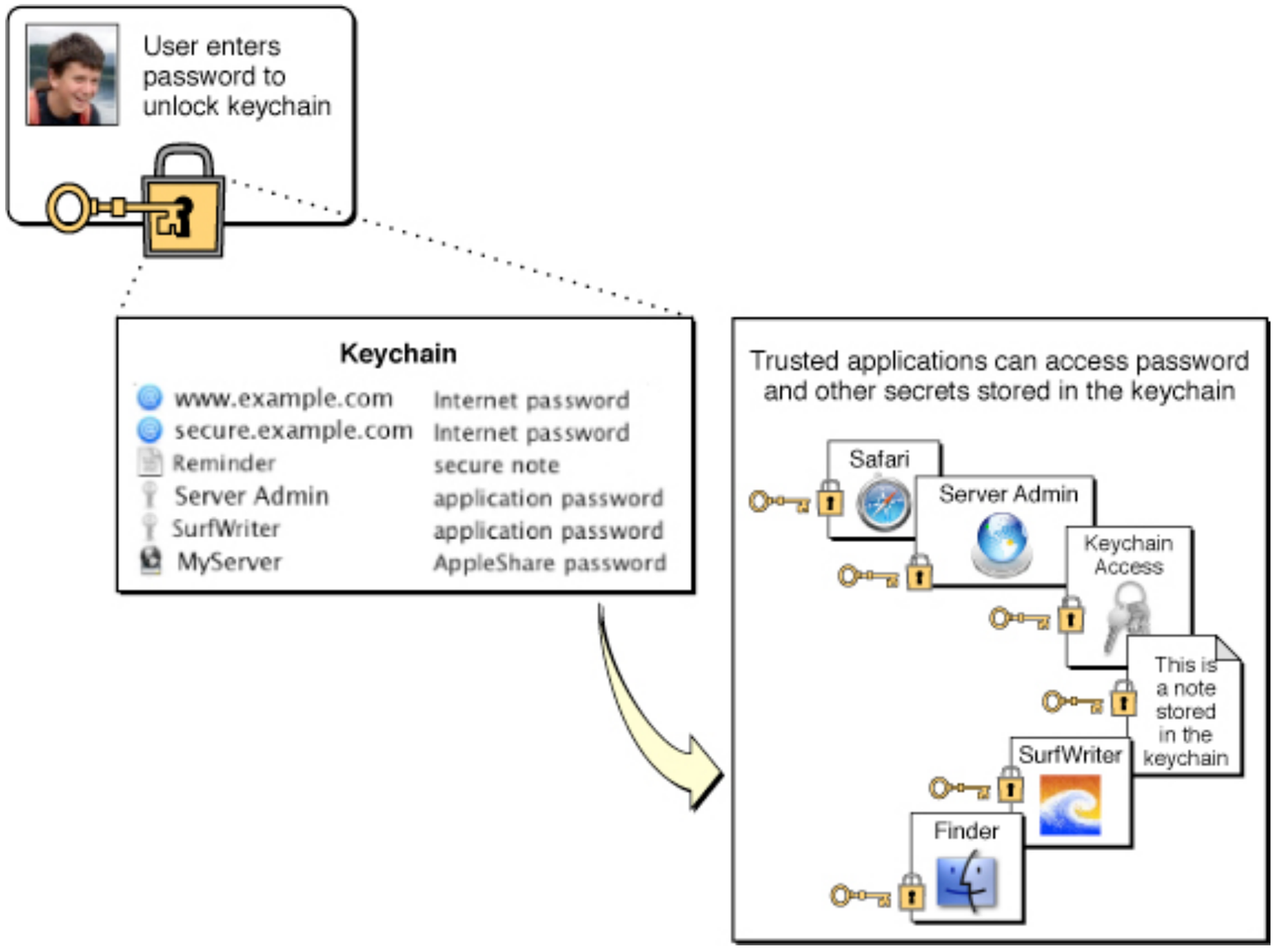
Mac OS X Forensic Artifacts

Keychain Analysis

keychain

- Keychain is password management system in Mac OS
- Introduced with Mac OS 8.6 ~
- keychain can contain various types of data
- File Format : Apple Database

keychain concept



reference: https://developer.apple.com/library/mac/#documentation/security/conceptual/keychainServConcepts/02concepts/concepts.html

