Mobile Ad hoc Networking (MANET)   U. Herberg
Internet-Draft   Fujitsu Laboratories of America
Obsoletes: 6622 (if approved)   T. Clausen
Intended status: Standards Track   LIX, Ecole Polytechnique
Expires: September 19, 2013   C. Dearlove
                                                     BAE Systems ATC
                                                      March 18, 2013

               Integrity Check Value and Timestamp TLV Definitions
                    for Mobile Ad Hoc Networks (MANETs)
                     draft-herberg-manet-rfc6622-bis-02

Abstract

   This document extends and replaces RFC 6622. It describes general
   and flexible TLVs for representing cryptographic Integrity Check
   Values (ICVs) (i.e., digital signatures or Message Authentication
   Codes (MACs)) as well as timestamps, using the generalized Mobile Ad
   Hoc Network (MANET) packet/message format defined in RFC 5444. It
   defines two Packet TLVs, two Message TLVs, and two Address Block TLVs
   for affixing ICVs and timestamps to a packet, a message, and an
   address, respectively.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   This document, which extends and replaces [RFC6622], specifies:

   o Two TLVs for carrying Integrity Check Values (ICVs) and timestamps
      in packets, messages, and address blocks as defined by [RFC5444].

   o A generic framework for ICVs, accounting (for Message TLVs) for
      mutable message header fields (<msg-hop-limit> and
      <msg-hop-count>), where these fields are present in messages.

   This document retains the IANA registries, defined in [RFC6622], for
   recording code points for hash-functions, cryptographic functions,
   and ICV calculations. This document requests additional allocations
   from these registries.

   Moreover, in Section 12, this document defines the following:

   o A method for generating ICVs using a combination of a
      cryptographic function and a hash function.

1.1. Differences from RFC6622

   This document obsoletes [RFC6622]. The changes introduced by this
   document are, however, small. In addition to editorial updates, this
   document adds a new type extension for the ICV TLV that is specified
   in Section 12 of this document. The TLV value of TLV with this type
   extension has the same internal structure as A TLV with type
   extension 1, but is calculated also over the source address of the IP
   datagram carrying the packet, message, or address block.

   The rationale for adding this type extension is that some MANET
   protocols, such as [RFC6130] and [OLSRv2], use the IP source address
   of the IP datagram carrying the packet, message or address block,
   e.g., to identify links with neighbor routers. If this address is
   not otherwise contained in the packet, message, or address block
   payload (which is permitted, e.g., in [RFC6130]), the address is not
   protected against tampering.

2. Terminology

   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
   [RFC2119].

   This document uses the terminology and notation defined in [RFC5444].
   In particular, the following TLV fields from [RFC5444] are used in

this specification:

<msg-hop-limit> is the hop limit of a message, as specified in
    Section 5.2 of [RFC5444].

<msg-hop-count> is the hop count of a message, as specified in
    Section 5.2 of [RFC5444].

<length> is the length of a TLV in octets, as specified in Section
    5.4.1 of [RFC5444].

3. Applicability Statement

MANET routing protocols using the format defined in [RFC5444] are
accorded the ability to carry additional information in control
messages and packets, through the inclusion of TLVs. Information so
included MAY be used by a MANET routing protocol, or by an extension
of a MANET routing protocol, according to its specification.

This document specifies how to include an ICV for a packet, a
message, and addresses in address blocks within a message, using such
TLVs. This document also specifies how to treat "mutable" fields,
specifically the <msg-hop-count> and <msg-hop-limit> fields, if
present in the message header when calculating ICVs, such that the
resulting ICV can be correctly verified by any recipient.

This document describes a generic framework for creating ICVs, and
how to include these ICVs in TLVs. In Section 12, an example method
for calculating such ICVs is given, using a cryptographic function
over the hash value of the content.

