

INTERNET DRAFT
Intended Status: Informational

T. Polk
NIST
R. Housley
Vigil Security
26 October 2009

Expires: 29 April 2010

Routing Authentication Using A Database of Long-Lived Cryptographic Keys
[draft-polk-saag-rtg-auth-keytable-01.txt](#)

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Abstract

This document describes the application of a database of long-lived cryptographic keys to establish session-specific cryptographic keys to support authentication services in routing protocols. Keys may be established between two peers for pair-wise communications, or between groups of peers for multicast traffic.

INTERNET DRAFT

19 October 2009

Table of Contents

1	Introduction	2
1.1	Terminology	2
2	Architecture and Design	2
3	Pair-wise Application	3
4	Identifier Mapping	5
4.1	Selected Range Reservation	6
4.2	Protocol Specific Mapping Tables	6
5	Worked Example: TCP-AO	6
5.1	Setup	7
5.2	Protocol Operation: Xp Initiates a Connection	8
5.3	Protocol Operation: Yp Initiates a Connection	8
6	Security Considerations	9
6	IANA Considerations	9
7	References	9
7.1	Normative References	9
7.2	Informative References	9
	Author's Addresses	9
	Full Copyright Statement	11

[1](#) Introduction

This document describes the application of a database of long-lived cryptographic keys, as defined in [\[KEYTAB\]](#), to establish session-specific cryptographic keys to provide authentication services in routing protocols. Keys may be established between two peers for pair-wise communications, or between groups of peers for multicast traffic.

[1.1](#) Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [\[RFC2119\]](#).

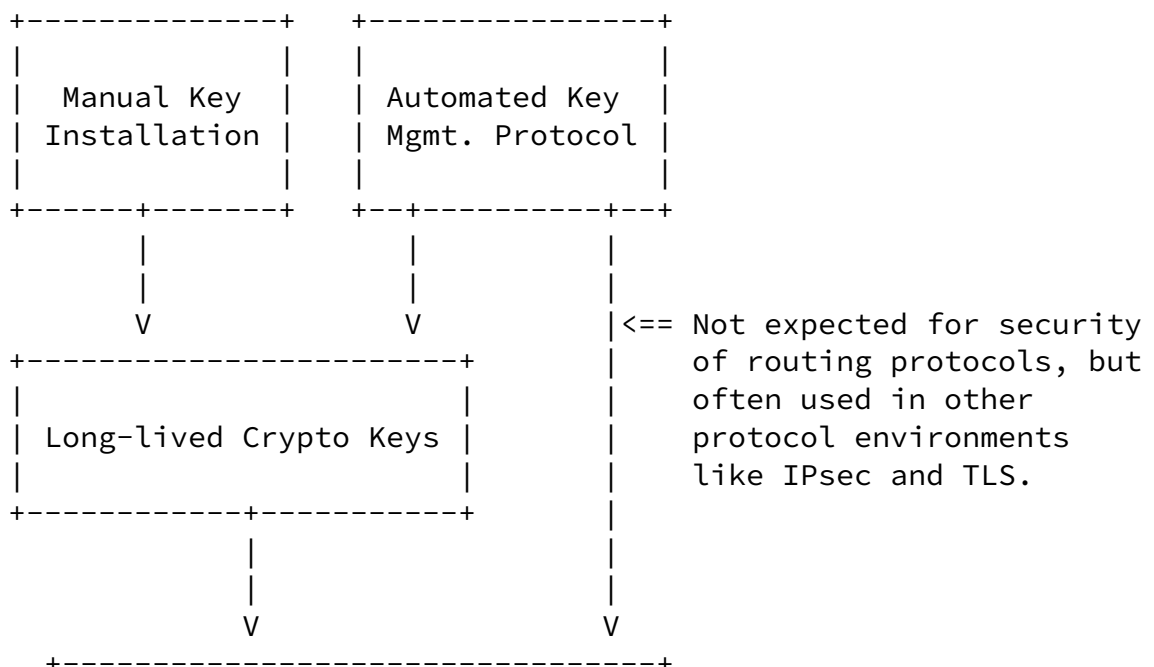
[2](#) Architecture and Design

INTERNET DRAFT

19 October 2009

Figure 1 illustrates the establishment and use of cryptographic keys for authentication in routing protocols. Long-lived cryptographic keys are inserted in a database manually. In the future, we anticipate an automated key management protocol to insert these keys in the database. In this future environment, we do not anticipate an environment where the automated key management protocol will be used to create short-lived cryptographic session keys. The structure of the database of long-lived cryptographic keys is described in [\[KEYTAB\]](#).

