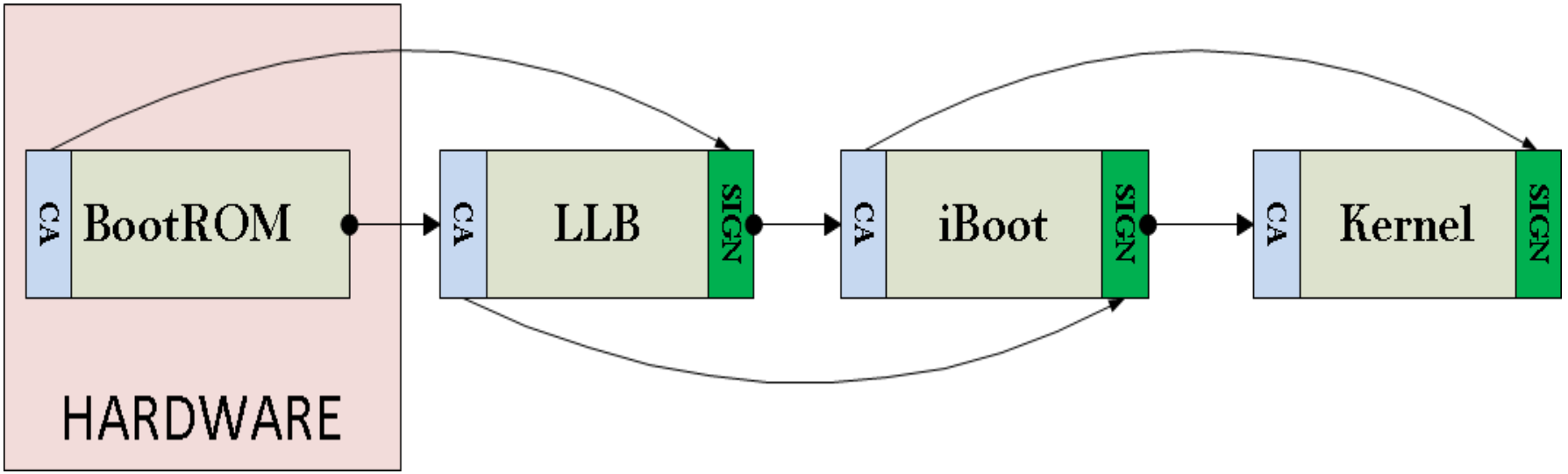
iOS MITM Attack
Technology and effects

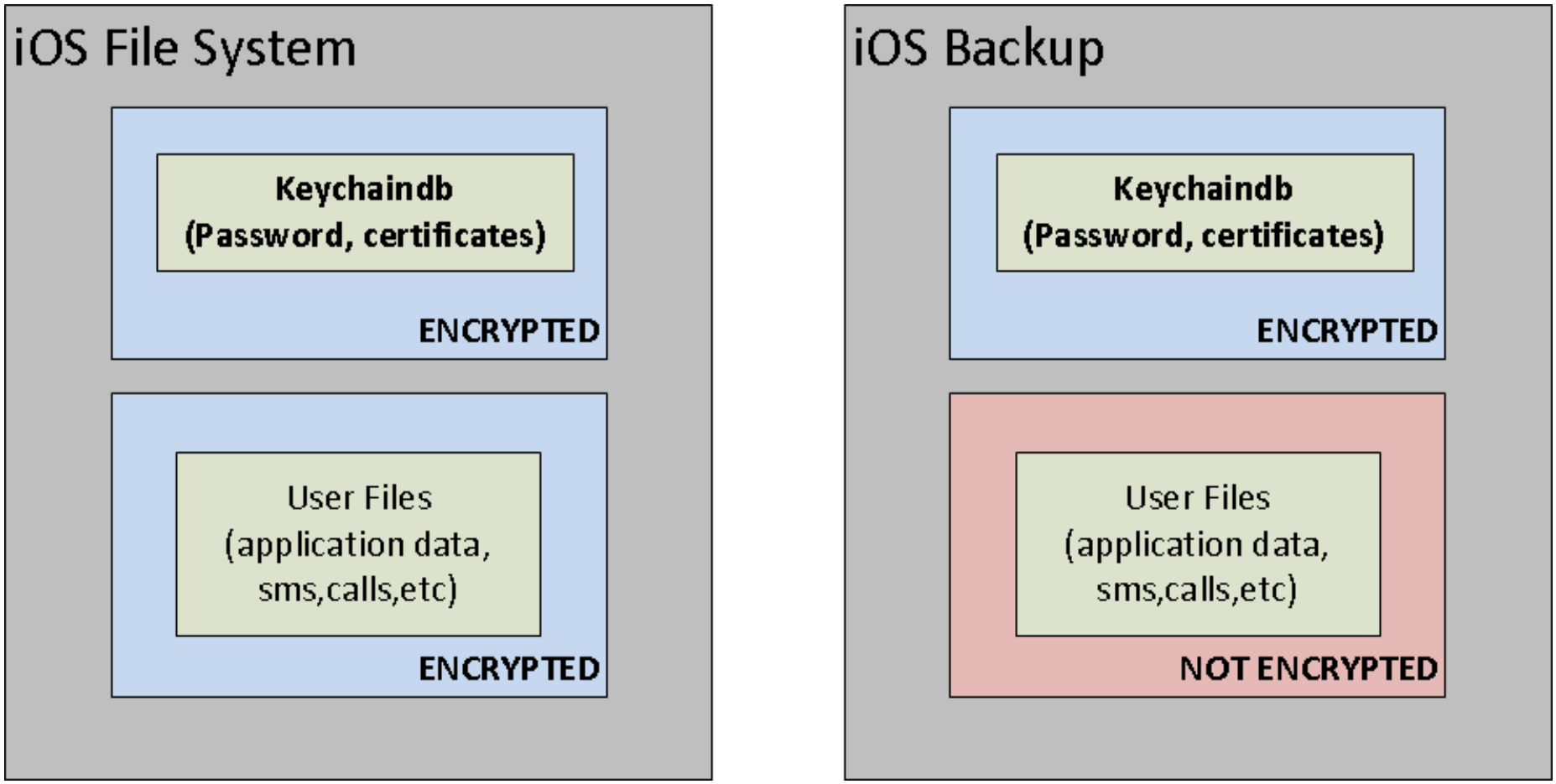

sieg.in

Boot validation

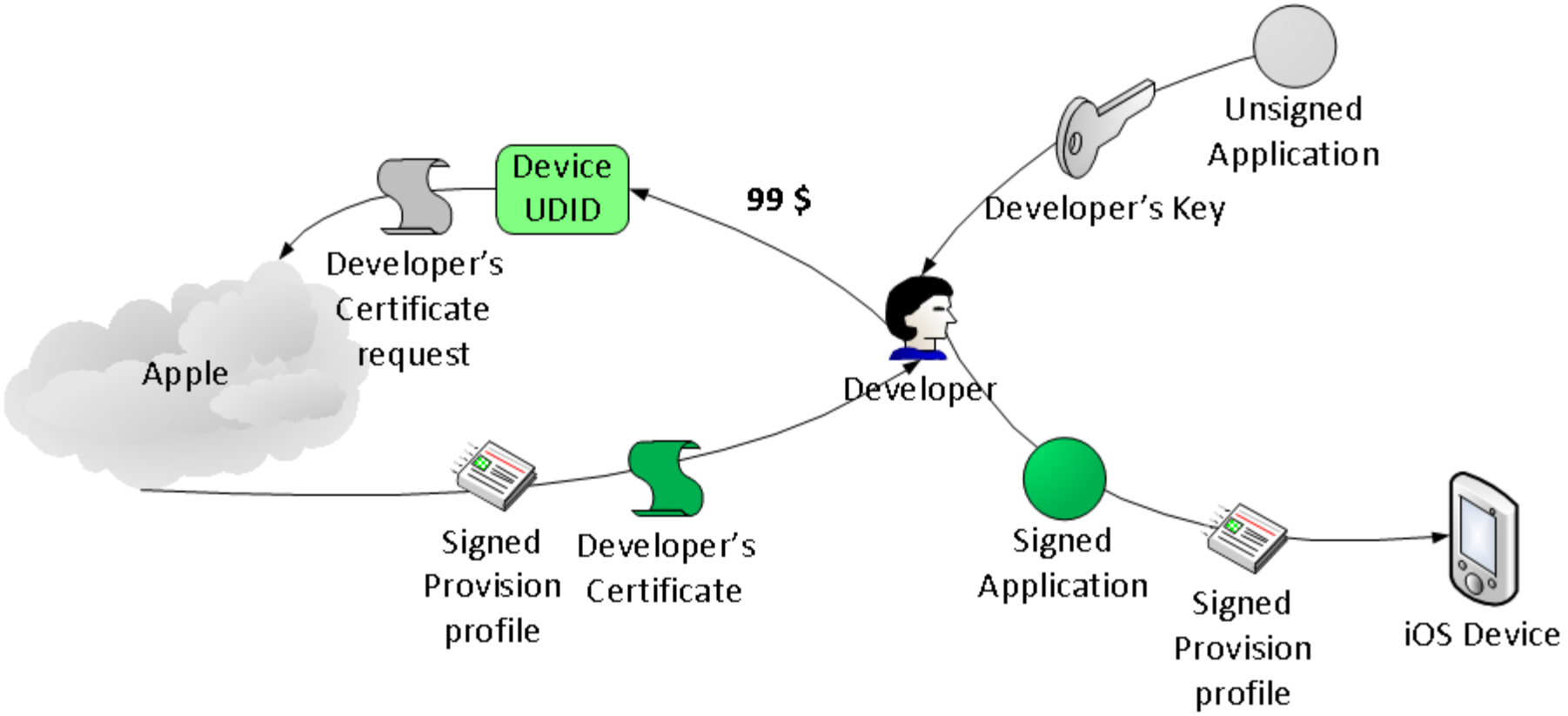


- CA – Apple Certificate Authority
- SIGN – Signature