4. Security Architecture

Basic MANET routing protocol specifications are often "oblivious to
security"; however, they have a clause allowing a control message to
be rejected as "badly formed" or "insecure" prior to the message
being processed or forwarded. MANET routing protocols such as the
Neighborhood Discovery Protocol (NHDP) [RFC6130] and the Optimized
Link State Routing Protocol version 2 [OLSRv2] recognize external
reasons (such as failure to verify an ICV) for rejecting a message
that would be considered "invalid for processing". This architecture
is a result of the observation that with respect to security in
MANETs, "one size rarely fits all" and that MANET routing protocol
deployment domains have varying security requirements ranging from
"unbreakable" to "virtually none". The virtue of this approach is
that MANET routing protocol specifications (and implementations) can
remain "generic", with extensions providing proper security
mechanisms specific to a deployment domain.

The MANET routing protocol "security architecture", in which this
specification situates itself, can therefore be summarized as
follows:

o Security-oblivious MANET routing protocol specifications, with a
   clause allowing an extension to reject a message (prior to
   processing/forwarding) as "badly formed" or "insecure".

o MANET routing protocol security extensions, rejecting messages as
   "badly formed" or "insecure", as appropriate for a given security
   requirement specific to a deployment domain.

o Code points and an exchange format for information, necessary for
   specification of such MANET routing protocol security extensions.

This document addresses the last of the issues listed above by
specifying a common exchange format for cryptographic ICVs, making
reservations from within the Packet TLV, Message TLV, and Address
Block TLV registries of [RFC5444], to be used (and shared) among
MANET routing protocol security extensions.

For the specific decomposition of an ICV into a cryptographic
function over a hash value (specified in Section 12), this document
reports the two IANA registries from [RFC6622] for code points for
hash functions and cryptographic functions adhering to [RFC5444].

With respect to [RFC5444], this document is:

o Intended to be used in the non-normative, but intended, mode of
   use described in Appendix B of [RFC5444].

o A specific example of the Security Considerations section of
   [RFC5444] (the authentication part).

5. Overview and Functioning

This document specifies a syntactical representation of security-
related information for use with [RFC5444] addresses, messages, and
packets, and also reports and updates IANA registrations (from
[RFC6622]) of TLV types and type extension registries for these TLV
types.

Moreover, this document provides guidelines for how MANET routing
protocols and MANET routing protocol extensions using this
specification should treat ICV and Timestamp TLVs, and mutable fields
in messages. This specification does not represent a stand-alone
protocol; MANET routing protocols and MANET routing protocol
extensions, using this specification, MUST provide instructions as to

how to handle packets, messages, and addresses with security
information, associated as specified in this document.

This document reports previously assigned TLV types (from [RFC6622])
from the registries defined for Packet, Message, and Address Block
TLVs in [RFC5444]. When a TLV type is assigned from one of these
registries, a registry for type extensions for that TLV type is
created by IANA. This document reports and updates these type
extension registries, in order to specify internal structure (and
accompanying processing) of the <value> field of a TLV.

For example, and as defined in this document, an ICV TLV with type
extension = 0 specifies that the <value> field has no pre-defined
internal structure but is simply a sequence of octets. An ICV TLV
with type extension = 1 specifies that the <value> field has a pre-
defined internal structure and defines its interpretation. An ICV
TLV with type extension = 2 specifies a modified version of this
definition. (Specifically, with type extension = 1 or type extension
= 2, the <value> field contains the result of combining a
cryptographic function and a hash function, calculated over the
contents of the packet, message or address block, with sub-fields
indicating which hash function and cryptographic function have been
used; this is specified in Section 12. The difference between the
two type extensions is that the ICV TLV with type extension = 2 is
calculated also over the source address of the IP datagram carrying
the packet, message, or address block.)

Other documents can request assignments for other type extensions; if
they do so, they MUST specify their internal structure (if any) and
interpretation.

6. General ICV TLV Structure

The value of the ICV TLV is:

       <value> := <ICV-value>

where

<ICV-value> is a field, of <length> octets, which contains the
   information to be interpreted by the ICV verification process, as
   specified by the type extension.

Note that this does not stipulate how to calculate the <ICV-value>
nor the internal structure thereof, if any; such information MUST be
specified by way of the type extension for the ICV TLV type. See
Section 13. This document specifies three such type extensions --
one for ICVs without pre-defined structures, and two for ICVs

constructed combining a cryptographic function and a hash function.

