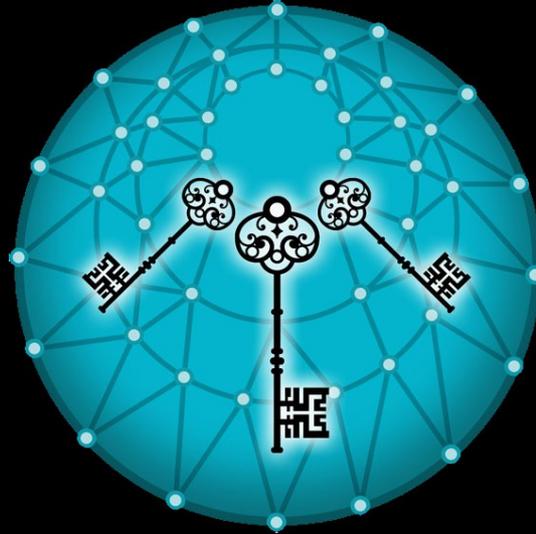


Structurally Insecure?



Several paradoxes in the market for Certificate Authorities,
and some ideas for resolving them

Peter Eckersley
pde@eff.org

TLS/SSL is really valuable

Millions of services use it

It is a necessary condition of most secure uses
of the Internet

Encryption is only as good as your ability to verify the
other party

Currently, most uses of TLS
(web, email, IM, many VPNs)
rely on PKIX

What is PKIX? Where did it come from?

Invented by Netscape, early 90s

In a hurry

Specifically to protect credit card #s

Before strong crypto was legal

Based on X.509, an ITU standard from the 1980s
(which predates the Web!)

Security via digital paperwork



X.509 certs can (and do) contain just about anything

```

#!/usr/bin/env python

# diversity.py -- estimate the number of different certificate types and
# combinations of fields in them

from dbconnect import dbconnect
db,dbc = dbconnect()
q = """
SELECT *,`X509v3 extensions:X509v3 Key Usage`,
        `X509v3 extensions:X509v3 Extended Key Usage`,
        `X509v3 extensions:X509v3 Basic Constraints:CA`,
        `X509v3 extensions:Netscape Cert Type`
FROM all_certs
WHERE certid >= %d and certid < %d
"""

dbc.execute("SELECT count(certid) from all_certs")
n = int(dbc.fetchone()[0])
print n, "rows"

fset = {}
for i in range(n / 1024):
    q1 = q % (i* 1024, (i+1) * 1024)
    dbc.execute(q1)
    batch= dbc.fetchall()
    for row in batch:
        cert, type_fields = row[:-4], row[-4:]
        bits = 0
        for field in cert:
            if field==None:
                bits |= 0x01
            elif type(field) == str and ("critical" in field):
                bits |=0x02
            bits <<= 2
        key = (type_fields, bits)
        fset[bits]=True

print len(fset)

```

By this approximate measure:

10,320 *kinds* of X.509 certs were observed

1,352 kinds were sometimes valid

Not as bad as a million kinds,
still *hard to process automatically*

How were the X.509 certificates to be issued?

Via a market!