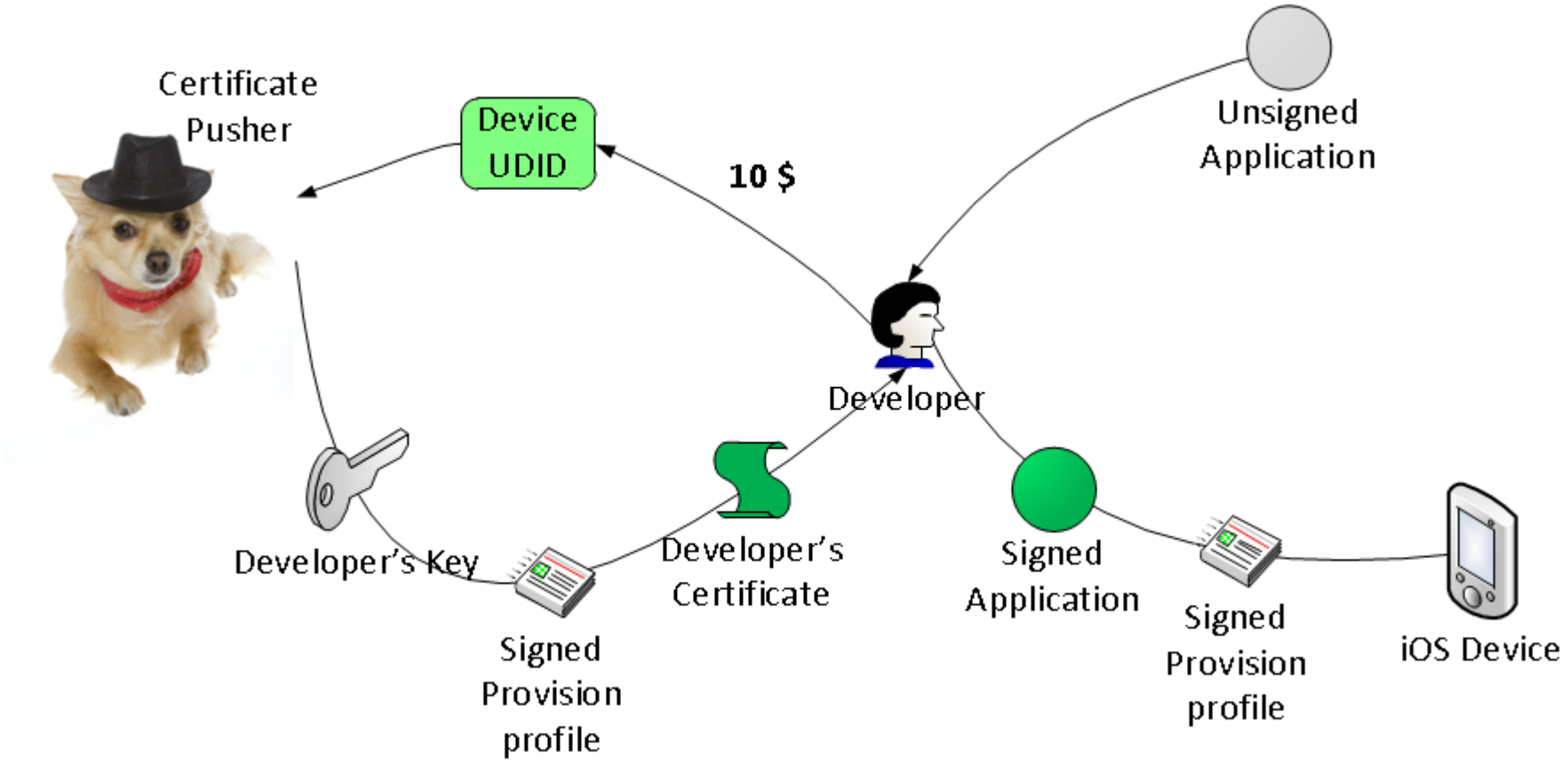
Files Protection



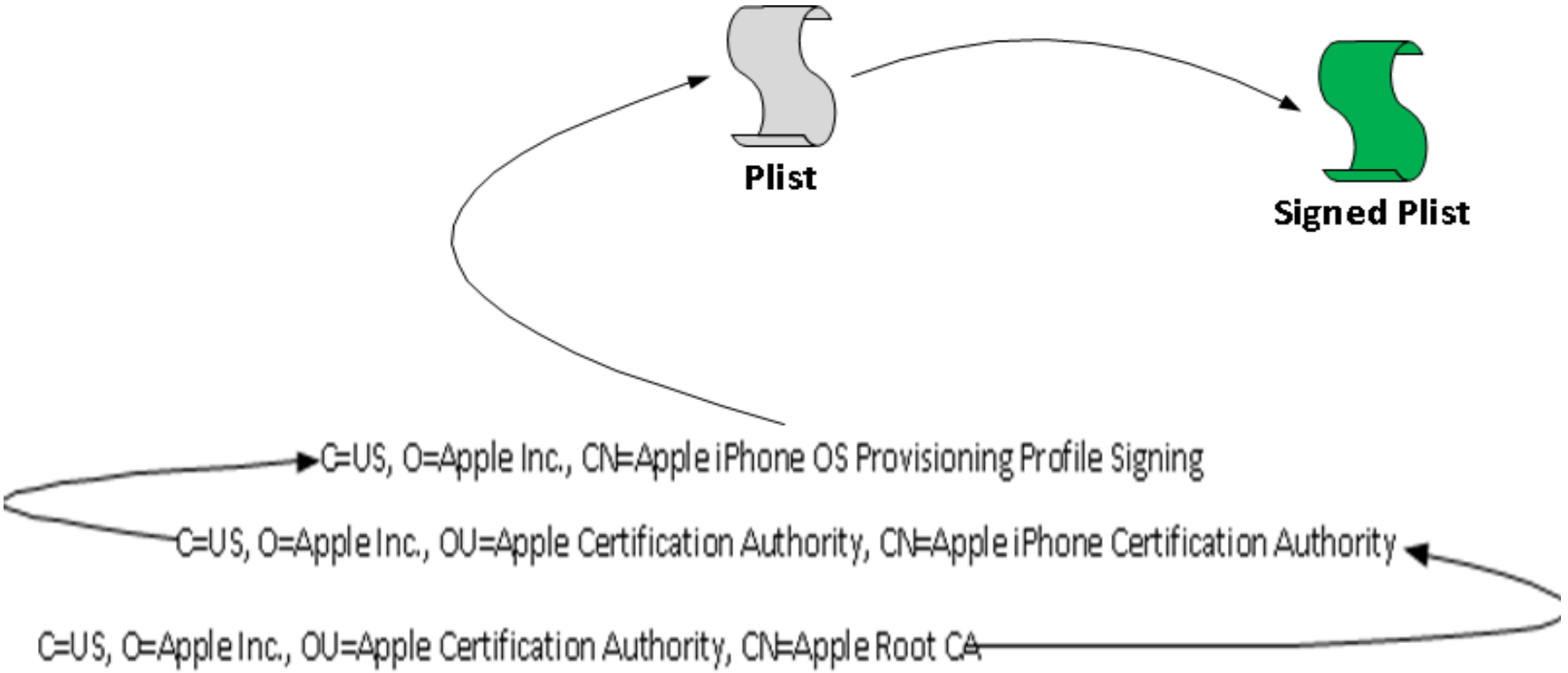
Classic provisioning



Actual provisioning

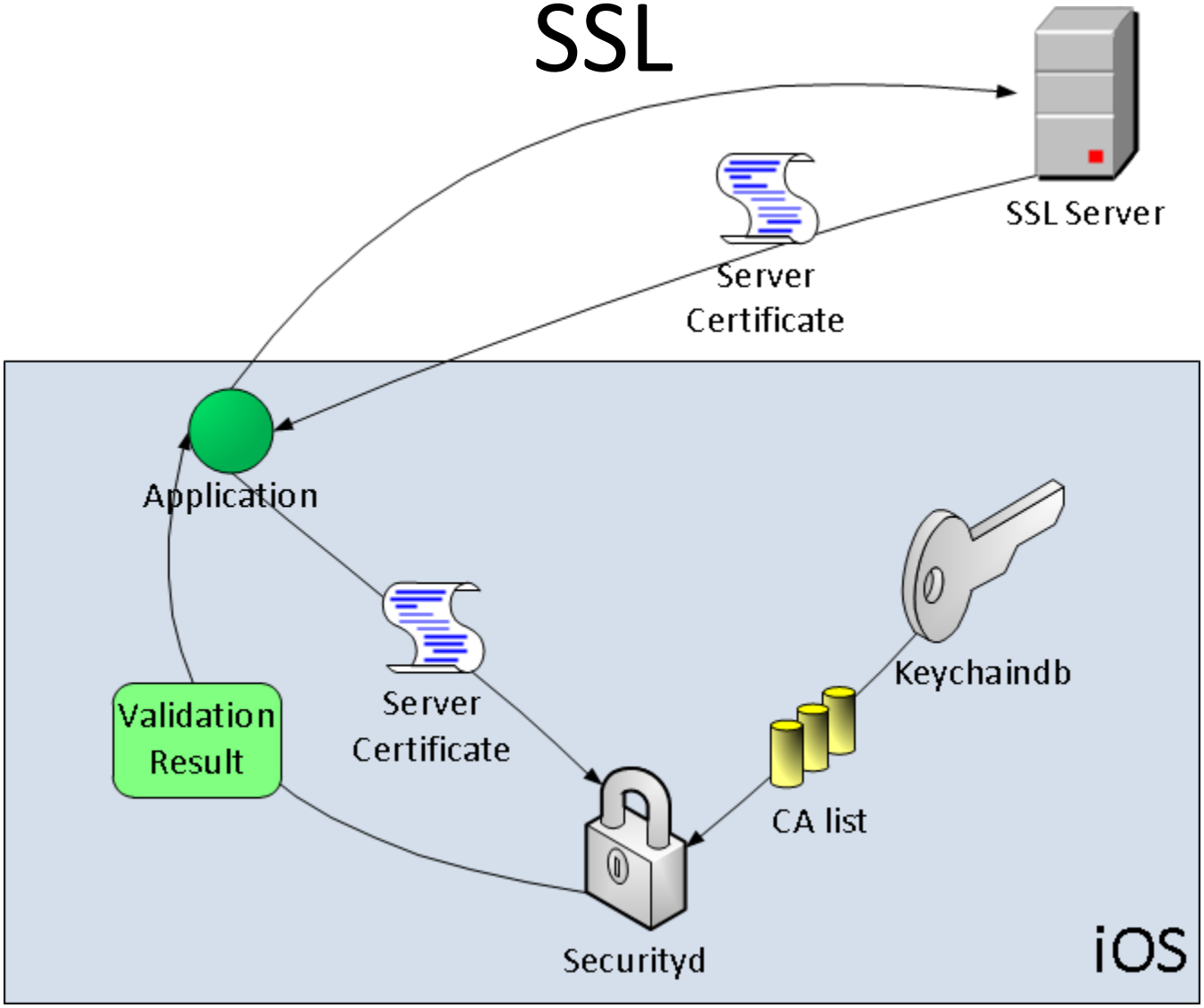


Why we can’t create fake signature?

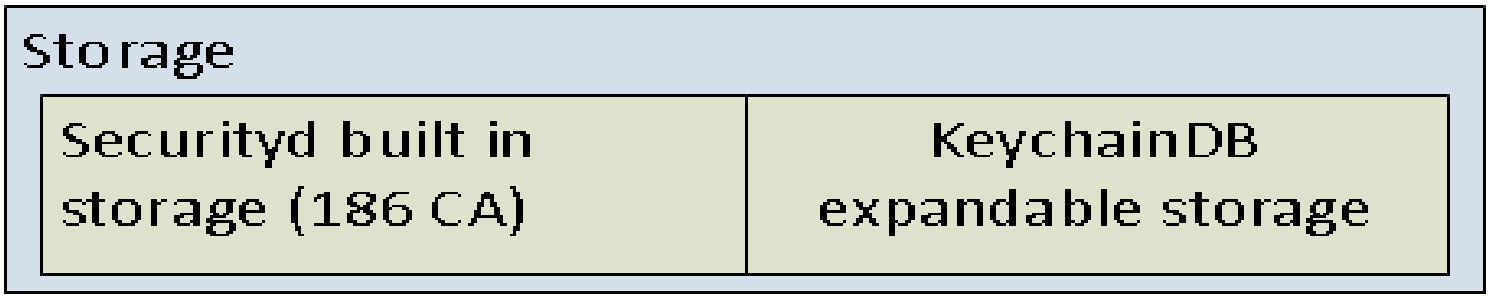


Because “Apple Root CA” fingerprint hardcoded into iOS and have to be **61:1E:5B:66:2C:59:3A:08:FF:58:D1:4A:E2:24:52:D1:98:DF:6C:60**

SSL


SSL Server
Server
Certificate
Application
Validation
Result
Server
Certificate
Keychaindb
CA list
Securityd
iOS

