
Secrets
&
Lies

English Letter	English word	Navajo Word
C	Cat	MOASI
D	Dog	LHA-CHA-EH
E	Elk	DZEH


CC0, courtesy skeeze


Security
Through
Obscurity

JOURNAL

DES

SCIENCES MILITAIRES.

Janvier 1883.

LA CRYPTOGRAPHIE MILITAIRE.

LA CRYPTOGRAPHIE MILITAIRE. (1883)

- The system must be practically, if not mathematically, indecipherable;
- It should not require secrecy, and it should not be a problem if it falls into enemy hands;
- It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will;
- It must be applicable to telegraph communications;
- It must be portable, and should not require several persons to handle or operate;
- Lastly, given the circumstances in which it is to be used, the system must be easy to use and should not be stressful to use or require its users to know and comply with a long list of rules.

Security through
obscurity is
no
Security

Public domain, courtesy Wikimedia

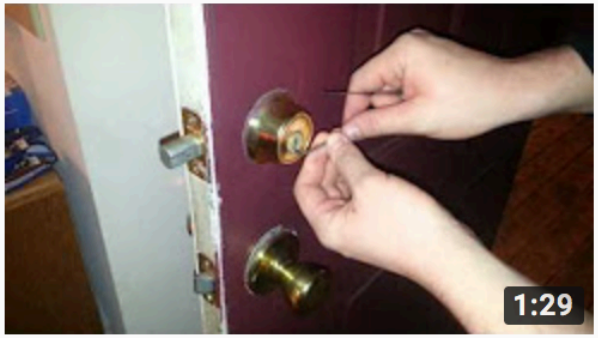
“Rogues are very keen in their profession, and know already much more than we can teach them.”

–Alfred Charles Hobbs

YouTube

lockpick deadbolt

About 15,500 results



How to Pick a Front Door Lock Deadbolt EASY

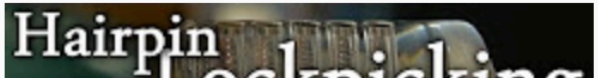
knifeandbladereviews • 357K views • 5 years ago

Easiest way to pick a **deadbolt** lock on a front door.

Lock picking a deadbolt fast!

charlieaf92 • 99K views • 7 years ago

A close up look at picking a brand name (kwikset) **deadbolt** lock. NEW VIDEO POSTED: Step by step instructions to pick your first ...



How to Pick a Lock With Hairpins

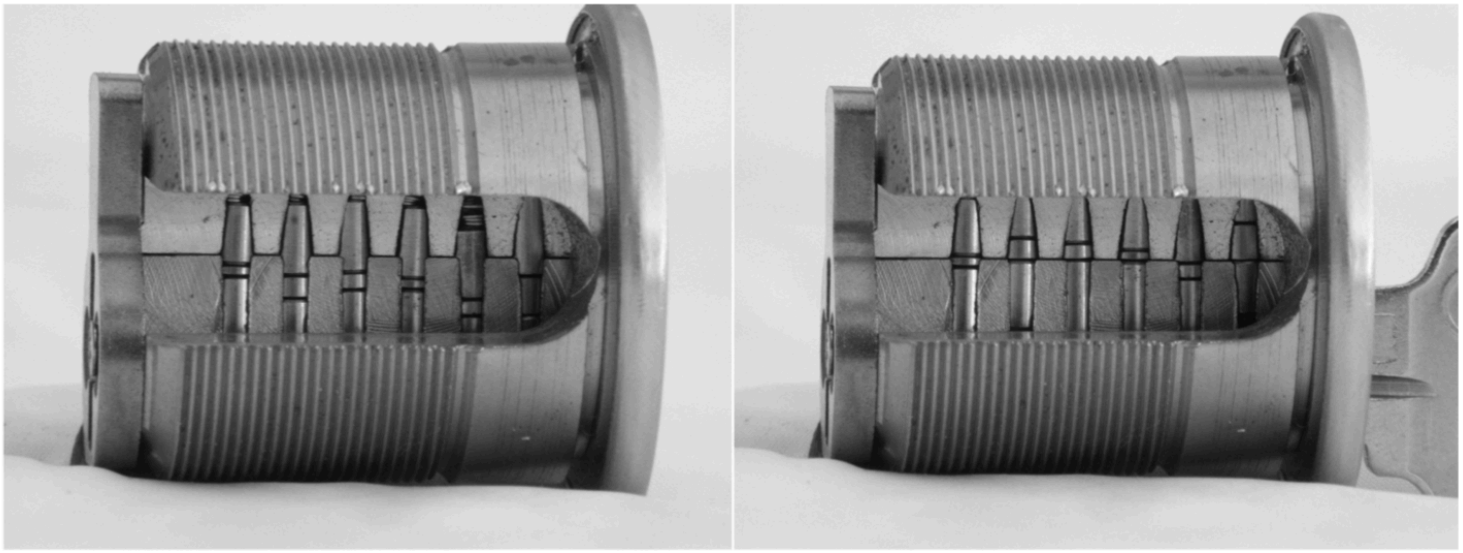


Figure 4: A master keyed pin tumbler lock. *Left:* Each of the six pin stacks has two cuts. *Right:* With the correct *change key* inserted, one of the cuts on each pin stack is aligned at the shear line. Observe that the other cut is sometimes above and sometimes below the shear line.

positions, although the positions will vary (rotate) from lock to lock. Both these schemes can implement a directed graph with several levels of master keys: "sub-master" keys that open a subset of locks in the system and "grand master" keys that open more[1]. The highest-level master key, which opens all locks in a multi-level system, is sometimes called the *Top Master Key (TMK)*.

Master keying has long been understood to reduce security in several important ways. First, of course, the master key represents a very valuable target; compromise of the master key compromises the entire

“Rogues are very keen in their profession, and know already much more than we can teach them.”

–Alfred Charles Hobbs

Security through
obscurity is
no
Security

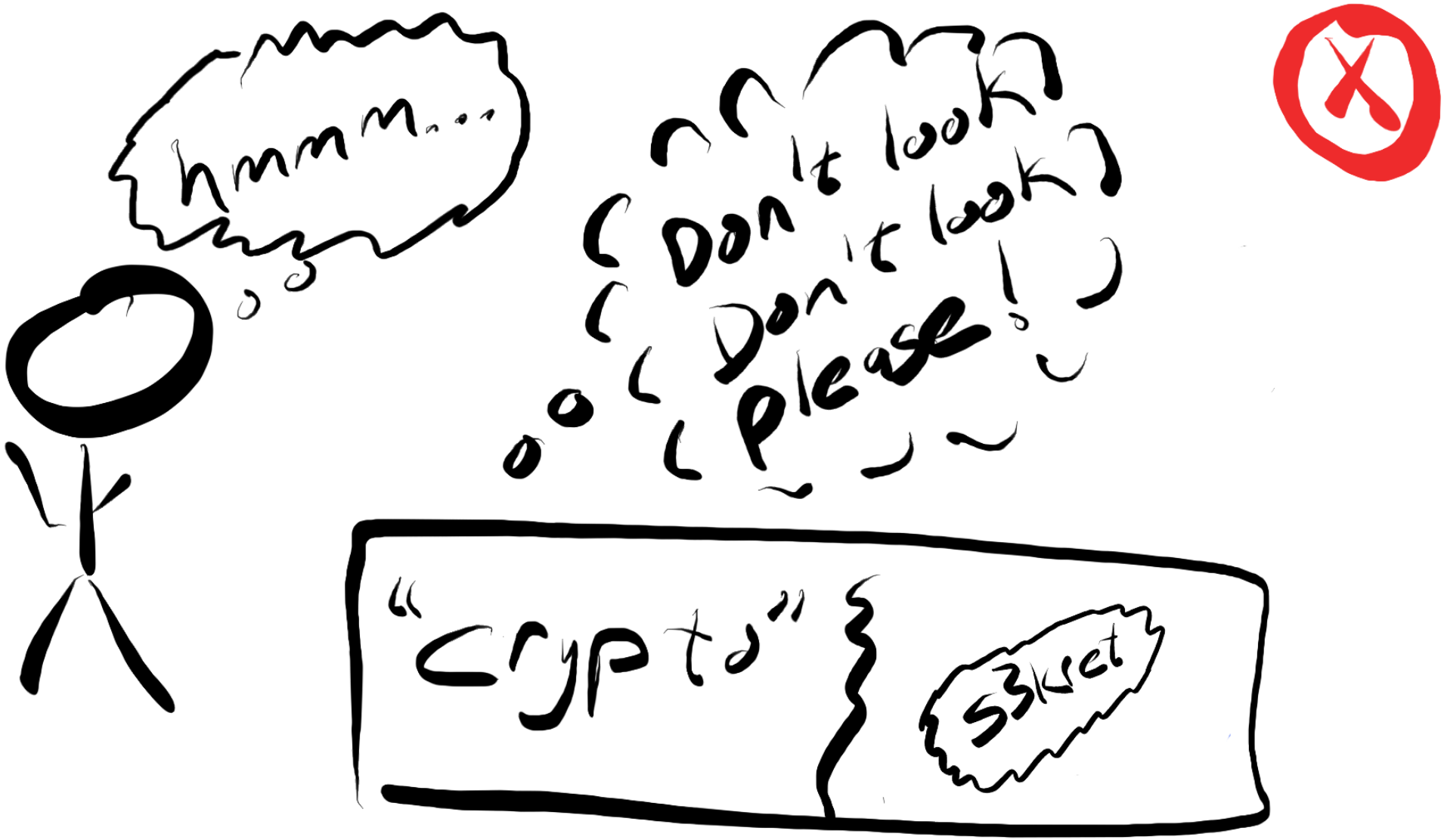
Security through obscurity is ~~no~~ a thin layer of Security

