IETF Key Management Activities

Russ Housley

IETF Chair

Vigil Security LLC

May 2010

Internet Engineering Task Force

- “We make the net work”
- Open, consensus-based Internet standards
- The mission of the IETF is to produce high quality, relevant technical and engineering documents that influence the way people design, use, and manage the Internet in such a way as to make the Internet work better. These documents include protocol standards, best current practices, and informational documents of various kinds. [RFC 3935]

Key Management Activities

- You probably already know a lot about IETF security protocols, such as IKE, IKEv2, CMS, TLS, EAP-TLS, EAP-TTLS, ...
- Recent developments in KeyProv were covered by Hannes
- I will cover recent activities in these areas:
  - TLS Key Extraction
  - Trust Anchor Management
  - Keying and Authentication for Routing Protocols

Vigil Security LLC

TLS Key Extraction

Vigil Security LLC

Internet Engineering Task Force (IETF)
Request for Comments: 5705
Category: Standards Track
ISSN: 2070-1721

E. Rescorla
RTFM, Inc.

March 2010

Keying Material Exporters for Transport Layer Security (TLS)

Abstract

A number of protocols wish to leverage Transport Layer Security (TLS) to perform key establishment but then use some of the keying material for their own purposes. This document describes a general mechanism for allowing that.

Requirements

- Both client and server need to be able to export the same Exported Keying Material (EKM) value
- EKM values should be indistinguishable from random data to attackers who don't know the TLS/DTLS master_secret
- It should be possible to export multiple EKM values from the same TLS/DTLS association
- Knowing one EKM value should not reveal any useful information about the TLS/DTLS master_secret or about other EKM values

Vigil Security LLC

Application Context Binding

- An application using EKM must to securely establish the upper-layer context where the keying material will be used
- Many ways to do so, some suggestions include:
  - Information about the upper-layer context can be included in the optional data after the exporter label (more on next slide)
  - Information about the upper-layer context can be exchanged in TLS extensions included in the ClientHello and ServerHello messages
  - The upper-layer protocol can include its own handshake, which can be protected using the keys exported by TLS

Vigil Security LLC

EKM Computation Inputs

- Three inputs:
  - a disambiguating label string
  - an optional per-association context value
  - a length value

EKM Computation

- If no context is provided, EKM is computed:

  PRF(SecurityParameters.master_secret, **label**,
  SecurityParameters.client_random +
  SecurityParameters.server_random
)[**length**]

- If context is provided, EKM is computed:

  PRF(SecurityParameters.master_secret, **label**,
  SecurityParameters.client_random +
  SecurityParameters.server_random +
  context_value_length + **context_value**
)[**length**]

Vigil Security LLC

Trust Anchor Management

Vigil Security LLC

Trust Anchor Management Protocol (TAMP)

- Primary goal: reduce need for out-of-band trust decisions
- Enables trust anchor stores to be initialized in a secure environment and managed thereafter using the protocol
- Provides support for disaster recovery
- Management operations are subject to strict subordination rules

Vigil Security LLC

TAMP Message Types

- Eleven message types (CMS content types)
- Five pairs of request/response messages plus TAMPError

Request (TA Manager-generated)	Response (TA Store-generated)
TAMPUpdate	**TAMPUpdateConfirm**
TAMPApexUpdate	**TAMPApexUpdateConfirm**
TAMPCommunityUpdate	**TAMPCommunityUpdateConfirm**
SequenceNumberAdjust	**SequenceNumberAdjustConfirm**
TAMPStatusQuery	**TAMPStatusResponse**
	TAMPError

CMS Content Constraints

- Certificate or trust anchor extension that describes the types of content that can be validated using a given public key
  - Content is described in terms of CMS content types and CMS attributes
- Can be used as an authorization mechanism with TAMP

Using Trust Anchor Constraints During Certification Path Processing

- Describes how to use constraints expressed in a trust anchor during certification path processing
  - Essentially describes how to combine values from trust anchor extensions with standard user-supplied path validation inputs
- Processing can be either
  - Incorporated into an RFC 5280 compliant implementation
  - Implemented as pre-processing of RFC 5280 inputs and post-processing of RFC 5280 outputs

Trust Anchor Format

- Self-signed certificates are de facto standard format
  - Security requires out-of-band establishment of trust
  - Format does not lend itself to association of constraints by relying parties
- TAMP supports three trust anchor formats
  - Certificate
    - Self-signed or otherwise
  - TBSCertificate
    - Certificate structure without signature
  - TrustAnchorInfo
    - Can be as small as name and public key, but may also include constraints
    - Can encapsulate a certificate to add constraints

Apex Trust Anchor

- Represents ultimate authority over the Trust Anchor Store
- Apex Trust Anchor contains two public keys
  - Operational: used as any other Trust Anchor
  - Transitional: used only one time to replace the Apex Trust Anchor
    - One time use for each Trust Anchor Store
    - Stored encrypted

