                     IKEv2-Derived Shared Secret Key for
          the One-Way Active Measurement Protocol (OWAMP) and
             Two-Way Active Measurement Protocol (TWAMP)

Abstract

   The One-Way Active Measurement Protocol (OWAMP) and Two-Way Active
   Measurement Protocol (TWAMP) security mechanisms require that both
   the client and server endpoints possess a shared secret. This
   document describes the use of keys derived from an IKEv2 security
   association (SA) as the shared key in OWAMP or TWAMP. If the shared
   key can be derived from the IKEv2 SA, OWAMP or TWAMP can support
   certificate-based key exchange; this would allow for more operational
   flexibility and efficiency. The key derivation presented in this
   document can also facilitate automatic key management.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7717.

Copyright Notice

Table of Contents

1. Introduction

   The One-Way Active Measurement Protocol (OWAMP) [RFC4656] and the
   Two-Way Active Measurement Protocol (TWAMP) [RFC5357] can be used to
   measure network performance parameters such as latency, bandwidth,
   and packet loss by sending probe packets and monitoring their
   experience in the network. In order to guarantee the accuracy of
   network measurement results, security aspects must be considered.
   Otherwise, attacks may occur and the authenticity of the measurement
   results may be violated. For example, if no protection is provided,
   an adversary in the middle may modify packet timestamps, thus
   altering the measurement results.

   According to [RFC4656] and [RFC5357], the OWAMP and TWAMP (O/TWAMP)
   security mechanisms require that endpoints (i.e., both the client and
   the server) possess a shared secret. In today's network deployments,
   however, the use of pre-shared keys is far from optimal. For
   example, in wireless infrastructure networks, certain network
   elements (which can be seen as the two endpoints from an O/TWAMP
   perspective) support certificate-based security. For instance,
   consider the case in which one wants to measure IP performance
   between an E-UTRAN Evolved Node B (eNB) and Security Gateway (SeGW),
   both of which are 3GPP Long Term Evolution (LTE) nodes and support
   certificate mode and the Internet Key Exchange Protocol version 2
   (IKEv2).

   The O/TWAMP security mechanism specified in [RFC4656] and [RFC5357]
   supports the pre-shared key (PSK) mode only, hindering large-scale
   deployment of O/TWAMP: provisioning and management of "shared
   secrets" for massive deployments consumes a tremendous amount of
   effort and is prone to human error. At the same time, recent trends
   point to wider IKEv2 deployment that, in turn, calls for mechanisms
   and methods that enable tunnel end-users, as well as operators, to
   measure one-way and two-way network performance in a standardized
   manner.

   With IKEv2 widely deployed, employing shared keys derived from an
   IKEv2 security association (SA) can be considered a viable
   alternative through the method described in this document. If the
   shared key can be derived from the IKEv2 SA, O/TWAMP can support
   certificate-based key exchange and practically increase operational
   flexibility and efficiency. The use of IKEv2 also makes it easier to
   extend automatic key management.

   In general, O/TWAMP measurement packets can be transmitted inside the
   IPsec tunnel, as typical user traffic is, or transmitted outside the
   IPsec tunnel. This may depend on the operator's policy and the
   performance evaluation goal, and it is orthogonal to the mechanism

described in this document. When IPsec is deployed, protecting
O/TWAMP traffic in unauthenticated mode using IPsec is one option.
Another option is to protect O/TWAMP traffic using the O/TWAMP
security established using the PSK derived from IKEv2 and bypassing
the IPsec tunnel.

Protecting unauthenticated O/TWAMP control and/or test traffic via
the Authentication Header (AH) [RFC4302] or Encapsulating Security
Payload (ESP) [RFC4303] cannot provide various security options,
e.g., it cannot authenticate part of an O/TWAMP packet as mentioned
in [RFC4656]. For measuring latency, a timestamp is carried in O/
TWAMP test traffic. The sender has to fetch the timestamp, encrypt
it, and send it. When the mechanism described in this document is
used, partial authentication of O/TWAMP packets is possible and
therefore the middle step can be skipped, potentially improving
accuracy as the sequence number can be encrypted and authenticated
before the timestamp is fetched. The receiver obtains the timestamp
without the need for the corresponding decryption step. In such
cases, protecting O/TWAMP traffic using O/TWAMP security but
bypassing the IPsec tunnel has its advantages.