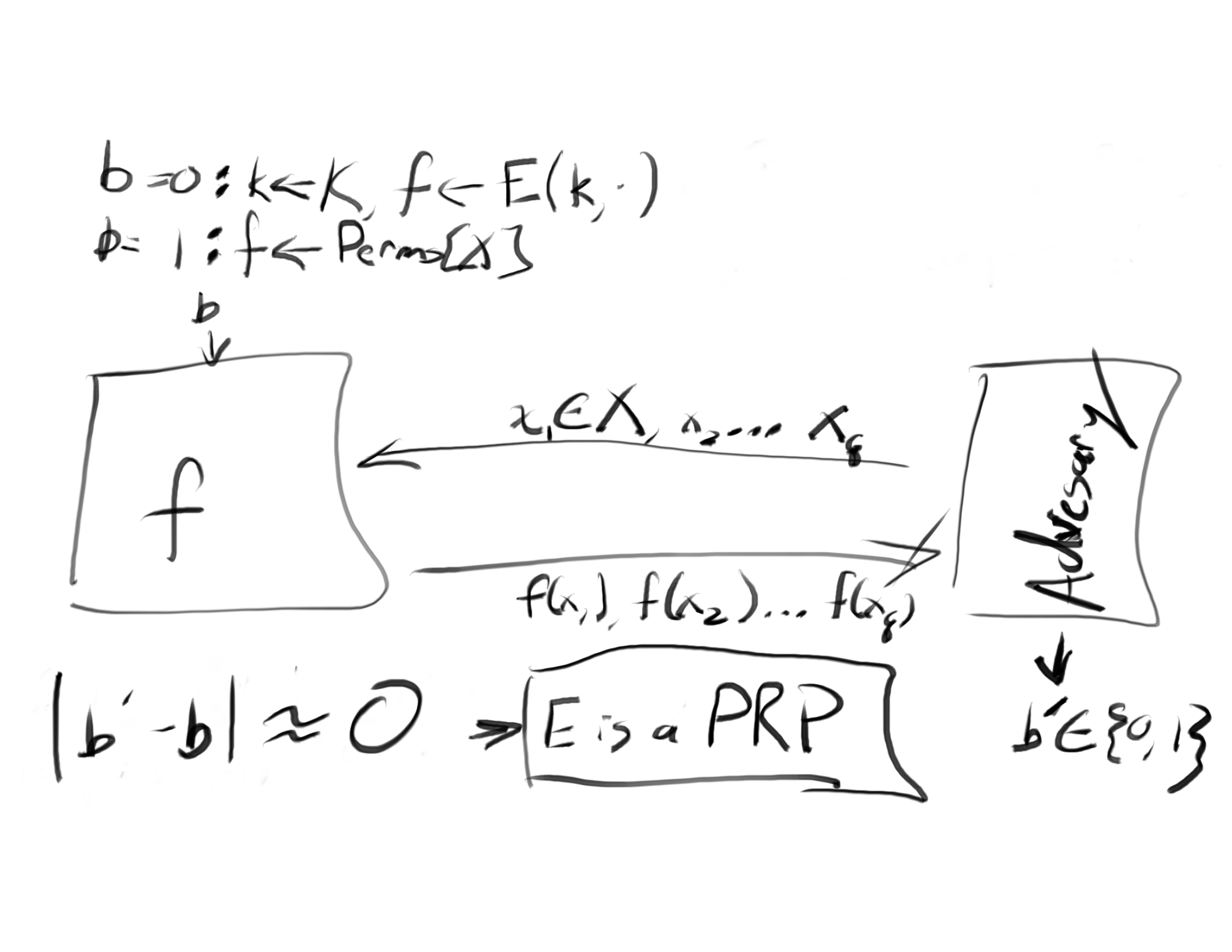
The Badlands of Security

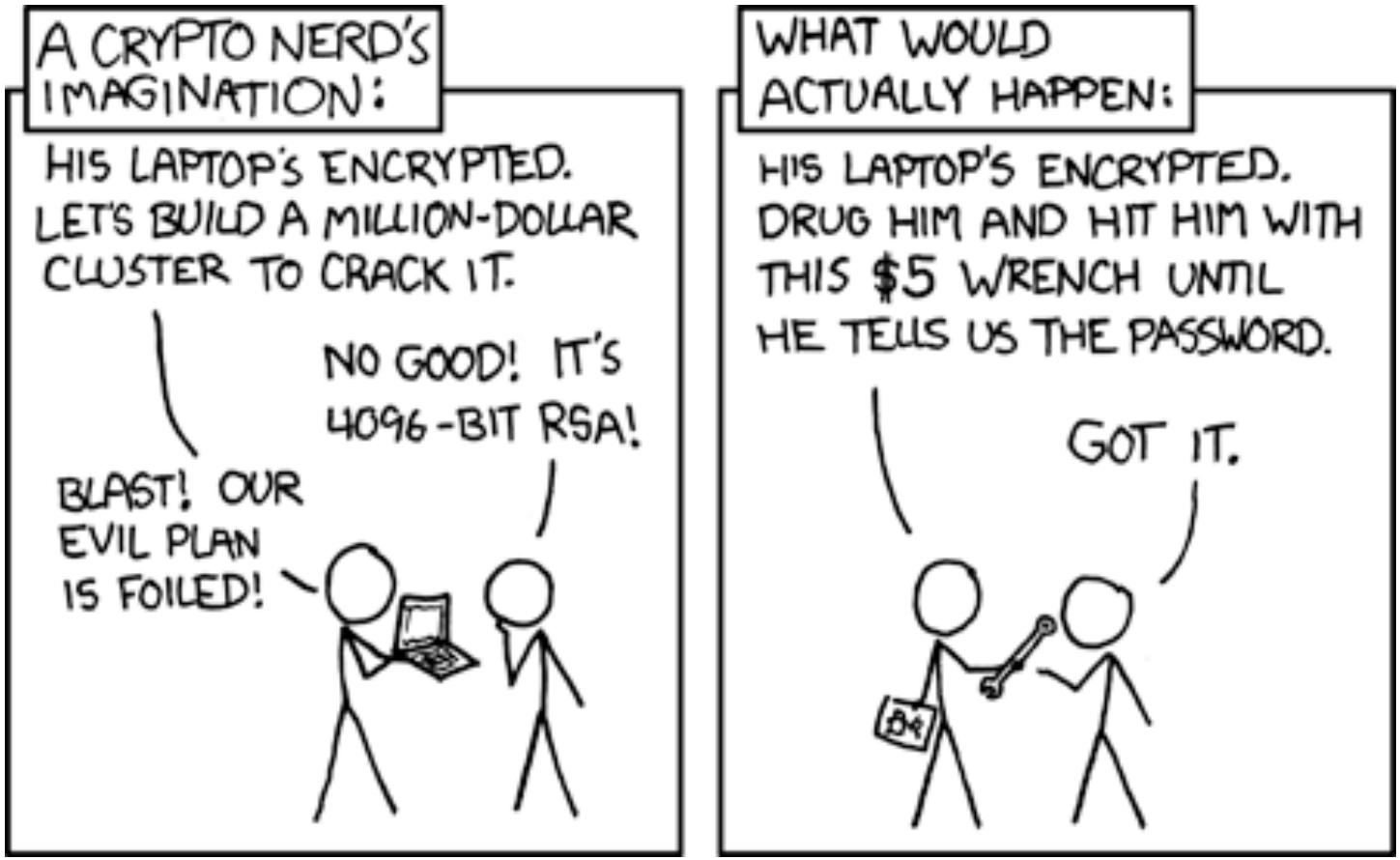
Obscurity ???

Obfu - what?