Certificate Authority Storage

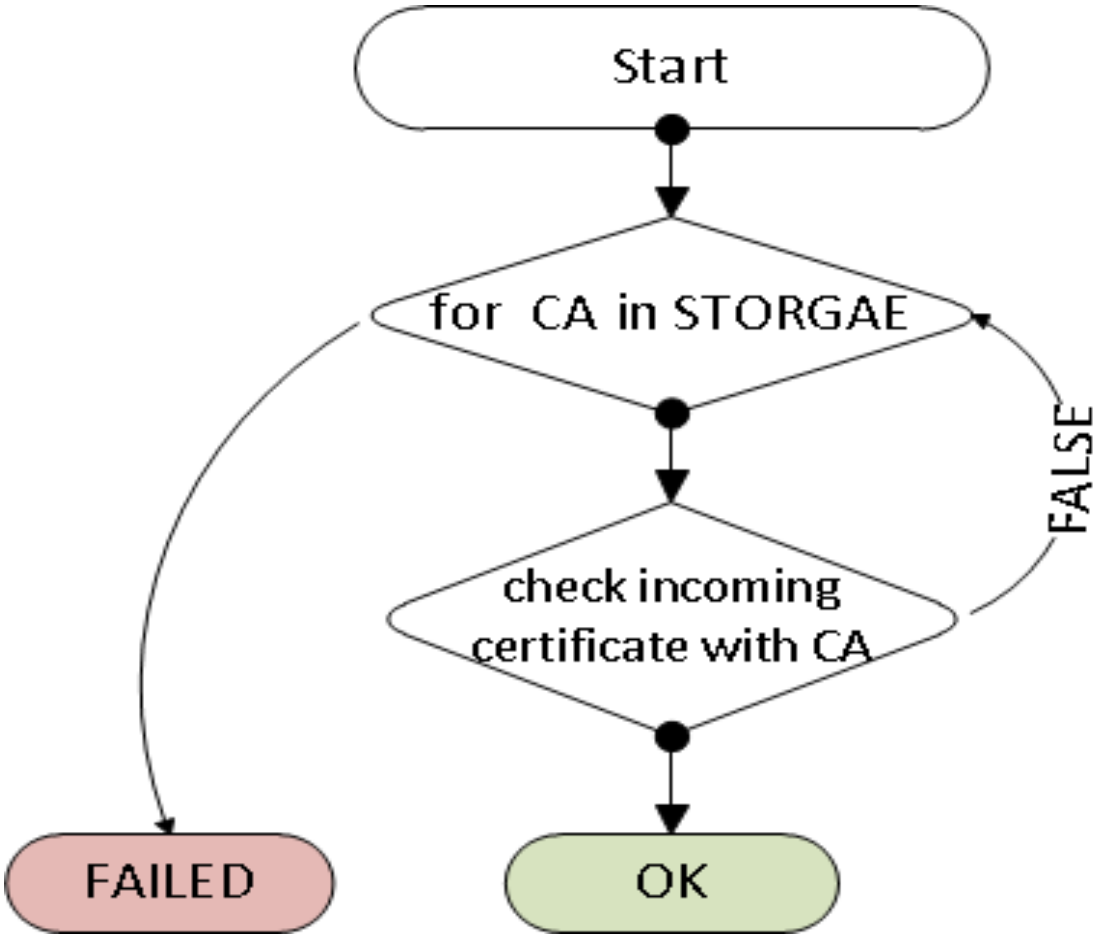


Few from 186 are quite interesting :

- C=US, O=U.S. Government, OU=**DoD**, OU=PKI, CN=DoD CLASS 3 Root CA
- C=JP, O=Japanese Government, OU=ApplicationCA
- C=CN, O=China Internet Network Information Center, CN=China Internet Network Information Center EV Certificates Root

…

Certificate authentication

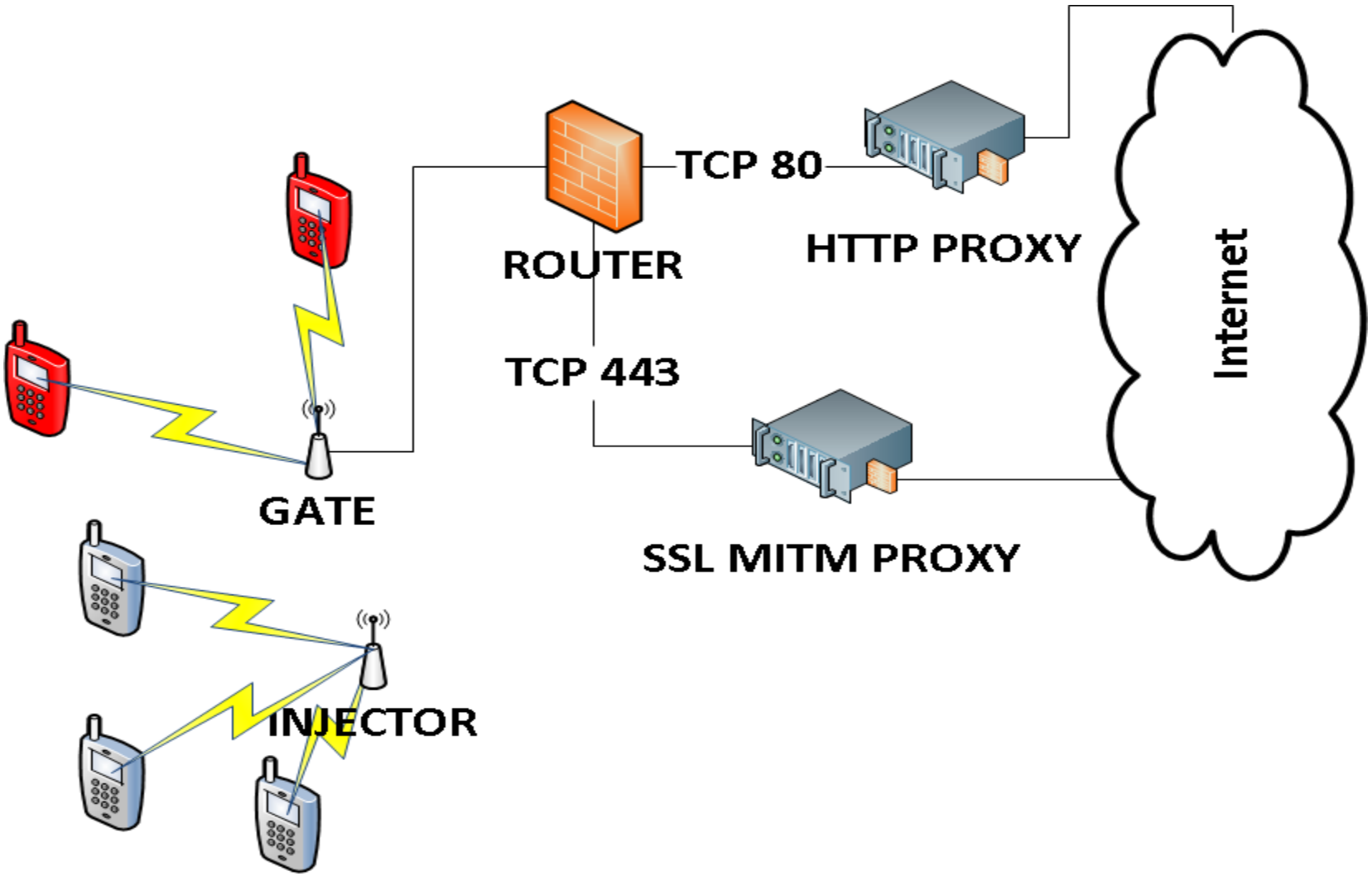


I want my CA in your iOS

Ways to install CA in iOS

- Safari
- Email attachment
- MDM
- ✓ With configuration profile
  - ➢ Can be installed with Safari

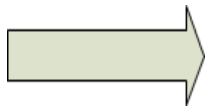
Attack

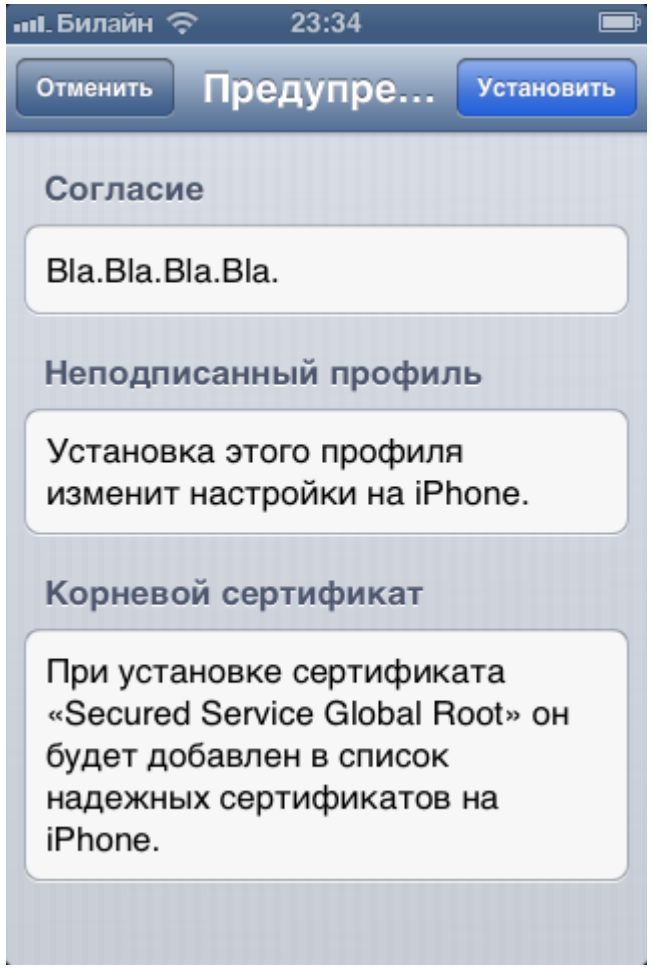

TCP 80
ROUTER
HTTP PROXY
Internet
TCP 443
GATE
SSL MITM PROXY
INJECTOR

Mobileconfig contains

- ✓ WiFi settings (pass, SSID) for “Gate”
- ✓ CA
- ✓ Proxy Settings, if we want victim’s traffic even it has left attack range. (Only for iOS6)
- ✓ iCloud backup (enable it, if not)

Mobileconfig installation





Looks bad =(

Let’s take a look on default CA list...

