hmac

HMAC: Keyed-Hashing for Message Authentication

Status of This Memo

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

Abstract

This document describes HMAC, a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function, e.g., MD5, SHA-1, in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

1. Introduction

Providing a way to check the integrity of information transmitted over or stored in an unreliable medium is a prime necessity in the world of open computing and communications. Mechanisms that provide such integrity check based on a secret key are usually called "message authentication codes" (MAC). Typically, message authentication codes are used between two parties that share a secret key in order to validate information transmitted between these parties. In this document we present such a MAC mechanism based on cryptographic hash functions. This mechanism, called HMAC, is based on work by the authors [BCK1] where the construction is presented and cryptographically analyzed. We refer to that work for the details on the rationale and security analysis of HMAC, and its comparison to other keyed-hash methods.

Given this, a naive example of MAC generation by the sender could be:

```
macCode = sha256('thisIsASecretKey1234' + 'my message here')
```

Then the verification by the receiver would be:

```
macCode == sha256('thisIsASecretKey1234' + 'my message here')
```

Note that MACs don't necessarily use a hash function, but a hash can be used as a "signing" mechanism. For a further reading look at the MAC Wikipedia article.

HMAC — Hash-Based Message Authentication Code

An HMAC is a kind of MAC. All HMACs are MACs but not all MACs are HMACs. The main difference is that an HMAC uses two rounds of hashing instead of one (or none). Each round of hashing uses a section of the secret key. As a naive example:

```
sha256('thisIsASe' + sha256('cretKey1234' + 'my message here'))
```

Which is a simplified version of the function given in RFC-2104.