Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 9, 2020

K. Rose
Akamai Technologies, Inc.

July 08, 2019

Asymmetric Loss-Tolerant Authentication
draft-krose-alta-00

Abstract

Establishing authenticity of a stream of datagrams in the presence of multiple receivers is naively achieved through the use of per-packet asymmetric digital signatures, but at high computational cost for both senders and receivers. Timed Efficient Stream Loss-Tolerant Authentication (TESLA) instead employs relatively cheap symmetric authentication, achieving asymmetry via time-delayed key disclosure, while adding latency to verification and imposing requirements on time synchronization between receivers and the sender to prevent forgery. This document introduces Asymmetric Loss-Tolerant Authentication (ALTA), which employs an acyclic graph of message authentication codes (MACs) transmitted alongside data payloads, with redundancy to enable authentication of all received payloads in the presence of certain patterns of loss, along with regularly paced digital signatures. ALTA requires no time synchronization and enables authentication of payloads as soon as sufficient authentication material has been received.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Authenticity of streaming data may be inexpensively established via symmetric message authentication codes (MACs) using keys pre-shared exclusively between two parties, as the receiver knows it did not originate the data and that only one other party has access to the key. In the presence of multiple receivers, however, this is not possible because all receivers must have access to the same key, giving any one of them the ability to forge messages. Consequently, authentication must be made asymmetric, such that only the sender has the ability to produce messages that correct receivers will verify as authentic.

Naively, a sender may sign individual datagrams using an asymmetric digital signature algorithm, such as RSA or Ed25519, but this carries high computational cost for both the sender and receivers. In the case of streaming video delivery, while the sender's computational load may be dominated by CPU-intensive video encoding, the receiver is often a device with hardware dedicated to efficient video decoding and with limited general purpose computing hardware and/or battery available for high-rate digital signature authentication.

Timed Efficient Stream Loss-Tolerant Authentication (TESLA) [RFC4082] addresses this problem through the use of symmetric authentication by delaying the release of keying material to a deadline at which any packets protected by said key that are subsequently received must be discarded by a receiver. While this reintroduces asymmetry between sender and receiver, it requires the sender and each receiver to (loosely) synchronize clocks and imposes authentication latency relative to RTT and to a pre-declared upper bound on clock skew.

Clock synchronization is not as trivial as it appears: internet-connected hosts often have significant clock skew relative to stratum 0 NTP servers [timeskew], and anyway enterprises serving valuable assets do not regard NTP as a reliable interdomain security protocol. Together with the need to avoid attacks that delay packets required for synchronization, this implies the need for an interactive unicast authenticated clock synchronization protocol, which is complicated by the need to maintain clock synchronization across both the stream publisher and multiple geographically-distributed nodes in a content delivery network (CDN).

This document introduces Asymmetric Loss-Tolerant Authentication (ALTA), which eschews time synchronization for an application of digital signatures to an acyclic graph of symmetric message authentication codes with redundancy sufficient to tolerate certain patterns of loss, and with digital signature authentication load greatly reduced relative to the naive approach. This algorithm is based on research by Golle and Modadugu, as published in [STRAUTH]. Live multicast streaming over an unreliable transport is the intended application for ALTA: object-based integrity solutions or transport security may be more appropriate for unicast transmission or for static objects pulled on-demand.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Protocol Overview

ALTA is intended for streaming datagram use cases in which the receiving application has a deadline for the utility of received data and can tolerate a degree of random packet loss. It combines a segment of application data with a variable-length authentication tag into an ALTA payload to be sent as a unit in a single datagram, with the authentication tags constructed in such a way that a receiver will be able to authenticate nearly all such ALTA payloads received by the deadline under certain patterns of random packet loss.

An authentication tag is a combination of zero or more symmetric message authentication codes (MACs) and either zero or one digital signature. Each MAC is of another ALTA payload in the stream, while the digital signature is of the containing ALTA payload with the signature field itself replaced by all zeroes.