Putting the pieces together …

- Trust Anchor format
  - Defines compact representation of trust anchor information (TrustAnchorInfo)
  - Enables relying parties to customize trust anchor information (TBSCertificate and TrustAnchorInfo)
  - Provides a migration path by supporting current trust anchor format (Certificate)
- Trust Anchor-based constraints are consistently enforced
- TAMP provides means for securing trust anchor representations and managing trust anchor store contents
- CMS content constraints provide an authorization mechanism

Keying and Authentication for Routing Protocols (KARP)

Vigil Security LLC

KARP

- Defines a conceptual model for a key table
- Describes the model's application to routing protocols

- Manual key management is today's reality in routing protocols, but want to enable future key establishment protocols in the that co-exist with manual keying
- Key establishment will employ separate protocols, not a handshake within routing protocols
- Accommodate textual description of database entries

Crypto Key Table

- Database is characterized as a table, with a row for each key
- Identifies 11 columns for the key and its attributes
- Describes rollover between long-lived keys

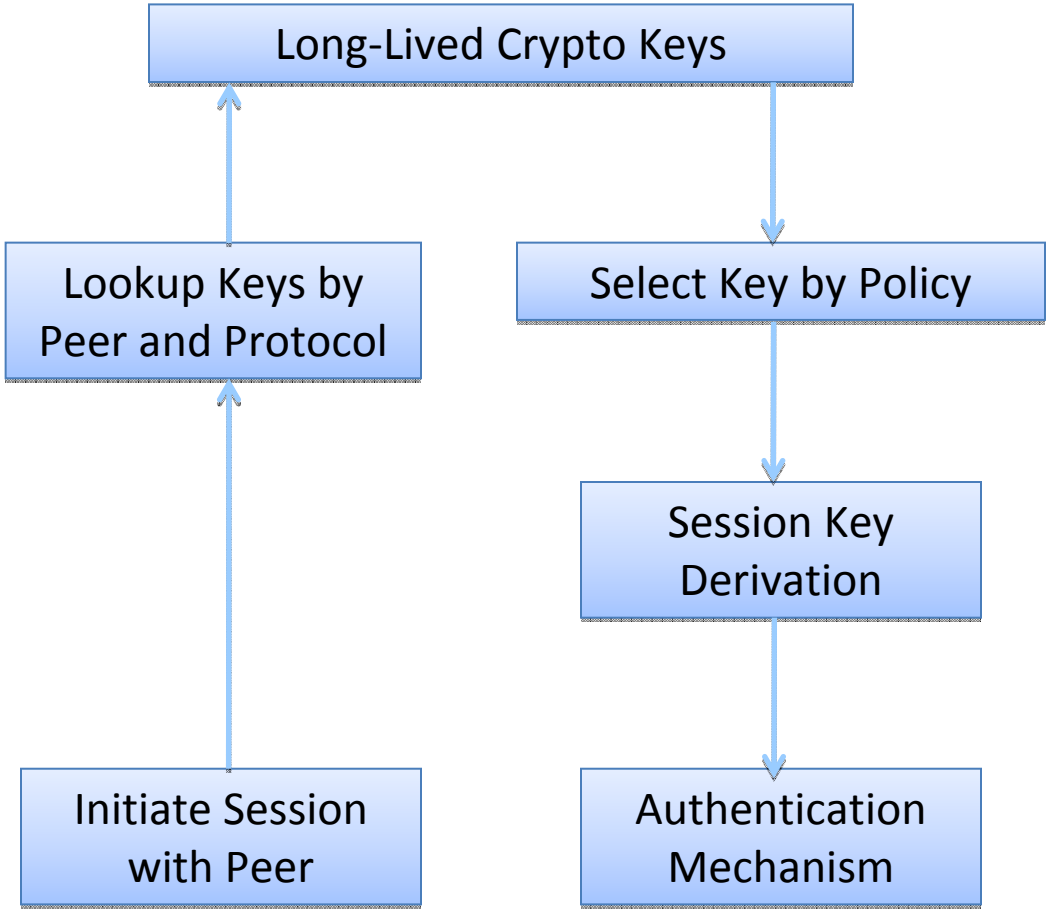
Crypto Key Table Columns (1 of 2)

- LocalKeyID
  - A 16-bit integer in hexadecimal, unique in the context of the database. The high order bit differentiates pairwise and group keys.
- PeerKeyID
  - For pairwise keys, the PeerKeyID field is a 16 bit integer in hexadecimal provided by the peer or "unknown" if the peer has not yet provided this value.
  - For group keying, the PeerKeyID field is set to "group", which easily accommodates group keys generated by a third party.
- KDF
  - Indicates which key derivation function (KDF) is used to generate short-lived keys (or "none“ when the long-term key is used directly).
- KDFInputs
  - Used when supplemental public or private data is supplied to the KDF.
- AlgID
  - Indicates which cryptographic algorithm to be used with the security protocol.

