Network Working Group                                          B. Weis
Request for Comments: 4359                                Cisco Systems
Category: Standards Track                                  January 2006


                  The Use of RSA/SHA-1 Signatures within
       Encapsulating Security Payload (ESP) and Authentication Header (AH)

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements.  Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol.  Distribution of this memo is unlimited.

Copyright Notice

Abstract

   This memo describes the use of the RSA digital signature algorithm as
   an authentication algorithm within the revised IP Encapsulating
   Security Payload (ESP) as described in RFC 4303 and the revised IP
   Authentication Header (AH) as described in RFC 4302.  The use of a
   digital signature algorithm, such as RSA, provides data origin
   authentication in applications when a secret key method (e.g., HMAC)
   does not provide this property.  One example is the use of ESP and AH
   to authenticate the sender of an IP multicast packet.

Table of Contents

1.  Introduction

   Encapsulating Security Payload  (ESP) [ESP] and Authentication Header
   (AH) [AH] headers can be used to protect both unicast traffic and
   group (e.g., IPv4 and IPv6 multicast) traffic.  When unicast traffic
   is protected between a pair of entities, HMAC transforms (such as
   [HMAC-SHA]) are sufficient to prove data origin authentication.  An
   HMAC is sufficient protection in that scenario because only the two
   entities involved in the communication have access to the key, and
   proof-of-possession of the key in the HMAC construct authenticates
   the sender.  However, when ESP and AH authenticate group traffic,
   this property no longer holds because all group members share the
   single HMAC key.  In the group case, the identity of the sender is
   not uniquely established, since any of the key holders has the
   ability to form the HMAC transform.  Although the HMAC transform
   establishes a group-level security property, data origin
   authentication is not achieved.

   Some group applications require true data origin authentication,
   where one group member cannot successfully impersonate another group
   member.  The use of asymmetric digital signature algorithms, such as
   RSA, can provide true data origin authentication.

   With asymmetric algorithms, the sender generates a pair of keys, one
   of which is never shared (called the "private key") and one of which
   is distributed to other group members (called the "public key").

Weis                        Standards Track                     [Page 2]

When the private key is used to sign the output of a cryptographic hash algorithm, the result is called a "digital signature".  A receiver of the digital signature uses the public key, the signature value, and an independently computed hash to determine whether or not the claimed origin of the packet is correct.

This memo describes how RSA digital signatures can be applied as an ESP and AH authentication mechanism to provide data origin authentication.

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].

2.  Algorithm and Mode

The RSA Public Key Algorithm [RSA] is a widely deployed public key algorithm commonly used for digital signatures.  Compared to other public key algorithms, signature verification is relatively efficient.  This property is useful for groups where receivers may have limited processing capabilities.  The RSA algorithm is commonly supported in hardware.

Two digital signature encoding methods are supported in [RSA]. RSASSA-PKCS1-v1_5 MUST be supported by a conforming implementation. RSASSA-PSS is generally believed to be more secure, but at the time of this writing is not ubiquitous.  RSASSA-PSS SHOULD be used whenever it is available.  SHA-1 [SHA] MUST be used as the signature hash algorithm used by the RSA digital signature algorithm.

When specified for ESP, the Integrity Check Value (ICV) is equal in size to the RSA modulus, unless the RSA modulus is not a multiple of 8 bits.  In this case, the ICV MUST be prepended with between 1 and 7 bits set to zero such that the ICV is a multiple of 8 bits.  This specification matches the output S [RSA, Section 8.1.1] (RSASSA-PSS) and [RSA, Section 8.2.1] (RSASSA-PKCS1-v1_5) when the RSA modulus is not a multiple of 8 bits.  No implicit ESP ICV Padding bits are necessary.

When specified for AH, the ICV is equal in size of the RSA modulus, unless the RSA modulus is not a multiple of 32 bits (IPv4) or 64 bits (IPv6) [AH, Section 2.6].  In this case, explicit ICV Padding bits are necessary to create a suitably sized ICV [AH, Section 3.3.3.2.1].

The distribution mechanism of the RSA public key and its replacement interval are a group policy matter.  The use of an ephemeral key pair with a lifetime of the ESP or AH Security Association (SA) is RECOMMENDED.  This recommended policy reduces the exposure of the RSA

private key to the lifetime of the data being signed by the private
key.  Also, this obviates the need to revoke or transmit the validity
period of the key pair.