The MACs included in a given authentication tag are determined by a scheme, as defined in section 3 of [STRAUTH]. Conceptually, a scheme is a mostly backward-looking directed acyclic graph of ALTA payloads such that the MAC of a given payload is contained in two or more other payloads in the stream, enabling the loss of one of these to be tolerated without losing the ability to authenticate the given payload.

For purposes of illustration, a simple example scheme is one in which the ith ALTA payload's authentication tag contains MACs for the (i-1)th and (i-2)th payload:

```
                                                 .
                                                 .
                                                 .
   +-------------+------+                        |
   | payload i+1 | MACs |                        |
   |             |      |                        |
   |             |   i <---------------------+   |
   |             | i-1 <-----------------+   |   |
   +---------+---+------+                |   |   |
             |                    +----------+---------+
             +------------------->| MAC of payload i+1 |
                                  +--------------------+
   +-------------+------+                |   |
   | payload i   | MACs |                |   |
   |             |      |                |   |
   |             | i-1 <-----------------+   |
   |             | i-2 <-------------+   |   |
   +---------+---+------+            |   |   |
             |                    +--------+---------+
             +------------------->| MAC of payload i |
                                  +------------------+
   +-------------+------+            |   |
   | payload i-1 | MACs |            |   |
   |             |      |            |   |
   |             | i-2 <-------------+   |
   |             | i-3 <--------+    |   |
   +---------+---+------+       |    |   |
             |                  | +----+-+-------------+
             +------------------->| MAC of payload i-1 |
                                | +--------------------+
   +-------------+------+       |    |
   | payload i-2 | MACs |       |    |
   |             |      |       |    |
   |             | i-3 <--------+    |
   |             | i-4 <---+   |    |
   +---------+---+------+   |   |    |
             |              |   | ++-+-----------------+
             +------------------->| MAC of payload i-2 |
                            |   | +--------------------+
                            |   |
                            |   |
                            .   .
                            .   .
                            .   .
```

The recommended scheme is more complex and will be covered in detail in Section 4.3.

Encoding a scheme relies on ALTA payloads being addressable deterministically by an index even in the presence of reordering or loss. This index may be deduced from the application data (e.g., making use of an existing sequence number) or by a payload index explicitly encoded in the authentication tag. Two modes are supported:

- If the index starts at zero and increments by exactly one for each payload in the stream, and if the scheme is known to both sender and receiver, then indices are not required to be encoded for each MAC in an authentication tag as they can be deduced from a given payload's index and from the DAG associated with the scheme. Hereafter, this is referred to as _implicit offset mode_.

- If the index increments unpredictably, or if the scheme is not known to the receiver, then each MAC in an authentication tag must be paired with the explicit index of the ALTA payload from which the MAC is computed. For compactness, this index will be encoded as an offset relative to the index of the containing payload. Hereafter this is referred to as _explicit offset mode_.

Authenticity of a payload is established by a chain of MACs rooted in an ALTA payload whose authentication tag contains a digital signature created by a key in which trust has been established out-of-band. Delivery of application data must be delayed until a payload has been authenticated. Note that a given payload may be authenticated by a digital signature as well as by one or more MAC chains; within authentication deadline constraints, receivers should prefer to authenticate by MAC, minimizing the computational load imposed by digital signature authentication.

The variable length of authentication tags in ALTA has implications for application data segmentation when constant-length datagrams are desired (e.g., to maximize data per UDP packet with a given path MTU while avoiding fragmentation).

4. Protocol Details

4.1. ALTA Payload

An ALTA payload comprises the following elements (defined below) concatenated in-order:

- Authentication tag

  - Options octet

  - Optional payload index

  - Sequence of chained MACs

  - Optional digital signature

- Application data

4.1.1. Authentication Tag

The authentication tag is the metadata emitted by an ALTA-compliant sender that is required, in combination with other out-of-band metadata, by an ALTA-compliant receiver to authenticate a stream of packets in a manner tolerant to loss and reordering.

