

VELOCITY
S O F T W A R E

zVWS and zSSL
Topics in SSL on z/VM

Velocity Software Inc.
196-D Castro Street
Mountain View CA 94041
650-964-8867

Velocity Software GmbH
Max-Joseph-Str. 5
D-68167 Mannheim
Germany
+49 (0)621 373844

Rick Troth
Velocity Software
<rickt@velocitysoftware.com>
<http://www.velocitysoftware.com/>

VM and Linux Workshop 2014
NC A&T, Greensboro

Copyright © 2014 Velocity Software, Inc. All Rights Reserved.
Other products and company names mentioned herein may be
trademarks of their respective owners.

The content of this presentation is informational only and is not intended to be an endorsement by Velocity Software. (ie: I am speaking only for myself.) The reader or attendee is responsible for his/her own use of the concepts and examples presented herein.

In other words: Your mileage may vary. “It Depends.”
Results not typical. Actual mileage will probably be less.
Use only as directed. Do not fold, spindle, or mutilate. Not to be taken on an empty stomach. Refrigerate after opening.

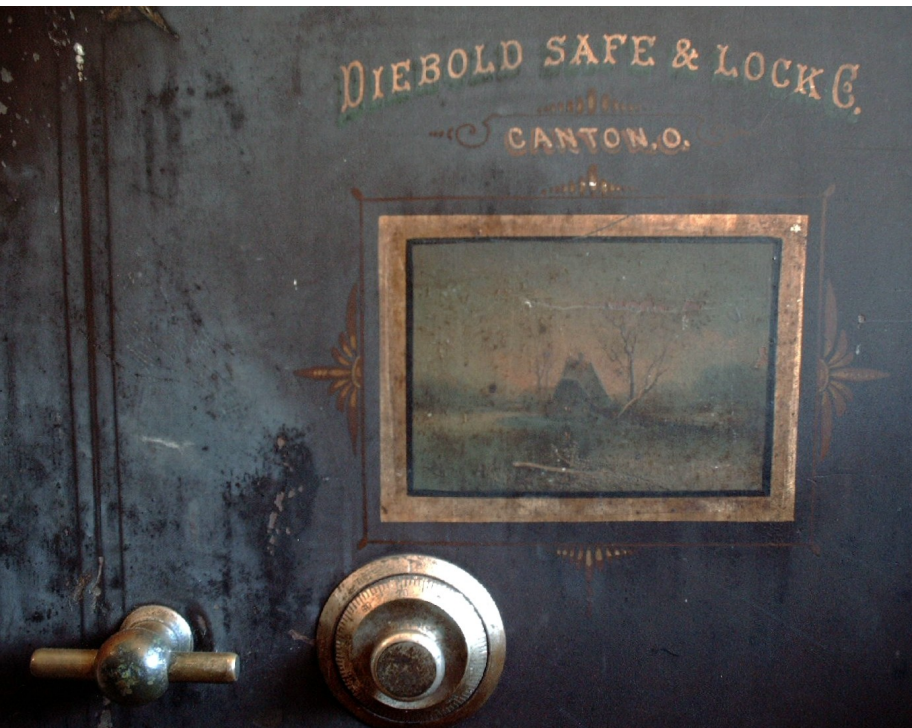
In all cases, *“If you can't measure it, I'm just not interested.”*

Ciphers: ancient to mechanical to now
PGP, SSL, SSH ... PKI, X.509, SSL/TLS
zSSL, VM SSL, client certs (smart cards)

Further Study:

- PGP Web-of-Trust
- SSH keys for log-on
- DNSSEC
- Keybase.io

Protecting Information



Data at Rest
Data in Transit



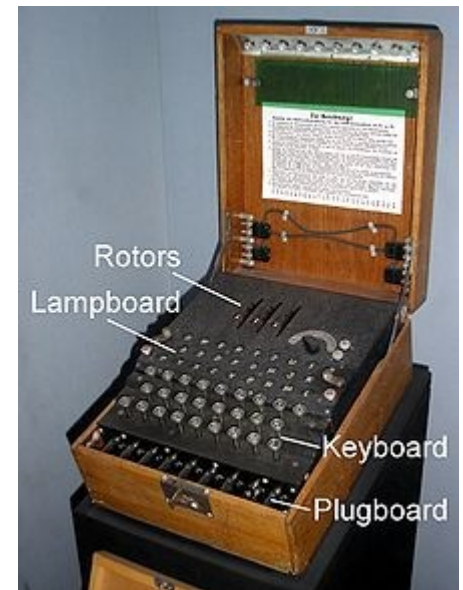
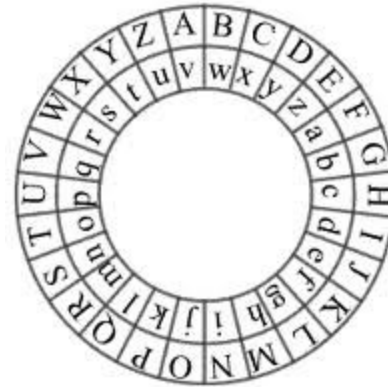
Symmetric Crypto

Early ciphers

- Caesar
- Jefferson
- Enigma, Lorenz

Passwords

One-time use



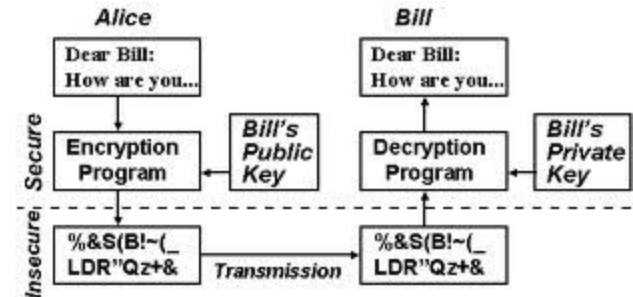
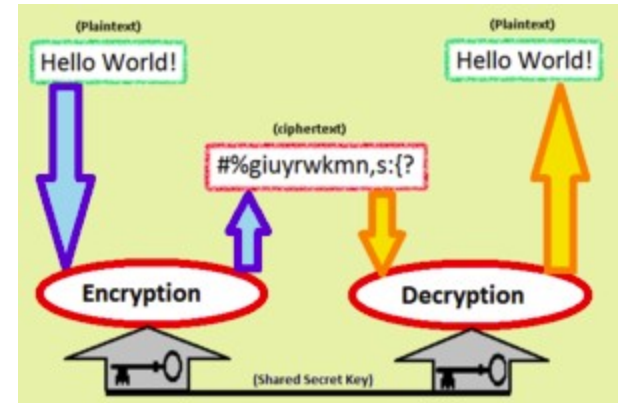
Asymmetric Crypto

What if someone
got the password?