7. General Timestamp TLV Structure

The value of the Timestamp TLV is:

    <value> := <time-value>

where:

<time-value> is an unsigned integer field, of length <length>, which
   contains the timestamp.

   Note that this does not stipulate how to calculate the
   <time-value> nor the internal structure thereof, if any; such
   information MUST be specified by way of the type extension for the
   TIMESTAMP TLV type. See Section 13.

A timestamp is essentially "freshness information". As such, its
setting and interpretation are to be determined by the MANET routing
protocol, or MANET routing protocol extension, that uses the
timestamp and can, for example, correspond to a UNIX timestamp, GPS
timestamp, or a simple sequence number.

8. Packet TLVs

Two Packet TLVs are defined: one for including the cryptographic ICV
of a packet and one for including the timestamp indicating the time
at which the cryptographic ICV was calculated.

8.1. Packet ICV TLV

A Packet ICV TLV is an example of an ICV TLV as described in
Section 6.

The following considerations apply:

o Because packets as defined in [RFC5444] are never forwarded by
   routers, no special considerations are required regarding mutable
   fields (e.g., <msg-hop-count> and <msg-hop-limit>), if present
   within any messages in the packet, when calculating the ICV.

o Any Packet ICV TLVs already present in the Packet TLV block MUST
   be removed before calculating the ICV, and the Packet TLV block
   size MUST be recalculated accordingly. Removed ICV TLVs MUST be
   restored after having calculated the ICV value.

The rationale for removing any Packet ICV TLV already present prior

   to calculating the ICV is that several ICVs may be added to the same
   packet, e.g., using different ICV functions.

8.2. Packet TIMESTAMP TLV

   A Packet TIMESTAMP TLV is an example of a Timestamp TLV as described
   in Section 7. If a packet contains a TIMESTAMP TLV and an ICV TLV,
   the TIMESTAMP TLV SHOULD be added to the packet before any ICV TLV,
   in order that it be included in the calculation of the ICV.

9. Message TLVs

   Two Message TLVs are defined: one for including the cryptographic ICV
   of a message and one for including the timestamp indicating the time
   at which the cryptographic ICV was calculated.

9.1. Message ICV TLV

   A Message ICV TLV is an example of an ICV TLV as described in
   Section 6. When determining the <ICV-value> for a message, the
   following considerations MUST be applied:

   o The fields <msg-hop-limit> and <msg-hop-count>, if present, MUST
      both be assumed to have the value 0 (zero) when calculating the
      ICV.

   o Any Message ICV TLVs already present in the Message TLV block MUST
      be removed before calculating the ICV, and the message size as
      well as the Message TLV block size MUST be recalculated
      accordingly. Removed ICV TLVs MUST be restored after having
      calculated the ICV value.

   The rationale for removing any Message ICV TLV already present prior
   to calculating the ICV is that several ICVs may be added to the same
   message, e.g., using different ICV functions.

9.2. Message TIMESTAMP TLV

   A Message TIMESTAMP TLV is an example of a Timestamp TLV as described
   in Section 7. If a message contains a TIMESTAMP TLV and an ICV TLV,
   the TIMESTAMP TLV SHOULD be added to the message before the ICV TLV,
   in order that it be included in the calculation of the ICV.

10. Address Block TLVs

   Two Address Block TLVs are defined: one for associating a
   cryptographic ICV to an address and one for including the timestamp
   indicating the time at which the cryptographic ICV was calculated.

10.1. Address Block ICV TLV

An Address Block ICV TLV is an example of an ICV TLV as described in
Section 6. The ICV is calculated over the address, concatenated with
any other values -- for example, any other Address Block TLV <value>
fields -- associated with that address. A MANET routing protocol or
MANET routing protocol extension using Address Block ICV TLVs MUST
specify how to include any such concatenated attribute of the address
in the verification process of the ICV. When determining the
<ICV-value> for an address, the following consideration MUST be
applied:

o If other TLV values are concatenated with the address for
   calculating the ICV, these TLVs MUST NOT be Address Block ICV TLVs
   already associated with the address.

The rationale for not concatenating the address with any ICV TLV
values already associated with the address when calculating the ICV
is that several ICVs may be added to the same address, e.g., using
different ICV functions.

