

A Comparison of End-to-End Security Solutions for SCTP

Stefan Lindskog

Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology, Trondheim, Norway
Email: stefan.lindskog@q2s.ntnu.no

Anna Brunstrom

Department of Computer Science
Karlstad University, Karlstad, Sweden
Email: anna.brunstrom@kau.se

Abstract—A comparison of three different end-to-end security solutions for the stream control transmission protocol (SCTP) is presented in this paper. The compared solutions are SCTP over IPsec, TLS over SCTP, and secure socket SCTP (SS-SCTP). The two former are standardized solutions, whereas the latter is a newly proposed solution that was designed to offer as much security differentiation support as possible using standardized solutions and mechanisms. The comparison focuses on three main issues: packet protection, security differentiation, and message complexity. SS-SCTP compares favorably in terms of offered security differentiation and message overhead. Confidentiality protection of SCTP control information is, however, only offered by SCTP over IPsec.

I. INTRODUCTION

Stream control transmission protocol (SCTP) [1] is a rather new full-fledged transport protocol with a rich set of functionalities. SCTP provides transport layer multihoming for enhanced network fault tolerance. Multistreaming is included to reduce the impact of head-of-line blocking of unrelated data. Furthermore, the provided transport service is message-oriented, and support for both ordered and unordered delivery of messages is offered. Messages in SCTP are transmitted as chunks. The same mechanism is also used to transfer SCTP control information. Through the chunk concept, SCTP is also easily extendable. Until today, four protocol extensions have reached a standard status [2–5].

Enhanced protection against denial-of-service (DoS) attacks is provided through a four-way handshake mechanism. Protection of message content is, however, not provided by SCTP. Such protection, which is referred to as end-to-end (E2E) security, must instead be implemented either above or beneath SCTP. SCTP over IPsec [6], TLS over SCTP [7], and secure socket SCTP (SS-SCTP) [8] are three proposed E2E security solutions, which are compared with respect to packet protection, security differentiation, and message complexity in this paper. SS-SCTP provides the most fine grained security differentiation and produces either less or similar message overhead depending on the traffic pattern. Only SCTP over IPsec, however, provides confidentiality protection of SCTP control information. In addition to the three studied E2E solutions, two other non-standardized solutions have been proposed [9, 10]. Due to space limitations, the comparison in the paper only considers the standardized solutions and

the new SS-SCTP. A brief qualitative comparison of all five solutions are, however, given in [8].

The remainder of the paper is organized as follows. A brief summary of the three compared E2E security solutions is given in Section II. In Section III, packet protection support, security differentiation support, and message complexity is compared for the three studied solutions. Concluding remarks is given in Section IV.

II. E2E SECURITY SOLUTIONS

A brief summary of the three studied E2E security solutions is given in this section. In common for all three solutions is that they offer both data integrity and data confidentiality protection of data in transit.

A. SCTP over IPsec

The use of IPsec to secure E2E SCTP traffic is described in [6]. IPsec [11] operates on the network layer and was designed to be independent of the transport protocol. The current version of IPsec has evolved to a fairly complex protocol. IPsec could in fact be seen as a number of protocols. When only integrity protection is needed the authentication header (AH) protocol [12] is used. If both integrity and confidentiality protection is needed, the encapsulating security payload (ESP) protocol [13] is used instead. To be able to dynamically establish security associations (SAs), a protocol referred to as the Internet key exchange (IKE) protocol [14] might be used.

When SCTP is used over IPsec all messages belonging to the same SCTP association are protected in the same way. Hence, no security differentiation may be applied at runtime. This is mainly due to the fact that IPsec is neither aware of SCTP messages nor of SCTP streams. Securing all data transferred between two hosts may produce unnecessary computational burden at the endpoints.

B. TLS over SCTP

Another solution to implement E2E security is through the use of TLS over SCTP as described in [7]. TLS is a byte-oriented protocol that requires an underlying reliable and in-sequence delivery service, which is offered by SCTP as long as it does not use the unordered delivery service or the extension for partially reliable transport available in the protocol.