Rivest, Shamir, Adleman
public key and private key ... *asymmetric*

[http://en.wikipedia.org/wiki/
Public-key_cryptography](http://en.wikipedia.org/wiki/Public-key_cryptography)

Cocks, et al, GCHQ 1973



Encryption plus Authentication

Encrypt with public key (of recipient)

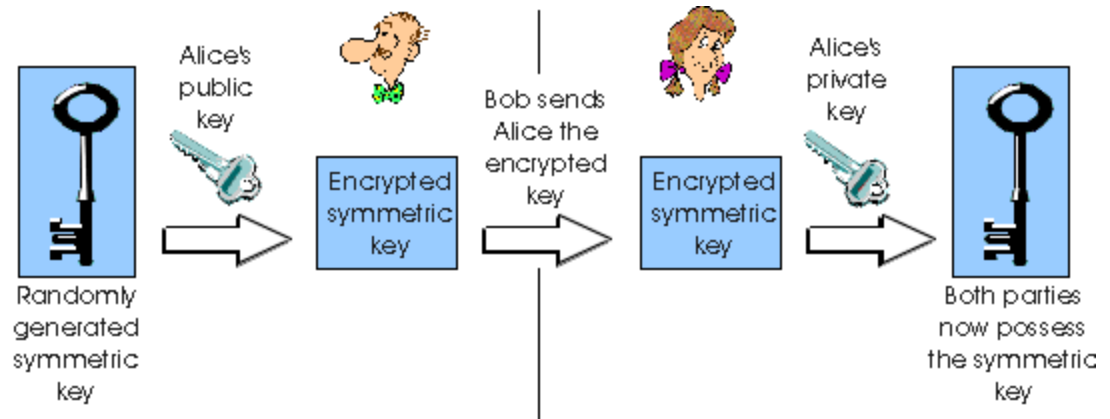
Decrypt with secret key

Sign with secret key

Verify with public key (of sender)

Combo Crypto

Random “session key” symmetric (single)
Encrypt that with asymmetric (dual)
Encrypt payload with session key
Send asym-encrypted session key
and sym-encrypted payload



Transport Layer Security

Handshake authenticates

SSL provides a “channel”

Compare to SSH

Contrast with PGP/GPG (data at rest)

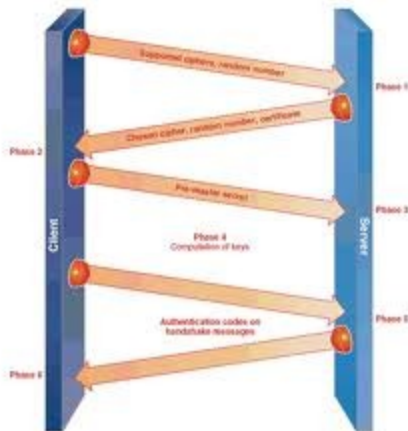
Content types

Hex	Dec	Type
0x14	20	ChangeCipherSpec
0x15	21	Alert
0x16	22	Handshake
0x17	23	Application

+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	Content type			
Bytes 1..4	Version		Length	
	(Major)	(Minor)	(bits 15..8)	(bits 7..0)
Bytes 5..(m-1)	Protocol message(s)			
Bytes m..(p-1)	MAC (optional)			
Bytes p..(q-1)	Padding (block ciphers only)			

SSL Handshake

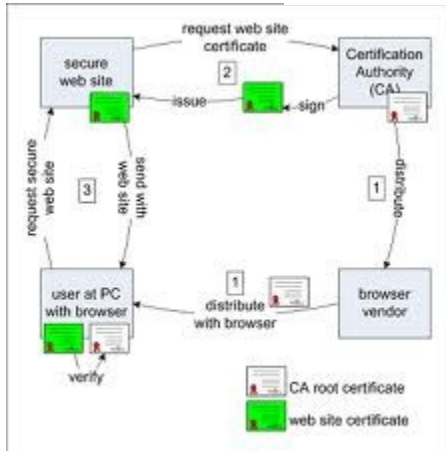
Authenticate the server
Establish a secure channel
Uses existing network



Message Types	
Code	Description
0	HelloRequest
1	ClientHello
2	ServerHello
11	Certificate
12	ServerKeyExchange
13	CertificateRequest
14	ServerHelloDone
15	CertificateVerify
16	ClientKeyExchange
20	Finished

Does not protect “data at rest”

Public Key Infrastructure



CA certificate(s) pre-loaded
WS admin requests assertion
CA signs WS request
WS admin loads that

.....
Browser hits WS,
compares signature chain
Browser/WS agree on
session keys



Got zVWS? Then install zSSL

Installation process for zSSL automatically generates a key pair and creates a self-signed server certificate.

Also creates a certificate request which you can submit to your CA of choice.

VSIMAINT - configure zSSL

```
x3270-3 192.168.5.67
File Options
VSIS SETUP Velocity Software Inc. ZSSL PR0D4152
Installation and Configuration

To Install a Product the following must be performed in the order shown
Some functions may not be valid for the selected Product

For additional help press PF1, Press PF3 to return to the previous screen
Press PF12 to exit the installer

Function Valid Function
-----
Run Pre-Installation Tasks Y
Product Installation Y
Run Post-Installation Tasks N
_Edit Run-Time Configuration Files Y
DCSS Configuration Y

PF1: Help PF2: Select PF3: Return PF12: Cancel
018/003
```


VSIMAINT - configure zSSL

```
x3270-3 192.168.5.67
File Options
EDITSCRN Velocity Software Inc. ZSSL PRD4152
Product Configuration / Post Installation Tasks

The following Configuration Files are used by ZSSL
Place the cursor on a Configuration File and press PF2 to configure

Configuration Files Disk Modified
-----
CONFIG ZWBSADM CONFIG 2012-12-08 21:11:38
CONFIG ZWBS01 CONFIG 2012-12-08 21:11:33
CONFIG ZWBS02 CONFIG 2012-12-08 21:11:29
CONFIG ZWBS03 CONFIG 2012-12-08 21:11:24
CONFIG ZWBS04 CONFIG 2012-12-08 21:11:19
CONFIG ZWBS05 CONFIG 2012-12-08 21:11:07
DEFAULT WEBHOST CONFIG 2013-05-21 11:31:02
DEFAULT ZADMIN CONFIG 2013-03-20 09:53:02
ESALOG SERVERS CONFIG 2011-06-10 12:59:28
_ ZSSL CONFIG CONFIG 2013-05-15 13:54:28

PF1: Help PF2: Select PF3: Return PF5: Add file PF12: Cancel
018/002
```

