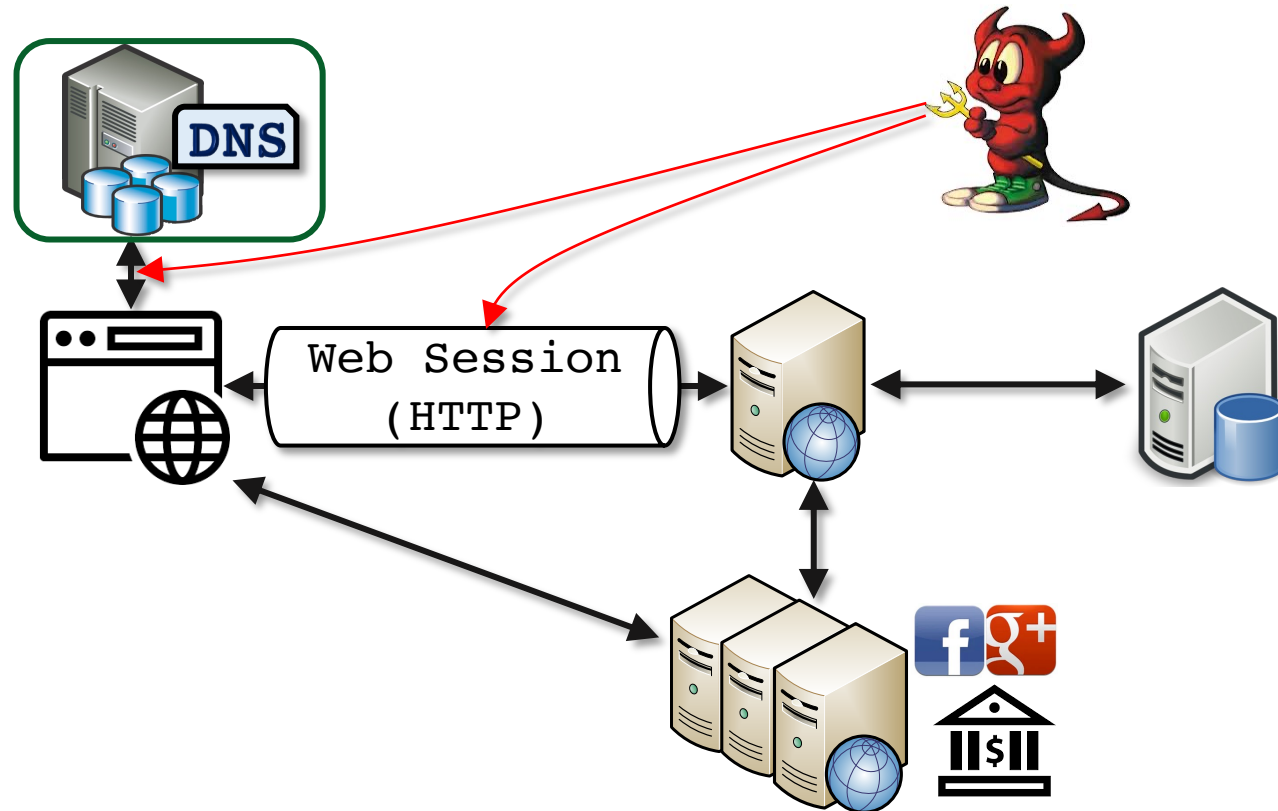


CS5331: Web Security

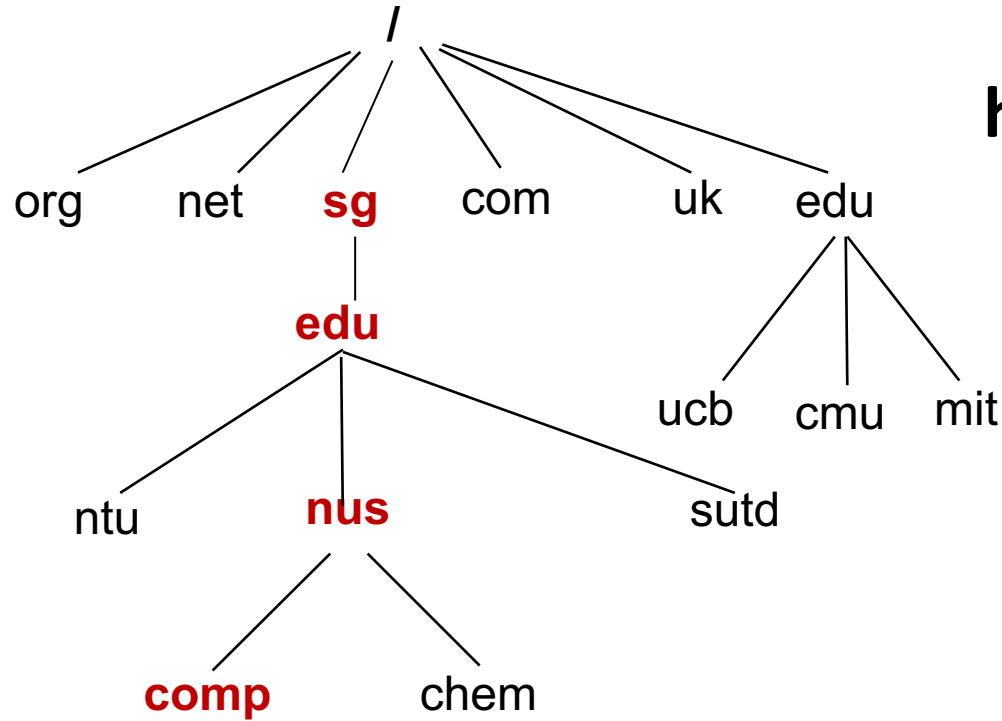
Lecture 2: Network-level Attacks

Network Threats to Web



Network Attacks: DNS (Optional Material)

