RFC7182: Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)

Thomas Heide Clausen, Ulrich Herberg, Christopher Chris Dearlove

HAL Id: hal-03172446

https://polytechnique.hal.science/hal-03172446

                Integrity Check Value and Timestamp TLV Definitions
                       for Mobile Ad Hoc Networks (MANETs)

Abstract

   This document revises, extends, and replaces RFC 6622. It describes
   general and flexible TLVs for representing cryptographic Integrity
   Check Values (ICVs) and 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 one or
   more addresses, respectively.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   This document specifies a syntactical representation of security-
   related information for use with [RFC5444] addresses, messages, and
   packets. It also specifies IANA registrations of TLV types and type
   extension registries for these TLV types. This specification does
   not represent a stand-alone protocol, but it is intended for use by
   MANET routing protocols or security extensions thereof.

   Specifically, this document, which revises, extends, and replaces
   [RFC6622], specifies:

   o Two kinds of TLV: one for carrying Integrity Check Values (ICVs)
      and one for timestamps in packets, messages, and Address Blocks as
      defined by [RFC5444].

   o A generic framework for use of these TLVs, accounting for specific
      features of Packet, Message, and Address Block TLVs.

   o IANA registrations for TLVs, and registries for TLV type
      extensions, replacing those from [RFC6622].

   This document specifies IANA registries for recording code points for
   ICV TLVs and TIMESTAMP TLVs, as well as timestamps, hash functions,
   and cryptographic functions.

   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 and for including such
      ICVs in the value field of a TLV.

1.1. Differences from RFC 6622

This document obsoletes [RFC6622], replacing that document as the
specification of two TLV types, TIMESTAMP and ICV, for packets,
messages and Address Blocks. For the ICV type, this document
specifies a new type extension, 2 (see Section 12), in addition to
including the original type extensions (0 and 1) from [RFC6622].

The TLV value of an ICV TLV with type extension = 2 has the same
internal structure as an ICV 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], 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]), then the address is not protected against tampering.

This document also incorporates a number of editorial improvements
over [RFC6622]. In particular, it makes it clear that an ICV TLV may
be used to carry a truncated ICV and that a single or multivalue
TIMESTAMP or ICV Address Block TLV may cover more than one address.
Moreover, to be consistent with the terminology in [RFC5444], the
name of the TLVs specified in this document have changed from "Packet
ICV TLV" to "ICV Packet TLV" and from "Packet TIMESTAMP TLV" to
"TIMESTAMP Packet TLV" (and similar for Message and Address Block
TLVs).

A normative requirement in Section 9.2 has changed from SHOULD to
MUST in the following sentence:

   If a message contains one or more TIMESTAMP TLVs and one or more
   ICV TLVs, then the TIMESTAMP TLVs (as well as any other Message
   TLVs) MUST be added to the message before the ICV TLVs....

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 and notation 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 the value field in a TLV in octets, as
   specified in Section 5.4.1 of [RFC5444].

single-length is the length of a single value in the value field in
   a TLV in octets, as specified in Section 5.4.1 of [RFC5444]. (It
   is equal to <length> except in a multivalue Address Block TLV.)

In addition to using the regular expressions defined in Section 2.1.1
of [RFC5444], this document defines the following:

+ - One or more occurrences of the preceding element or group.

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 an Address Block within a message, using
   such TLVs. This document also specifies how to treat an empty Packet
   TLV Block, and "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
   and a hash function, for which two TLV type extensions are allocated.

   This document does not specify a protocol. Protocol specifications
   that make use of the framework, specified in this document, will
   reference this document in a normative way, and they may require the
   implementation of some or all of the algorithms described in this
   document. As this document does not specify a protocol itself, key

management and key exchange mechanisms are out of scope and may be
specified in the protocol or protocol extension using this
specification.

