

Advanced smartphone forensics

Apple iCloud: backups, FindMyPhone, document storage, keychain

Windows Live: SMS, mail, notes

BlackBerry 10 backup encryption

Vladimir Katalov, ElcomSoft Co. Ltd



Smartphone forensics methods

| | iOS | WP8 | BB 10 |
|----------------------|------------------------|-------------------------|-------|
| Logical acquisition | Yes ^(*) | No | No |
| Advanced logical | Yes | No | No |
| Physical acquisition | Yes/No ^(**) | No | No |
| Chip-off | No | ? | ? |
| Local backup | Yes | No | Yes |
| Cloud backup | Yes | Yes/No ^(***) | No |
| iCloud keychain | Yes | N/A | N/A |
| Documents in cloud | Yes | Yes | No |
| Location service | Yes | Yes | No |

() In fact, same as local backup*

*(**) For A5+ devices, subject of jailbreak availability*

*(***) No complete backup, only selected categories*

iOS forensics

- **Logical acquisition**
 - “Ask” device to produce backup
 - Device must be unlocked(*)
 - Device may produce encrypted backup
 - Limited amount of information (but more than you think)
- **Physical acquisition**
 - Boot-time exploit to run unsigned code or jailbreak
 - Device lock state isn't relevant, can bruteforce passcode
 - Can get all information from the device (incl. deleted data)
- **Advanced logical acquisition**
 - By direct access to some services running on the iPhone
 - Device must be unlocked*
 - Limited amount of information (some as in local backup, but plus something extra)
 - Backup password isn't relevant
 - Can be performed over Wi-Fi
- **iCloud**
 - Need Apple ID and password
 - Can be performed without the device itself
 - Almost the same information as in local backup
 - Can get the documents and location data, too

* *But there is a workaround ;)*

Backups – what and when

- *Contacts and Contact Favorites*
- *Messages (including iMessages)*
- *Call history*
- *Application data*
- *Device settings*
- *Camera roll (photos and videos)*
- *Purchases (music, movies, TV, apps, books)*
- *Mail accounts*
- *Network settings (saved Wi-Fi hotspots, VPN settings etc)*
- *Paired Bluetooth devices*
- *Safari bookmarks, cookies, history*
- *... and much more*



★ Local backups

- iTunes create backups when:
 - Sync with iTunes
 - [File] | [Devices backup]

★ iCloud backups


- Backup runs daily when device is:
 - Connected to the Internet over Wi-Fi
 - Connected to a power source
 - Locked
- Can force backup
 - [Settings] | [iCloud] | [Storage & Backup] | [Back Up Now]

Backups
iBackupBot

Vladimir's iPhone 5 (Tue Oct 22 09:59:11 2013)

- System Files
 - CameraRollDomain
 - DatabaseDomain
 - HomeDomain
 - Library
 - Accounts
 - AddressBook
 - Application Support
 - BackBoard
 - BulletinBoard
 - Caches
 - Calendar
 - ConfigurationProfiles
 - Cookies
 - DataAccess
 - Keyboard
 - Mail
 - Maps
 - MobileInstallation
 - MusicLibrary
 - Notes
 - Passes
 - Preferences
 - SMS
 - Safari
 - SpringBoard
 - TCC
 - Voicemail
 - WebClips
 - YouTube
 - com.apple.itunesstored
 - KeychainDomain
 - ManagedPreferencesDomain
 - MediaDomain

iPhone 5 Global



Vladimir's iPhone 5

iOS 7.0.2 (build 11A501)

Phone Number: +7 (985) 998-68-20

Serial Number: C39JT465F39D

Unique Identifier: b026b06988940617008c9f81dd9e6ab13e07bbcc

IMEI: 013420001267660

What's In Backup

(NOTE: Before making any changes please select Duplicate from the File menu to make a copy of the backup)

Backup Date: Tue Oct 22 09:59:11 2013

Backup Location:
/Users/vkatalov/Downloads/b026b06988940617008c9f81dd9e6ab13e07bbcc

Total: 8677 Files, 13.4 GB

System Files: [4698 Files, 12.8 GB](#)

User Information:

- [Contacts](#)
- [Messages](#)
- [Call History](#)
- [Calendars](#)
- [Notes](#)
- [Recent Email Address](#)
- [Safari Bookmarks](#)
- [Safari History](#)

Multimedia Files:

- [Camera Roll](#)
- [Voice Mails](#)
- [Voice Memos](#)
- [Other Multimedia Files](#)

User App: [134 Apps, 3979 Data Files, 578.5 MB](#)

Devices
Welcome to iBackupBot

Total: 1 Selected: 0 Selected Size: 0

Wait, there is more...

- **Google Apps data: Search, Maps, YouTube etc**

AppDomain-com.google.*

- **Social networking & communications**

AppDomain-net.whatsapp.WhatsApp*

AppDomain-com.burbn.instagram*

AppDomain-com.facebook.Facebook*

AppDomain-com.facebook.Messenger*

AppDomain-com.skype.skype*

AppDomain-com.atebits.Tweetie2*

AppDomain-com.linkedin.Linkedin*

AppDomain-com.naveenium.foursquare*

AppDomain-com.viber*

- **Other**

HomeDomain\Library\Keyboard*

HomeDomain\Library\Passes*

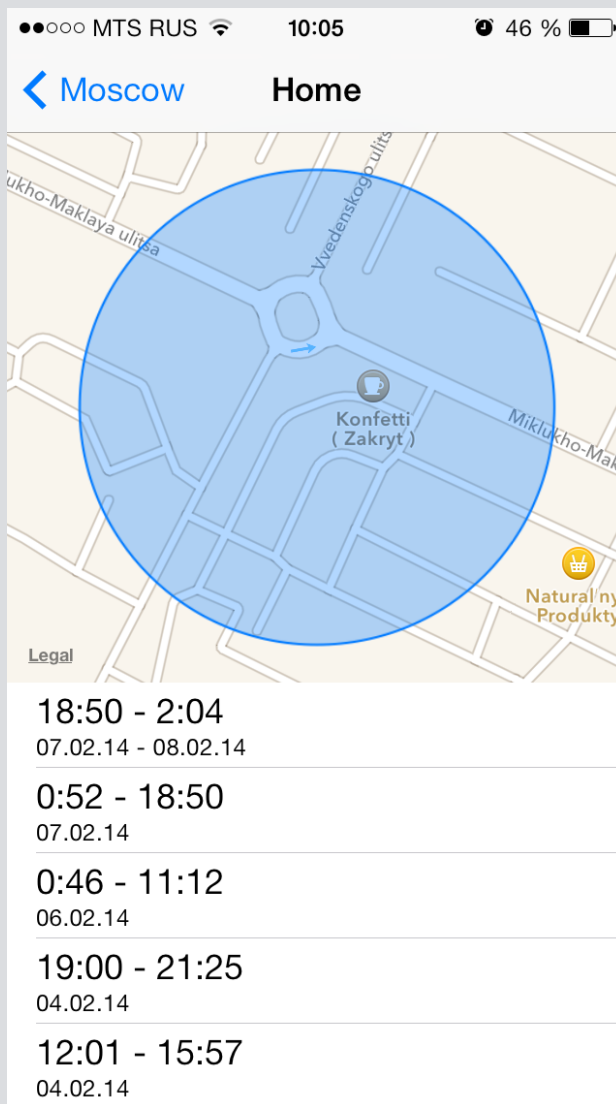
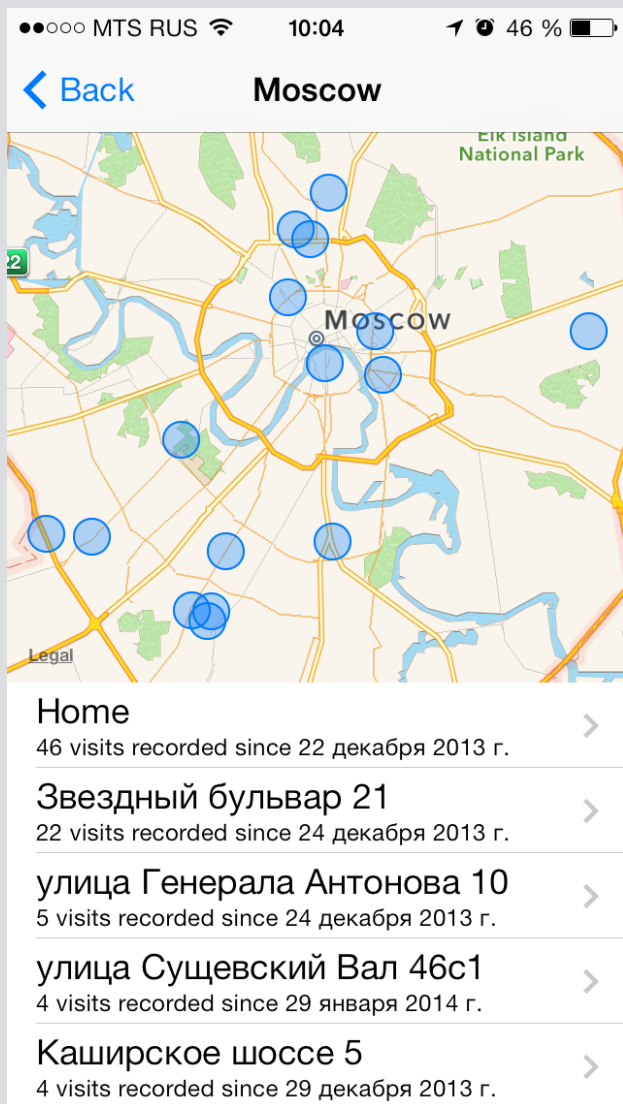
HomeDomain\Library\Voicemail*

HomeDomain\Library\Maps*



- Message attachments (even from deleted conversations!)
- Pictures from twitter posts
- Last backup date & time
- Info on Wi-Fi access points you ever connected to (SSID, security, signal etc)
- ... a lot of other interesting stuff :)

Frequent locations



Improve Maps **Done**

If you choose to enable Improve Maps, Apple will collect the GPS coordinates obtained through the Frequent Locations feature on your device and correlate them with the street address associated with your Apple ID. This will enable Apple to better approximate the geographic location of that and other addresses. Apple will only retain the resulting coordinates in an anonymous form to improve Maps and other Apple location-based products and services. You can turn off Improve Maps or Frequent Locations at any time under Settings > Privacy > Location Services > System Services > Frequent Locations.