This document specifies a method for enabling network measurements
between a TWAMP client and a TWAMP server. In short, the shared key
used for securing TWAMP traffic is derived from IKEv2 [RFC7296].
TWAMP implementations signal the use of this method by setting
IKEv2Derived (see Section 7). IKEv2-derived keys SHOULD be used
instead of shared secrets when O/TWAMP is employed in a deployment
using IKEv2. From an operations and management perspective
[RFC5706], the mechanism described in this document requires that
both the TWAMP Control-Client and Server support IPsec.

The remainder of this document is organized as follows. Section 4
summarizes O/TWAMP protocol operation with respect to security.
Section 5 presents the method for binding TWAMP and IKEv2 for network
measurements between the client and the server that both support
IKEv2. Finally, Section 6 discusses the security considerations
arising from the proposed mechanisms.

2. 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 [RFC2119].

3. Scope

   This document specifies a method using keys derived from an IKEv2 SA
   as the shared key in O/TWAMP. O/TWAMP implementations signal the use
   of this method by setting IKEv2Derived (see Section 7).

4. O/TWAMP Security

   Security for O/TWAMP-Control and O/TWAMP-Test are briefly reviewed in
   the following subsections.

4.1. O/TWAMP-Control Security

   O/TWAMP uses a simple cryptographic protocol that relies on

   o AES-CBC for confidentiality

   o HMAC-SHA1 truncated to 128 bits for message authentication

   Three modes of operation are supported in the OWAMP-Control protocol:
   unauthenticated, authenticated, and encrypted. In addition to these
   modes, the TWAMP-Control protocol also supports a mixed mode, i.e.,
   the TWAMP-Control protocol operates in encrypted mode while TWAMP-
   Test protocol operates in unauthenticated mode. The authenticated,
   encrypted, and mixed modes require that endpoints possess a shared
   secret, typically a passphrase. The secret key is derived from the
   passphrase using a password-based key derivation function PBKDF2
   (PKCS #5) [RFC2898].

   In the unauthenticated mode, the security parameters are left unused.
   In the authenticated, encrypted, and mixed modes, the security
   parameters are negotiated during the control connection
   establishment.

   Figure 1 illustrates the initiation stage of the O/TWAMP-Control
   protocol between a Control-Client and a Server. In short, the
   Control-Client opens a TCP connection to the Server in order to be
   able to send O/TWAMP-Control commands. The Server responds with a
   Server Greeting, which contains the Modes, Challenge, Salt, Count,
   and MBZ ("MUST be zero") fields (see Section 3.1 of [RFC4656]). If
   the Control-Client preferred mode is available, the client responds
   with a Set-Up-Response message, wherein the selected Mode, as well as
   the KeyID, Token, and Client initialization vector (IV) are included.
   The Token is the concatenation of a 16-octet Challenge, a 16-octet
   AES Session-key used for encryption, and a 32-octet HMAC-SHA1
   Session-key used for authentication. The Token is encrypted using
   AES-CBC.

```
    +----------------+                      +--------+
    | Control-Client |                      | Server |
    +----------------+                      +--------+
            |                                   |
            |<------ TCP Connection-- ----->|
            |                                   |
            |<------ Greeting message ------|
            |                                   |
            |------- Set-Up-Response ------>|
            |                                   |
            |<------ Server-Start ----------|
            |                                   |
```
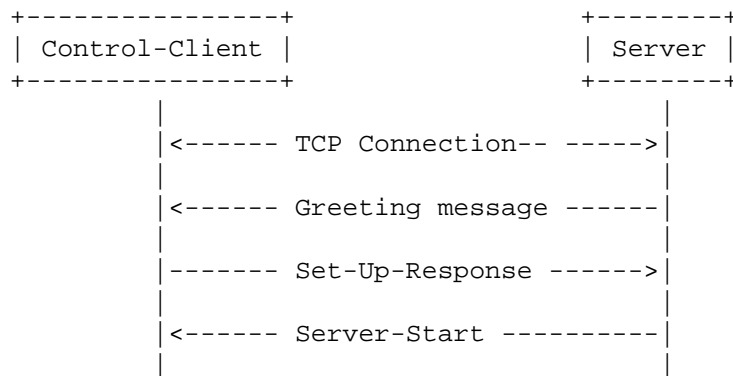
                Figure 1: Initiation of O/TWAMP-Control

