INTERNET DRAFT
Intended Status: Informational

T. Polk
NIST
R. Housley
Vigil Security

Expires: January 29, 2011
July 28, 2010

Routing Authentication Using A Database of Long-Lived Cryptographic Keys

draft-polk-saag-rtg-auth-keytable-03.txt

Status of this Memo

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.

Table of Contents

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

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. (While this future environment conceivably includes automated key management protocols to negotiate short-lived cryptographic session keys, such keys are out of scope for this database.) 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.

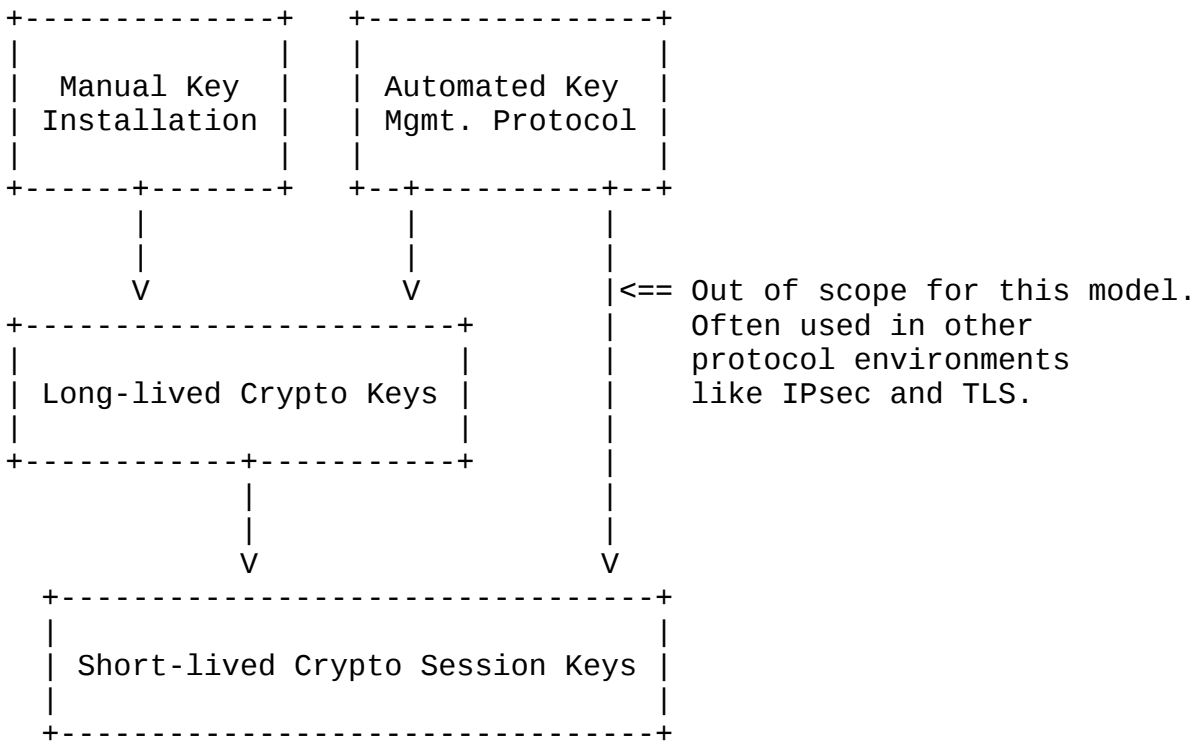

Figure 1. Cryptographic key establishment and use.

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.