4. Security Architecture

   MANET routing protocol specifications may have a clause allowing a
   control message to be rejected as "badly formed" or "insecure" prior
   to the message being processed or forwarded. In particular, MANET
   routing protocols such as the Neighborhood Discovery Protocol (NHDP)
   [RFC6130] and the Optimized Link State Routing Protocol version 2
   [RFC7181] 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 MANET routing protocol specifications, each 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, each 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 points above, by specifying a
   common exchange format for cryptographic ICVs and timestamps, making
   reservations from within the Packet TLV, Message TLV, and Address
   Block TLV registries of [RFC5444], to be used by (and shared among)
   MANET routing protocol security extensions.

   For the specific decomposition of an ICV using a cryptographic
   function and a hash function (specified in Section 12), this document
   specifies two IANA registries (see Section 13) for code points for
   hash functions and cryptographic functions.

   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 specifies IANA registrations (see Section 13) 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 specifies TLV type assignments (see Section 13) 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 specifies these type extension registries, in order to
   specify internal structure (and accompanying processing) of the
   <value> field of a TLV.

   For example, and as specified in this document, an ICV TLV with type
   extension = 0 specifies that the <value> field has no predefined
   internal structure, but is simply a sequence of octets. An ICV TLV
   with type extension = 1 specifies that the <value> field has a
   predefined internal structure and defines its interpretation. An ICV
   TLV with type extension = 2 (added in this document) is the same as
   an ICV TLV with type extension = 1, except that the integrity
   protection also covers the source address of the IP datagram carrying
   the packet, message, or Address Block.

   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. The <value> field contains sub-
   fields indicating which hash function and cryptographic function have
   been used, as specified in Section 12.

   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 <length> octets (except in a
      multivalue Address Block TLV, where each <ICV-value> is of length
      single-length octets) that contains the information to be
      interpreted by the ICV verification process, as specified by the
      type extension.

   Note that this does not specify how to calculate the <ICV-value> nor
   the internal structure thereof, if any; such information MUST be
   specified by the type extension for the ICV TLV type; see Section 13.
   This document specifies three such type extensions: one for ICVs
   without predefined 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 a field, of length <length> octets (except in a
      multivalue Address Block TLV, where each <time-value> is of length
      single-length octets) that contains the timestamp.

   Note that this does not specify how to calculate the <time-value> nor
   the internal structure thereof, if any; such information MUST be
   specified by 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 POSIX timestamp, GPS
   timestamp, or a simple sequence number. Note that ensuring time
   synchronization in a MANET may be difficult because of the

decentralized architecture as well as highly dynamic topology due to
mobility or other factors. It is out of scope for this document to
specify a time synchronization mechanism.

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. ICV Packet TLV

   An ICV Packet TLV is an example of an ICV TLV as described in
   Section 6. When determining the <ICV-value> for a packet, and adding
   an ICV Packet TLV to a packet, the following considerations MUST be
   applied:

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

   o Any ICV Packet 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.

   o If the Packet TLV Block now contains no Packet TLVs, the Packet
      TLV Block MUST be removed, and the phastlv bit in the <pkt-flags>
      field in the Packet Header MUST be cleared ('0').

   o Any removed ICV Packet TLVs MUST be restored after having
      calculated the ICV, and the Packet TLV Block size MUST be
      recalculated accordingly.

   o When any removed ICV Packet TLVs, and the newly calculated ICV
      Packet TLV, are added to the packet, if there is no Packet TLV
      Block, then one MUST be added, including setting ('1') the phastlv
      bit in the <pkt-flags> field in the Packet Header.

   The rationale for removing any ICV Packet TLVs already present prior
   to calculating the ICV is that several ICV TLVs may be added to the
   same packet, e.g., using different ICV cryptographic and/or hash
   functions. The rationale for removing an empty Packet TLV Block is
   because the receiver of the packet cannot tell the difference between
   what was an absent Packet TLV Block, and what was an empty Packet TLV
   Block when removing and verifying the ICV Packet TLV if no other
   Packet TLVs are present.

8.2. TIMESTAMP Packet TLV

   A TIMESTAMP Packet TLV is an example of a Timestamp TLV as described
   in Section 7. If a packet contains one or more TIMESTAMP TLVs and
   one or more ICV TLVs, then the TIMESTAMP TLVs (as well as any other
   Packet TLVs) MUST be added to the packet before the ICV TLVs, in
   order to include the timestamps and other TLVs in the calculation of
   the ICVs.

