

Mobile Ad hoc Networking (MANET)	U. Herberg
Internet-Draft	T. Clausen
Intended status: Standards Track	LIX, Ecole Polytechnique
Expires: January 12, 2012	July 11, 2011

MANET Cryptographical Signature TLV Definition  
draft-ietf-manet-packetbb-sec-04

## [Abstract](#)

This document describes general and flexible TLVs (type-length-value structure) for representing cryptographic signatures as well as timestamps, using the generalized MANET packet/message format [\[RFC5444\]](#). It defines two Packet TLVs, two Message TLVs, and two Address Block TLVs, for affixing cryptographic signatures and timestamps to a packet, message and address, respectively.

## [Status of this Memo](#)

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on January 12, 2012.

## [Copyright Notice](#)

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## [Table of Contents](#)

- \*1. [Introduction](#)
- \*2. [Terminology](#)

- \*3. [Applicability Statement](#)
- \*4. [Security Architecture](#)
- \*5. [Protocol Overview and Functioning](#)
- \*6. [Imported TLV Fields](#)
- \*7. [General Signature TLV Structure](#)
- \*8. [General Timestamp TLV Structure](#)
- \*9. [Packet TLVs](#)
  - \*9.1. [Packet SIGNATURE TLV](#)
  - \*9.2. [Packet TIMESTAMP TLV](#)
- \*10. [Message TLVs](#)
  - \*10.1. [Message SIGNATURE TLV](#)
  - \*10.2. [Message TIMESTAMP TLV](#)
- \*11. [Address Block TLVs](#)
  - \*11.1. [Address Block SIGNATURE TLV](#)
  - \*11.2. [Address Block TIMESTAMP TLV](#)
- \*12. [IANA Considerations](#)
  - \*12.1. [TLV Registrations](#)
    - \*12.1.1. [Expert Review: Evaluation Guidelines](#)
    - \*12.1.2. [Packet TLV Type Registrations](#)
    - \*12.1.3. [Message TLV Type Registrations](#)
    - \*12.1.4. [Address Block TLV Type Registrations](#)
  - \*12.2. [New IANA Registries](#)
    - \*12.2.1. [Expert Review: Evaluation Guidelines](#)
    - \*12.2.2. [Hash Function](#)
    - \*12.2.3. [Cryptographic Algorithm](#)
- \*13. [Security Considerations](#)

- \*14. [Acknowledgements](#)
- \*15. [References](#)
  - \*15.1. [Normative References](#)
  - \*15.2. [Informative References](#)
- \*Appendix A. [Signature Decomposition into Cryptographic Function of a Hash Value](#)
  - \*Appendix A.1. [General Signature TLV Structure](#)
    - \*Appendix A.1.1. [Rationale](#)
  - \*Appendix A.2. [Considerations for Calculating the Signature](#)
    - \*Appendix A.2.1. [Packet SIGNATURE TLV](#)
    - \*Appendix A.2.2. [Message SIGNATURE TLV](#)
    - \*Appendix A.2.3. [Address Block SIGNATURE TLV](#)
  - \*Appendix A.3. [Example of a Signed Message](#)
- \*[Authors' Addresses](#)

## **[1. Introduction](#)**

This document specifies:

- \*two TLVs for carrying cryptographic signatures and timestamps in packets, messages and address blocks as defined by [\[RFC5444\]](#),
- \*a generic framework for calculating cryptographic signatures, taking (for Message TLVs) into account the mutable message header fields (<msg-hop-limit> and <msg-hop-count>) where these fields are present in messages,
- \*a specific calculation of signatures, decomposed as a cryptographic function over the hash value of the content to be signed, in the [Appendix Appendix A](#) of this document.

This document requests from IANA:

- \*allocations for these Packet, Message, and Address Block TLVs from the 0-223 Packet TLV range, the 0-127 Message TLV range and the 0-127 Address Block TLV range from [\[RFC5444\]](#),
- \*creation of two IANA registries for recording code points for hash function and signature calculation, respectively.