Location services

SQLite Expert Personal 3.5.0.2410

File Database Table View SQL Transaction Tools Help

Database: cache_encryptedA Table: main.CellLocation File: D:\location\cache_encryptedA.db SQLite Library: [internal] version 3.8.1

Database SQL Data Design DDL

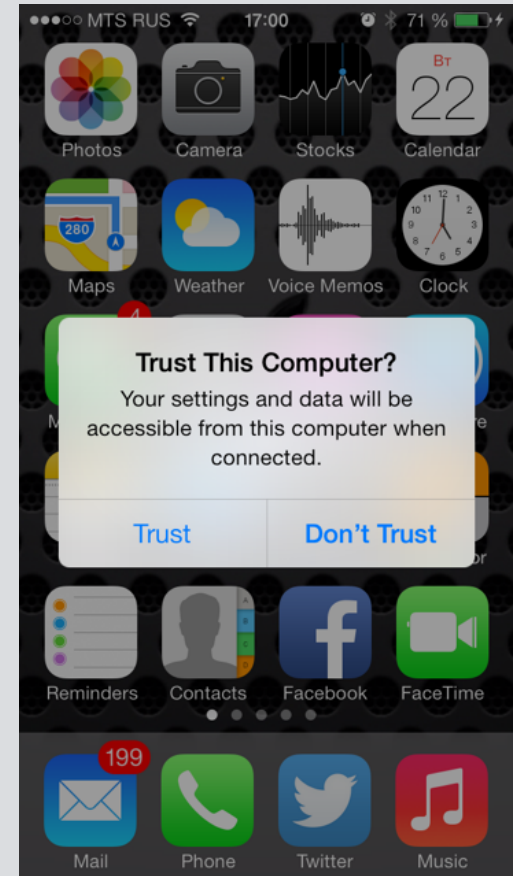
Refresh

| RecNo | MCC | MNC | LAC | CI | U.. | P.. | Timestamp | Latitude | Longitude | HorizontalAccuracy | Alt |
|-------|-----|-----|-------|----------|-----|-----|------------------|-------------|-------------|--------------------|-----|
| 1 | 250 | 2 | 7753 | 10295262 | -1 | -1 | 411552390.649334 | 55.63924331 | 37.5346787 | 1414 | |
| 2 | 250 | 2 | 7701 | 10534644 | -1 | -1 | 411569213.988531 | 55.97641917 | 37.40668065 | 1869 | |
| 3 | 250 | 1 | 531 | 6648044 | -1 | -1 | 411569213.988531 | 55.97718005 | 37.40913832 | 1414 | |
| 4 | 250 | 2 | 7701 | 9905516 | -1 | -1 | 411569213.988531 | 55.98150714 | 37.40640527 | 1847 | |
| 5 | 250 | 2 | 7710 | 9905516 | -1 | -1 | 411569213.988531 | 55.98641572 | 37.40042053 | 1417 | |
| 6 | 250 | 2 | 5099 | 9915948 | -1 | -1 | 411569213.988531 | 55.98036073 | 37.41394693 | 1451 | |
| 7 | 250 | 99 | 27392 | 16409562 | -1 | -1 | 411569213.988531 | 55.98164837 | 37.41057585 | 1414 | |
| 8 | 250 | 1 | 6355 | 6674244 | -1 | -1 | 411569213.988531 | 55.97435053 | 37.41451161 | 1414 | |
| 9 | 250 | 1 | 6355 | 6667103 | -1 | -1 | 411569213.988531 | 55.98595007 | 37.39681366 | 1464 | |
| 10 | 250 | 99 | 27392 | 25562 | -1 | -1 | 411569213.988531 | 55.98254984 | 37.41435664 | 1414 | |
| 11 | 250 | 99 | 27392 | 3122 | -1 | -1 | 411569213.988531 | 55.98132423 | 37.4073644 | 1414 | |
| 12 | 250 | 99 | 27392 | 16404996 | -1 | -1 | 411569213.988531 | 55.97895022 | 37.40839352 | 1414 | |
| 13 | 250 | 2 | 7701 | 9903235 | -1 | -1 | 411569213.988531 | 55.9828584 | 37.40648087 | 1414 | |
| 14 | 250 | 2 | 7733 | 9905516 | -1 | -1 | 411569213.988531 | 55.97988936 | 37.40808087 | 1414 | |
| 15 | 250 | 99 | 27392 | 16426 | -1 | -1 | 411569213.988531 | 55.97958107 | 37.41297526 | 1414 | |
| 16 | 250 | 1 | 6355 | 6648044 | -1 | -1 | 411569213.988531 | 55.97083864 | 37.41326473 | 1414 | |
| 17 | 250 | 99 | 27392 | 16405509 | -1 | -1 | 411569213.988531 | 55.9695994 | 37.3995729 | 1414 | |
| 18 | 250 | 2 | 7701 | 42477 | -1 | -1 | 411569213.988531 | 55.97348067 | 37.39981076 | 1414 | |
| 19 | 250 | 2 | 7701 | 49269 | -1 | -1 | 411569213.988531 | 55.98165978 | 37.40615409 | 1414 | |
| 20 | 250 | 2 | 5099 | 9905516 | -1 | -1 | 411569213.988531 | 55.98089591 | 37.40480887 | 1414 | |
| 21 | 250 | 99 | 27392 | 16422 | -1 | -1 | 411569213.988531 | 55.97682268 | 37.40994813 | 1414 | |
| 22 | 250 | 1 | 6355 | 60376 | -1 | -1 | 411569213.988531 | 55.97885874 | 37.40365231 | 1414 | |
| 23 | 250 | 2 | 7701 | 20018 | -1 | -1 | 411569213.988531 | 55.9749972 | 37.40963459 | 1896 | |
| 24 | 250 | 99 | 27392 | 16412305 | -1 | -1 | 411569213.988531 | 55.97472347 | 37.40230776 | 1414 | |

<Filter is Empty> Customize...

Record 18 of at least 512

Do you really trust your charger?

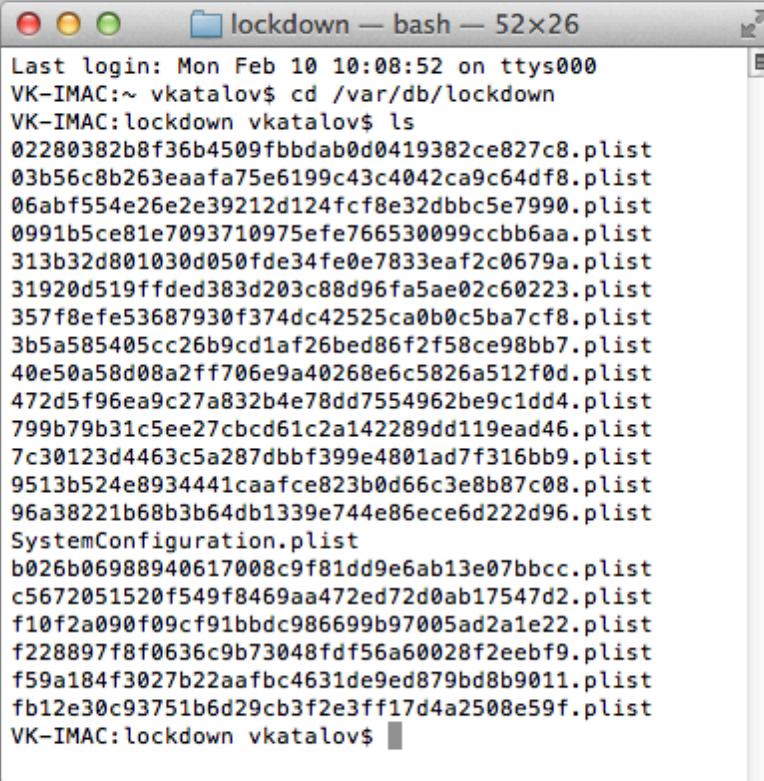


Pair-locking

- iOS device: `/var/root/Library/Lockdown`
- Mac: `/var/db/lockdown`

lockdownd service

- backup service
- software installation service
- get device name & UDID
- sync data
- retrieve a screenshot
- request iOS diagnostic information
- put device into recovery mode
- manage provisioning profiles



```
lockdown — bash — 52x26
Last login: Mon Feb 10 10:08:52 on ttys000
VK-IMAC:~ vkatalov$ cd /var/db/lockdown
VK-IMAC:lockdown vkatalov$ ls
02280382b8f36b4509fbbdab0d0419382ce827c8.plist
03b56c8b263eaafa75e6199c43c4042ca9c64df8.plist
06abf554e26e2e39212d124fcf8e32dbbc5e7990.plist
0991b5ce81e7093710975efe766530099ccbb6aa.plist
313b32d801030d050fde34fe0e7833eaf2c0679a.plist
31920d519ffded383d203c88d96fa5ae02c60223.plist
357f8efe53687930f374dc42525ca0b0c5ba7cf8.plist
3b5a585405cc26b9cd1af26bed86f2f58ce98bb7.plist
40e50a58d08a2ff706e9a40268e6c5826a512f0d.plist
472d5f96ea9c27a832b4e78dd7554962be9c1dd4.plist
799b79b31c5ee27cbcd61c2a142289dd119ead46.plist
7c30123d4463c5a287dbbf399e4801ad7f316bb9.plist
9513b524e8934441caafce823b0d66c3e8b87c08.plist
96a38221b68b3b64db1339e744e86ece6d222d96.plist
SystemConfiguration.plist
b026b06988940617008c9f81dd9e6ab13e07bbcc.plist
c5672051520f549f8469aa472ed72d0ab17547d2.plist
f10f2a090f09cf91bbdc986699b97005ad2a1e22.plist
f228897f8f0636c9b73048fdf56a60028f2eebf9.plist
f59a184f3027b22aafbc4631de9ed879bd8b9011.plist
fb12e30c93751b6d29cb3f2e3ff17d4a2508e59f.plist
VK-IMAC:lockdown vkatalov$
```

Advanced logical acquisition

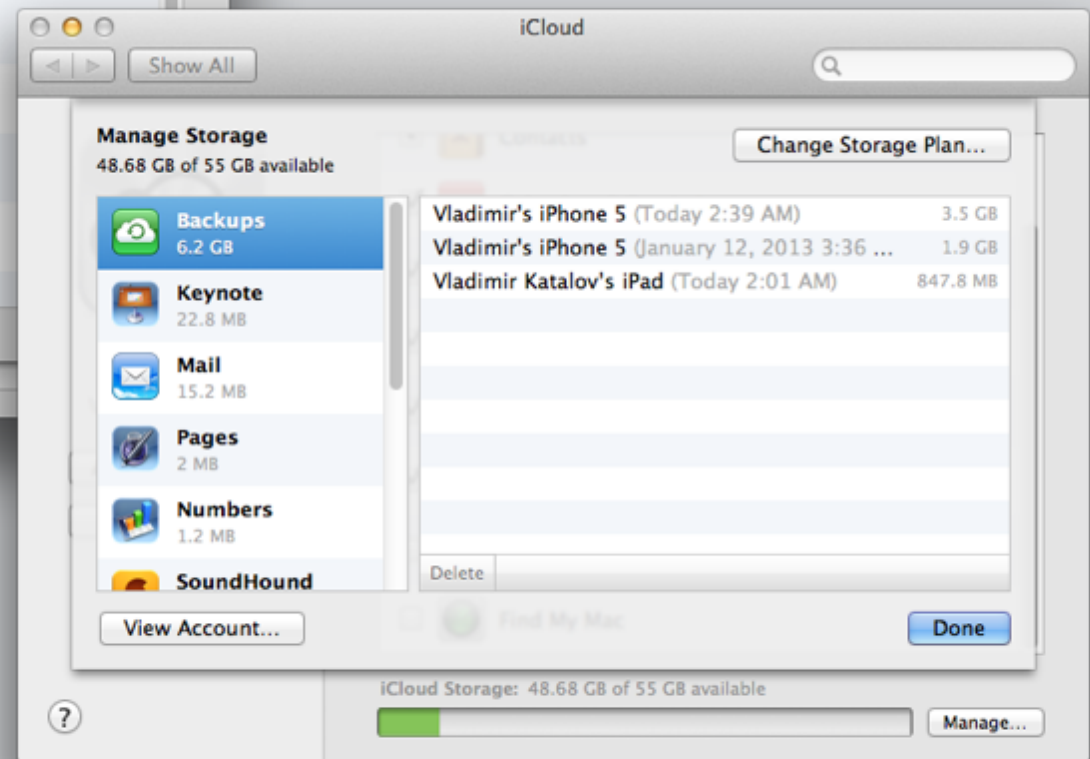
- *for jailbroken devices - the entire file system*
- device information: name, model, IMEI, UUID, serial number etc
- all the media (photos, videos, iTunes library, iBooks)
- application data (including temporary files and *caches* folder)
- various device settings
- log files and diagnostic information
- cached web data (e.g. pictures from social networks)
- keyboard typing caches
- SMS and iMessages (including attachments, even to deleted messages)
- address book
- calendar
- voice mail
- WAL (Write-Ahead Logging) files for most SQLite databases
- ...and more

