Perspectives:

Can Host Authentication be Secure AND Cheap?

Demo + Software : http://www.cs.cmu.edu/~perspectives/

Dan Wendlandt - danwent@gmail.com
Carnegie Mellon University

Joint work with:
David G. Andersen and Adrian Perrig

Why should you care?

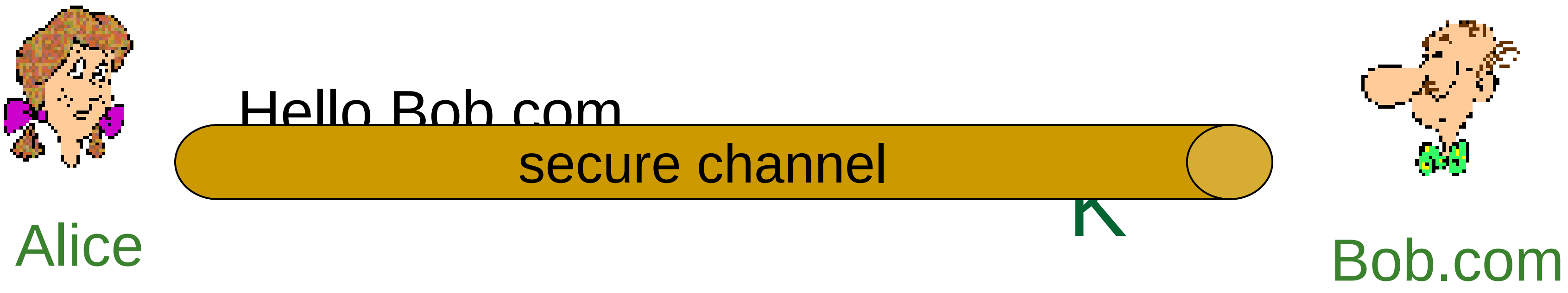
- Using a traditional host PKI can be costly in $$ and admin time.

- Perspectives used automated network probing to create a “lightweight PKI”:
  - Makes SSH/self-signed HTTPS more secure + useable.
  - Potential to offer cheap alternative to existing PKI solutions.

- What I’m looking for:
  - Your feedback / flames.
  - If interested, your participation.

“Man in the Middle” (MitM) Attacks

Alice needs Bob.com’s public key to establish a secure channel (e.g., SSL/SSH) to him.



Hello Bob.com

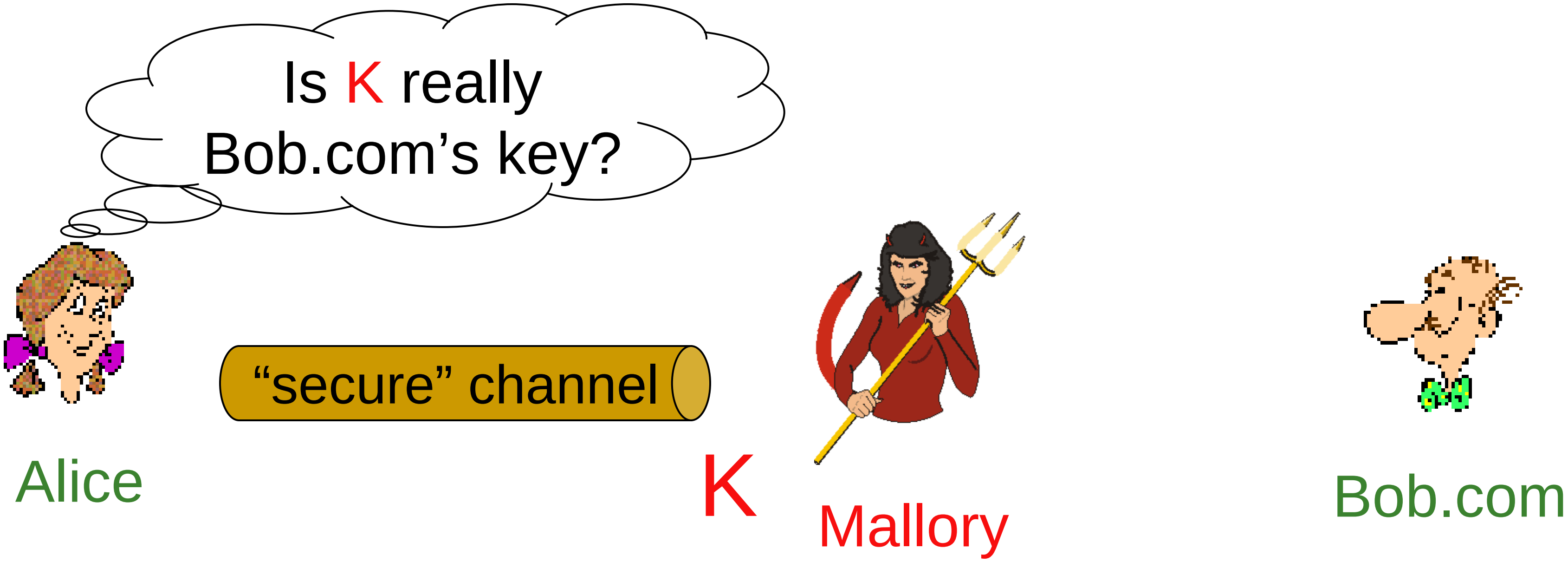
secure channel

K

Alice

Bob.com

“Man in the Middle” (MitM) Attacks



Is K really Bob.com’s key?

“secure” channel

Alice

K

Mallory

Bob.com

If Alice accepts K, Mallory can snoop and modify all traffic!

