IPPM WG K. Pentikousis, Ed.
Internet-Draft EICT
Intended status: Standards Track Y. Cui
Expires: April 24, 2014 E. Zhang
Huawei Technologies
October 21, 2013

Network Performance Measurement for IPsec
draft-ietf-ippm-ipsec-01

Abstract

IPsec is a mature technology with several interoperable implementations. Indeed, the use of IPsec tunnels is increasingly gaining popularity in several deployment scenarios, not the least in what used to be solely areas of traditional telecommunication protocols. Wider IPsec deployment 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. Unfortunately, however, standard IP performance measurement security mechanisms cannot be readily used with IPsec. This document makes the case for employing IPsec to protect the One-way and Two-Way Active Measurement Protocols (O/TWAMP) and proposes a method which combines IKEv2 and O/TWAMP as defined in RFC 4656 and RFC 5357, respectively. This specification aims, on the one hand, to ensure that O/TWAMP can be secured with the best mechanisms we have at our disposal today while, on the other hand, it facilitates the applicability of O/TWAMP to networks that have already deployed IPsec.

Status of This Memo

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.

Cryptographic security mechanisms, such as IPsec, have been considered during the early stage of the specification of the two active measurement protocols mentioned above. However, due to several reasons, it was decided to avoid tying the development and deployment of O/TWAMP to such security mechanisms. In practice, for many networks, the issues listed in [RFC4656], Sec. 6.6 with respect to IPsec are still valid. However, we expect that in the near future IPsec will be deployed in many more hosts and networks than today. For example, IPsec tunnels may be used to secure wireless channels. In this case, what we are interested in is measuring network performance specifically for the traffic carried by the secured tunnel, not over the wireless channel in general. This document makes the case that O/TWAMP should be cognizant when IPsec and other security mechanisms are in place and can be leveraged upon. In other words, it is now time to specify how O/TWAMP is used in a network environment where IPsec is already deployed. We expect that in such an environment, measuring IP performance over IPsec tunnels with O/TWAMP is an important tool for operators.

For example, when considering the use of O/TWAMP in networks with IPsec deployed, we can take advantage of the IPsec key exchange protocol [RFC5996]. In particular, we note that it is not necessary to use distinct keys in OWAMP-Control and OWAMP-Test layers. One key for encryption and another for authentication is sufficient for both Control and Test layers. This obviates the need to generate two keys for each layer and reduces the complexity of O/TWAMP protocols in an IPsec environment. This observation comes from the fact that separate session keys in the OWAMP-Control and OWAMP-Test layers were designed for preventing reflection attacks when employing the current mechanism. Once IPsec is employed, such a potential threat is alleviated.

The remainder of this document is organized as follows. Section 3 motivates this work by revisiting the arguments made in [RFC4656] against the use of IPsec; this section also summarizes protocol operation with respect to security. Section 4 presents a method of binding O/TWAMP and IKEv2 for network measurements between a sender and a receiver which both support IPsec. Finally, Section 5 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. Motivation

In order to motivate the solutions proposed in this document, let us first revisit Section 6.6 of [RFC4656]. As we explain below, the reasons originally listed therein may not apply in many cases today.

RFC 4656 opts against using IPsec and instead favors the use of "a simple cryptographic protocol (based on a block cipher in CBC mode)". The first argument justifying this decision in [RFC4656] is that partial authentication in OWAMP authentication mode is not possible with IPsec. IPsec indeed cannot authenticate only a part of a packet. However, in an environment where IPsec is already deployed and actively used, partial authentication for OWAMP contradicts the operational reasons dictating the use of IPsec. It also increases the operational complexity of OWAMP (and TWAMP) in networks where IPsec is actively used and may in practice limit its applicability.

The second argument made is the need to keep separate deployment paths between OWAMP and IPsec. In several currently deployed types of networks IPsec is widely used to protect the data and signaling planes. For example, in mobile telecommunication networks, the deployment rate of IPsec exceeds 95% with respect to the LTE serving network. In older technology cellular networks, such as UMTS and GSM, IPsec use penetration is lower, but still quite significant. Additionally, there is a great number of IPsec-based VPN applications which are widely used in business applications to provide end-to-end security over, for instance, publicly open or otherwise untrusted IEEE 802.11 wireless LANs. At the same time, many IETF-standardized protocols make use of IPsec/IKE, including MIPv4/v6, HIP, SCTP, BGP, NAT and SIP, just to name a few.

