iCloud Keychain
and
iOS 7 Data Protection

Andrey Belenko
Sr. Security Engineer @ viaForensics

iOS Data Protection

Keychain encryption since the very beginning

- Before iOS 4: AES-CBC, per-device key
- iOS 4: AES-CBC, per-record key based on desired item accessibility
- iOS 5+: AES-GCM, per-record key, encrypts metadata

iOS Data Protection

Storage encryption since iOS 4*

- File-level (much like Windows EFS), so carving is challenging
- Per-file key based on protection class: NSFileProtectionNone or NSFileProtectionComplete
- iOS 5 adds …CompleteUntilFirstUserAuthentication and … CompleteUnlessOpen (uses DJB's curve25519)

* Pre-iOS 4 encryption was used to wipe data

iOS 7?

- No visible changes to file encryption (i.e. existing tools work and don’t screw things up)
- Keychain record format has changed
    - ASN.1 BER encoding instead of proprietary Binary Property List encoding
- Keychain encryption has not changed: AES-GCM with per-record key

ASN.1?

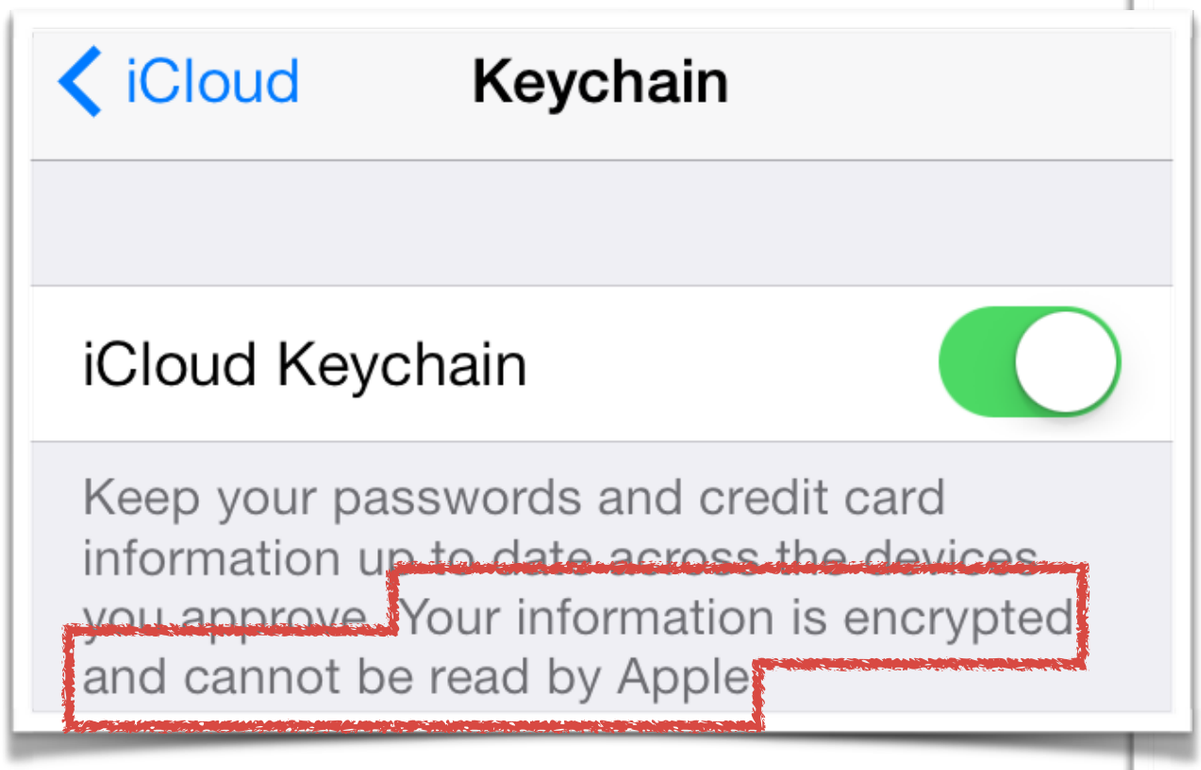
Apple switching from something proprietary to something standard? Why?

