Network Working Group
Request for Comments: 4894
Category: Informational

P. Hoffman
VPN Consortium

May 2007

# Use of Hash Algorithms in Internet Key Exchange (IKE) and IPsec

## Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

## Copyright Notice

## Abstract

This document describes how the IKEv1 (Internet Key Exchange version 1), IKEv2, and IPsec protocols use hash functions, and explains the level of vulnerability of these protocols to the reduced collision resistance of the MD5 and SHA-1 hash algorithms.

## Table of Contents

## 1. Introduction

Recently, attacks on the collision-resistance properties of MD5 and SHA-1 hash functions have been discovered; [HashAttacks] summarizes the discoveries. The security community is now reexamining how various Internet protocols use hash functions. The goal of this reexamination is to be sure that the current usage is safe in the face of these new attacks, and whether protocols can easily use new hash functions when they become recommended.

Different protocols use hash functions quite differently. Because of this, the IETF has asked for reviews of all protocols that use hash functions. This document reviews the many ways that three protocols (IKEv1 [IKEv1], IKEv2 [IKEv2], and IPsec [ESP] and [AH]) use hash functions.

In this document, "IKEv1" refers to only "Phase 1" of IKEv1 and the agreement process. "IKEv2" refers to the IKE_SA_INIT and IKE_AUTH exchanges. "IPsec" refers to IP encapsulated in either the Authentication Header (AH) or Encapsulating Security Payload (ESP).

## 2. Hashes in IKEv1 and IKEv2

Both IKEv1 and IKEv2 can use hash functions as pseudo-random functions (PRFs). The inputs to the PRFs always contain nonce values from both the initiator and the responder that the other party cannot predict in advance. In IKEv1, the length of this nonce is at least 64 bits; in IKEv2, it is at least 128 bits. Because of this, the use of hash functions in IKEv1 and IKEv2 are not susceptible to any known collision-reduction attack.

IKEv1 also uses hash functions on the inputs to the PRF. The inputs are a combination of values from both the initiator and responder, and thus the hash function here is not susceptible to any known collision-reduction attack.

In IKEv2, hashes are used as integrity protection for all messages after the IKE_SA_INIT Exchange. These hashes are used in Hashed Message Authentication Codes (HMACs). As described in [HMAC-reduction], MD5 used in HMACs is susceptible to forgery, and it is suspected that full SHA-1 used in HMAC is susceptible to forgery. There is no known reason for the person who creates legitimate integrity protection to want to spoof it.

Both IKEv1 and IKEv2 have authentication modes that use digital signatures. Digital signatures use hashes to make unique digests of the message being signed. With the current known attacks, the only party that can create the two messages that collide is the IKE entity that generates the message. As shown in [Target-collisions], an attacker can create two different Public Key Infrastructure using X.509 (PKIX) certificates with different identities that have the same signatures.

IKEv1 has two modes, "public key encryption" and "revised public key encryption", that use hashes to identify the public key used. The hash function here is used simply to reduce the size of the identifier. In IKEv2 with public-key certificates, a hash function is used for similar purposes, both for identifying the sender's public key and the trust anchors. Using a collision-reduction attack, an individual could create two public keys that have the same hash value. This is not considered to be a useful attack because the key generator holds both private keys.

IKEv1 can be used together with Network Access Translator (NAT) traversal support, as described in [NAT-T]; IKEv2 includes this NAT traversal support. In both of these cases, hash functions are used to obscure the IP addresses used by the initiator and/or the responder. The hash function here is not susceptible to any known collision-reduction attack.

## 3. Hashes in IPsec

AH uses hash functions for authenticating packets; the same is true for ESP when ESP is using its own authentication. For both uses of IPsec, hash functions are always used in hashed MACs (HMACs). As described in [HMAC-reduction], MD5 used in HMACs is susceptible to forgery, and it is suspected that full SHA-1 used in HMAC is susceptible to forgery. There is no known reason for the person who creates legitimate packet authentication to want to spoof it.

## 4. PKIX Certificates in IKEv1 and IKEv2

Some implementations of IKEv1 and IKEv2 use PKIX certificates for authentication. Any weaknesses in PKIX certificates due to particular ways hash functions are used, or due to weaknesses in particular hash functions used in certificates, will be inherited in IKEv1 and IKEv2 implementations that use PKIX-based authentication.