## 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 [\[RFC5444\]](#).

## 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 inclusion of TLVs. Information so included MAY be used by a routing protocol, or by an extension of a routing protocol, according to its specification.

This document specifies how to include a cryptographic signature for a packet, message or address by way of such TLVs. This document also specifies how to treat "mutable" fields (<msg-hop-count> and <msg-hop-limit>), if present, in the message header when calculating signatures, such that the resulting signature can be correctly verified by any recipient, and how to include this signature.

This document is split into two parts: (i) a generic framework of creating signatures in the presence of mutable fields, and how to include these signatures in TLVs, (ii) a specific description of how to calculate a signature, using a cryptographic function over the hash value of the content to be signed, in the [Appendix Appendix A](#) of this document. Note that (ii) is a possible and widely-used way of calculating a signature, but other means may exist. Such other means of calculating a signature have to be specified in another document. That new document MUST use the TLV structures specified in this document, as well as the described considerations when calculating the signatures.

## 4. Security Architecture

Basic MANET routing protocol specifications are often "oblivious to security", however have a clause allowing a control message to be rejected as "badly formed" prior to it being processed or forwarded. Protocols such as [\[RFC6130\]](#) and [\[OLSRv2\]](#) recognize external reasons (such as failure to verify a signature) for rejecting a message as "badly formed", and therefore "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 deployment-domain specific security mechanisms.

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

- \*Security-oblivious MANET routing protocol specifications, with a clause allowing an extension to reject a message (prior to processing/forwarding) as "badly formed".
- \*MANET routing protocol security extensions, rejecting messages as "badly formed", as appropriate for a given deployment-domain specific security requirement.
- \*Code-points and an exchange format for information, necessary for specification of such MANET routing protocol security extensions.

This document addresses the last of these issues, by specifying a common exchange format for cryptographic signatures, 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 a signature into a cryptographic function over a hash value, specified in [Appendix Appendix A](#), this document establishes two IANA registries for code-points for hash functions and cryptographic functions adhering to [\[RFC5444\]](#).

With respect to [\[RFC5444\]](#), this document:

- \*is intended to be used in the non-normative, but intended, mode of use of [\[RFC5444\]](#) as described in its Appendix B.
- \*is a specific example of the Security Considerations section of [\[RFC5444\]](#) (the authentication part).

## **5. Protocol Overview and Functioning**

This specification does not describe a protocol, nor does it mandate specific router or protocol behavior. It represents a purely syntactical representation of security related information for use with [\[RFC5444\]](#) addresses, messages and packets, as well as establishes IANA registrations and registries.

## **6. Imported TLV Fields**

In this specification, the following TLV fields from [\[RFC5444\]](#) are used:

- <msg-hop-limit> - hop limit of a message, as specified in Section 5.2 of [\[RFC5444\]](#).
- <msg-hop-count> - hop count of a message, as specified in Section 5.2 of [\[RFC5444\]](#).

**<length>**

- length of a TLV in octets, as specified in Section 5.4.1 of [\[RFC5444\]](#).

## **7. General Signature TLV Structure**

The following data structure allows a generic representation of a cryptographic signature. This <signature> data structure is specified, using the regular expression syntax of [\[RFC5444\]](#), as:

**<signature> := <signature-value>**

This generic specification allows for adding a signature in a TLV, using TLV type extension 0, and does not stipulate how to calculate the signature-value. [Appendix Appendix A](#) specifies a concrete calculation of the signature-value, using a cryptographic function over a hash function of the content to be signed. Other methods of how to calculate the signature-value may be specified in future documents.

## **8. General Timestamp TLV Structure**

The following data structure allows the representation of a timestamp. This <timestamp> data structure is specified as:

**<timestamp> := <time-value>**

where:

**<time-value>** is an unsigned integer field, whose length is <length>, and which contains the timestamp. The value of this variable is to be interpreted by the routing protocol as specified by the type extension of the Timestamp TLV, see [Section 12](#).