10.2. Address Block TIMESTAMP TLV

An Address Block TIMESTAMP TLV is an example of a Timestamp TLV as
described in Section 7. If both a TIMESTAMP TLV and an ICV TLV are
associated with an address, the TIMESTAMP TLV <value> MUST be covered
when calculating the value of the ICV to be contained in the ICV TLV
value (i.e., concatenated with the associated address and any other
values as described in Section 10.1).

11. ICV: Basic

The basic ICV, represented by way of an ICV TLV with type extension =
0, is a simple bit-field containing the cryptographic ICV. This
assumes that the mechanism stipulating how ICVs are calculated and
verified is established outside of this specification, e.g., by way
of administrative configuration or external out-of-band signaling.
Thus, the <ICV-value>, when using type extension = 0, is

       <ICV-value> := <ICV-data>

where:

<ICV-data> is an unsigned integer field, of length <length>, which
   contains the cryptographic ICV.

12. ICV: Hash Function and Cryptographic Function

   One common way of calculating an ICV is combining a cryptographic
   function and a hash function applied to the content. This
   decomposition is specified in this section, using either type
   extension = 1 or type extension = 2, in the ICV TLVs.

12.1. General ICV TLV Structure

   The following data structure allows representation of a cryptographic
   ICV, including specification of the appropriate hash function and
   cryptographic function used for calculating the ICV:

                    <ICV-value> := <hash-function>
                                   <cryptographic-function>
                                   <key-id-length>
                                   <key-id>
                                   <ICV-data>

   where:

   <hash-function> is an 8-bit unsigned integer field specifying the
      hash function.

   <cryptographic-function> is an 8-bit unsigned integer field
      specifying the cryptographic function.

   <key-id-length> is an 8-bit unsigned integer field specifying the
      length of the <key-id> field in number of octets. The value 0x00
      is reserved for using a pre-installed, shared key.

   <key-id> is a field specifying the key identifier of the key that
      was used to calculate the ICV of the message, which allows unique
      identification of different keys with the same originator. It is
      the responsibility of each key originator to make sure that
      actively used keys that it issues have distinct key identifiers.
      If <key-id-length> equals 0x00, the <key-id> field is not
      contained in the TLV, and a pre-installed, shared key is used.

   <ICV-data> is an unsigned integer field, whose length is
      <length> - 3 - <key-id-length>, and which contains the
      cryptographic ICV.

   The version of this TLV, specified in this section, assumes that,
   unless otherwise specified, calculating the ICV can be decomposed
   into:

        ICV-value = cryptographic-function(hash-function(content))

   In some cases a different combination of cryptographic function and
   hash function may be specified. This is the case for the HMAC
   function, which is specified as defined in Section 13.6, which
   applies the hash function twice.

   The hash function and the cryptographic function correspond to the
   entries in two IANA registries, which are reported by this
   specification and are described in Section 13.

12.1.1. Rationale

   The rationale for separating the hash function and the cryptographic
   function into two octets instead of having all combinations in a
   single octet -- possibly as a TLV type extension -- is that adding
   further hash functions or cryptographic functions in the future may
   lead to a non-contiguous number space.

   The rationale for not including a field that lists parameters of the
   cryptographic ICV in the TLV is that, before being able to validate a
   cryptographic ICV, routers have to exchange or acquire keys (e.g.,
   public keys). Any additional parameters can be provided together
   with the keys in that bootstrap process. It is therefore not
   necessary, and would even entail an extra overhead, to transmit the
   parameters within every message. One implicitly available parameter
   is the length of the ICV, which is <length> - 3 - <key-id-length>,
   and which depends on the choice of the cryptographic function.

12.2. Considerations for Calculating the ICV

   The considerations listed in the following subsections MUST be
   applied when calculating the ICV for Packet, Message, and Address
   Block ICV TLVs, respectively.

