

YubiKey

YubiHSM

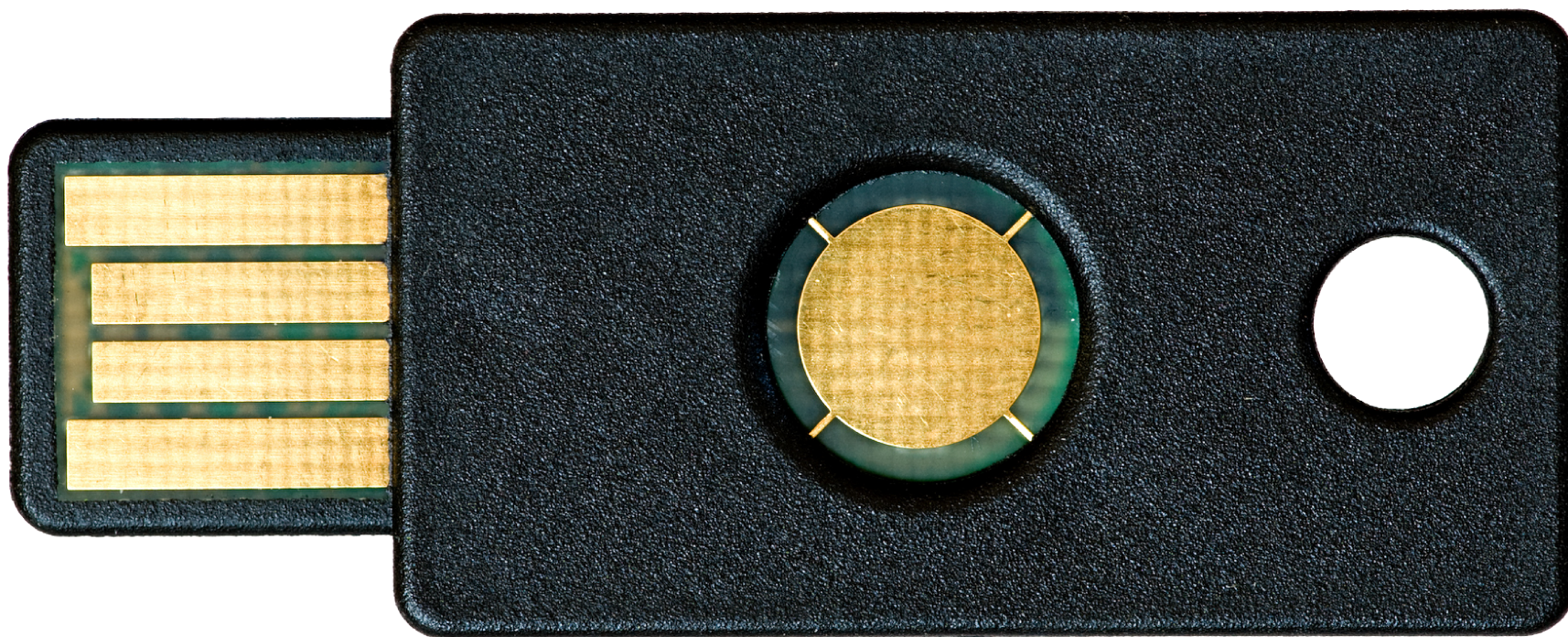
Passwords¹¹

Simon Josefsson

About Yubico

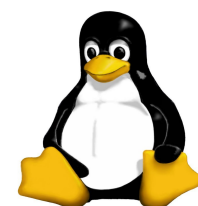
- Started in 2007 in Stockholm
- Founder and CEO is Stina Ehrensvärd
- Presence today in Sweden, UK and US
- Team of ~15 people
- Core invention is the YubiKey
- Online web shop and (in)direct sales
- Web shop sales to anyone - \$25 per unit
- Free software friendly

YubiKey



YubiKey Quick Facts

- The YubiKey generates one-time passwords for identification and authentication
- Two factor, One Touch, Zero drivers!
- No batteries, no display, no mechanical buttons
- Unique AES key in every YubiKey
- YubiKey configuration is customizable