VSIMAINT - X.509 data

```
x3270-3 192.168.5.67
File Options
ZSSL Velocity Software Inc. ZSSL PRD4152
ZSSL CONFIG Configuration

CN rmtzvm.velocitysoftware.com
O Velocity Software
OU TrothR
L Grove City
ST OH
C US
Email rickt@velocitysoftware.com
key size 2048
certificate serial number 20
comment Velocity Software zSSL Generated Certificate

PF1: Help PF2: Validate/Save PF3: Exit PF10: Default PF12: Cancel
004/015
```

VSIMAINT - keys, cert, req

```
x3270-3 192.168.5.67
```

File	Options
------	---------

```
ZVPS      FILELIST A0  V 169  Trunc=169  Size=3  Line=1  Col=1  Alt=36
Cmd  Filename  Filetype  Fm  Format  Lrecl  Records  Blocks  Date      Time
-   ZSSLCERT  CRQ       A1  V       64      18       1  2013-05-10 15:28:40
    ZSSLCERT  X509CERT C1  V       980     1       1  2012-09-28 15:58:09
    ZSSLCERT  KEYP     C1  V      1192    4       1  2012-09-28 14:25:55
```

1= Help 2= Refresh 3= Quit 4= Sort(type) 5= Sort(date) 6= Sort(size)
7= Backward 8= Forward 9= FL /n 10= Share 11= XEDIT/LIST 12= Cursor

====>

X E D I T 1 File

003/001

Got zVWS? Then install zSSL

It's that easy!

Self-signed certificate is immediately ready.
Certificate request is available too.
Submit it to your CA of choice, if needed.

Server with Self-Signed Cert



This Connection is Untrusted

You have asked Firefox to connect securely to **192.168.5.44:2983**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

- ▶ **Technical Details**
- ▶ **I Understand the Risks**

Certificate Authorities - StartSSL



StartSSL™ Free (Class 1)

128/256-bit Encryption, **1 Year** Validity
Legitimate SSL/TLS + S/MIME Certificates
No Charge, Unlimited + 100 % Free



StartSSL™ Extended Validation

128/256-bit Encryption, **2 Years** Validity
Highest Level Third Party Assurance
Green Extended Trust Indicator
Multiple Domain Names (UCC)
Special Offer - US\$ 199.90



Hardware

Aladdin® USB eToken Pro
Aladdin® Smart Cards + Reader
Original Driver Software + PKI Client
Enterprise PKI Customized Solutions



Internationally Recognized

WebTrust for CAs + WebTrust EV Certified
Recognized by major browsers + software vendors



StartSSL™ Verified (Class 2)

128/256-bit Encryption, **2 Years** Validity
Legitimate SSL/TLS + S/MIME + Object Code
Wild Cards, Multiple Domain Names (UCC)
Unlimited Certificates - US\$ 59.90



High Protection

StartSSL™ High Level Protection
No MD5 Hashes, Weak Key Scans
Minimum 2048-bit Strong RSA Keys



Authentication

StartSSL™ Authentication SSL Protected
Open Identity Authentication Provider
Click here to log into your
StartSSL™ Account



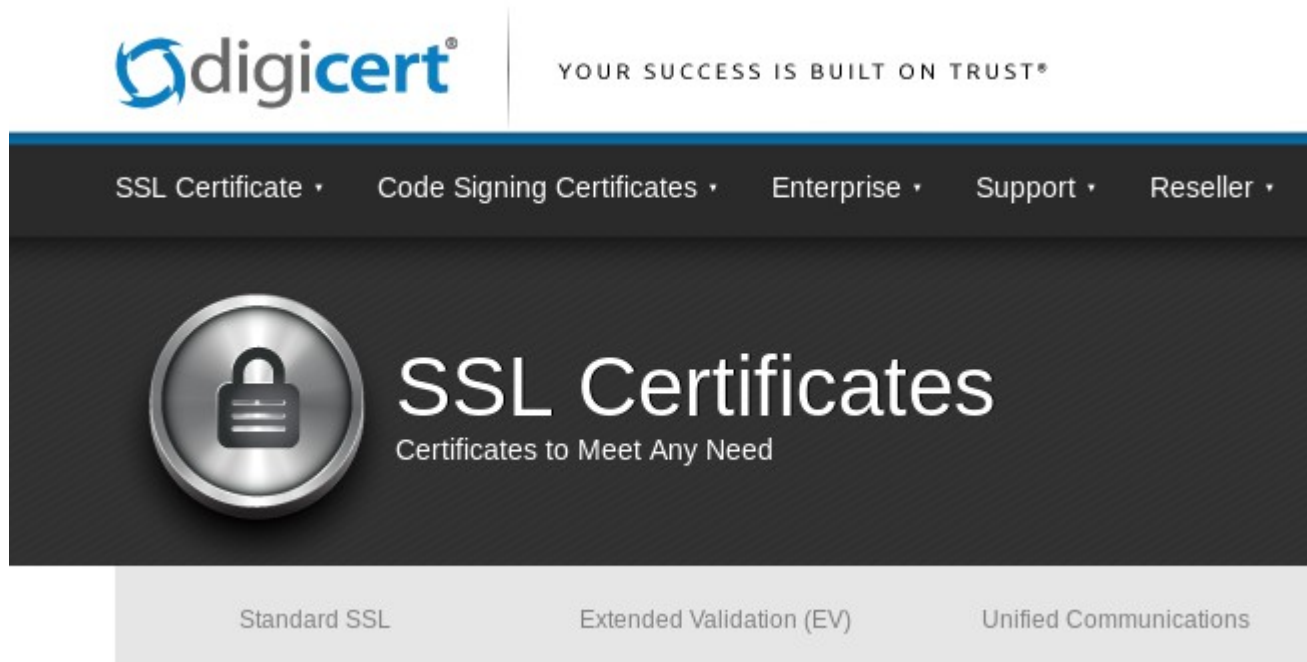
Easy Enrollment

Sign-up and you will receive right away an S/MIME
client-certificate and a digital StartSSL™ Open Identity
without charge during the easy three-step enrollment!