A timestamp is essentially "freshness information". As such, its setting and interpretation is to be determined by the routing protocol (or the extension to a routing protocol) that uses it, and may e.g. correspond to a UNIX-timestamp, GPS timestamp or a simple sequence number.

## **9. Packet TLVs**

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

### 9.1. Packet SIGNATURE TLV

A Packet SIGNATURE TLV is an example of a Signature TLV as described in [Section 7](#).

The following considerations apply:

- \*As packets defined in [\[RFC5444\]](#) are never forwarded by routers, it is unnecessary to consider mutable fields (e.g. <msg-hop-count> and <msg-hop-limit>), if present, when calculating the signature.
- \*any Packet SIGNATURE TLVs already present in the Packet TLV block MUST be removed before calculating the signature, and the Packet TLV block size MUST be recalculated accordingly. The TLVs can be restored after having calculated the signature value.

The rationale for removing any Packet SIGNATURE TLV already present prior to calculating the signature, is that several signatures may be added to the same packet, e.g., using different signature functions.

### 9.2. Packet TIMESTAMP TLV

A Packet TIMESTAMP TLV is an example of a Timestamp TLV as described in [Section 8](#). If a packet contains a TIMESTAMP TLV and a SIGNATURE TLV, the TIMESTAMP TLV SHOULD be added to the packet before any SIGNATURE TLV, in order that it be included in the calculation of the signature.

## 10. Message TLVs

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

### 10.1. Message SIGNATURE TLV

A Message SIGNATURE TLV is an example of a Signature TLV as described in [Section 7](#). When determining the <signature-value> for a message, the following considerations must be applied:

- \*the fields <msg-hop-limit> and <msg-hop-count>, if present, MUST both be assumed to have the value 0 (zero) when calculating the signature.
- \*any Message SIGNATURE TLVs already present in the Message TLV block MUST be removed before calculating the signature, and the message size as well as the Message TLV block size MUST be recalculated accordingly. The TLVs can be restored after having calculated the signature value.

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

## **10.2. Message TIMESTAMP TLV**

A Message TIMESTAMP TLV is an example of a Timestamp TLV as described in [Section 8](#). If a message contains a TIMESTAMP TLV and a SIGNATURE TLV, the TIMESTAMP TLV SHOULD be added to the message before the SIGNATURE TLV, in order that it be included in the calculation of the signature.

## **11. Address Block TLVs**

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

### **11.1. Address Block SIGNATURE TLV**

An Address Block SIGNATURE TLV is an example of a Signature TLV as described in [Section 7](#). The signature is calculated over the address, concatenated with any other values, for example, any other TLV value that is associated with that address. A routing protocol or routing protocol extension using Address Block SIGNATURE TLVs MUST specify how to include any such concatenated attribute of the address in the verification process of the signature.

### **11.2. Address Block TIMESTAMP TLV**

An Address Block TIMESTAMP TLV is an example of a Timestamp TLV as described in [Section 8](#). If both a TIMESTAMP TLV and a SIGNATURE TLV are associated with an address, the timestamp value should be considered when calculating the value of the signature.

## **12. IANA Considerations**

This section specifies requests to IANA.

### **12.1. TLV Registrations**

This specification defines:

- \*two Packet TLV types which must be allocated from the 0-223 range of the "Assigned Packet TLV Types" repository of [\[RFC5444\]](#) as specified in [Table 1](#),
- \*two Message TLV types which must be allocated from the 0-127 range of the "Assigned Message TLV Types" repository of [\[RFC5444\]](#) as specified in [Table 2](#),



\*and two Address Block TLV types which must be allocated from the 0-127 range of the "Assigned Address Block TLV Types" repository of [\[RFC5444\]](#) as specified in [Table 3](#).