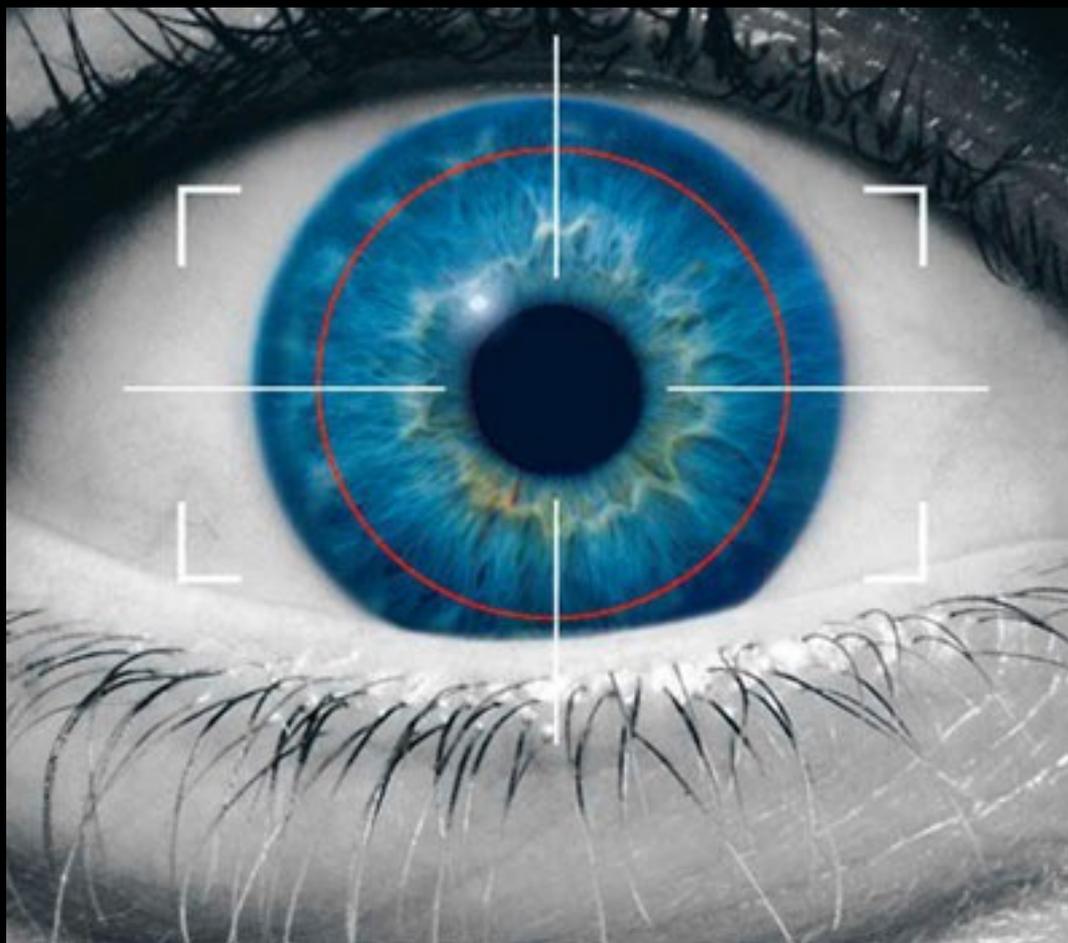
Markets are a great idea!

CAs can compete with each other to offer the best services at the best prices

Competition will drive down prices,

and

there will be different levels of certification services



Expensive, thorough validation



Cheaper, weaker validation

Actually that isn't how the CA market works at all!

How the CA marketplace works



Regardless of what you pay for, you get every restaurant's food mixed together



But PKIX is holding us back

Thesis: this is a product of the PKIX market structure

4 problems

Problem 1

TLS/SSL Authentication



SSL Observatory





Private keys safely locked in an HSM?