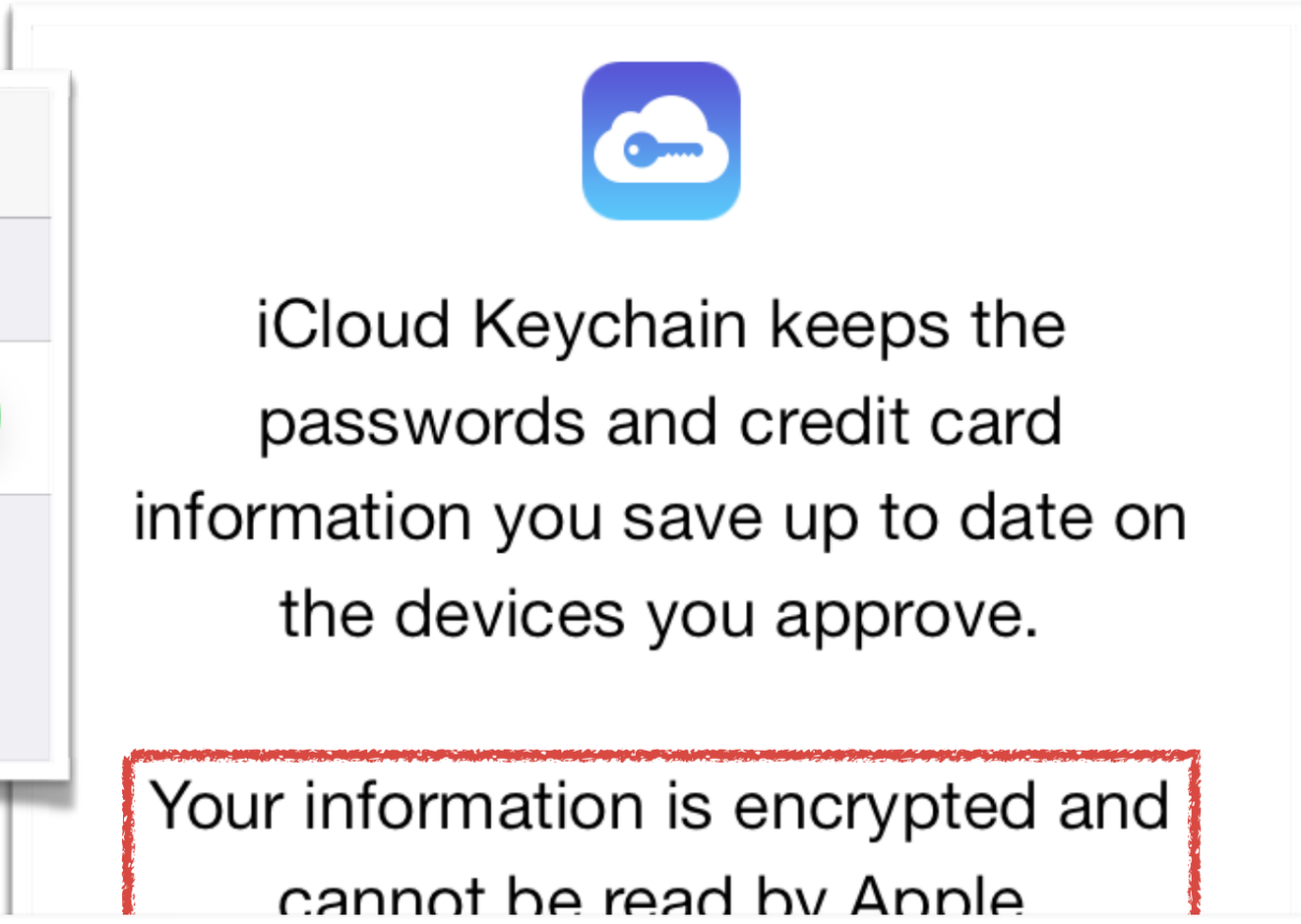
- ASN.1 BER is sequential: no need to read TOC from the end of the stream first
- ASN.1 BER is more compact
- ASN.1 is cross-platform: will we see Keychain on other platforms?

iCloud Keychain

Image: Apple Inc.

Motivation





iCloud Keychain

- iCloud Keychain encryption keys are created on your devices, and Apple can't access those keys. Only encrypted keychain data passes through Apple's servers, and Apple can't access any of the key material that could be used to decrypt that data.

http://support.apple.com/kb/HT4865

Motivation

iCloud Keych

iCloud Keychain

Keep your passwo
information up to d
you approve. Your i
and cannot be read

keeps the
redit card
up to date on
approve.

encrypted and
ad by Apple

iCloud Keychain

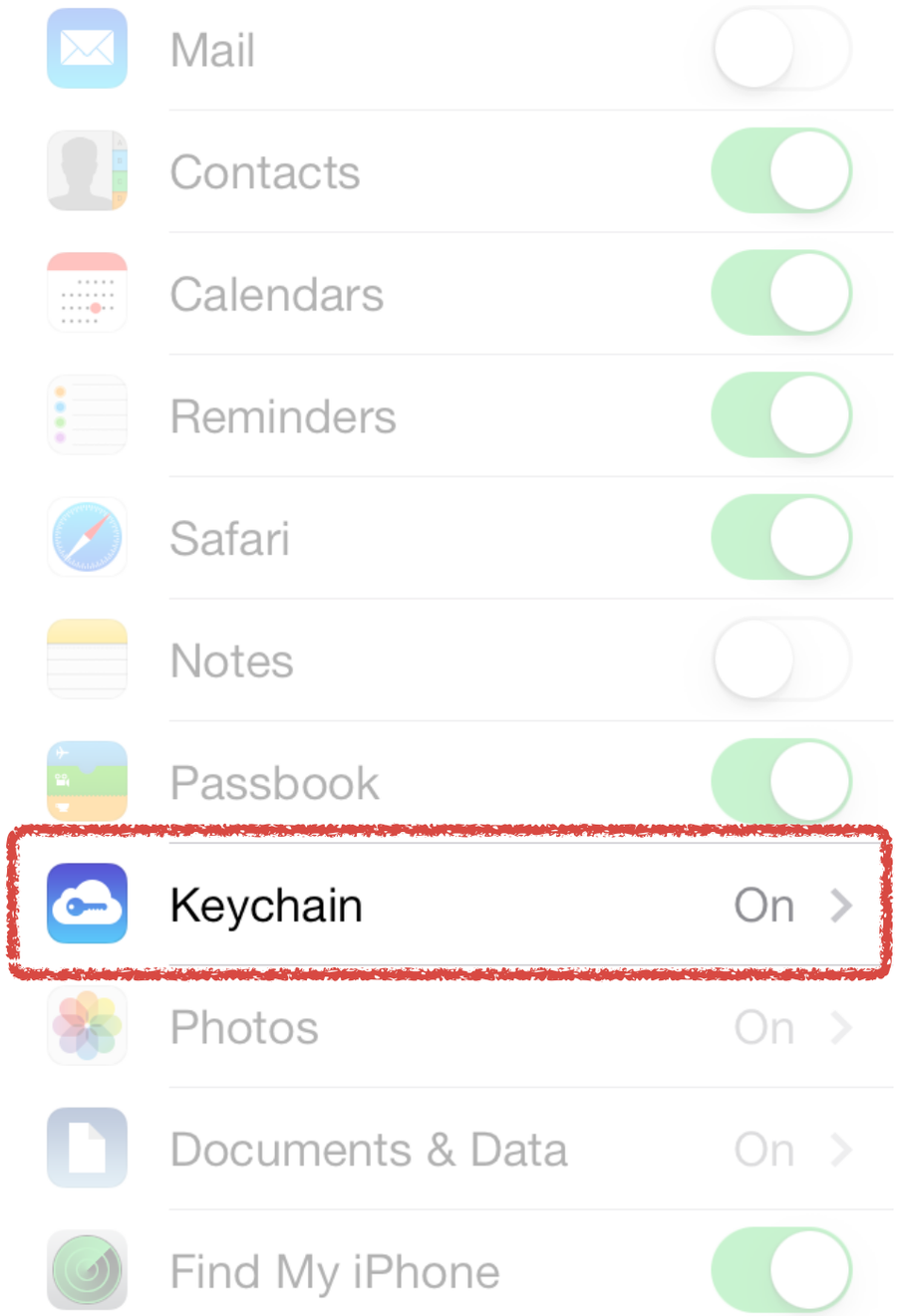
- iCloud Keychain encryption keys a
  encrypted keychain data passes through.
  could be used to decrypt that data.