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            4e:81:2d:8a:82:65:e0:0b:02:ee:3e:35:02:46:e5:3d
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO
        Validity
            Not Before: Dec  1 00:00:00 2006 GMT
            Not After : Dec 31 23:59:59 2029 GMT
```

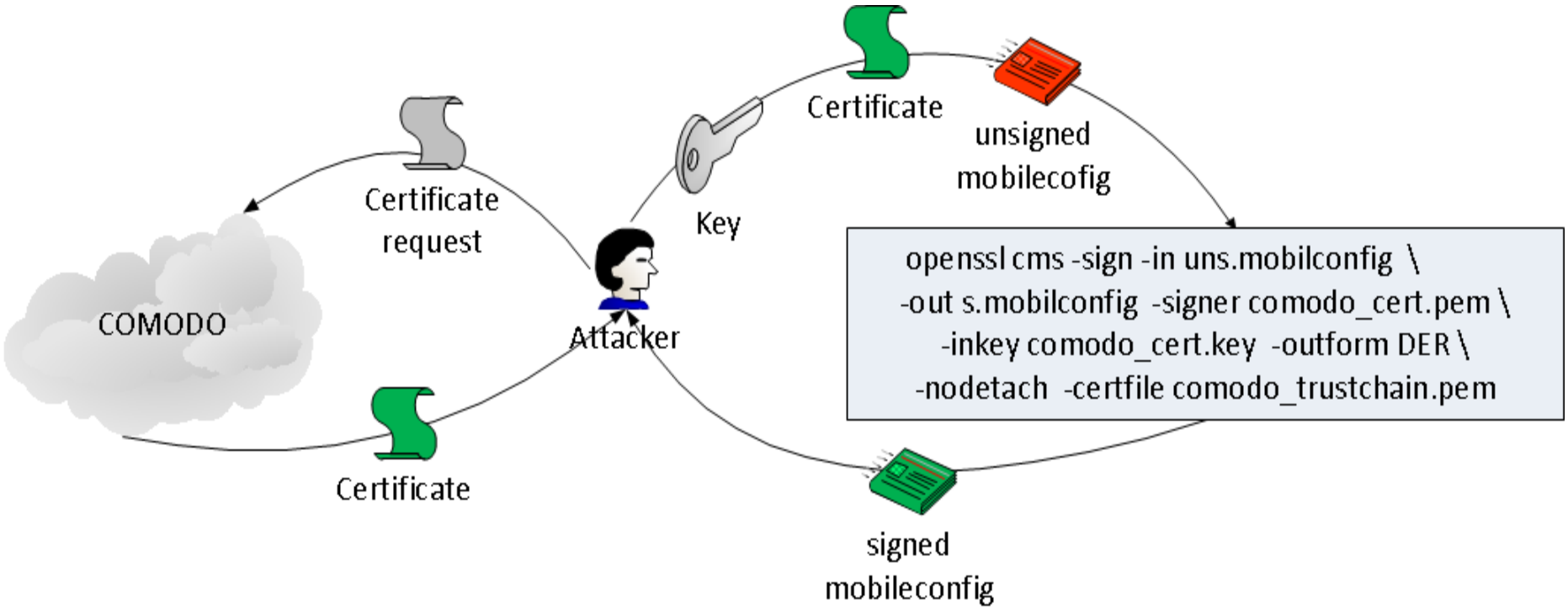
COMODO trial certificate

- You only need valid admin@yourdomain.com mail for confirmation
- Can be used for signing

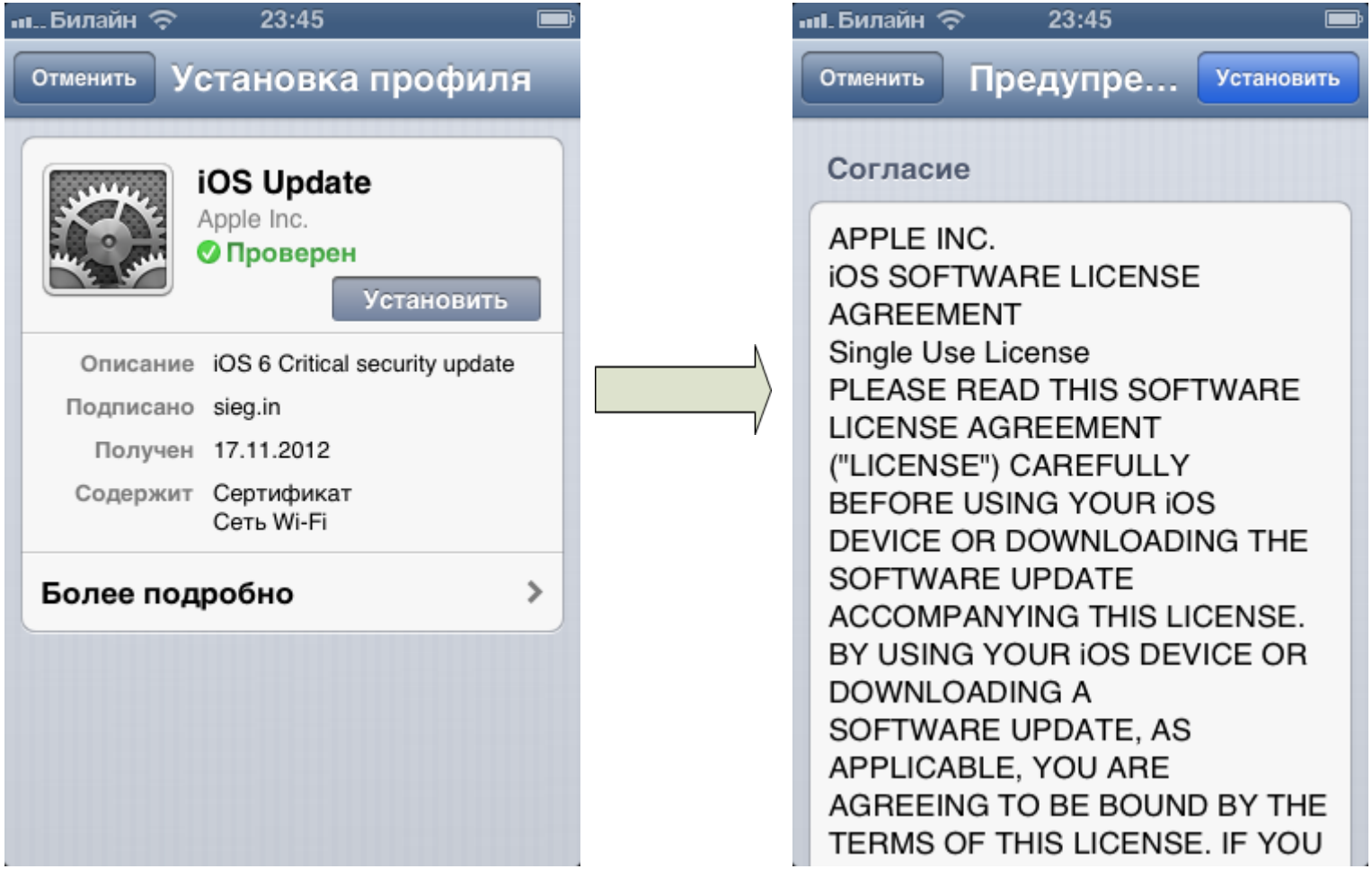

```
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:DA:CB:EA:AD:5B:08:5D:CC:FF:FC:26:5