12.2.1. Packet ICV TLV

   When determining the <ICV-value> for a packet, with type extension =
   1, the ICV is calculated over the fields <hash-function>,
   <cryptographic-function>, <key-id-length>, and -- if present --
   <key-id> (in that order), concatenated with the entire packet,
   including the packet header, all Packet TLVs (other than Packet ICV
   TLVs), and all included Messages and their message headers, in
   accordance with Section 8.1. When determining the <ICV-value> for a
   packet, with type extension = 2, the same procedure is used, except
   that the source address of the IP datagram carrying the packet is
   also concatenated, in the first position, with the data used.

12.2.2. **Message ICV TLV**

   When determining the <ICV-value> for a message, with type extension =
   1, the ICV is calculated over the fields <hash-function>,
   <cryptographic-function>, <key-id-length>, and -- if present --
   <key-id> (in that order), concatenated with the entire message. The
   considerations in Section 9.1 MUST be applied. When determining the
   <ICV-value> for a message, with type extension = 2, the same
   procedure is used, except that the source address of the IP datagram
   carrying the message is also concatenated, in the first position,
   with the data used.

12.2.3. **Address Block ICV TLV**

   When determining the <ICV-value> for an address, block, with type
   extension = 2, the ICV is calculated over the fields <hash-function>,
   <cryptographic-function>, <key-id-length>, and -- if present --
   <key-id> (in that order), concatenated with the address, and
   concatenated with any other values -- for example, any other address
   block TLV <value> that is associated with that address. A MANET
   routing protocol or MANET routing protocol extension using Address
   Block ICV TLVs MUST specify how to include any such concatenated
   attribute of the address in the verification process of the ICV. The
   considerations in Section 10.1 MUST be applied. When determining the
   <ICV-value> for an address block, with type extension = 2, the same
   procedure is used, except that the source address of the IP datagram
   carrying the address block is also concatenated, in the first
   position, with the data used.

12.3. **Example of a Message Including an ICV**

   The sample message depicted in Figure 1 is derived from Appendix D of
   [RFC5444]. The message contains an ICV Message TLV, with the value
   representing an ICV that is 16 octets long of the whole message, and
   a key identifier that is 4 octets long. The type extension of the
   Message TLV is 1, for the specific decomposition of an ICV using a
   cryptographic function and a hash function, as specified in
   Section 12.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | PV=0 |  PF=8  |    Packet Sequence Number    | Message Type  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | MF=15 | MAL=3 |    Message Length = 44       | Msg Orig Addr |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       Message Originator Address (cont)      |   Hop Limit   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Count   |    Message Sequence Number   | Msg TLV Block |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Length = 27   |    ICV      |  MTLVF = 144  |  MTLVExt = 1  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Value Len = 23 |  Hash Func  |  Crypto Func  |Key ID length=4|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Key Identifier                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       ICV Value                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    ICV Value (cont)                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    ICV Value (cont)                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    ICV Value (cont)                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                   Figure 1: Example Message with ICV

13. IANA Considerations

This specification reports the following, originally specified in
[RFC6622]:

o Two Packet TLV types, which have been allocated from the 0-223
   range of the "Packet TLV Types" repository of [RFC5444], as
   specified in Table 1.

o Two Message TLV types, which have been allocated from the 0-127
   range of the "Message TLV Types" repository of [RFC5444], as
   specified in Table 2.

o Two Address Block TLV types, which have been allocated from the
   0-127 range of the "Address Block TLV Types" repository of
   [RFC5444], as specified in Table 3.

This specification updates the following, created in [RFC6622]:

   o A type extension registry for each of these TLV types with values
      as listed in Tables 1, 2, and 3.

   The following terms are used as defined in [BCP26]: "Namespace",
   "Registration", and "Designated Expert".

   The following policy is used as defined in [BCP26]: "Expert Review".

13.1. Expert Review: Evaluation Guidelines

   For TLV type extensions registries where an Expert Review is
   required, the Designated Expert SHOULD take the same general
   recommendations into consideration as those specified by [RFC5444].

   For the Timestamp TLV, the same type extensions for all Packet,
   Message, and Address Block TLVs SHOULD be numbered identically.

13.2. Packet TLV Type Registrations

   IANA has, in accordance with [RFC6622], made allocations from the
   "Packet TLV Types" namespace of [RFC5444] for the Packet TLVs
   specified in Table 1. IANA are requested to modify this allocation
   (defining type extension = 2) as indicated.