Do MitM Attacks Really Matter?

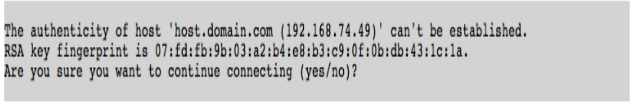
- Recent trends increase MitM vulnerability
  - Other hosts on a wifi LAN can spoof ARP/DNS.
    e.g., ARPIFrame worm
  - Known vulnerabilities in home routers/APs.
    e.g., “Pharming” attacks
  - Recent “Kaminsky” DNS attack vector.

- Attacks are often automated & profit driven

Authenticating Public Keys

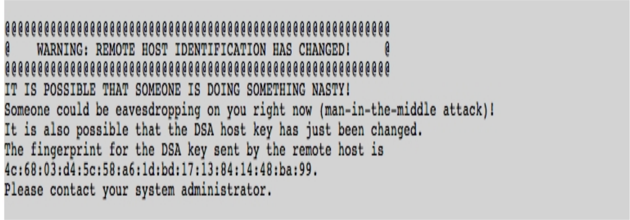
Two standard approaches to handling MitM attacks:

- Public Key Infrastructure (e.g., Verisign certs)
- Prayer (e.g., SSH and self-signed HTTPS)

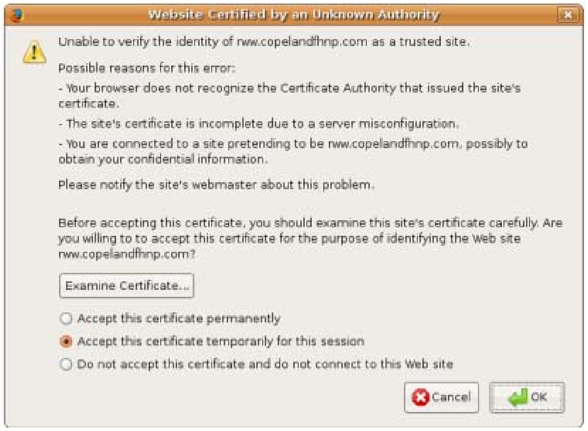


```
The authenticity of host 'host.domain.com (192.168.74.49)' can't be established.
RSA key fingerprint is 07:fd:fb:9b:03:a2:b4:e8:b3:c9:0f:0b:db:43:1c:1a.
Are you sure you want to continue connecting (yes/no)?
```

or



```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the DSA host key has just been changed.
The fingerprint for the DSA key sent by the remote host is
4c:68:03:d4:5c:58:a6:1d:bd:17:13:84:14:48:ba:99.
Please contact your system administrator.
```



Website Certified by an Unknown Authority

Unable to verify the identity of rww.copelandfhnp.com as a trusted site.

Possible reasons for this error:

- Your browser does not recognize the Certificate Authority that issued the site's certificate.

- The site's certificate is incomplete due to a server misconfiguration.

- You are connected to a site pretending to be rww.copelandfhnp.com, possibly to obtain your confidential information.

Please notify the site's webmaster about this problem.

Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to to accept this certificate for the purpose of identifying the Web site rww.copelandfhnp.com?

○ Accept this certificate permanently
● Accept this certificate temporarily for this session
○ Do not accept this certificate and do not connect to this Web site

download code at:
http://www.cs.cmu.edu/~perspectives/

Prayer (aka SSH-style Authentication)

Definition of SSH-style Authentication:

1) **Pray for no adversary on first connection, cache key.**
2) **If key changes on a subsequent connection, panic!**
3) **If you feel lucky, pray again and connect anyway.**

Why would anyone use prayer?

Unlike a PKI, it is <u>cheap</u> and <u>simple to use.</u>

A <u>secure</u> PKI traditionally requires:

- Costly (often manual) verification by a Certificate Authority
- Admin time to submit, install and replace certificates on each server.

SSH-style auth requires neither cost. It is “Plug-and-Play”