This specification requests:

\*set up of type extension registries for these TLV types.

IANA is requested to assign the same numerical value to the Packet TLV, Message TLV and Address Block TLV types with the same name.

#### **[12.1.1.](#) Expert Review: Evaluation Guidelines**

For the registries for TLV type extensions where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [\[RFC5444\]](#). For the Timestamp TLV, the same type extensions for all Packet, Message and Address TLVs should be numbered identically.

#### **[12.1.2.](#) Packet TLV Type Registrations**

The Packet TLVs as specified in [Table 1](#) must be allocated from the "Packet TLV Types" namespace of [\[RFC5444\]](#).

Name	Type	Type Extension	Description
SIGNATURE	TBD1	0	Signature of a packet
		1	Signature, decomposed into cryptographic function over a hash value, as specified in <a href="#">Appendix Appendix A</a> in this document.
		2-223	Expert Review
		224-255	Experimental Use
TIMESTAMP	TBD2	0	Unsigned timestamp of arbitrary length, given by the TLV length field. The MANET routing protocol has to define how to interpret this timestamp
		1-223	Expert Review
		224-255	Experimental Use

Packet TLV types

#### **[12.1.3.](#) Message TLV Type Registrations**

The Message TLVs as specified in [Table 2](#) must be allocated from the "Message TLV Types" namespace of [\[RFC5444\]](#).

Name	Type	Type Extension	Description
SIGNATURE	TBD3	0	Signature of a message
		1	Signature, decomposed into cryptographic function over a hash value, as specified in <a href="#">Appendix Appendix A</a> in this document.
		2-223	Expert Review
		224-255	Experimental Use
TIMESTAMP	TBD4	0	Unsigned timestamp of arbitrary length, given by the TLV length field.
		1-223	Expert Review
		224-255	Experimental Use

Message TLV types

#### [12.1.4.](#) Address Block TLV Type Registrations

The Address Block TLVs as specified in [Table 3](#) must be allocated from the "Address Block TLV Types" namespace of [\[RFC5444\]](#).

Name	Type	Type Extension	Description
SIGNATURE	TBD5	0	Signature of an object (e.g. an address)
		1	Signature, decomposed into cryptographic function over a hash value, as specified in <a href="#">Appendix Appendix A</a> in this document.
		2-223	Expert Review
		224-255	Experimental Use
TIMESTAMP	TBD6	0	Unsigned timestamp of arbitrary length, given by the TLV length field.
		1-223	Expert Review
		224-255	Experimental Use

Address Block TLV types

#### [12.2.](#) New IANA Registries

This document introduces three namespaces that have been registered: Packet TLV Types, Message TLV Types, and Address Block TLV Types. This section specifies IANA registries for these namespaces and provides guidance to the Internet Assigned Numbers Authority regarding registrations in these namespaces.

The following terms are used with the meanings defined in [\[BCP26\]](#): "Namespace", "Assigned Value", "Registration", "Unassigned", "Reserved", "Hierarchical Allocation", and "Designated Expert".

The following policies are used with the meanings defined in [\[BCP26\]](#): "Private Use", "Expert Review", and "Standards Action".

#### [12.2.1. Expert Review: Evaluation Guidelines](#)

For the registries for the following tables where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [\[RFC5444\]](#).

#### [12.2.2. Hash Function](#)

IANA is requested to create a new registry for the hash functions that can be used when creating a signature, as specified in the [Appendix Appendix A](#) of this document. The initial assignments and allocation policies are specified in [Table 4](#).

Hash function value	Algorithm	Description
0	none	The "identity function": the hash value of an object is the object itself
1-223		Expert Review
224-255		Experimental Use

Hash-Function registry

#### [12.2.3. Cryptographic Algorithm](#)

IANA is requested to create a new registry for the cryptographic function, as specified in the [Appendix Appendix A](#) of this document. Initial assignments and allocation policies are specified in [Table 5](#).