The cryptographic keying material for individual sessions is derived from the keying material stored in the database of long-lived cryptographic keys. A key derivation function (KDF) and its inputs are named in the database of long-lived cryptographic keys; session specific values based on the routing protocol are input the the KDF. Protocol specific key identifiers may be assigned to the cryptographic keying material for individual sessions if needed.



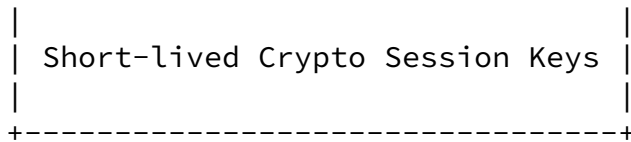


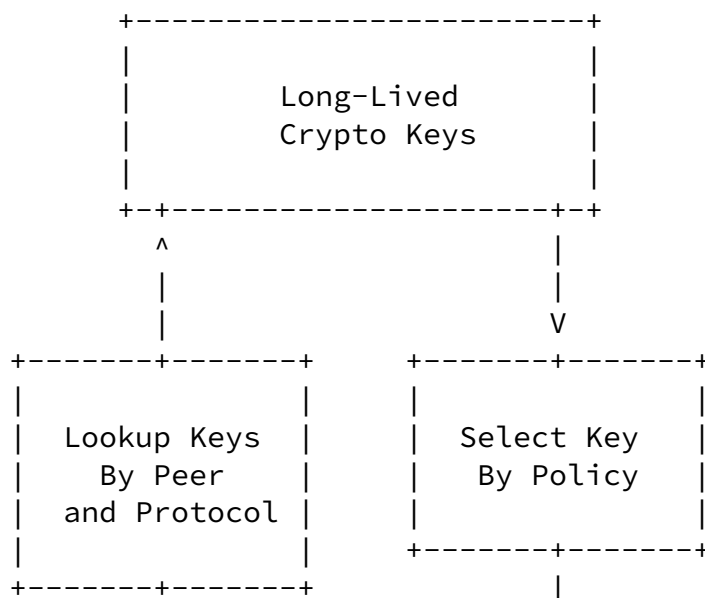
Figure 1. Cryptographic key establishment and use.

3 Pair-wise Application

Figure 2 illustrates how the long-lived cryptographic keys are accessed and employed when an entity wishes to establish a protected

session with a peer. As one step in the initiation process, the initiator requests the set of long term keys associated with the peer for the particular protocol. If the set contains more than one key, the initiator selects one long-term key based on the local policy. The long-term key is provided as an input, along with session-specific information (e.g., ports or initial counters), to a key derivation function. The result is session-specific key material which is used to generate cryptographic authentication.

Where the initiator is establishing a multicast session, the Peer in the key request identifies the set of systems that will receive this information.



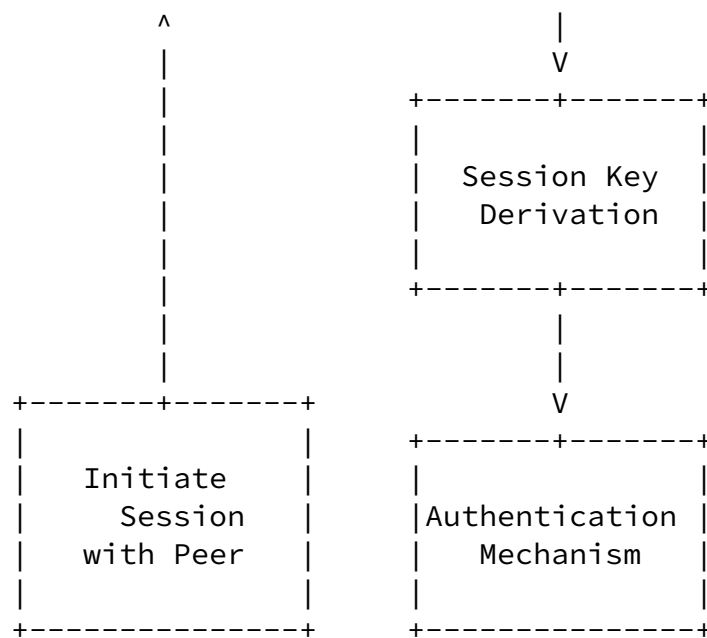
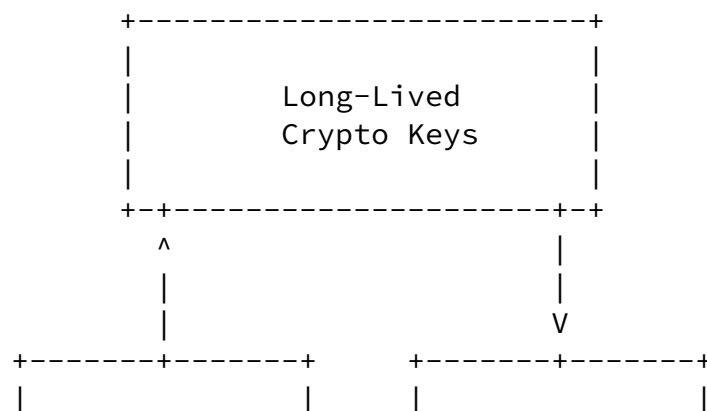
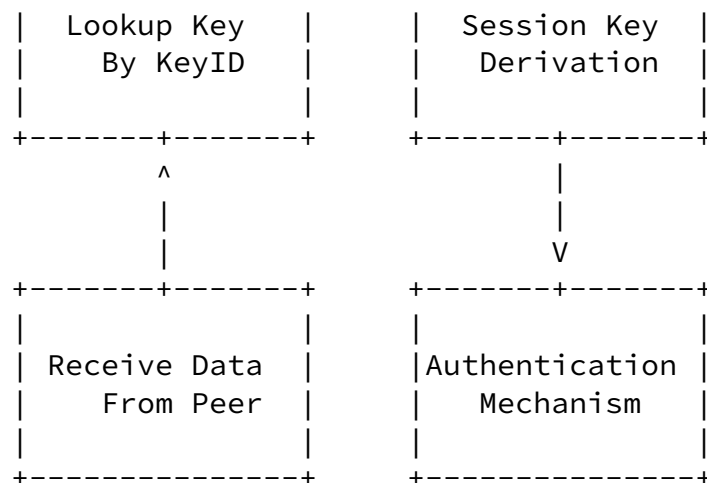