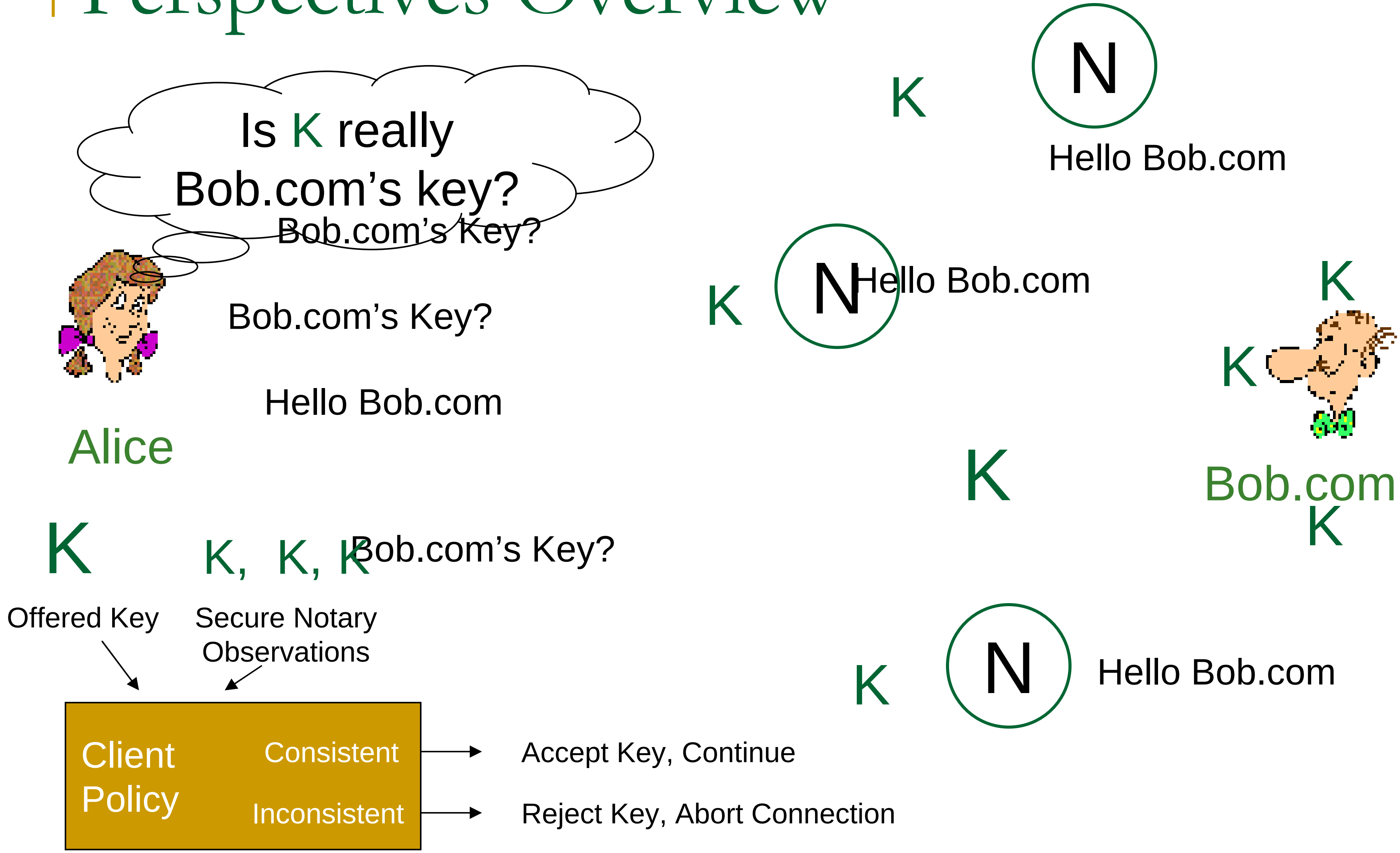
➔ SSH quickly + ubiquitously SSH replaced telnet.

Our Approach: Strengthen the SSH Model

We design “**Perspectives**” to:

- Keep SSH-style “Plug-n-Play” simplicity + low-cost.
- Significantly improve attack resistance

Perspectives Overview



Is K really Bob.com's key?

Bob.com's Key?

Bob.com's Key?

Hello Bob.com

Alice

K

N

Hello Bob.com

K

N

Hello Bob.com

K

K

Bob.com

K

K

K

N

Hello Bob.com

K

K, K, K

Bob.com's Key?

Offered Key

Secure Notary Observations

Client Policy

Consistent → Accept Key, Continue

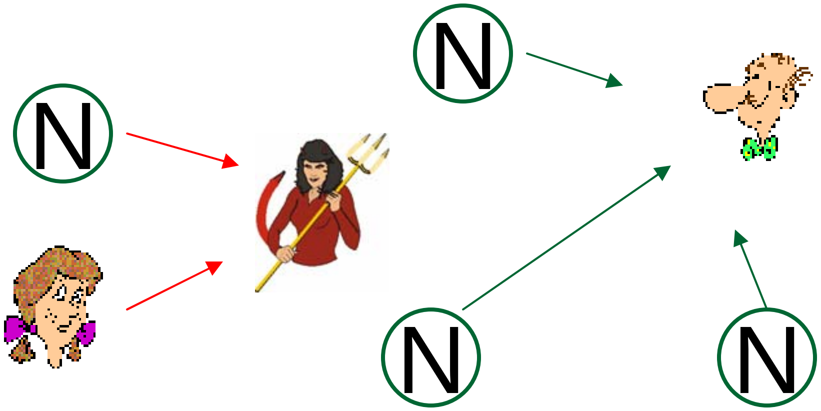
Inconsistent → Reject Key, Abort Connection

