

# SecItem attributes for keys

!

This thread has been locked. Questions are automatically locked after two months of inactivity, or sooner if deemed necessary by a moderator.



1.4k

I can never remember which key attributes perform which function, so I've created a summary and I'm posting it here for the benefit of Future Quinn™ (and possibly others :-).

- `kSecAttrLabel` ('`labl`')
  - `CFString`
  - Is **not** a component of key uniqueness
  - `kSecKeyPrintName` (1) in the legacy keychain on macOS
  - This is the user-visible description of the key; this is not particularly useful on iOS but super handy on macOS because it shows up in the *Name* column in Keychain Access
- `kSecAttrApplicationLabel` ('`klbl`')
  - `CFData` (but may be a `CFString` containing a UUID)
  - Is a component of key uniqueness
  - `kSecKeyLabel` (6) in the legacy keychain on macOS
  - Not user visible
  - For asymmetric keys this holds the public key hash which allows digital identity formation (to form a digital identity, this value must match the `kSecAttrPublicKeyHash` ('`pkhh`') attribute of the certificate)
- `kSecAttrApplicationTag` ('`atag`')
  - `CFData`
  - Is a component of key uniqueness
  - `kSecKeyApplicationTag` (7) in the legacy keychain on macOS
  - On macOS, this shows up in the *Comments* field in the info window in Keychain Access (accessed via *File* > *Get Info*)
  - The content of this is entirely up to the app

In the context of `kSecAttrApplicationLabel`, the public key hash is a SHA-1 digest of the bytes in the `subjectPublicKey` element of the `SubjectPublicKeyInfo` structure within the certificate (see Section 4.1 of [RFC 5280](#)). For example, with this key:

```
1 $ dumpasn1 -p key.asn1
2 SEQUENCE {
3   SEQUENCE {
4     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
5     OBJECT IDENTIFIER prime256v1 (1 2 840 10045 3 1 7)
6   }
7   BIT STRING
8   04 07 88 9B 18 EF 92 93 6C B1 04 7F F3 81 A4 31
9   0A 08 1A CE 9D E7 13 B4 B9 5A E9 04 0C 10 A2 02
10  C0 0D 78 71 01 21 EE 57 C4 40 C3 86 AE 05 25 F3
11  31 96 49 C9 28 31 10 A8 B9 0A 57 E1 E3 36 2C 3D
12  F7
13 }
```

the digest covers the bytes 04 07 88 9B ... 36 2C 3D F7.

**Note** `dumpasn1` is a third-party tool for inspecting ASN.1 files. Download its source code from [www.cs.auckland.ac.nz/~pgut001/dumpasn1.c](#).

Share and Enjoy

—

Quinn “The Eskimo!” @ Developer Technical Support @ Apple

```
let myEmail = "eskimo" + "1" + "@" + "apple.com"
```

Changes:

- 17 Jan 2017 — First posted.
- 13 Mar 2019 — Added a note about the public key hash. Made minor editorial changes.
- 18 Mar 2019 — Added a download link for `dumpasn1`. Made minor typographical changes.
- 1 Apr 2021 — Fixed the formatting.

Security

Asked 4 years ago by eskimo

Reply to this question

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the [Apple Developer Forums Participation Agreement](#).

Discover	Design	Develop	Distribute	Support
macOS	Human Interface Guidelines	Xcode	Developer Program	Articles
iOS	Resources	Swift	App Store	Developer Forums
watchOS	Videos	Swift Playgrounds	App Review	Feedback & Bug Reporting
tvOS	Apple Design Awards	TestFlight	Mac Software	System Status
Safari and Web	Fonts	Documentation	Apps for Business	Contact Us
Games	Accessibility	Videos	Safari Extensions	<b>Account</b>
Business	Internationalization	Downloads	Marketing Resources	Certificates, Identifiers & Profiles
Education	Accessories		Trademark Licensing	App Store Connect
WWDC				