Works even if device is passcode-locked and backup encryption is set

Can be done over Wi-Fi

Only need the pairing record

iCloud Control Panel



iCloud backups reverse engineering

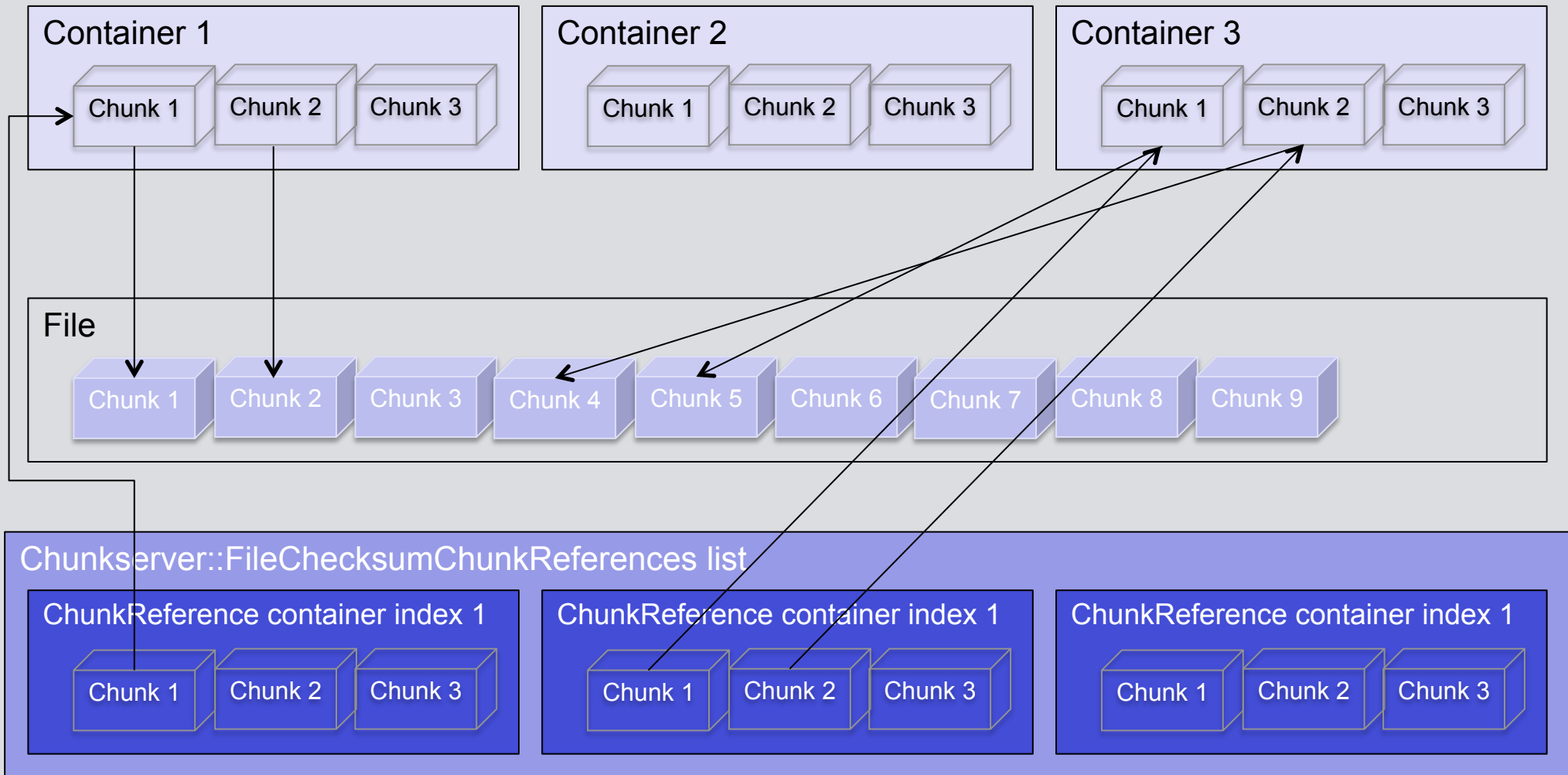
- no backup to iCloud from iTunes :(

so...

- jailbreak iPhone
- Install Open SSH, get keychain (keychain-2.db)
- [Settings] | [iCloud] | [Delete Account] | [Delete from My iPhone]
- [Settings] | [General] | [Reset] | [Reset All Settings]
- reboot
- set up Wi-Fi connection (proxy)
- replace keychain with our own trusted root certificate
- ... read all the traffic :)



Files in iCloud



iCloud backups – authentication, get token, get keys

Authentication:

[https://setup.icloud.com/setup/authenticate/\\$APPLE_ID\\$](https://setup.icloud.com/setup/authenticate/$APPLE_ID$)

Authorization:Basic <authentication data>

authentication data = mime64 (AppleID:password)

returns: mmeAuthToken, dsPrsID

https://setup.icloud.com/setup/get_account_settings

Authorization:Basic <authentication data>

authentication data = mime64 (dsPrsID:mmeAuthToken)

returns: mmeAuthToken (new/other one!!)

[https://p11-mobilebackup.icloud.com/mbs/\(dsPrsID\)](https://p11-mobilebackup.icloud.com/mbs/(dsPrsID))

Authorization: <authentication data>

authentication data = mime64 (dsPrsID:mmeAuthToken)

returns: list of backup IDs (backupudid)

[https://p11-mobilebackup.icloud.com/mbs/2005111682/\(backupudid\)/getKeys](https://p11-mobilebackup.icloud.com/mbs/2005111682/(backupudid)/getKeys)

Get snapshots, file auth. tokens, chunks URLs

Enumerate snapshots

HTTPS GET

[https://p11-mobilebackup.icloud.com/mbs/\(dsPrsID\)/\(backupudid\)/\(snapshotid\)/listFiles?\[...\]](https://p11-mobilebackup.icloud.com/mbs/(dsPrsID)/(backupudid)/(snapshotid)/listFiles?[...])

Get file authentication tokens

HTTPS POST

[https://p11-mobilebackup.icloud.com/mbs/\(dsPrsID\)/\(backupudid\)/\(snapshotid\)/getFiles](https://p11-mobilebackup.icloud.com/mbs/(dsPrsID)/(backupudid)/(snapshotid)/getFiles)

Get URLs for file chunks

HTTPS POST

[https://p11-content.icloud.com/\(dsPrsID\)/authorizeGet](https://p11-content.icloud.com/(dsPrsID)/authorizeGet)

Download chunks

Windows Azure:

[http://msbnx000004.blob.core.windows.net:80/cnt/g6YMJKQBPxQruxQAr30C?\[...\]](http://msbnx000004.blob.core.windows.net:80/cnt/g6YMJKQBPxQruxQAr30C?[...])

Amazon AWS:

[http://us-std-00001.s3-external-1.amazonaws.com/I9rh20QBPX4jizMAr3vY?\[...\]](http://us-std-00001.s3-external-1.amazonaws.com/I9rh20QBPX4jizMAr3vY?[...])

iCloud backups – data encryption

- get keyData from iCloud
- wrappedOffset = keyDataSize - (ECP_LEN + WRAPPED_KEY_LEN)
- get wrappedKey (at wrappedOffset)
- get CLASS_KEY
- iOS 5/6: ((UINT32*)(keyData + wrappedOffset))[-1]
- iOS 7: ((UINT32*)(keyData + wrappedOffset))[-3]
- decrypt wrappedKey using CLASS_KEY
- get AES_KEY from wrappedKey
- file decryption: by 0x1000 blocks (unique IV for every block)

```
#define HFS_IV_GENERATOR 0x80000061
#define IV_GEN(x) (((x) >> 1) ^ (((x) & 1) ?
HFS_IV_GENERATOR : 0))
```

```
static UInt8 *genIV (UInt32 seed, void *pIV) {
    UInt32 *pdw = (UInt32*)pIV;
    pdw[0] = seed = IV_GEN(seed);
    pdw[1] = seed = IV_GEN(seed);
    pdw[2] = seed = IV_GEN(seed);
    pdw[3] = seed = IV_GEN(seed);
    return (UInt8*)pIV;
}
```

```
AES_KEY aesIV;
makeIVkey (&aesIV, abKey, SYSTEM_KEY_LEN);
(abKey is AES_KEY we have got from wrappedKey)
```

```
static AES_KEY *makeIVkey (AES_KEY *pAES, UInt8 *pb,
size_t cb) {
    SHA_CTX sha;
    UInt8 abHash[SHA_DIGEST_LENGTH];
    SHA1_Init (&sha);
    SHA1_Update (&sha, pb, cb);
    SHA1_Final (abHash, &sha);
    AES_set_encrypt_key (abHash, 128, pAES);
    return pAES;
}
```