keychain concept

keychain storage

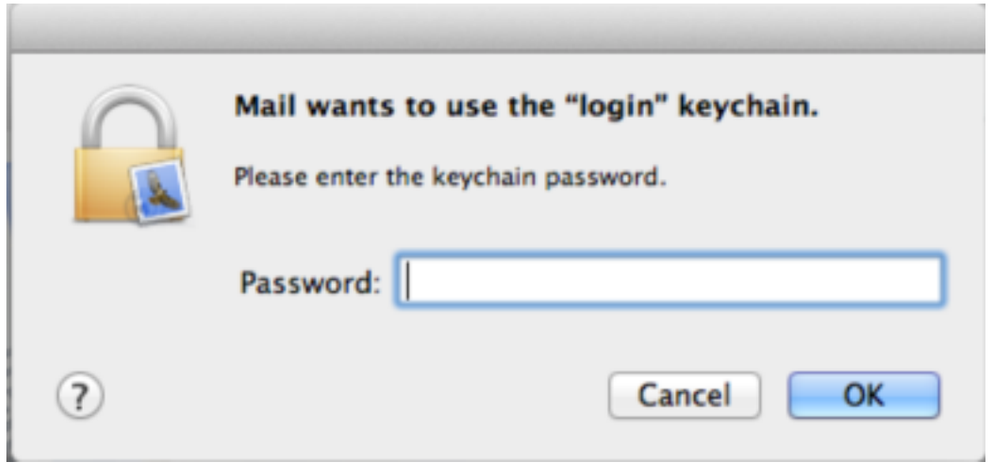
- ~/Library/Keychains
- /Library/Keychains
- /Network/Library/Keychains

Keychain Locking

- 사용자로 패스워드 인증을 성공하면 해당 사용자의 login.keychain을 사용할 수 있음.
- 로그아웃하거나 키체인을 잠그는(locking) 경우에는 다시 사용자 패스워드를 입력해야 키체인 데이터에 접근할 수 있음.

Keychain Unlocking

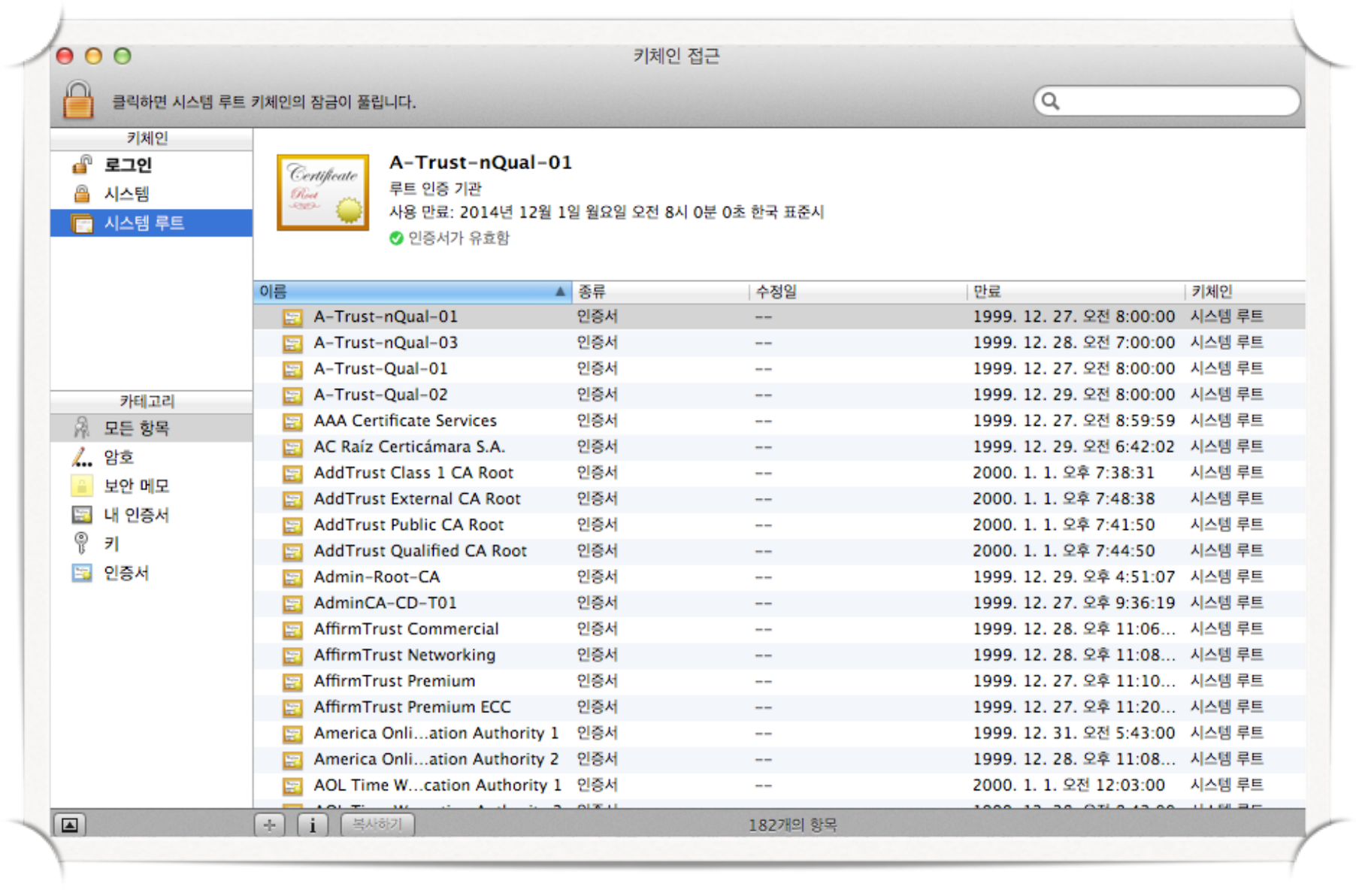
- 새로운 이메일 계정 등 키체인에 저장할 정보를 선택할 경우 다음과 같은 다이얼로그 제공
- 패스워드 입력 후에 자동으로 Unlocking 됨