   Encryption uses a key derived from the shared secret associated with
   KeyID. In the authenticated, encrypted, and mixed modes, all further
   communication is encrypted using the AES Session-key and
   authenticated with the HMAC Session-key. After receiving the Set-Up-
   Response, the Server responds with a Server-Start message containing
   the Server-IV. The Control-Client encrypts everything it transmits
   through the just established O/TWAMP-Control connection using stream
   encryption with Client-IV as the IV. Correspondingly, the Server
   encrypts its side of the connection using Server-IV as the IV. The
   IVs themselves are transmitted in cleartext. Encryption starts with
   the block immediately following that containing the IV.

   The AES Session-key and HMAC Session-key are generated randomly by
   the Control-Client. The HMAC Session-key is communicated along with
   the AES Session-key during O/TWAMP-Control connection setup. The
   HMAC Session-key is derived independently of the AES Session-key.

4.2. O/TWAMP-Test Security

   The O/TWAMP-Test protocol runs over UDP, using the Session-Sender and
   Session-Reflector IP and port numbers that were negotiated during the
   Request-Session exchange. O/TWAMP-Test has the same mode with O/
   TWAMP-Control and all O/TWAMP-Test sessions inherit the corresponding
   O/TWAMP-Control session mode except when operating in mixed mode.

   The O/TWAMP-Test packet format is the same in authenticated and
   encrypted modes. The encryption and authentication operations are,
   however, different. Similarly, with the respective O/TWAMP-Control
   session, each O/TWAMP-Test session has two keys: an AES Session-key
   and an HMAC Session-key. However, there is a difference in how the
   keys are obtained:

   O/TWAMP-Control: the keys are generated by the Control-Client and
           communicated to the Server during the control connection
           establishment with the Set-Up-Response message (as part of
           the Token).

   O/TWAMP-Test: the keys are derived from the O/TWAMP-Control keys and
           the session identifier (SID), which serve as inputs to the
           key derivation function (KDF). The O/TWAMP-Test AES Session-
           key is generated using the O/TWAMP-Control AES Session-key,
           with the 16-octet SID, for encrypting and decrypting the
           packets of the particular O/TWAMP-Test session. The O/TWAMP-
           Test HMAC Session-key is generated using the O/TWAMP-Control
           HMAC Session-key, with the 16-octet SID, for authenticating
           the packets of the particular O/TWAMP-Test session.

4.3. O/TWAMP Security Root

   As discussed above, the O/TWAMP-Test AES Session-key and HMAC
   Session-key are derived, respectively, from the O/TWAMP-Control AES
   Session-key and HMAC Session-key. The AES Session-key and HMAC
   Session-key used in the O/TWAMP-Control protocol are generated
   randomly by the Control-Client, and encrypted with the shared secret
   associated with KeyID. Therefore, the security root is the shared
   secret key. Thus, for large deployments, key provision and
   management may become overly complicated. Comparatively, a
   certificate-based approach using IKEv2 can automatically manage the
   security root and solve this problem, as we explain in Section 5.

5. O/TWAMP for IPsec Networks

   This section presents a method of binding O/TWAMP and IKEv2 for
   network measurements between a client and a server that both support
   IPsec. In short, the shared key used for securing O/TWAMP traffic is
   derived using IKEv2 [RFC7296].

5.1. Shared Key Derivation

   In the authenticated, encrypted, and mixed modes, the shared secret
   key MUST be derived from the IKEv2 SA. Note that we explicitly opt
   to derive the shared secret key from the IKEv2 SA, rather than the
   child SA, since it is possible that an IKEv2 SA is created without
   generating any child SA [RFC6023].

   When the shared secret key is derived from the IKEv2 SA, SK_d must be
   generated first. SK_d must be computed as per [RFC7296].

The shared secret key MUST be generated as follows:

    Shared secret key = prf(SK_d, "IPPM")

Wherein the string "IPPM" is encoded in ASCII and "prf" is a
pseudorandom function.