Cryptographic function value	Algorithm	Description
0	none	The "identity function": the value of an encrypted hash is the hash itself
1-223		Expert Review
224-255		Experimental Use

Cryptographic function registry

### [13. Security Considerations](#)

This document does not specify a protocol itself. However, it provides a syntactical component for cryptographic signatures of messages and packets as defined in [\[RFC5444\]](#). It can be used to address security issues of a protocol or extension that uses the component specified in this document. As such, it has the same security considerations as [\[RFC5444\]](#).

In addition, a protocol that includes this component MUST specify the usage as well as the security that is attained by the cryptographic signatures of a message or a packet.

As an example, a routing protocol that uses this component to reject "badly formed" messages if a control message does not contain a valid signature, should indicate the security assumption that if the signature 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).

## **14. Acknowledgements**

The authors would like to thank Jerome Milan (Ecole Polytechnique) for his advice as cryptographer. In addition, many thanks to Bo Berry (Cisco), Alan Cullen (BAE), Justin Dean (NRL), Christopher Dearlove (BAE), Paul Lambert (Marvell), and Henning Rogge (FGAN) for their constructive comments on the document.

## **15. References**

### **15.1. Normative References**

<b>[BCP26]</b>	Narten, T. and H. Alvestrand, " <a href="#">Guidelines for Writing an IANA Considerations Section in RFCs</a> ", RFC 5226, BCP 26, May 2008.
<b>[RFC2119]</b>	Bradner, S., " <a href="#">Key words for use in RFCs to Indicate Requirement Levels</a> ", RFC 2119, BCP 14, March 1997.
<b>[RFC5444]</b>	Clausen, T.H., Dearlove, C.M., Dean, J.W. and C. Adjih, " <a href="#">Generalized MANET Packet/Message Format</a> ", RFC 5444, February 2009.

### **15.2. Informative References**

<b>[RFC6130]</b>	Clausen, T.H., Dean, J.W. and C.M. Dearlove, " <a href="#">MANET Neighborhood Discovery Protocol (NHDP)</a> ", RFC 6130, March 2011.
<b>[OLSRv2]</b>	Clausen, T.H., Dearlove, C.M. and P. Jacquet, "The Optimized Link State Routing Protocol version 2", work in progress draft-ietf-manet-olsrv2-11.txt, April 2010.

## **Appendix A. Signature Decomposition into Cryptographic Function of a Hash Value**

This section specifies how to calculate the signature-value in a Signature TLV, as described in [Section 7](#). A common way of calculating a signature is applying a cryptographic function on a hash value of the content. This decomposition is specified in the following, using a type extension of 1 in the Signature TLVs.

## [Appendix A.1.](#) General Signature TLV Structure

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

```
<signature> := <hash-function>
               <cryptographic-function>
               <key-index>
               <signature-value>
```

where: [Section 12](#).

**<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-index>** is an 8-bit unsigned integer field specifying the key index of the key which was used to sign 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 indices and that all key indices have a value unequal to 0x00. Value 0x00 is reserved for a pre-installed, shared key.

**<signature-value>** is an unsigned integer field, whose length is <length> - 3, and which contains the cryptographic signature.

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

$$\text{signature-value} = \text{cryptographic-function}(\text{hash-function}(\text{content}))$$

The hash function and the cryptographic function correspond to the entries in two IANA registries, set up by this specification in

### [Appendix A.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 TLV type extension - is twofold: First, if further hash functions or cryptographic functions are added in the future, the number space might not remain continuous. More importantly, the number space of possible combinations would be rapidly exhausted. As new or improved cryptographic mechanism are continuously being developed and introduced, this format should be able to accommodate such for the foreseeable future.

The rationale for not including a field that lists parameters of the cryptographic signature in the TLV is, that before being able to validate a cryptographic signature, 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 inherently included parameter is the length of the signature, which is `<length> - 3` and which depends on the choice of the cryptographic function.