Digital signature generation is performed as described in [RSA,
Section 8.1.1] (RSASSA-PSS) and [RSA, Section 8.2.1](RSASSA-PKCS1-
v1_5).  The authenticated portion of the AH or ESP packet ([AH,
Section 3.3.3], [ESP, Section 3.3.2]) is used as the message M, which
is passed to the signature generation function.  The signer's RSA
private key is passed as K.  Summarizing, the signature generation
process computes a SHA-1 hash of the authenticated packet bytes,
signs the SHA-1 hash using the private key, and encodes the result
with the specified RSA encoding type.  This process results in a
value S, which is known as the ICV in AH and ESP.

Digital signature verification is performed as described in [RSA,
Section 8.1.2] (RSASSA-PSS) and [RSA, Section 8.2.2]
(RSASSA-PKCS1-v1_5).  Upon receipt, the ICV is passed to the
verification function as S.  The authenticated portion of the AH or
ESP packet is used as the message M, and the RSA public key is passed
as (n, e).  In summary, the verification function computes a SHA-1
hash of the authenticated packet bytes, decrypts the SHA-1 hash in
the ICV, and validates that the appropriate encoding was applied and
was correct.  The two SHA-1 hashes are compared, and if they are
identical the validation is successful.

2.1.  Key Size Discussion

The choice of RSA modulus size must be made carefully.  If too small
of a modulus size is chosen, an attacker may be able to reconstruct
the private key used to sign packets before the key is no longer used
by the sender to sign packets.  This order of events may result in
the data origin authentication property being compromised.  However,
choosing a modulus size larger than necessary will result in an
unnecessarily high cost of CPU cycles for the sender and all
receivers of the packet.

A conforming implementation MUST support a modulus size of 1024 bits.

Recent guidance [TWIRL, RSA-TR] on key sizes makes estimates as to
the amount of effort an attacker would need to expend in order to
reconstruct an RSA private key.  Table 1 summarizes the maximum
length of time that selected modulus sizes should be used.  Note that
these recommendations are based on factors such as the cost of
processing and memory, as well as cryptographic analysis methods,
which were current at the time these documents were published.  As
those factors change, choices of key lifetimes should take them into
account.

```
            Number of      Recommended Maximum
           Modulus Bits         Lifetime
           ------------     -------------------
               768          1 week
              1024          1 year
```

Table 1.  RSA Key Use Lifetime Recommendations

3.  Performance

   The RSA asymmetric key algorithm is very costly in terms of
   processing time compared to the HMAC algorithms.  However, processing
   cost is decreasing over time.  Faster general-purpose processors are
   being deployed, faster software implementations are being developed,
   and hardware acceleration support for the algorithm is becoming more
   prevalent.

   Care should be taken that RSA signatures are not used for
   applications when potential receivers are known to lack sufficient
   processing power to verify the signature.  It is also important to
   use this scheme judiciously when any receiver may be battery powered.

   The RSA asymmetric key algorithm is best suited to protect network
   traffic for which:

    o The sender has a substantial amount of processing power, and

    o The network traffic is small enough that adding a relatively large
      authentication tag (in the range of 62 to 256 bytes) does not
      cause packet fragmentation.

   RSA key pair generation and signing are substantially more expensive
   operations than signature verification, but these are isolated to the
   sender.

   The size of the RSA modulus affects the processing required to create
   and verify RSA digital signatures.  Care should be taken to determine
   the size of modulus needed for the application.  Smaller modulus
   sizes may be chosen as long as the network traffic protected by the
   private key flows for less time than it is estimated that an attacker
   would take to discover the private key.  This lifetime is
   considerably smaller than most public key applications that store the
   signed data for a period of time.  But since the digital signature is
   used only for sender verification purposes, a modulus that is
   considered weak in another context may be satisfactory.

The size of the RSA public exponent can affect the processing
required to verify RSA digital signatures.  Low-exponent RSA
signatures may result in a lower verification processing cost.  At
the time of this writing, no attacks are known against low-exponent
RSA signatures that would allow an attacker to create a valid
signature using the RSAES-OAEP scheme.

The addition of a digital signature as an authentication tag adds a
significant number of bytes to the packet.  This increases the
likelihood that the packet encapsulated in ESP or AH may be
fragmented.

4.  Interaction with the ESP Cipher Mechanism

The RSA signature algorithm cannot be used with an ESP Combined Mode
algorithm that includes an explicit ICV.  The Combined Mode algorithm
will add the ESP ICV field, which does not allow use of a separate
authentication algorithm to add the ESP ICV field.  One example of
such an algorithm is the ESP Galois/Counter Mode algorithm [AES-GCM].

