Structure

HKDF

A standards-based implementation of an HMAC-based Key Derivation Function (HKDF).

iOS 14.0+ | iPadOS 14.0+ | Mac Catalyst 14.0+ | macOS 11.0+ | tvOS 14.0+ | visionOS 1.0+ | watchOS 7.0+

```
struct HKDF<H> where H : HashFunction
```

Overview

The key derivation functions allow you to derive one or more secrets of the size of your choice from a main key or passcode. The key derivation function is compliant with IETF RFC 5869. Use one of the `deriveKey` functions, such as `deriveKey(inputKeyMaterial:outputByteCount:)` or `deriveKey(inputKeyMaterial:salt:info:outputByteCount:)`, to derive a key from a main secret or passcode in a single function.

To derive a key with more fine-grained control, use `extract(inputKeyMaterial:salt:)` to create cryptographically strong key material in the form of a hashed authentication code, then call `expand(pseudoRandomKey:info:outputByteCount:)` using that key material to generate a symmetric key of the length you specify.

Topics

Deriving a key

```
static func deriveKey(inputKeyMaterial: SymmetricKey, outputByteCount: Int) -> SymmetricKey
```

Derives a symmetric encryption key from a main key or passcode using HKDF key derivation.

```
static func deriveKey<Info>(inputKeyMaterial: SymmetricKey, info: Info, outputByteCount: Int) -> SymmetricKey
```

Derives a symmetric encryption key from a main key or passcode using HKDF key derivation with information you specify.

```
static func deriveKey<Salt>(inputKeyMaterial: SymmetricKey, salt: Salt, outputByteCount: Int) -> SymmetricKey
```

Derives a symmetric encryption key from a main key or passcode using HKDF key derivation with salt that you specify.

```
static func deriveKey<Salt, Info>(inputKeyMaterial: SymmetricKey, salt: Salt, info: Info, outputByteCount: Int) -> SymmetricKey
```

Derives a symmetric encryption key from a main key or passcode using HKDF key derivation with information and salt you specify.

Controlling key derivation

```
static func extract<Salt>(inputKeyMaterial: SymmetricKey, salt: Salt?) -> HashedAuthenticationCode<H>
```

Creates cryptographically strong key material from a main key or passcode that you specify.

```
static func expand<PRK, Info>(pseudoRandomKey: PRK, info: Info?, outputByteCount: Int) -> SymmetricKey
```

Expands cryptographically strong key material into a derived symmetric key.

Relationships

Conforms To

`Sendable`, `SendableMetatype`