Application data transfers are made over a TLS connection. Such a connection must be bidirectional. TLS provides key negotiation and endpoint authentication, data integrity through hashed message authentication codes (HMACs), and/or data confidentiality through encryption and decryption algorithms. The most serious disadvantage with TLS over SCTP is that SCTP control chunks exchanged between two peers are completely unprotected. This is caused by the fact that TLS operates on a layer above SCTP and is thus not aware of the control chunks.

C. SS-SCTP

The third solution described in this paper is SS-SCTP, which was initially proposed in [8]. SS-SCTP aims to offer a high degree of security differentiation based on features in the base SCTP protocol as well as in standardized extensions. The flexible message concept provided in the base protocol plays a central role in the design of SS-SCTP.

SS-SCTP implements E2E security at different communication layers. Key negotiation and authentication is implemented through TLS [15]. Data integrity protection is provided through the authenticated chunks extension [4], which among other things defines a new chunk type called AUTH. The AUTH chunk is used to carry a HMAC produced using either secure hash algorithm number 1 (SHA-1) or SHA-256. Data confidentiality, finally, is provided through encryption and decryption at the socket layer using the OpenSSL crypt library [16].

III. COMPARISON

A comparison of the three different E2E security solutions described in the previous section is given below. A description of SCTP packets is, however, first provided. Then, packet protection and security differentiation are compared, followed by a comparison of the message complexity when transferring user data using the different solutions.

A. SCTP packets

As mentioned in the introduction, SCTP is message-oriented and supports framing of individual messages. Chunks are the concept used to frame messages within SCTP packets. Both control and user data are transferred as chunks. Furthermore, one or more messages may be bundled into the same SCTP packet. An example of an SCTP packet with two control chunks and three data chunks is illustrated in Fig. 1.

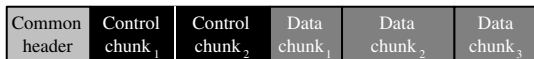


Fig. 1. An example of an SCTP packet with the mandatory SCTP common header (light grey), two control chunks (black), and three data chunks (dark grey).

As can be seen in Fig. 1, a common header is placed first in an SCTP packet. After the common header, the control chunks are included followed by as many data chunks as will fit in the packet. The number of chunks that can be carried

within the same SCTP packet is limited by the path maximum transmission unit (PMTU). If a single message is larger than the PMTU minus the IP header and the common header, SCTP will fragment the message and send the fragments within multiple SCTP packets.

B. Packet Protection and Security Differentiation

When protecting an SCTP packet, additional data is added. Depending on the chosen E2E security solution data is added at different places. Fig. 2 depicts the IP version 4 (IPv4) datagrams that result when transferring the SCTP packet introduced in Fig. 1 in five different protection scenarios. In (a), no protection at all is applied. In scenario (b) and (c), IPsec is used with the AH and ESP protocols, respectively. Scenario (d) illustrates when TLS is used, and, finally, in (e) SS-SCTP is used.

As is illustrated in Fig. 2, IPsec provides two different protection modes. If data integrity protection is sufficient, an AH is added to each IP datagram. If both data integrity and confidentiality protection is needed, an ESP header is used instead. When the choice of protection type has been made, all SCTP packets are protected in the same way, i.e., an all-or-nothing approach is applied to all SCTP chunks for both user and control data.

When TLS is used to protect data transferred over SCTP, only user data is protected as can be seen in Fig. 2. This is due to the fact that the data protection algorithms are applied to data above the SCTP layer, which implies that SCTP control data is transferred unprotected. For user data, security differentiation could be conducted on a per-stream basis.

In SS-SCTP, both user and control chunks could be integrity protected using the AUTH extension. The selection of which chunk types to integrity protect is decided on a per-association basis. In the example given in Fig. 2, it is assumed that all illustrated control chunks and data chunks are integrity protected. Data confidentiality protection in SS-SCTP is provided through encryption and decryption implemented at the socket layer. Encryption is thus only applied on user data and could be selectively applied on a per-message basis.