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. ICV Message TLV

   An ICV Message 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 in the
      Message Header, MUST both be assumed to have the value 0 (zero)
      when calculating the ICV.

   o Any ICV Message 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. Also, all relevant TLVs other than ICV TLVs MUST be
      added prior to ICV value calculation.

   o Any removed ICV Message TLVs MUST be restored after having
      calculated the ICV, and the message size as well as the Message
      TLV Block size MUST be recalculated accordingly.

   The rationale for removing any ICV Message TLVs already present prior
   to calculating the ICV is that several ICV TLVs may be added to the
   same message, e.g., using different ICV cryptographic and/or hash
   functions.

9.2. TIMESTAMP Message TLV

   A TIMESTAMP Message TLV is an example of a Timestamp TLV as described
   in Section 7. If a message contains one or more TIMESTAMP TLVs and
   one or more ICV TLVs, then the TIMESTAMP TLVs (as well as any other
   Message TLVs) MUST be added to the message before the ICV TLVs, in
   order to include the timestamps and other Message TLVs in the
   calculation of the ICV.

10. Address Block TLVs

   Two Address Block TLVs are defined: one for associating a
   cryptographic ICV to one or more addresses and their associated
   information and one for including the timestamp indicating the time
   at which the cryptographic ICV was calculated.

10.1. ICV Address Block TLV

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

   o If other TLV values are concatenated with the addresses for
      calculating the ICV, the corresponding TLVs MUST NOT be ICV
      Address Block TLVs already associated with any of the addresses.

   The rationale for not concatenating the addresses with any ICV TLV
   values already associated with the addresses when calculating the ICV
   is that several ICVs may be added to the same address or addresses,
   e.g., using different ICV cryptographic and/or hash functions, and
   the order of addition is not known to the recipient.

10.2. TIMESTAMP Address Block TLV

   A TIMESTAMP Address Block TLV is an example of a Timestamp TLV as
   described in Section 7. If one or more TIMESTAMP TLVs and one or
   more ICV TLVs are associated with an address, the relevant TIMESTAMP
   TLV <time-value>(s) MUST be included before calculating the value of
   the ICV to be contained in the ICV TLV value (i.e., concatenated with
   the associated addresses 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, has as TLV value a simple bit-field without specified
   structure (i.e, without explicitly included hash function, crypto
   function, key ID or other parameters). Moreover, it is not specified
   how to calculate the <ICV-value>. It is assumed that the mechanism
   specifying how ICVs are calculated and verified, as well as which
   parameters (if any) need to be exchanged prior to using the TLV with

type extension = 0, is established outside of this specification,
e.g., by administrative configuration or external out-of-band
signaling.

The <ICV-value>, when using type extension = 0, is:

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

   where:

      <ICV-data> is a field, of length <length> octets (or single-length
      octets in a multivalue Address Block TLV) that 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 a one-octet unsigned integer field specifying
      the hash function.

      <cryptographic-function> is a one-octet unsigned integer field
      specifying the cryptographic function.

      <key-id-length> is a one-octet unsigned integer field specifying
      the length of the <key-id> field as a number of octets. The value
      zero (0x00) is reserved for using a single 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 zero (0x00), the <key-id> field is not
contained in the TLV, and a single pre-installed, shared key is
used.

<ICV-data> is a field with length <length> - 3 - <key-id-length>
octets (except in a multivalue Address Block TLV, in which it is
single-length - 3 - <key-id-length> octets) and that 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 Hashed
Message Authentication Code (HMAC) function, which is specified as
defined in Section 13.12, using the hash function twice. Using
cryptographic-function "none" is provided for symmetry and possible
future use, but it SHOULD NOT be used with any currently specified
hash function.