    X509v3 Subject Key Identifier:
        C4:C8:4E:8B:8B:06:8F:49:10:3E:EB:7A:23:2
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Basic Constraints: critical
        CA:FALSE
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication, Microsoft
```

How to sign

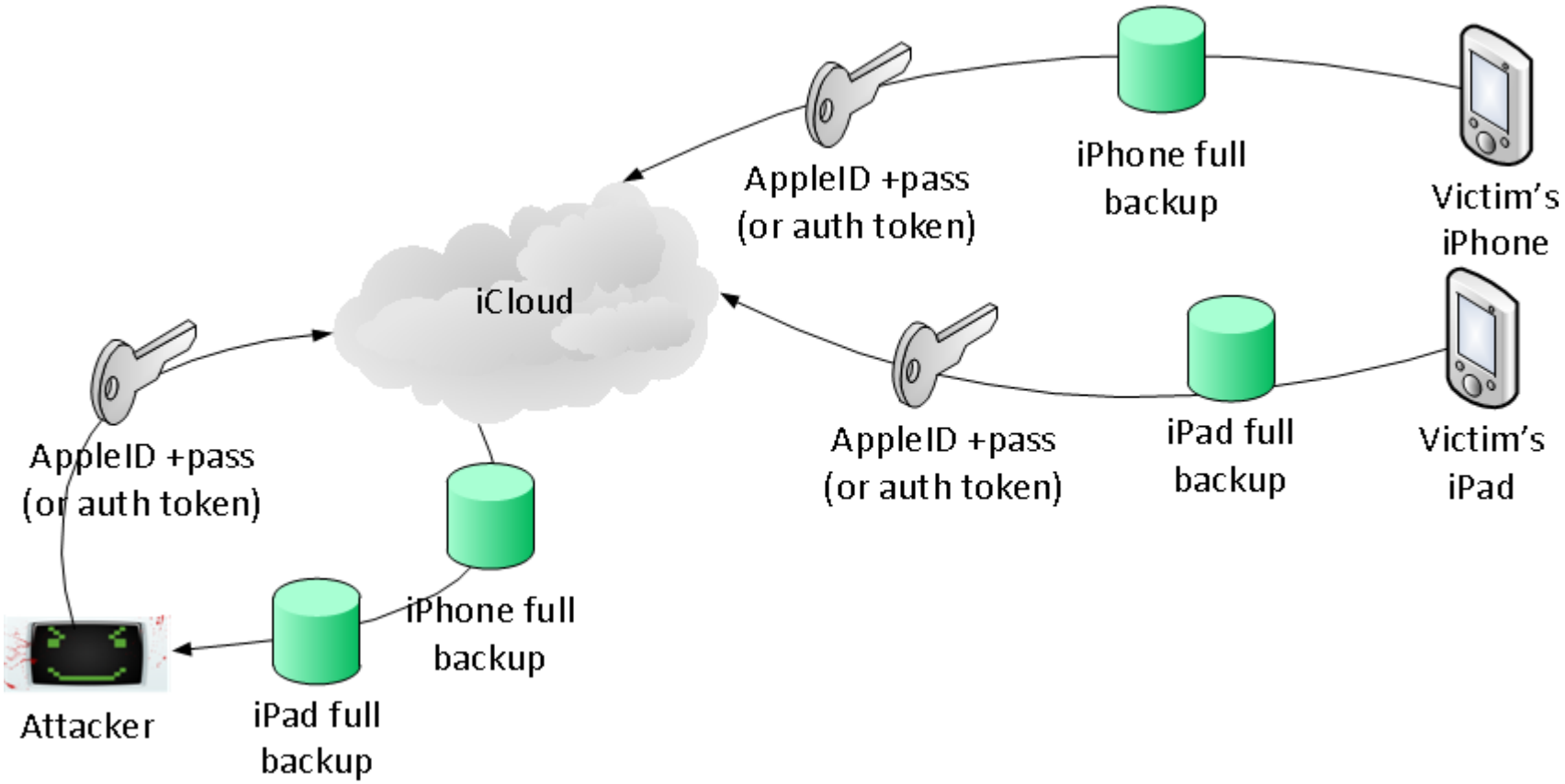

Certificate
unsigned
mobilecofig
Certificate
request
Key
COMODO
Attacker
openssl cms -sign -in uns.mobilconfig \
-out s.mobilconfig -signer comodo_cert.pem \
-inkey comodo_cert.key -outform DER \
-nodetach -certfile comodo_trustchain.pem