s3kret
crypto


hmmm...
Don't look
Don't look
Don't look
Please!
"crypto"
secret

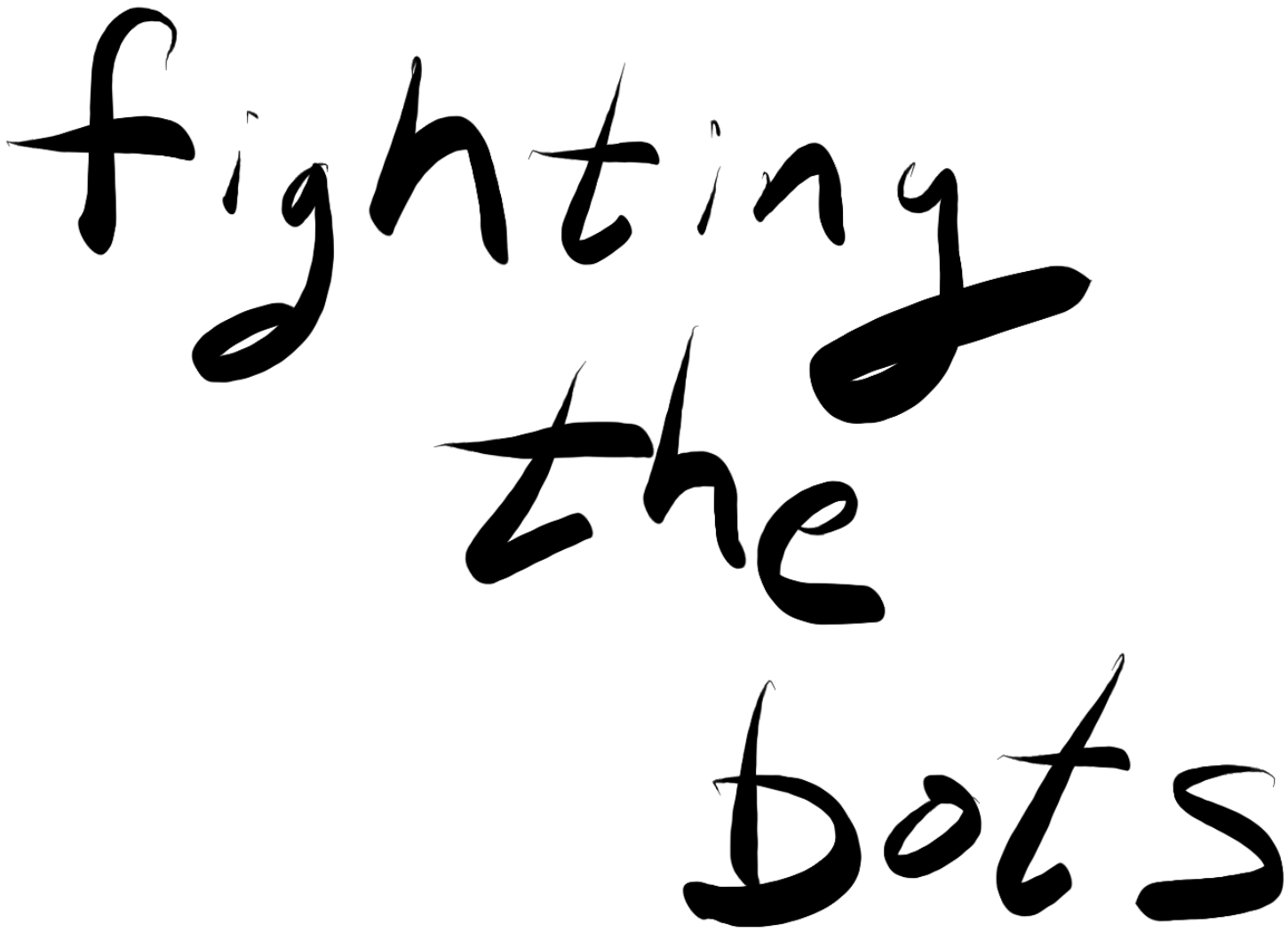

b=0 : k ← K, f ← E(k, ·)
b=1 : f ← Perm[X]
b
f
x1 ∈ X, x2 ... xq
f(x1), f(x2) ... f(xq)
Adversary
b' ∈ {0,1}
|b' - b| ≈ 0
E is a PRP