<https://www.startssl.com/>

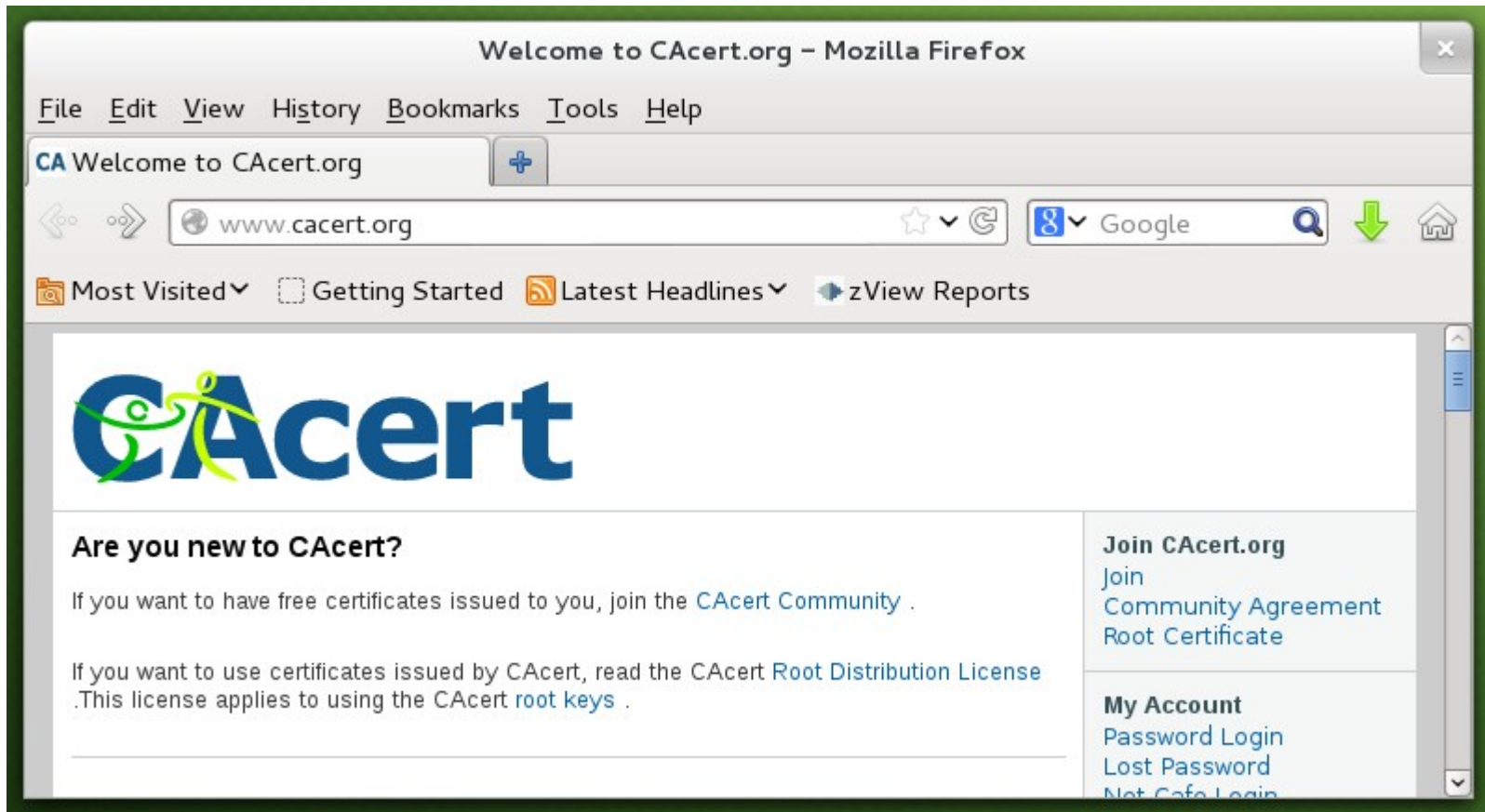
Certificate Authorities - DigiCert



The screenshot shows the DigiCert website header with the logo and the tagline "YOUR SUCCESS IS BUILT ON TRUST®". Below the header is a navigation menu with links for "SSL Certificate", "Code Signing Certificates", "Enterprise", "Support", and "Reseller". The main content area features a large circular icon of a padlock, followed by the text "SSL Certificates" and "Certificates to Meet Any Need". At the bottom of the main content area, there are three buttons: "Standard SSL", "Extended Validation (EV)", and "Unified Communications".