Certificate
signed
mobileconfig

Looks much better

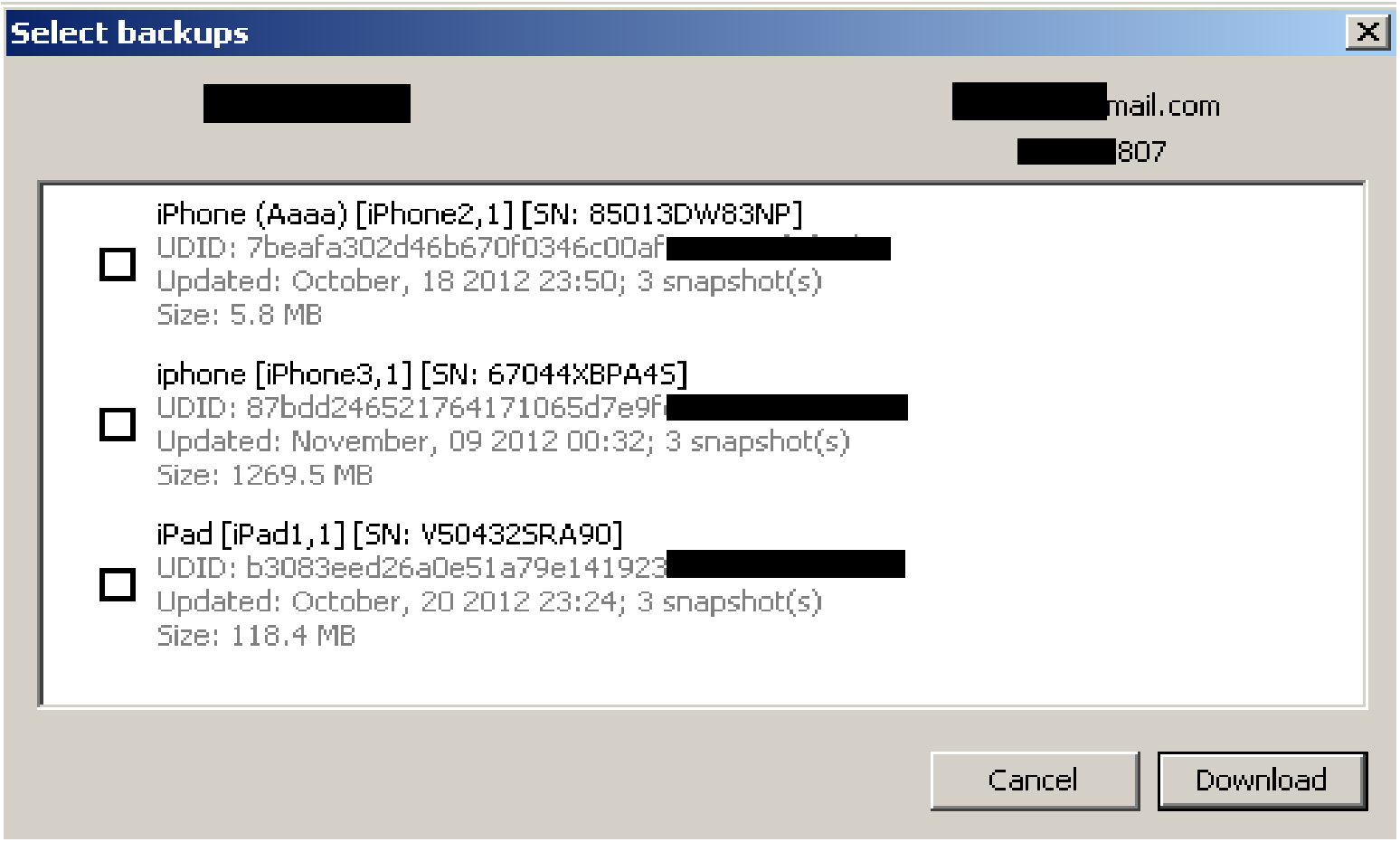


SSL Defeated
But we want more

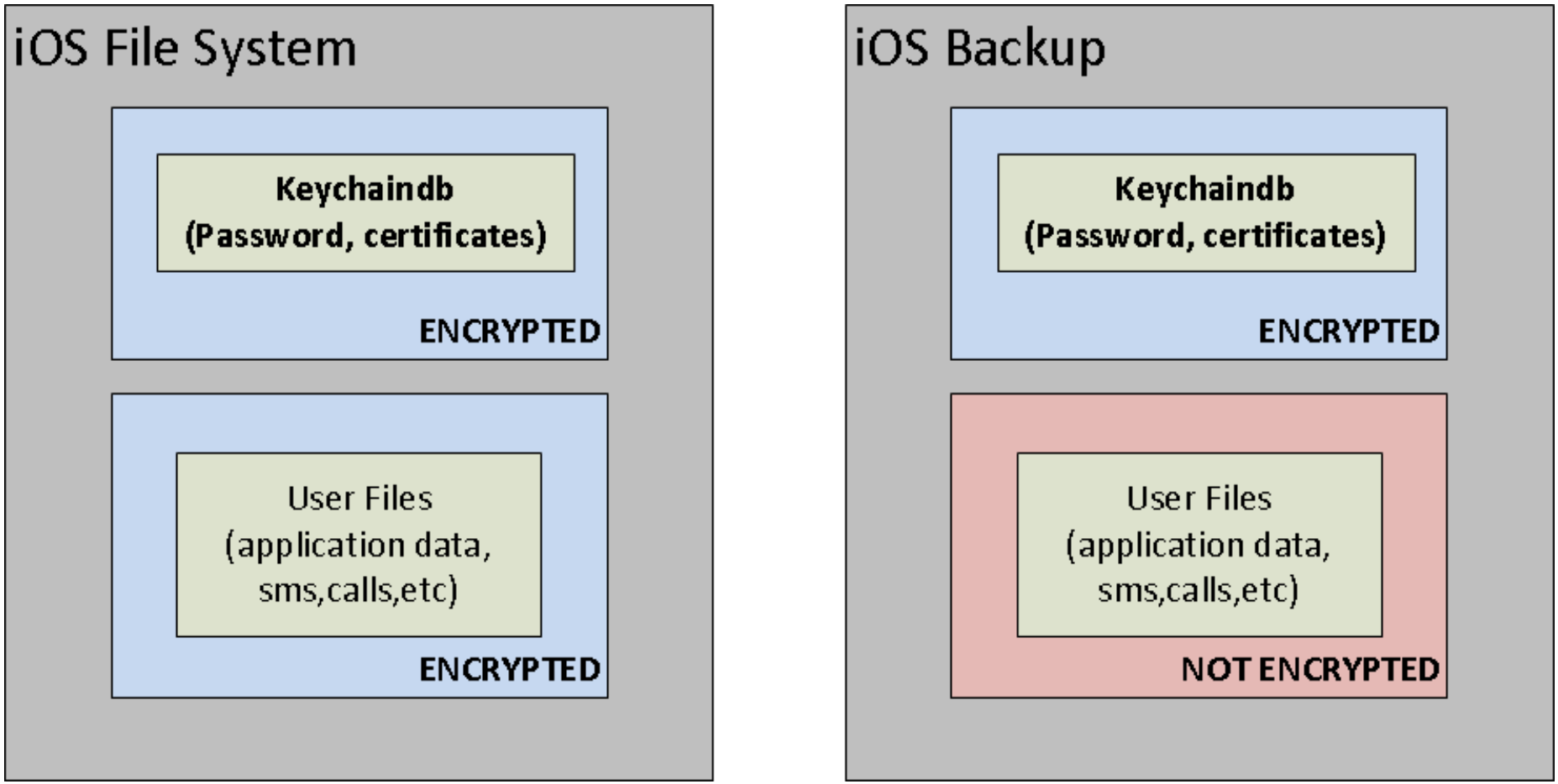
How to get files from device



Elcomsoft Phone Password Breaker

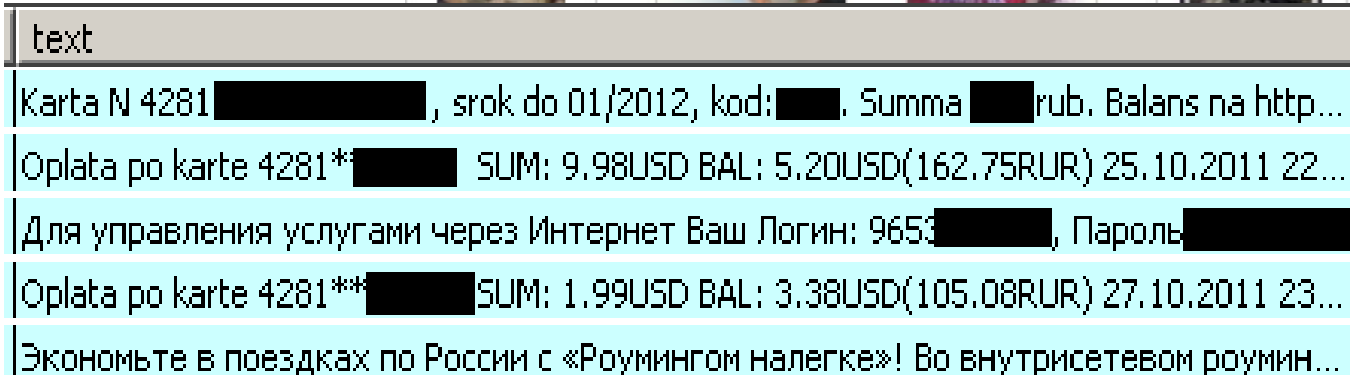


Once again


iOS File System
Keychaindb
(Password, certificates)
ENCRYPTED
User Files
(application data,
sms,calls,etc)
ENCRYPTED
iOS Backup
Keychaindb
(Password, certificates)
ENCRYPTED
User Files
(application data,
sms,calls,etc)
NOT ENCRYPTED

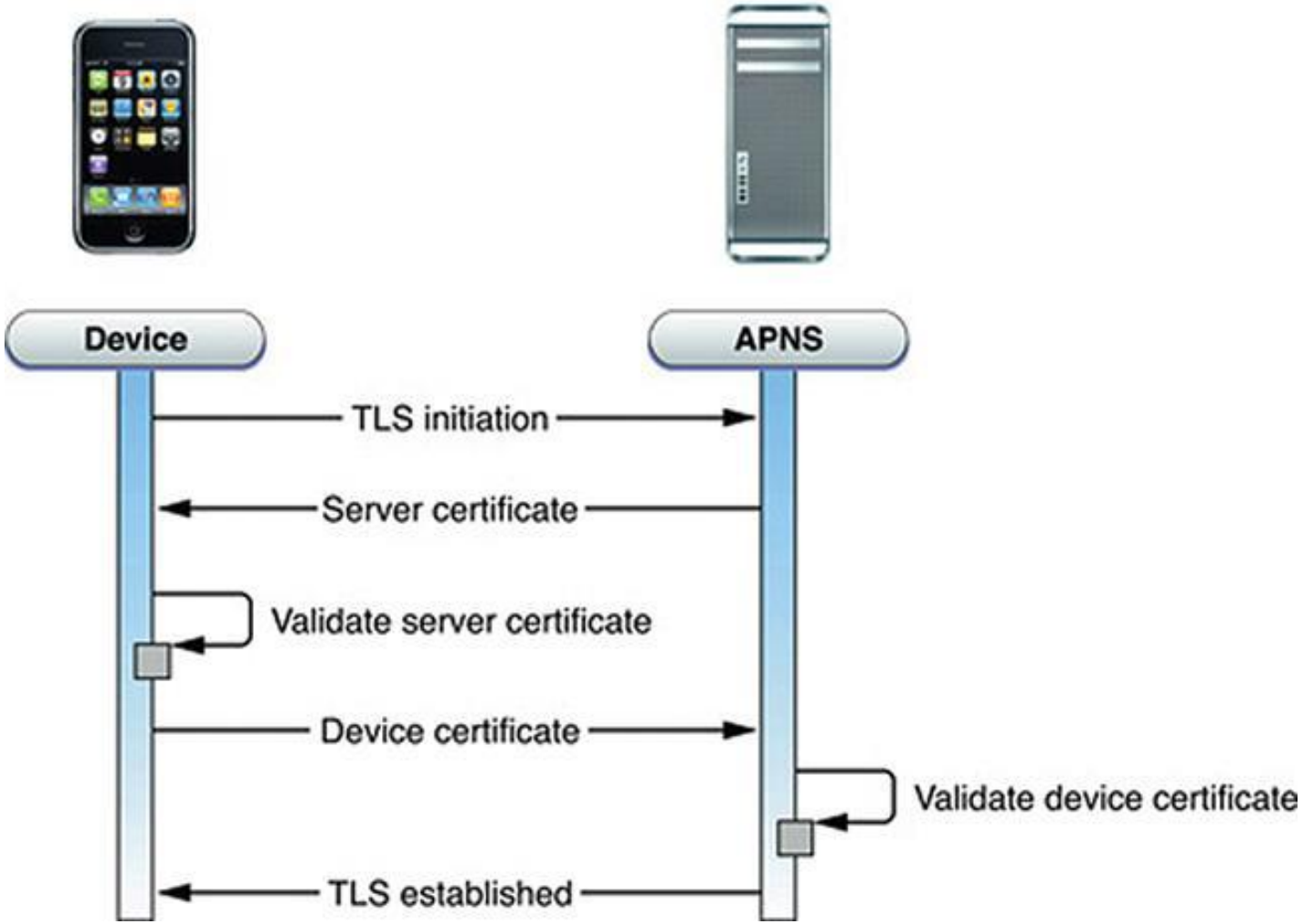