download code at:
http://www.cs.cmu.edu/~perspectives/

Perspectives: Attack Resistance Model

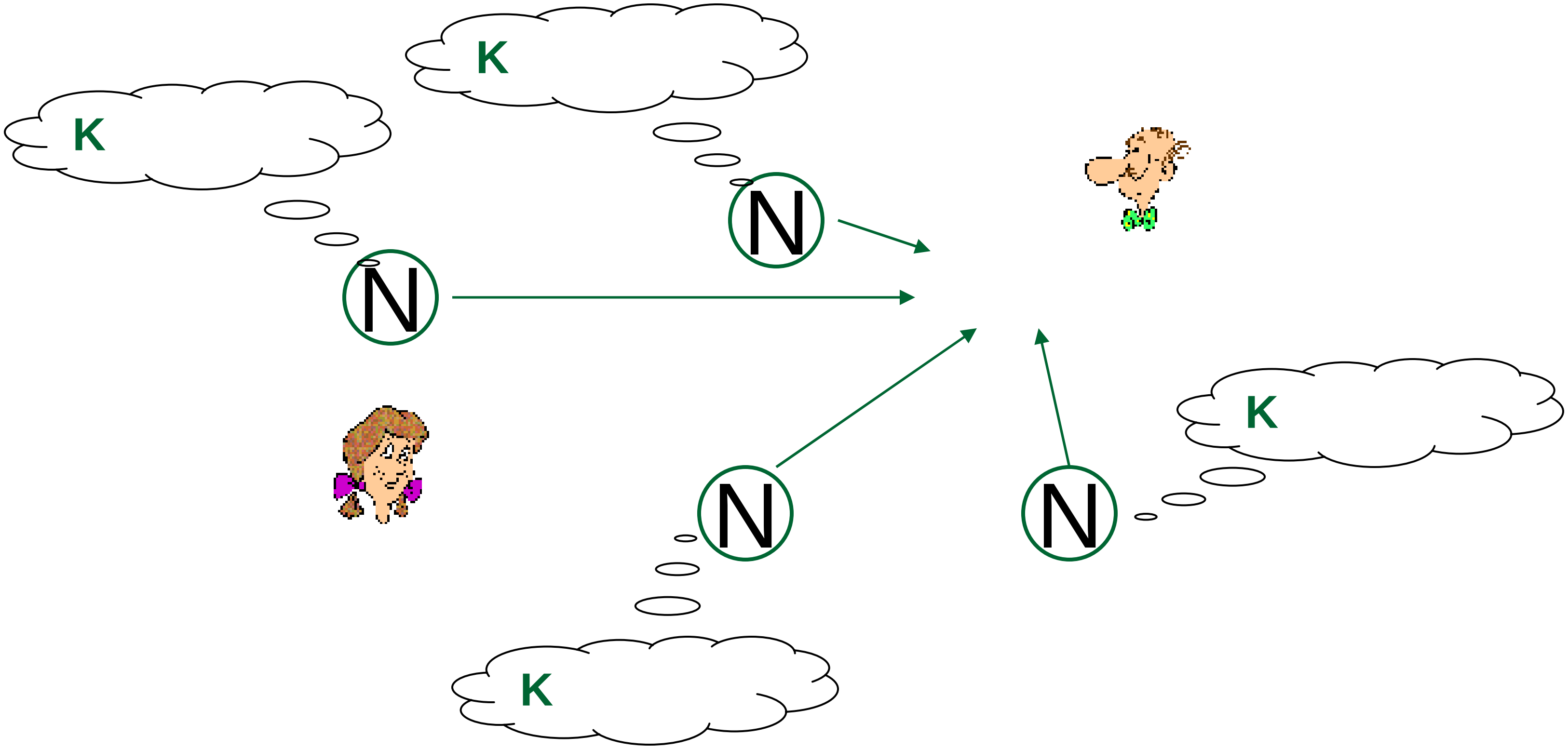
Spatial Resistance:

Multiple vantage points to circumvent localized attackers



Perspectives: Attack Resistance Model

Temporal Resistance:
Key history raises alarm even if all paths are compromised.