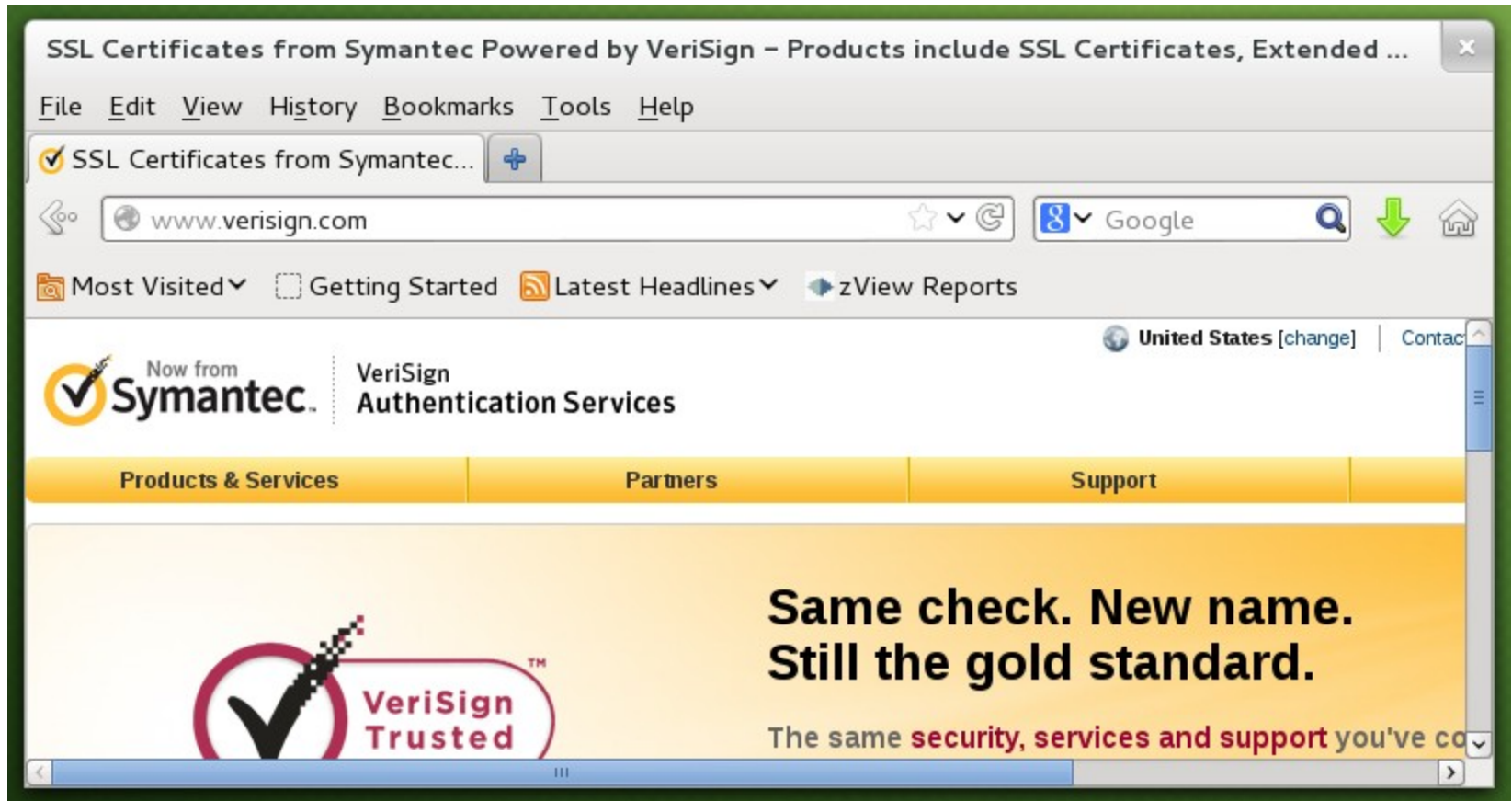
<http://www.digicert.com/ssl-certificate.htm>

Certificate Authorities – CAcert



<http://www.cacert.org/>

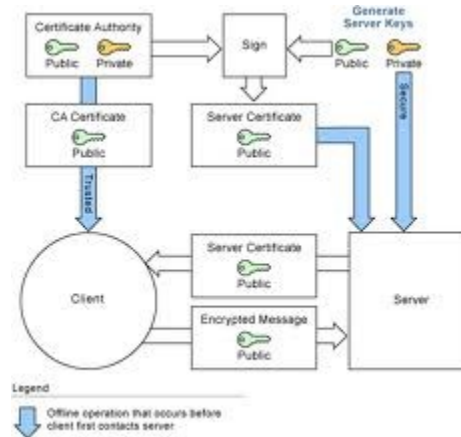
Certificate Authorities - VeriSign



<http://www.verisign.com/>

VM SSL Key Management

Set up GSKADMIN and wire it into the stack



Sign onto GSKADMIN

Use 'gskkyman' command

VM SSL Key Management

```
File Options
Ready; T=0.01/0.01 12:51:46
gskkyman

Database Menu

1 - Create new database
2 - Open database
3 - Change database password
4 - Change database record length
5 - Delete database
6 - Create key parameter file
7 - Display certificate file (Binary or Base64 ASN.1 DER)

0 - Exit program

Enter option number:

-
RUNNING ZVMV5R40
031/001
```

Create a key database ...

- Option 1
- Filename "Database.kdb"
- 3700 days = 10 years, 6 weeks
- Default record size

Fix file access ...

```
openvm permit /etc/gskadm/Database.kdb rw- r-- ---  
openvm permit /etc/gskadm/Database.sth rw- r-- ---
```


VM SSL Key Management

```
File      Options
Key Management Menu

Database: /etc/gskadm/Database.kdb
Expiration: 2022/07/30 21:38:58

1 - Manage keys and certificates
2 - Manage certificates
3 - Manage certificate requests
4 - Create new certificate request
5 - Receive requested certificate or a renewal certificate
6 - Create a self-signed certificate
7 - Import a certificate
8 - Import a certificate and a private key
9 - Show the default key
10 - Store database password
11 - Show database record length

0 - Exit program

Enter option number (press ENTER to return to previous menu):

-
RUNNING  ZVMV5R40
031/001
```

Create a self-signed certificate ...

- Option 6
- Option 7, server cert with 4096-bit RSA key
- Option 3, SHA-256 signature digest
- Enter a label, UPPER CASE
- Enter X.509 stuff

Apply that label to a “secured” TCP port

Create new certificate request ...

- Option 4
- Option 3, cert with 4096-bit RSA key
- Enter filename
- Enter a label, UPPER CASE again
- Enter X.509 stuff

File is PEM encoded; send it to your CA

To use client certificates, or devices like common access cards, install a “CA bundle”.

CABUNDLE CRT ← in CONFIG directory

a collection of “signing certificates”

Copy **ca-bundle.crt** (eg: from Apache)

Create by hand (PEM encoded)

Create from example

Sample CA bundle can be found at:

<http://curl.haxx.se/ca/cacert.pem>

CGI variables

`SSL_CLIENT_S_DN`, `SSL_CLIENT_I_DN`,
`SSL_CLIENT_M_VERSION`, `SSL_CLIENT_M_SERIAL`,
`SSL_CLIENT_V_START`, `SSL_CLIENT_V_END`,
`SSL_CLIENT_A_KEY`, and `SSL_CLIENT_A_SIG`

Crypto Concepts - Trust Models

Peer-to-Peer

- PGP style

Third Party / Centralized

- PKI style

Manual Assertion

- Self-signed certificates

Question:

which works best for the application?