## [Appendix A.2.](#) **Considerations for Calculating the Signature**

In the following, considerations are listed, which have to be applied when calculating the signature for Packet, Message and Address SIGNATURE TLVs, respectively.

### [Appendix A.2.1.](#) **Packet SIGNATURE TLV**

When determining the `<signature-value>` for a Packet, the signature is calculated over the three fields `<hash-function>`, `<cryptographic-function>` and `<key-index>` (in that order), concatenated with the entire Packet, including the packet header, all Packet TLVs (other than Packet SIGNATURE TLVs) and all included Messages and their message headers.

### [Appendix A.2.2.](#) **Message SIGNATURE TLV**

When determining the `<signature-value>` for a message, the signature is calculated over the three fields `<hash-function>`, `<cryptographic-function>`, and `<key-index>` (in that order), concatenated with the entire message.

### [Appendix A.2.3.](#) **Address Block SIGNATURE TLV**

When determining the `<signature-value>` for an address, the signature is calculated over the three fields `<hash-function>`, `<cryptographic-function>`, and `<key-index>` (in that order), concatenated with the address, concatenated with any other values, for example, any other TLV value that is associated with that address. A routing protocol or routing protocol extension using Address Block SIGNATURE TLVs MUST specify how to include any such concatenated attribute of the address in the verification process of the signature.

## [Appendix A.3.](#) **Example of a Signed Message**

The sample message depicted in [Figure 4](#) is derived from the appendix of [\[RFC5444\]](#). A SIGNATURE Message TLV has been added, with the value representing a 14 octet long signature of the whole message. The type extension of the Message TLV is 1, for the specific decomposition of a signature into a cryptographic function over a hash value, as specified in [Appendix Appendix A](#).

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								
0 0 0 0 1 0 0 0										Packet Sequence Number										Message Type																			
1 1 1 1 0 0 1 1										0 0 0 0 0 0 0 0 1 0 0 1 1 0 0										Orig Addr																			
										Originator Address (cont)										Hop Limit																			
Hop Count										Message Sequence Number										0 0 0 0 0 0 0 0																			
0 0 0 1 1 1 1 0										SIGNATURE   1 0 0 1 0 0 0 0										0 0 0 0 0 0 0 1																			
0 0 0 1 0 0 1 0										Hash Func   Crypto Func										Key Index																			
										Signature Value																													
										Signature Value (cont)																													
										Signature Value (cont)																													
Signature Value (cont)										TLV Type										0 0 0 1 0 0 0 0																			
0 0 0 0 0 1 1 0										Value																													
										Value (cont)										0 0 0 0 0 0 1 0																			
0 0 1 1 0 0 0 0										0 0 0 0 0 0 1 0										Mid																			
										Mid   Prefix Length										0 0 0 0 0 0 0 0																			
0 0 0 0 0 0 0 0										0 0 0 0 0 0 1 1										1 0 0 0 0 0 0 0 0										0 0 0 0 0 0 1 0									
										Head										Mid																			
										Mid										Mid																			
0 0 0 0 0 0 0 0										0 0 0 0 0 0 1 0 0 1										TLV Type										0 0 0 1 0 0 0 0									
0 0 0 0 0 0 1 0										Value										TLV Type																			
0 0 1 0 0 0 0 0										Index Start										Index Stop																			

### Authors' Addresses

Ulrich Herberg Herberg LIX, Ecole Polytechnique 91128 Palaiseau  
Cedex, France Phone: +33 1 6933 4126 EMail: [ulrich@herberg.name](mailto:ulrich@herberg.name) URI:  
<http://www.herberg.name/>

Thomas Heide Clausen Clausen LIX, Ecole Polytechnique 91128  
Palaiseau Cedex, France Phone: +33 6 6058 9349 EMail:  
[T.Clausen@computer.org](mailto:T.Clausen@computer.org) URI: <http://www.thomasclausen.org/>