Comparing the protocols, SCTP over IPsec provides the strongest protection, and SS-SCTP the most flexible security differentiation. Note also that all three E2E security solutions compared in this paper allow security differentiation through algorithm selection, i.e., selection of key exchange algorithm, HMAC algorithm, and encryption algorithm.

C. Message Complexity

A comparison of the message complexity when transferring user data using SCTP over IPsec, TLS over SCTP, and SS-SCTP is emphasized in this section. We consider cases when user messages are unbundled as well as bundled. For data integrity and confidentiality protection, the SHA-1 and the advanced encryption standard (AES) algorithm in cipher block chaining (CBC) mode with a key length of 128 bits are assumed. Since the focus is on the message overhead introduced when transferring user data, the secure association

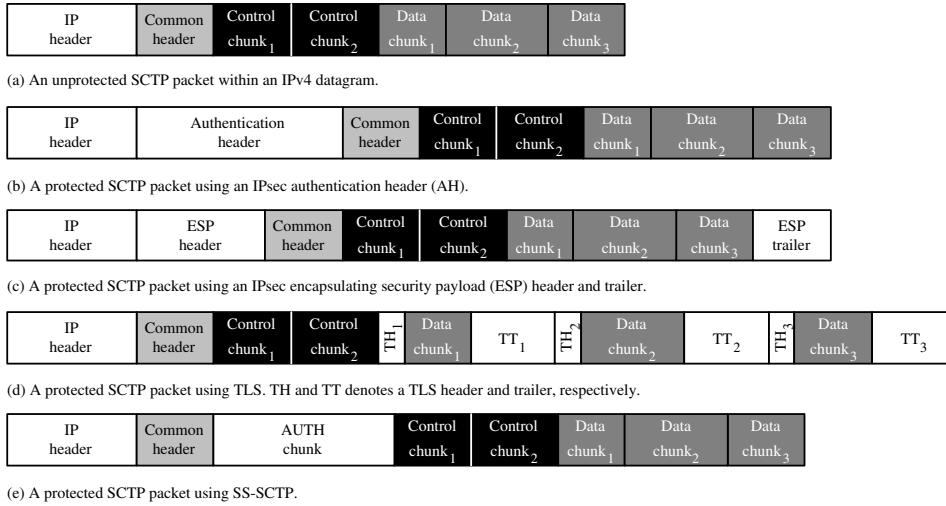


Fig. 2. An illustration of the extra overhead added when using (b) the IPsec AH protocol, (c) the IPsec ESP protocol, (d) TLS, and (e) SS-SCTP to protect the SCTP packet in (a). The extra headers when transferring the SCTP packet in Fig. 1, including the IPv4 header, are illustrated in white.

establishment is not considered. For the same reason, re-authentication and re-keying are not considered either. These events are also infrequent in comparison to data transfers.

1) *Unbundled Data Transfers*: Sending unbundled user messages means that each SCTP packet only carries a single user message. Such transfers will therefore result in a higher message overhead compared to a bundled transfer, simply due to the fact that some protocol headers must be duplicated in each SCTP packet. Unbundled data transfers are typically used to achieve as low latency as possible, which may be a requirement in signaling and multimedia applications.

A summary of the message complexity for the three different investigated E2E security solutions when transferring user messages unbundled is provided in Table I. The figures in the table include all higher layer protocol headers starting with the IP layer and a user message of the specified size. IPv4 is assumed to be used here. Hence, when SCTP over IPsec is used, an IPv4 header (20 bytes), an IPsec header/trailer (32 bytes in total independent of if AH or ESP is used), an SCTP common header (12 bytes), and a DATA chunk header (16 bytes), and as much of the user message that fits is included in each SCTP packet. Similarly, TLS over SCTP includes an IPv4 header (20 bytes), an SCTP common header (12 bytes), a DATA chunk header (16 bytes), and as much of the user message that fits in each SCTP packet. In this case, the user message contains both the payload data, the TLS header (5 bytes for type, version, and length) and the TLS trailer (23 bytes for HMAC and padding). Finally, when SS-SCTP is used, an IPv4 header (20 bytes), an SCTP common header (12 bytes), an AUTH chunk (28 bytes), and a DATA chunk header (16 bytes), and as much of the user message that fits is included in each SCTP packet.