ple can't access those keys. Only
pple can't access any of the key material that

http://support.apple.com/kb/HT4865

iCloud

```
<key>com.apple.mobileme</key>
<dict>
    <key>availableFeatures</key>
    <array>
        <string>com.apple.Dataclass.Reminders</string>
        <string>com.apple.Dataclass.DeviceLocator</string>
        <string>com.apple.Dataclass.KeychainSync</string>
        <string>com.apple.Dataclass.Content</string>
        <string>com.apple.Dataclass.Mail</string>
        <string>com.apple.Dataclass.KeyValue</string>
        <string>com.apple.Dataclass.Notes</string>
        <string>com.apple.Dataclass.Ubiquity</string>
        <string>com.apple.Dataclass.MediaStream</string>
        <string>com.apple.Dataclass.Contacts</string>
        <string>com.apple.Dataclass.Bookmarks</string>
        <string>com.apple.Dataclass.Calendars</string>
        <string>com.apple.Dataclass.Backup</string>
        <string>com.apple.Dataclass.SharedStreams</string>
</array>
```



```
<key>com.apple.Dataclass.KeychainSync</key>
<dict>
    <key>authMechanism</key>
    <string>token</string>

    <key>escrowProxyUrl</key>
    <string>https://p19-escrowproxy.icloud.com:443</string>

</dict>
```

```
<key>com.apple.Dataclass.KeyValue</key>
<dict>
    <key>beta</key>
    <false/>

    <key>authMechanism</key>
    <string>token</string>

    <key>apsEnv</key>
    <string>production</string>

    <key>configURL</key>
    <string>https://keyvalueservice.icloud.com:443/config</string>

    <key>url</key>
    <string>https://p19-keyvalueservice.icloud.com:443</string>

</dict>
```

The Big Picture

Keychain (encrypted)
Keybag (encrypted)

*.keyvalueservice.icloud.com

HTTPS
NO PINNING

Some Secret

*.escrowproxy.icloud.com

Setup Options

Cancel **Keychain Setup**

Your iCloud Security Code can be used to set up iCloud Keychain on a new device.

– – – –

Advanced Options

Back Next

Advanced Security Code Options

Use a Complex Security Code

Get a Random Security Code

Don’t Create Security Code

If you don’t create a security code, setting up iCloud Keychain on a new device will require your approval from a different device.

4-digit iCSC [Default]

iCloud Security Code

1234

PBKDF2
SHA-256 x 10'000

Random Password

BL7Z-EBTJ-UBKD-X7NM-4W6D-J2N4

AES-CBC
256 bit

*.escrowproxy.icloud.com

AES-Wrap Keys
RFC 3394

*.keyvalueservice.icloud.com

Backup Keybag

Key 1
Key 2
Key 3

AES-GCM
256 bit

Keychain Passwords

yMa9ohCJ
tzzcVhE7
sDVoCnb

Key-Value Store

- Not new
- Used extensively by many apps e.g. to keep preferences in sync across devices
- iCloud Keychain utilises two stores:
  - `com.apple.security.cloudkeychainproxy3`
  - `com.apple.sbd3 (securebackupd3)`

Key-Value Store

Key	Description
`com.apple.securebackup.enabled`	Is Keychain data saved in KVS?
`com.apple.securebackup.record`	Keychain records, encrypted
`SecureBackupMetadata`	iCSC complexity, timestamp, country
`BackupKeybag`	Keybag protecting Keychain records
`BackupUsesEscrow`	Is keybag password escrowed?
`BackupVersion`	Version, currently @“1”
`BackupUUID`	UUID of the backup