Name	Type	Type Extension	Description
ICV	5	0	ICV of a packet
		1-2	ICV, using a cryptographic function and a hash function, as specified in [Section 12](#) of this document
		3-251	Unassigned; Expert Review
		252-255	Experimental Use
TIMESTAMP	6	0	Unsigned timestamp of arbitrary length, given by the TLV Length field. The MANET routing protocol has to define how to interpret this timestamp
		1	Unsigned 32-bit timestamp, as specified in [IEEE 1003.1-2008 (POSIX)]
		2	NTP timestamp format, as defined in [RFC5905]
		3	Signed timestamp of arbitrary length with no constraints such as monotonicity. In particular, it may represent any random value
		4-251	Unassigned; Expert Review
		252-255	Experimental Use

Table 1: Packet TLV Types

More than one ICV Packet TLV with the same type extension MAY be included in a packet if these represent different ICV calculations (e.g., with type extension 1 or 2 and different cryptographic function and/or hash function). ICV Packet TLVs that carry what is declared to be the same information MUST NOT be included in the same packet.

13.3. Message TLV Type Registrations

IANA has, in accordance with [RFC6622], made allocations from the "Message TLV Types" namespace of [RFC5444] for the Message TLVs specified in Table 2. IANA are requested to modify this allocation (defining type extension = 2) as indicated.

Name	Type	Type Extension	Description
ICV	5	0	ICV of a message
		1-2	ICV, using a cryptographic function and a hash function, as specified in Section 12 of this document
		3-251	Unassigned; Expert Review
		252-255	Experimental Use
TIMESTAMP	6	0	Unsigned timestamp of arbitrary length, given by the TLV Length field.
		1	Unsigned 32-bit timestamp, as specified in [IEEE 1003.1-2008 (POSIX)]
		2	NTP timestamp format, as defined in [RFC5905]
		3	Signed timestamp of arbitrary length with no constraints such as monotonicity. In particular, it may represent any random value
		4-251	Unassigned; Expert Review
		252-255	Experimental Use

Table 2: Message TLV Types

More than one ICV Message TLV with the same type extension MAY be included in a message if these represent different ICV calculations (e.g., with type extension 1 or 2 and different cryptographic function and/or hash function). ICV Message TLVs that carry what is declared to be the same information MUST NOT be included in the same message.

13.4. Address Block TLV Type Registrations

IANA has, in accordance with [RFC6622], made allocations from the "Address Block TLV Types" namespace of [RFC5444] for the Packet TLVs specified in Table 3. IANA are requested to modify this allocation (defining type extension = 2) as indicated.

```
+-----------+------+-----------+----------------------------------+
|    Name   | Type |    Type   |            Description            |
|           |      | Extension |                                  |
+-----------+------+-----------+----------------------------------+
|    ICV    |   5  |     0     |     ICV of an object (e.g., an    |
|           |      |           |              address)            |
|           |      |    1-2    |     ICV, using a cryptographic    |
|           |      |           |   function and a hash function, as |
|           |      |           |   specified in Section 12 of this |
|           |      |           |              document            |
|           |      |   3-251   |     Unassigned; Expert Review    |
|           |      |  252-255  |        Experimental Use          |
| TIMESTAMP |   6  |     0     |  Unsigned timestamp of arbitrary |
|           |      |           |   length, given by the TLV Length |
|           |      |           |              field               |
|           |      |     1     |    Unsigned 32-bit timestamp, as  |
|           |      |           |   specified in [IEEE 1003.1-2008  |
|           |      |           |             (POSIX)]             |
|           |      |     2     |  NTP timestamp format, as defined |
|           |      |           |         in [RFC5905]             |
|           |      |     3     |   Signed timestamp of arbitrary  |
|           |      |           | length with no constraints such as |
|           |      |           |   monotonicity.  In particular, it |
|           |      |           |   may represent any random value  |
|           |      |   4-251   |     Unassigned; Expert Review    |
|           |      |  252-255  |        Experimental Use          |
+-----------+------+-----------+----------------------------------+
```

Table 3: Address Block TLV Types

More than one ICV Address Block TLV with the same type extension MAY
be associated with an Address Block if these represent different ICV
calculations (e.g., with type extension 1 or 2 and different
cryptographic function and/or hash function). ICV Address Block TLVs
that carry what is declared to be the same information MUST NOT be
associated with the same Address Block.