Typical Usage



Password

YubiKey



IDENTITY

ONE TIME PASSWORD

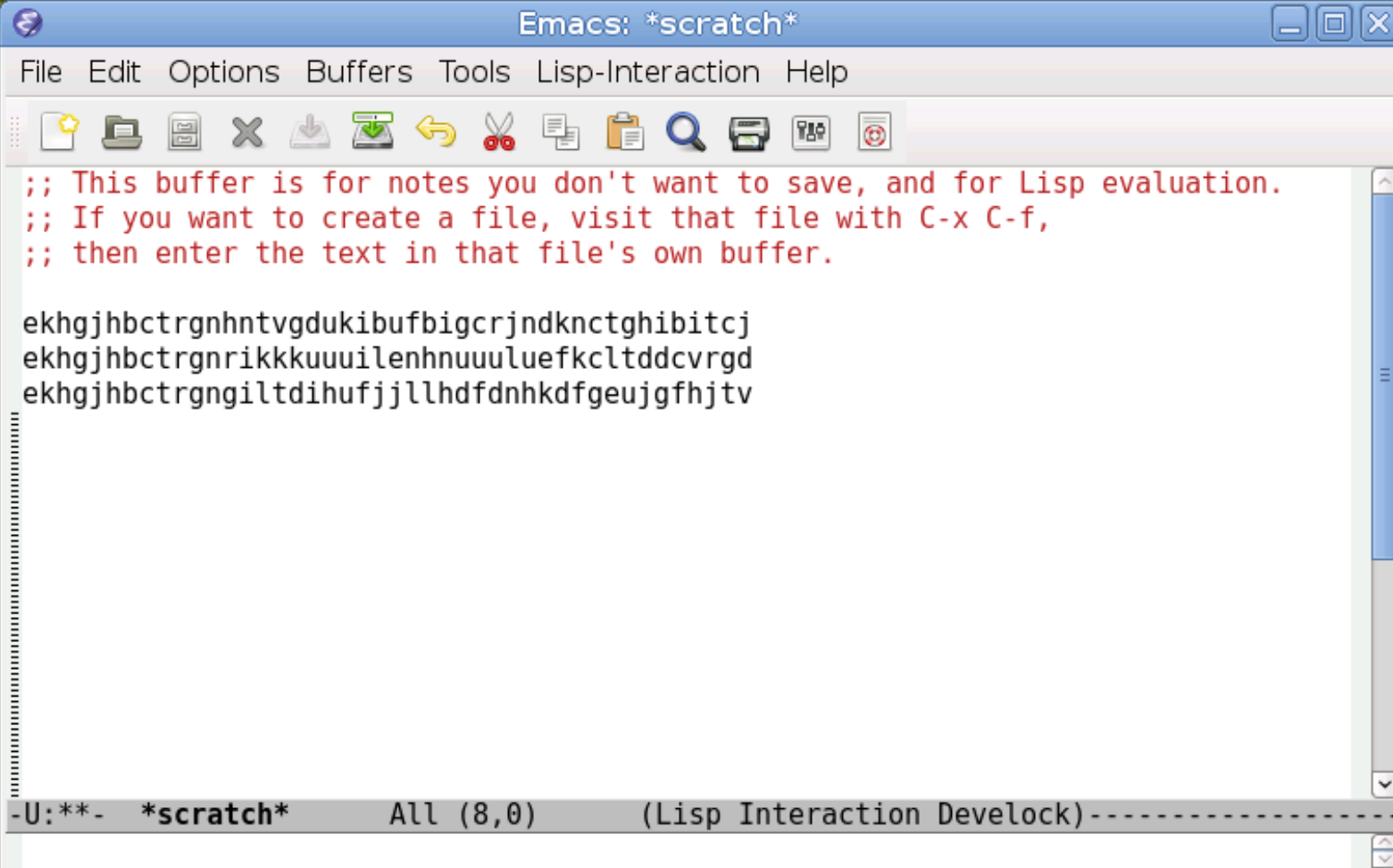


cccccccchllvjtitleikcffjndtjkgnrejudfrjncun
cccccccchllcrnhttrgbgikrcctihnlhclrvhkldcdj



