Protocol

HashFunction

A type that performs cryptographically secure hashing.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 13.0+ | visionOS 1.0+ | watchOS 6.0+

```
@preconcurrency
protocol HashFunction : Sendable
```

Overview

The `HashFunction` protocol describes an interface for computing a fixed-length digest from an arbitrarily large collection of bytes. Because the digest is small, you can quickly compare the digests to detect a difference in two corresponding data sets. Alternatively, transmit or store data with its digest to detect changes introduced after initially calculating the digest.

Use one of the protocol's adopters, like `SHA256`, `SHA384`, or `SHA512`, to output a digest whose value varies significantly over even small differences in the input data.

Checking a digest doesn't guard against changes made by a malicious user who also changes the digest to match. To handle this, compute a message authentication code (MAC) like `HMAC` instead. MACs rely on hashing, but incorporate a secret cryptographic key into the digest computation. Only a user that has the key can generate a valid MAC.

Topics

Specifying the output type

`associatedtype Digest : Digest`

Required

`protocol Digest`

A type that represents the output of a hash.

Computing a hash in one call

`static func hash<D>(data: D) -> Self.Digest`

Computes the digest of the bytes in the given data instance and returns the computed digest.

Computing a hash iteratively

`init()`

Creates a hash function.

Required

`func update<D>(data: D)`

Incrementally updates the hash function with the given data.

`func update(bufferPointer: UnsafeRawBufferPointer)`

Incrementally updates the hash function with the contents of the buffer.

Required

`func finalize() -> Self.Digest`

Finalizes the hash function and returns the computed digest.

Required

Inspecting hash information

`static var blockByteCount: Int`

The number of bytes that represents the hash function's internal state.

Required

Relationships

Inherits From

Sendable, SendableMetatype

Conforming Types

Insecure.MD5
Insecure.SHA1
SHA256
SHA384
SHA3_256
SHA3_384
SHA3_512
SHA512

See Also

Cryptographically secure hashes

`struct` SHA512

An implementation of Secure Hashing Algorithm 2 (SHA-2) hashing with a 512-bit digest.

`struct` SHA384

An implementation of Secure Hashing Algorithm 2 (SHA-2) hashing with a 384-bit digest.

`struct` SHA256

An implementation of Secure Hashing Algorithm 2 (SHA-2) hashing with a 256-bit digest.