iCloud backups – encryption, summary

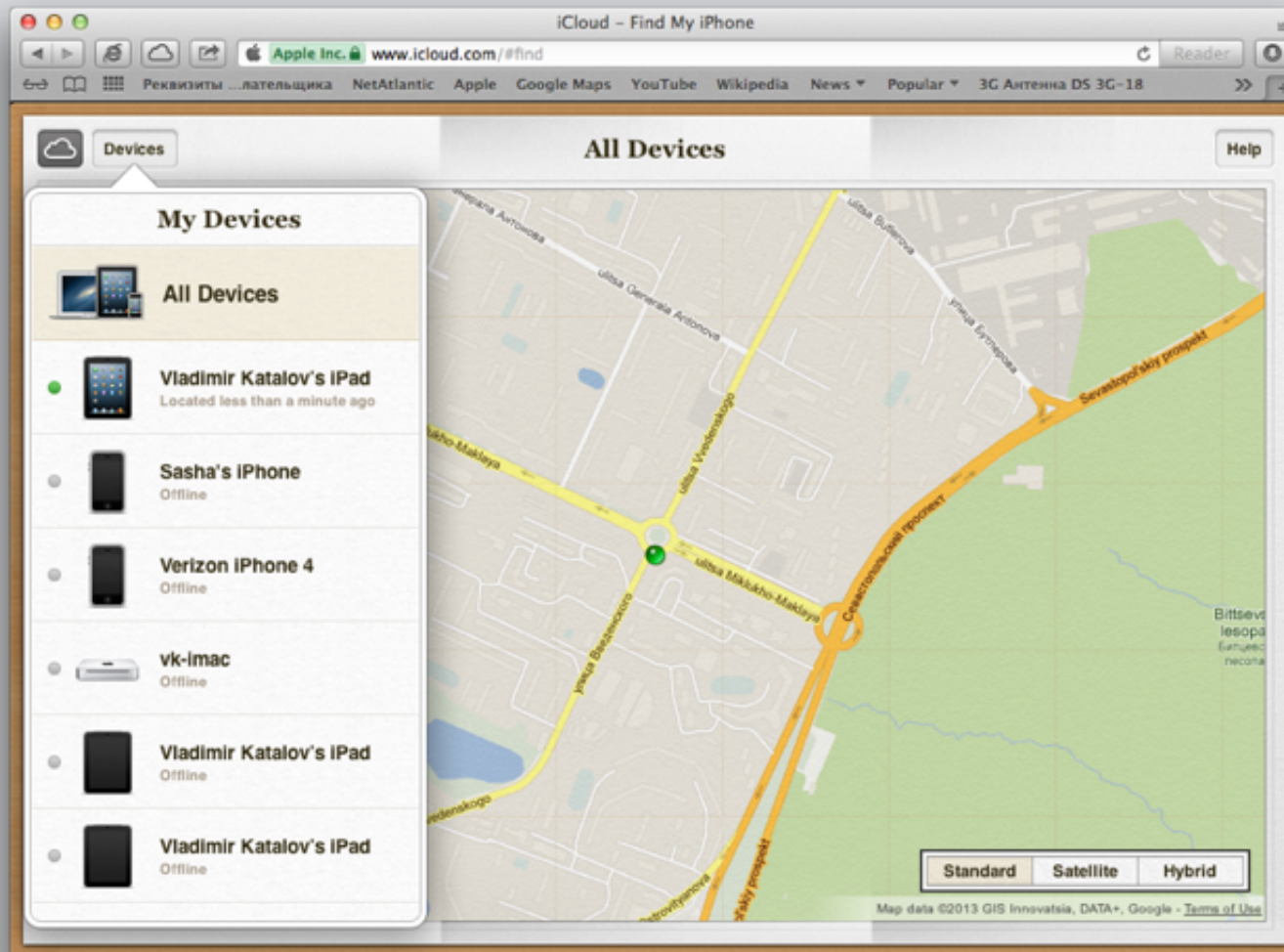
iCloud encryption

- Data stored at 3rd-party storage providers is encrypted
- Apple has encryption keys to that data
- Few files are further encrypted using keys from OTA backup keybag
- Keychain items are encrypted using keys from OTA backup keybag
- Need key 0x835 (securityd) to decrypt most keys from OTA backup keybag

iCloud backups – summary

- There is no user-configurable encryption for iCloud backups
- iCloud backups are stored in Microsoft and Amazon clouds in encrypted form
- Apple holds encryption keys and thus have access to data in iCloud backups
- If Apple stores 0x835 keys then it can also have access to Keychain data
- Apple may have legal obligations to do this (e.g. LE)
- *No notification after backup downloading (as with device restore)!*

Find My Phone



Find My Phone protocol

Authentication

validate

<https://setup.icloud.com/setup/ws/1/validate> (ClientBuildNumber, ClientId)
ClientBuildNumber=1M.63768 (constant), ClientId is random GUID
<- instance

login

<https://setup.icloud.com/setup/ws/1/login> (id, password)
id=sha1(apple_id+instance)
<- dsid

Get devices with location

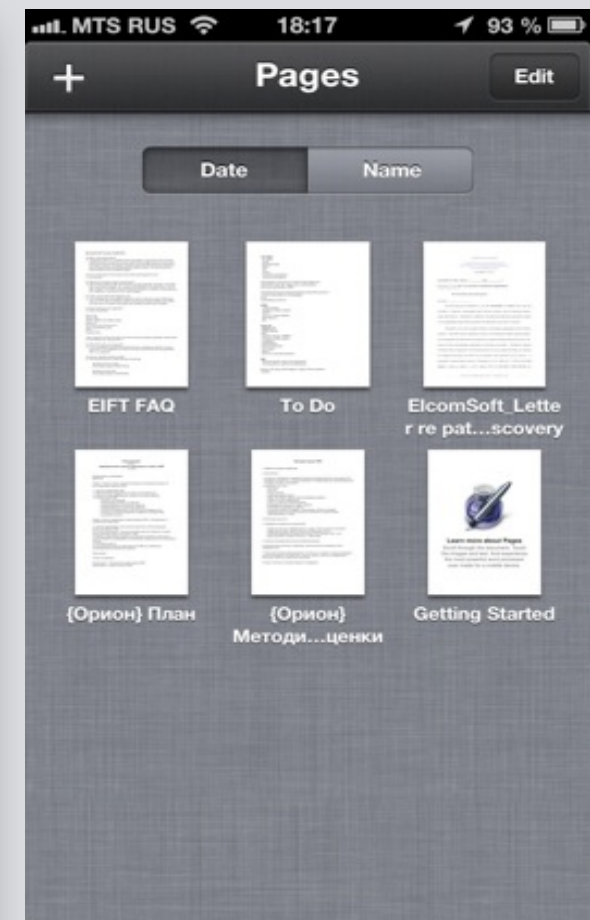
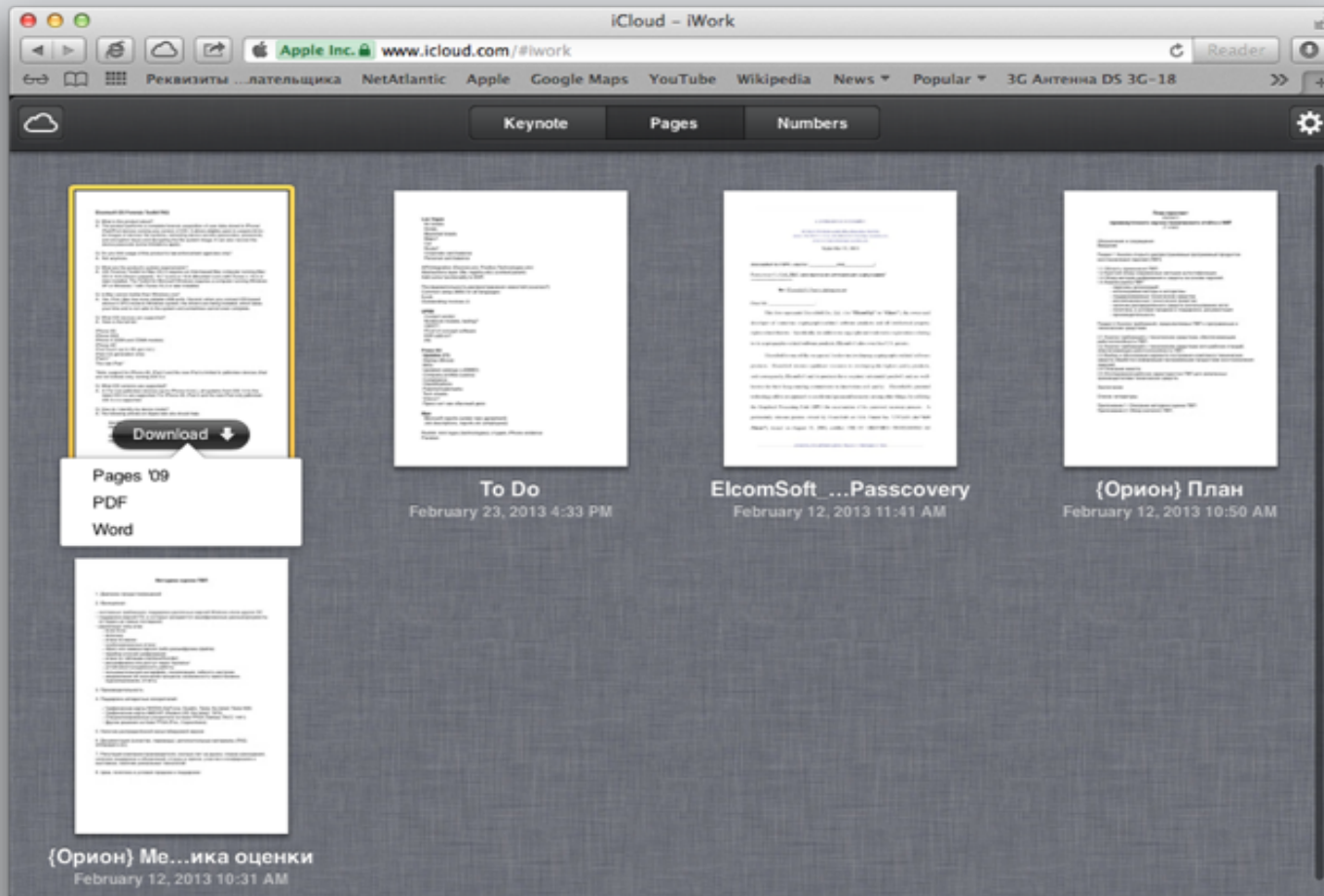
initClient

<https://p11-fmipweb.icloud.com/fmipservice/client/web/initClient>

refreshClient

<https://p11-fmipweb.icloud.com/fmipservice/client/web/refreshClient> (id, dsid)
<- content (location)

iCloud documents



Get files from iCloud

To get list of files

- Authentication request (with given AppleID & password). Client gets mmeAuthToken in return; which, in order, is used to create authentication token (together with dsid). dsid (Destination Signaling Identifier) is an unique ID assigned to the user when registering at iCloud.com.
- Request to get AccountSettings. Client gets an URL (ubiquityUrl) with an address to get UUID (unique user identifier), file list, info on file tokens and for authorization.
- Request to get file list (POST). Output (for every file):
 - file name
 - file id
 - parent folder id
 - last change time
 - checksum
 - access rights

To download given file

- Request to get file token (using file id, checksum and aliasMap).
- Authorization request. Returns information on file chunks and containers. Output: container list (with URLs) and chunk information.

iCloud files: packages

- KeyNote: PDF, Microsoft PowerPoint, KeyNote
- Pages: PDF, Microsoft Word, Pages
- Numbers: PDF, Microsoft Excel, Numbers
- Some other programs: 1Password, WhatsApp etc

Storage: plist + content (text, media files)

Validate, login

<https://setup.icloud.com/setup/ws/1/validate>

<https://setup.icloud.com/setup/ws/1/login>

Export

[https://p15-ubiquityws.icloud.com/iw/export/\(dsid\)/export_document?...](https://p15-ubiquityws.icloud.com/iw/export/(dsid)/export_document?...)

Check export status

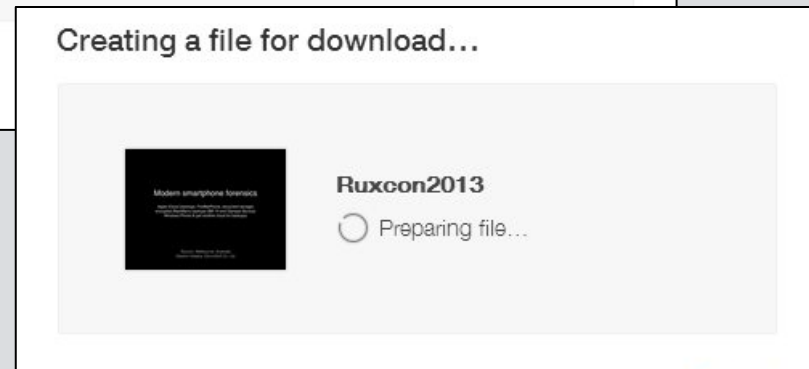
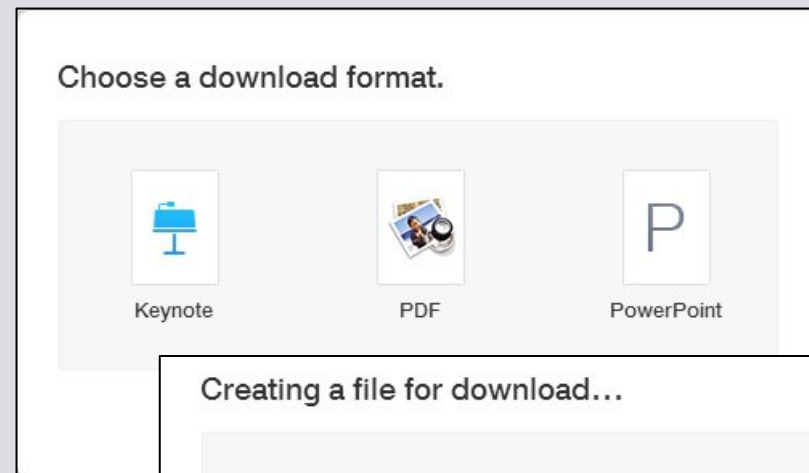
[https://p15-ubiquityws.icloud.com/iw/export/\(dsid\)/check_export_status?...](https://p15-ubiquityws.icloud.com/iw/export/(dsid)/check_export_status?...)