DEMO

- 1. Insert YubiKey**
- 2. Launch text editor**
- 3. Touch YubiKey**



The image shows a screenshot of the Emacs editor window titled "Emacs: *scratch*". The window has a standard macOS-style title bar with minimize, maximize, and close buttons. Below the title bar is a menu bar with the following items: File, Edit, Options, Buffers, Tools, Lisp-Interaction, and Help. Underneath the menu bar is a toolbar containing various icons for file operations (like open, save, delete, download, upload, undo, redo, copy, paste, search, print, and refresh). The main editing area contains the following text:

```
;; This buffer is for notes you don't want to save, and for Lisp evaluation.  
;; If you want to create a file, visit that file with C-x C-f,  
;; then enter the text in that file's own buffer.  
  
ekhgjhbctrghntvgdukibufbigcrjndknctghibitcj  
ekhgjhbctrgnrikkkuuilenhnuuuluefkcltddcvrgd  
ekhgjhbctrngiltdihufjjllhdfdnhkdfgeujgfhjtv
```

At the bottom of the window is a status bar with the following text: `-U:**- *scratch* All (8,0) (Lisp Interaction Develock)-----`

ModHex

- USB keyboards returns scan codes, not characters! Keyboard layout matters...
- Modhex encoding is hex encoding with another alphabet
 - cbdefghijklnrtuv (modhex)
 - 0123456789abcdef (hex)
- For example hex string 00 is cc in modhex
 - Modhex ekhgjhbctrgn is 39658610dc5b hex
- Goal with alphabet is keyboard layout independent character input

YubiKey OTP Format

- One YubiKey OTP consists of two parts:
 - Variable length 0-16 modhex characters for identity
 - 32 modhex characters with OTP data
- The two parts are concatenated:
 - `ekhgjhbctrgn`kutgvrvkinllgnkejtlgidhbubeuebdb
- Yubico ships 12 character identities
 - Splitting PASSWORDOTP concern
- Identity string is configurable

Encrypted OTP data

- Internal format of the encrypted OTP:
 - 6 byte: internal identity string
 - 2 byte: session counter (non-volatile)
 - 2 byte: 8Hz timestamp (low part)
 - 1 byte: 8Hz timestamp (high part)
 - 1 byte: session use (volatile)
 - 2 byte: non-predictable data “nonce”
 - 2 byte: CRC-16 of all fields with this field 0
- Final OTP is AES-ECB encrypted plaintext

Counters and time

- The YubiKey OTP has two monotonously incrementing counters:
 - One that is stored in long-term memory: incremented by one on first use after each powerup
 - One in volatile memory: incremented by one on every use during a powerup-cycle
- The YubiKey OTP contains time information:
 - However it is not wall-clock time but instead time since last power-up (because there is no battery)
 - Requires two OTPs from the same powerup-cycle to detect time-delaying phishing

Static password

- Static password mode
 - Generate the same strong password on every YubiKey touch
- Vulnerable to keyloggers!
- Can provide some security advantages compared to human-recalled passwords
- Useful when evaluating user-acceptance of YubiKey - no server-side changes

OATH HOTP

- Open AuTHentication
 - <http://www.openauthentication.org/>
- HMAC-based One-Time Password (HOTP)
 - RFC 4226. Code is 6-8 digits, e.g. “673821”
- Enables one-time-password systems with tokens from multiple vendors
- The YubiKey can be programmed to generate OATH HOTP codes
 - Version 2.x only – since December 2009

Challenge Response

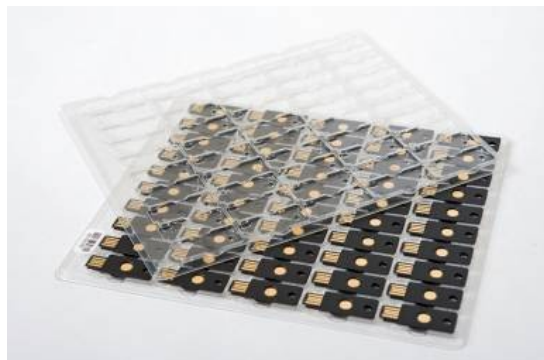
- Combined with client-software the YubiKey supports challenge-response
- Algorithm is HMAC-SHA1
- The YubiKey can sign data authorized by user by touch
- Use-case is software license management, improved security, pay-TV boxes etc
- YubiKey version 2.2 and later only