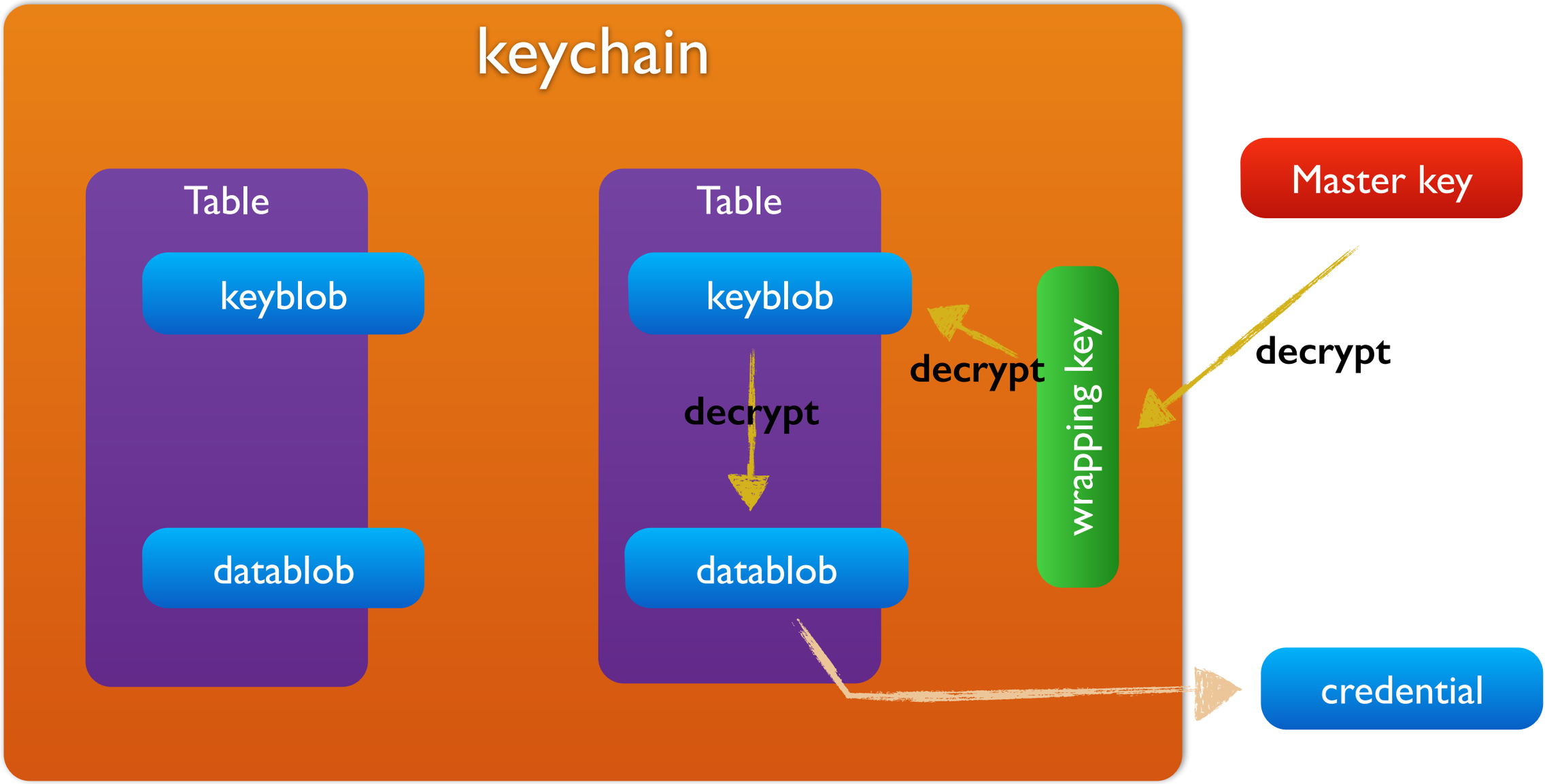
Keychain Tool

- Keychain Access - GUI
- /usr/bin/security - Console
  - list-keychains, dump-keychain ....

Keychain Access



Keychain Structure



Key management

- Encrypted Algorithm: 3DES-CBC
- Master Key : Password Based generation
- Wrapping Key : Master key로 암호화
- KeyBlob :Wrapping key로 암호화
- DataBlob : KeyBlob에서 복호화한 키로 암호화

keychaindump

Breaking into the OS X keychain

September 5, 2012. Tagged security, osx, keychain.

***Update**: I want to clear up some misconceptions. This is not a security bug in OS X. Everything works as designed. The point of this post was to show a post-exploitation technique and to release a tool for the job. I found this particular technique interesting because it is instantaneous, reliable across OS X versions, and requires no persistent changes in the system.*

TL;DR: Root can read plaintext keychain passwords of logged-in users in OS X. Open source proof-of-concept.

There is a design compromise in Apple's keychain implementation that sacrifices some security for a lot of usability.

keychaindump

- 발생조건
  - 시스템 관리자 권한이 필요함.
  - 시스템에 바이너리를 업로드하고 구동해야함.

- 발생조건
  - 사용자가 키체인을 언락해야함.

keychaindump

- vmmap을 이용하여 securityd 프로세스의 메모리 영역에서 MALLOC_TINY 영역을 찾는다.
- MALLOC_TINY 영역에서 마스터키 구조체를 찾는다.
- 구조체의 내용을 기반으로 마스터키 후보군을 선정한다.