Download converted file

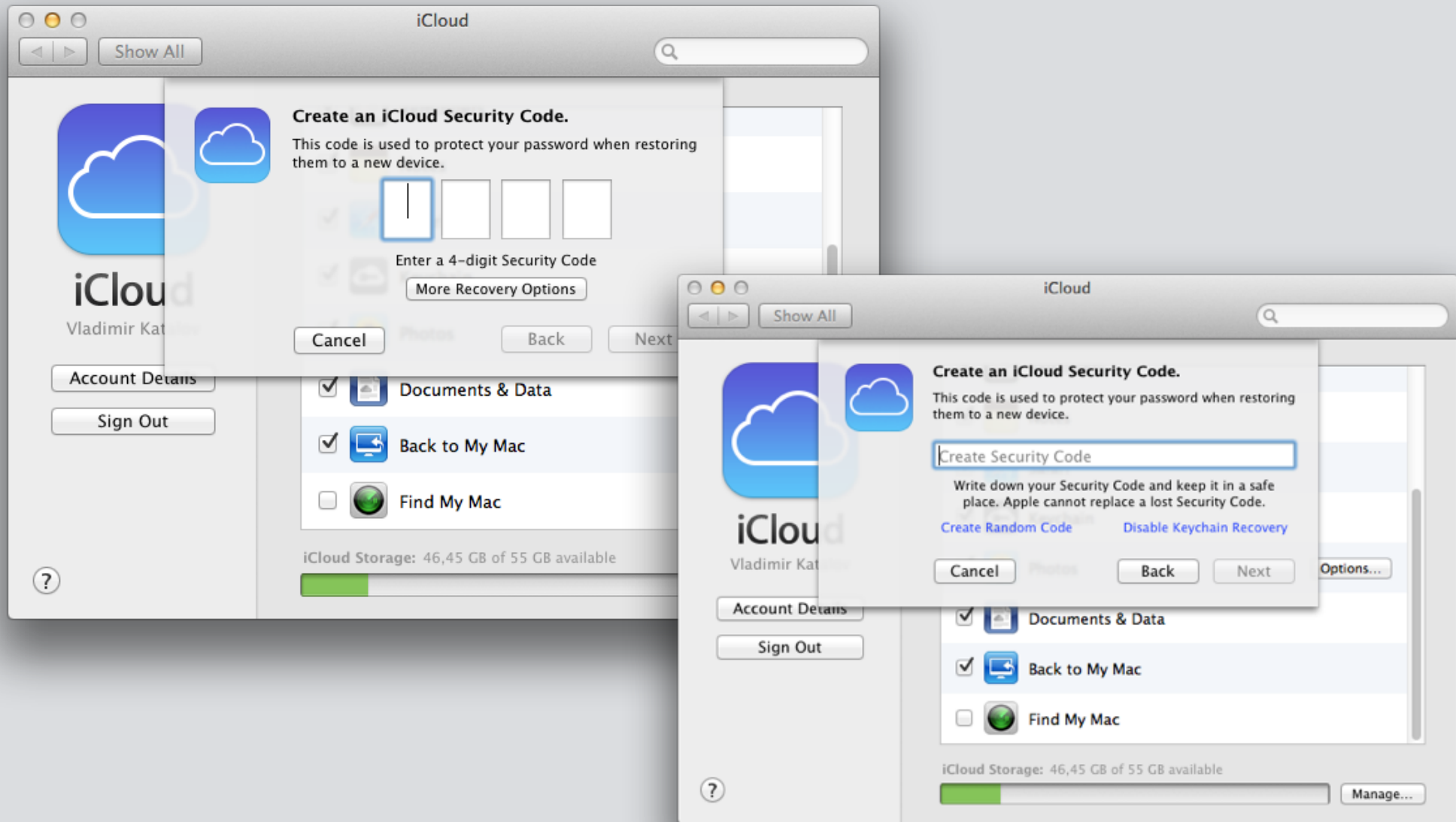
[https://p15-ubiquityws.icloud.com/iw/export/\(dsid\)/download_exported_document?](https://p15-ubiquityws.icloud.com/iw/export/(dsid)/download_exported_document?)

iWork'2013 files w/password: only available as packages

.iwph: password hint, .iwph2: password hash (pbkdf2(sha1, 100000))



iCloud keychain



iCloud keychain



iCloud keychain (cont'd)

Keychain Access

Click to lock the iCloud keychain.

Keychains

- login
- iCloud
- System
- System Roots

Category

- All Items
- Passwords
- Secure Notes
- My Certificates
- Keys
- Certificates

appleid.apple.com (apple@elcomsoft.com)
Kind: Web form password
Account: apple@elcomsoft.com
Where: https://appleid.apple.com
Modified: 11 Jun 2013 07:31:55

| Name | Kind | Date Modified |
|---|------------------------|---------------|
| @ accounts.google.com (Passwords not saved) | Web form password | 11 Jun 2013 |
| AirPort | application password | 11 Jun 2013 |
| aknet | AirPort network pas... | 11 Jun 2013 |
| @ appleid.apple.com (apple@elcomsoft.com) | Web form password | 11 Jun 2013 |
| @ appleid.apple.com (qtt.test@gmail.com) | Web form password | 11 Jun 2013 |
| @ appleid.apple.com (qtt.test@icloud.com) | Web form password | 11 Jun 2013 |
| @ appleid.apple.com (vkatalov@mail.ru) | Web form password | 11 Jun 2013 |
| @ calendar.google.com | Internet password | 11 Jun 2013 |
| @ calendar.google.com | Internet password | 11 Jun 2013 |
| @ daw.apple.com (elcomsoft) | Web form password | 11 Jun 2013 |
| @ daw.apple.com (info@elcomsoft.com) | Web form password | 11 Jun 2013 |
| @ daw.apple.com (vkatalov@gmail.com) | Web form password | 11 Jun 2013 |

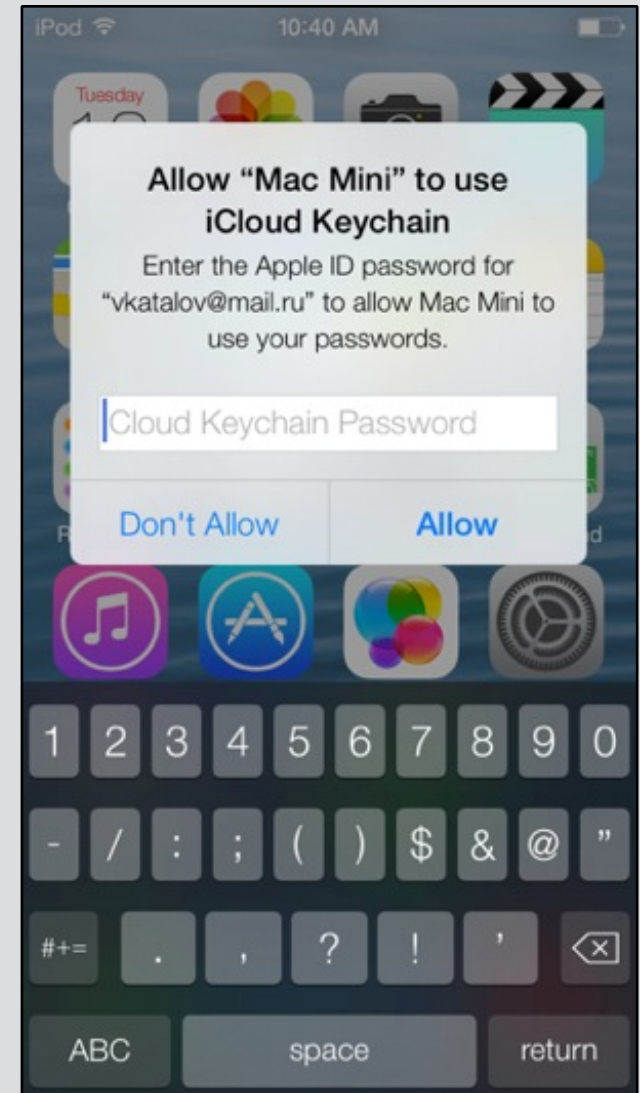
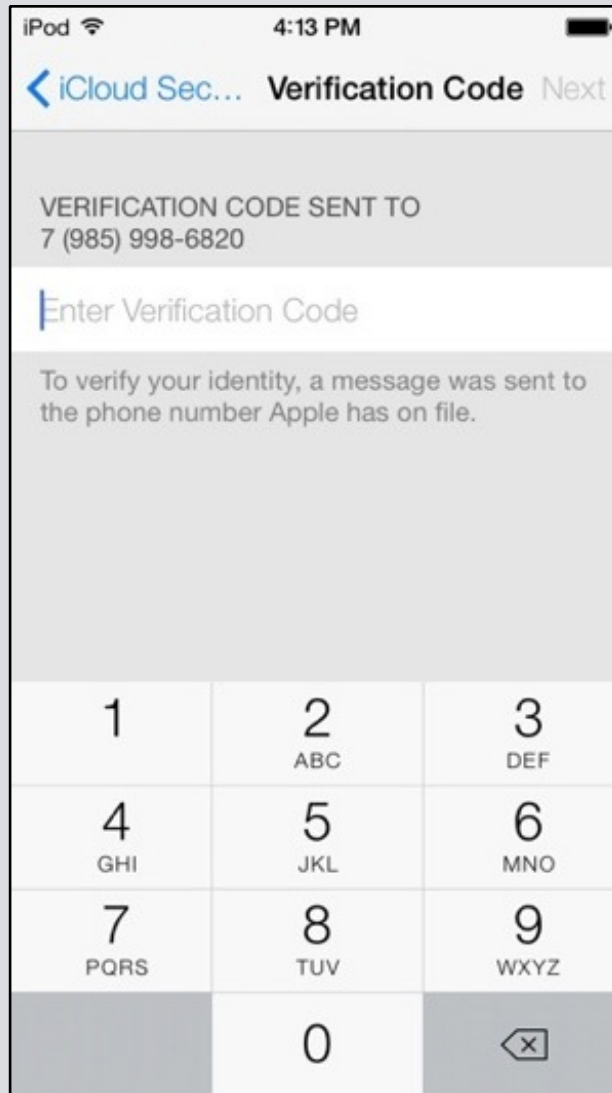
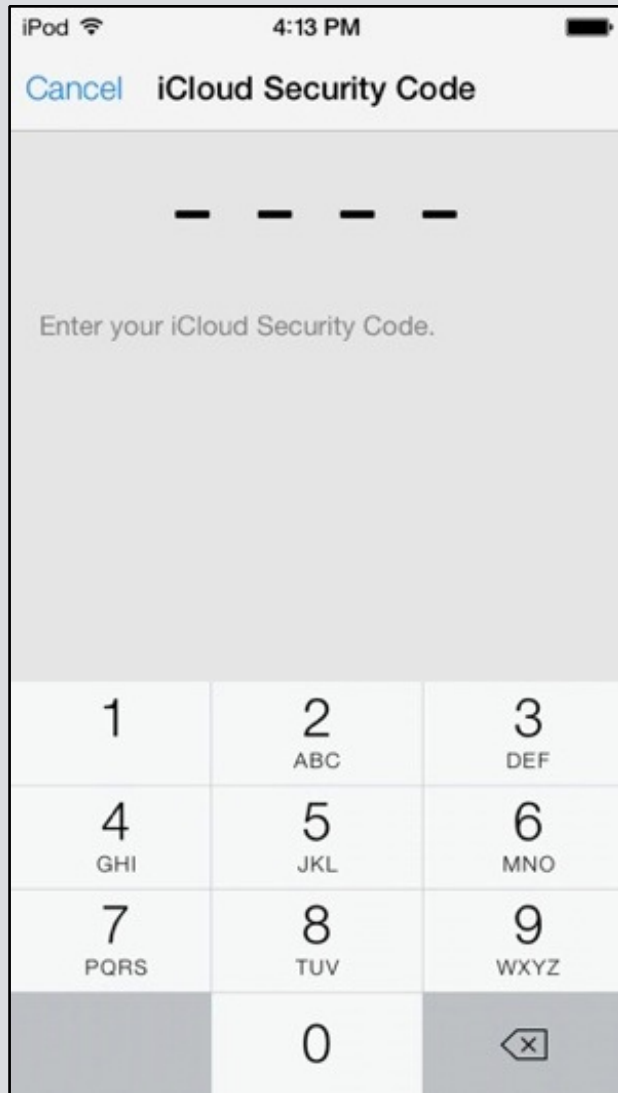
69 items