RFID YubiKey

- YubiKey combined with RFID chip
- Provides security in both digital and physical world



Automated Logistics



Yubico Provides

- YubiKey – different variants
- Personalization software
- Low-level OTP parsing libraries
- Validation protocol specification
- Clients to validation server
- Online Validation server
- Hosted demo servers

Yubico Provides (contd)

- Yubico Forum for support
 - <http://forum.yubico.com/>
- Yubico Wiki for knowledge
 - <http://wiki.yubico.com/>
- PAM module
 - Documentation describing how FreeRadius is used to provide a Radius server
- OpenID server - <http://openid.yubico.com/>
- YubiKey plugin to simpleSAMLphp

Personalization Software

- <http://yubico.com/developers/personalization/>
- Alternatives:
 - 1.Windows Personalization Tool
 - 2.Windows COM/ActiveX component
 - 3.Free software portable library + tool
 - C code, BSD license - packaged by Debian etc
 - <http://code.google.com/p/yubikey-personalization/>
 - 4.Third-party Mac graphical interface

Lock code

- YubiKeys can be protected with a lock code
- Prevents unauthorized re-programming of the YubiKey
- The AES key can never be read out from the device
- Recommendation: If you personalize YubiKeys yourself, set a random locking code on each device

Low-level OTP parsing

- <http://code.google.com/p/yubico-c/>
- Core library written in C
- BSD license – included in Debian, Fedora etc
- Functionality ported to Java, PHP, Perl, Python, ...
- Low-level, example interfaces:

```
extern void yubikey_parse (const uint8_t token[YUBIKEY_BLOCK_SIZE],
                          const uint8_t key[YUBIKEY_KEY_SIZE], yubikey_token_t out);
extern void yubikey_modhex_encode (char *dst, const char *src,
                                   size_t srctime);
extern int yubikey_modhex_p (const char *str);
extern uint16_t yubikey_crc16 (const uint8_t * buf, size_t buf_size);
extern void yubikey_aes_decrypt (uint8_t * state, const uint8_t * key);
...

```

DEMO

- 1.Reprogram a YubiKey with 'ykpersonalize'**
- 2.Debug generated OTP using 'ykdebug'**

Validation Server Protocol

- Protocol specification online:
 - <http://yubico.com/developers/api/>
- Concept of client identity
- Optional HMAC signing of requests/response
- Simple Query and response (v1):
 - <http://api.yubico.com/wsapi/verify?id=42&otp=vvvvvcurikvhjcvnlInbecbkubjvuittbifhndhn>
 - h=hhbVQZYvkEWUdhYjx1hjB/yeW/Y=
t=2008-01-11T03:51:21Z0079
status=OK

Client ID & Key

- Generate your own client identity & HMAC key online:
 - <http://yubico.com/developers/api/>
- You will be allocated one integer and a newly generated random base64 string
- Used by client software to sign requests and validate responses

DEMO

- 1. Validate OTP against online demo**
- 2. Verify an OTP against Yubico Validation Server using command line tools**



Basic Login Demo

Demo YubiKey only

Congratulations simon!

You have been successfully authenticated with the YubiKey.

- » [Try again](#)
- » [Demo YubiKey + password](#)
- » [Demo YubiKey + username/password](#)
- » [Set username/password for Demo](#)
- » [Back to main page](#)

Technical details

More information about the performed transaction:

HTML fields

```

mode      one-factor
key       ekhgjhbctrngcdjkbttbkhkhriubhjiubfburtjjukb
identity  ekhgjhbctrng
db_realname simon

```

Authentication Output

```

h=CwtSC//DjEbDahmF2O6ZWka5kDg=
t=2010-04-19T09:04:13Z0877
status=OK