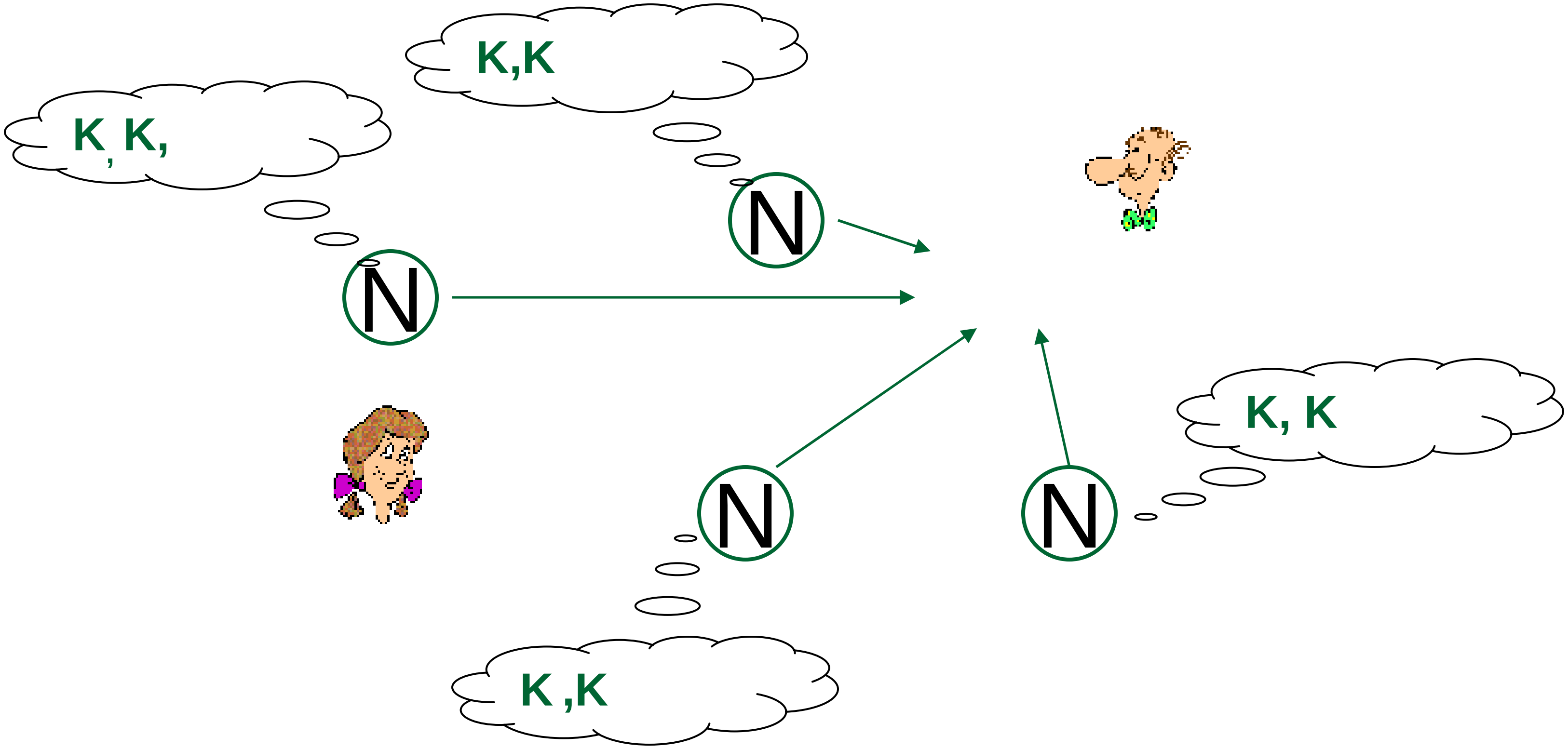


download code at:
http://www.cs.cmu.edu/~perspectives/

Perspectives: Attack Resistance Model

Temporal Resistance:

Key history raises alarm even if all paths are compromised.

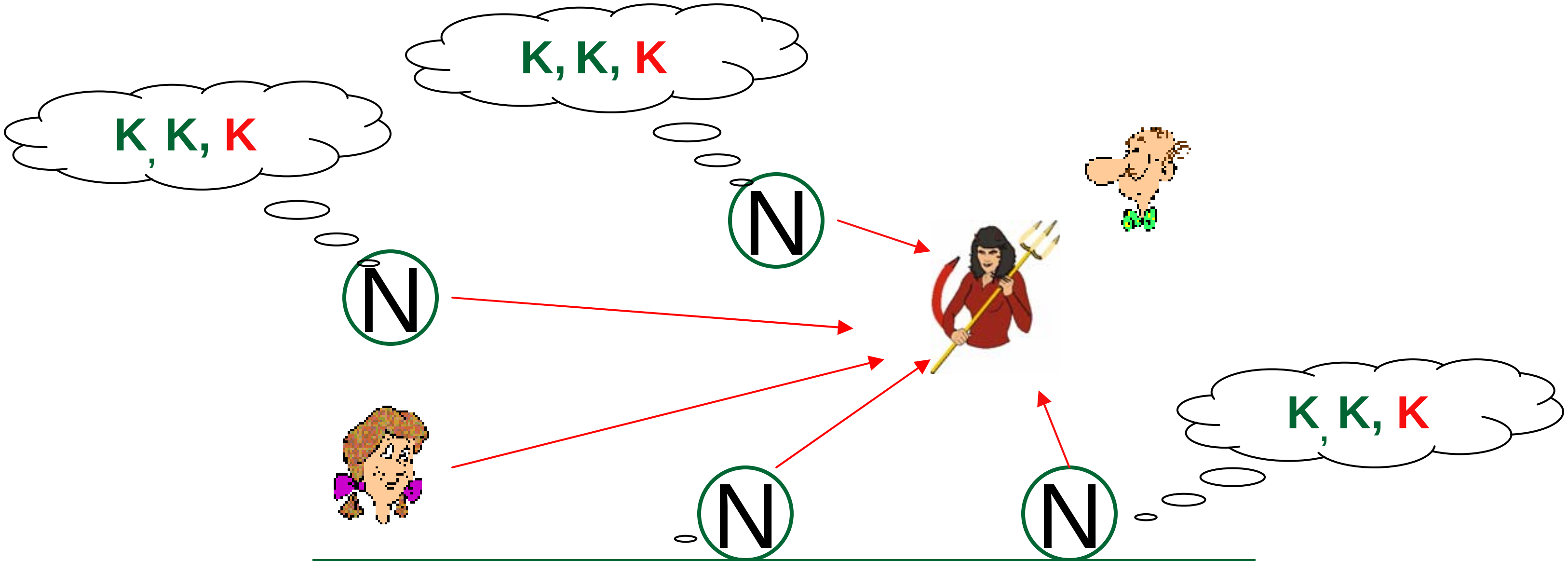


download code at:
http://www.cs.cmu.edu/~perspectives/

Perspectives: Attack Resistance Model

Temporal Resistance:

Key history raises alarm even if all paths are compromised.



Not bullet-proof, but significantly improves attack resistance.

download code at:
http://www.cs.cmu.edu/~perspectives/

Perspectives Design

- Who runs these network notaries?
- How do notaries monitor keys/certificates?
- How do clients securely retrieve notary data and decide to accept or reject a key?

Who runs “network notary” servers?

- Could be single player (e.g., Mozilla, Google, or EFF)
- Or a “community deployment” with ISPs, universities, webhosts, etc. volunteering single nodes. Similar to:
  - Public traceroute & looking-glass servers
  - Academic network testbeds like PlanetLab and RON.
- Our design + security analysis assumes that some notaries may be malicious/compromised at any time.

Who runs “network notary” servers?