XKCD #538, used with permission

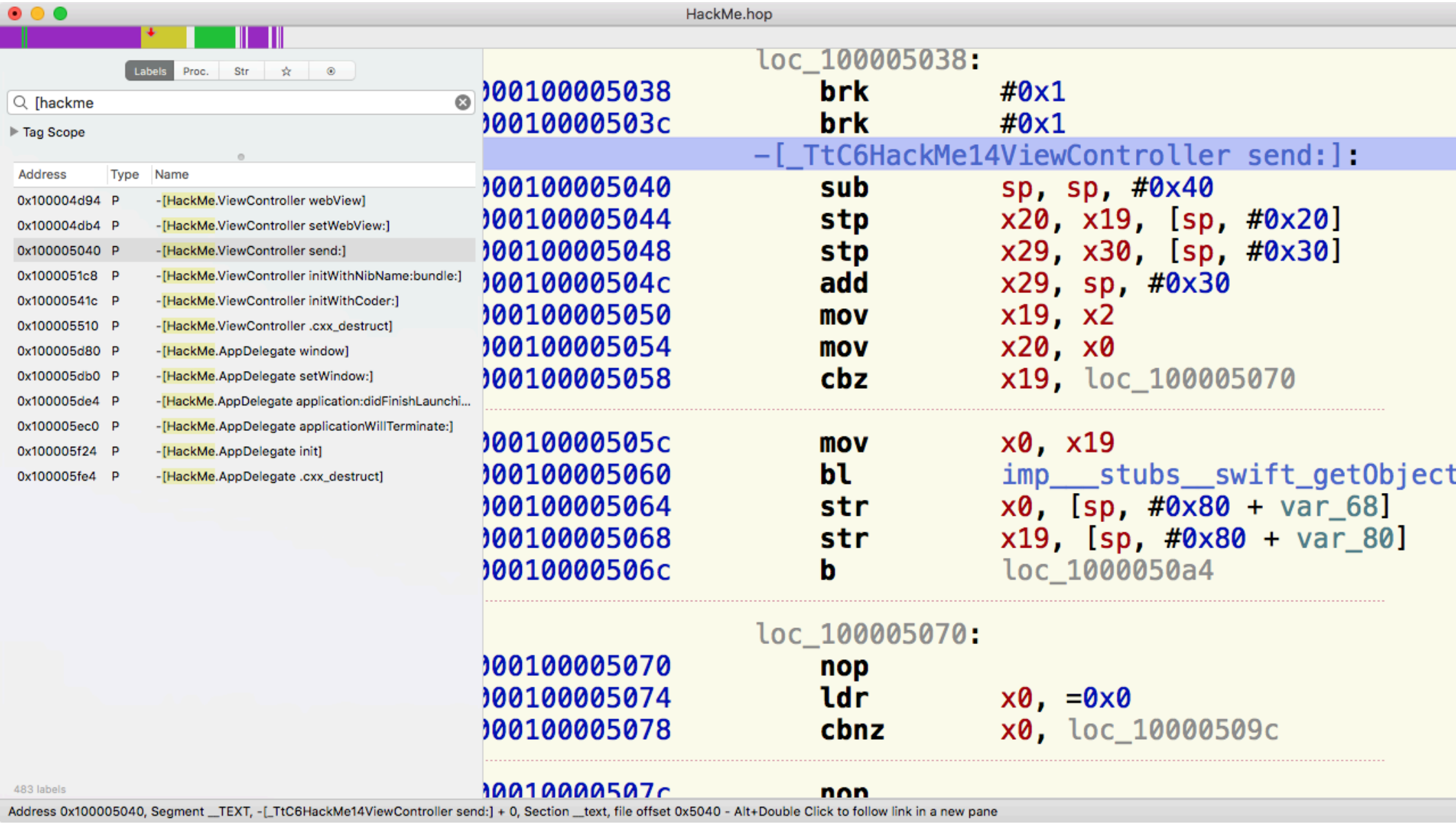

Raising
the
Bar

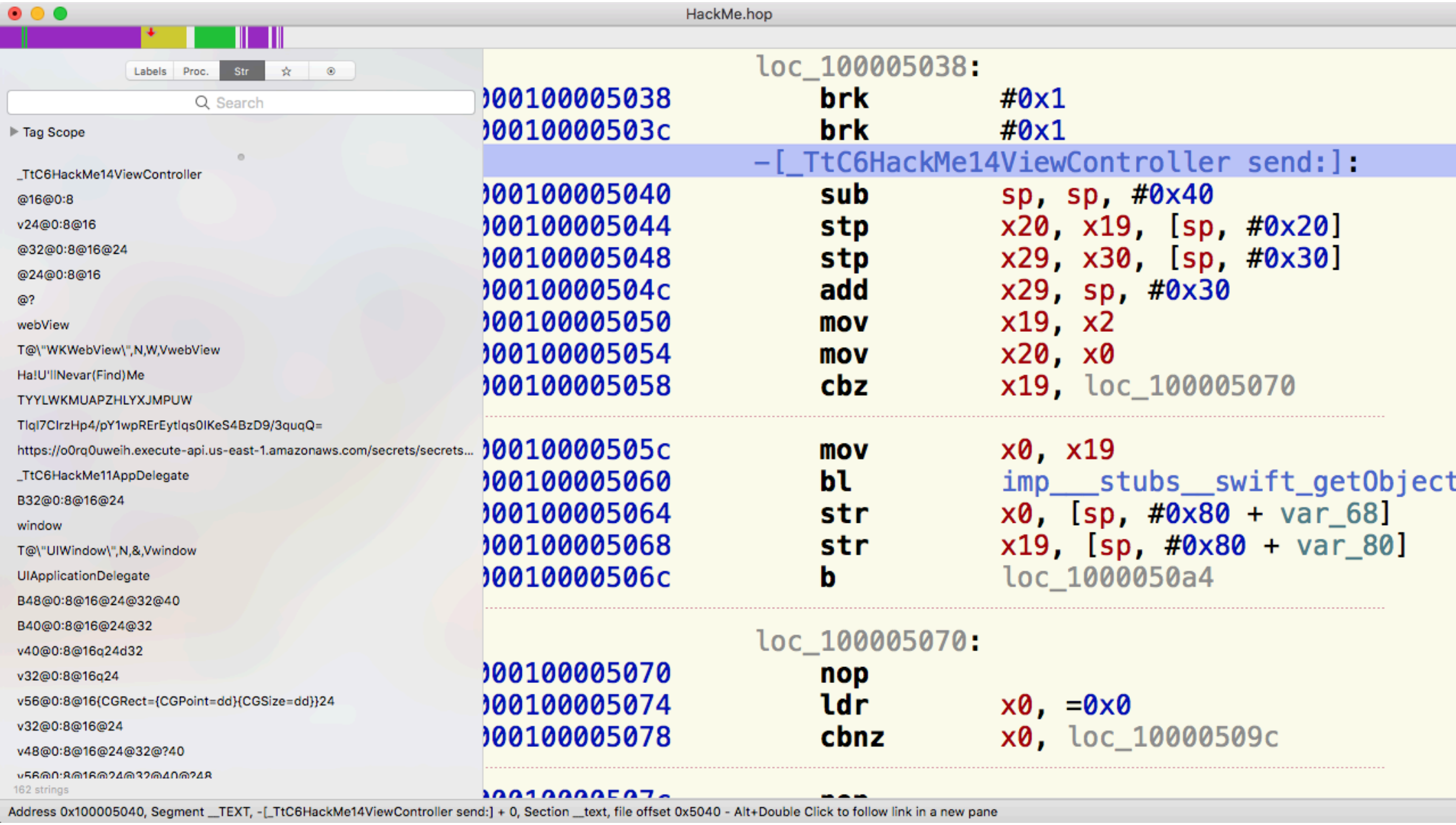
Racing the Bear

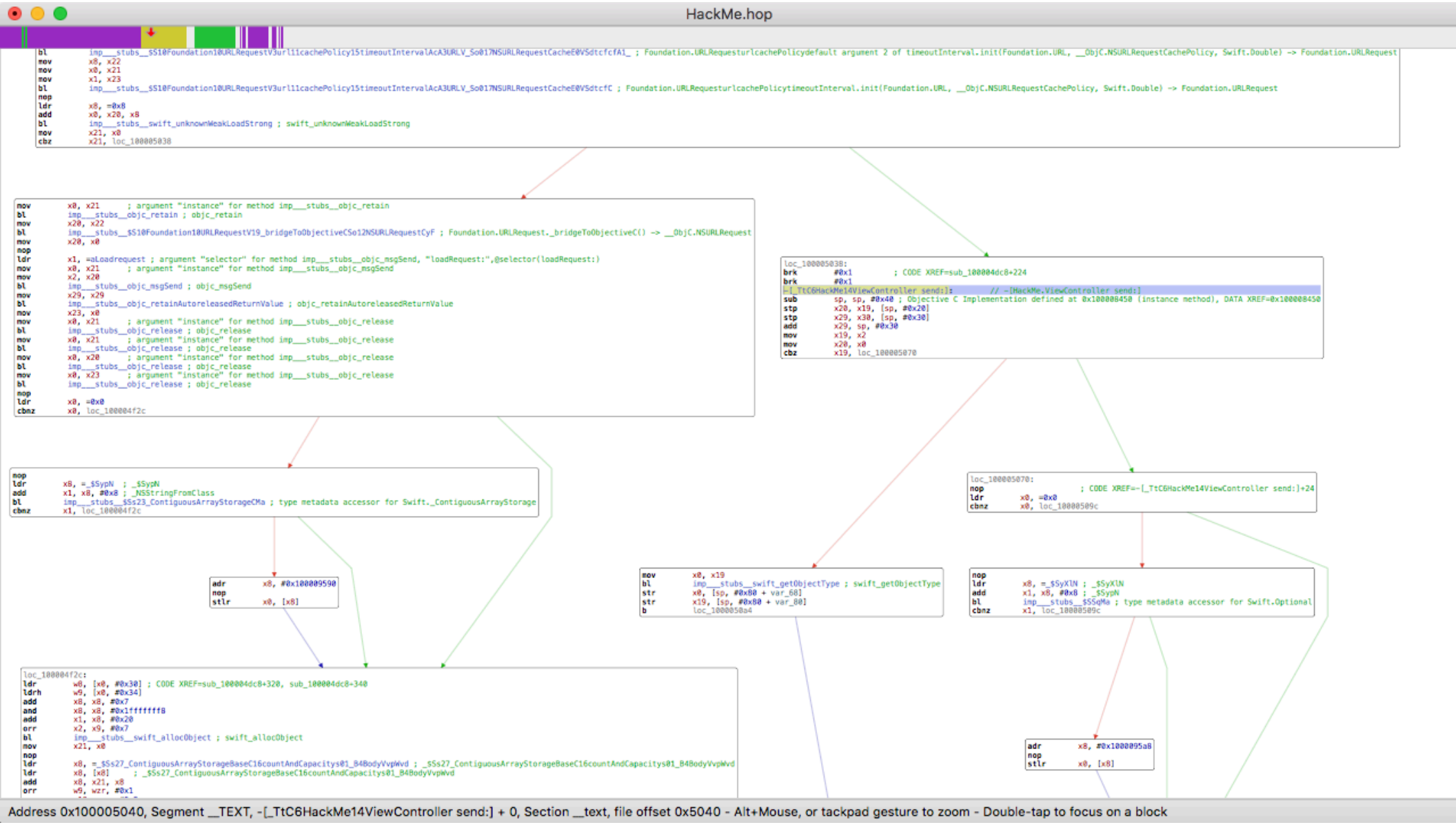

fighting
the
bots

```
$ strings -n 8 HackMe
UIApplicationLaunchOptionsKey
ViewController
AppDelegate
_TtC6HackMe14ViewController
v24@0:8@16
@32@0:8@16@24
@24@0:8@16
T@"WKWebView",N,W,VwebView
https://o0rq0uweih.execute-api.us-east-1.amazonaws.com/secrets/secrets?token=
Ha!U'llNevar(Find)Me
TYYLWKMUAPZHLYXJMPUW
TlqI7CIrzHp4/pY1wpRErEytlqs0IKeS4BzD9/3quqQ=
_TtC6HackMe11AppDelegate
B32@0:8@16@24
 ...
```

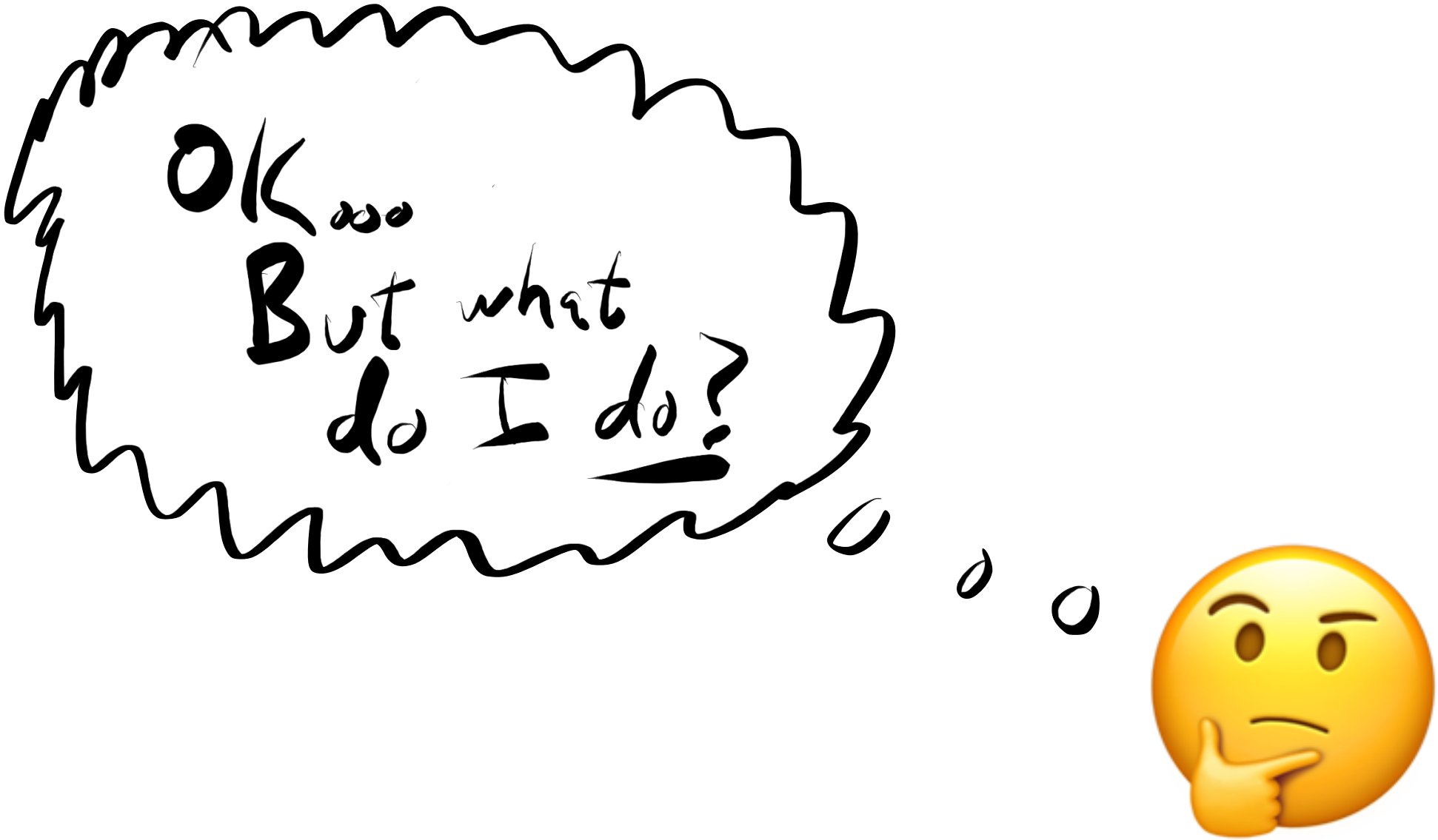
```
$ strings -n 8 HackMe
UIApplicationLaunchOptionsKey
ViewController
AppDelegate
_TtC6HackMe14ViewController
v24@0:8@16
@32@0:8@16@24
@24@0:8@16
T@"WKWebView",N,W,VwebView
https://o0rq0uweih.execute-api.us-east-1.amazonaws.com/secrets/secrets?token=
Ha!U'llNevar(Find)Me
TYYLWKMUAPZHLYXJMPUW
TlqI7CIrzHp4/pY1wpRErEytlqs0IKeS4BzD9/3quqQ=
_TtC6HackMe11AppDelegate
B32@0:8@16@24
 ...
```