13.5. Hash Functions

IANA has, in accordance with [RFC6622], created a new registry for
hash functions that can be used when creating an ICV, as specified in
Section 12 of this document. The initial assignments and allocation
policies are specified in Table 4. This registry is unchanged by
this specification.

```
+-------------+----------+---------------------------------------+
|    Hash     | Algorithm |              Description               |
|  Function   |          |                                       |
|   Value     |          |                                       |
+-------------+----------+---------------------------------------+
|      0      |   none   | The "identity function": The hash value |
|             |          |    of an object is the object itself   |
|      1      |   SHA1   |           [NIST-FIPS-180-2]            |
|      2      |  SHA224  |        [NIST-FIPS-180-2-change]        |
|      3      |  SHA256  |           [NIST-FIPS-180-2]            |
|      4      |  SHA384  |           [NIST-FIPS-180-2]            |
|      5      |  SHA512  |           [NIST-FIPS-180-2]            |
|    6-251    |          |        Unassigned; Expert Review       |
|   252-255   |          |           Experimental Use             |
+-------------+----------+---------------------------------------+
```

                    Table 4: Hash Function Registry

13.6. Cryptographic Functions

   IANA has, in accordance with [RFC6622], created a new registry for
   the cryptographic functions, as specified in Section 12 of this
   document. Initial assignments and allocation policies are specified
   in Table 5. This registry is unchanged by this specification.

```
+----------------+----------+---------------------------------------+
| Cryptographic  | Algorithm |              Description               |
| Function Value |          |                                       |
+----------------+----------+---------------------------------------+
|       0        |   none   | The "identity function": The value     |
|                |          |   of an encrypted hash is the hash     |
|                |          |               itself                   |
|       1        |   RSA    |             [RFC3447]                  |
|       2        |   DSA    |         [NIST-FIPS-186-3]              |
|       3        |   HMAC   |             [RFC2104]                  |
|       4        |   3DES   |          [NIST-SP-800-67]              |
|       5        |   AES    |           [NIST-FIPS-197]              |
|       6        |  ECDSA   |          [ANSI-X9-62-2005]             |
|     7-251      |          |      Unassigned; Expert Review         |
|    252-255     |          |          Experimental Use              |
+----------------+----------+---------------------------------------+
```

               Table 5: Cryptographic Function Registry

14. Security Considerations

This document does not specify a protocol. It provides a syntactical
component for cryptographic ICVs of messages and packets, as defined
in [RFC5444]. It can be used to address security issues of a MANET
routing protocol or MANET routing protocol extension. As such, it
has the same security considerations as [RFC5444].

In addition, a MANET routing protocol or MANET routing protocol
extension that uses this specification MUST specify how to use the
framework, and the TLVs presented in this document. In addition, the
protection that the MANET routing protocol or MANET routing protocol
extensions attain by using this framework MUST be described.

As an example, a MANET routing protocol that uses this component to
reject "badly formed" or "insecure" messages if a control message
does not contain a valid ICV SHOULD indicate the security assumption
that if the ICV is valid, the message is considered valid. It also
SHOULD indicate the security issues that are counteracted by this
measure (e.g., link or identity spoofing) as well as the issues that
are not counteracted (e.g., compromised keys).

15. Acknowledgements

The authors would like to thank Bo Berry (Cisco), Alan Cullen (BAE
Systems), Justin Dean (NRL), Paul Lambert (Marvell), Jerome Milan
(Ecole Polytechnique), and Henning Rogge (FGAN) for their
constructive comments on [RFC6622].

The authors also appreciate the detailed reviews of [RFC6622] from
the Area Directors, in particular Stewart Bryant (Cisco), Stephen
Farrell (Trinity College Dublin), and Robert Sparks (Tekelec), as
well as Donald Eastlake (Huawei) from the Security Directorate.