It is recommended that the shared secret key is derived in the IPsec
layer so that IPsec keying material is not exposed to the O/TWAMP
client. Note, however, that the interaction between the O/TWAMP and
IPsec layers is host internal and implementation specific.
Therefore, this is clearly outside the scope of this document, which
focuses on the interaction between the O/TWAMP client and server.
That said, one possible way could be the following: at the Control-
Client side, the IPsec layer can perform a lookup in the Security
Association Database (SAD) using the IP address of the Server and
thus match the corresponding IKEv2 SA. At the Server side, the IPsec
layer can look up the corresponding IKEv2 SA by using the Security
Parameter Indexes (SPIs) sent by the Control-Client (see
Section 5.3), and therefore extract the shared secret key.

If both the client and server do support IKEv2 but there is no
current IKEv2 SA, two alternative ways could be considered. First,
the O/TWAMP Control-Client initiates the establishment of the IKEv2
SA, logs this operation, and selects the mode that supports IKEv2.
Alternatively, the O/TWAMP Control-Client does not initiate the
establishment of the IKEv2 SA, logs an error for operational
management purposes, and proceeds with the modes defined in
[RFC4656], [RFC5357], and [RFC5618]. Again, although both
alternatives are feasible, they are in fact implementation specific.

If rekeying for the IKEv2 SA or deletion of the IKEv2 SA occurs, the
corresponding shared secret key generated from the SA MUST continue
to be used until the O/TWAMP session terminates.

5.2. Server Greeting Message Update

To trigger a binding association between the key generated from IKEv2
and the O/TWAMP shared secret key, the Modes field in the Server
Greeting Message (Figure 2) must support key derivation as discussed
in Section 5.1. Support for deriving the shared key from the IKEv2
SA is indicated by setting IKEv2Derived (see Section 7). Therefore,
when this method is used, the Modes value extension MUST be
supported.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Unused (12 octets)                        |
|                                                               |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Modes                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Challenge (16 octets)                      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Salt (16 octets)                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Count (4 octets)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       MBZ (12 octets)                         |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
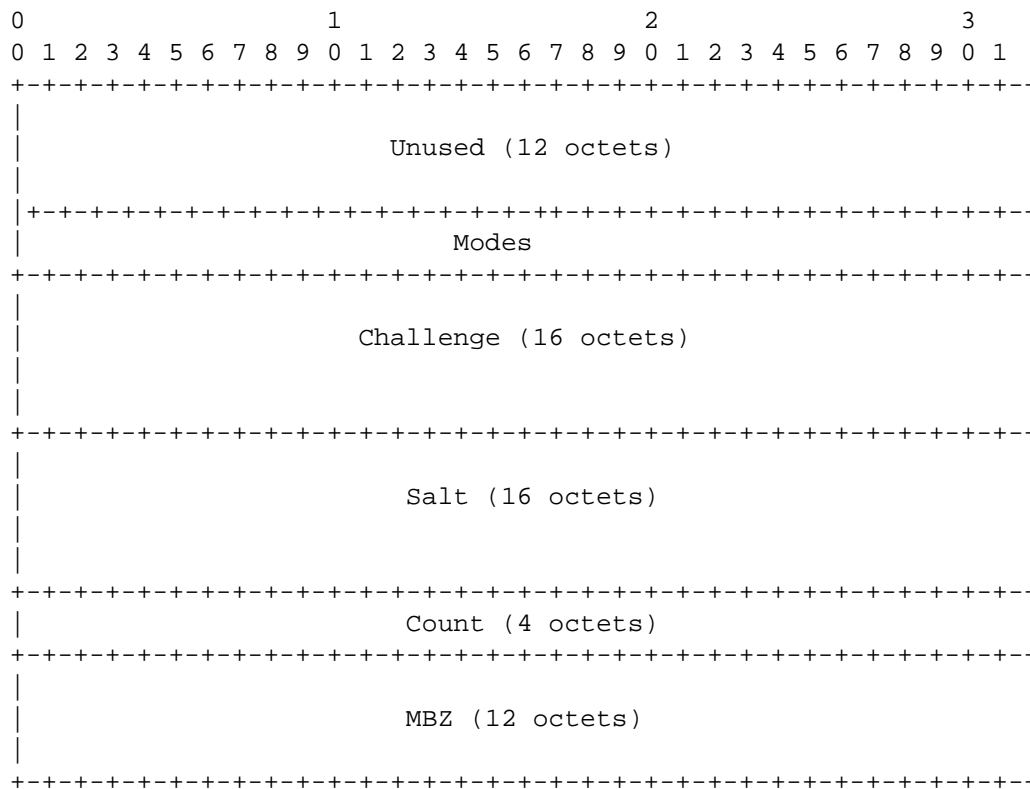
                   Figure 2: Server Greeting Format