5.  Key Management Considerations

Key management mechanisms negotiating the use of RSA signatures MUST
include the length of the RSA modulus during policy negotiation using
the Authentication Key Length SA Attribute.  This gives a device the
opportunity to decline use of the algorithm.  This is especially
important for devices with constrained processors that might not be
able to verify signatures using larger key sizes.

Key management mechanisms negotiating the use of RSA signatures also
MUST include the encoding method during policy negotiation using the
Signature Encoding Algorithm SA Attribute.

A receiver must have the RSA public key in order to verify integrity
of the packet.  When used with a group key management system (e.g.,
RFC 3547 [GDOI]), the public key SHOULD be sent as part of the key
download policy.  If the group has multiple senders, the public key
of each sender SHOULD be sent as part of the key download policy.

Use of this transform to obtain data origin authentication for
pairwise SAs is NOT RECOMMENDED.  In the case of pairwise SAs (such
as negotiated by the Internet Key Exchange [IKEV2]), data origin
authentication can be achieved with an HMAC transform.  Because the
performance impact of an RSA signature is typically greater than an
HMAC, the value of using this transform for a pairwise connection is
limited.

6.  Security Considerations

   This document provides a method of authentication for ESP and AH
   using digital signatures.  This feature provides the following
   protections:

   o Message modification integrity.  The digital signature allows the
     receiver of the message to verify that it was exactly the same as
     when the sender signed it.

   o Host authentication.  The asymmetric nature of the RSA public key
     algorithm allows the sender to be uniquely verified, even when the
     message is sent to a group.

   Non-repudiation is not claimed as a property of this transform.  At
   times, the property of non-repudiation may be applied to digital
   signatures on application-level objects (e.g., electronic mail).
   However, this document describes a means of authenticating network-
   level objects (i.e., IP packets), which are ephemeral and not
   directly correlated to any application.  Non-repudiation is not
   applicable to network-level objects (i.e., IP packets).

   A number of attacks are suggested by [RFC3552].  The following
   sections describe the risks those attacks present when RSA signatures
   are used for ESP and AH packet authentication.

   SHA-1 has been scheduled to be phased out in 2010, due to the steady
   advances in technology by which an adversary can double its computing
   power in roughly eighteen months.  Recent attacks on SHA-1 underscore
   the importance of replacing SHA-1, but have not motivated replacing
   it before that date [SHA-COMMENTS].  The use of this transform after
   that date SHOULD be preceded by an analysis as to its continued
   suitability.

6.1.  Eavesdropping

   This document does not address confidentiality.  That function, if
   desired, must be addressed by an ESP cipher that is used with the RSA
   signatures authentication method.  The RSA signature itself does not
   need to be protected from an eavesdropper.

6.2.  Replay

   This document does not address replay attacks.  That function, if
   desired, is addressed through use of ESP and AH sequence numbers as
   defined in [ESP] and [AH].

6.3.  Message Insertion

   This document directly addresses message insertion attacks.  Inserted
   messages will fail authentication and be dropped by the receiver.

6.4.  Deletion

   This document does not address deletion attacks.  It is concerned
   only with validating the legitimacy of messages that are not deleted.

6.5.  Modification

   This document directly addresses message modification attacks.
   Modified messages will fail authentication and be dropped by the
   receiver.

6.6.  Man in the Middle

   As long as a receiver is given the sender RSA public key in a trusted
   manner (e.g., by a key management protocol), it will be able to
   verify that the digital signature is correct.  A man in the middle
   will not be able to spoof the actual sender unless it acquires the
   RSA private key through some means.

   The RSA modulus size must be chosen carefully to ensure that the time
   a man in the middle needs to determine the RSA private key through
   cryptanalysis is longer than the amount of time that packets are
   signed with that private key.

6.7.  Denial of Service

   According to IPsec processing rules, a receiver of an ESP and AH
   packet begins by looking up the Security Association in the SA
   database.  If one is found, the ESP or AH sequence number in the
   packet is verified.  No further processing will be applied to packets
   with an invalid sequence number.

   An attacker that sends an ESP or AH packet matching a valid SA on the
   system and also having a valid sequence number will cause the
   receiver to perform the ESP or AH authentication step.  Because the
   process of verifying an RSA digital signature consumes relatively
   large amounts of processing, many such packets could lead to a denial
   of service (DoS) attack on the receiver.

   If the message was sent to an IPv4 or IPv6 multicast group, all group
   members that received the packet would be under attack
   simultaneously.