The difference between the two type extensions is that in addition to
the information covered by the ICV using type extension = 1 (which is
detailed in the following sections), the ICV using type extension = 2
also MUST cover the source address of the IP datagram carrying the
corresponding packet, message, or Address Block.

The <ICV-data> field MAY be truncated after being calculated, this is
indicated by its length, calculated as described above. The
truncation MUST be as specified for the relevant cryptographic
function (and, if appropriate, hash function).

o When using truncation, the guidelines for minimal ICV length set
   out in [NIST-SP-800-107] MUST be followed. In particular the
   <ICV-data> field when using HMAC MUST NOT be truncated below 4
   octets.

o The truncated ICV length MUST be so large that the probability of
   success of a dictionary attack is acceptably small. Such a
   success will arise if the ICV of a spoofed packet or message is
   verified. The probability of success is a function of (a) how

   many routers can be attacked, (b) how fast a router can receive
   packets or messages and attempt to verify their ICV, (c) the
   truncated ICV length, and (d) the lifetime of the network. If the
   truncated ICV length in bits is L, then 2^L packets or messages
   are required to attack with certainty of success. With a
   verification rate of R packets/messages per second, applied to N
   routers over an available time of T, the probability of success is
   given by NRT/2^L. If this is not to exceed a probability of P,
   then L > log2(NRT/P). For example, if N is 32, R is 1000, T is
   86400 (I day) and P is 10^-6, then L must be at least 52 bits.

Some of the cryptographic and hash functions listed in Section 13
require the length of the content to be digitally signed to be a
multiple of a certain number of octets. As a consequence, they
specify padding mechanisms, e.g., AES-CMAC [RFC4493] specifies a
padding mechanism for message lengths that are not equal to a
multiple of 16 octets. Implementations of the framework in this
document MUST support appropriate padding mechanisms, as specified in
the cryptographic or hash function specifications.

The hash function and the cryptographic function correspond to the
entries in two IANA registries, which 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 as well as a smaller overall
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. Any
additional parameters can be provided together with the keys in that
bootstrap process. Therefore, it is not necessary, and would even
entail an extra overhead, to transmit the parameters within every
message.

The rationale for the addition of type extension = 2 is that the
source address is used in some cases, such as when processing HELLO
messages in [RFC6130]. This is applicable only to packets (which
only ever travel one hop) and messages (and their Address Blocks)
that only travel one hop. It is not applicable to messages that may
be forwarded more than one hop, such as Topology Control (TC)
messages in [RFC7181].

12.1.2. Parameters

   As described in Section 12.1.1, parameters are selected
   administratively on each router before using this framework in a
   MANET, in addition to exchanging the keys between MANET routers.
   This was a design decision in [RFC6622] and is kept in this
   specification for reasons of backwards compatibility.

   The following parameters are RECOMMENDED and SHOULD be those chosen
   administratively, unless there are good reasons otherwise:

   o For crypto function RSA:

      * Signature scheme: RSASSA-PSS with the default parameters:
         rSASSA-PSS-Default-Identifier (as defined in [RFC3447])

      * Common exponent: 65537

   o For crypto function ECDSA:

      * Curve name: exchanged as part of key distribution

      * Hash function: The hash function MUST be pinned to the curve,
         i.e., use SHA-256 for the p-256 curve, SHA-384 for p-384, etc.

   o For crypto function AES:

      * Authentication algorithm: Cipher-Based Message Authentication
         Code (CMAC) (as defined in [RFC4493])

      * Hash function: None

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 TLVs, respectively.

12.2.1. ICV Packet TLV

   When determining the <ICV-data> for a packet, with type
   extension = 1:

   o The ICV is calculated over the fields <hash-function>,
      <cryptographic-function>, <key-id-length>, and -- if present --
      <key-id> (in that order), followed by the entire packet, including

the Packet Header, including all Packet TLVs (other than ICV
Packet TLVs), and all included messages. The considerations of
Section 8.1 MUST be applied.