- Currently targeting 10-30 global notary servers.
- “master” public key shipped with client software.
- Clients regularly fetch & verify a “notary list”:

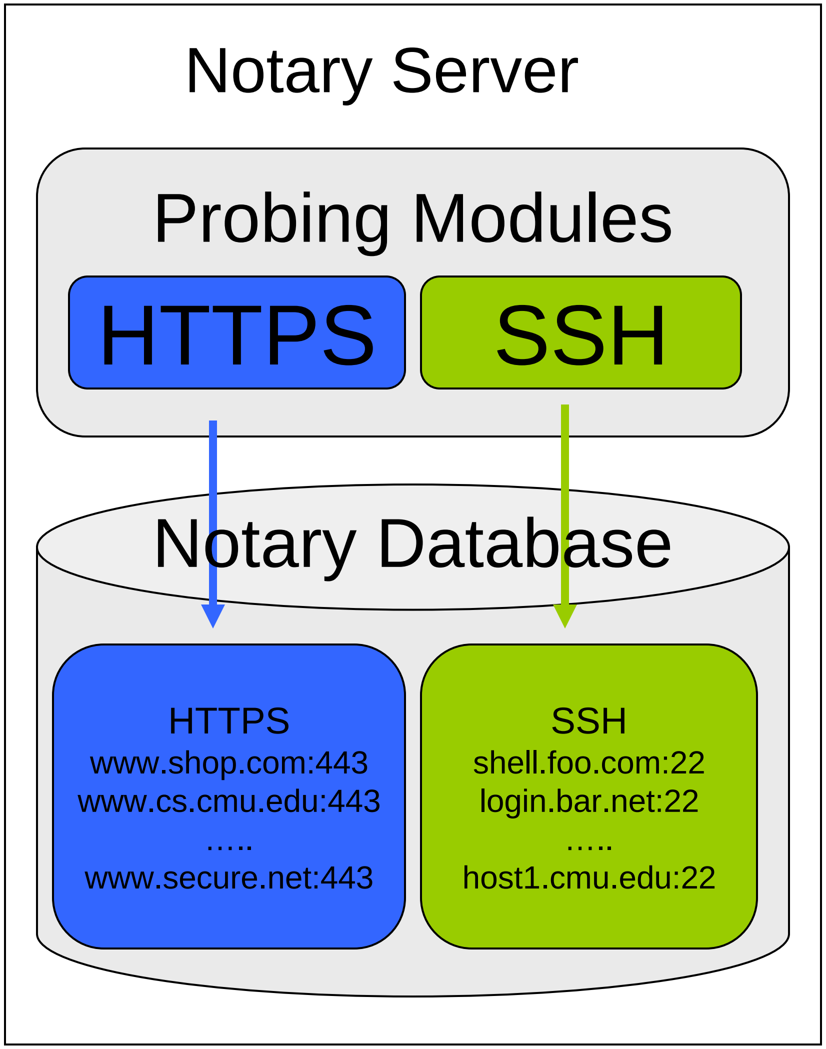
  [notary ip, notary public key]

  [notary ip, notary public key]

  ……

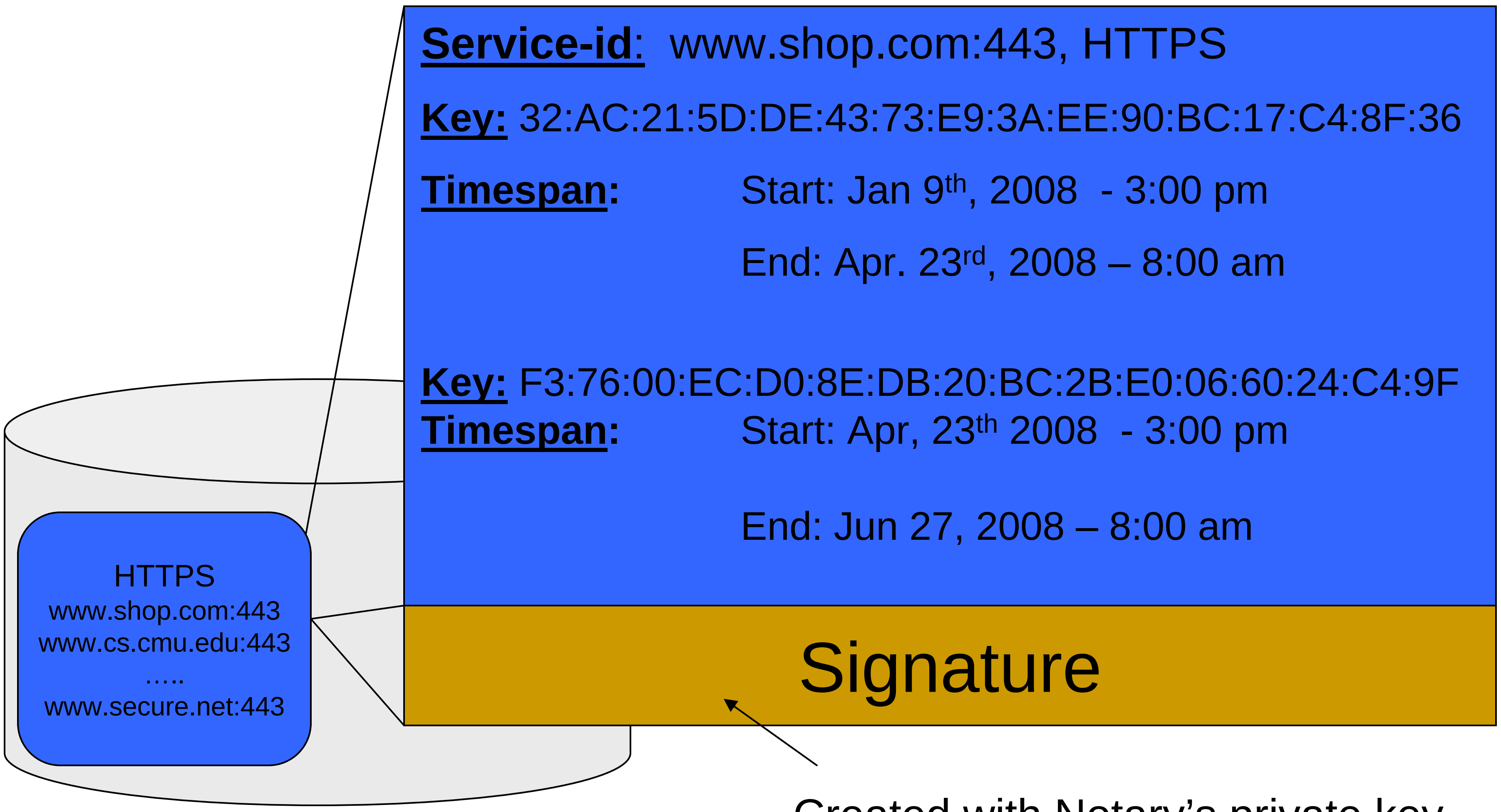
  [notary ip, notary public key]