SSL and TLS (PKI)

- originally for HTTPS, now many protocols
- third party trust
- X.509 certificates (contain public keys)

SSH

- variable trust models
- keys

PGP/GPG

- peer-to-peer trust
- keys

'ssh-keygen' command

- Generates pub (“ .pub”) and sec, two files

Append pub to “authorized_keys” file of target user(s) on target system(s)

```
$ id
```

```
uid=51668(rickt) gid=51668(rickt)
```

```
$ ssh trothr@rmtlinux
```

```
Last login: Fri Jun 27 05:52:53 2014
```

```
-sh-3.2$ id
```

```
uid=51667(trothr) gid=51667(trothr)
```

Generate a key pair

```
gpg --gen-key
```

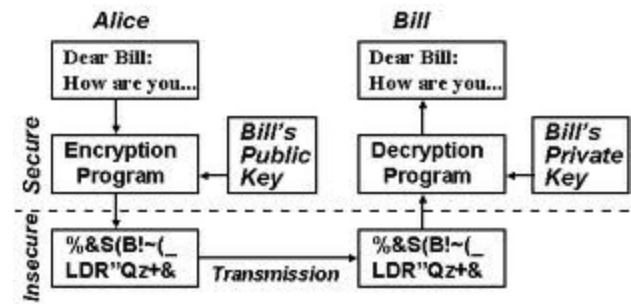
Export your pub key, sign others

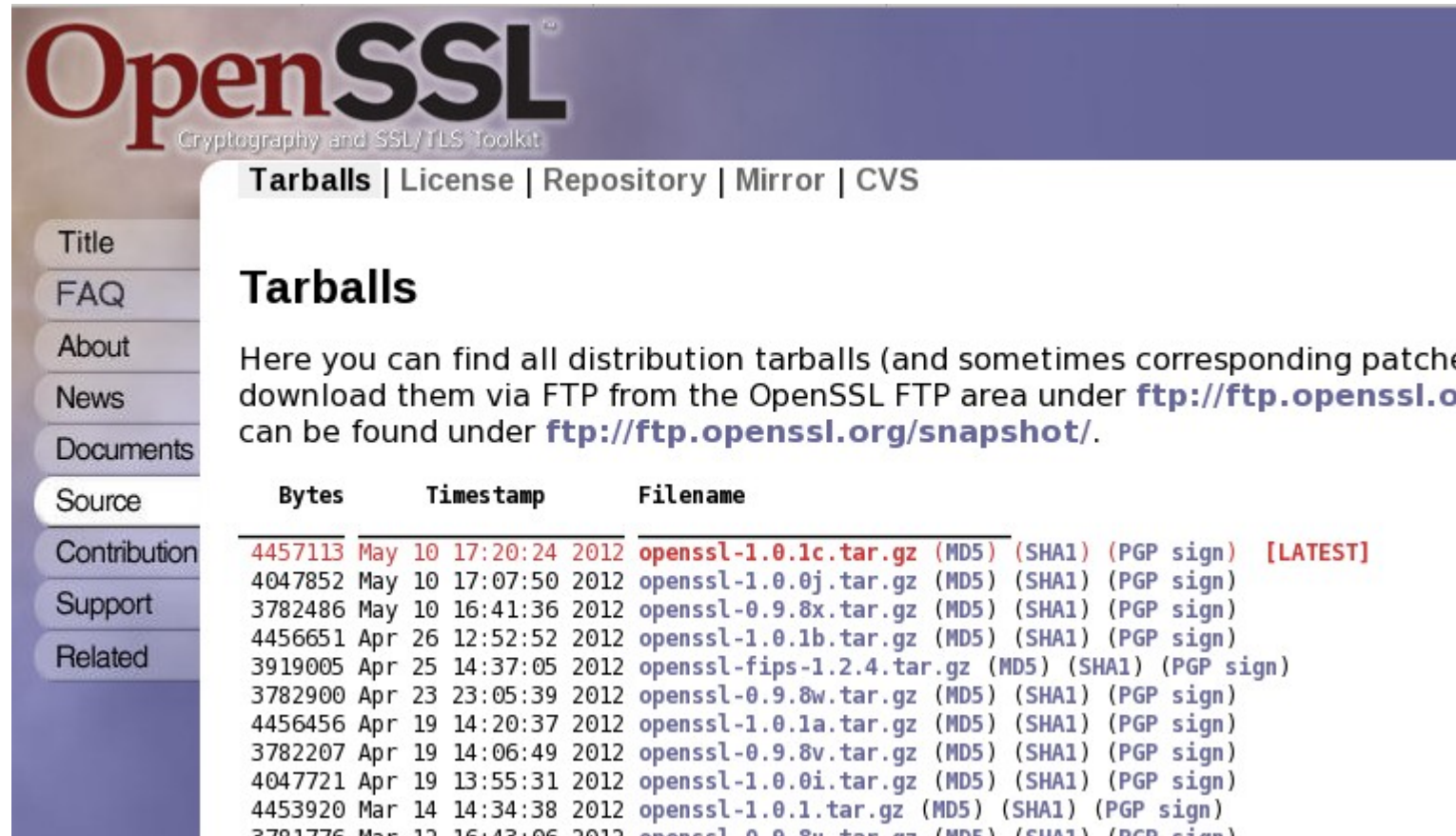
```
gpg --armor --export
```

```
gpg --sign-key other-user's-key
```

Import signed keys and signatures

```
gpg --import
```





OpenSSL
Cryptography and SSL/TLS Toolkit

[Tarballs](#) | [License](#) | [Repository](#) | [Mirror](#) | [CVS](#)

Tarballs

Here you can find all distribution tarballs (and sometimes corresponding patches) download them via FTP from the OpenSSL FTP area under <ftp://ftp.openssl.org> can be found under <ftp://ftp.openssl.org/snapshot/>.

Bytes	Timestamp	Filename
4457113	May 10 17:20:24 2012	openssl-1.0.1c.tar.gz (MD5) (SHA1) (PGP sign) [LATEST]
4047852	May 10 17:07:50 2012	openssl-1.0.0j.tar.gz (MD5) (SHA1) (PGP sign)
3782486	May 10 16:41:36 2012	openssl-0.9.8x.tar.gz (MD5) (SHA1) (PGP sign)
4456651	Apr 26 12:52:52 2012	openssl-1.0.1b.tar.gz (MD5) (SHA1) (PGP sign)
3919005	Apr 25 14:37:05 2012	openssl-fips-1.2.4.tar.gz (MD5) (SHA1) (PGP sign)
3782900	Apr 23 23:05:39 2012	openssl-0.9.8w.tar.gz (MD5) (SHA1) (PGP sign)
4456456	Apr 19 14:20:37 2012	openssl-1.0.1a.tar.gz (MD5) (SHA1) (PGP sign)
3782207	Apr 19 14:06:49 2012	openssl-0.9.8v.tar.gz (MD5) (SHA1) (PGP sign)
4047721	Apr 19 13:55:31 2012	openssl-1.0.0i.tar.gz (MD5) (SHA1) (PGP sign)
4453920	Mar 14 14:34:38 2012	openssl-1.0.1.tar.gz (MD5) (SHA1) (PGP sign)
3781776	Mar 12 16:43:06 2012	openssl-0.9.8u.tar.gz (MD5) (SHA1) (PGP sign)