When determining the <ICV-data> for a packet, with type
extension = 2:

o The same procedure as for type extension = 1 is used, except that
   the data used consists of a representation of the source address
   of the IP datagram carrying the packet, followed by the remaining
   data (as for type extension = 1). The representation of the
   source address consists of a single octet containing the address
   length, in octets, followed by that many octets containing the
   address in network byte order.

12.2.2. ICV Message TLV

When determining the <ICV-data> for a message, with type
extension = 1:

o The ICV is calculated over the fields <hash-function>,
   <cryptographic-function>, <key-id-length>, and -- if present --
   <key-id> (in that order), followed by the entire message. The
   considerations in Section 9.1 MUST be applied.

When determining the <ICV-data> for a message, with type
extension = 2:

o The same procedure as for type extension = 1 is used, except that
   the data used consists of a representation of the source address
   of the IP datagram carrying the message, followed by the remaining
   data (as for type extension = 1). The representation of the
   source address consists of a single octet containing the address
   length, in octets, followed by that many octets containing the
   address in network byte order.

12.2.3. ICV Address Block TLV

When determining the <ICV-data> for one or more addresses, with type
extension = 1:

o The ICV is calculated over the fields <hash-function>,
   <cryptographic-function>, <key-id-length>, and -- if present --
   <key-id> (in that order), followed by the addresses, and followed
   by any other values -- for example, other Address Block TLV
   <value>s that are associated with those addresses. A MANET
   routing protocol, or MANET routing protocol extension, using ICV
   Address Block TLVs MUST specify how to include any such

concatenated attribute of the addresses in the verification
process of the ICV. The consideration in Section 10.1 MUST be
applied.

When determining the <ICV-data> for one or more addresses, with type
extension = 2:

o The same procedure as for type extension = 1 is used, except that
   the data used consists of a representation of the source address
   of the IP datagram carrying the Address Block, followed by the
   remaining data (as for type extension = 1). The representation of
   the source address consists of a single octet containing the
   address length, in octets, followed by that many octets containing
   the address in network byte order.

12.3. Example of a Message Including an ICV

The sample message depicted in Figure 1 is derived from Appendix E of
[RFC5444]. The message contains an ICV Message TLV, with the value
representing an ICV that is 16 octets long 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Message Type  | MF=15 | MAL=3 |      Message Length = 82      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Message Originator Address                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Hop Limit   |   Hop Count   |    Message Sequence Number    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Message TLV Block Length = 36 |   TLV Type    |   MTLVF = 16  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Value Len = 6 |                    Value                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  Value (cont)                 |TLV Type = ICV |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   MTLVF = 144  | MTLVExt = 1  |Value Len = 23 |   Hash Func   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Crypto Func  | KeyID Len = 4 |        Key Identifier         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      Key Identifier (cont)    |          ICV Value            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        ICV Value (cont)                       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        ICV Value (cont)                       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        ICV Value (cont)                       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |        ICV Value (cont)       |  Num Addr = 2 |    ABF = 48   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Tail Len = 2  |            Mid 0              |     Mid 1     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Mid 1 (cont)  | Prefix Length |    ABTLV Block Length = 0     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Num Addr = 3  |   ABF = 128   | Head Len = 2  |     Head      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Head (cont)  |            Mid 0              |     Mid 1     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Mid 1 (cont)  |            Mid 2             |ABTLV Block ...|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |... Length = 9 |   TLV Type    |  ABTLVF = 16  | Value Len = 2 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |             Value             |   TLV Type    |  ABTLVF = 32  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Index Start  |  Index Stop   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                    Figure 1: Example Message with ICV

       MF: Message Flags, see Section 5.2 of [RFC5444].
       MAL: Message Address Length, see Section 5.2 of [RFC5444].
       MTLVF: Message TLV Flags, see Section 5.4.1 of [RFC5444].
       MTLVExt: Message TLV Type Extension, see Section 5.4.1 of [RFC5444].
       AF: Address Block Flags, see Section 5.3 of [RFC5444].
       ABTLV: Address Block TLV, see Section 5.4 of [RFC5444].
       ABTLVF: Address Block TLV Flags, see Section 5.4.1 of [RFC5444].