As can be seen in the Table I, SS-SCTP and TLS over SCTP produce the same amount of traffic for user messages that fit in a single SCTP packet. When message fragmentation is needed, i.e., when transferring user messages that do not

TABLE I
MESSAGE COMPLEXITY EXPRESSED IN BYTES FOR DIFFERENT SIZED USER MESSAGES WHEN TRANSFERRED UNBUNDLED USING IPV4. THE PMTU IS ASSUMED TO BE 1500 BYTES.

E2E security solution	User message size in bytes				
	128	256	1024	4096	16384
SCTP over IPsec	208	336	1104	4336	17344
TLS over SCTP	204	332	1100	4268	16988
SS-SCTP	204	332	1100	4324	17296

fit in a single SCTP packet, TLS causes slightly less traffic compared to SS-SCTP. From the table it is also evident that SCTP over IPsec results in the highest amount of network traffic although the difference with SS-SCTP is small.

In Fig. 3, the message overhead added for unbundled data transfers by the different solutions are depicted. The message overheads for all solutions are high when transferring small user messages, but rather small when transferring large messages.

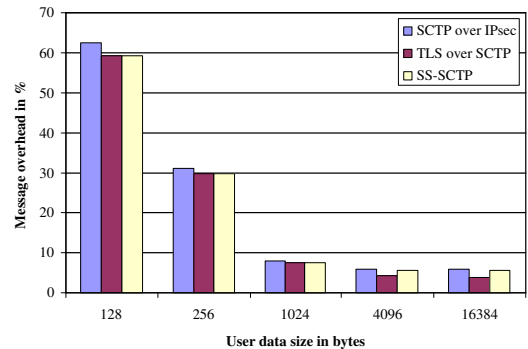


Fig. 3. Message overhead for different sized user messages.

2) *Bundled Data Transfers*: When bundled data transfers are used, multiple user messages could potentially be carried by a single SCTP packet. Bundling is, however, only used when multiple chunks are ready to be sent and fit completely in a single SCTP packet. If that is not the case, each message is sent in one or more SCTP packets depending on the size of the data. By using bundling, the message overhead is reduced. Bundled data transfers could with advantage therefore be used by, e.g., notification and data log services. In such services, messages exchanged are typically small, i.e., a few hundred bytes. Slightly increased latency is also typically acceptable in such services.

The message complexity for the three studied E2E security solutions with different amount of message bundling is depicted in Table II. All messages are assumed to be of equal length, i.e., 256 bytes in this case. Such an assumption is fair when considering notification as well as data log services. When bundling is used, TLS causes the highest message complexity. This is due to the extra protocol overhead added by TLS for each message, i.e., the TLS header (5 bytes) and the TLS trailer (23 bytes) as described above. When TLS is used, five bundled user messages will therefore not fit into a single SCTP packet. Instead, the four first messages are sent in one SCTP packet and the fifth is sent in a separate one, assuming that no further messages arrive to the send queue before the second packet is sent. When comparing SS-SCTP and SCTP over IPsec, the message complexity is again similar.

TABLE II
MESSAGE COMPLEXITY EXPRESSED IN BYTES FOR DIFFERENT AMOUNT OF MESSAGE BUNDLING USING IPV4. EACH USER MESSAGE IS 256 BYTES AND THE PMTU IS ASSUMED TO BE 1500 BYTES.

E2E security solution	Number of bundled user messages				
	1	2	3	4	5
SCTP over IPsec	336	608	880	1152	1424
TLS over SCTP	332	632	932	1232	1564 ^a
SS-SCTP	332	604	876	1148	1420

^a For this case, two SCTP packets are needed to transfer these five user messages, since the total data size is greater than the PMTU.