```



```
jas@mocca: ~  
File Edit View Terminal Help  
jas@mocca:~$ wget -q -O - 'https://api.yubico.com/wsapi/verify?id=1&otp=ekhgjhbc  
trgnvvkftttuhlrkibeutukkgkdhibljhr'  
h=WsK3+VXb9vU/KVnny7xV4Wd1fsA=  
t=2010-04-19T09:32:27Z0185  
status=OK  
  
jas@mocca:~$ wget -q -O - 'https://api.yubico.com/wsapi/verify?id=1&otp=ekhgjhbc  
trgnvvkftttuhlrkibeutukkgkdhibljhr'  
h=nFjt9rtSyseUFRXosXtgk1K/Vjw=  
t=2010-04-19T09:32:32Z0165  
status=REPLAYED_OTP  
  
jas@mocca:~$ wget -q -O - 'https://api.yubico.com/wsapi/verify?id=1&otp=ekhgjhbc  
trgnvvkftttuhlrkibeutukkgkdhibljhr'  
h=UGPNBDMAMfy0JQCgjh1z6MLLMAM=  
t=2010-04-19T09:32:33Z0765  
status=REPLAYED_OTP  
  
jas@mocca:~$
```

Validation Protocol v2.0

- Supports distributed servers
- Each client query in parallel all servers
- Servers all talk to each other
- Clients waits for positive validation
- While waiting, will reject OTP if any negative response is received
- Some servers may respond “replayed request” if they became aware of the query through another validation server first

Validation server clients

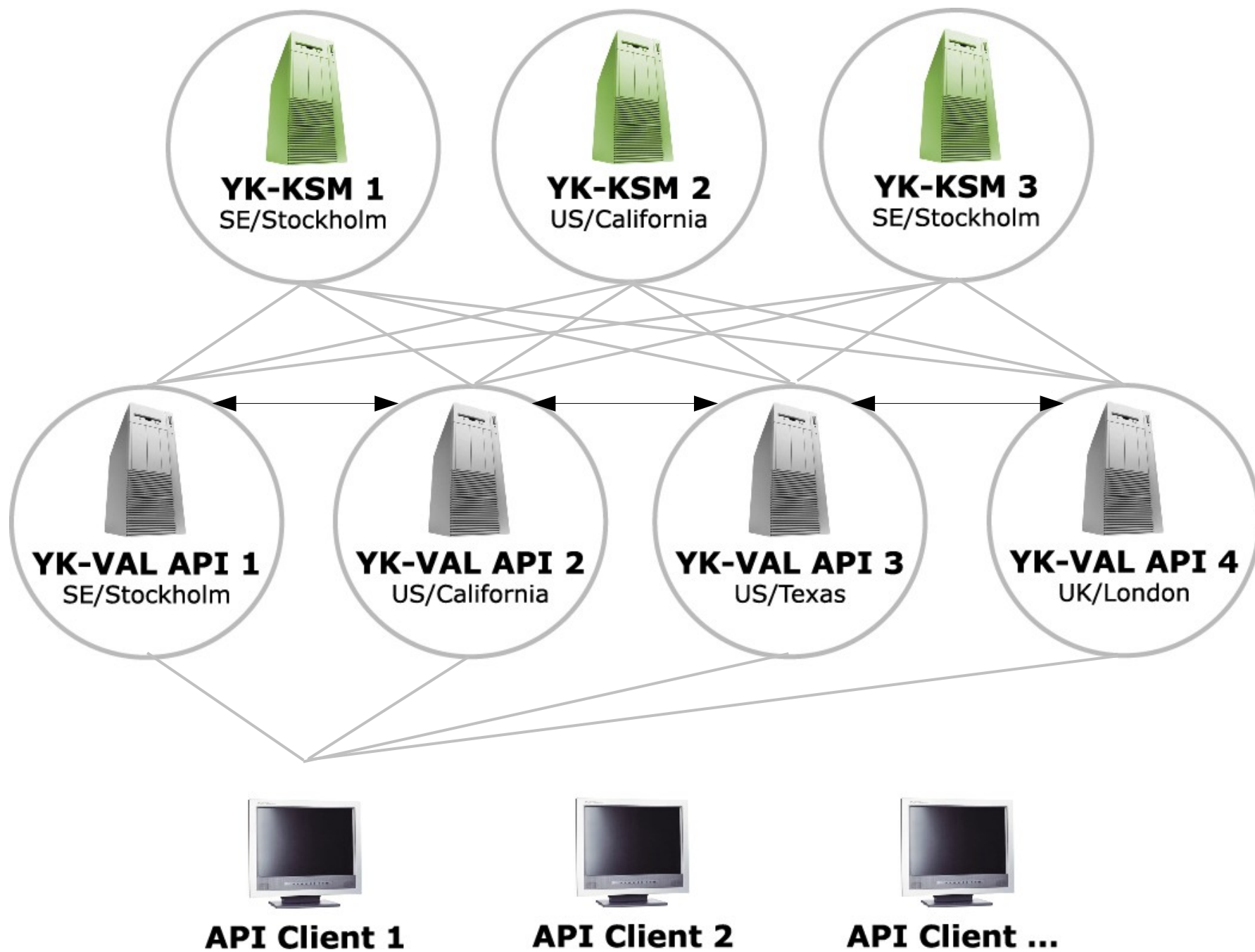
- C library, PHP module, many others...
- PHP code easy to install and use
 - `wget http://php-yubico.googlecode.com/files/Auth_Yubico-1.9.tgz`
`pear install Auth_Yubico-1.9.tgz`

```
<?php
require_once 'Auth/Yubico.php';
$otp = "ccbdddeertkrctjkkcglfndnliahnvekchkcctif";

# Generate a new id+key from https://api.yubico.com/get-api-key/
$yubi = &new Auth_Yubico('42', 'FOOBAR=');
$auth = $yubi->verify($otp);
if (PEAR::isError($auth)) {
    print "<p>Authentication failed: " . $auth->getMessage();
    print "<p>Debug output from server: " . $yubi->getLastResponse();
} else {
    print "<p>You are authenticated!";
}
?>
```