Domain Names to IP addresses: DNS

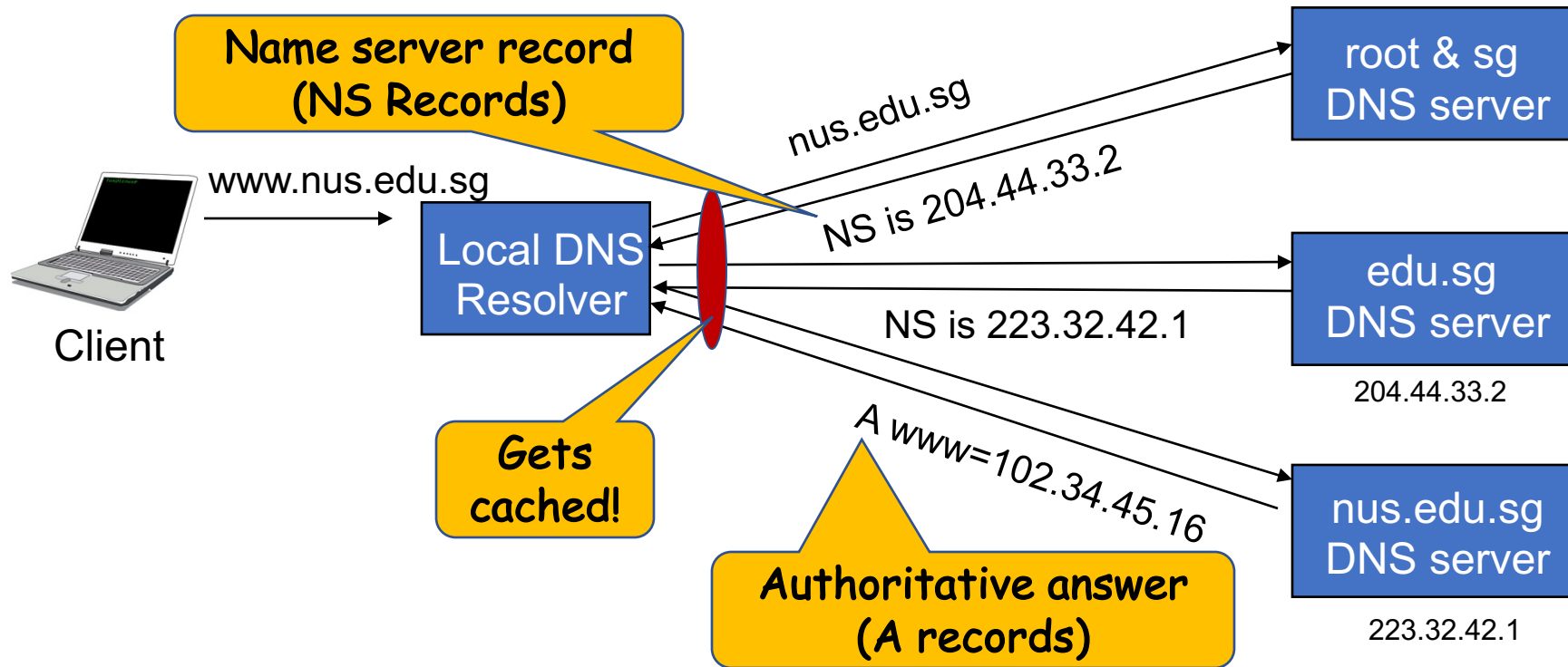


http://comp.nus.edu.sg



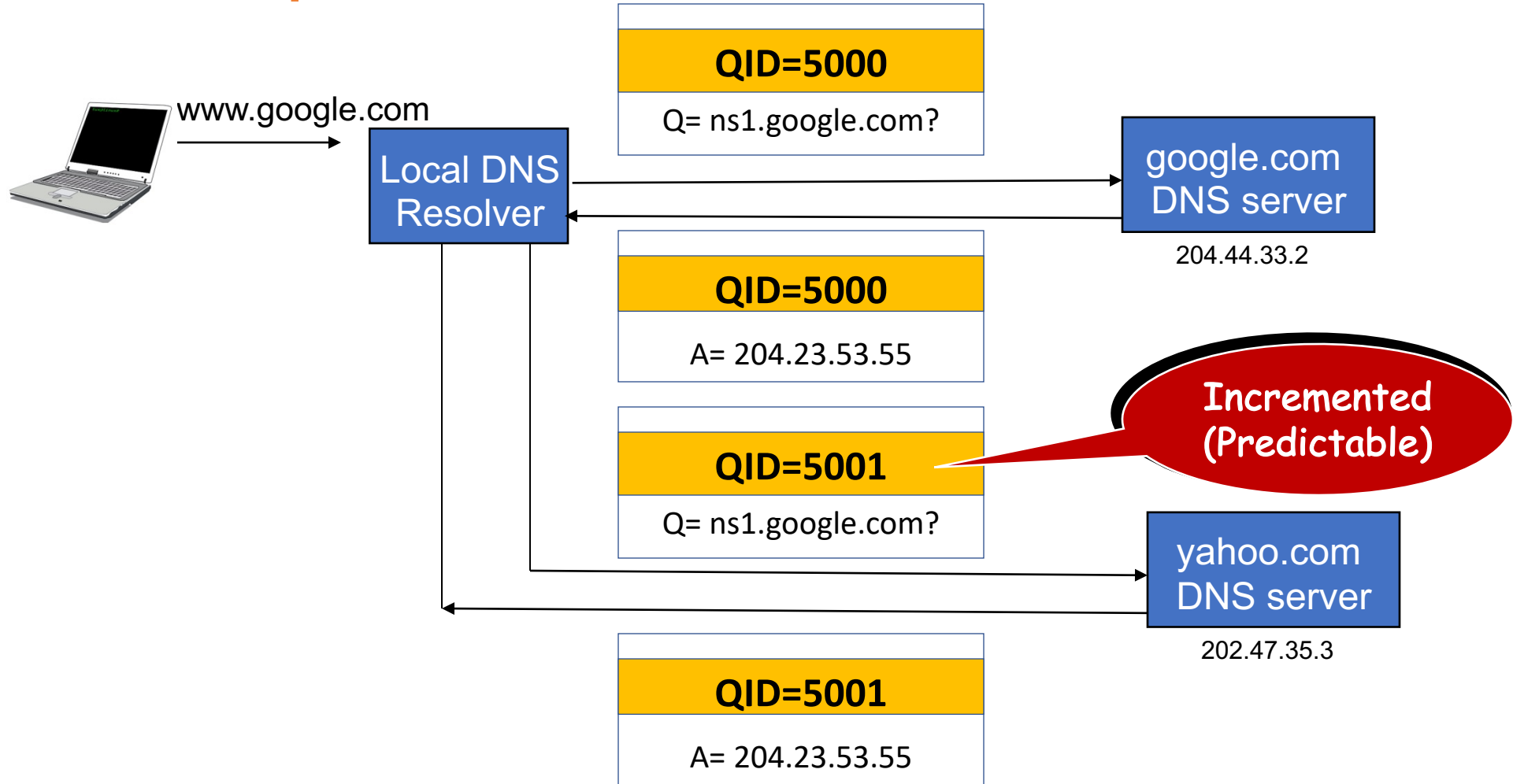
205.135.94.60

How DNS Works!



- Classical attacks
 - Modify in-transit DNS responses
 - Find software vulnerabilities in DNS software
- A more advanced (and easy) attack!
- Let's see how you can own *.google.com !