HackMe.hop
Labels
Proc.
Str
[hackme
Tag Scope
Address
Type
Name
0x100004d94 P -[HackMe.ViewController webView]
0x100004db4 P -[HackMe.ViewController setWebView:]
0x100005040 P -[HackMe.ViewController send:]
0x1000051c8 P -[HackMe.ViewController initWithNibName:bundle:]
0x10000541c P -[HackMe.ViewController initWithCoder:]
0x100005510 P -[HackMe.ViewController .cxx_destruct]
0x100005d80 P -[HackMe.AppDelegate window]
0x100005db0 P -[HackMe.AppDelegate setWindow:]
0x100005de4 P -[HackMe.AppDelegate application:didFinishLaunchi...
0x100005ec0 P -[HackMe.AppDelegate applicationWillTerminate:]
0x100005f24 P -[HackMe.AppDelegate init]
0x100005fe4 P -[HackMe.AppDelegate .cxx_destruct]
483 labels
loc_100005038:
00100005038 brk #0x1
0010000503c brk #0x1
-[_TtC6HackMe14ViewController send:]:
00100005040 sub sp, sp, #0x40
00100005044 stp x20, x19, [sp, #0x20]
00100005048 stp x29, x30, [sp, #0x30]
0010000504c add x29, sp, #0x30
00100005050 mov x19, x2
00100005054 mov x20, x0
00100005058 cbz x19, loc_100005070
0010000505c mov x0, x19
00100005060 bl imp___stubs__swift_getObject
00100005064 str x0, [sp, #0x80 + var_68]
00100005068 str x19, [sp, #0x80 + var_80]
0010000506c b loc_1000050a4
loc_100005070:
00100005070 nop
00100005074 ldr x0, =0x0
00100005078 cbnz x0, loc_10000509c
Address 0x100005040, Segment __TEXT, -[_TtC6HackMe14ViewController send:] + 0, Section __text, file offset 0x5040 - Alt+Double Click to follow link in a new pane


HackMe.hop
Labels
Proc.
Str
Search
Tag Scope
_TtC6HackMe14ViewController
@16@0:8
v24@0:8@16
@32@0:8@16@24
@24@0:8@16
@?
webView
T@\"WKWebView\",N,W,VwebView
Ha!U'llNevar(Find)Me
TYYLWKMUAPZHLYXJMPUW
Tlql7ClrzHp4/pY1wpRErEytlqs0lKeS4BzD9/3quqQ=
https://o0rq0uweih.execute-api.us-east-1.amazonaws.com/secrets/secrets...
_TtC6HackMe11AppDelegate
B32@0:8@16@24
window
T@\"UIWindow\",N,&,Vwindow
UIApplicationDelegate
B48@0:8@16@24@32@40
B40@0:8@16@24@32
v40@0:8@16q24d32
v32@0:8@16q24
v56@0:8@16{CGRect={CGPoint=dd}{CGSize=dd}}24
v32@0:8@16@24
v48@0:8@16@24@32@?40
162 strings
loc_100005038:
00100005038 brk #0x1
0010000503c brk #0x1
-[_TtC6HackMe14ViewController send:]:
00100005040 sub sp, sp, #0x40
00100005044 stp x20, x19, [sp, #0x20]
00100005048 stp x29, x30, [sp, #0x30]
0010000504c add x29, sp, #0x30
00100005050 mov x19, x2
00100005054 mov x20, x0
00100005058 cbz x19, loc_100005070
0010000505c mov x0, x19
00100005060 bl imp___stubs__swift_getObject
00100005064 str x0, [sp, #0x80 + var_68]
00100005068 str x19, [sp, #0x80 + var_80]
0010000506c b loc_1000050a4
loc_100005070:
00100005070 nop
00100005074 ldr x0, =0x0
00100005078 cbnz x0, loc_10000509c
Address 0x100005040, Segment __TEXT, -[_TtC6HackMe14ViewController send:] + 0, Section __text, file offset 0x5040 - Alt+Double Click to follow link in a new pane


HackMe.hop
bl imp___stubs__$S10Foundation10URLRequestV3url11cachePolicy15timeoutIntervalAcA3URLV_So017NSURLRequestCacheE0VSdtcfcfA1_ ; Foundation.URLRequesturlcachePolicydefault argument 2 of timeoutInterval.init(Foundation.URL, __ObjC.NSURLRequestCachePolicy, Swift.Double) -> Foundation.URLRequest
mov x0, x22
mov x0, x21
mov x1, x23
bl imp___stubs__$S10Foundation10URLRequestV3url11cachePolicy15timeoutIntervalAcA3URLV_So017NSURLRequestCacheE0VSdtcfC ; Foundation.URLRequesturlcachePolicytimeoutInterval.init(Foundation.URL, __ObjC.NSURLRequestCachePolicy, Swift.Double) -> Foundation.URLRequest
nop
ldr x8, =0x8
add x0, x20, x8
bl imp___stubs__swift_unknownWeakLoadStrong ; swift_unknownWeakLoadStrong
mov x21, x0
cbz x21, loc_100005038
mov x0, x21 ; argument "instance" for method imp___stubs__objc_retain
bl imp___stubs__objc_retain ; objc_retain
mov x20, x22
bl imp___stubs__$S10Foundation10URLRequestV19_bridgeToObjectiveCSo12NSURLRequestCyF ; Foundation.URLRequest._bridgeToObjectiveC() -> __ObjC.NSURLRequest
mov x20, x0
nop
ldr x1, =aLoadrequest ; argument "selector" for method imp___stubs__objc_msgSend, "loadRequest:",@selector(loadRequest:)
mov x0, x21 ; argument "instance" for method imp___stubs__objc_msgSend
mov x2, x20
bl imp___stubs__objc_msgSend ; objc_msgSend
mov x29, x29
bl imp___stubs__objc_retainAutoreleasedReturnValue ; objc_retainAutoreleasedReturnValue
mov x23, x0
mov x0, x21 ; argument "instance" for method imp___stubs__objc_release
bl imp___stubs__objc_release ; objc_release
mov x0, x21 ; argument "instance" for method imp___stubs__objc_release
bl imp___stubs__objc_release ; objc_release
mov x0, x20 ; argument "instance" for method imp___stubs__objc_release
bl imp___stubs__objc_release ; objc_release
mov x0, x23 ; argument "instance" for method imp___stubs__objc_release
bl imp___stubs__objc_release ; objc_release
nop
ldr x0, =0x0
cbnz x0, loc_100004f2c
loc_100005038:
brk #0x1 ; CODE XREF=sub_100004dc8+224
brk #0x1
-[_TtC6HackMe14ViewController send:]: // -[HackMe.ViewController send:]
sub sp, sp, #0x40 ; Objective C Implementation defined at 0x100008450 (instance method), DATA XREF=0x100008450
stp x20, x19, [sp, #0x20]
stp x29, x30, [sp, #0x30]
add x29, sp, #0x30
mov x19, x2
mov x20, x0
cbz x19, loc_100005070
nop
ldr x8, =_$SypN ; _$SypN
add x1, x8, #0x8 ; _NSStringFromClass
bl imp___stubs__$Ss23_ContiguousArrayStorageCMa ; type metadata accessor for Swift._ContiguousArrayStorage
cbnz x1, loc_100004f2c
loc_100005070:
nop ; CODE XREF=-[_TtC6HackMe14ViewController send:]+24
ldr x0, =0x0
cbnz x0, loc_10000509c
adr x8, #0x100009590
nop
stlr x0, [x8]
mov x0, x19
bl imp___stubs__swift_getObjectType ; swift_getObjectType
str x0, [sp, #0x80 + var_60]
str x19, [sp, #0x80 + var_80]
b loc_1000050a4
nop
ldr x8, =_$SyXlN ; _$SyXlN
add x1, x8, #0x8 ; _$SypN
bl imp___stubs__$SSqMa ; type metadata accessor for Swift.Optional
cbnz x1, loc_10000509c
loc_100004f2c:
ldr w8, [x0, #0x30] ; CODE XREF=sub_100004dc8+320, sub_100004dc8+340
ldrh w9, [x0, #0x34]
add x8, x8, #0x7
and x8, x8, #0x1ffffff8
add x1, x8, #0x20
orr x2, x9, #0x7
bl imp___stubs__swift_allocObject ; swift_allocObject
mov x21, x0
nop
ldr x8, =_$Ss27_ContiguousArrayStorageBaseC16countAndCapacitys01_B4BodyVvpWvd ; _$Ss27_ContiguousArrayStorageBaseC16countAndCapacitys01_B4BodyVvpWvd
ldr x8, [x8] ; _$Ss27_ContiguousArrayStorageBaseC16countAndCapacitys01_B4BodyVvpWvd
add x8, x21, x8
orr w9, wzr, #0x1
adr x8, #0x1000095a8
nop
stlr x0, [x8]
Address 0x100005040, Segment __TEXT, -[_TtC6HackMe14ViewController send:] + 0, Section __text, file offset 0x5040 - Alt+Mouse, or tackpad gesture to zoom - Double-tap to focus on a block