This attack can be mitigated against most attackers by encapsulating
ESP or AH using an RSA signature for authentication within ESP or AH
using an HMAC transform for authentication.  In this case, the HMAC
transform would be validated first, and as long as the attacker does
not possess the HMAC key no digital signatures would be evaluated on
the attacker packets.  However, if the attacker does possess the HMAC
key (e.g., the attacker is a legitimate member of the group using the
SA), then the DoS attack cannot be mitigated.

7.  IANA Considerations

   An assigned number is required in the "IPSec Authentication
   Algorithm" name space in the Internet Security Association and Key
   Management Protocol (ISAKMP) registry [ISAKMP-REG].  The mnemonic
   should be "SIG-RSA".

   An assigned number is also required in the "IPSEC AH Transform
   Identifiers" name space in the ISAKMP registry.  Its mnemonic should
   be "AH_RSA".

   A new "IPSEC Security Association Attribute" is required in the
   ISAKMP registry to pass the RSA modulus size.  The attribute class
   should be called "Authentication Key Length", and it should be a
   Variable type.

   A second "IPSEC Security Association Attribute" is required in the
   ISAKMP registry to pass the RSA signature encoding type.  The
   attribute class should be called "Signature Encoding Algorithm", and
   it should be a Basic type.  The following rules apply to define the
   values of the attribute:

                   Name                Value
                   ----                -----
                   Reserved            0
                   RSASSA-PKCS1-v1_5    1
                   RSASSA-PSS           2

   Values 3-61439 are reserved to IANA.  New values MUST be added due to
   a Standards Action as defined in [RFC2434].  Values 61440-65535 are
   for private use and may be allocated by implementations for their own
   purposes.

8.  Acknowledgements

   Scott Fluhrer and David McGrew provided advice regarding applicable
   key sizes.  Scott Fluhrer also provided advice regarding key
   lifetimes.  Ian Jackson, Steve Kent, and Ran Canetti provided many
   helpful comments.  Sam Hartman, Russ Housley, and Lakshminth Dondeti
   provided valuable guidance in the development of this document.

9.  References

9.1.  Normative References

   [AH]            Kent, S., "IP Authentication Header", RFC 4302,
                   December 2005.

   [ESP]           Kent, S., "IP Encapsulating Security Payload (ESP)",
                   RFC 4303, December 2005.

   [ISAKMP-REG]    http://www.iana.org/assignments/isakmp-registry

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

   [RFC3552]       Rescorla, E. and B. Korver, "Guidelines for Writing
                   RFC Text on Security Considerations", BCP 72, RFC
                   3552, July 2003.

   [RSA]           Jonsson, J. and B. Kaliski,  "Public-Key Cryptography
                   Standard (PKCS) #1: RSA Cryptography Specifications
                   Version 2.1", RFC 3447, February 2003.

   [SHA]           FIPS PUB 180-2: Specifications for the Secure Hash
                   Standard, August 2002.  http://csrc.nist.gov/
                   publications/fips/fips180-2/fips180-2.pdf.

9.2.  Informative References

   [AES-GCM]       Viega, J. and D. McGrew, "The Use of Galois/Counter
                   Mode (GCM) in IPsec Encapsulating Security Payload
                   (ESP)", RFC 4106, June 2005.

   [GDOI]          Baugher, M., Weis, B., Hardjono, T., and H. Harney,
                   "The Group Domain of Interpretation", RFC 3547,
                   December 2002.

   [HMAC-SHA]      Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96
                   within ESP and AH", RFC 2404, November 1998.

   [IKEV2]            Kaufman, C., "Internet Key Exchange (IKEv2) Protocol",
                      RFC 4306, December 2005.

   [RFC2434]          Narten, T. and H. Alvestrand, "Guidelines for Writing
                      an IANA Considerations Section in RFCs", BCP 26, RFC
                      2434, October 1998.

   [RSA-TR]           B. Kaliski, "TWIRL and RSA Key Size", RSA Laboratories
                      Technical Note, http://www.rsasecurity.com/rsalabs/
                      node.asp?id=2004, May 6, 2003.

   [SHA-COMMENTS]  NIST Brief Comments on Recent Cryptanalytic Attacks on
                      Secure Hashing Functions and the Continued Security
                      Provided by SHA-1, August, 2004.
                      http://csrc.nist.gov/hash_standards_comments.pdf.

   [TWIRL]            Shamir, A., and E. Tromer, "Factoring Large Numbers
                      with the TwIRL Device", Work in Progress, February 9,
                      2003.

Author's Address

   Brian Weis
   Cisco Systems
   170 W. Tasman Drive,
   San Jose, CA 95134-1706, USA

   Phone: (408) 526-4796
   EMail: bew@cisco.com