Figure 3 illustrates how an entity that receives a session generates the necessary long-lived cryptographic keys to verify data when a protected session is requested. As step one in the initiation process, the receiver extracts the keyID for the long-term keyID from the received data. The receiver then requests the specified long-term key from the table. The long-term key is provided as an input, along with session-specific information (e.g., ports or initial counters), to a key derivation function. The result is session-specific key material which is used to verify the cryptographic authentication information.





[4 Identifier Mapping](#)

[KEYTAB] specifies a 16-bit identifier, but protocols already exist with key identifiers of various sizes. Where the identifiers are of different sizes, an extra mapping step may be required. Note that mapping mechanisms are local – that is, different mapping mechanisms could be employed on different peers.

In practice, the mapping process need only be applied to the LocalKeyID, whose value must be unique in the context of the

database, as defined in [[KEYTAB](#)]. Uniqueness is not required for the PeerKeyID, so mapping is generally restricted to truncation. Mapping would only be needed to expand PeerKeyID's value beyond 16 bits.

[4.1 Selected Range Reservation](#)

Where a protocol uses an index of less than 16 bits, a selected range of the local index space can be reserved for a particular protocol. For example, consider two protocols P1 and P2 that each use 8 bit key identifiers. Without identifier mapping these protocols would share the space {0x0000 through 0x00ff} which would limit the pair of protocols to 256 keys in total. By reserving the ranges {0x7f00 through 0x7fff} and {0x7e00 through 0x7eff} for P1 and P2 respectively permits each protocol to use the full 256 key identifiers and establishes an unambiguous mapping for the protocol

key identifiers and local table identifiers.

When an initiator selects a key from the set in the table, the given key identifier needs to be masked or shifted to the on-the-wire range. Before requesting a specific key, the receiver would use a shim layer would need to map the on-the-wire identifier into the reserved range.

[4.2](#) Protocol Specific Mapping Tables

Each protocol can also maintain a simple mapping table with two fields: the 16 bit index and the protocol specific value

KEYTAB index (16 bits) | Protocol specific index (8 bits)

In this case, the host system would maintain separate mapping tables for protocols P1 and P2.

[5](#) Worked Example: TCP-AO

This section describes the way a TCP-AO implementation could use the database. [[tcpao](#)] TCP-AO protocol is an example where the key identifier is limited to 8 bits, so an identifier mapping is needed.

We will assume two peers Xp and Yp. Xp employs the range reservation method for mapping and has reserved the range {0x7f00 ... 0x7fff} for LocalKeyIDs for TCP-AO, mapping to {0x00 ... 0xff}. Yp employs a protocol specific mapping table in its TCP-AO implementation.

The following subsections describe how peers Xp and Yp make use of the database of long-lived cryptographic keys when Xp and Yp respectively initiate a session. (Note: Rollover to new sessions keys during a session is described in [[tcpao](#)].)