HackMe.hop
r1 = r23;
r0 = Foundation.URLRequesturlcachePolicytimeoutInterval.init();
r0 = r20 + *objc_ivar_offset__TtC6HackMe14ViewController_webView;
r0 = swift_unknownWeakLoadStrong();
r21 = r0;
if (r21 != 0x0) {
r0 = [r21 retain];
r20 = r22;
r20 = Foundation.URLRequest._bridgeToObjectiveC();
r0 = [r21 loadRequest:r20];
r29 = r29;
r23 = [r0 retain];
r0 = [r21 release];
r0 = [r21 release];
r0 = [r20 release];
r0 = [r23 release];
r0 = *qword_100009590;
if (r0 == 0x0) {
r1 = *type metadata for Any + 0x8;
r0 = type metadata accessor for Swift._ContiguousArrayStorage();
if (r1 == 0x0) {
r8 = 0x100009590;
asm { stlr x0, [x8] };
}
Address 0x100005040, Segment __TEXT, -[_TtC6HackMe14ViewController send:] + 0, Section __text, file offset 0x5040


OK...
But what
do I do?

Three can keep a
secret,
If two of them
are dead

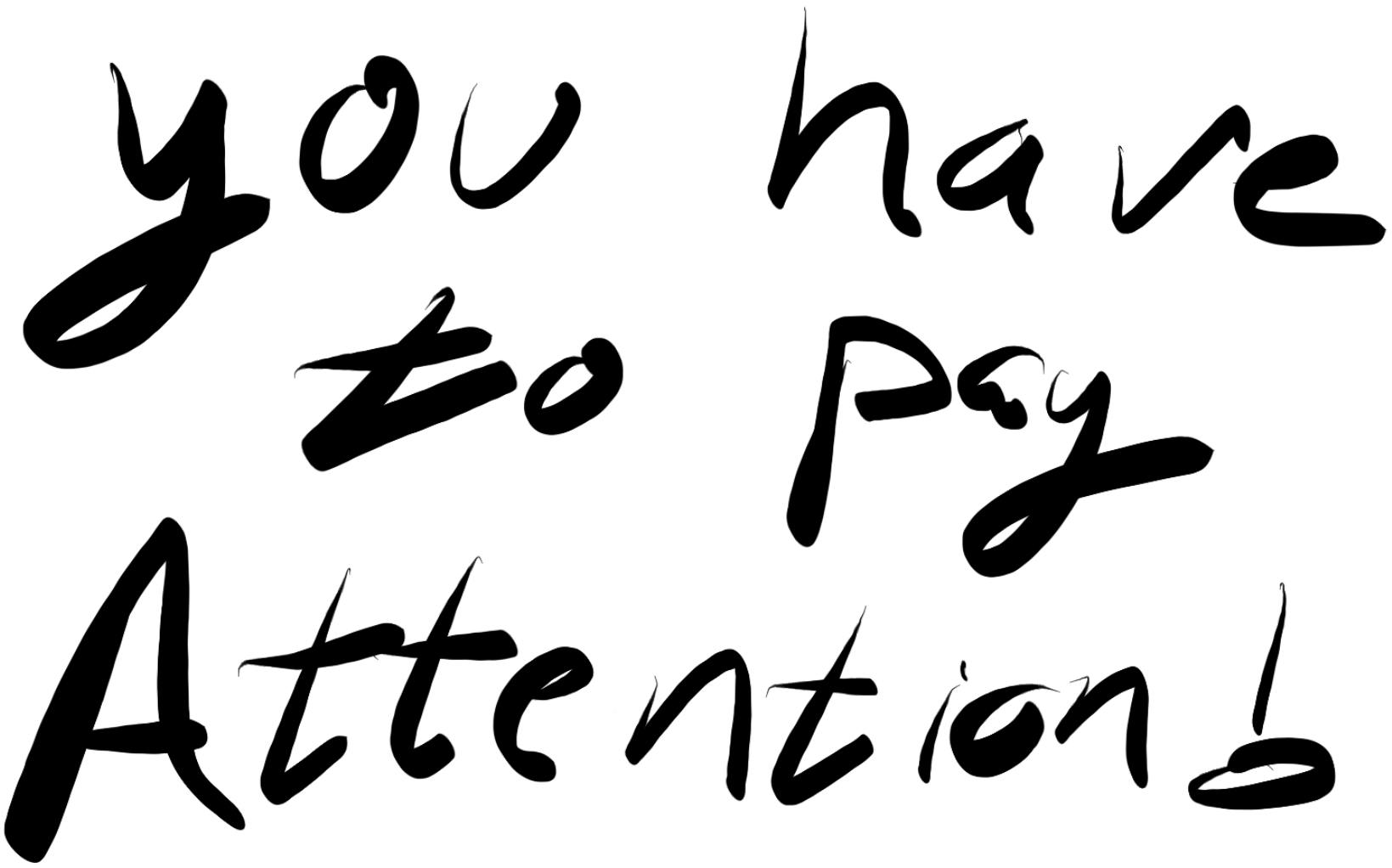
-B. Franklin

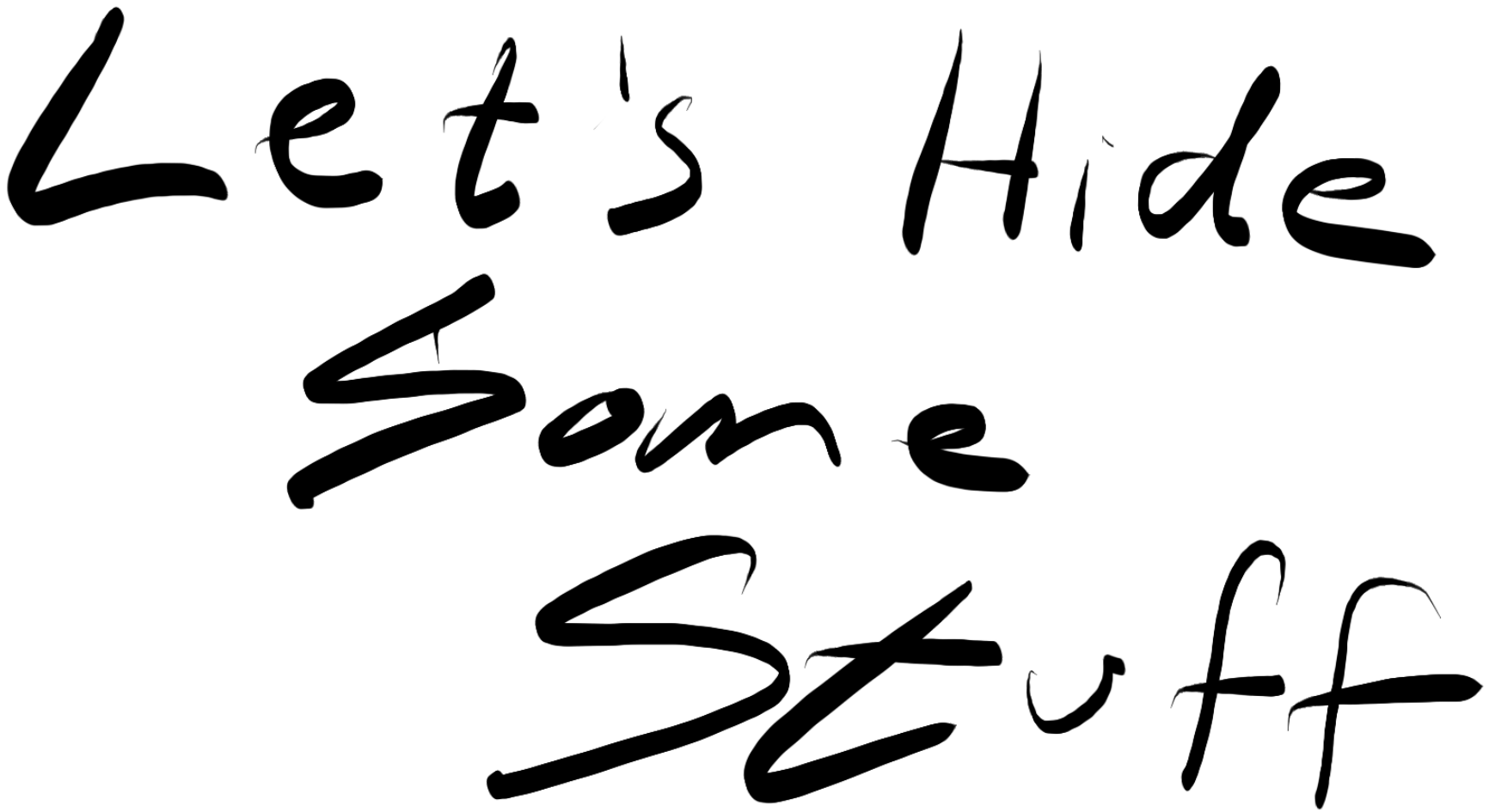
Fewer

Secrets

Authenticate Users.
Not Apps


HONEY


you have
to pay
Attention!