iPod 10:42 AM

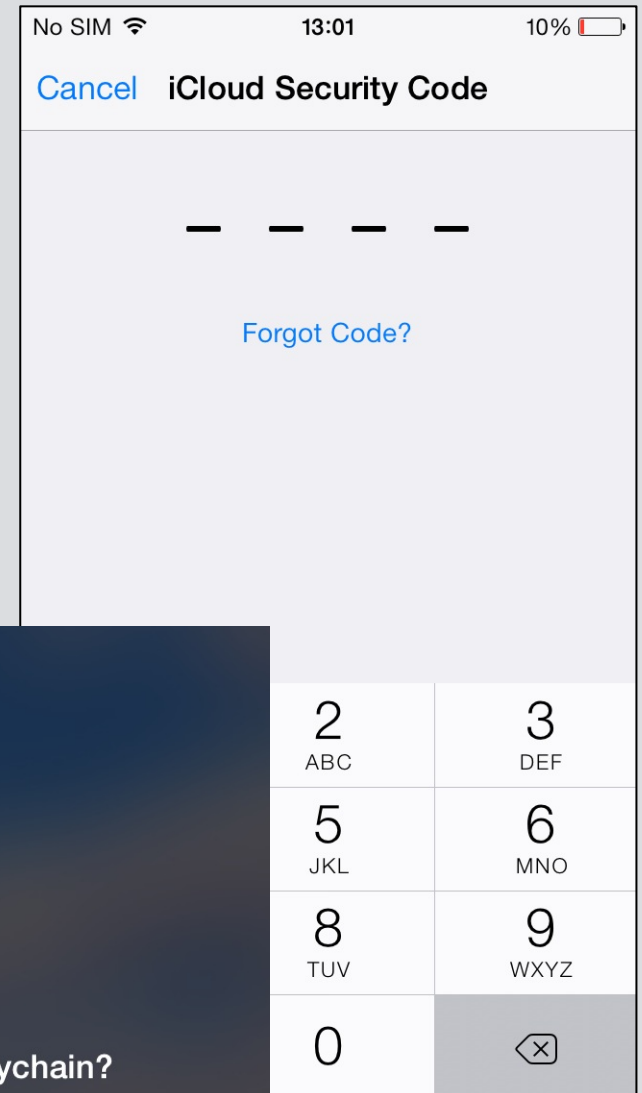
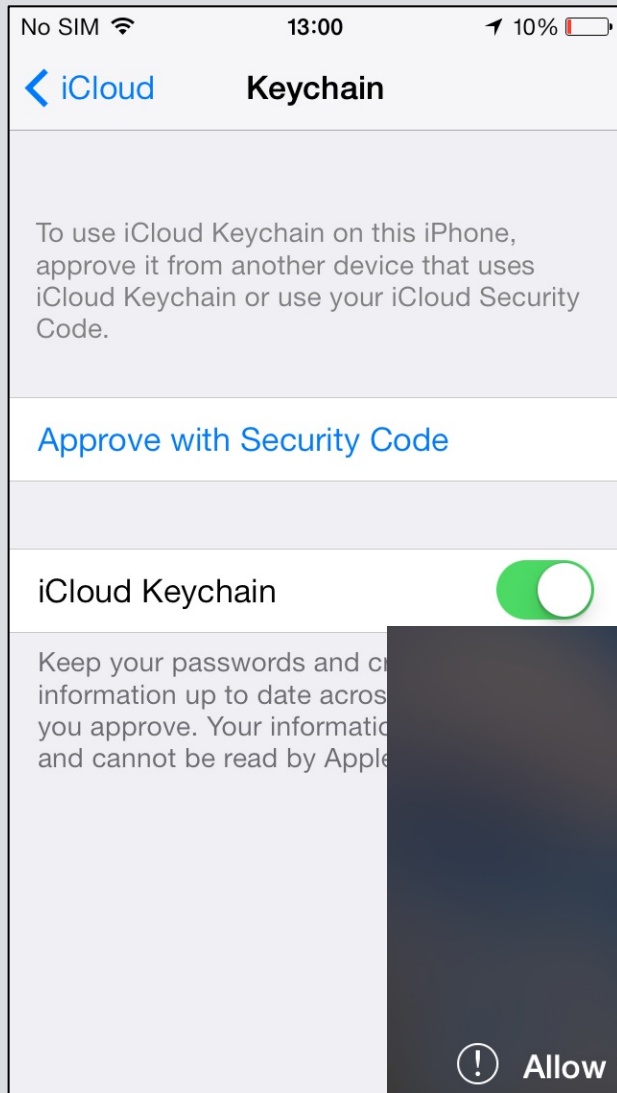
AutoFill Passwords Clear All

- appleid.apple.com
- apple@elcomsoft.com
- appleid.apple.com
- qtt.test@gmail.com
- appleid.apple.com
- qtt.test@icloud.com
- appleid.apple.com
- vkatalov@mail.ru
- daw.apple.com
- elcomsoft
- daw.apple.com
- info@elcomsoft.com
- daw.apple.com
- vkatalov@gmail.com
- daw.apple.com
- vkatalov@mail.ru

iCloud keychain (cont'd)



iCloud keychain (cont'd)

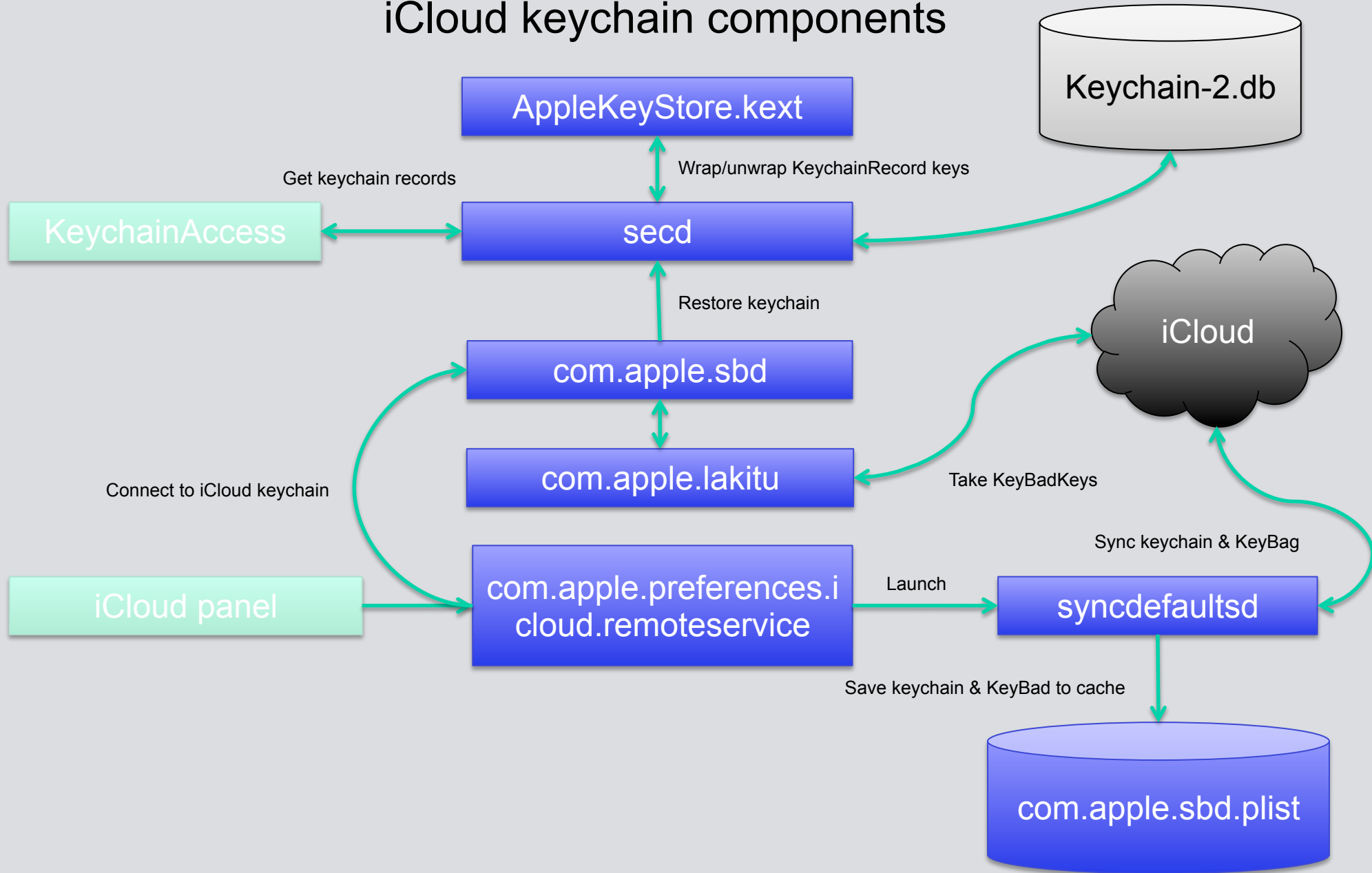


12:59

Вторник, 11 февраля

ⓘ Allow “Элкомсофт iPhone 5 (JB)” to use iCloud Keychain?
 Enter the Apple ID password for “apple@elcomsoft.com” to allow Элкомсофт iPhone 5 (JB) to use iCloud Keychain.

iCloud keychain components



iCloud keychain components

com.apple.preferences.icloud.remoteservice

process that interacts with iCloud control

com.apple.sbd

(daemon) caching and restoring keychain, get notifications

com.apple.lakitu

(daemon) talks to iCloud (get requests from com.apple.sbd, make queries to iCloud, get and decrypt responses, passes them back to com.apple.sbd)

secd

(daemon) caching and restoring keychain, get notifications

AppleKeyStore.kext

driver to interact with KVS (key-value storage)

Security.framework

functions to restore keychain, save it to SQLite database, send notifications (e.g. to Keychain access)

libcorecrypto.dylib

encryption/decryption

iCloud keychain retrieval

GetAccountSettings

POST https://setup.icloud.com/setup/get_account_settings HTTP/1.1

Host: setup.icloud.com

Authorization: Basic MTc3MzgyNT ... INLy9TbkU9

[...]