   The choice of this set of Modes values poses no backwards-
   compatibility problems to existing O/TWAMP clients. Robust legacy
   Control-Client implementations would disregard the fact that the
   IKEv2Derived Modes bit in the Server Greeting is set. On the other
   hand, a Control-Client implementing this method can identify that the
   O/TWAMP Server contacted does not support this specification. If the
   Server supports other Modes, as one could assume, the Control-Client
   would then decide which Mode to use and indicate such accordingly as
   per [RFC4656] and [RFC5357]. A Control-Client that is implementing
   this method and decides not to employ IKEv2 derivation can simply
   behave as a client that is purely compatible with [RFC4656] and
   [RFC5357].

5.3. Set-Up-Response Update

   The Set-Up-Response message Figure 3 is updated as follows. When an
   O/TWAMP Control-Client implementing this method receives a Server
   Greeting indicating support for Mode IKEv2Derived, it SHOULD reply to
   the O/TWAMP Server with a Set-Up-Response that indicates so. For

example, a compatible O/TWAMP Control-Client choosing the
authenticated mode with IKEv2 shared secret key derivation should set
the Mode bits as per Section 7.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Mode                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      KeyID (80 octets)                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Token (16 octets)                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Client-IV (12 octets)                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
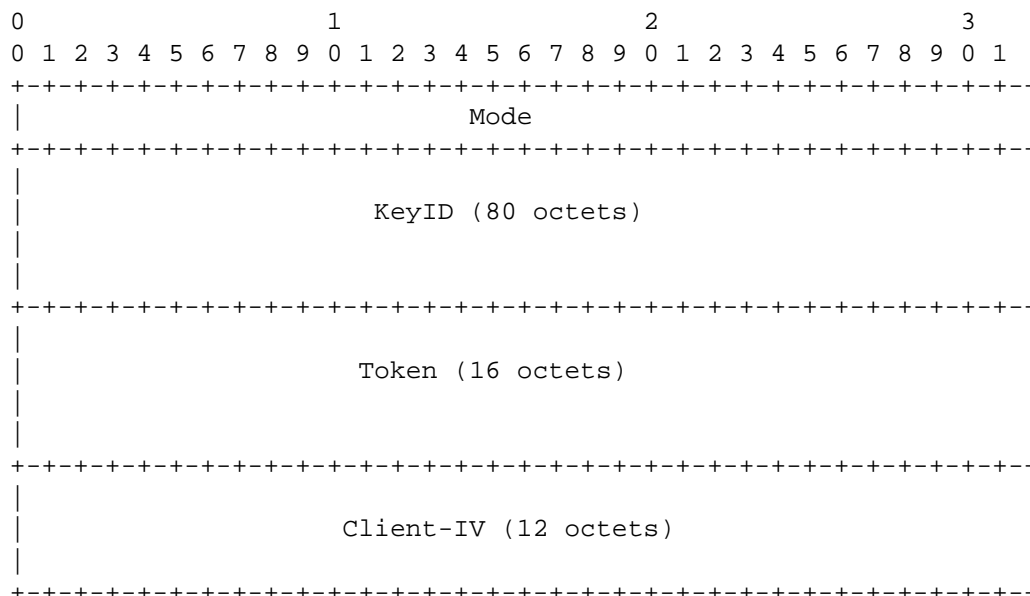
                    Figure 3: Set-Up-Response Message

The Security Parameter Index (SPI), as described in [RFC4301] and
[RFC7296], uniquely identifies the SA. If the Control-Client
supports shared secret key derivation for the IKEv2 SA, it will
choose the corresponding Mode value and carry SPIi and SPIr in the
KeyID field. SPIi and SPIr MUST be included in the KeyID field of
the Set-Up-Response Message to indicate the IKEv2 SA from which the
O/TWAMP shared secret key was derived. The length of SPI is 8
octets. Therefore, the first 8 octets of the KeyID field MUST carry
SPIi, and the second 8 octets MUST carry SPIr. The remaining bits of
the KeyID field MUST be set to zero.