Domain Name System Security Extensions

DNSSEC



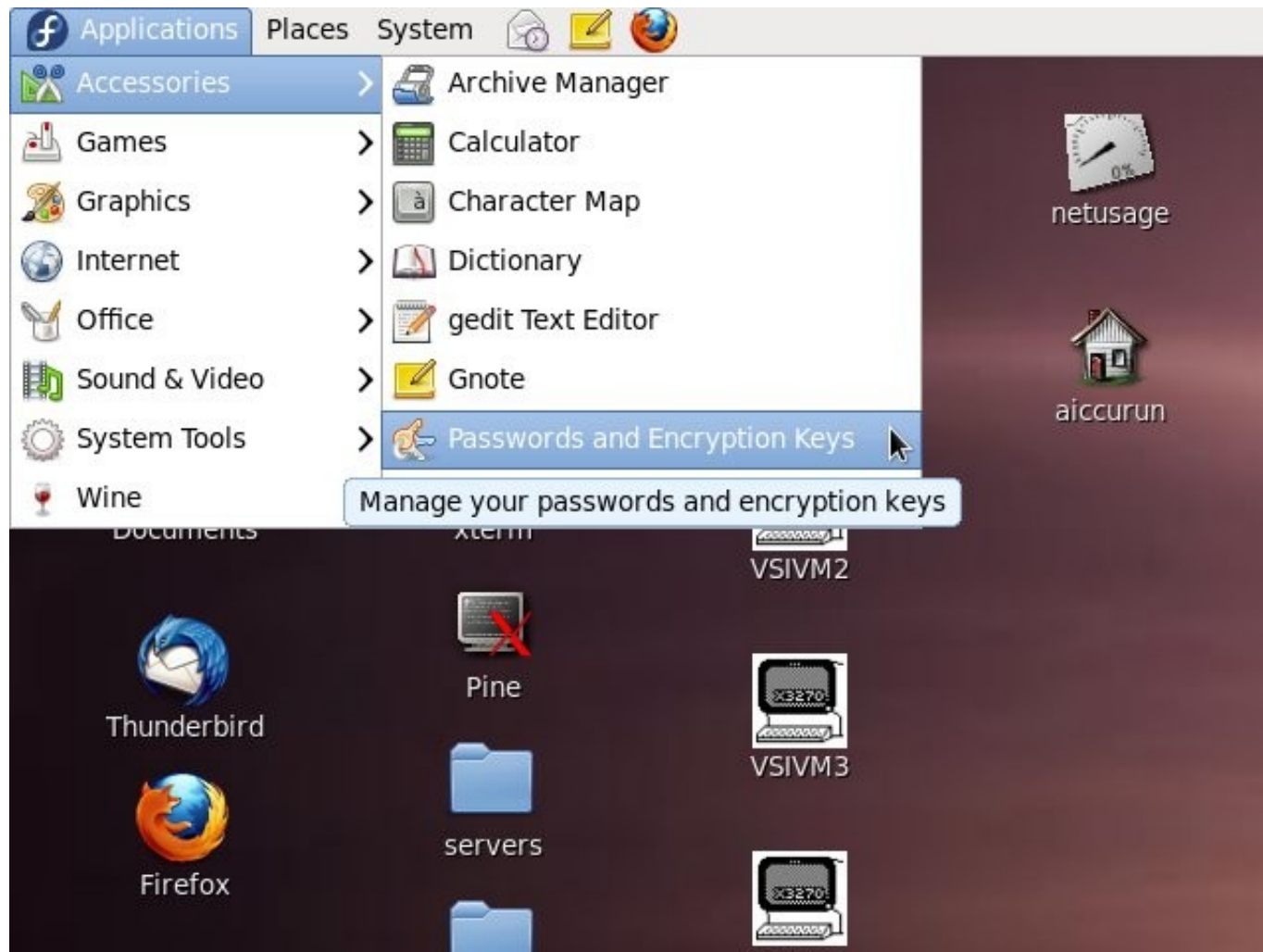
© istock photo / benoitb

Why It Matters

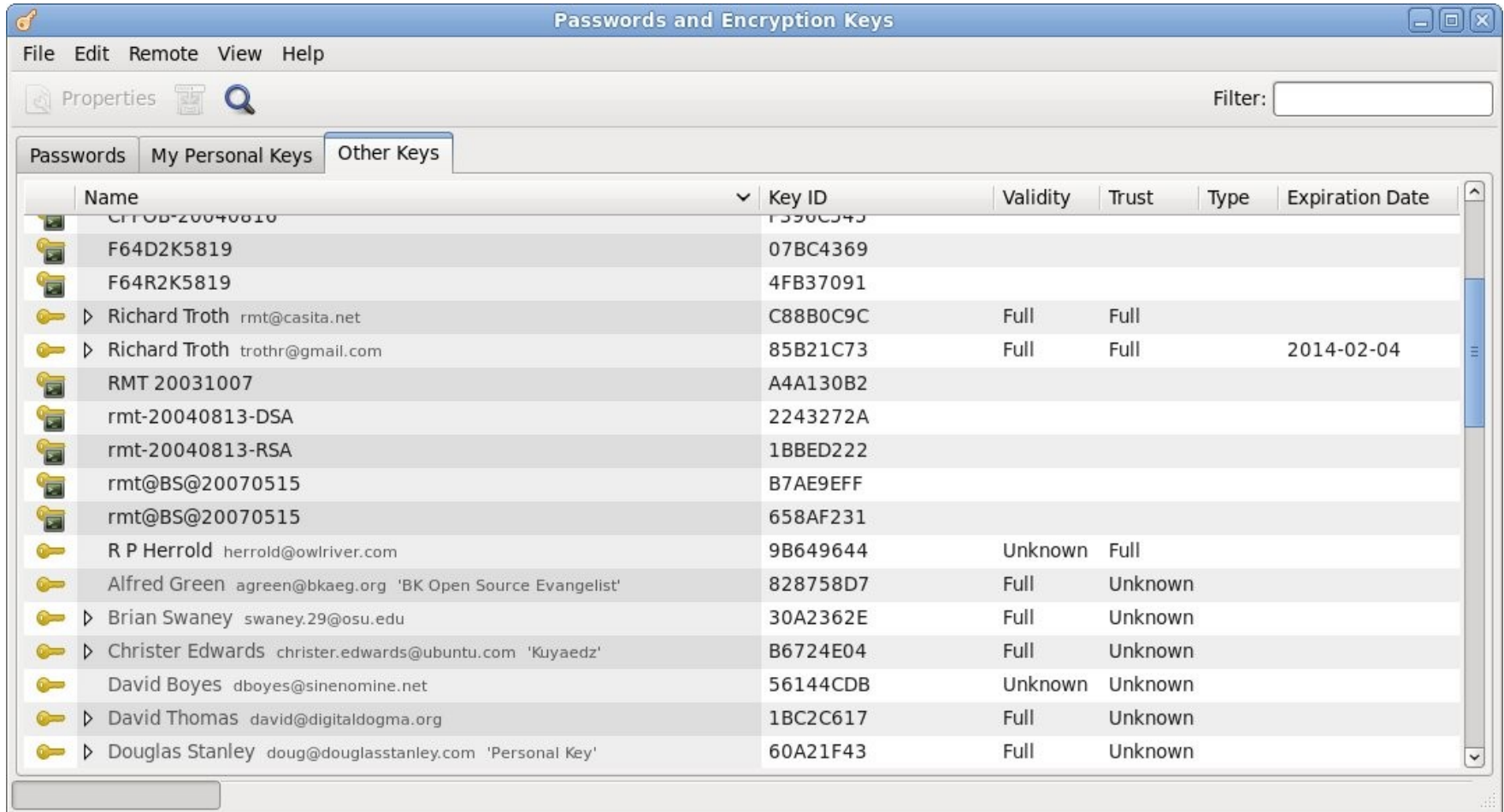
Domain Name System Security Extensions - or DNSSEC - allows users to have more confidence in the online activities that are increasingly becoming a part of our lives at work, home, and school. DNSSEC acts like tamper-proof packaging for domain name data, helping to ensure that you are communicating with the correct website or service.

Crypto Signing of Internet Domain Data

Key Management - Seahorse



Key Management - Seahorse



The screenshot shows the 'Passwords and Encryption Keys' application window. The window title is 'Passwords and Encryption Keys'. The menu bar includes 'File', 'Edit', 'Remote', 'View', and 'Help'. There is a search bar labeled 'Filter:' and a 'Properties' button. The main content area has three tabs: 'Passwords', 'My Personal Keys', and 'Other Keys'. The 'Other Keys' tab is selected, displaying a table of keys.

Name	Key ID	Validity	Trust	Type	Expiration Date
CTFOB-20040810	F590C343				
F64D2K5819	07BC4369				
F64R2K5819	4FB37091				
▷ Richard Troth rmt@casita.net	C88B0C9C	Full	Full		
▷ Richard Troth trothr@gmail.com	85B21C73	Full	Full		2014-02-04
RMT 20031007	A4A130B2				
rmt-20040813-DSA	2243272A				
rmt-20040813-RSA	1BBED222				
rmt@BS@20070515	B7AE9EFF				
rmt@BS@20070515	658AF231				
R P Herrold herrold@owriver.com	9B649644	Unknown	Full		
Alfred Green agreen@bkaeg.org 'BK Open Source Evangelist'	828758D7	Full	Unknown		
▷ Brian Swaney swaney.29@osu.edu	30A2362E	Full	Unknown		
▷ Christer Edwards christer.edwards@ubuntu.com 'Kuyaedz'	B6724E04	Full	Unknown		
David Boyes dboyes@sinenomine.net	56144CDB	Unknown	Unknown		
▷ David Thomas david@digitaldogma.org	1BC2C617	Full	Unknown		
▷ Douglas Stanley doug@douglasstanley.com 'Personal Key'	60A21F43	Full	Unknown		

Terms and Tools to Learn

Certificates identified by SDN,
“subject distinguished name”

X.509 verbiage abounds

Need overview of BFS files (for VM SSL)

- x /etc/gskadm/mycert.crq (nam bfs)

CA here is “Certificate Authority”

What is a “subject”?

What is the “subject”? (from SDN)