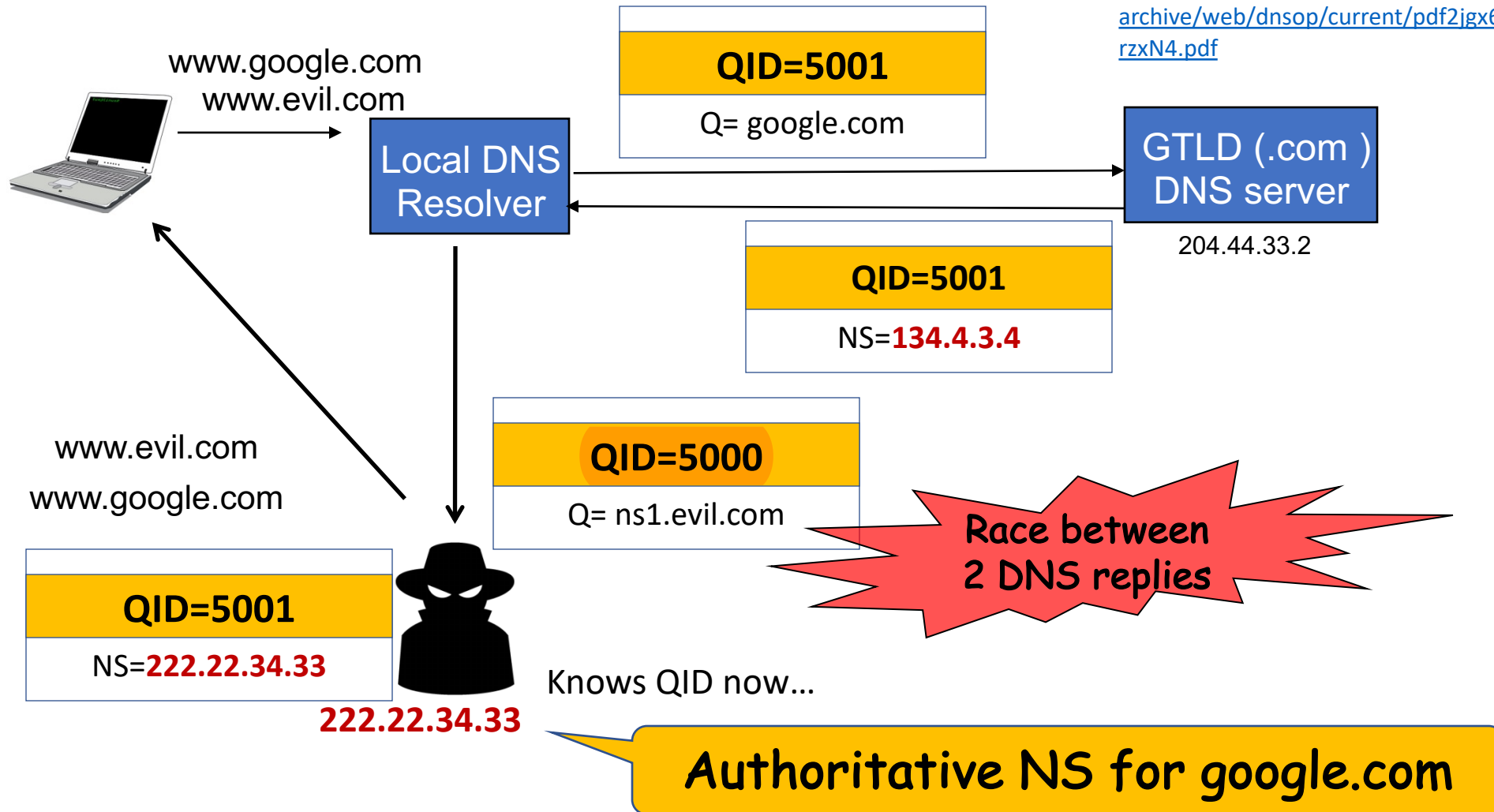
DNS Resolution: A Bit of Implementation Detail



DNS Cache Poisoning [Kaminsky'08]

Reference:

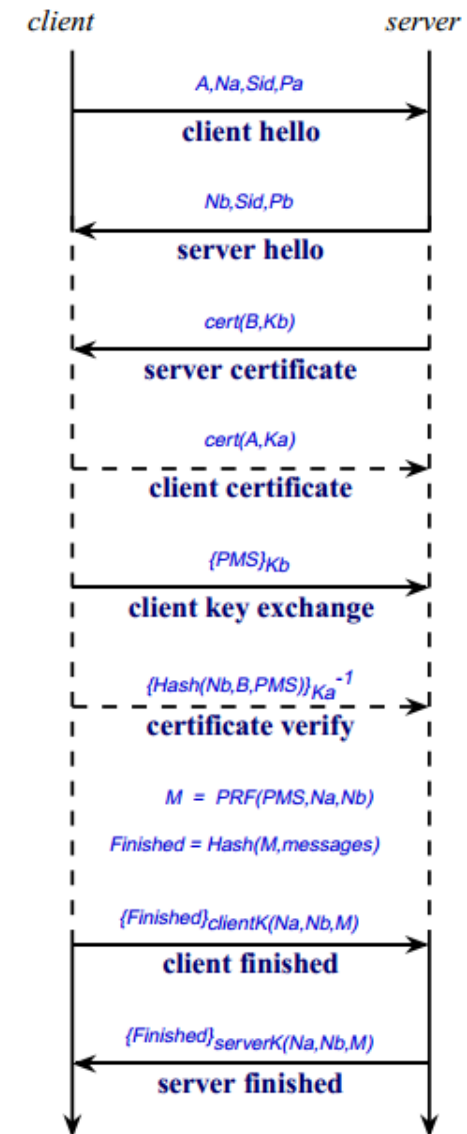
<https://www.ietf.org/mail-archive/web/dnsop/current/pdf2jgx6rzn4.pdf>



Secure Channels: Defense against Network Attacks

Secure Channel on the Internet: SSL + HTTP = HTTPS

- Used in HTTPS, SMTP, fax, ...
- Netscape SSL 2.0 [1993] .. TLS 1.2 [2008]
- How does it work?
 - Ciphers Negotiation
 - Authenticated Key Exchange (AKE):
the exchange of session key, which
also authenticates the identities of
parties involved
 - Symmetric Key Encryption



A Very High-Level of TLS



Negotiation Phase



Key Exchange (RSA, DHE, ...)