Fig. 4 illustrates the high overhead introduced when TLS over SCTP is used to transfer bundled user messages. In the worst case illustrated in Fig. 4, i.e., five bundled messages, the overhead is more than twice as big as compared to SS-SCTP and SCTP over IPsec. This implies that TLS is not a preferable E2E security solution when transferring small messages that can be bundled together in SCTP packets. Finally, the difference between SS-SCTP and SCTP over IPsec with respect to message overhead is almost negligible.

IV. CONCLUDING REMARKS

Packet protection, security differentiation, and message complexity of three different end-to-end security solutions for SCTP have been compared in this paper. In conclusion, SCTP over IPsec offers the lowest degree of security differentiation, but the highest level of security. TLS over SCTP produces the

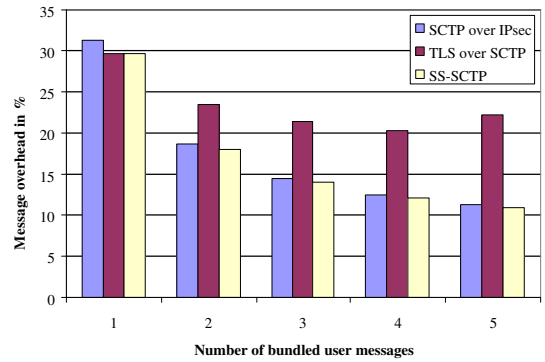


Fig. 4. Message overhead when user messages are bundled.

least communication overhead for large messages. Finally, SS-SCTP provides the finest degree of security differentiation and produces the least communication overhead when messages are bundled. The next step in the development of SS-SCTP is to finalize the prototype implementation, and then perform an experimental evaluation.

Acknowledgment: The work at Karlstad University is supported by grants from the Knowledge Foundations of Sweden with TietoEnator and Ericsson as industrial partners.

REFERENCES

- [1] R. Stewart, "RFC 4960: Stream control transmission protocol," September 2007.
- [2] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "RFC 3578: Stream control transmission protocol (SCTP) partial reliability extension," May 2004.
- [3] M. Tuexen, R. Stewart, and P. Lei, "RFC 4820: Padding chunk and parameter for the stream control transmission protocol (SCTP)," March 2007.
- [4] M. Tuexen, R. Stewart, P. Lei, and E. Rescorla, "RFC 4895: Authenticated chunks for stream control transmission protocol (SCTP)," August 2007.
- [5] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, and M. Kozuka, "RFC 5061: Stream control transmission protocol (SCTP) dynamic address reconfiguration," September 2007.
- [6] S. Bellovin, J. Ioannidis, A. Keromytis, and R. Stewart, "RFC 3554: On the use of stream control transmission protocol (SCTP) with IPsec," July 2003.
- [7] A. Jungmair, E. Rescorla, and M. Tuexen, "RFC 3436: Transport layer security over stream control transmission protocol," December 2002.
- [8] S. Lindskog and A. Brunstrom, "An end-to-end security solution for SCTP," in *Proceedings of the Third International Conference on Availability, Reliability and Security (ARES 2008)*, Barcelona, Spain, March 4–7, 2008, to appear.
- [9] E. Unurkhaan, E. P. Rathgeb, and A. Jungmair, "Secure SCTP: A versatile secure transport protocol," *Telecommunication Systems*, vol. 27, no. 2–4, pp. 273–296, 2004.
- [10] C. Hohendorf, E. P. Rathgeb, E. Unurkhaan, and M. Tüxen, "Secure end-to-end transport over SCTP," *Journal of Computers*, vol. 2, no. 4, pp. 31–40, June 2007.
- [11] S. Kent and K. Seo, "RFC4301: Security architecture for the Internet protocol," December 2005.
- [12] S. Kent, "RFC4302: IP authentication header," December 2005.
- [13] —, "RFC4303: IP encapsulating security payload (ESP)," December 2005.
- [14] C. Kaufman, "RFC4306: Internet key exchange (IKEv2) protocol," December 2005.
- [15] T. Dierks and E. Rescorla, "RFC 4346: The transport layer security (TLS) protocol version 1.1," April 2006.
- [16] OpenSSL homepage, "http://www.openssl.org/," November 30, 2007.