An O/TWAMP Server implementation MUST obtain the SPIi and SPIr from
the first 16 octets and ignore the remaining octets of the KeyID
field. Then, the Control-Client and the Server can derive the shared
secret key based on the Mode value and SPI. If the O/TWAMP Server
cannot find the IKEv2 SA corresponding to the SPIi and SPIr received,
it MUST log the event for operational management purposes. In
addition, the O/TWAMP Server SHOULD set the Accept field of the
Server-Start message to the value 6 to indicate that the Server is
not willing to conduct further transactions in this O/TWAMP-Control
session since it cannot find the corresponding IKEv2 SA.

5.4. O/TWAMP over an IPsec Tunnel

   The IPsec Authentication Header (AH) [RFC4302] and Encapsulating
   Security Payload (ESP) [RFC4303] provide confidentiality and data
   integrity to IP datagrams. An IPsec tunnel can be used to provide
   the protection needed for O/TWAMP Control and Test packets, even if
   the peers choose the unauthenticated mode of operation. In order to
   ensure authenticity and security, O/TWAMP packets between two IKEv2
   systems SHOULD be configured to use the corresponding IPsec tunnel
   running over an external network even when using the O/TWAMP
   unauthenticated mode.

6. Security Considerations

   As the shared secret key is derived from the IKEv2 SA, the key
   derivation algorithm strength and limitations are as per [RFC7296].
   The strength of a key derived from a Diffie-Hellman exchange using
   any of the groups defined here depends on the inherent strength of
   the group, the size of the exponent used, and the entropy provided by
   the random number generator employed. The strength of all keys and
   implementation vulnerabilities, particularly denial-of-service (DoS)
   attacks are as defined in [RFC7296].

7. IANA Considerations

   During the production of this document, the authors and reviewers
   noticed that the TWAMP-Modes registry should describe a field of
   single bit position flags, rather than the existing registry
   construction with assignment of integer values. In addition, the
   Semantics Definition column seemed to have spurious information in
   it. The registry has been reformatted to simplify future
   assignments. Thus, the contents of the TWAMP-Modes registry are as
   follows:

```
   Bit|Description                                |Semantics  |Reference
   Pos|                                           |Definition |
   ---|-------------------------------------------|-----------|---------
   0   Unauthenticated                             Section 3.1 [RFC4656]
   1   Authenticated                               Section 3.1 [RFC4656]
   2   Encrypted                                   Section 3.1 [RFC4656]
   3   Unauth. TEST protocol, Encrypted CONTROL    Section 3.1 [RFC5618]
   4   Individual Session Control                              [RFC5938]
   5   Reflect Octets Capability                               [RFC6038]
   6   Symmetrical Size Sender Test Packet Format              [RFC6038]
```

                      Figure 4: TWAMP Modes

The new description and registry management instructions follow.

Registry Specification: TWAMP-Modes are specified in TWAMP Server
Greeting messages and Set-Up-Response messages consistent with
Section 3.1 of [RFC5357]. Modes are indicated by setting single bits
in the 32-bit Modes field.

Registry Management: Because the "TWAMP-Modes" are based on only 32
bit positions with each position conveying a unique feature, and
because TWAMP is an IETF protocol, this registry must be updated only
by "IETF Review" as specified in [RFC5226]. IANA SHOULD allocate
monotonically increasing bit positions when requested.

Experimental Numbers: No experimental bit positions are currently
assigned in the Modes registry, as indicated in the initial contents
above.

In addition, per this document, a new entry has been added to the
TWAMP-Modes registry:

```
Bit|Description                                   |Semantics  |Reference
Pos|                                              |Definition |
---|----------------------------------------------|-----------|---------
7   IKEv2Derived Mode Capability                   Section 5    RFC 7717
```

                Figure 5: TWAMP IKEv2-Derived Mode Capability

For the new OWAMP-Modes registry, see the IANA Considerations in
[RFC7718].

8. References

8.1. Normative References

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
               Requirement Levels", BCP 14, RFC 2119,
               DOI 10.17487/RFC2119, March 1997,
               <http://www.rfc-editor.org/info/rfc2119>.