4.1.1.1. Options Octet

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|MACct|S| rsvd  |
+-+-+-+-+-+-+-+-+
```

Figure 1: Options Octet

The first octet of the authentication tag contains the count of MACs included ("MACct") as well as a flag "S" indicating whether the tag also contains a digital signature. It also contains four reserved bits which MUST be set to 0 by senders and ignored by receivers.

4.1.1.2. Payload Index

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|          payload index ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2: Payload Index

If the payload index cannot be deduced from the application data in this payload, it must be specified explicitly in the authentication tag as an unsigned quantity of a fixed length specified by out-of-band metadata.

Whether explicit or deduced, the payload index uniquely identifies a single ALTA stream payload within a rollover window of size "2^N" for some "N" specified in out-of-band metadata. The payload index MUST start at zero and increment by one for each payload transmitted, with rollover to zero on overflow.

4.1.1.3. Chained MACs

```
    0                   1
    0 1 2 3 4 5 6 7 8 9 0 1 ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-
   |    offset     | MAC ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 3: Example MAC with explicit index

In explicit offset mode, each MAC encoded in the payload comprises an offset from the payload's index, expressed as a signed octet in two's complement, followed by a fixed-length MAC. The length and semantics of the MAC are a function of the MAC algorithm, which is specified by out-of-band metadata. The offset space given in the example in Figure 3 is one octet, ranging from -128 to 127, but may be of any number of whole octets, as specified by out-of-band metadata.

In implicit offset mode, the receiver knows the scheme being employed and so can deduce the indices of the chained MACs from the current payload's index. Consequently, the MACs are simply concatenated in ascending order of source index according to the scheme.

4.2. Digital Signature

```
    0 1 2 3 4 5 6 ...
   +-+-+-+-+-+-+-+-+-
   |  signature ...
   +-+-+-+-+-+-+-+-+-
```

Figure 4: Digital Signature

If "S=1" in the options octet, then a digital signature is included in the tag. The length and content of this digital signature are a function of the signature algorithm, which is specified by out-of-band metadata.

4.2.1. Application Data

The application data is opaque, with the exception of the payload index if not specified explicitly in the authentication tag.

4.3. Scheme Construction

In the ALTA context, a scheme describes the directed acyclic graph of payload MACs embedded in other payloads for purposes of chained authentication. The recommended scheme is that described in section 3.2 of [STRAUTH], with "a=3" and "p=5".

FIXME: Describe how to construct this scheme in pseudocode.

5. ALTA Configuration

5.1. Performance Considerations

5.1.1. MAC selection

5.1.2. Digital signature selection

5.2. Out-of-band Metadata

6. Operational Considerations

As ALTA requires an out-of-band channel for provisioning of metadata, including digital signature keys and cryptographic algorithms, versioning of the protocol to support a future ALTA revision may be performed there and acted upon by the application.

7. Security Considerations

7.1. Parsing an ill-formed or inconsistent payload

7.2. Index overflow

7.3. Truncated MACs

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, DOI 10.17487/RFC4082, June 2005, <https://www.rfc-editor.org/info/rfc4082>.

[STRAUTH] Modadugu, N., "Authenticating Streamed Data in the Presence of Random Packet Loss", 2001, <https://crypto.stanford.edu/~pgolle/papers/auth.pdf>.

ISOC Network and Distributed System Security Symposium

[timeskew] "FIXME reference for how bad time sync is", n.d..

Acknowledgments

The author wishes to acknowledge the contributions of his colleague, Jake Holland, whose work with interdomain multicast live video delivery drove the need for a robust solution to the streaming authentication problem, and Eric Rescorla, who introduced the author to the paper describing the loss-tolerant symmetric authentication scheme used as the basis for ALTA.

Author's Address

Kyle Rose
Akamai Technologies, Inc.

Email: krose@krose.org