Escrow Proxy

- Designed to store precious secrets
- Access to service requires auth token
- Access to escrowed data requires iCSC
  - Need to receive SMS challenge
  - Must successfully complete SRP auth
- User-Agent: `com.apple.lakitu` (iOS/OS X)

Image: mariowiki.com

Secure Remote Password

- Zero-knowledge password proof scheme
- Combats sniffing/MITM
- One password guess per connection attempt
- Password verifier is not sufficient for impersonation
- Escrow Proxy uses SRP-6a

Agreed-upon parameters:

H – one-way hash function
N, g – group parameters
k ← H(N, g)

Password verifier:

SALT ← random
x ← H(SALT,Password)
v ← g^x

Key Negotiation

a ← random, A ← g^a	ID, A →	
	← SALT, B	b ← random, B ← kv + g^b
u ← H(A, B)		u ← H(A, B)
x ← H(SALT, Password) S ← (B - kg^x) ^ (a + ux) K ← H(S)		S ← (Av^u) ^ b K ← H(S)

Key Verification

M ← H(H(N) ⊕ H(g), H(ID), SALT, A, B, K)

M →	(Aborts if M is invalid)
← H(A, M, K)	

Agreed-upon parameters:

H – SHA-256
N, g – RFC 5054 w. 2048-bit group
k ← H(N, g)

Password verifier:

SALT ← random
x ← H(SALT,Password)
v ← g^x

Key Negotiation

a ← random, A ← g^a	ID, A, SMS CODE →	
	← SALT, B	b ← random, B ← kv + g^b
u ← H(A, B)		u ← H(A, B)
x ← H(SALT, Password) S ← (B - kg^x) ^ (a + ux) K ← H(S)		S ← (Av^u) ^ b K ← H(S)

Key Verification

M ← H(H(N) ⊕ H(g), H(ID), SALT, A, B, K)

M, SMS CODE →	(Aborts if M is invalid)
← H(A, M, K)	

Escrowed Data Recovery

/get_records

List of escrowed records

/get_sms_targets

List of phone numbers*

/generate_sms_challenge

OK

/srp_init [DsID, A, SMS CODE]

[UUID, DsID, SALT, B]

/recover [UUID, DsID, M, SMS CODE]

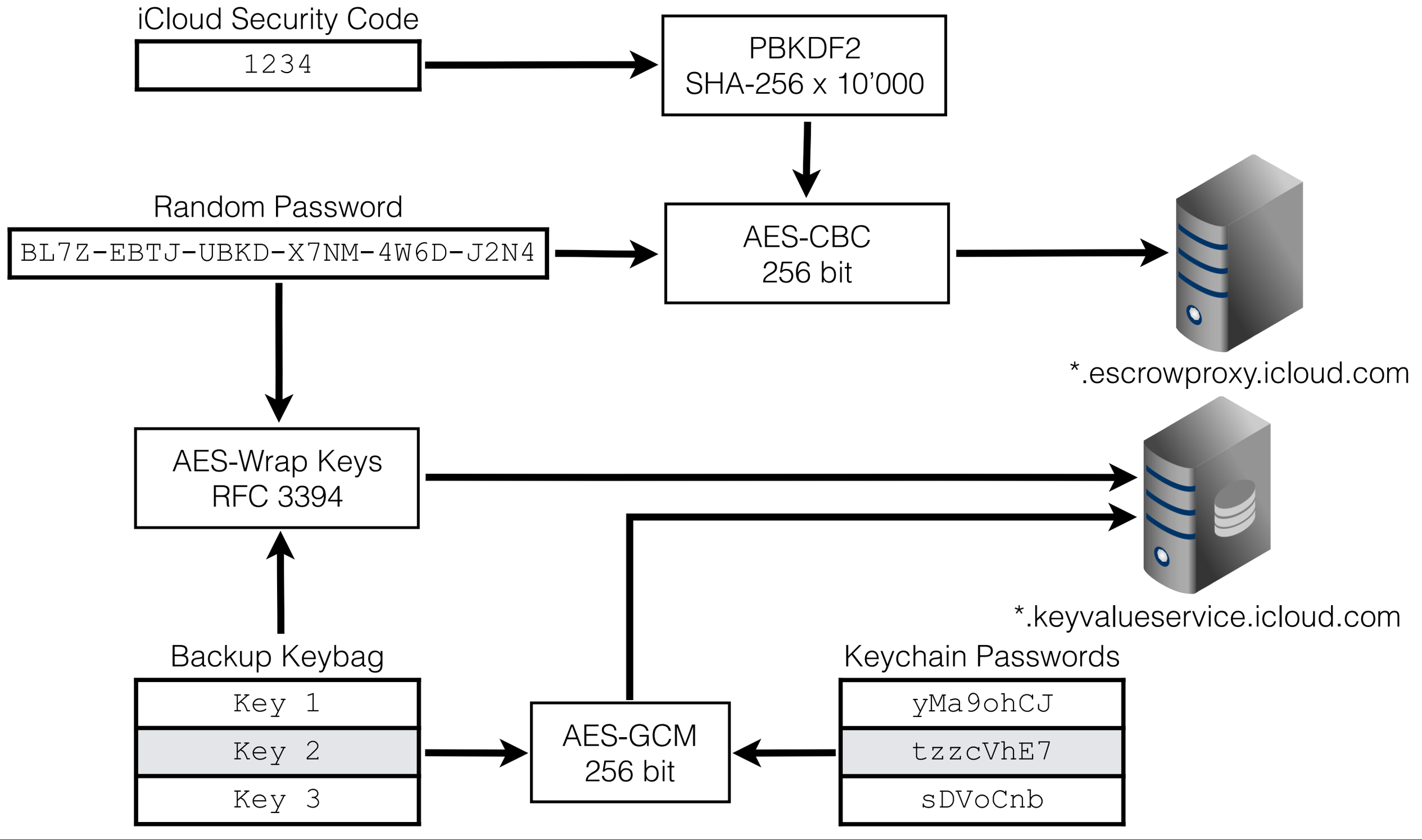
[IV, AES-CBC(K_{SRP}, Escrowed Record)]