16. References

16.1. Normative References

[BCP26]   Narten, T. and H. Alvestrand, "Guidelines
                           for Writing an IANA Considerations
                           Section in RFCs", BCP 26, RFC 5226,
                           May 2008.

[RFC2119]   Bradner, S., "Key words for use in RFCs
                           to Indicate Requirement Levels", BCP 14,
                           RFC 2119, March 1997.

[RFC5444]   Clausen, T., Dearlove, C., Dean, J., and

                               C. Adjih, "Generalized Mobile Ad Hoc
                               Network (MANET) Packet/Message Format",
                               RFC 5444, February 2009.

   [RFC5905]   Mills, D., Martin, J., Ed., Burbank, J.,
                               and W. Kasch, "Network Time Protocol
                               Version 4: Protocol and Algorithms
                               Specification", RFC 5905, June 2010.

   [RFC3447]   Jonsson, J. and B. Kaliski, "Public-Key
                               Cryptography Standards (PKCS) #1: RSA
                               Cryptography Specifications Version 2.1",
                               RFC 3447, February 2003.

   [RFC2104]   Krawczyk, H., Bellare, M., and R.
                               Canetti, "HMAC: Keyed-Hashing for Message
                               Authentication", RFC 2104, February 1997.

   [NIST-FIPS-197]   National Institute of Standards and
                               Technology, "Specification for the
                               Advanced Encryption Standard (AES)",
                               FIPS 197, November 2001.

   [NIST-FIPS-186-3]   National Institute of Standards and
                               Technology, "Digital Signature Standard
                               (DSS)", FIPS 186-3, June 2009.

   [ANSI-X9-62-2005]   American National Standards Institute,
                               "Public Key Cryptography for the
                               Financial Services Industry: The Elliptic
                               Curve Digital Signature Algorithm
                               (ECDSA)", ANSI X9.62-2005, November 2005.

   [NIST-SP-800-67]   National Institute of Standards and
                               Technology, "Recommendation for the
                               Triple Data Encryption Algorithm
                               (TDEA) Block Cipher", Special
                               Publication 800-67, May 2004.

   [NIST-FIPS-180-2]   National Institute of Standards and
                               Technology, "Specifications for the
                               Secure Hash Standard", FIPS 180-2,
                               August 2002.

   [NIST-FIPS-180-2-change]   National Institute of Standards and
                               Technology, "Federal Information
                               Processing Standards Publication 180-2 (+
                               Change Notice to include SHA-224)",

FIPS 180-2, August 2002.

[IEEE 1003.1-2008 (POSIX)] IEEE Computer Society, "1003.1-2008
                            Standard for Information Technology -
                            Portable Operating System Interface
                            (POSIX) Base Specifications, Issue 7",
                            December 2008.

16.2. Informative References

[RFC6130]   Clausen, T., Dearlove, C., and J. Dean,
                            "Mobile Ad Hoc Network (MANET)
                            Neighborhood Discovery Protocol (NHDP)",
                            RFC 6130, April 2011.

[RFC6622]   Herberg, U. and T. Clausen, "Integrity
                            Check Value and Timestamp TLV Definitions
                            for Mobile Ad Hoc Networks (MANETs)",
                            RFC 6622, May 2012.

[OLSRv2]   Clausen, T., Dearlove, C., Jacquet, P.,
                            and U. Herberg, "The Optimized Link State
                            Routing Protocol version 2", Work
                            in Progress, March 2012.

Authors' Addresses

   Ulrich Herberg
   Fujitsu Laboratories of America
   1240 E. Arques Ave.
   Sunnyvale, CA 94085
   USA

   EMail: ulrich@herberg.name
   URI: http://www.herberg.name/

   Thomas Heide Clausen
   LIX, Ecole Polytechnique
   91128 Palaiseau Cedex
   France

   Phone: +33 6 6058 9349
   EMail: T.Clausen@computer.org
   URI: http://www.thomasclausen.org/

Christopher Dearlove
BAE Systems ATC

Phone: +44 1245 242194
EMail: chris.dearlove@baesystems.com
URI: http://www.baesystems.com/