                     Example Message with ICV - Legend

13. IANA Considerations

   The IANA registrations for TLV Types and the TLV type extension
   registries given in this specification replace the identical
   registrations and registries from [RFC6622].

   This specification defines the following TLV Types, replacing the
   original specifications 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 registries that were 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 both TIMESTAMP and ICV TLVs, functionally similar extensions for
Packet, Message, and Address Block TLVs SHOULD be numbered
identically.

13.2. Packet TLV Types

   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 has modified this allocation as
   indicated.

```
                +------+-------------+-----------+
                | Type | Description | Reference |
                +------+-------------+-----------+
                |  5   |     ICV     | RFC 7182  |
                |  6   |  TIMESTAMP  | RFC 7182  |
                +------+-------------+-----------+
```

                   Table 1: Packet TLV Types

13.3. Message TLV Types

   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 has modified this allocation as
   indicated.

```
                +------+-------------+-----------+
                | Type | Description | Reference |
                +------+-------------+-----------+
                |  5   |     ICV     | RFC 7182  |
                |  6   |  TIMESTAMP  | RFC 7182  |
                +------+-------------+-----------+
```

                  Table 2: Message TLV Types

13.4. Address Block TLV Types

   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 has modified this allocation as
   indicated.

```
                  +------+-------------+-----------+
                  | Type | Description | Reference |
                  +------+-------------+-----------+
                  |  5   |     ICV     | RFC 7182  |
                  |  6   |  TIMESTAMP  | RFC 7182  |
                  +------+-------------+-----------+
```

                   Table 3: Address Block TLV Types

13.5. ICV Packet TLV Type Extensions

   IANA has, in accordance with [RFC6622], made allocations from the
   "ICV Packet TLV Type Extensions" namespace of [RFC6622] for the
   Packet TLVs specified in Table 4. IANA has modified this allocation
   (including defining type extension = 2) as indicated.

```
   +-----------+-------------------------------------------+-----------+
   |   Type    |                Description                | Reference |
   | Extension |                                           |           |
   +-----------+-------------------------------------------+-----------+
   |     0     |              ICV of a packet              | RFC 7182  |
   |     1     | ICV, using a cryptographic function and a | RFC 7182  |
   |           | hash function, as specified in Section 12 |           |
   |           |              of this document             |           |
   |     2     | ICV, using a cryptographic function and a | RFC 7182  |
   |           |      hash function, and including the IP  |           |
   |           |   datagram source address, as specified in|           |
   |           |       Section 12 of this document         |           |
   |   3-251   |          Unassigned; Expert Review        |           |
   |  252-255  |       Reserved for Experimental Use       | RFC 7182  |
   +-----------+-------------------------------------------+-----------+
```

                 Table 4: ICV Packet TLV Type Extensions

   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 or with a different key identifier).
   ICV Packet TLVs that carry what is declared to be the same
   information MUST NOT be included in the same packet.

13.6. TIMESTAMP Packet TLV Type Extensions

   IANA has, in accordance with [RFC6622], made allocations from the
   "TIMESTAMP Packet TLV Type Extensions" namespace of [RFC6622] for the
   Packet TLVs specified in Table 5. IANA has modified this allocation
   as indicated.

+-----------+---+-----------+
| Type | Description | Reference |
| Extension | | |
+-----------+---+-----------+
0	Unsigned timestamp of arbitrary length,	RFC 7182
	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	RFC 7182
	in [IEEE1003.1-2008]	
2	NTP timestamp format, as specified in	RFC 7182
	[RFC5905]	
3	Signed timestamp of arbitrary length with	RFC 7182
	no constraints such as monotonicity. In	
	particular, it may represent any random	
	value	
4-251	Unassigned; Expert Review	
252-255	Reserved for Experimental Use	RFC 7182
+-----------+---+-----------+

                Table 5: TIMESTAMP Packet TLV Type Extensions

   More than one TIMESTAMP Packet TLV with the same type extension MUST
   NOT be included in a packet.