- 각 마스터키 후보군을 이용하여 wrapping key가 올바르게 복호화되는지 확인한다.

Security Server

- Security Server(securityd) is daemon running in OS X and iOS that implements several security protocol

- *https://developer.apple.com/library/mac/ipad/#documentation/Security/Conceptual/Security_Overview/Architecture/Architecture.html*

keychaindump

$ sudo vmmap SECURITYDPID

securityd

MALLOC_TINY

MALLOC_TINY

MALLOC_TINY

MALLOC_TINY

matching

keychain analysis routine

```
struct _masterkey {
uint32 masterkeybit;
int* masterkeyaddr;
}
```

keychaindump

- DEMO - keychaindump 바이너리를 이용하여 해당 시스템에서 키체인 정보를 추출

keychaindump

- 이 데이터를 메모리에서 찾아보면 어떨까?

volafox : keychaindump

- volafox의 keychaindump는 security daemon의 메모리 영역을 분석할 수 있음.
  - 이 중 MALLOC_TINY 영역만 찾으면 동일한 분석이 가능

  - MALLOC_TINY 영역은 항상 1MB(1024KB)의 크기를 가지므로 이 영역에서 후보를 찾으면 가능

volafox : keychaindump

```
$ python vol.py -i key.mem -o keychaindump
```

securityd

MALLOC_TINY

MALLOC_TINY

MALLOC_TINY

MALLOC_TINY

```
[*] master key candidate: ACEFBB5CECE8398E780016DAF685E39A8E36FDE8EC8B6B6E
[*] master key candidate: C7A8DC12E05F1BEEF70228C304E6567CFDA07BEAB86606C7
[*] master key candidate: E340E90C2EC1E55BC116A2775624FC3D4ADD35E248E543C1
[*] master key candidate: 9AB40E7378E195335019C5A3577123D5AC814C7D22588B2F
[*] master key candidate: 9AB40E7378E195335019C5A3577123D5AC814C7D22588B2F
```

