

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 26, 2018

T. Lemon
Nominum, Inc.
October 23, 2017

Babel Security Model
draft-lemon-homenet-babel-security-latest-00

Abstract

This document describes how to add authenticity to Babel messages so as to prevent malicious tampering or black hole attacks. Peer trust is outside the scope of this document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 26, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Internet-Draft

Babel Security Model

October 2017

Table of Contents

1.	Introduction	2
2.	Message Authenticity	2
3.	Babel Extensions	3
3.1.	Data Structures	3
3.2.	Messages	3
3.2.1.	MAC TLV	4
3.2.2.	Signature TLV	4
3.3.	Message Processing	4
4.	Pairing and Trust	5
5.	IANA Considerations	5
6.	Security Considerations	5
7.	Acknowledgments	5
8.	Normative References	5
	Author's Address	6

[1.](#) Introduction

Babel is a loop-avoiding distance-vector routing protocol suitable for wired and wireless mesh networks. As defined in [\[RFC6126\]](#), Babel is a completely secure protocol. It offers no message authenticity or confidentiality, making it vulnerable to the following attacks:

- o Attacker black holes: An attacker advertises cheap routes to attract direct legitimate traffic to an invalid host.
- o Advertisement tampering: An attacker can steer legitimate traffic away from legitimate hosts by maliciously increasing advertisement costs.

The specification suggests that one of two approaches can mitigate these attacks:

1. Lower-layer security mechanisms, e.g., link-layer authenticated encryption, or
2. Authenticating Babel packets directly via, e.g., a cryptographic MAC computed using a shared key.

In this document, we outline the mechanics necessary for the second strategy. Namely, building message authentication into Babel.

[2.](#) Message Authenticity

Message authenticity requires receivers to verify the contents of each received message. This can be done in one of two ways, depending on the type of destination address used in the message:

Lemon

Expires April 26, 2018

[Page 2]

Internet-Draft

Babel Security Model

October 2017

- o For multicast addresses, the message must be digitally signed. This allows any recipient with that trusts the public key to verify the message. We recommend EdDSA-Ed25519 [[RFC8032](#)] for digital signatures. (EdDSA-Ed25519 signatures have 64-octet signatures instead of 114-octet signatures.)
- o For unicast addresses, the message must contain a cryptographic MAC generated with a secret key shared between the sender and receiver. We recommend HMAC [[RFC2104](#)] or CMAC [[RFC4493](#)] for as the MAC algorithm.

It is assumed that each Babel speaker, i.e., each speaker ID, has an associated public and private key pair. Private keys are used to sign multicast messages. Receivers use (trusted) public keys to verify said messages. Two speakers that trust one another can use these keys to establish a shared secret using mutually authenticated DTLS [[RFC6347](#)]. DTLS is not used to encrypt and authenticate messages afterwards. It is only used to derive a shared secret.

In addition to these keys, routers maintain a monotonically increasing sequence number that is incremented whenever a message is signed or MAC'd. This serves as a unique nonce suitable for replay detection, if desired.

[3.](#) Babel Extensions

The Babel message protocol and data structures must be amended to store peer trust information, i.e., cryptographic keying material.

[3.1.](#) Data Structures

Neighbor tables must be extended to store an optional shared key and corresponding sequence number for each (interface, address) tuple. If the address is unicast, the key **MUST** be present. Otherwise, the address is multicast, and each message is signed using the speaker's private key.

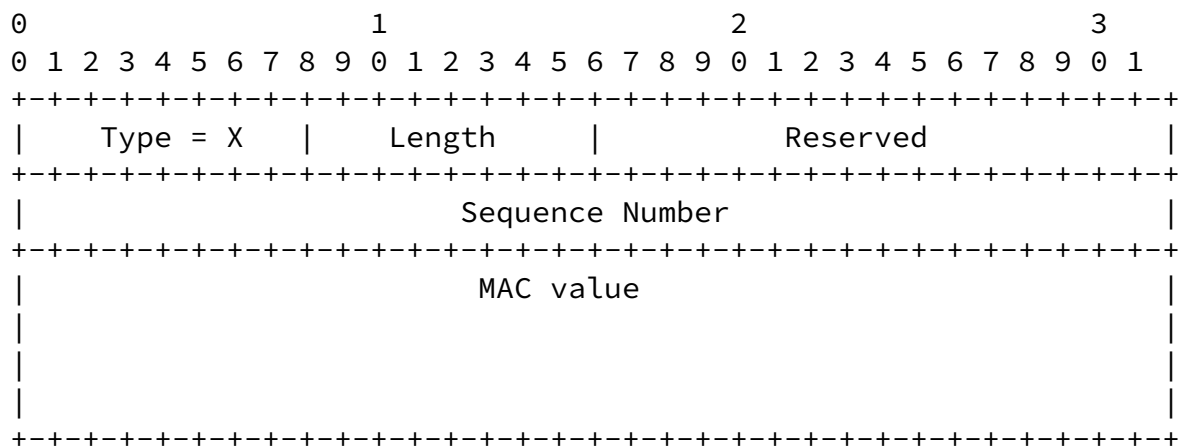
3.2. Messages

Each authenticated Babel message MUST carry one of the two following new TLVs: MAC or Signature. These TLVs MUST be the last TLV in a single Babel message. Their authenticator values are computed over all preceding TLVs, as well as the (T, L, Reserved, Sequence Number) headers in the parent TLV. This authenticates the entire message contents.