[5.1](#) Setup

The owners of Xp and Yp determine a need for authenticated communication using TCP-AO. They decide to use AES-CMAC-128 for authentication, so a 128 bit key is needed. They decide to use the same key for both directions (inbound and outbound), and that the key will be available from 12/31/2010 through 12/31/2011. Through an out-of-band channel, the administrators establish the shared secret:

0x0123456789ABCDEF0123456789ABCDEF

Peer Xp selects the first available TCP-A0 identifier in the reserved range, which is 0x7f05 and maps to an eight bit identifier 0x05. Peer Yp selects the next available TCP-A0 identifier, 0x12, and the next available LocalKeyID, which is 0x0107. Peer Yp also adds an entry to its TCP-A0 mapping table mapping the LocalKeyID to the TCP-A0 identifier, as shown in Figure 5:

LocalKeyID	TCP-A0 identifier

0x001a	0x01
0x004d	0x02
...	...
0x0107	0x12

Figure 5. Protocol Specific KeyID Mapping Table for TCP-A0

After exchanging the TCP-A0 identifiers, the peers have sufficient information to establish their [\[KEYTAB\]](#) entries. Peer Xp's [\[KEYTAB\]](#) entry is shown as Figure 6:

```

LocalKeyID    0x7f05
PeerKeyID     0x0012
KDFInputs     none
AlgID         AES-CMAC-128
Key           0x0123456789ABCDEF0123456789ABCDEF
Direction     both
NotBefore     12/31/2010
NotAfter      12/31/2011
Peers         yp.example.com
Protocol      TCP-A0

```

Figure 6. Key Table Entry on Xp

Peer Yp's [\[KEYTAB\]](#) entry is shown as Figure 6:

```

LocalKeyID    0x0107
PeerKeyID     0x0005
KDFInputs     none
AlgID         AES-CMAC-128
Key           0x0123456789ABCDEF0123456789ABCDEF

```


NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	xp.example.com
Protocol	TCP-A0

Figure 7. Key Table Entry on Yp

5.2 Protocol Operation: Xp Initiates a Connection

Peer Xp wishes to initiate a connection with Peer Yp.

- (1) Xp performs a key lookup for {Peer=Yp, Protocol=TCP-A0}, and the entry with LocalKeyID 0x7f05 is returned.
- (2) The LocalKeyID 0x7f05 is range mapped by Xp to the TCP-A0 identifier 0x05.
- (3) Xp performs the session key derivation using the mechanism specified for the TCP-A0 protocol in [[ao-crypto](#)].
- (4) Xp generates the AES-CMAC-128 MACs for the outgoing traffic using the derived key, and asserts the key identifier 0x05 in the packets.
- (5) Yp receives a protected packet from Xp, and extracts the key identifier 0x05.
- (6) Yp performs a key lookup for {Peer=Xp, Protocol=TCP-A0, PeerKeyID=0x05}, and the entry with LocalKeyID 0x0107 is returned.
- (7) Yp performs the session key derivation using the mechanism specified for the TCP-A0 protocol in [[ao-crypto](#)].
- (8) Yp verifies the MACs for the incoming traffic using the derived key.

5.3 Protocol Operation: Yp Initiates a Connection

Where Peer Yp establishes the connection, the same process is followed, except that range mapping process from step (2) is replaced by a table lookup.

[6](#) Security Considerations

<Security considerations text>

[6](#) IANA Considerations

This document requires no actions by IANA.

[7](#) References

[7.1](#) Normative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [KEYTAB] R. Housley and Polk, T. "Database of Long-Lived Cryptographic Keys", [draft-housley-saag-crypto-key-table-00.txt](#), September 2009.

[7.2](#) Informative References

- [tcipao] J. Touch, Mankin A., and Bonica R. "The TCP Authentication Option", [draft-ietf-tcpm-tcp-auth-opt-05.txt](#), July 2009.
- [ao-crypto] Lebovitz, G., "Cryptographic Algorithms, Use, & Implementation Requirements for TCP Authentication Option", [draft-lebovitz-ietf-tcpm-tcp-ao-crypto-02.txt](#), July 2009.

Author's Addresses

Tim Polk
National Institute of Standards and Technology
100 Bureau Drive, Mail Stop 8930
Gaithersburg, MD 20899-8930
USA
EMail: tim.polk@nist.gov

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive

Herndon, VA 20170
USA

Polk & Housley

Expires 19 April 2010

[Page 9]

INTERNET DRAFT

19 October 2009

EMail: housley@vigilsec.com

INTERNET DRAFT

19 October 2009

Full Copyright Statement

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