13.7. ICV Message TLV Type Extensions

   IANA has, in accordance with [RFC6622], made allocations from the
   "ICV Message TLV Type Extensions" namespace of [RFC6622] for the
   Message TLVs specified in Table 6. IANA has modified this allocation
   (including defining type extension = 2) as indicated.

```
+-----------+-------------------------------------------+-----------+
|   Type    |               Description                 | Reference |
| Extension |                                           |           |
+-----------+-------------------------------------------+-----------+
|     0     |              ICV of a message             | RFC 7182  |
|     1     | ICV, using a cryptographic function and a | RFC 7182  |
|           | hash function, as specified in Section 12 |           |
|           |               of this document            |           |
|     2     | ICV, using a cryptographic function and a | RFC 7182  |
|           |   hash function, and including the IP      |           |
|           |  datagram source address, as specified in |           |
|           |      Section 12 of this document          |           |
|   3-251   |          Unassigned; Expert Review        |           |
|  252-255  |        Reserved for Experimental Use      | RFC 7182  |
+-----------+-------------------------------------------+-----------+
```

                  Table 6: ICV Message TLV Type Extensions

   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 or with a different key identifier).
   ICV Message TLVs that carry what is declared to be the same
   information MUST NOT be included in the same message.

13.8. TIMESTAMP Message TLV Type Extensions

   IANA has, in accordance with [RFC6622], made allocations from the
   "TIMESTAMP Message TLV Type Extensions" namespace of [RFC6622] for
   the Message TLVs specified in Table 7. IANA has modified this
   allocation as indicated.

```
+-----------+--------------------------------------+-----------+
|   Type    |             Description              | Reference |
| Extension |                                      |           |
+-----------+--------------------------------------+-----------+
|     0     | Unsigned timestamp of arbitrary      | RFC 7182  |
|           | 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        | RFC 7182  |
|           | specified in POSIX [IEEE1003.1-2008] |           |
|     2     | NTP timestamp format, as specified   | RFC 7182  |
|           | in [RFC5905]                         |           |
|     3     | Signed timestamp of arbitrary length | RFC 7182  |
|           | with no constraints such as          |           |
|           | monotonicity.  In particular, it may |           |
|           | represent any random value           |           |
|   4-251   | Unassigned; Expert Review            |           |
|  252-255  | Reserved for Experimental Use        | RFC 7182  |
+-----------+--------------------------------------+-----------+
```

Table 7: TIMESTAMP Message TLV Type Extensions

More than one TIMESTAMP Message TLV with the same type extension MUST
NOT be included in a message.

13.9. ICV Address Block TLV Type Extensions

IANA has, in accordance with [RFC6622], made allocations from the
"ICV Address Block TLV Type Extensions" namespace of [RFC6622] for
the Address Block TLVs specified in Table 8. IANA has modified this
allocation (including defining type extension = 2) as indicated.

```
+-----------+-------------------------------------------+-----------+
|   Type    |               Description                 | Reference |
| Extension |                                           |           |
+-----------+-------------------------------------------+-----------+
|     0     |     ICV of an object (e.g., an address)   | RFC 7182  |
|     1     |  ICV, using a cryptographic function and a| RFC 7182  |
|           |  hash function, as specified in Section 12|           |
|           |              of this document             |           |
|     2     |  ICV, using a cryptographic function and a| RFC 7182  |
|           |    hash function, and including the IP     |           |
|           |  datagram source address, as specified in |           |
|           |     Section 12 of this document           |           |
|  3-251    |         Unassigned; Expert Review          |           |
|  252-255  |        Reserved for Experimental Use       | RFC 7182  |
+-----------+-------------------------------------------+-----------+
```

                Table 8: ICV Address Block TLV Type Extensions

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

13.10. TIMESTAMP Address Block TLV Type Extensions

   IANA has, in accordance with [RFC6622], made allocations from the
   "TIMESTAMP Address Block TLV Type Extensions" namespace of [RFC6622]
   for the Address Block TLVs specified in Table 9. IANA has modified
   this allocation as indicated.