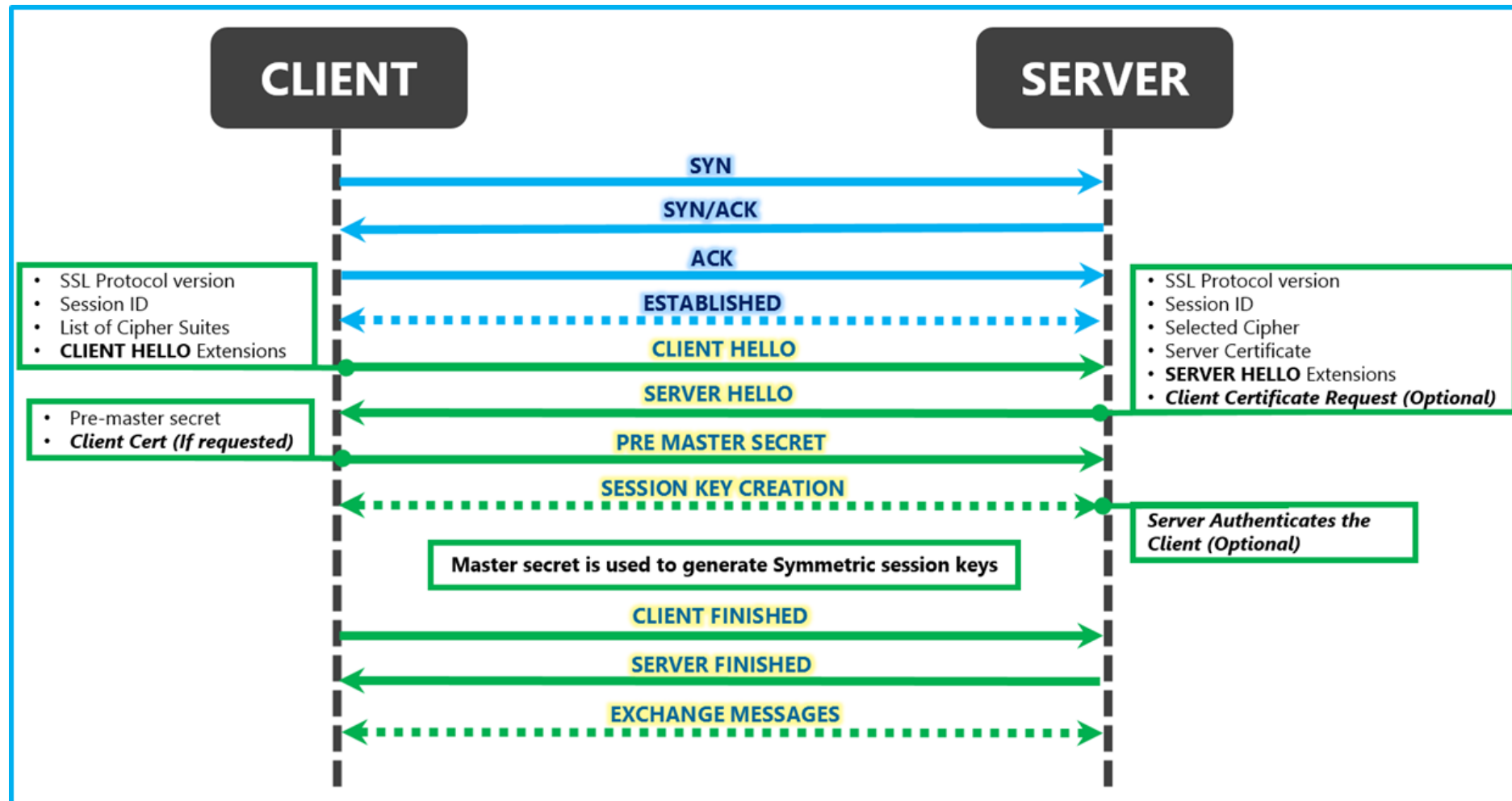
Symmetric Key Encrypted Session



Re-negotiation



TLS Handshake: Sample Session

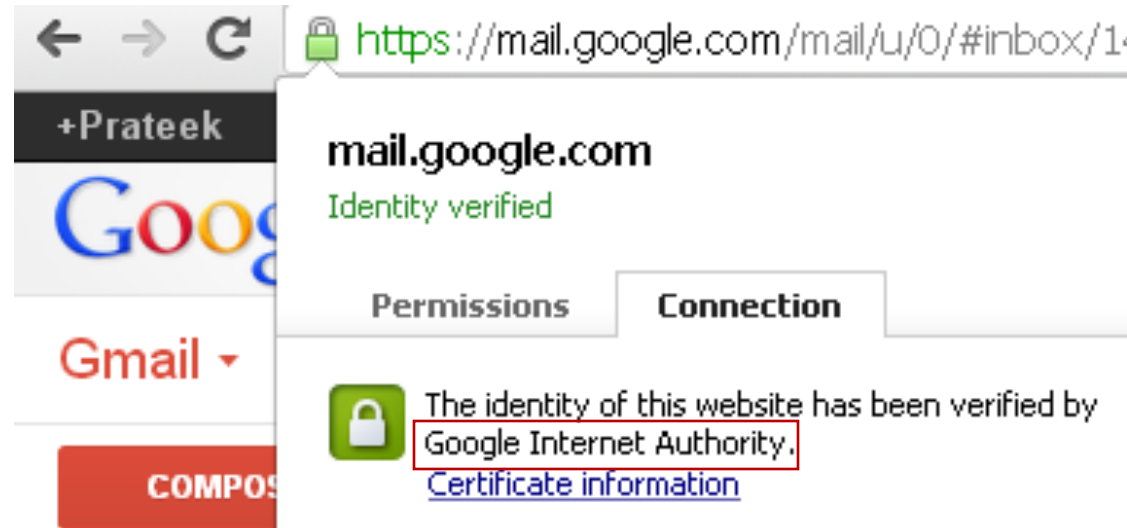


Panday, Microsoft Developer's Blog:

[https://blogs.msdn.microsoft.com/kaushal/2013/08/02/ssl-handshake-and-https-](https://blogs.msdn.microsoft.com/kaushal/2013/08/02/ssl-handshake-and-https-bindings-on-iis/)

[bindings-on-iis/](#)