Authorization = Base64("apple_id" + ":" + "password")

Get mmeAuthToken in return

X-MobileMe-AuthToken = Base64("dsid" + ":" + "mmeAuthToken")

Sync (get Keychain and KeyBag) - query

POST <https://p18-keyvalueservice.icloud.com/sync> HTTP/1.1

Host: p18-keyvalueservice.icloud.com

Authorization: X-MobileMe-AuthToken MTc3Mzg ... meWRINDg9

[...]

```
<dict>
  <key>apns-token</key>
  <data>
    D7wxUEz2av7JaSgJD6j2lyQKENzH0e4DGJzfOeLBbYA=
  </data>
  <key>apps</key>
  <array>
    <dict>
      <key>bundle-id</key>
      <string>com.apple.security.cloudkeychainproxy3</string>
      <key>kvstore-id</key>
      <string>com.apple.security.cloudkeychainproxy3</string>
      <key>registry-version</key>
      <string>FT=-@RU=40c72786-6f77-4190-85d8-3ae1f4df91ca@S=1286</string>
    </dict>
    <dict>
      <key>bundle-id</key>
      <string>com.apple.sbd</string>
      <key>kvstore-id</key>
      <string>com.apple.sbd3</string>
      <key>registry-version</key>
      <string>FT=-@RU=40c72786-6f77-4190-85d8-3ae1f4df91ca@S=1259</string>
    </dict>
  </array>
  <key>service-id</key>
  <string>iOS</string>
</dict>
```

Sync (get Keychain and KeyBag) - response

```
<dict>
  <key>apps</key>
  <array>
    <dict>
      <key>kvstore-id</key>
      <string>com.apple.security.cloudkeychainproxy3</string>
      <key>keys</key>
      <array>
        <dict>
          <key>data</key>
          <data>AYYkF93rOBg ... AABVag==</data>
          <key>name</key>
          <string>com.apple.securebackup.record</string>
        </dict>
      </array>
      <key>bundle-id</key>
      <string>com.apple.sbd</string>
    </dict>
  </array>
  <key>timestamp</key>
  <integer>1384690786479</integer>
</dict>
```

SMS challenge?

POST https://p18-escrowproxy.icloud.com:443/escrowproxy/api/get_sms_targets HTTP/1.1
Authorization: X-MobileMe-AuthToken MTc3Mzgy...WkwdXM9

```
...
<dict>
  <key>command</key>
  <string>GET_SMS_TARGETS</string>
  <key>version</key>
  <integer>1</integer>
</dict>
</plist>

<dict>
  <key>success</key>
  <true/>
  <key>enrollRequiresPhoneNumber</key>
  <true/>
  <key>isHSAEnabled</key>
  <true/>
...
</dict>
```

SMS challenge (cont'd)

POST https://p15-escrowproxy.icloud.com:443/escrowproxy/api/generate_sms_challenge HTTP/1.1

Authorization: X-MobileMe-AuthToken MTY5NT...T0INS3FnQnM9

...

```
<dict>
<key>command</key>
<string>GENERATE_SMS_CHALLENGE</string>
<key>label</key>
<string>com.apple.securebackup.record</string>
<key>version</key>
<integer>1</integer>
</dict>
```

```
<dict>
<key>version</key>
<integer>1</integer>
<key>success</key>
<true/>
<key>smsChallengeLength</key>
<integer>6</integer>
<key>message</key>
<string>Success</string>
</dict>
```

Get KeyBagKey – srp_init

POST https://p18-escrowproxy.icloud.com:443/escrowproxy/api/srp_init HTTP/1.1

Host: p18-escrowproxy.icloud.com:443

```
<dict>
  <key>blob</key>
  <string>dSyhi0M/...CQ==</string>
  <key>command</key>
  <string>SRP_INIT</string>
  <key>label</key>
  <string>com.apple.securebackup.record</string>
  <key>phoneNumberToken</key>
  <string>AQAAAABSidUhUkYydkSNDx8dc4r/QMudn0Q1ctg=</string>
  <key>version</key>
  <integer>1</integer>
  <key>smsChallengeCode</key>
  <string>код</string>
</dict>
```

HTTP/1.1 200 OK

[...]

```
<plist version="1.0">
  <dict>
    <key>respBlob</key>
    <string>AAABiAAA...638rrzw8=</string>
    <key>dsid</key>
    <string>1773825601</string>
  </dict>
</plist>
```

Keychain recovery (get KeyBagKey)

POST <https://p18-escrowproxy.icloud.com:443/escrowproxy/api/recover> HTTP/1.1

Host: p18-escrowproxy.icloud.com:443

[...]

<dict>

<key>blob</key>

<string>AAAAYAAA ... +m8</string>

<key>command</key>

<string>RECOVER</string>

<key>version</key>

<integer>1</integer>

</dict>

HTTP/1.1 200 OK

[...]

<dict>

<key>respBlob</key>

<string>AAADKAA...1FHUaEwbQ==</string>

</dict>

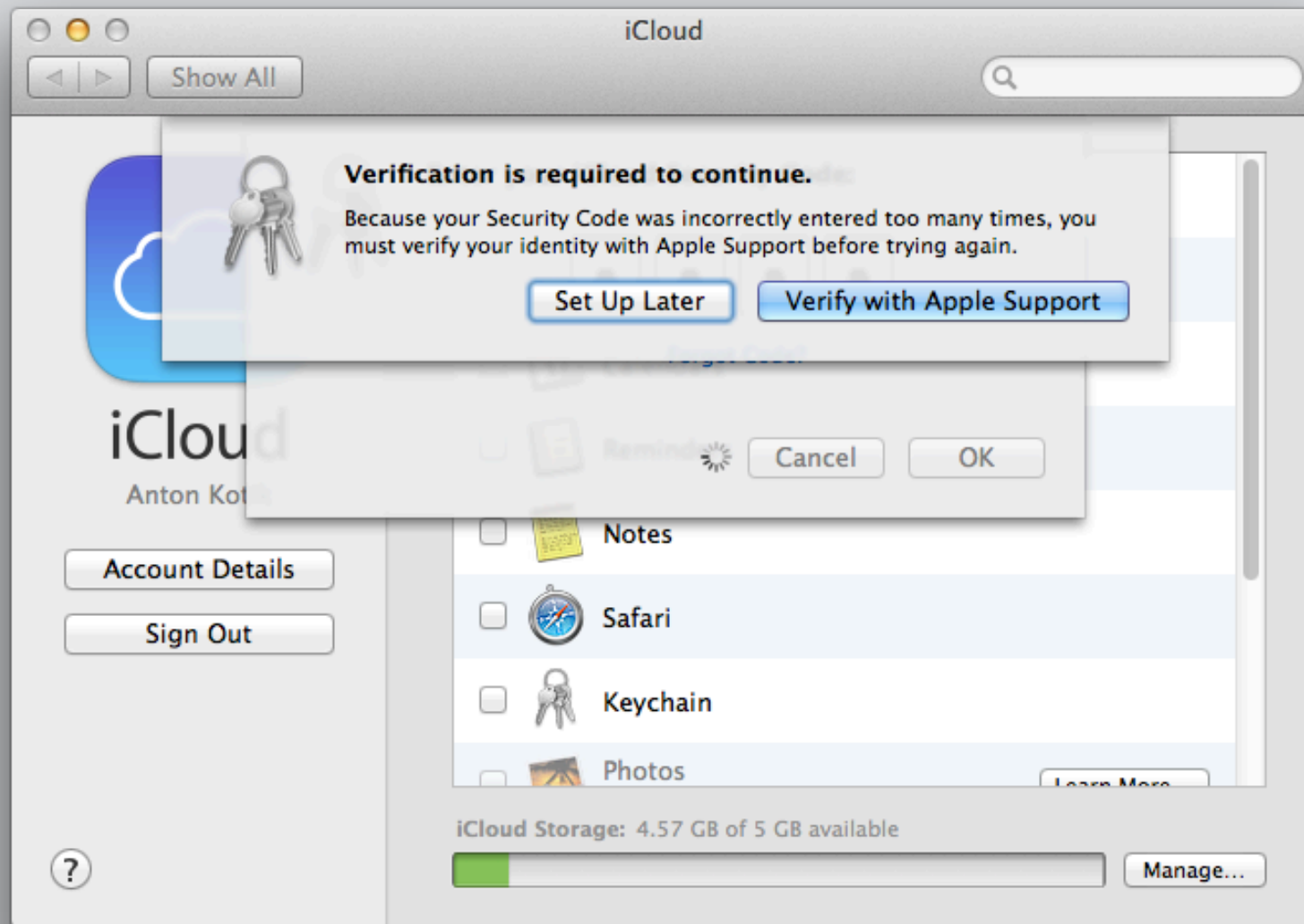
No prompt whether to allow the new device to use the keychain!!

Keychain decryption

- *genp* class: Wi-Fi passwords, credit card data
- *inet* class: saved Safari passwords and autocomplete fields
- Version (4 bytes)
- KeyClass (4 bytes): index in CLASS_KEY array
- KeyDataSize (4 bytes)
- KeyData (0x48 bytes)
 - RecordPublicKey (0x20 bytes)
 - WrappedKey (0x28 bytes)
- CryptaData
- SharedKey = curve25519(CLASS_KEY.privateKey, RecordPublicKey)
- md = sha256("\x00\x00\x00\x01" + SharedKey + RecordPublicKey + CLASS_KEY.publicKey)
- AES_unwrap_key(md)
- gcm_init_key
- gcm_decrypt_message

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00000000 | 03 | 00 | 00 | 00 | 06 | 00 | 00 | 00 | 48 | 00 | 00 | 00 | 8C | B0 | 01 | CE |
| 00000010 | F5 | 80 | CD | 22 | 77 | 5F | 83 | CD | C7 | 9B | 4A | 91 | 0E | 47 | B2 | 3E |
| 00000020 | DE | A4 | 2B | 1D | 26 | A2 | 17 | 25 | B2 | CF | A1 | 20 | 1A | CA | A9 | 3A |
| 00000030 | BB | 4B | 00 | F4 | 7E | 0E | 7D | A4 | 7E | 5F | 07 | 4D | C2 | 4A | D9 | 31 |
| 00000040 | 3D | D1 | 2E | D6 | DC | C0 | 31 | 62 | 4C | 56 | 99 | 5B | 43 | 5C | F6 | B7 |
| 00000050 | 61 | 8E | 6B | A6 | 79 | B2 | BF | D9 | 68 | 74 | FA | 18 | 31 | 89 | 26 | 49 |

iCloud keychain – no brute-force available



Apple iCloud: conclusion

- Balance between security, privacy and convenience
- iCloud security risks
- Use additional encryption
- Better 2FA implementation
- Need further work
 - My Photo Stream
 - Photo Sharing
 - 3rd party apps data
 - Back To My Mac
 - Touch ID (iPhone 5S)



Windows Phone backups

What is saved:

- Internet Explorer Favourites
- List of installed apps
- Theme and accent configuration
- Call history
- App settings (where applicable - email and accounts, lock screen etc)
- Text messages (SMS conversations)
- Photos (good quality - uses data allowance)

Can get with LiveSDK:

- Basic user information
- Contacts
- Calendars
- Files, photos, videos, documents

Download full backup?