Domain validation

Alice asks for cert

CA emails root@domain.com

Someone clicks on link in email

CA gives cert to Alice

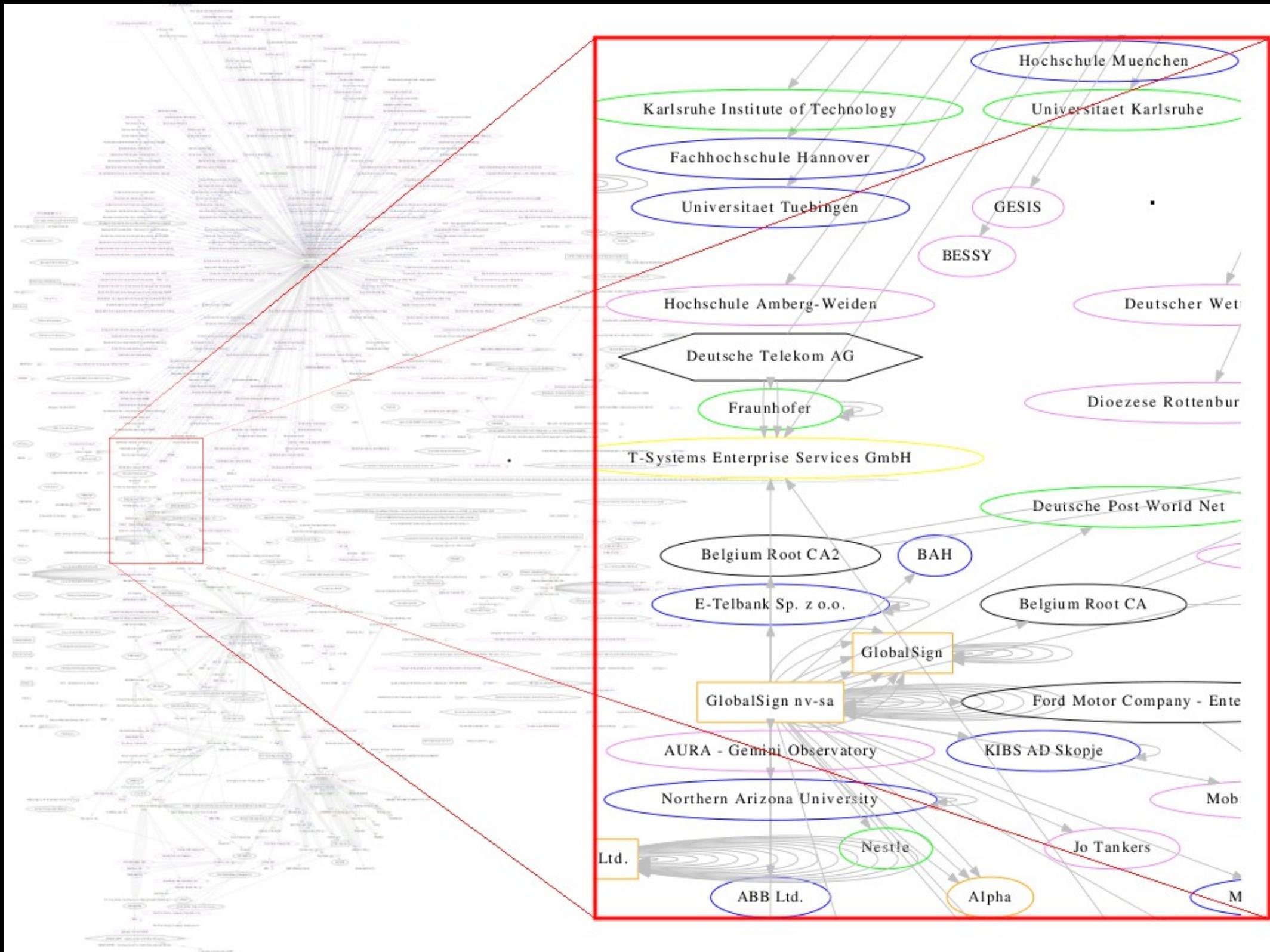
Attack surface:

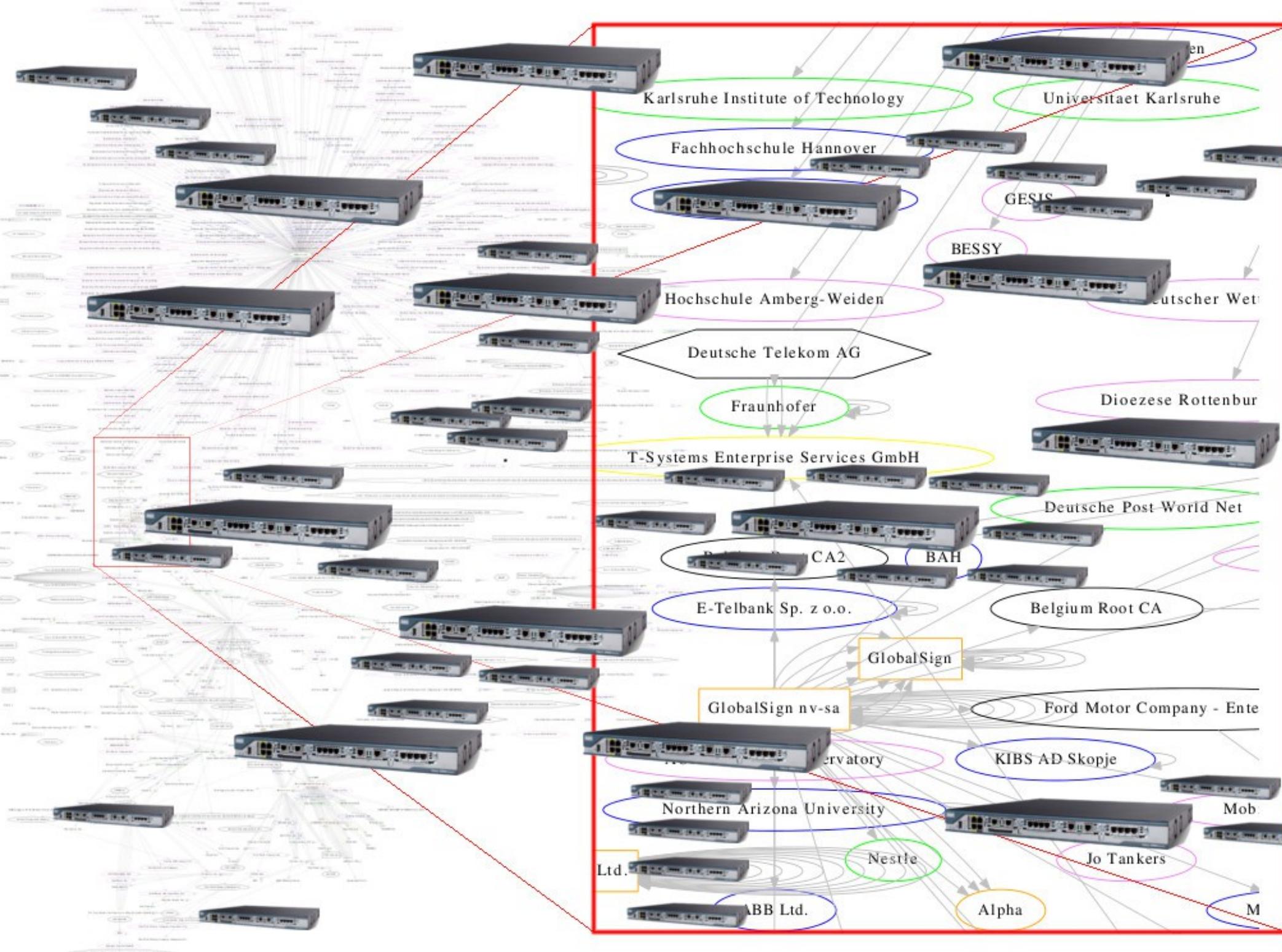
The CAs

Their routers

Their ISPs' routers...

All of the DNS infrastructure







Problem 2

The screenshot shows a web browser window with a single tab titled "Certificate Error: Navigation...". The address bar displays the URL "https://localhost/Test/test.txt". The main content area features a red shield icon with a white 'X' on the left. The text reads: "There is a problem with this website's security certificate." followed by "The security certificate presented by this website was not issued by a trusted certificate authority." and "Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server." Below this, a bold recommendation states: "We recommend that you close this webpage and do not continue to this website." Three options are listed: "Click here to close this webpage." (with a green checkmark icon), "Continue to this website (not recommended)." (with a red 'X' icon), and "More information" (with a blue downward arrow icon). The browser's status bar at the bottom right shows a magnifying glass icon and "100%".

https://localhost/Test/test.txt Certificate Error: Navigation... x

There is a problem with this website's security certificate.

The security certificate presented by this website was not issued by a trusted certificate authority.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

- [Click here to close this webpage.](#)
- [Continue to this website \(not recommended\).](#)
- [More information](#)

100%



Verify Certificate



Fetch can not verify the identity of server "ftps-masquerade.fetchsoftworks.com".

An FTP connection to "ftps-masquerade.fetchsoftworks.com" could not be opened because the hostname of the server's SSL certificate does not match the hostname used to connect to the server.