How do notaries monitor keys?



- Protocol-specific probing modules mimic client behavior.
- Notary regularly (e.g. daily) probes each service listed in database and updates its info.

Notary Database Records



HTTPS
www.shop.com:443
www.cs.cmu.edu:443
…..
www.secure.net:443

Service-id: www.shop.com:443, HTTPS

Key: 32:AC:21:5D:DE:43:73:E9:3A:EE:90:BC:17:C4:8F:36

Timespan: Start: Jan 9th, 2008 - 3:00 pm

End: Apr. 23rd, 2008 – 8:00 am

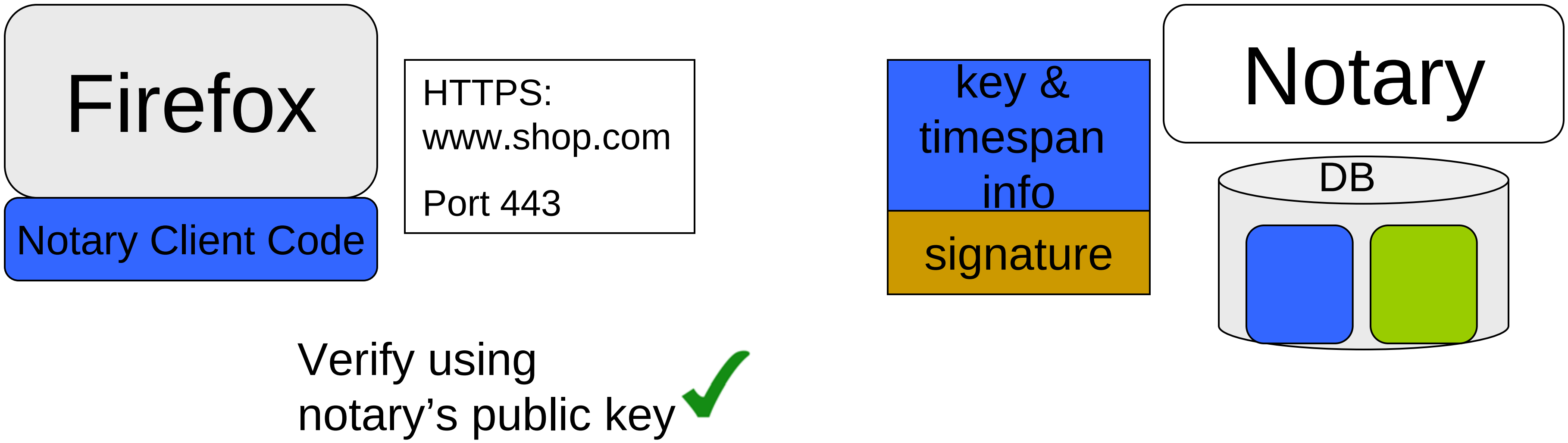
Key: F3:76:00:EC:D0:8E:DB:20:BC:2B:E0:06:60:24:C4:9F
Timespan: Start: Apr, 23th 2008 - 3:00 pm

End: Jun 27, 2008 – 8:00 am

Signature

Created with Notary’s private key

How do clients receive notary data?



Firefox

Notary Client Code

HTTPS:
www.shop.com

Port 443

key & timespan info

signature

Notary

DB

Verify using
notary's public key ✓

- Query & Response are UDP datagrams, like DNS.
- Attacker cannot "spoof" notary reply.

Client Policies to accept/reject a key.

- Test spatial and temporal "consistency".

- Many possible approaches to policies:
  - Manual (power users)

    or
  - Automatic (normal users)

Manual Key Policies: Power Users

Give sophisticated users <u>more detailed info</u>:

- 6/6 notaries have consistently seen the offered key from this service over the past 200 days.
- 4/6 notaries currently see a different key!
- All notaries have seen the offered key for the past 8 hours, but previously all consistently saw key Y!

Power user would determine if offered key passes a “consistency threshold”.

Automated Key Policies: Normal Users

Automated "Consistency Thresholds" can be tailored to the individual client's high-level security needs:

I really want to connect, just make sure I'm protected against simple (e.g., wifi) attacks.

ng is fishy, be and don't onnect.

y

100% of No have seen offered key consistently for the past 3 days

At least 50% of Notaries currently see offered key.

