario is required.

Explanation of RANDOM Parameter



3.1. Random Parameter (RANDOM)

This parameter is used to carry a random number of an arbitrary length.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Parameter Type = 0x8002   |       Parameter Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
\                          Random Number                        /
/                               +-------------------------------\
|                               |           Padding             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: RANDOM Parameter

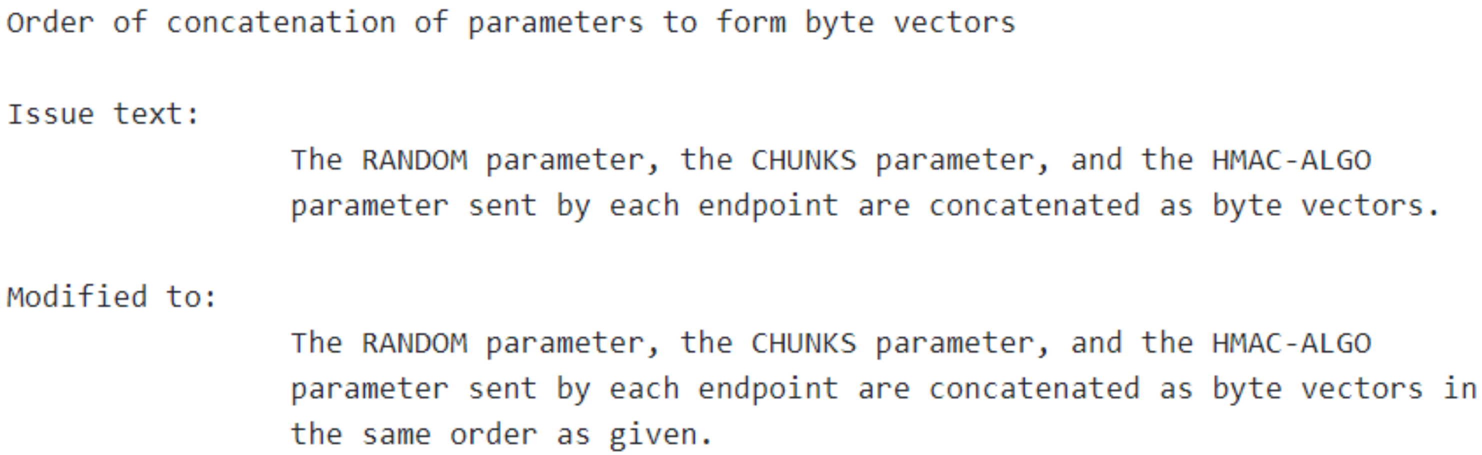
Parameter Type: 2 bytes (unsigned integer)
This value MUST be set to a value of 0x8002.

Parameter Length: 2 bytes (unsigned integer)
This value is the length of the Random Number in bytes plus 4.

Random Number: n bytes (unsigned integer)
This value represents an arbitrary Random Number in network byte order.

The RANDOM parameter MUST be included once in the INIT or INIT-ACK chunk, if the sender wants to send or receive authenticated chunks, to provide a 32-byte Random Number. For 32-byte Random Numbers, the Padding is empty.

Formation of Byte Vectors



Order of concatenation of parameters to form byte vectors

Issue text:

The RANDOM parameter, the CHUNKS parameter, and the HMAC-ALGO parameter sent by each endpoint are concatenated as byte vectors.

Modified to:

The RANDOM parameter, the CHUNKS parameter, and the HMAC-ALGO parameter sent by each endpoint are concatenated as byte vectors in the same order as given.