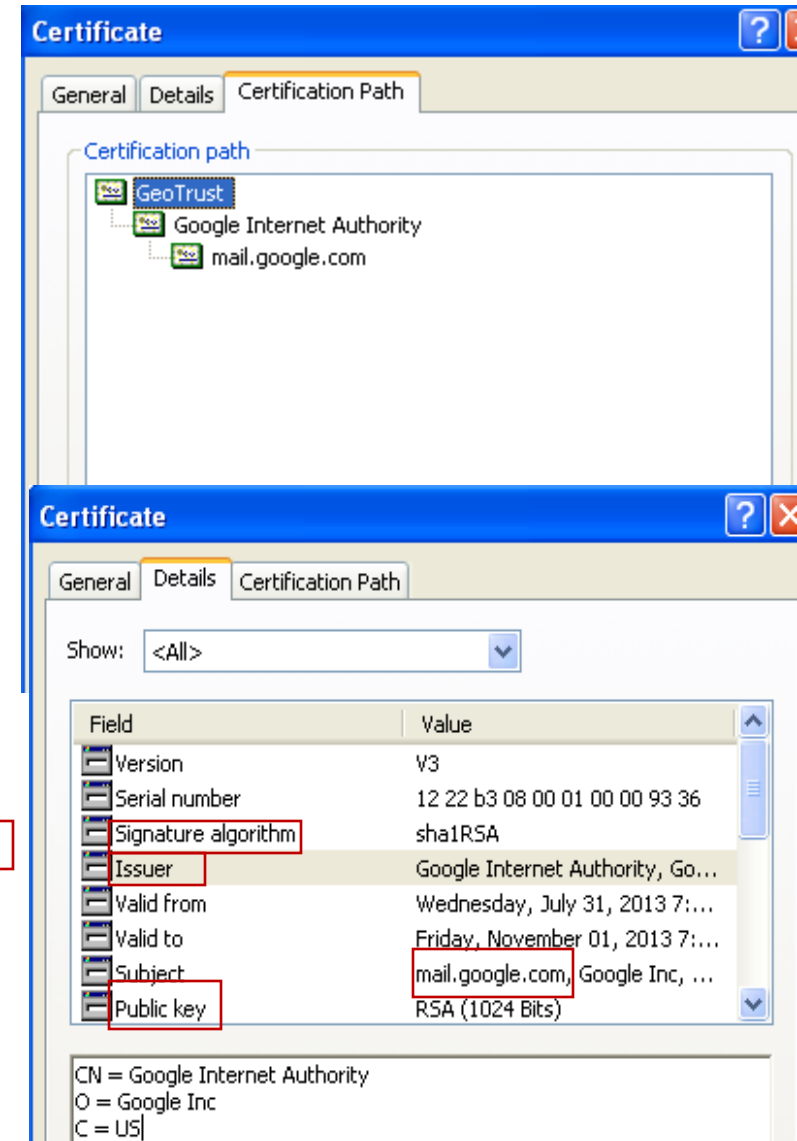
HTTPS



MAC

Encrypt

KE
protocol

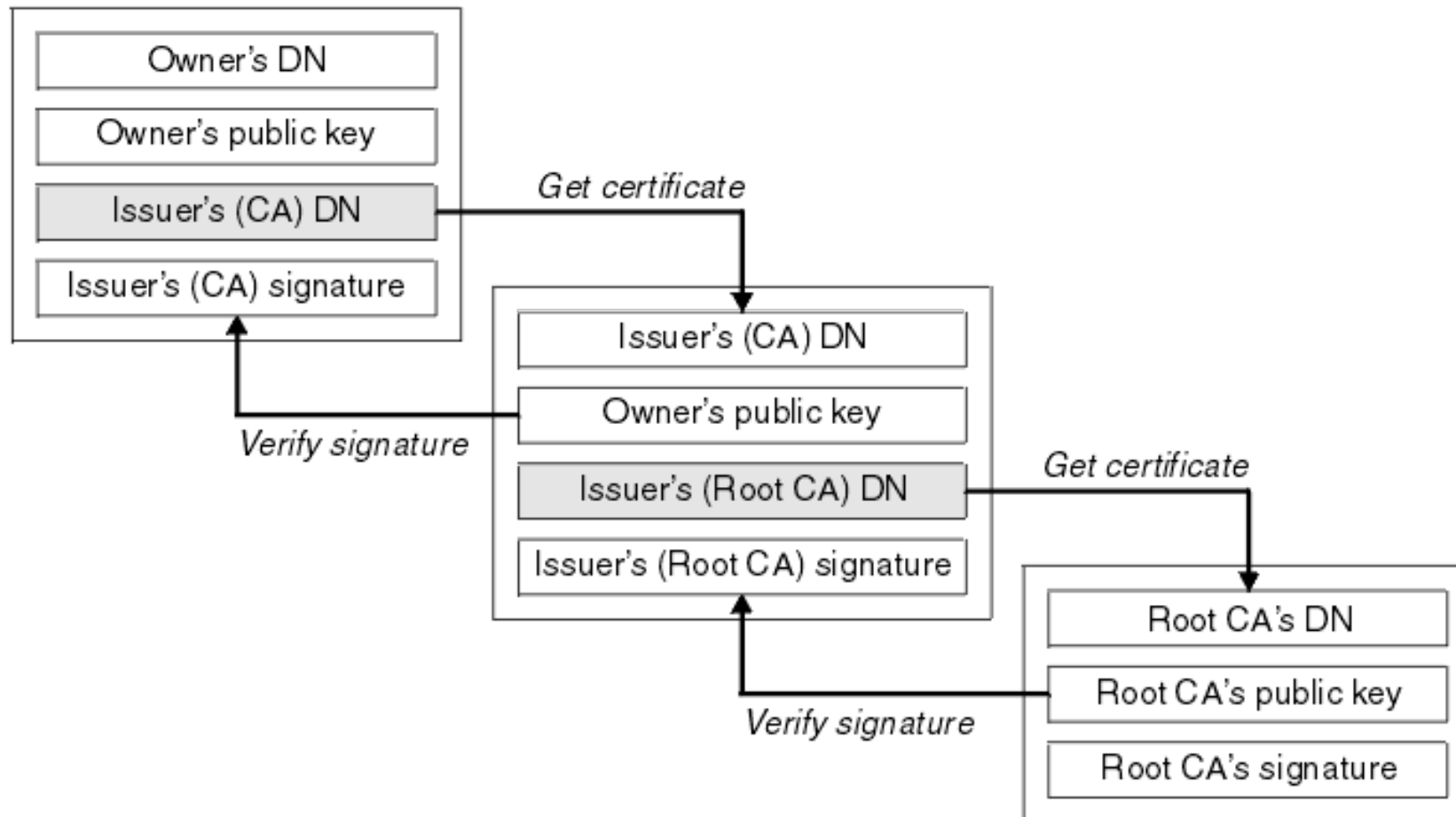


Chain Of Trusted Certificates

- Root CAs (e.g. GeoTrust)
 - Can designate Intermediate CA
 - E.g. Google Internet Authority
 - Restricted to signing certs for its subdomains
- Where does the browser start trusting?
 - Root CA's certificates are baked in your browser
 - ~50
- Who are the root CAs for the web?
 - Symantec (GeoTrust) – 38%
 - Comodo – 20%
 - GoDaddy – 13%, GlobalSign – 10%

Chain Of Trusted Certificates

- How Does the Chain Get Verified?