matching

```
struct _masterkey {
uint32 masterkeybit;
int* masterkeyaddr;
}
```

volafox : keychaindump

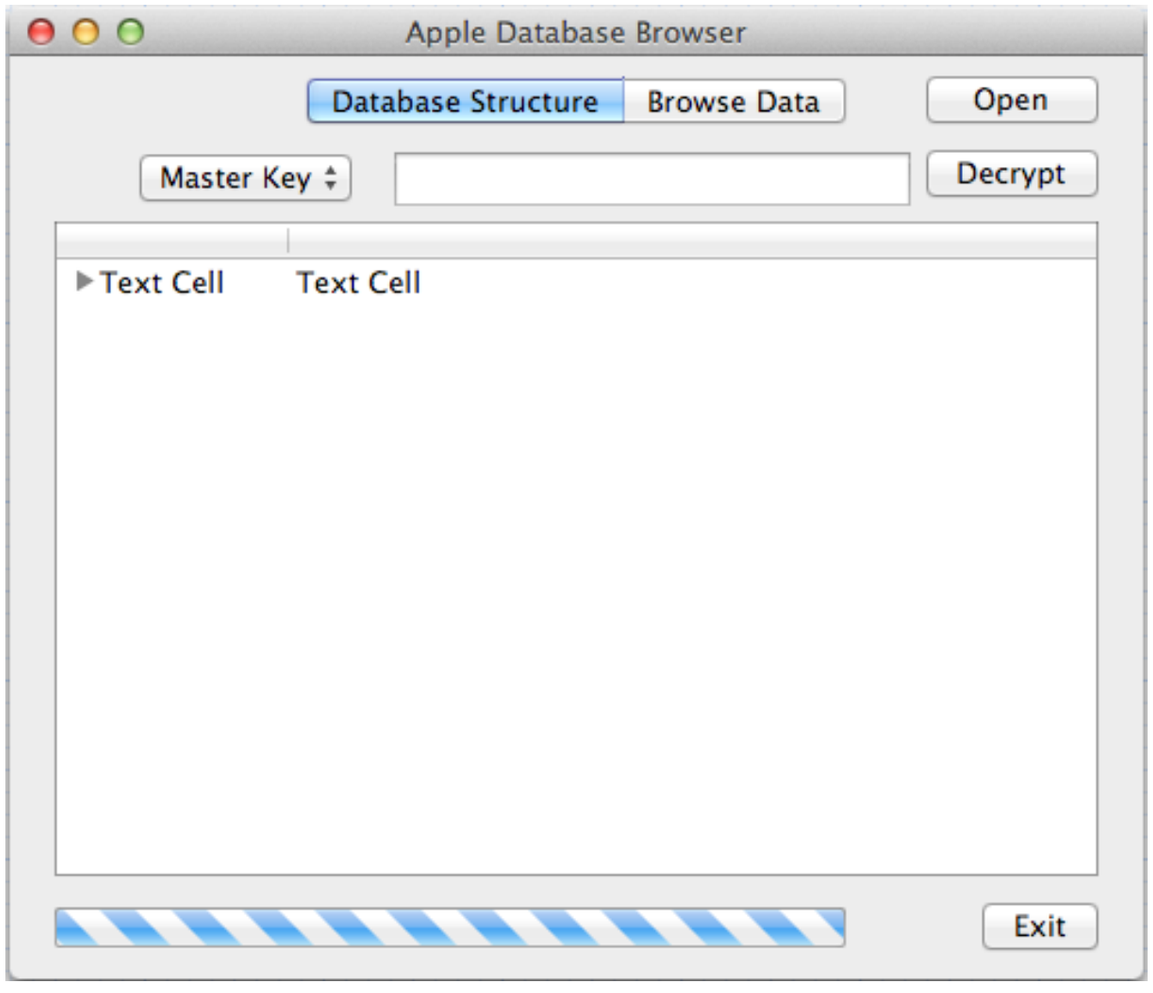
- DEMO - 메모리에서 마스터키 후보군을 추출하고 이 후보군을 이용하여 키체인 분석