Let's Hide
Some
Stuff

```
let key = "0rlsRU5vjPvHUad70ysqX67k4DN50/sNYRD5zqnCLbg="
```

```
echo 0rlsRU5vjPvHUad70ysqX67k4DN50/sNYRD5zqnCLbg= | base64 -D | xxd -i
```

```
let key = Data([
    0×45, 0×6c, 0×b9, 0×d2, 0×fb, 0×8c, 0×6f, 0×4e, 0×7b, 0×a7, 0×51, 0×c7,
    0×5f, 0×2a, 0×2b, 0×d3, 0×33, 0×e0, 0×e4, 0×ae, 0×0d, 0×fb, 0×d3, 0×79,
    0×ce, 0×f9, 0×10, 0×61, 0×b8, 0×2d, 0×c2, 0×a9
   ])
```

```
$ otool -d HackMe
HackMe:
Contents of (__DATA,__data) section
0000000100009538  00000000 00000000 00000000 00000000
0000000100009548  00000000 00000000 00000000 00000000
0000000100009558  00000000 00000000 00000000 00000000
0000000100009568  00000020 00000000 00000040 00000000
0000000100009578  456cb9d2 fb8c6f4e 7ba751c7 5f2a2bd3
0000000100009588  33e0e4ae 0dfbd379 cef91061 b82dc2a9
0000000100009598  00000000 00000000 00000000 00000000
00000001000095a8  00000000 00000000 00000001 00000000
00000001000095b8  00006540 00000001 00000000 00000000
00000001000095c8  00000000 00000000 00000000 00000000
00000001000095d8  00000000 00000000 00000000 00000000
00000001000095e8  00000000 00000000 00000000 00000000
00000001000095f8  00000000 00000000 00008408 00000001
0000000100009608  00000008 00000000 00000000 00000000
0000000100009618  00000000 00000000 00000000 00000000
0000000100009628  00000000 00000000 00000000 00000000
0000000100009638  00000000 00000000 00000000 00000000
0000000100009648  00000000 00000000 00000000 00000000
```

```
let englishKey = "Ha!U'llNevar(Find)Me!I’mSoSekret"
```

```
head -c32 /dev/random | xxd -i
```

```
let aesKey = Data([
      0×45, 0×6c, 0×b9, 0×d2, 0×fb, 0×8c, 0×6f, 0×4e, 0×7b, 0×a7, 0×51, 0×c7,
      0×5f, 0×2a, 0×2b, 0×d3, 0×33, 0×e0, 0×e4, 0×ae, 0×0d, 0×fb, 0×d3, 0×79,
      0×ce, 0×f9, 0×10, 0×61, 0×b8, 0×2d, 0×c2, 0×a9
    ])
```

Insert

Dr. Evil GIF

Here

you know the one...

Letters + Numbers + Symbols

≈ 96

$96^{32} / 256^{32} \approx 100$ trillion

TYYLWKMUAPZHLYXJMPUW

swiftmask

head -c 20 /dev/random | xxd -i

```
let mask = Data([
    0×77, 0×a6, 0×7f, 0×09, 0×1f, 0×74, 0×2d, 0×f3, 0×b6, 0×88, 0×b6, 0×ff,
    0×05, 0×c3, 0×9f, 0×3c, 0×02, 0×3a, 0×b1, 0×ac
    ])
```

print(zip(api, mask).map(^))

```
let masked = Data(
    [35, 255, 38, 69, 72, 63, 96, 166, 247, 216,
     236, 183, 73, 154, 199, 118, 79, 106, 228, 251]
)
```

let demasked = String(bytes: zip(masked, mask).map(^),
encoding: .utf8)!

```
$ otool -d HackMe
HackMe:
Contents of (__DATA,__data) section
0000000100009538  00000000 00000000 00000000 00000000
0000000100009548  00000000 00000000 00000000 00000000
0000000100009558  00000000 00000000 00000000 00000000
0000000100009568  00000020 00000000 00000040 00000000
0000000100009578  456cb9d2 fb8c6f4e 7ba751c7 5f2a2bd3
0000000100009588  33e0e4ae 0dfbd379 cef91061 b82dc2a9
0000000100009598  00000000 00000000 00000000 00000000
00000001000095a8  00000000 00000000 00000001 00000000
00000001000095b8  00006540 00000001 00000000 00000000
00000001000095c8  00000000 00000000 00000000 00000000
00000001000095d8  00000000 00000000 00000000 00000000
00000001000095e8  00000000 00000000 00000000 00000000
00000001000095f8  00000000 00000000 00008408 00000001
0000000100009608  00000008 00000000 00000000 00000000
0000000100009618  00000000 00000000 00000000 00000000
0000000100009628  00000000 00000000 00000000 00000000
0000000100009638  00000000 00000000 00000000 00000000
0000000100009648  00000000 00000000 00000000 00000000
```

```
head -c20 /dev/random | xxd -i
```

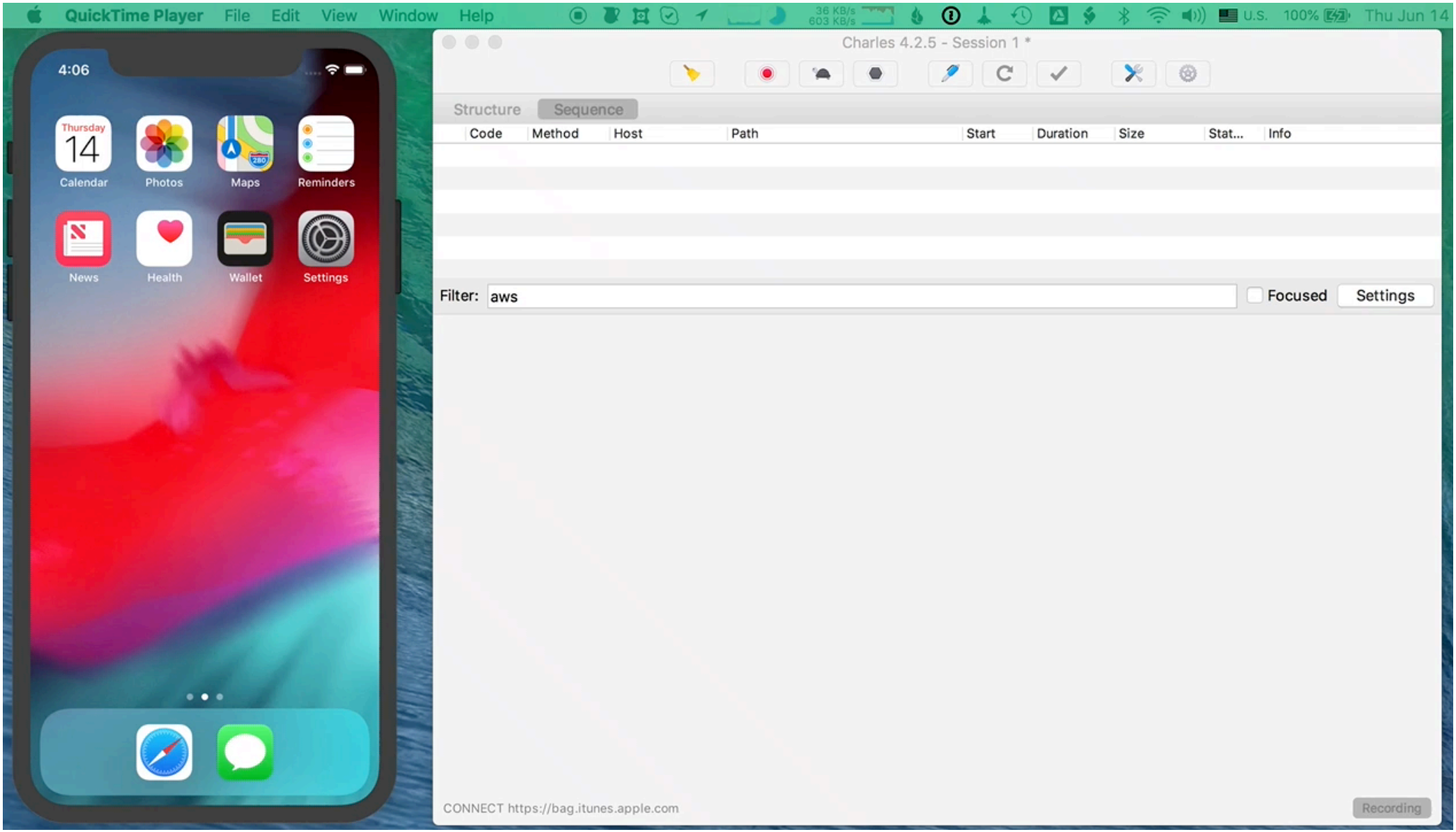
```
let key = Data([
  0x84, 0x26, 0x7e, 0xa2, 0xbc, 0x00, 0x72, 0xff, 0x58, 0x59, 0x40, 0x88,
  0x8c, 0x12, 0x53, 0xfa, 0x0f, 0x75, 0x20, 0xdc,
  0x45, 0x6c, 0xb9, 0xd2, 0xfb, 0x8c, 0x6f, 0x4e, 0x7b, 0xa7, 0x51, 0xc7,
  0x5f, 0x2a, 0x2b, 0xd3, 0x33, 0xe0, 0xe4, 0xae, 0x0d, 0xfb, 0xd3, 0x79,
  0xce, 0xf9, 0x10, 0x61, 0xb8, 0x2d, 0xc2, 0xa9
])
```

```
let key = heap[20 ..< 52]
```


```
EN SPACE = 0
EM SPACE = 1
```

https://github.com/rnapier/SwiftStego


Don't
Overthink


QuickTime Player File Edit View Window Help
36 KB/s 603 KB/s
U.S. 100% Thu Jun 14
4:06
Thursday 14
Calendar
Photos
Maps
Reminders
News
Health
Wallet
Settings
Charles 4.2.5 - Session 1 *
Structure
Sequence
Code
Method
Host
Path
Start
Duration
Size
Stat...
Info
Filter: aws
Focused
Settings
CONNECT https://bag.itunes.apple.com
Recording

Certificate Pinning

https://talk.objc.io/episodes/S01E57-certificate-pinning

A Lot of Trust

You Expect...