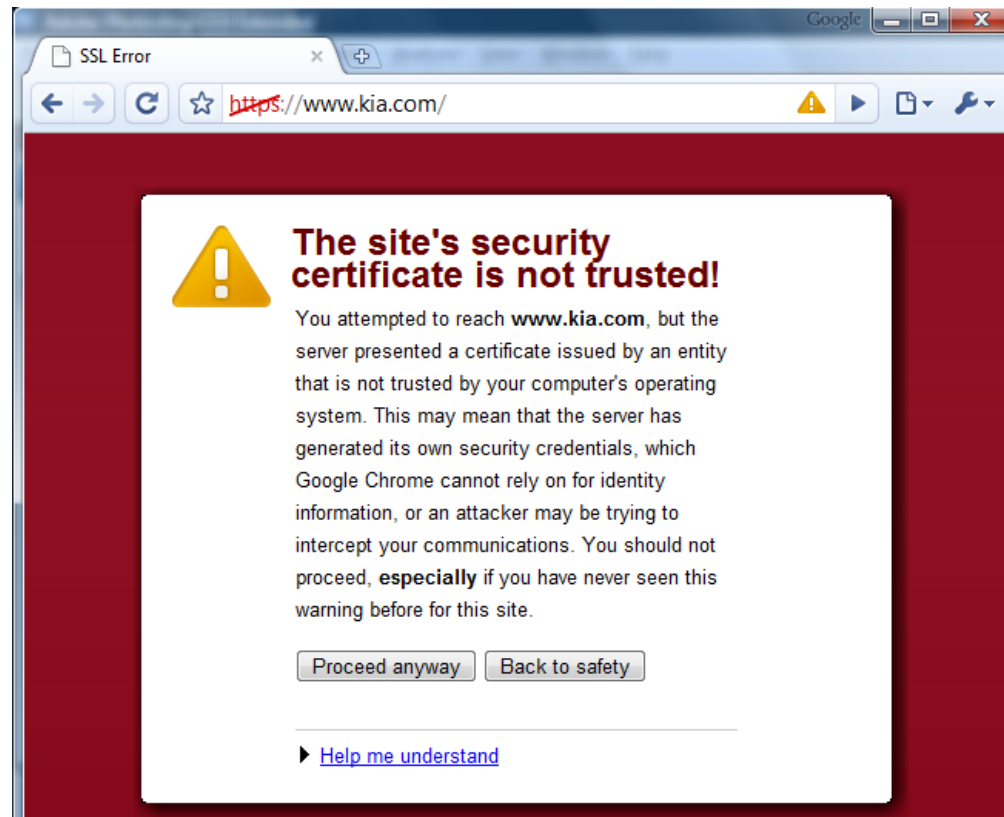


How Do I Get an SSL cert?

- Get a Root CA to issue you one
 - You can get some for free 😊
 - Paid ones: \$10 - \$50 / yr (not costly)
- What do they check for before issuing cert?
 - Valid email
 - You own the domain you want cert for?
 - E.g. you are the admin at <http://evil.com>
 - Sometimes a bit deeper, but basically that's it!

Can I Be A CA?

- Yes,
 - Self-sign certificates
 - Customers need to add you as root CA
- Otherwise:



Defeating HTTPS (I): HTTPS Downgrade

HTTPS Downgrade: Scenario 1

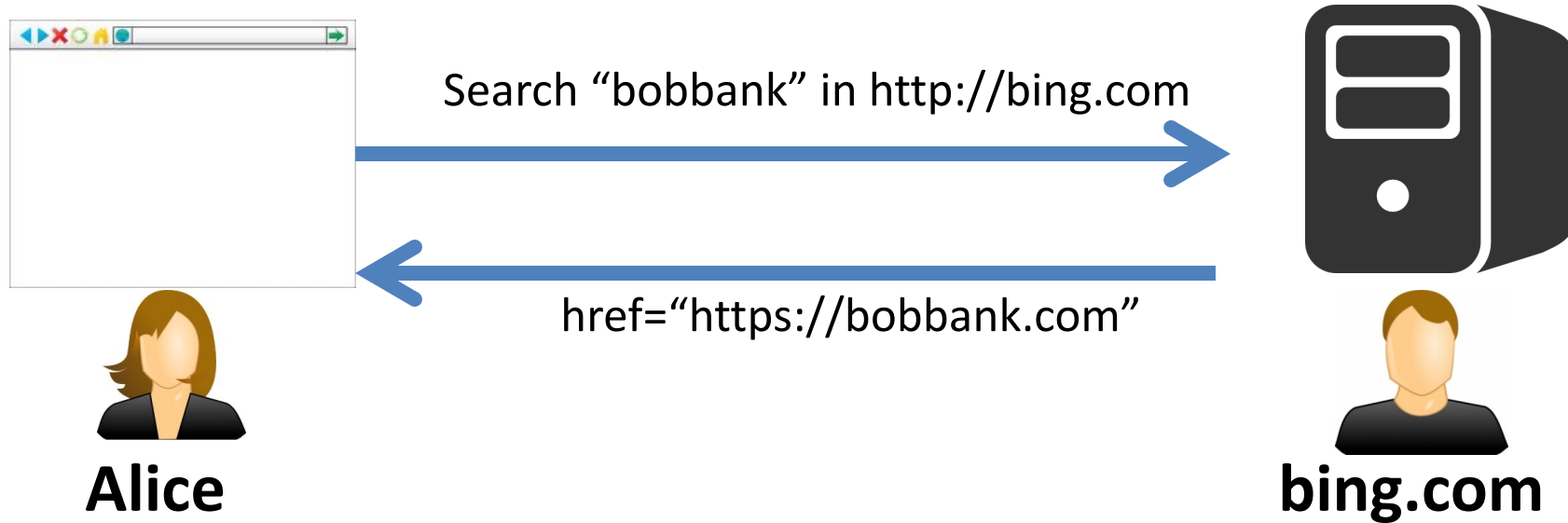
- Is this a good security practice?

http://example.com

```
<script src="https://example.com/lib.js">
```

Is this safe?