- That which is “signed” (issued) by an “authority”

What is the “authority”? (as in CA)

- That which cryptographically signs the “subject”

What is the “issuer”? (from IDN)

- The authority issuing a certificate

maximum entropy, minimum energy

maximum entropy, minimum “order”

Entropy ==> Randomness

Strong encryption

requires reliable randomness

Human factors remain the biggest risk

- Easy passwords
- Gullible to scams
- Easy-click assertion
- Profiled for info
- Unsecured hardware
- Lost hardware



Back Channels?



Preventing Data Loss Through Privileged Access Channels



This white paper focuses on how organizations facing the issues of privileged access can effectively balance the challenges of cost, risk and compliance. It describes how privileged access governance can be made minimally invasive, scale to enterprise requirements and most importantly, prevent costly losses.

[Download a copy today](#)

A security auditor for our servers has demanded the following within two weeks:

- A list of current usernames and plain-text passwords for all user accounts on all servers
- A list of all password changes for the past six months, again in plain-text
- A list of "every file added to the server from remote devices" in the past six months
- The public and private keys of any SSH keys
- An email sent to him every time a user changes their password, containing the plain text password

We're running Red Hat Linux 5/6 and CentOS 5 boxes with LDAP authentication.



Anonymity Network, uses “onion routing”
<https://www.torproject.org/index.html.en>

See also: TAILS

<https://tails.boum.org/index.en.html>



Tor “Hidden Services”

```
HiddenServiceDir /some/restricted/directory/  
HiddenServicePort 22 127.0.0.1:2222  
HiddenServicePort 80 192.168.5.67:80  
HiddenServicePort 608 127.0.0.1:608
```

<http://zynn8tqupxhroqmn.onion/>

Only reachable via Tor network

PGP based service

Multiple varied “proofs” of ID and ownership

<https://keybase.io>

Recommend: do not upload your private key



You need SSL

Apply SSL carefully

Understand the concepts

Be prepared:

SSL is a moving target!

And practice. Play with the stuff.