   [RFC4656]   Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.
               Zekauskas, "A One-way Active Measurement Protocol
               (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006,
               <http://www.rfc-editor.org/info/rfc4656>.

   [RFC5226]   Narten, T. and H. Alvestrand, "Guidelines for Writing an
               IANA Considerations Section in RFCs", BCP 26, RFC 5226,
               DOI 10.17487/RFC5226, May 2008,
               <http://www.rfc-editor.org/info/rfc5226>.

   [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J.
              Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",
              RFC 5357, DOI 10.17487/RFC5357, October 2008,
              <http://www.rfc-editor.org/info/rfc5357>.

   [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the
              Two-Way Active Measurement Protocol (TWAMP)", RFC 5618,
              DOI 10.17487/RFC5618, August 2009,
              <http://www.rfc-editor.org/info/rfc5618>.

   [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T.
              Kivinen, "Internet Key Exchange Protocol Version 2
              (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October
              2014, <http://www.rfc-editor.org/info/rfc7296>.

   [RFC7718] Morton, A., "Registries for the One-Way Active Measurement
              Protocol (OWAMP)", RFC 7718, DOI 10.17487/RFC7718,
              December 2015, <http://www.rfc-editor.org/info/rfc7718>.

8.2. Informative References

   [RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography
              Specification Version 2.0", RFC 2898,
              DOI 10.17487/RFC2898, September 2000,
              <http://www.rfc-editor.org/info/rfc2898>.

   [RFC4301] Kent, S. and K. Seo, "Security Architecture for the
              Internet Protocol", RFC 4301, DOI 10.17487/RFC4301,
              December 2005, <http://www.rfc-editor.org/info/rfc4301>.

   [RFC4302] Kent, S., "IP Authentication Header", RFC 4302,
              DOI 10.17487/RFC4302, December 2005,
              <http://www.rfc-editor.org/info/rfc4302>.

   [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)",
              RFC 4303, DOI 10.17487/RFC4303, December 2005,
              <http://www.rfc-editor.org/info/rfc4303>.

   [RFC5706] Harrington, D., "Guidelines for Considering Operations and
              Management of New Protocols and Protocol Extensions",
              RFC 5706, DOI 10.17487/RFC5706, November 2009,
              <http://www.rfc-editor.org/info/rfc5706>.

   [RFC5938] Morton, A. and M. Chiba, "Individual Session Control
              Feature for the Two-Way Active Measurement Protocol
              (TWAMP)", RFC 5938, DOI 10.17487/RFC5938, August 2010,
              <http://www.rfc-editor.org/info/rfc5938>.

   [RFC6023] Nir, Y., Tschofenig, H., Deng, H., and R. Singh, "A
              Childless Initiation of the Internet Key Exchange Version
              2 (IKEv2) Security Association (SA)", RFC 6023,
              DOI 10.17487/RFC6023, October 2010,
              <http://www.rfc-editor.org/info/rfc6023>.

   [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement
              Protocol (TWAMP) Reflect Octets and Symmetrical Size
              Features", RFC 6038, DOI 10.17487/RFC6038, October 2010,
              <http://www.rfc-editor.org/info/rfc6038>.

Acknowledgements

   We thank Eric Chen, Yaakov Stein, Brian Trammell, Emily Bi, John
   Mattsson, Steve Baillargeon, Spencer Dawkins, Tero Kivinen, Fred
   Baker, Meral Shirazipour, Hannes Tschofenig, Ben Campbell, Stephen
   Farrell, Brian Haberman, and Barry Leiba for their reviews, comments,
   and text suggestions.

   Al Morton deserves a special mention for his thorough reviews and
   text contributions to this document as well as the constructive
   discussions over several IPPM meetings.

Authors' Addresses

   Kostas Pentikousis (editor)
   EICT GmbH
   EUREF-Campus Haus 13
   Torgauer Strasse 12-15
   10829 Berlin
   Germany

   Email: k.pentikousis@eict.de

   Emma Zhang
   Huawei Technologies
   Huawei Building, No.3, Rd. XinXi
   Haidian District, Beijing 100095
   China

   Email: emma.zhanglijia@huawei.com

   Yang Cui
   Huawei Technologies
   Otemachi First Square 1-5-1 Otemachi
   Chiyoda-ku, Tokyo 100-0004
   Japan

   Email: cuiyang@huawei.com