## 5. Choosing Cryptographic Functions

Recently, there has been more discussion in the IETF about the ability of one party in a protocol to tell the other party which cryptographic functions the first party prefers the second party to use. The discussion was spurred in part by [Deploying]. Although that paper focuses on hash functions, it is relevant to other cryptographic functions as well.

There are (at least) three distinct subtopics related to choosing cryptographic functions in protocols:

- The ability to pick between different cryptographic functions instead of having just one specified in the protocol

- If there are multiple functions, the ability to agree on which function will be used in the main protocol

- The ability to suggest to the other party which kinds of cryptographic functions should be used in the other party's public key certificates

### 5.1. Different Cryptographic Functions

Protocols that use cryptographic functions can either specify a single function, or can allow different functions. Protocols in the first category are susceptible to attack if the specified function is later found to be too weak for the stated purpose; protocols in the second category can usually avoid such attacks, but at a cost of increased protocol complexity. In the IETF, protocols that allow a choice of cryptographic functions are strongly preferred.

IKEv1, IKEv2, and IPsec already allow different hash functions in every significant place where hash functions are used (that is, in every place that has any susceptibility to a collision-reduction attack).

### 5.2. Specifying Cryptographic Functions in the Protocol

Protocols that allow a choice of cryptographic functions need to have a way for all parties to agree on which function is going to be used. Some protocols, such as secure electronic mail, allow the initiator to simply pick a set of cryptographic functions; if the responder does not understand the functions used, the transmission fails. Other protocols allow for the two parties to agree on which cryptographic functions will be used. This is sometimes called "negotiation", but the term "negotiation" is inappropriate for protocols in which one party (the "proposer") lists all the functions it is willing to use, and the other party (the "chooser") simply picks the ones that will be used.

When a new cryptographic function is introduced, one party may want to tell the other party that they can use the new function. If it is the proposer who wants to use the new function, the situation is easy: the proposer simply adds the new function to its list, possibly removing other parallel functions that the proposer no longer wants to use.

On the other hand, if it is the chooser who wants to use the new function and the proposer didn't list it, the chooser may want to signal the proposer that they are capable of using the new function or the chooser may want to say that it is only willing to use the new function. If a protocol wants to handle either of these cases, it has to have a way for the chooser to specify this information to the proposer in its acceptance and/or rejection message.

It is not clear from a design standpoint how important it might be to let the chooser specify the additional functions it knows. As long as the proposer offers all the functions it wants to use, there is no reason for the chooser to say "I know one you don't know". The only place where the chooser is able to signal the proposer with different functions is in protocols where listing all the functions might be prohibitive, such as where they would add additional round trips or significant packet length.

IKEv1 and IKEv2 allow the proposer to list all functions. Neither allows the chooser to specify which functions that were not proposed it could have used, either in a successful or unsuccessful Security Association (SA) establishment.

## 5.3. Specifying Cryptographic Functions in Authentication

Passing public key certificates and signatures used in authentication creates additional issues for protocols. When specifying cryptographic functions for a protocol, it is an agreement between the proposer and the chooser. When choosing cryptographic functions for public key certificates, however, the proposer and the chooser are beholden to functions used by the trusted third parties, the certification authorities (CAs). It doesn't really matter what either party wants the other party to use, since the other party is not the one issuing the certificates.

In this discussion, the term "certificate" does not necessarily mean a PKIX certificate. Instead, it means any message that binds an identity to a public key, where the message is signed by a trusted third party. This can be non-PKIX certificates or other types of cryptographic identity-binding structures that may be used in the future.

The question of specifying cryptographic functions is only relevant if one party has multiple certificates or signatures with different cryptographic functions. In this section, the terms "proposer" and "chooser" have a different meaning than in the previous section.

Here, both parties act as proposers of the identity they want to use and the certificates with which they are backing up that identity, and both parties are choosers of the other party's identity and certificate.

Some protocols allow the proposer to send multiple certificates or signatures, while other protocols only allow the proposer to send a single certificate or signature. Some protocols allow the proposer to send multiple certificates but advise against it, given that certificates can be fairly large (particularly when the CA loads the certificate with lots of information).

IKEv1 and IKEv2 allow both parties to list all the certificates that they want to use. [PKI4IPsec] proposes to restrict this by saying that all the certificates for a proposer have to have the same identity.

## 6. Suggested Changes

In investigating how protocols use hash functions, the IETF is looking at (at least) two areas of possible changes to individual protocols: how the IETF might need to change the protocols, and how implementors of current protocols might change what they do. This section describes both of these areas with respect to IKEv1, IKEv2, and IPsec.