Validation Server

- YK-VAL: YubiKey Validation server
 - Free software <http://code.google.com/p/yubikey-ksm/>
 - YK-VAL responsible for verifying YubiKey OTPs following Yubico's web service API protocol
 - YK-VAL requests AES decryption from YK-KSM
- YK-KSM: YubiKey Key Storage Module
 - Free software <http://code.google.com/p/yubikey-val-server-php/>
 - YK-KSM responsible for storing AES keys and decrypting incoming OTP



Scalability

- Internal redundancy: YK-VAL is configured to query any number of YK-KSM machines and will use the first valid answer
- The YK-KSM can be cloned easily:
 - No synchronization of data necessary beyond loading of AES keys
- The YK-VAL can be replicated
 - Requires loose synchronization of OTP counter fields between YK-VAL instances

It is currently Mon Apr 19, 2010 7:52 am



The Yubico Forum is intended for anyone who wants to learn, question, comment or contribute to Yubico's technology. To avoid spam and misuse we only allow YubiKey owners to post comments. If you do not have a YubiKey, you can send your question to forum@yubico.com or order a YubiKey at www.yubico.com/products/order.

[View unanswered posts](#) | [View active topics](#)

[Board index](#) All times are UTC - 8 hours

Forum	Topics	Posts	Last post
General Meta-issues and general announcements.	33	89	Thu Apr 15, 2010 6:04 am samir ➔
Yubikey			
Hardware / Firmware / Config Tools Questions and discussions related to hardware aspects of the Yubikey.	102	526	Wed Apr 14, 2010 2:52 am JakobE ➔
Web Service Client Software Using our online verification server for validating Yubico One-Time Passwords. Here goes questions about the PHP class, the PAM module, the Java client library, and so on.	62	262	Sat Apr 17, 2010 7:19 am ionr ➔
Server Side Software Validating Yubikey OTPs using the AES key directly, typically only for server integration or disconnected use. Here goes questions related to 'yubico-c' and 'yubico-j' projects.	76	325	Tue Mar 30, 2010 11:08 am asq ➔
Hi-Priority Projects, Help Needed, Request for Proposals These are solutions many are asking for, and a place to post Yubikey integration related jobs.	9	56	Wed Apr 14, 2010 10:13 pm rootinflux ➔
Community Projects Discussions about new projects to use the YubiKey with a new protocol, language or environment. Ideas include Python or Perl based basic server libraries, Windows login support, but can be anything.	32	139	Sun Feb 28, 2010 7:23 am murlock ➔
YubiKey 2.0 Discussions related to YubiKey v2.0 specific features.	28	105	Tue Apr 13, 2010 7:15 am samir ➔
OpenID OpenID related uses of the Yubikey, including discussions for how to use Yubico's OpenID Server.	21	83	Fri Mar 19, 2010 3:21 pm adedommelin ➔
Promote Your YubiKey-related Solutions Developers can list well tested, documented YubiKey applications here. Buyers can look for YubiKey related applications and developers here.	18	104	Sat Feb 20, 2010 4:38 pm marks ➔