The structure of each TLV defined in the following sections.

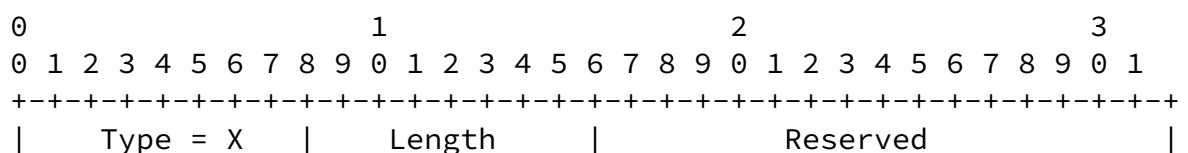
3.2.1. MAC TLV

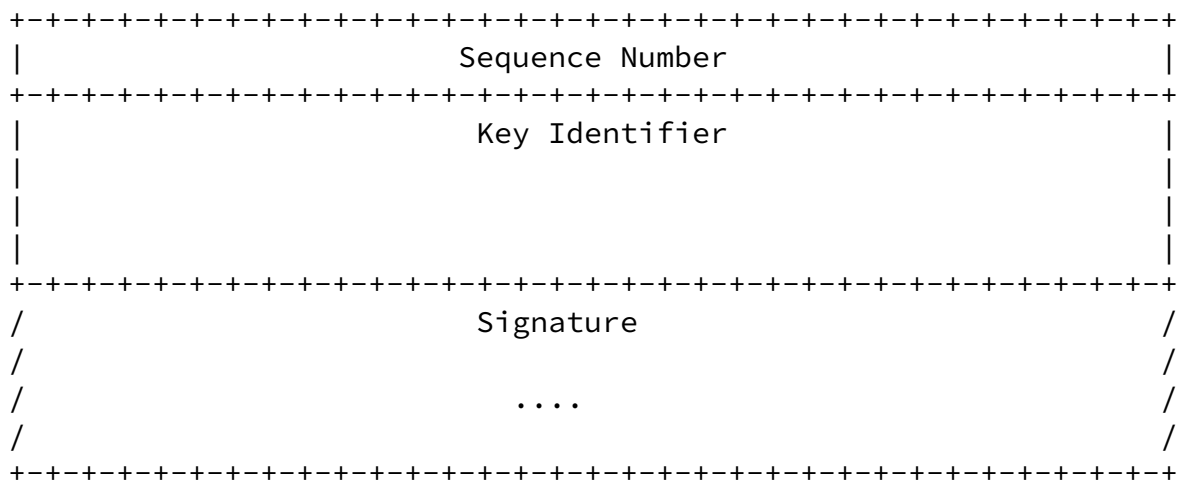
The MAC TLV contains the 4-octet sequence number and 16-octet MAC value, as shown below.



3.2.2. Signature TLV

The Signature TLV contains the 4-octet sequence number, 16-octet key identifier, and 64-octet signature. The key identifier is the (truncated) SHA-256 hash of the sender's public key. The signature is the EdDSA signature, formatted according to [\[RFC8032\]](#).





[3.3.](#) Message Processing

MACs and signatures are computed over all data preceding the actual MAC or signature payload, including the headers of the MAC or Signature TLV. Upon receipt of message with a MAC or Signature TLV,

the receipient must verify its correctness before processing. The verification process for unicast messages works as follows:

1. If there is no MAC TLV, ignore the message.
2. Compute and verify the MAC using the secret key associated with the sender. If the MAC is invalid, ignore the packet.
3. If the MAC is valid, process the message as per normal.

Verification of multicast messages works as follows:

1. If there is no Signature TLV, ignore the message.
2. If there is no public key whose identifier matches the key identifier in the Signature TLV, ignore the message.
3. Verify the signature in the Signature TLV. If invalid, ignore the message.
4. If valid, process the message as per normal.

4. Pairing and Trust

Device pairing and trust establishment is done via HNCP [[RFC7788](#)].

5. IANA Considerations

This document makes no requests to IANA at this time.

6. Security Considerations

This document describes a mechanism to protect Babel protocol messages. Trust in keys used to derive shared secrets and protect is deferred to HNCP [[RFC7788](#)].

7. Acknowledgments

The author would like to thank members of the HomeNet security group for helpful discussions that led to the production of this draft.

8. Normative References

- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.

- [RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", [RFC 4493](#), DOI 10.17487/RFC4493, June 2006, <<https://www.rfc-editor.org/info/rfc4493>>.
- [RFC6126] Chroboczek, J., "The Babel Routing Protocol", [RFC 6126](#), DOI 10.17487/RFC6126, April 2011, <<https://www.rfc-editor.org/info/rfc6126>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", [RFC 6347](#), DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", [RFC 7788](#), DOI 10.17487/RFC7788, April 2016, <<https://www.rfc-editor.org/info/rfc7788>>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", [RFC 8032](#), DOI 10.17487/RFC8032, January 2017, <<https://www.rfc-editor.org/info/rfc8032>>.

Author's Address

Ted Lemon
Nominum, Inc.
800 Bridge Parkway, Suite 100
Redwood City, California 94065
United States of America

Email: Ted.Lemon@nominum.com