The third argument in [RFC4656] is that, effectively, the adoption of IPsec in OWAMP may be problematic for "lightweight embedded devices." However, since the publication of RFC 4656, a large number of limited-resource and low-cost hardware, such as Ethernet switches, DSL modems, set-top boxes and other such devices come with support for IPsec "out of the box". Therefore concerns about implementation, although likely valid a decade ago, are not well founded today.

Finally, everyday use of IPsec applications by field technicians and good understanding of the IPsec API by many programmers should no longer be a reason for concern. On the contrary: By now, IPsec open source code is available for anyone who wants to use it. Therefore, although IPsec does need a certain level of expertise to deal with

it, in practice, most competent technical personnel and programmers have no problems using it on a daily basis.

OWAMP and TWAMP actually consist of two inter-related protocols: O/TWAMP-Control and O/TWAMP-Test. With respect to TWAMP, since "TWAMP and OWAMP use the same protocol for establishment of Control and Test procedures" [RFC5357] (Section 6), IPsec is also not considered. O/TWAMP-Control is used to initiate, start, and stop test sessions and to fetch their results, whereas O/TWAMP-Test is used to exchange test packets between two measurement nodes.

In the remainder of this section we review security for O/TWAMP-Control and O/TWAMP-Test separately and then make the case for using them over IPsec.

3.1. O/TWAMP-Control Security

O/TWAMP uses a simple cryptographic protocol which relies on

- AES in Cipher Block Chaining (AES-CBC) for confidentiality

- HMAC-SHA1 truncated to 128 bits for message authentication

Three modes of operation are supported: unauthenticated, authenticated, and encrypted. The authenticated and encrypted 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 and encrypted modes, security parameters are negotiated during the control connection establishment.

Figure 1 illustrates the initiation stage of the O/TWAMP-Control protocol between a client and the server. In short, the client opens a TCP connection to the server in order to be able to send OWAMP-Control commands. The server responds with a Server Greeting, which contains the Modes, Challenge, Salt, Count, and MBZ fields (see Section 3.1 of [RFC4656]). If the 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 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.

```
+---------+                   +---------+
| 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 and encrypted modes, all further communication is encrypted using the AES Session-key and authenticated with the HMAC Session-key. The 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 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.

3.2. O/TWAMP-Test Security

The O/TWAMP-Test protocol runs over UDP, using the sender and receiver IP and port numbers that were negotiated during the Request-Session exchange. O/TWAMP-Test has the same three modes as with O/TWAMP-Control (unauthenticated, authenticated, and encrypted) and all O/TWAMP-Test sessions inherit the corresponding O/TWAMP-Control session 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 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 of 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 session identifier (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 session identifier (SID), for authenticating the packets of the particular O/TWAMP-Test session.

3.3. O/TWAMP Security Root

As discussed above, the AES Session-key and HMAC Session-key used in the O/TWAMP-Test protocol are derived from the AES Session-key and HMAC Session-key which are used in O/TWAMP-Control protocol. The AES Session-key and HMAC Session-key used in the O/TWAMP-Control protocol are generated randomly by the 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/IPsec can automatically manage the security root and solve this problem, as we explain in Section 4.

3.4. O/TWAMP and IPsec

According to [RFC4656] the "deployment paths of IPsec and OWAMP could be separate if OWAMP does not depend on IPsec." However, the problem that arises in practice is that the security mechanism of O/TWAMP and IPsec cannot coexist at the same time without adding overhead or increasing complexity.

IPsec provides confidentiality and data integrity to IP datagrams. Distinct protocols are provided: Authentication Header (AH), Encapsulating Security Payload (ESP) and Internet Key Exchange (IKE v1/v2). AH provides only integrity protection, while ESP can also provide encryption. IKE is used for dynamical key negotiation and automatic key management.

When sender and receiver implement O/TWAMP over IPsec, they need to agree on a shared secret key during the IPsec tunnel establishment. Subsequently, all IP packets sent by the sender are protected. If the AH protocol is used, IP packets are transmitted in plaintext.

The authentication part covers the entire packet. So all test information, such as UDP port number, and the test results will be visible to any attacker, which can intercept these test packets, and introduce errors or forge packets that may be injected during the transmission. In order to avoid this attack, the receiver must validate the integrity of these packets with the negotiated secret key. If ESP is used, IP packets are encrypted, and hence only the receiver can use the IPsec secret key to decrypt the IP packet, and obtain the test data in order to assess the IP network performance based on the measurements. Both the sender and receiver must support IPsec to generate the security secret key of IPsec.

Currently, after the test packets are received by the receiver, it cannot execute active measurement over IPsec. That is because the receiver knows only the shared secret key but not the IPsec key, while the test packets are protected by the IPsec key ultimately. Therefore, it needs to be considered how to measure IP network performance in an IPsec tunnel with O/TWAMP. Without this functionality, the use of OWAMP and TWAMP over IPsec is hindered.

Of course, backward compatibility should be considered as well. That is, the intrinsic security method based on shared key as specified in the O/TWAMP standards can also still be suitable for other network settings. There should be no impact on the current security mechanisms defined in O/TWAMP for other use cases. This document describes possible solutions to this problem which take advantage of the secret key derived by IPsec, in order to provision the key needed for active network measurements based on [RFC4656] and [RFC5357].

4. 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 which both support IPsec. In short, the shared key used for securing O/TWAMP traffic is derived using IKEv2 [RFC5996].

4.1. Shared Key Derivation

If the AH protocol is used, the IP packets are transmitted in plaintext, but all O/TWAMP traffic is integrity-protected by IPsec. Therefore, even if the peers choose to opt for the unauthenticated mode, IPsec integrity protection is extended to O/TWAMP. In the authenticated and encrypted modes, the shared secret can be derived from the IKEv2 Security Association (SA), or IPsec SA.

If the shared secret key is derived from the IKEv2 SA, SKEYSEED must be generated firstly. SKEYSEED and its derivatives are computed as per [RFC5996], where prf is a pseudorandom function:

```
SKEYSEED = prf( Ni | Nr, g^ir )
```

Ni and Nr are, respectively, the initiator and responder nonces, which are negotiated during the initial exchange (see Section 1.2 of [RFC5996]). g^ir is the shared secret from the ephemeral Diffie-Hellman exchange and is represented as a string of octets. Note that this SKEYSEED can be used as the O/TWAMP shared secret key directly.

Alternatively, the shared secret key can be generated as follows:

```
Shared secret key = PRF{ SKEYSEED, Session ID }
```

wherein the Session ID is the O/TWAMP-Test SID.

If the shared secret key is derived from the IPsec SA, instead, the shared secret key can be equal to KEYMAT, wherein

```
KEYMAT = prf+( SK_d, Ni | Nr )
```

The term "prf+" stands for a function that outputs a pseudorandom stream based on the inputs to a prf, while SK_d is defined in [RFC5996] (Sections 2.13 and 1.2, respectively). The shared secret key can alternatively be generated as follows:

```
Shared secret key = PRF{ KEYMAT, Session ID }
```

wherein the session ID is is the O/TWAMP-Test SID.

If rekeying for the IKE SA and IPsec SA occurs, the corresponding key of the SA is updated. Generally, ESP and AH SAs always exist in pairs, with one SA in each direction. If the SA is deleted, the key generated from the IKE SA or IPsec SA should also be updated.

4.2. Server Greeting Message Update

As discussed above, a binding association between the key generated from IPsec and the O/TWAMP shared secret key needs to be considered. The Security Association (SA) can be identified by the Security Parameter Index (SPI) and protocol uniquely for a given sender and receiver pair. Therefore, these parameters should be agreed upon during the initiation stage of O/TWAMP-Control. At the stage that the sender and receiver negotiate the integrity key, the IPsec protocol and SPI MUST be checked. Only if the two parameters are matched with the IPsec information, MUST the O/TWAMP connection be established.

The Security Parameter Index (SPI) and protocol type (see [RFC4301] [RFC5996]) will need to be included in the Server Greeting of the O/

TWAMP-Control protocol depicted in Figure 1. After the client receives the greeting, it MUST close the connection if it receives a greeting with an erroneous SPI and protocol value (Figure 2). Otherwise, the client SHOULD generate the shared secret key as discussed in Section 4.1 and respond with the server-expected Set-Up-Response message.

The Modes field in Figure 2 will need to allow for support of key derivation as discussed in Section 4.1. As such, pending discussion in the IPPM WG, Modes value 8 MUST be supported by compatible implementations, indicating support for IPsec. Server implementations compatible with this document MUST set the first 28 bits of the Modes field to zero. A client compatible with this specification MUST ignore the first 28 bits of the Modes field. For backward compatibility, the server is obviously allowed to indicate support for the Modes defined in [RFC4656]

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Protocol                             |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SPIi                                 |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SPIr                                 |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Modes                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Challenge (16 octets)                     |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                        Salt (16 octets)                       |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Count (4 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         MBZ (12 octets)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Server Greeting format

A compatible O/TWAMP client implementation would then interpret the originally unused 12 bits of the Server Greeting (see sec. 3.1 of [RFC4656]) as follows: The first 4 octets of the Server Greeting message indicate the protocol type, while the following 8 octets indicate the initiator (SPIi) and responder (SPIr) SPIs as illustrated in Figure 2. Note that in this case, the remaining fields of the Server Greeting message remain as per [RFC4656].

EDITOR'S NOTE:

> We expect that this implementation option would pose the least backwards compatibility problems to existing O/TWAMP clients. Robust client implementations of [RFC4656] would disregard that the 29th Modes bit in the Server Greeting is set, and should ignore the information contained in the newly defined fields (Protocol, SPIi, SPIr). If the server supports other Modes, as one would assume, the client would then indicate any of the Modes defined in [RFC4656] and effectively indicate that it does not support the IPsec mode. At this point, the Server would need to use the Modes defined in [RFC4656] only.