HTTPS Downgrade: Scenario 2

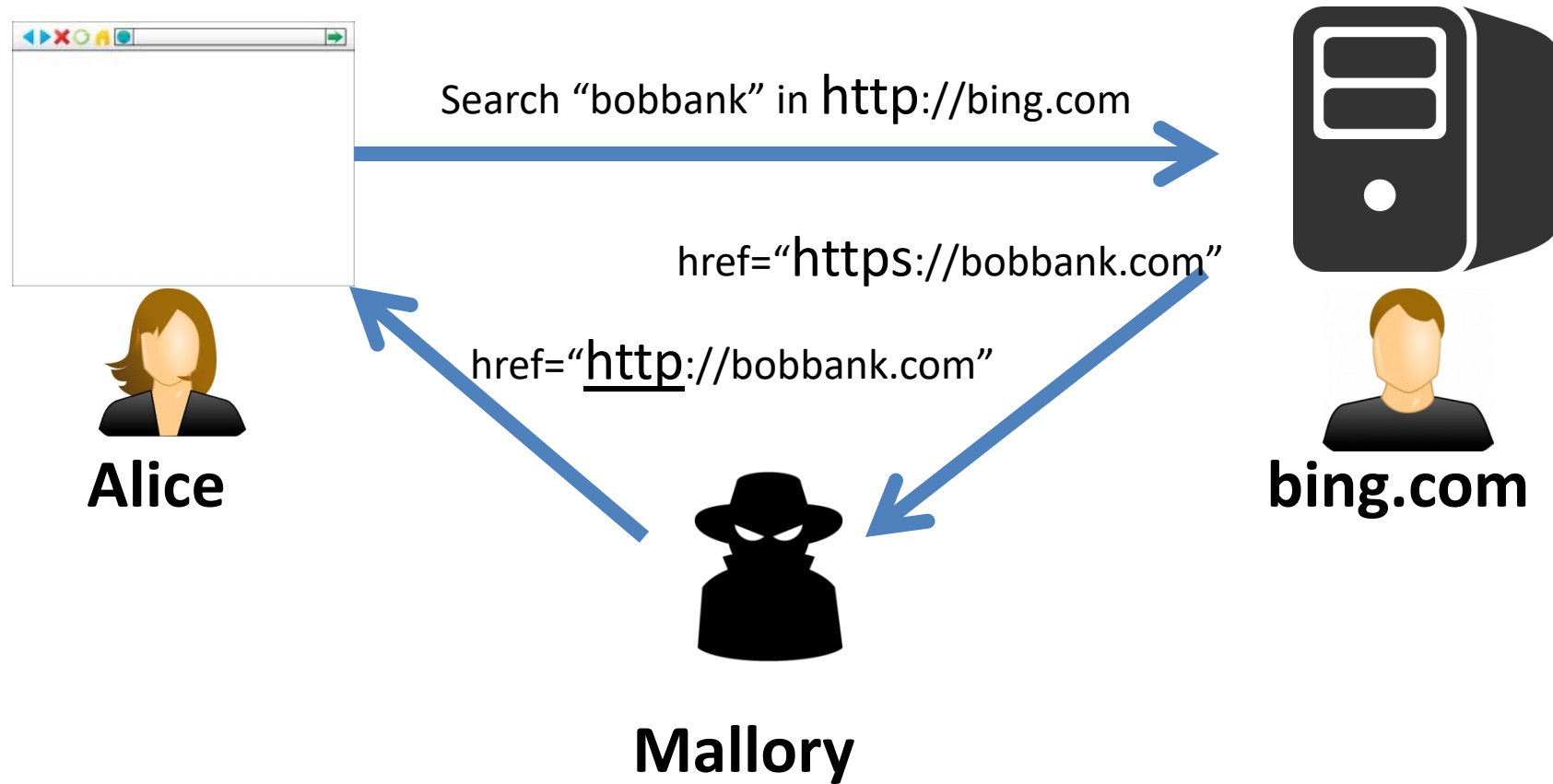


An HTTPS link access is very often a result of:

- An HTTPS link that a user clicks on from an HTTP page
- A 302 redirect from an HTTP URL to an HTTPS URL

[Moxie Marlinspike, "More Tricks For Defeating SSL", DEFCON 17](#)

HTTPS Downgrade: Scenario 2



More Downgrade Tricks

- Drop-in a clever favicon, which is shown in the address bar in older browsers
(Note: modern browsers show a favicon in the title bar, and not the address bar)
- Inject “Set-cookie” headers to delete existing session cookies. This forces re-login!
- Careful users can be careless at times too!

Defense Against HTTPS Downgrade

- **HSTS: HTTP Strict Transport Security**
- Idea: Server supplies a header over HTTPS

```
Strict-Transport-Security: max-age=63072000; includeSubDomains
```

- Browser subsequently never issues any non-HTTP request to this site
- In other words, connections to the site will always use TLS/SSL
- HSTS flag will be deleted when user clears private data: security vs privacy issue

Issue with HSTS

- First-visit problem
- “HSTS preloaded list”: <https://hstspreload.org/>
- See the complete list at:
https://cs.chromium.org/chromium/src/net/http/transport_security_state_static.json
- The solution is clearly not scalable
- Other workarounds:
 - Incorporating DNS records to declare HSTS Policy, and accessing them securely via DNSSEC
 - Manual inclusion: chrome://net-internals/#hsts
 - An extra client-side protection: HTTPS Everywhere browser extension (caveat: it is also based on a whitelist approach)

Defeating HTTPS (II): User Coopting & Caching Flaws

Some UI Tricks

- Include many lock icons/images
- Show logo images from the actual site
- Spoof status bar with images
- Picture-in-picture technique
- URL shown in the real status bar can also be spoofed easily?

```
<a href="http://www.bobbank.com/">  
onclick="this.href='http://www.hacker.com/';"> Click to go  
to BobBank</a>
```

Coopting the User to Click-through

- Do users pay attention to cert warnings?



This

You h:
that y

Norm:
that y

What

If you
trying

Get

► **Tecl**

► **I Un**

Operating System	SSL Warnings	
	Firefox	Chrome
Windows	32.5%	71.1%
MacOS	39.3%	68.8%
Linux	58.7%	64.2%
Android	NC	64.6%