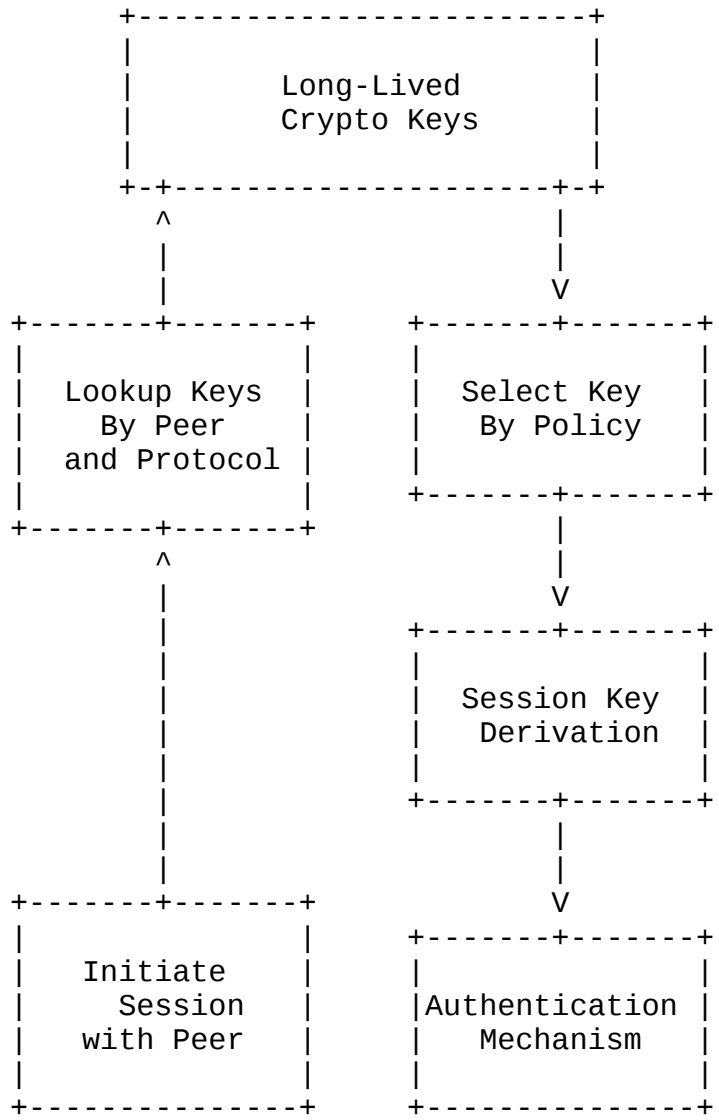

Figure 2. Session Initiation

Figure 3 illustrates how the long-lived cryptographic keys are accessed and employed when an entity receives a request establish a protected session with a peer. As step one in the session establishment 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.

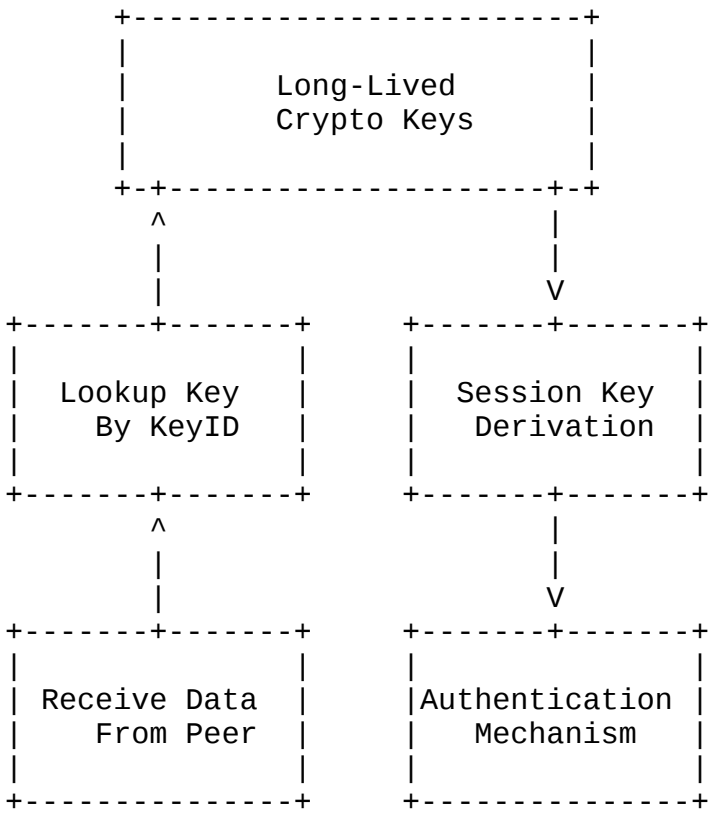

Figure 3. Session Acceptance

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 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 Database Maintenance

The previous sections focus upon installing and using the cryptographic keys in the database. A mechanism or mechanisms to remove unneeded keys is also needed to ensure that the key material up-to-date. [KEYTAB] provides mechanisms for expiration of entries; such key management could be performed in a fully automated fashion. Other reasons for key removal, such as severing a business relationship, or deciding a long lived key has been compromised before its expiration date, would inherently require a manual key removal process.

6 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].)

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

LocalKeyID	TCP-AO identifier
0x001a	0x01
0x004d	0x02
...	...
0x0107	0x12

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

After exchanging the TCP-AO 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-AO
```

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
Direction    both
NotBefore    12/31/2010
NotAfter     12/31/2011
Peers        xp.example.com
Protocol     TCP-AO
```

Figure 7. Key Table Entry on Yp

6.2 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-AO}, and the entry with LocalKeyID 0x7f05 is returned.
(2) The LocalKeyID 0x7f05 is range mapped by Xp to the TCP-AO identifier 0x05.
(3) Xp performs the session key derivation using the mechanism specified for the TCP-AO 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 a key lookup for {Peer=Xp, Protocol=TCP-AO, PeerKeyID=0x05}, and the entry with LocalKeyID 0x0107 is returned.
(7) Yp performs the session key derivation using the mechanism specified for the TCP-AO protocol in [ao-crypto].
(8) Yp verifies the MACs for the incoming traffic using the derived key.

6.3 Protocol Operation: Yp Initiates a Connection

Where Peer Yp establishes the connection, the same process is followed, except that the range mapping process from step (2) is

replaced by a table lookup.

7 Security Considerations

<Security considerations text>

8 IANA Considerations

This document requires no actions by IANA.

9 References

9.1 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-02.txt, September 2010.

9.2 Informative References

[tcpao] J. Touch, Mankin A., and Bonica R. "The TCP Authentication Option", draft-ietf-tcpm-tcp-auth-opt-08.txt, October 2009.

[ao-crypto] Lebovitz, G., "Cryptographic Algorithms, Use, & Implementation Requirments 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
EMail: housley@vigilsec.com

Full Copyright Statement