Windows Phone: Live! SDK

Identity API

- Get basic information on user

Hotmail API

- Manage contacts
- Manage calendars & events

SkyDrive API

- Files & documents
- Photos
- Videos



Windows Phone: Live! SDK

- Authentication
- Needs client_id of registered application
- Several requests to https://login.live.com to get redirects and some parameters
- Send “SetDeviceInfo” request (same for Contacts/SMS/email/notes)

POST [https://blu-m.hotmail.com/\[...\]](https://blu-m.hotmail.com/[...])

Authorization: Basic dGVzdGluZy53aW5kb3dzMUBtYWlsLnJ1OnF3ZXJ0eTEyMzQ=

```
<Settings xmlns="Settings:">
```

```
  <DeviceInformation>
```

```
    <Set>
```

```
      <Model>RM-820_nam_canada_246</Model>
```

```
      <IMEI>353653050052892</IMEI>
```

```
      <FriendlyName>Lumia 920</FriendlyName>
```

```
      <OS>Windows Phone 8.0.9903</OS>
```

```
      <OSLanguage>English</OSLanguage>
```

```
      <PhoneNumber>+*****</PhoneNumber>
```

```
      <UserAgent>MSFT-WP/8.0.9903</UserAgent>
```

```
      <EnableOutboundSMS>0</EnableOutboundSMS>
```

```
      <MobileOperator>Developer Device</MobileOperator>
```

```
    </Set>
```

```
  </DeviceInformation>
```

```
  <UserInformation>
```

```
    <Get/>
```

```
  </UserInformation>
```

```
</Settings>
```

Authorization: Base64(“login” + “.” + “password”)

- Request folder hierarchy
- Request sync key for folders
- GetItemEstimate to get item count

Windows Phone: folder hierarchy

| Value | Folder |
|--------|--|
| 1 | User-created (generic) |
| 2 | Default Inbox |
| 3 | Default Drafts |
| 4 | Default Deleted Items |
| 5 | Default Sent Items |
| 6 | Default Outbox |
| 7 | Default Tasks |
| 8 | Default Calendar |
| 9 | Default Contacts |
| 10 | Default Notes |
| 12..17 | User-defined Mail, Calendar, Contacts etc. |

WP8: get SMS

Server: *https://???-m.hotmail.com*
(to get correct name: send request to *blum-hotmail.com*)
Protocol: ASHTTP
Data format: wbxml
Can be compressed ("Accept-Encoding: gzip, deflate")

Get (login, password in base64

✓ success

FolderHierarchy

✓ success

SyncKeys

✓ success

CategoriesItems (CollectionId)

✓ return: SyncKey

GetItemEstimate (SyncKey, CollectionId)

✓ number of SMS

GetSMS (SyncKey, CollectionId)

✓ new SyncKey and SMS

GetClosedSMS

✓ new SyncKey

WP8: get mail

GET <http://mail.live.com/> HTTP/1.1

Host: mail.live.com

Connection: keep-alive

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36

Accept-Encoding: gzip, deflate, sdch

< redirect to authentication (<https://login.live.com/login.srf?...>)

> POST <https://login.live.com/ppsecure/post.srf?...> (login, password)

< redirect to mail.live.com?id=XXX

> GET <https://mail.live.com/?id=XXX>

< redirect to mailbox:

HTTP/1.1 302 Found

Location: <https://col131.mail.live.com/default.aspx?id=XXX&rru=inbox>

Go to mailbox:

GET <https://col131.mail.live.com/default.aspx?id=XXX&rru=inbox> HTTP/1.1

Host: col131.mail.live.com

Connection: keep-alive

Windows Phone: find my phone

- **Map phone's location**

GET <https://www.windowsphone.com/ru-ru/my/find HTTP/1.1>

or

GET <https://www.windowsphone.com/ru-ru/my/phones/.../locate-status?request=17>

- **Make the phone ring**

<https://www.windowsphone.com/ru-ru/my/phones/c34a5c89b6aabc87cdc457b49e5f3abbbf81c72e0b19d48bdbd3918e36785f646/ring>

- **Lock the phone and show a message**

- **Erase the phone**

Authentication is required, of course!

1. *If device is not found, service sends an email (with the link to windowsphone.com) ones it becomes available*
2. *To get device location, we have to select it (so it becomes 'active'), and the state is saved on the site*
3. *The number of device location requests is limited (per day)*

BlackBerry backups

Old format:

- IPD files (all databases in a single container)
- BBB files (in fact, ZIP archives with several IPDs, one database per IPD)

New format:

- Unencrypted BBB-QNX (three .tar files inside); for PlayBook with firmware <2.0
- Encrypted BBB-QNX (all .tar files are encrypted); for BB OS 10 (created w/ BlackBerry Link)

For old formats - simple password protection:

- Encryption: AES-256
- Password verification:
 - BlackBerry Desktop Software 5: pbkdf2 (1) - *yes, just one iteration*
 - BlackBerry Desktop Software 6: pbkdf2 (20,000)

BB 10 backups

- mounting QNX6 partitions
- backup encryption: AES-256
- authentication/verification: HMAC-SHA1
- backup.cgi:backuparch
- backup.cgi:scramble

bbid (BlackBerry ID)

(libbbid.so:bbid_profile_get_user_properties(urn:bbid:username)

qbek

(libbbid.so:bbid_profile_get_user_properties(urn:bbid:backupandrestorekey)

cache storage: /accounts/<id>/sys/bbid/keyCache

if not found: request to BB Olympia Service (blackberryid.blackberry.com)

BlackBerry token service

- request: bbid, password, pin, salt (client's entropy)
- response:

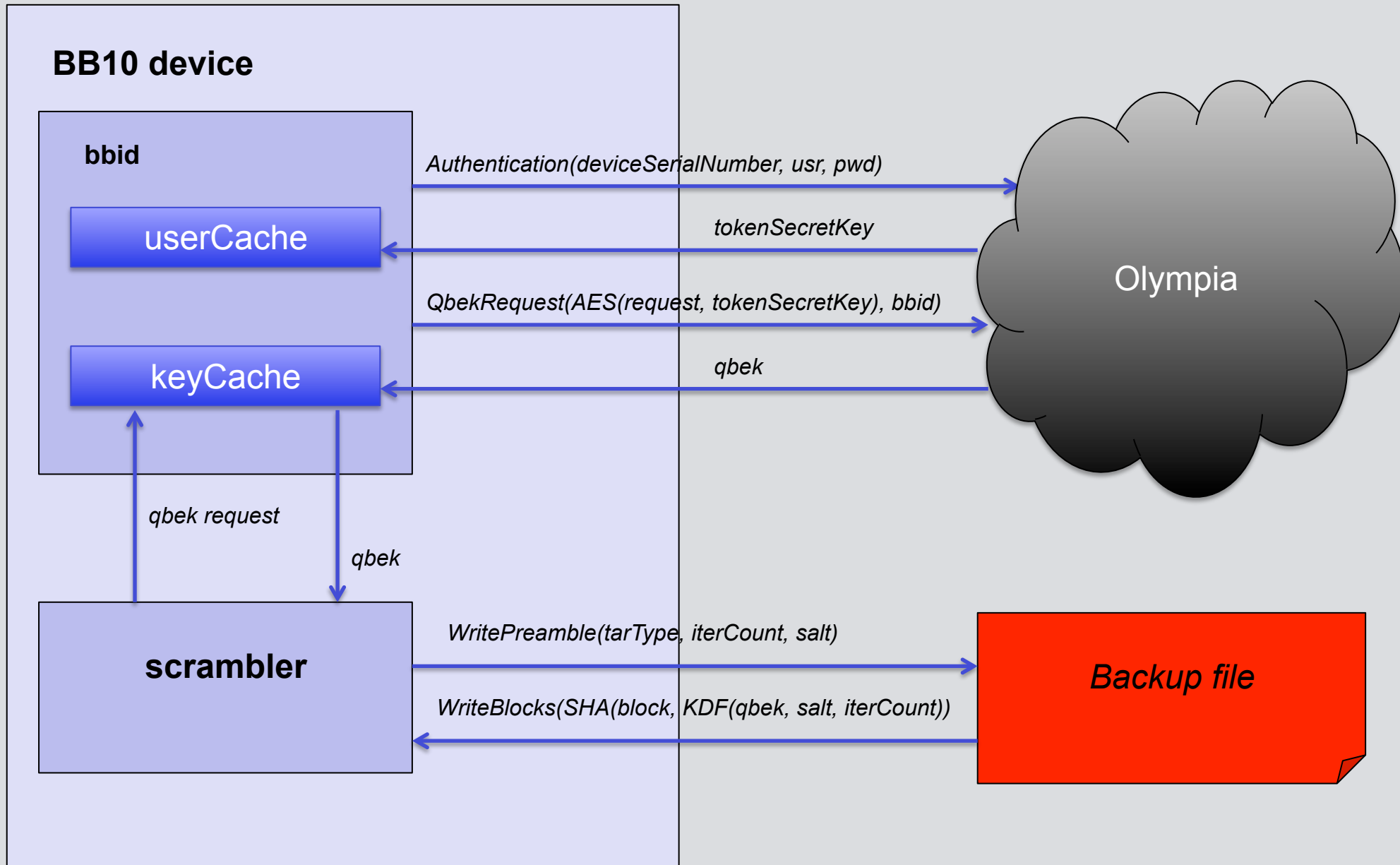
```
Hct=1379081439336&st=1379168703336&se=PF3V5ikbH8fx2wSb2mbHITGy0q1xlcGZZ66Oma3o66  
k&et=1381673439336&fn=John&ln=Doe&nn=johndoe-59094&un=john.doe  
%40gmail.com&ec=AcDGzWbVM12nd0BigqlfJYw%3D&em=john.doe  
%40gmail.com&at=AQ:AQ:zTh0_L5BwTuZf0w0L2CYVGmMyrzSbs7OszPBq72NIYY:ibKt2ZKGOsAjO  
Dk6lITmQA:asSsJMYRzS8Tf2IMQY44_HiCDaWzCBRwQj68XDDH0z6Qhp7gCXuKqSk6_v4KTQ8p  
WMtpVriBNBWO4t2lg879MY_Oro2upCzw32EmCgAKapUPGTleAIKeo3kr13v-  
Td2lpWU0b3kQJVJsTMz9GBjG29RFkcxw-039ksxUJYnDxkCrgbrAwVFpw5Pg5XmAZxtA
```

- se - server entropy
- at - authentication token
- ec - user ID for BB cloud services (saved to /dev/rpmb/BBID_BDEK)
- at (creation time), st (server time), et (expiry time)
- further requests: RST (Request Secure Token) with token type and service name

to get qbek:

- get authentication token
- get BBIDAuthN_1 token for urn:bbid:v1:olympia)
- send request for authzo:qbek token
- register device on BB server (using authzo:qbek token)
- get request on backupAndRestoreKey info (two IDs)
- get janusUrl by request to kronos.bbprotect.blackberry.com
- get qbek from %janusUrl%/FlashGetFile

- **Valid device PIN should be supplied**
- **qbek depends on BlackBerry ID only**



Special thanks to our team:

Researches

- Anton K.
- Sergey P.
- Denis B.

Developers

- Alina P.
- Oleg K.

Testers

- Artem G.
- Oxana B.

Managers

- Anna S.
- Vladimir S.



I could not have done it without you! ;)

Thank you!

*Vladimir Katalov, ElcomSoft Co. Ltd.
(twitter: @vkatalov)*

<http://www.elcomsoft.com>

<http://blog.crackpassword.com>

Facebook: ElcomSoft

Twitter: @elcomsoft