*Display purposes only

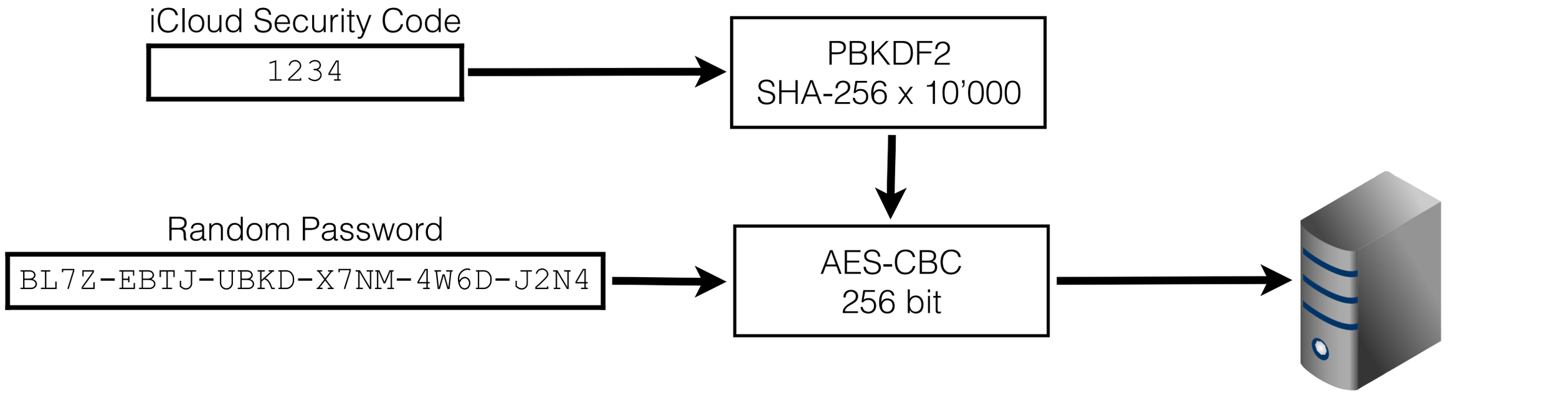
Escrow Proxy Endpoints

Endpoint	Description
`get_club_cert`	**[?]** Obtain certificate
`enroll`	Submit escrow record
`get_records`	List escrowed records
`get_sms_targets`	List SMS numbers for escrowed records
`generate_sms_challenge`	Generate and send challenge code
`srp_init`	First step of SRP protocol
`recover`	Second step of SRP protocol
`alter_sms_target`	Change SMS number

Escrow Record



Escrow Record




```
Key ← PBKDF2-SHA256(iCSC, 10’000)

EscrowRecord ← AES-CBC(Key, RandomPassword)
```

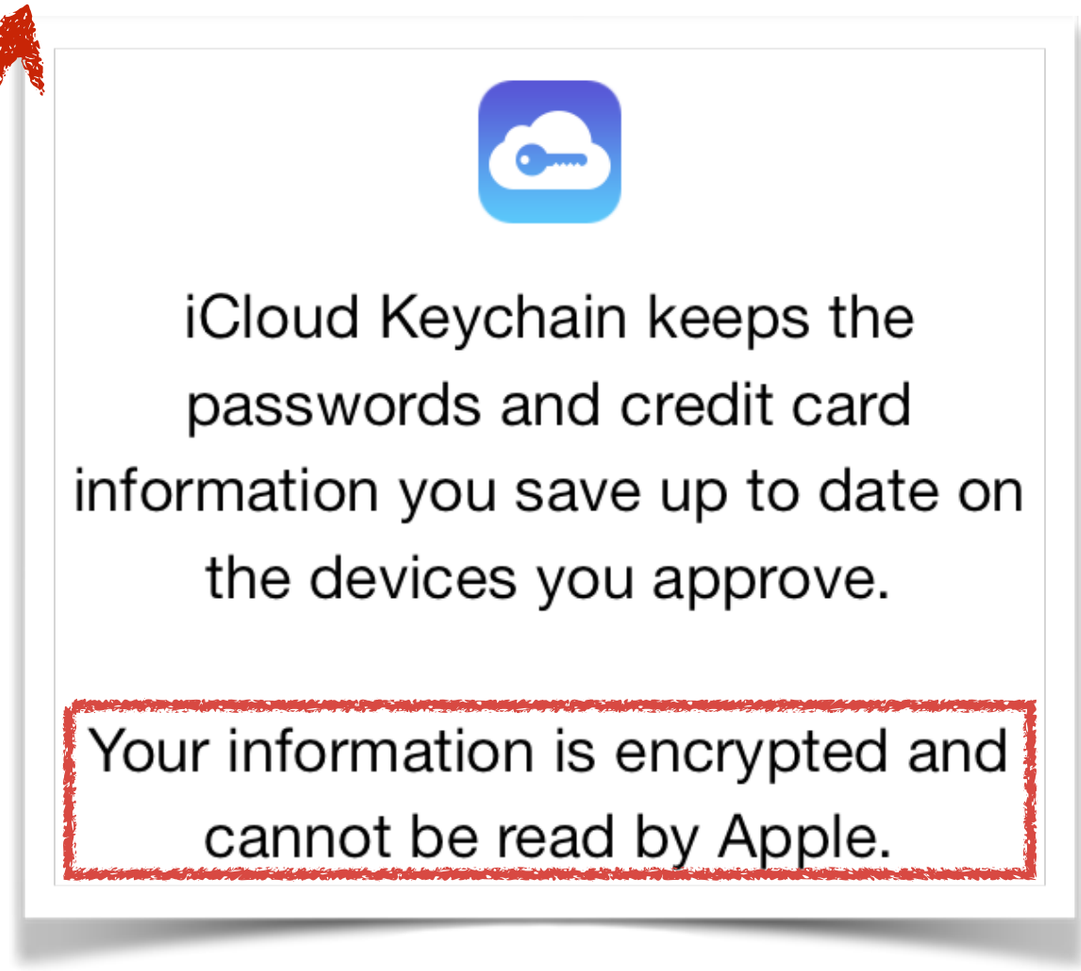
Escrow Record

```
Key ← PBKDF2-SHA256(iCSC, 10’000)

EscrowRecord ← AES-CBC(Key, RandomPassword)
```