Main Page

yubico

- [Main Page](#)
- [Integrators](#)
- [New Ideas](#)
- [Wiki Support](#)

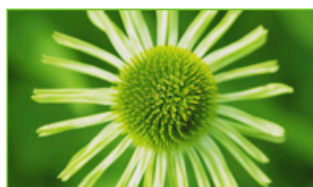
authors/administrator

- [Recent changes](#)
- [Help](#)
- [Installed extensions](#)

search

toolbox

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)



The YubiKey Wiki

The portal for software and services supporting the YubiKey

Everyone is welcome to read, but to avoid misuse only YubiKey users can edit.

To promote your solution [Order your YubiKey](#)

VPN, RADIUS, Windows login

Organization	Product Description	Authentication Mode	Business Model	Region
ActivIdentity	Fortress authentication server	OTP	Licensed software	USA, Global
AuthLite	Windows Active Directory integrated login	OTP	Licensed software	USA, Global
Cybercom	Trusted Security Server			
Mike Clark	Yubidus - YubiKey enabled radius server		Free open software	Global
MobilityGuard	MG authentication platform		Licensed software	Sweden, Chile
Mi-Token	Authentication software	OTP, OATH	Licensed software	APAC
Radiator Radius	Enterprise authentication server	OTP	Licensed software	
Rohos	Windows Local login, Remote Desktop login	OTP, YubiKey ID	Licensed software	Europe, Global
Sun	Open SSO authentication server	OTP	Free open software	Global
YubiRadius	Basic YubiKey enabled RADIUS authentication service	OTP	Free open software	Global
RADIUS_on_Premise	Proof of concept implementation of YubiKey enabled RADIUS Server	OTP	Free open software	Global

CMS & editing

Organization	Product Description	Authentication Mode	Business Model	Region
Crasman	Crasmanager CMS		Licensed software	Finland, Global
Drupal	CMS software	OTP	Free open software	Global

PAM

- Pluggable Authentication Module (PAM)
- User authentication and authorization under GNU/Linux & Solaris
- Used in other environments to achieve modularity, e.g., Radius
- Challenge-Response approach
 - <http://code.google.com/p/yubico-pam/>
 - C code, BSD/GPL, Debian packages
- Useful for SSH and Desktop login

OpenID

- Decentralized web-based authentication system
- Serious phishing security issues!
 - One-time passwords are cost effective solution
 - SMS passcodes, X.509 https other approaches
- Three parties:
 - 1.Identity Provider (IdP)
 - 2.Relying Partner (RP)
 - 3.User - identified by an OpenID URL

Yubico OpenID server

- Based on JanRain's OpenID library and their example OpenID Server
- Minimally modified to support YubiKey
- <http://code.google.com/p/yubico-openid-server/>
- Running on <http://openid.yubico.com/> as free service – all existing YubiKeys have an OpenID URL automatically
- Easy to use with your own URL, just add two HEAD META tags to your HTML page
- No vendor lock-in!

Demo!

The image is a collage of browser screenshots demonstrating the Wikitravel OpenID login process. The screenshots are as follows:

- Wikitravel Main Page:** The top-most screenshot shows the Wikitravel Main Page with the URL `http://wikitravel.org/en/Main_Page`.
- Login with OpenID:** The second screenshot shows the "Login with OpenID" page with the URL `http://wikitravel.org/en/Special:OpenIDLogin`. It explains that Wikitravel supports the OpenID standard for single sign-on.
- Yubico - Trust the net.:** The third screenshot shows the Yubico verification page with the URL `http://openid.yubico.com/server.php?openid.assoc_handle=...`. It features a green background with the Yubico logo and the text "trust the net".
- Verification succeeded:** The bottom-most screenshot shows the Wikitravel "Verification succeeded" page with the URL `http://wikitravel.org/en/Special:OpenIDFinish?nonce=JaCt...`. It displays the message "Verification succeeded" and a link to "Return to Main Page".