### 6.1. Suggested Changes for the Protocols

Protocols might need to be changed if they rely on the collision-resistance of particular hash functions. They might also need to be changed if they do not allow for the agreement of hash functions because it is expected that the "preferred" hash function for different users will change over time.

IKEv1 and IKEv2 already allow for the agreement of hash functions for both IKE and IPsec, and thus do not need any protocol change.

IKEv1 and IKEv2, when used with public key authentication, already allow each party to send multiple PKIX certificates, and thus do not need any protocol change.

There are known weaknesses in PKIX with respect to collision-resistance of some hash functions. Because of this, it is hoped that there will be changes to PKIX fostered by the PKIX Working Group. Some of the changes to PKIX may be usable in IKEv1 and IKEv2 without having to change IKEv1 and IKEv2. Other changes to PKIX may require changes to IKEv1 and IKEv2 in order to incorporate them, but that will not be known until the changes to PKIX are finalized.

### 6.2. Suggested Changes for Implementors

As described in earlier sections, IKE and IPsec themselves are not susceptible to any known collision-reduction attacks on hash functions. Thus, implementors do not need to make changes such as prohibiting the use of MD5 or SHA-1. The mandatory and suggested algorithms for IKEv2 and IPsec are given in [IKEv2Algs] and [IPsecAlgs].

Note that some IKE and IPsec users will misunderstand the relevance of the known attacks and want to use "stronger" hash functions. Thus, implementors should strongly consider adding support for alternatives, particularly the AES-XCBC-PRF-128 [AES-PRF] and AES-XCBC-MAC-96 [AES-MAC] algorithms, as well as forthcoming algorithms based on the SHA-2 family [SHA2-HMAC].

Implementations of IKEv1 and IKEv2 that use PKIX certificates for authentication may be susceptible to attacks based on weaknesses in PKIX. It is widely expected that PKIX certificates in the future will use hash functions other than MD5 and SHA-1. Implementors of IKE that allow certificate authentication should strongly consider allowing the use of certificates that are signed with the SHA-256, SHA-384, and SHA-512 hash algorithms. Similarly, those implementors should also strongly consider allowing the sending of multiple certificates for identification.

## 7. Security Considerations

This entire document is about the security implications of reduced collision-resistance of common hash algorithms for the IKE and IPsec protocols.

The Security Considerations section of [HashAttacks] gives much more detail about the security of hash functions.

## 8. Informative References

[AES-MAC] Frankel, S. and H. Herbert, "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec", RFC 3566, September 2003.

[AES-PRF] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4434, February 2006.

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

[Deploying] Bellovin, S. and E. Rescorla, "Deploying a New Hash Algorithm", NDSS '06, February 2006.

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

[HashAttacks] Hoffman, P. and B. Schneier, "Attacks on Cryptographic Hashes in Internet Protocols", RFC 4270, November 2005.

[HMAC-reduction] Contini, S. and YL. Yin, "Forgery and Partial Key-Recovery Attacks on HMAC and NMAC Using Hash Collisions", Cryptology ePrint Report 2006/319, September 2006.

[IKEv1] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

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

[IKEv2Algs] Schiller, J., "Cryptographic Algorithms for use in the Internet Key Exchange Version 2", RFC 4307, December 2005.

[IPsecAlgs] Eastlake, D., "Cryptographic Algorithm Implementation Requirements For ESP And AH", RFC 4305, December 2005.

[NAT-T] Kivinen, T., Swander, B., Huttunen, A., and V. Volpe, "Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.

[PKI4IPsec] Korver, B., "The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX", Work in Progress, April 2007.

[SHA2-HMAC] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 With IPsec", RFC 4868, May 2007.

[Target-collisions] Stevens, M., Lenstra, A., and B. de Weger, "Target Collisions for MD5 and Colliding X.509 Certificates for Different Identities", Cryptology ePrint Report 2006/360, October 2006.

## Appendix A.  Acknowledgments

Tero Kivinen helped with ideas in the first version of this document. Many participants on the SAAG and IPsec mailing lists contributed ideas in later versions.  In particular, suggestions were made by Alfred Hoenes, Michael Richardson, Hugo Krawczyk, Steve Bellovin, David McGrew, Russ Housley, Arjen Lenstra, and Pasi Eronen.

## Author's Address

Paul Hoffman
VPN Consortium
127 Segre Place
Santa Cruz, CA  95060
US

EMail: paul.hoffman@vpnc.org

## Full Copyright Statement

## Intellectual Property

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.