- This is stored by Apple
- iCSC is <u>4 digits by default</u>

Can you spot the problem yet?



Escrow Record

`Key ← PBKDF2-SHA256(iCSC, 10’000)`

- Offline iCSC guessing is possible
- Almost instant recovery [for default settings]
- iCSC decrypts keybag password
- Keybag password unlocks keybag keys
- Keybag keys decrypt Keychain items

Apple, or other adversary with similar access level, can near-instantly decrypt “master” password and read synced iCloud Keychain records

(for default settings)

Setup Options

Cancel **Keychain Setup**

Your iCloud Security Code can be used to set up iCloud Keychain on a new device.

– – – –

Advanced Options

Back Next

Advanced Security Code Options

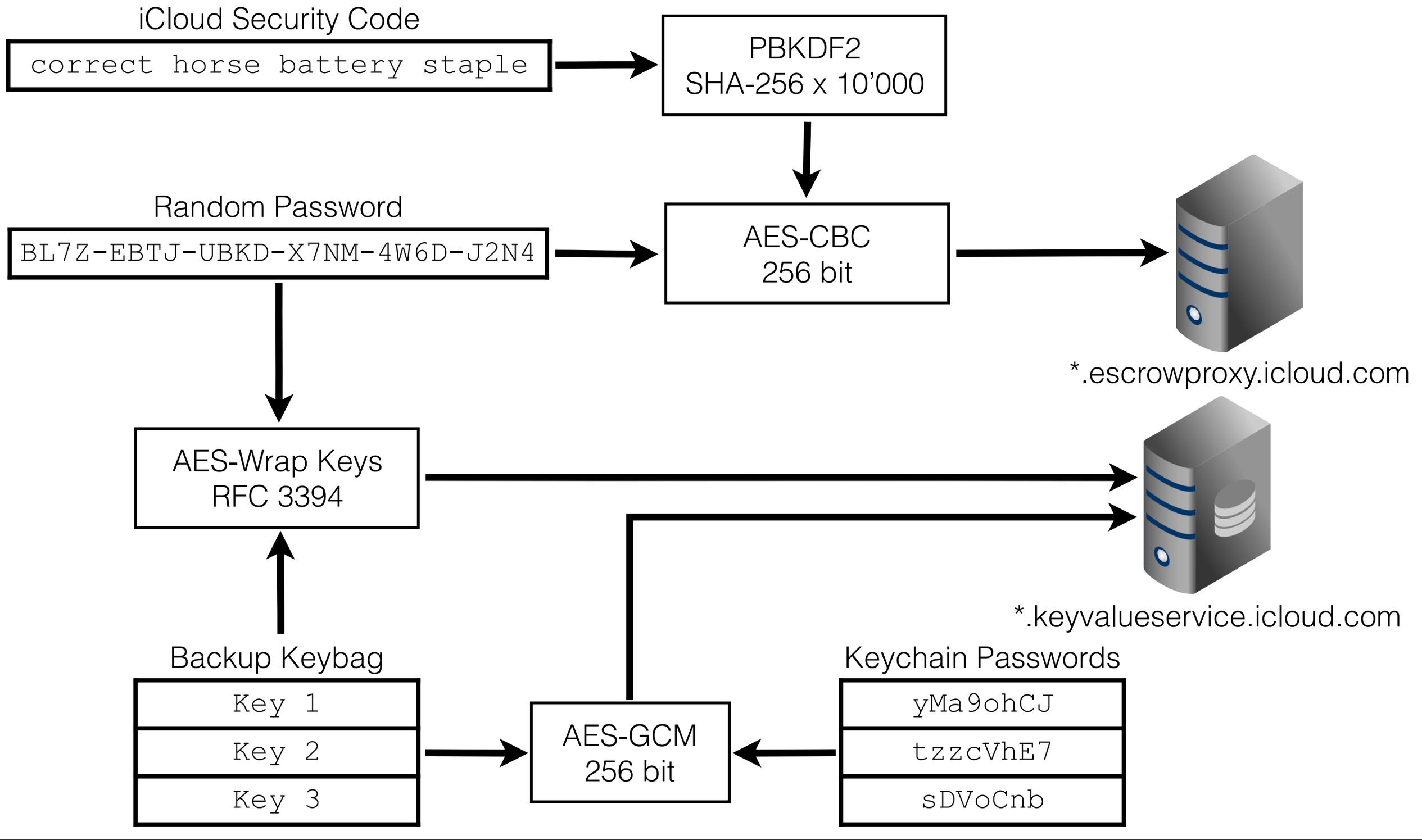
Use a Complex Security Code

Get a Random Security Code

Don’t Create Security Code

If you don’t create a security code, setting up iCloud Keychain on a new device will require your approval from a different device.

Complex iCSC



Complex iCSC

- Mechanics are the same as with simple iCSC
- Offline password recovery attack is still possible, although pointless if password is complex enough

Setup Options

Cancel **Keychain Setup**

Your iCloud Security Code can be used to set up iCloud Keychain on a new device.