When using ESP, all IP packets are encrypted, and therefore only the receiver can use the IPsec key to decrypt the IP active measurement packets. In this case, the IPsec tunnel between the sender and receiver provides additional security: even if the peers choose the unauthenticated mode, IPsec encryption and integrity protection is provided to O/TWAMP. If the sender and receiver decide to use the authenticated or encrypted mode, the shared secret can also be derived from IKE SA or IPsec SA. The method for key generation and binding association is the same discussed above for the AH protocol mode.

There is an encryption-only configuration in ESP, though this is not recommended due to its limitations. Since it does not produce integrity key in this case, either encryption-only ESP should be prohibited for O/TWAMP, or a decryption failure should be distinguished due to possible integrity attack.

4.3. Session Key Derivation

Section 4.1 described a method for deriving the shared key for O/TWAMP by capitalizing on IPsec. This is a step forward in terms of facilitating O/TWAMP deployment at scale in IPsec networks as it allows for greater and secure automation of standardized network performance measurements. We note, however, that the O/TWAMP protocol uses distinct encryption and integrity keys for O/TWAMP-Control and O/TWAMP-Test. Consequently, four keys are generated to protect O/TWAMP-Control and O/TWAMP-Test messages.

In fact, once IPsec is employed, one key for encryption and another for authentication is sufficient for both the Control and Test protocols. Therefore, in an IPsec environment we can further reduce the operational complexity of O/TWAMP protocols in a straightforward manner, as discussed below.

EDITOR'S NOTE:

> We expect that both session key derivation proposals and optimization alternatives will be discussed in the IPPM working group and we are looking forward to community comments and feedback.

4.3.1. Alternative 1

If an IPsec SA is established between the server and the client, or both server and client support IPsec, the root key for O/TWAMP-based active network measurements can be derived from the IKE or IPsec SA.

If the root key that will be used in O/TWAMP network performance measurements is derived from the IKE SA, SKEYSEED must be generated first. SKEYSEED and its derivatives are computed as per [RFC5996]. SKEYSEED can be used as the root key of O/TWAMP directly; then the root key of O/TWAMP is equal to SKEYSEED. If the root key of O/TWAMP is derived from the IPsec SA, the shared secret key can be equal to KEYMAT. KEYMAT and its derivatives are computed as per usual

[RFC5996].

Then, the session keys for encryption and authentication can be derived from the root key of O/TWAMP, wherein:

Session key for enc = PRF{ root key of O/TWAMP, "O/TWAMP enc" }

Session key for auth = PRF{ root key of O/TWAMP, "O/TWAMP auth" }

The former can provide encryption protection for O/TWAMP-Control and O/TWAMP-Test messages, while the latter can provide integrity protection.

Note that there are cases where rekeying the IKE SA and IPsec SA is necessary, and after which the corresponding key of SA is updated. If the SA is deleted, the O/TWAMP shared key generated from the IKE SA or IPsec SA should also be updated.

EDITOR'S NOTE:

In addition to optimizing session key derivation, we can also reduce the verbosity of the Server Greeting and Set-Up-Response messages, as explained below. Note, however, that such O/TWAMP message simplification poses backward compatibility challenges, which should be discussed in the IPPM WG.

In this optimization, the O/TWAMP-Control message exchange flow remains as per Figure 1. However, the optimized Server Greeting (Figure 3) can do without the Salt and Count parameters (cf. Figure 2) since the root key of O/TWAMP is derived from IKE SA or IPsec SA. O/TWAMP security can rely on IPsec and the SPI can uniquely identify the IPsec SA from which the root key was derived from.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Protocol                             |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SPIi                                |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SPIr                                |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Modes                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Challenge (16 octets)                     |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                        MBZ (12 octets)                        |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Optimized Server Greeting format

The format of the Set-Up-Response is illustrated in Figure 4. The Token carried in the Set-Up-Response is calculated as follows:

```
      Token = Enc_root-key( Challenge )
```

where Challenge is the value received earlier in the Server Greeting (Figure 3)

```
   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                               |
```

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                     Token (16 octets)                         |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                    Client-IV (12 octets)                      |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Set-Up-Response in Alternative 1

If the server authenticates the token successfully, then the O/TWAMP-Control message exchange flow can continue.

4.3.2. Alternative 2

Another way for optimizing the shared key use is to set the O/TWAMP session keys equal to the keys of the IPsec SA directly, i.e:

Session key for enc = encryption key of the IPsec SA

Session key for auth = integrity key of the IPsec SA

The former session key can provide encryption protection for O/TWAMP-Control and O/TWAMP-Test messages, while the latter can provide integrity protection. The point made in the previous subsection about rekeying the IPsec SA applies here too.

EDITOR'S NOTE:

> As noted in the previous subsection, here too we can reduce the verbosity of the Server Greeting and Set-Up-Response messages even further, as explained below. Note, however, that such O/TWAMP message simplification poses backward compatibility challenges, which should be discussed in the IPPM WG.

The O/TWAMP control message exchange flow remains the same (i.e. as per Figure 1), while the Server Greeting format is illustrated in Figure 5. The Challenge, Salt, and Count parameters can be eliminated since the session keys of O/TWAMP are equal to the keys of an IPsec SA directly. SPI can identify the IPsec SA where the session keys derived from. The similarly optimized Set-Up-Response message is illustrated in Figure 6.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|                           Protocol                            |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SPIi                                |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SPIr                                |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Modes                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                        MBZ (12 octets)                        |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Optimized Server Greeting format

```
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                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Client-IV (12 octets)                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Set-Up-Response in Alternative 2

5. Security Considerations

As the shared secret key is derived from IPsec, the key derivation algorithm strength and limitations are as per [RFC5996]. 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 [RFC5996].

EDITOR'S NOTE:

> As a general note, the IPPM community may want to revisit the arguments listed in [RFC4656], Sec. 6.6. Other widely-used Internet security mechanisms, such as TLS and DTLS, may also be considered for future use over and above of what is already specified in [RFC4656] [RFC5357].

6. IANA Considerations

IANA may need to allocate additional values for the Modes options presented in this document. The values of the protocol field may need to be assigned from the numbering space.

7. Acknowledgments

Emily Bi contributed to an earlier version of this document.

We thank Eric Chen and Yakov Stein for their comments on this draft, and Al Morton for the discussion on related earlier work in IPPM WG.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

8.2. Informative References

[RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, September 2000.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

Authors' Addresses

Kostas Pentikousis (editor)
EICT GmbH
Torgauer Strasse 12-15
10829 Berlin
Germany

Email: k.pentikousis@eict.de

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

Email: cuiyang@huawei.com

Emma Zhang
Huawei Technologies
Huawei Building, Q20, No.156, Rd. BeiQing
Haidian District , Beijing   100095
P. R. China

Email: emma.zhanglijia@huawei.com