What's in backup?

- SMS
- Private photo
- Emails
- Application data
- And more …

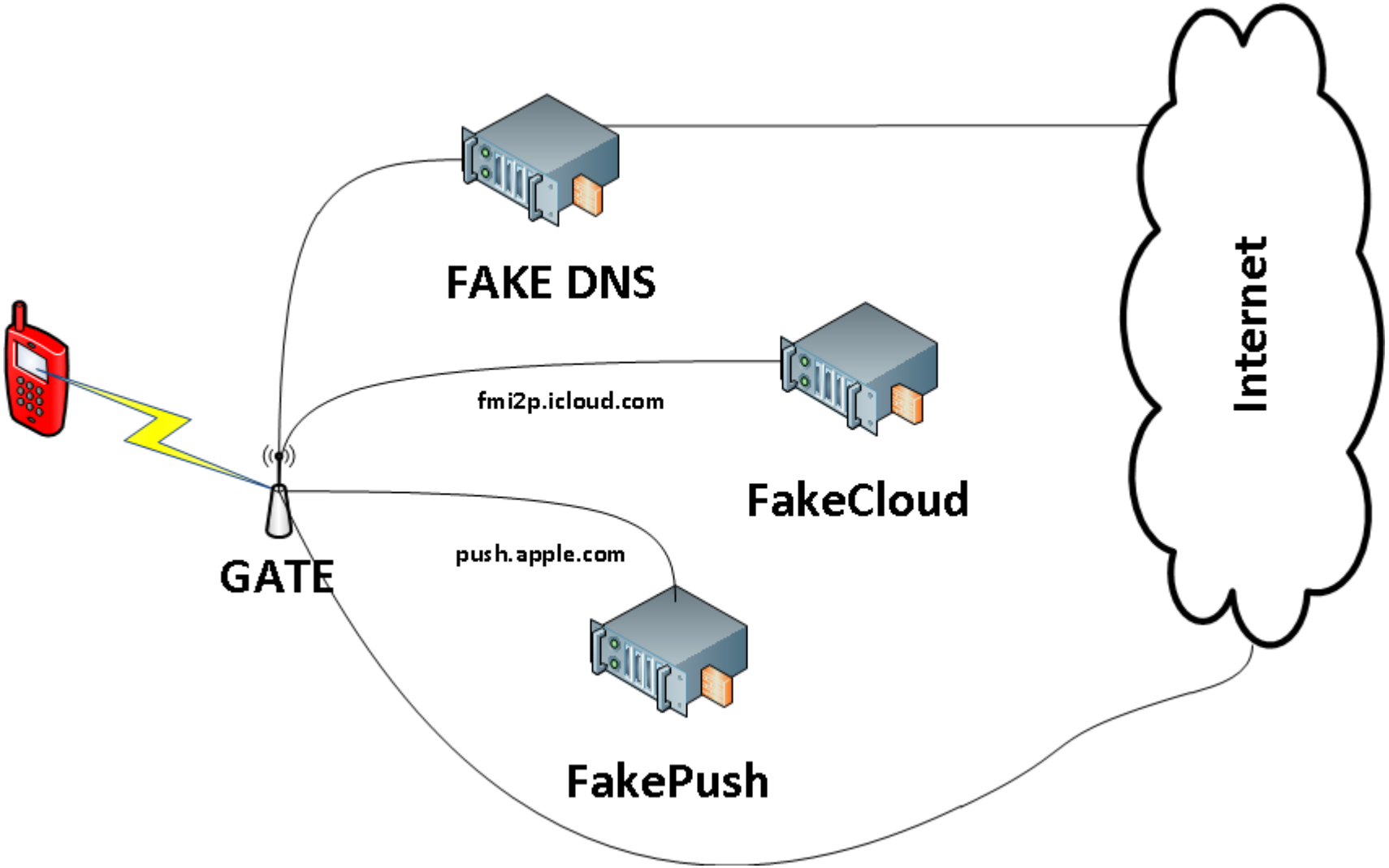


Files done
But we want more

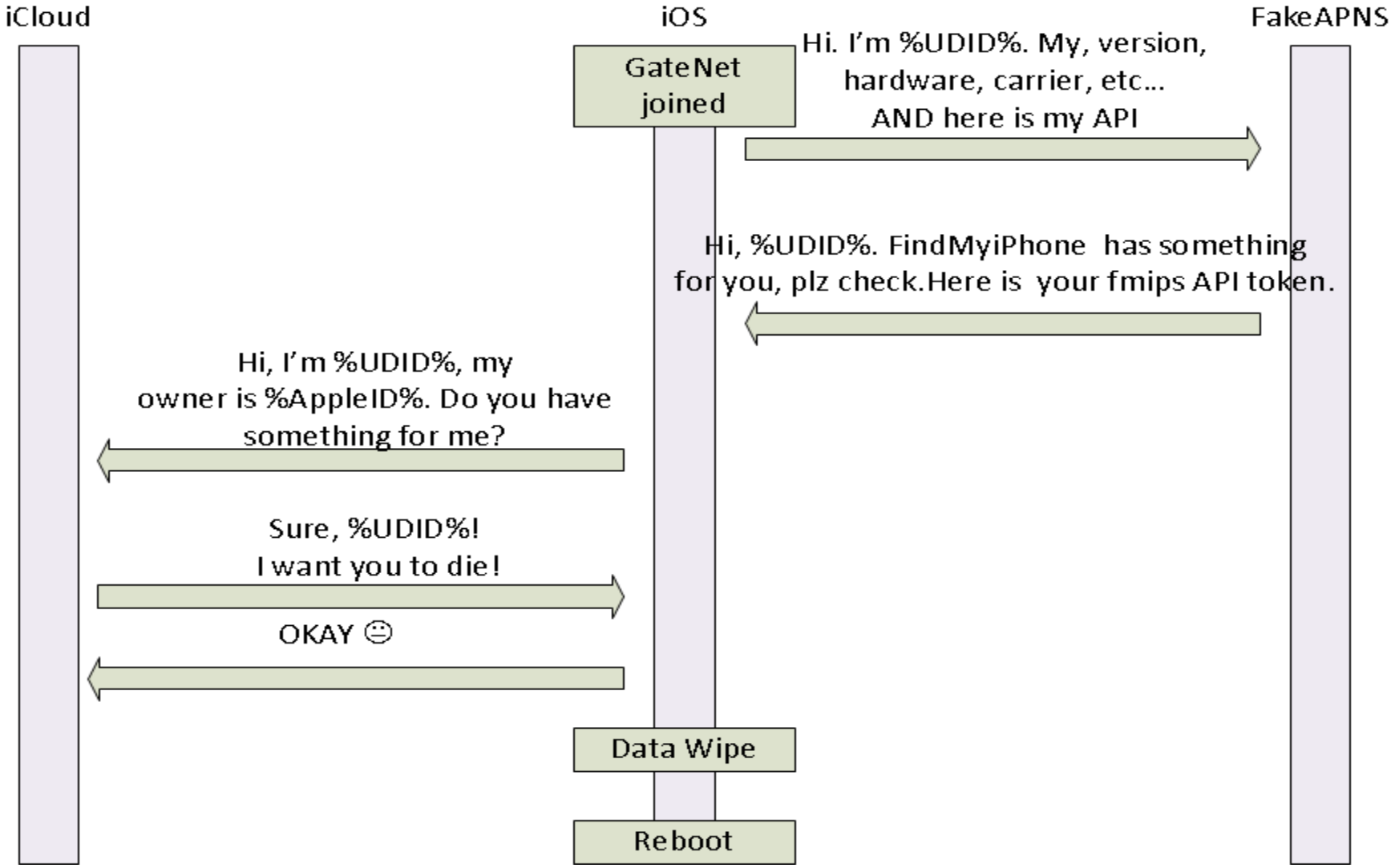
Apple Push Notification Service



Fake! Fake! Fake!



Wipe Tragedy (act 1/1)



Summary

User only have to tap ‘Install’ two times to make us able to :

- Sniff all his SSL traffic (cookies,passwords, etc)
- Steal his backup (call log, sms log, photos and application data)
- Send him funny push messages or just wipe device

sieg.in
al@sieg.in
@siegin