– – – –

Advanced Options

Back Next

Advanced Security Code Options

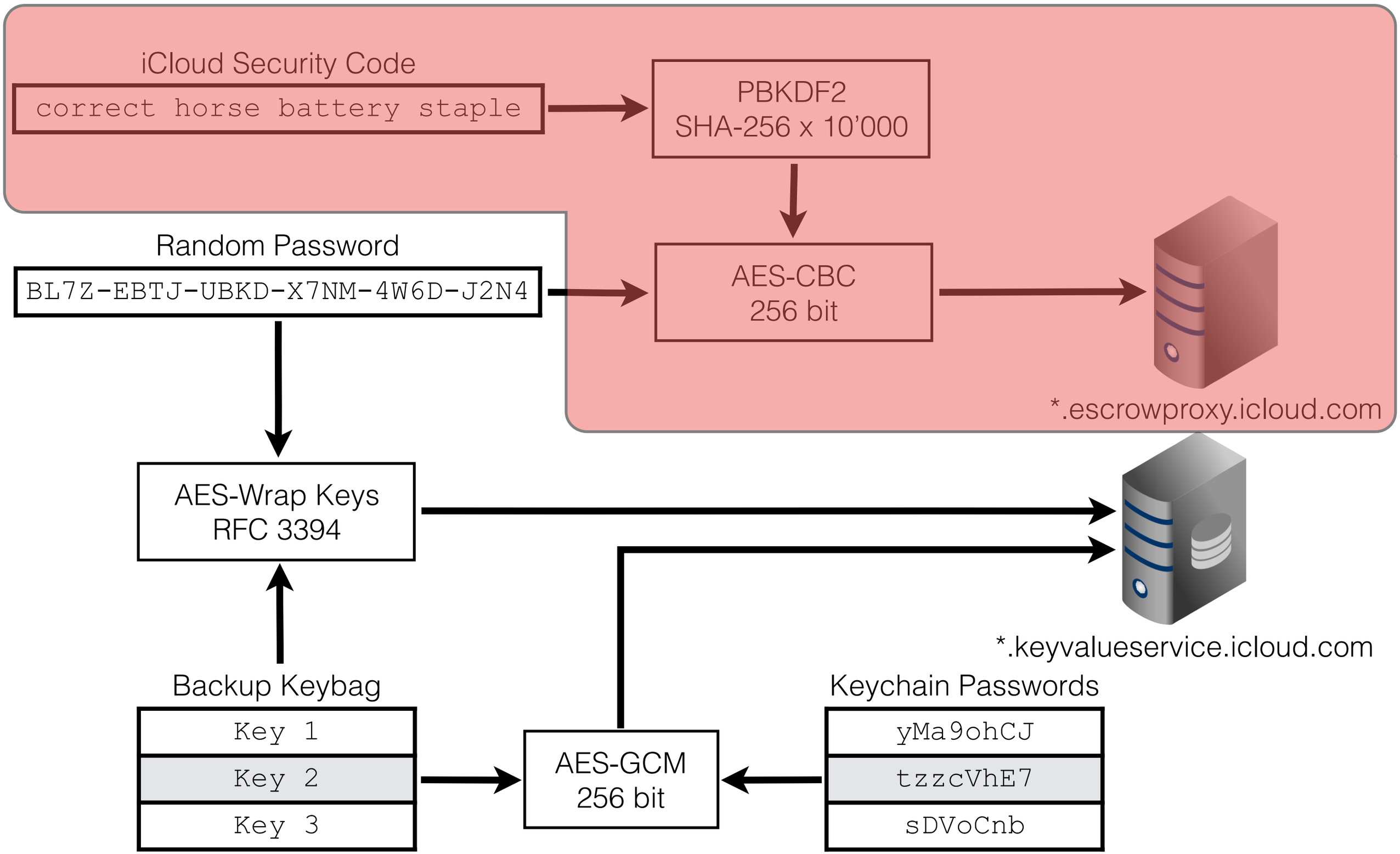
Use a Complex Security Code

Get a Random Security Code

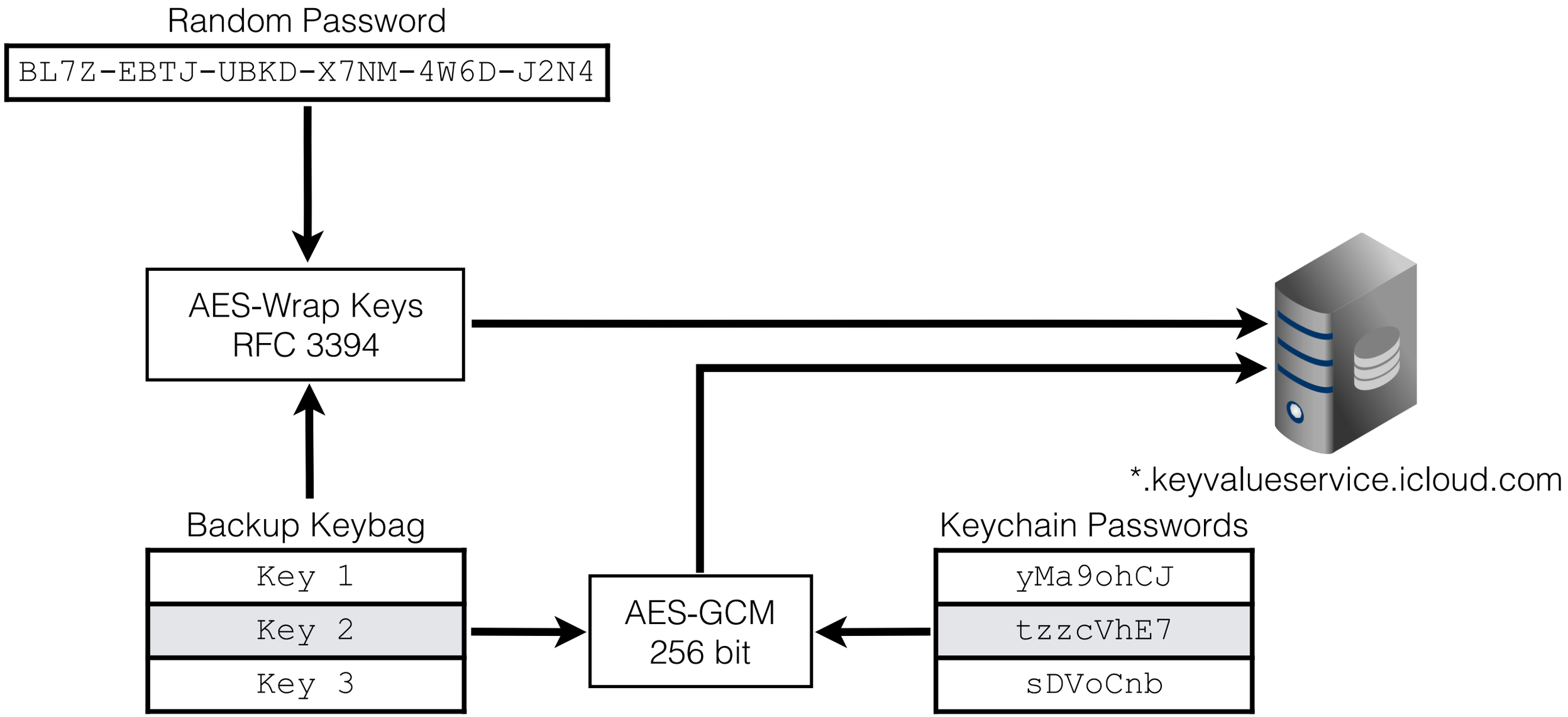
Don’t Create Security Code

If you don’t create a security code, setting up iCloud Keychain on a new device will require your approval from a different device.

Random iCSC


iCloud Security Code
correct horse battery staple
PBKDF2
SHA-256 x 10'000
Random Password
BL7Z-EBTJ-UBKD-X7NM-4W6D-J2N4
AES-CBC
256 bit
*.escrowproxy.icloud.com
AES-Wrap Keys
RFC 3394
*.keyvalueservice.icloud.com
Backup Keybag
Key 1
Key 2
Key 3
AES-GCM
256 bit
Keychain Passwords
yMa9ohCJ
tzzcVhE7
sDVoCnb

Random iCSC


Random Password
BL7Z-EBTJ-UBKD-X7NM-4W6D-J2N4
AES-Wrap Keys
RFC 3394
*.keyvalueservice.icloud.com
Backup Keybag
Key 1
Key 2
Key 3
AES-GCM
256 bit
Keychain Passwords
yMa9ohCJ
tzzcVhE7
sDVoCnb

Random iCSC

- Escrow Proxy is not used
- Random iCSC (or derived key) stored on the device [haven’t verified]

Setup Options

Cancel **Keychain Setup**

Your iCloud Security Code can be used to set up iCloud Keychain on a new device.

– – – –

Advanced Options

Back Next

Advanced Security Code Options

Use a Complex Security Code

Get a Random Security Code

Don’t Create Security Code

If you don’t create a security code, setting up iCloud Keychain on a new device will require your approval from a different device.

No iCSC

Work in Progress

Conclusions

Image: Apple Inc.

Conclusions

- Trust your vendor but verify his claims
- Never ever use simple iCloud Security Code
- Do not think that SMS Apple sends you is a 2FA
- Yet, iCK is reasonably well engineered although not without shortcomings

Thank You!

Questions are welcome :-)

@abelenko

ABelenko@viaforenscs.com

VIAFORENSICS
advancing mobile security

passwords