In the bottom-left corner, there is an inset photograph of a hand inserting a USB drive into a laptop's USB port.

At the bottom of the browser window, there is a status bar with the text "Done" on the left and "Tor Disabled" on the right.

SAML

- Security Assertion Markup Language
- Format to exchange authentication and authorization information between security domains
- Specified by OASIS: www.oasis-open.org
- Primary use case is web browser sign on but protocol is transport agnostic

Yubico SAML Server

- simpleSAMLphp (SSP) PHP based SAML server with YubiKey plugin
- Sun/Oracle's OpenSSO server with YubiKey plugin
- Both are free software, commercial alternatives exists
- YubiKey hosts SSP as <http://saml.yubico.com/>
- Free service for all YubiKey owners

YubiHSM



YubiHSM Quick Facts

- Currently in beta testing with customers
- Small USB device (0.2W) acting like a serial device - GNU/MAC/Windows-friendly
- Priced at \$500 with no maintenance fee
- AES encrypt/decrypt/decrypt-compare using key in YubiHSM
- HMAC-SHA1 with key in YubiHSM (HOTP/TOTP)
- AES-based NIST SP800-90 CTR-DRBG random number generator

More facts

- Holds 40 AES/HMAC keys indexed by a 32-bit key handle
- Fairly small set of interface functions
 - YSM_NULL, YSM_SYSTEM_INFO_QUERY, YSM_ECHO, YSM_KEY_STORAGE_UNLOCK, YSM_BUFFER_LOAD, YSM_BUFFER_RANDOM_LOAD, YSM_NONCE_GET, YSM_AEAD_GENERATE, YSM_RANDOM_AEAD_GENERATE, YSM_BUFFER_AEAD_GENERATE, YSM_AEAD_DECRYPT_CMP, YSM_AEAD_YUBIKEY_OTP_DECODE, YSM_DB_YUBIKEY_AEAD_STORE, YSM_DB_YUBIKEY_OTP_VALIDATE, YSM_TEMP_KEY_LOAD, YSM_AES_ECB_BLOCK_ENCRYPT, YSM_AES_ECB_BLOCK_DECRYPT, YSM_AES_ECB_BLOCK_DECRYPT_CMP, YSM_HMAC_SHA1_GENERATE, YSM_RANDOM_GENERATE, YSM_RANDOM_RESEED
- Reference Python code available on GitHub
 - Third-party java code being published
- Documented interface, please write your own!

Background

- Yubico operates validation server for a fleet of YubiKey's
- We needed to secure millions of AES keys stored on servers world-wide
- Traditional HSMs are expensive, cannot store millions of keys and only offer encrypt/decrypt interfaces
 - Attackers getting root would get our AES keys!
- We needed an inexpensive solution and interfaces for native YubiKey OTP parsing and decrypt-and-compare

Wider usage

- Threat model: someone roots your server
 - Physical attacks (stealing the machine) is outside of our threat model – we use the traditional security industry to mitigate that.
- Goal: Minimize what the attacker can achieve by becoming root
- How #1: Make the data stored on the server useless to an attacker

YubiHSM Indirect Mode

- Based on AES CCM - RFC 3610
 - Early AEAD cipher mode, easy to implement
- Enables support of millions of “virtual” keys protected by YubiHSM
- Used here to do “key wrap”, i.e., encrypt an AES key or a (hashed) password
- Encrypted AEAD-blob stored on server
- On request, YubiHSM takes the AEAD-protected key and either an OTP or (hashed) password for comparison

Validating a password

- Let's say you are building a server to validate passwords for millions of users
- Perform a PBKDF2 iterated hash as early as possible, using a per-user salt/count
- Query a server with a YubiHSM with input (AEAD-blob, potential-PBKDF2)
- Server uses `AEAD_DECRYPT_CMP` and returns yes/no
- No data stored on server is useful for the attacker!

Caveats

- Key management of the YubiHSM keys becomes critical
- Authorization of AEAD generation and storage is important
- Best practice is to generate a random key with the same key handle and configure two YubiHSMs in pair at the same time on trusted machine
- One YubiHSM will have permissions to generate AEADs (the set-password machine) and another to validate passwords using the AEADs (the validate-password machine)

Thank you for listening!

Questions?

yubico
trust the net