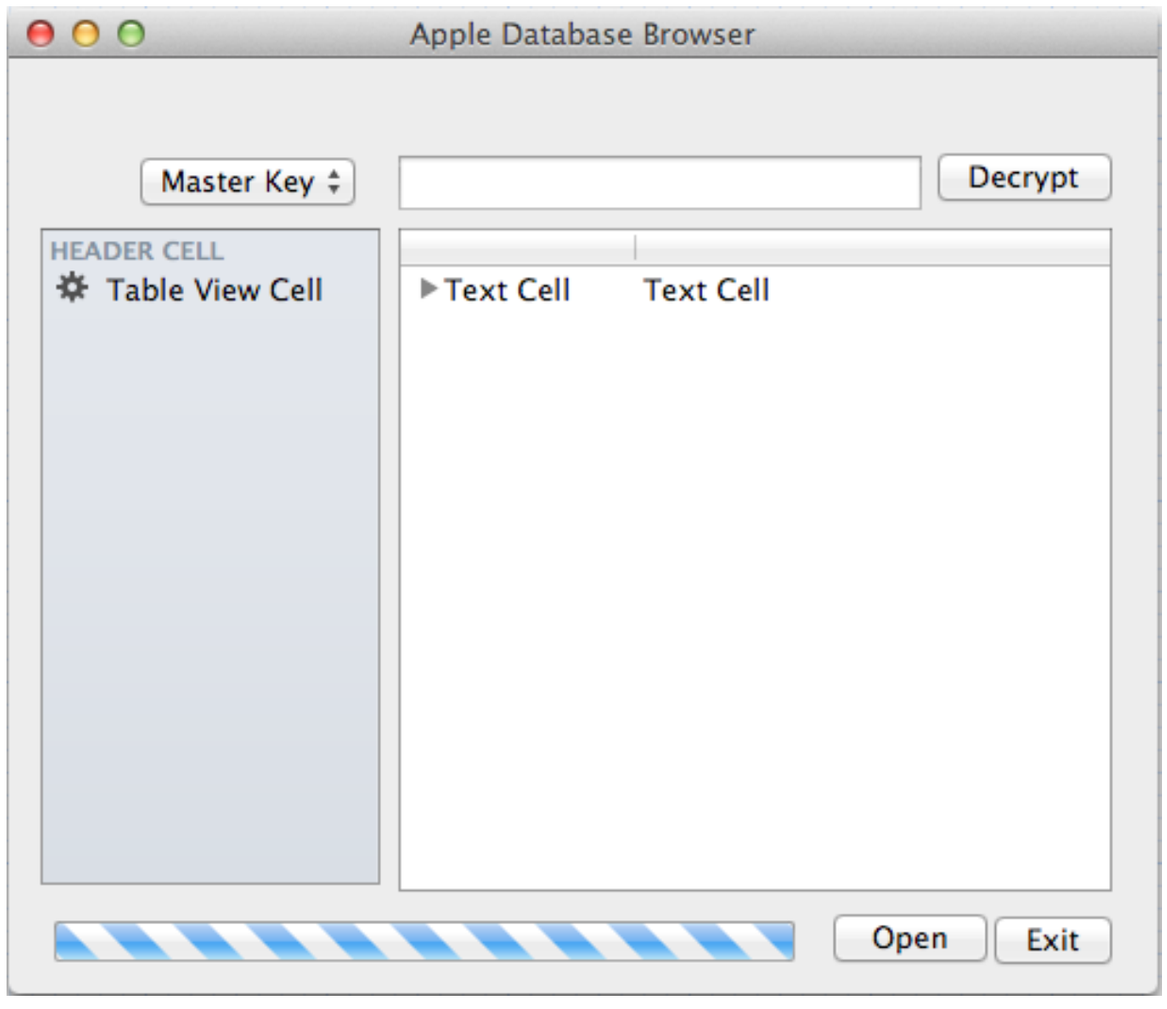
Limitation

- 키체인의 모든 데이터를 분석해주지는 않음.
  - WIFI Password 등 몇몇 정보를 복원하지 않음
- 두 방법 다 루트 권한을 필요로 함.
  - FW기반 이미징의 경우 10.7.2부터 사용할 수 없도록 변경 됨.

To-Do

- Keychain에서 아직 분석되지 않은 정보 분석





To-Do

- FileVault Master Key 추출 방법 연구
  - *Infiltrate the Vault: Security Analysis and Decryption of Lion Full Disk Encryption*
  - *http://eprint.iacr.org/2012/374.pdf*

Q & A

feedbeef.blogspot.com
rapfer@gmail.com