Crypto Key Table Columns (2 of 2)

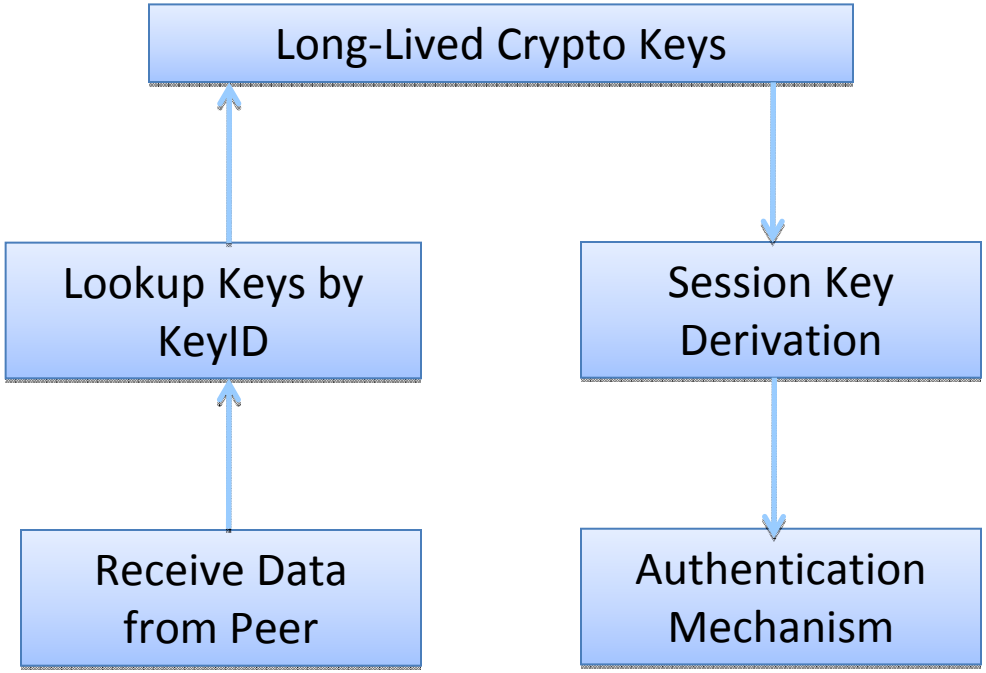
- Key
  - A hexadecimal string representing a ling-lived symmetric cryptographic key.
- KeyDirection
  - Indicates whether this key may be used for inbound traffic, outbound traffic, or both.
- NotBefore
  - Specifies the earliest date and time at which this key should be considered for use.
- NotAfter
  - Specifies the latest date and time at which this key should be considered for use.
- Peers
  - Identifies a peer system or set of peer systems
- Protocol
  - Identifies the security protocol where this key is to be used to provide cryptographic protection.

Initiator's View



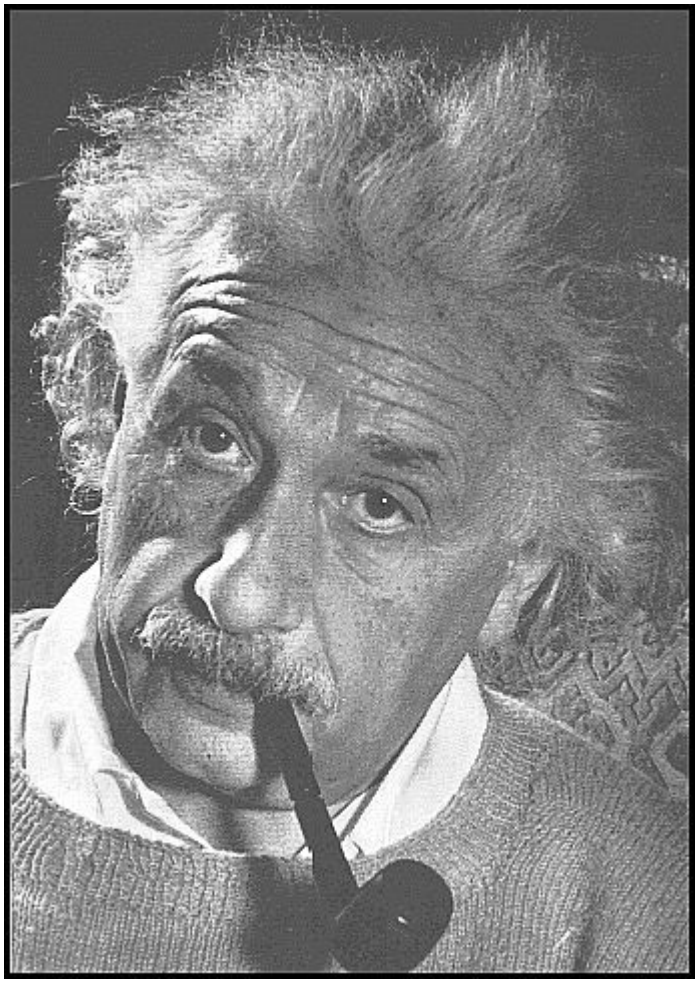
Vigil Security LLC

Receiver's View



Questions?

"The problems that exist in the world today cannot be solved by the level of thinking that created them."



Vigil Security LLC