Click Show Certificate for more information and an opportunity to have this certificate always accepted. Click Continue to proceed, or Cancel to disconnect.



Show Certificate

Cancel

Continue



99% of the time, this is not an attack!



Certificate warnings are useless for humans!

Problem 3

Surveillance and censorship by governments

X.509 PKI was not designed
to survive this!

Examples

Problem 1 & 3



Problems 2 & 3

Syria attacks Facebook

Could not verify this certificate because the issuer is not trusted.

Issued To		Issued To	
Common Name (CN)	s.static.ak.facebook.com	Common Name (CN)	www.facebook.com
Organization (O)	Facebook, Inc.	Organization (O)	Facebook, Inc.
Organizational Unit (OU)	Facebook	Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	00:C6:4F:50:11:B3:65:DC:B9	Serial Number	0C:6F:C8:59:57:FA:1F:5F:C9:67:2C:9F:E6:5C:DB:E6
Issued By		Issued By	
Common Name (CN)	s.static.ak.facebook.com	Common Name (CN)	DigiCert High Assurance CA-3
Organization (O)	Facebook, Inc.	Organization (O)	DigiCert Inc
Organizational Unit (OU)	Facebook	Organizational Unit (OU)	www.digicert.com
Validity		Validity	
Issued On	01/05/2011	Issued On	15/11/2010
Expires On	30/04/2012	Expires On	03/12/2013
Fingerprints		Fingerprints	
SHA1 Fingerprint	DD:F4:94:F4:A7:93:52:DD:83:BC:8E:16:58:DE:0C:BC:B8:4F:70:41	SHA1 Fingerprint	63:08:84:E2:79:CB:11:07:F1:FB:8A:6B:11:A6:4D:1B:14:76:3F:8E
MD5 Fingerprint	5D:F0:36:75:2A:65:16:4B:3D:79:88:A8:8F:26:40:51	MD5 Fingerprint	42:03:FC:BE:01:3E:2D:6C:57:48:E8:49:79:B4:8E:26

Some cases of problem 3 to consider

Internet censorship in...

Australia (?!), Bahrain, Burma, China, Cuba, Iran,
Syria, Thailand, Turkey, UAE, USA (?!)

...and dozens of other countries

Kazhakstan, June 2011:

demand for access to all Kazakh Google users' data

(Google briefly unplugged google.kz)

Note:

The Kazhakh government can control
<https://google.kz> if it wishes

Google cannot stop users from typing
“google.kz” into a browser

The lesson of vb.ly

vb.ly

the internet's first and only sex-positive url shortener

VB'ize your url:

http://

Shorten

optional custom keyword:

feedback

[contact](#) | [did we mention we have an API?](#)

powered by the marvelous yourls. hosted by the amazing laughing squid. © 2009 the internet's violet blue ®. use at your own risk. no animals are harmed during the shortening of your linkage.

The lesson of vb.ly?



Who else uses .ly?

bit.ly – 2 billion URLs / month

bit.ly – we were lucky!



The country ↔ TLD mapping was a cute idea

That ship has sailed

Problem 4

HTTP needs to be retired

That won't happen with PKIX

Solutions to these problems?

For Problem 1...

	<u>DANE</u> Select table column	Pinning	Perspectives/ Convergence	Sovereign Keys	Cert Transparency	<u>Mecai</u>	<u>Decentralised Observatory</u>
Protects against attacks owning the server's upstream router	Y	Y	N	Y	Y	N	Y
Protects against victim's DNS zone / <u>gov'ts</u> with leverage	N	Y	N	Y	not really	N	Y
Copes with messy/partial cert rollover	?	Y	N	Y	Y	with work	Y
HTTPS server <u>admins</u> don't need to do anything	N	N	Y	N	N	N	sort of
Works for all SSL/TLS protocols (not just HTTPS)	Y	N	?	Y	Y	Y	N
Protects 1st connection	Y	If preloaded	Y	Y	undefined	Y	Y
Protection, once in place, is automatic and robust	Y	Y	N	Y	Y	Y	N
Avoids cert warnings	N	N	some	Y	N	N	N

Also: NameCoin, other projects?

But what about problem 4?