- Verisign
- Network Solutions
- Thawte
- DigiCert
- Digital Signature Trust

But Also...

- Amazon, Cisco, Apple, ...
- US, Belgium, Germany, Netherlands, Switzerland, Turkey, …
- Actalis, Atos, Buypass, Certigna, Certinomis, Chambers of Commerce, E-Tugra, Echoworx, Izenpe, QuoVadis, Starfield, …

http://support.apple.com/kb/ht5012

```
extension CertificateValidator: URLSessionDelegate {
    func urlSession(_ session: URLSession, didReceive challenge: URLAuthenticationChallenge,
                    completionHandler: @escaping (URLSession.AuthChallengeDisposition,
                                                  URLCredential?) -> Void) {
        let protectionSpace = challenge.protectionSpace

        if (protectionSpace.authenticationMethod == NSURLAuthenticationMethodServerTrust) {

            if let trust = protectionSpace.serverTrust {
                SecTrustSetAnchorCertificates(trust, trustedCertificates as CFArray)
                SecTrustSetAnchorCertificatesOnly(trust, true)

                var result = SecTrustResultType.invalid
                let status = SecTrustEvaluate(trust, &result)
                if status == errSecSuccess {
                    switch result {
                    case .proceed, .unspecified:
                        completionHandler(.useCredential, URLCredential(trust: trust))
                        return

                    default:
                        print("Could not verify certificate: \(result)")
                    }
                }
            }
        }

        // Something failed. Cancel
        completionHandler(.cancelAuthenticationChallenge, nil)
    }
}
```

```
guard let url = Bundle.main.url(forResource: name, withExtension: "cer") else {
    preconditionFailure("\(name) not found")
}

try! validator = CertificateValidator(certificateURL: url)
session = URLSession(configuration: .default, delegate: validator, delegateQueue: nil)
```

https://github.com/rnapier/CertificateValidator


Secret
Knock

1. generate secret
2. ??? (also send 404)
3. Secure!


True
Story

Spend an hour,

(ok, maybe a few)

Save the day

```
xcrun simctl launch booted net.robnapier.HackMe --wait-for-debugger
```


$ Spending ?
? some $
$ money ?

```
func a(b: URL, c: @escaping (Data?) → Void) {
    URLSession.shared.dataTask(with: b) { (d, e, f) in
        if let g = f {
            print(g)
            c(nil)
        } else if let data = d {
            c(data)
        } else {
            c(nil)
        }
    }.resume()
}
```

```
func o1(_ o7: o2, o3: @escaping (o4) → o5) {
    o20(o6(o7) {
        if let o8 = $2 {
            o9(o8)
            o3(nil)
        } else if let o10 = $0 {
            o3(o10)
        } else {
            o3(nil)
        }
    })()
}
```

```
func o1(_ o7: URL, o3: @escaping (Data?) → Void) {
    o20(o6(o7: URL) { (Data?, URLResponse, Error?) in
        if let o8: Error = $2: Error? {
            o9(o8: Error)
            o3(nil: Data)
        } else if let o10: Data = $0: Data? {
            o3(o10: Data)
        } else {
            o3(nil: Data)
        }
    }: URLSessionDataTask)()
}
```

```
func o1(_ o7: URL, o3: @escaping (Data?) → Void) {
    o20(urlSession.dataTask(with: o7) {
        if let o8: Error = $2: Error? {
            o9(o8: Error)
            o3(nil: Data)
        } else if let o10: Data = $0: Data? {
            o3(o10: Data)
        } else {
            o3(nil: Data)
        }
    }: URLSessionDataTask)()
}
```

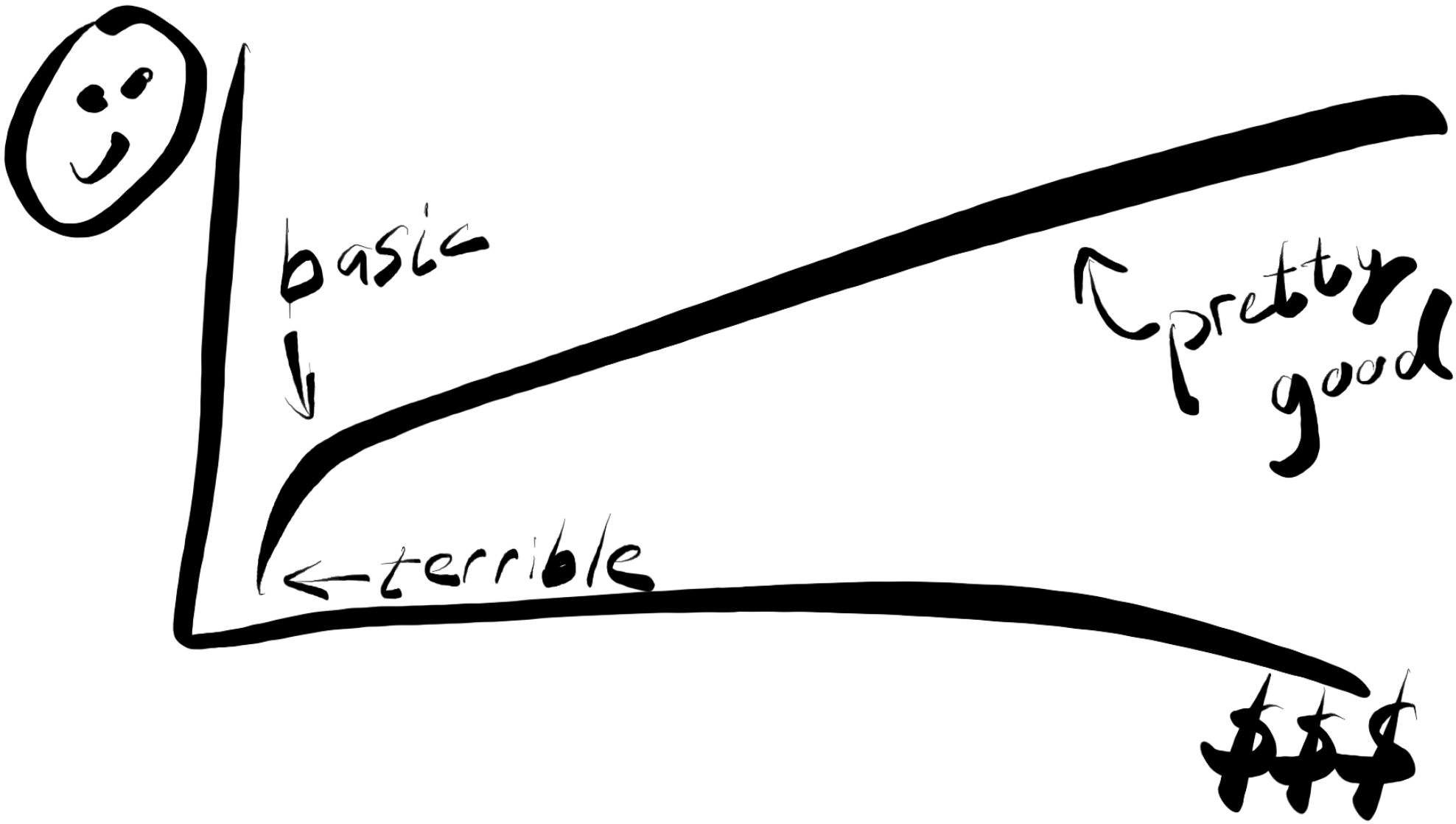
```
func o1(_ o7: URL, o3: @escaping (Data?) → Void) {
    urlSession.dataTask(with: o7) {
        if let o8: Error = $2: Error? {
            o9(o8: Error)
            o3(nil: Data)
        } else if let o10: Data = $0: Data? {
            o3(o10: Data)
        } else {
            o3(nil: Data)
        }
    }.resume()
}
```

```
func o1(_ url: URL,
    completion: @escaping (Data?) → Void) {
    urlSession.dataTask(with: url) {
        if let error = $2 {
            o9(error)
            completion(nil)
        } else if let data = $0 {
            completion(data)
        } else {
            completion(nil)
        }
    }.resume()
}
```

Hardening

No Pain, No Gain

But so much pain


basic
pretty good
terrible
$$$

1. Obfuscation is for you, not users!

2. Focus on the basics strings, network

3. Hopper & Charles

Strings bad

Data less bad

ObjC very bad

(but I still ♥ ObjC)

https://github.com/rnapier/secrets