The Implementation of Secure Socket SCTP

Nicklas Hasselström and Gunnar Hjern
Department of Mathematics and Computer Science
Karlstad University, Sweden
nickhass100@student.kau.se
gunnhjer100@student.kau.se

Introduction

- **Secure Socket SCTP (S²-SCTP) is an End-to-End security solution for the SCTP protocol.**
- **This poster provides an overview of the involved protocols and the implementation of S²-SCTP.**
- **The implementation wraps around the Berkley sockets API and use TLS for authentication, see Figure 1.**

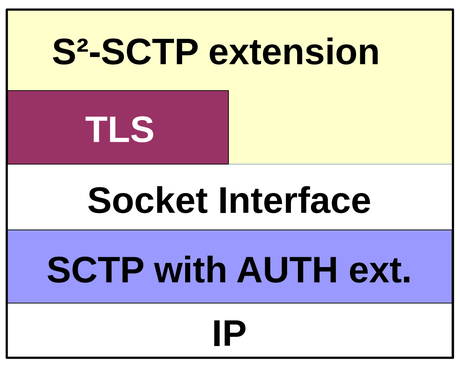

Figure 1. The S²-SCTP layer structure

Protocols

- **The data format protocol governs how the cryptographic info, IV and data padding is laid out. This was established by Lindskog and Brunstrom, 2007.**
- **The key and algorithm numbers (ref. to as “crinfo” field) constitutes the first 4 bytes of the payload field of each message. The crinfo-field is always present whether encryption is applied or not.**
- **If the encryption algorithm needs an initialization vector (IV), this IV is added explicitly to each message.**
- **If the encryption algorithm used is a block cipher, it is padded to an even block length.**
- **The length of IV- and padding-fields are dependent on the encryption algorithm used.**
- **The last byte of the padding field holds the padding length. This byte must always be present if padding is used, hence the padding is always 1 to blockLen bytes long, see Figure 2.**

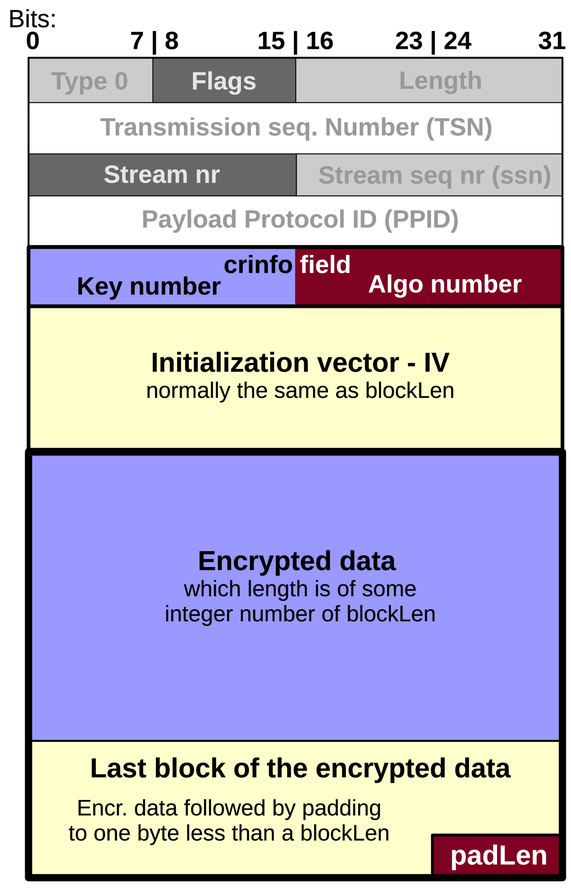

Figure 2. An S²-SCTP data chunk with chunk header (gray text and backgr.) and data part with crinfo, IV, encrypted data and padding.

Protocols cont.

- **To handle S²-SCTP management tasks, notably the peer authentication and key exchange, there is a newly developed management protocol.**
- **The management protocol messages are sent over stream 0, and have a length header value format, see Figure 3.**
- **Reception of these messages and the actions connected to them are handled internally by the receiving function.**

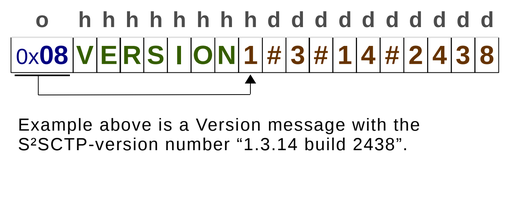

Figure 3. The length header value format of the S²SCTP management protocol

Implementation

- **The library is written in the C programming language for the FreeBSD operating system.**
- **An object oriented approach is used, see Figure 4.**
- **Flexibility is provided by user defined callback routines.**
- **Memory allocation and CPU utilization are minimized by using message object pools and stored key evolutions.**

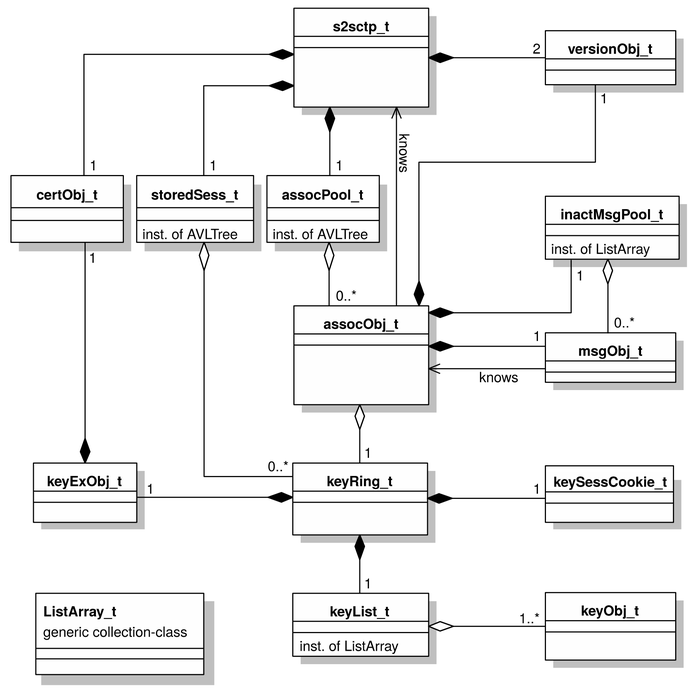

Figure 4. UML-diagram showing the object model of S²-SCTP. Collection objects are foremost realized as instantiations of the ListArray generic container class.

Conclusion and Future Work

- **The S²-SCTP library contains all the functionality needed for secure SCTP communication. Its only dependency is the Berkleys sockets API, and users are free to chose key exchange and encryption mechanism.**
- **In the future, S²-SCTP could easily be improved by adding support for:**
  - **Fragmented messages and partial reliability.**
  - **Multihoming**
  - **The One-to-Many SCTP socket model.**
  - **Stored authentication and keyExchange sessions**

Read the full report at:
http://www.cs.kau.se/proj/s2sctp