+-----------+--+-----------+
| Type | Description | Reference |
| Extension | | |
+-----------+--+-----------+
0	Unsigned timestamp of arbitrary length,	RFC 7182
	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	RFC 7182
	in POSIX [IEEE1003.1-2008]	
2	NTP timestamp format, as specified in	RFC 7182
	[RFC5905]	
3	Signed timestamp of arbitrary length with	RFC 7182
	no constraints such as monotonicity. In	
	particular, it may represent any random	
	value	
4-251	Unassigned; Expert Review	
252-255	Reserved for Experimental Use	RFC 7182
+-----------+--+-----------+

          Table 9: TIMESTAMP Address Block TLV Type Extensions

   More than one TIMESTAMP Address Block TLV with the same type
   extension MUST NOT be associated with any address.

13.11. Hash Functions

   IANA has, in accordance with [RFC6622], created a 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 10. IANA has modified this
   allocation as indicated.

Value	Algorithm	Description	Reference
0	none	The "identity function": The hash value of an object is the object itself	RFC 7182
1	SHA-1	[NIST-FIPS-180-4]	RFC 7182
2	SHA-224	[NIST-FIPS-180-4]	RFC 7182
3	SHA-256	[NIST-FIPS-180-4]	RFC 7182
4	SHA-384	[NIST-FIPS-180-4]	RFC 7182
5	SHA-512	[NIST-FIPS-180-4]	RFC 7182
6-251		Unassigned; Expert Review	
252-255		Reserved for Experimental Use	RFC 7182

Table 10: Hash Function Registry

13.12. Cryptographic Functions

   IANA has, in accordance with [RFC6622], created a registry for the
   cryptographic functions, as specified in Section 12 of this document.
   Initial assignments and allocation policies are specified in
   Table 11. IANA has modified this allocation as indicated.

Value	Algorithm	Description	Reference
0	none	The "identity function": The value of an encrypted hash is the hash itself	RFC 7182
1	RSA	[RFC3447]	RFC 7182
2	DSA	[NIST-FIPS-186-4]	RFC 7182
3	HMAC	[RFC2104]	RFC 7182
4	3DES	[NIST-SP-800-67]	RFC 7182
5	AES	[NIST-FIPS-197]	RFC 7182
6	ECDSA	[RFC6090]	RFC 7182
7-251		Unassigned; Expert Review	
252-255		Reserved for Experimental Use	RFC 7182

Table 11: 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.

   The authors would like to thank Justin Dean (NRL) and Henning Rogge
   (FGAN) for their constructive comments on this specification.

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.

   [IEEE1003.1-2008]
              IEEE, "Portable Operating System Interface (POSIX)", IEEE
              1003.1-2008, Base Specifications, Issue 7, December 2008.

   [NIST-FIPS-180-4]
              National Institute of Standards and Technology, "Secure
              Hash Standard (SHS)", FIPS 180-4, March 2012.

   [NIST-FIPS-186-4]
              National Institute of Standards and Technology, "Digital
              Signature Standard (DSS)", FIPS 186-4, July 2013.

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

   [NIST-SP-800-107]
              National Institute of Standards and Technology,
              "Recommendation for Applications Using Approved Hash
              Algorithms", SP 800-107, Revision 1, August 2012.

   [NIST-SP-800-67]
              National Institute of Standards and Technology,
              "Recommendation for the Triple Data Encryption Algorithm
              (TDEA) Block Cipher", Special Publication 800-67, Revision
              1, January 2012.

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

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

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

   [RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The
              AES-CMAC Algorithm", RFC 4493, June 2006.

   [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., Burbank, J., and W. Kasch, "Network
              Time Protocol Version 4: Protocol and Algorithms
              Specification", RFC 5905, June 2010.

   [RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic
              Curve Cryptography Algorithms", RFC 6090, February 2011.

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.

   [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg,
              "The Optimized Link State Routing Protocol Version 2", RFC
              7181, April 2014.

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 Advanced Technology Centre
    West Hanningfield Road
    Great Baddow, Chelmsford
    United Kingdom

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