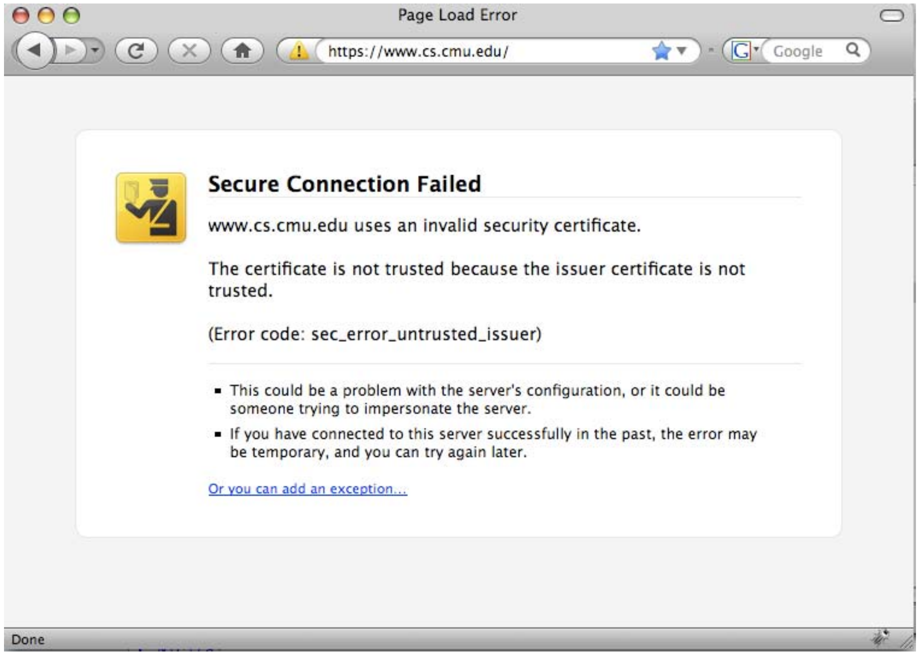
Our paper provides a detailed description and security analysis.

The Story so Far…

- Traditional PKI model is costly and cumbersome.
- Perspectives retains the low-cost and simplicity of SSH-style authentication while greatly improving attack resistance.
- Not bullet-proof, but provides a security trade-off suitable for many non-critical websites.

Three Potential uses of Perspectives

#1: Strengthen existing use of SSH and self-signed SSL



Secure Connection Failed

www.cs.cmu.edu uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is not trusted.

(Error code: sec_error_untrusted_issuer)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

Or you can add an exception…

- Recent changes to IE and Firefox make self-signed certs harder to use.
- More than 10K people have downloaded and used our Firefox extension.

download code at:
http://www.cs.cmu.edu/~perspectives/

#2: Alternative for “low-end” CA-signed certs.

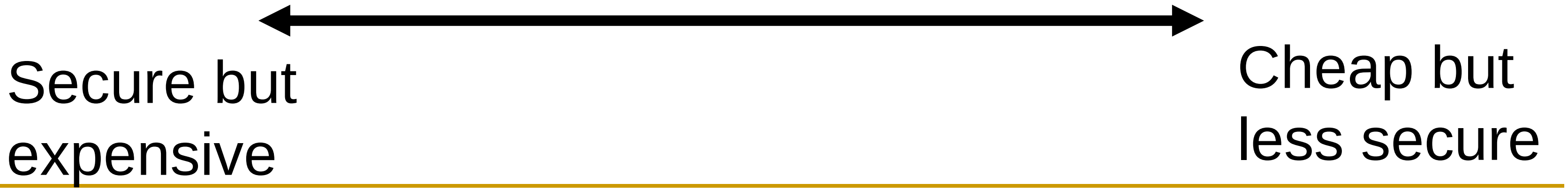
The HTTPS certificate market is splitting:

High-end certificates granted after manual verification of real-world identity.

(e.g., Extended Validation)

Low-end certificates granted after automated email to WHOIS address.

(e.g., Godaddy.com)



Secure but expensive

Cheap but less secure

#2: Alternative for “low-end” CA-signed certs.

Compared to current “low-end”, Perspectives:

- Offers comparable security:
  - A widespread attacker can likely spoof “verification” emails.
  - This spoofing attack need not be long-lasting.
- Is more convenient for server admins:
  - No need to manually request/install a cert.
  - Plays nicely with virtual hosting on a shared IP address.
- Is based on freely available data:
  - Server owners do not pay yearly “certificate tax”.
  - Clients can make an individualized security trade-off.

#3: Provide an additional layer of security for root-signed SSL certificates

- If an attacker can trick or compromise any one of the 30+ CAs, it can potentially spoof any website.

- A client can detect that the attacker’s cert differs from the cert being seen by Notaries.

- Also, website owners/third parties can monitor notary data to proactively detect attacks.

Publicly Available Notary Deployment

- Currently running on the RON testbed.
- Probes new services “on-demand”, adds them to DB.

Existing Notary Clients:

- OpenSSH: “power user” policy if key is not cached.
- Firefox 3: Automatically overrides security error page if notary data validates key.
- Query via Web: If you can’t install software on the client.

download code at:
http://www.cs.cmu.edu/~perspectives/

Notary Server Benchmarks

	Probes / day	Queries / Sec
Modern Server: 4-core 2GHz, 8 GB RAM	16.8 million	25,000
3 year-old Workstation: 1-core 2.4GHz, 512MB RAM	2.2 million	21,000

Good News:

- Current probing code is highly UNoptimized.
- Operations are “trivially parallel” => easily scales with addition machines/cores.

download code at:
http://www.cs.cmu.edu/~perspectives/

Thanks!

Source and binaries available at:

http://www.cs.cmu.edu/~perspectives/

Interested in helping? danwent@gmail.com

Academic Paper:

http://www.cs.cmu.edu/perspectives_usenix08.pdf