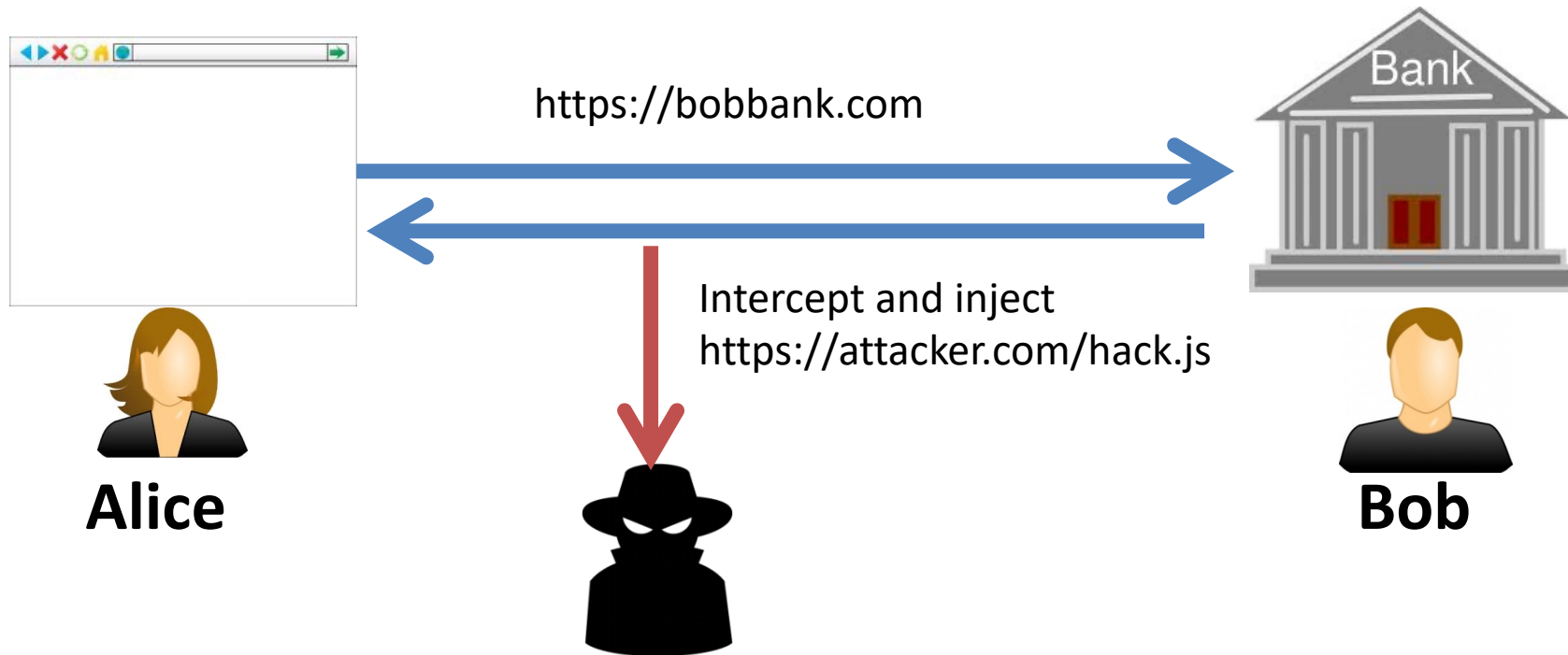
Table 3: User operating system vs. clickthrough rates for SSL warnings. The Google Chrome data is from the stable channel, and the Mozilla Firefox data is from the beta channel.

Channel	SSL Warnings	
	Firefox	Chrome
Release	NC	70.2%
Beta	32.2%	73.3%
Dev	35.0%	75.9%
Nightly	43.0%	74.0%

Table 4: Channel vs. clickthrough rates for SSL warnings.

Akhawe, Devdatta, and Adrienne Porter Felt. "Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness." Usenix Security. 2013.

Let's See an Example: HTTPS Hijacking



Defeating HTTPS(III): Mixed Content

Limits of HTTPS

- **Mixed Content Bugs** (HTTP + HTTPS)

<https://example.com>

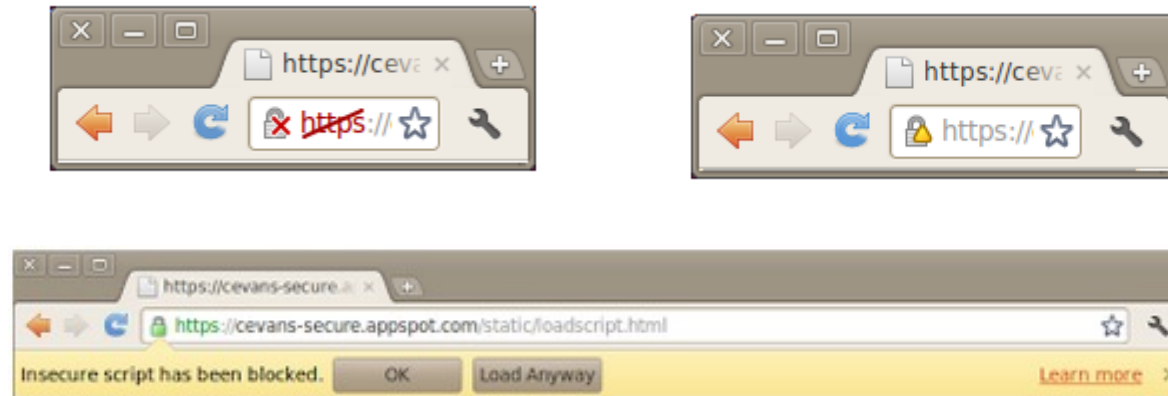
```
<script src="http://example.com/lib.js">
```

Is this safe?

Attacker can corrupt JS
and include payload

Mixed Content and Browser

- Mixed Content: What do browsers do?
 - Legacy: Ignore, No security warning.
 - Recently: New UI Indicators
 - Are these effective?



Mixed Content Types & Risks

- Two types of mixed content: active and passive
- **Passive mixed content:**
 - Content that doesn't interact with the rest of the page
 - Examples: images, video, audio content
 - A MITM attack is restricted to changing that content. Any big deal with this?
 - But again, it may leak cookies
 - Also a privacy issue
- **Active mixed content:**
 - Interacts with the page as a whole
 - Examples: **scripts**, iframes, flash resources
 - Allows an attacker to do almost anything with the page!

Mixed Content Defense

- How to deal with *legacy pages* with multiple insecure URLs that need to be rewritten?
- Use “find and replace”? Is this fool proof?
- We can use “protocol-relative URLs”:
``
- Additionally use Content Security Policy (CSP) `upgrade-insecure-requests` directive:
 - Browser treats all of a site's insecure URLs (served over HTTP) as though they have been replaced with secure URLs (served over HTTPS)
 - Example:
// header
`Content-Security-Policy: upgrade-insecure-requests;`

// meta tag
`<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">`

Defeating HTTPS (IV): CA Compromise

Limits of HTTPS

- Trust the Root CAs?

Four CAs Have Been Compromised Since June

Posted by **Soulskill** on Friday October 28 2011 , @04:08PM
from the four-whole-californias-wow dept.

News

Hackers spied on 300,000 Iranians using fake Google certificate

Investigation reveals month-long, massive Gmail snooping campaign

By **Gregg Keizer**

September 6, 2011 05:43 AM ET  4 Comments

DigiNotar Breach

- Incident took place in 2011
- 500+ fraudulent certs issued, including for *.google.com, *.mozilla.com, *.windowsupdate.com, *.torproject.org
- 300K+ IP addresses got MITM-ed, mostly from Iran
- DigiNotar was immediately removed from Root CA list by major browsers
- What happened to DigiNotar afterwards?

Added Root CAs by System Providers

- System providers may include their root CAs, usually together with some forged certs
- They claimed the forged certs were used for monitoring, giving extra features, ...

Gogo Inflight Internet Used Fake SSL Certificates

Home > Article > Gogo Inflight Internet Used Fake SSL Certificates

March 4, 2015 SSL Support Team

Lenovo PCs ship with man-in-the-middle adware that breaks HTTPS connections [Updated]

Superfish may make it trivial for attackers to spoof any HTTPS website.

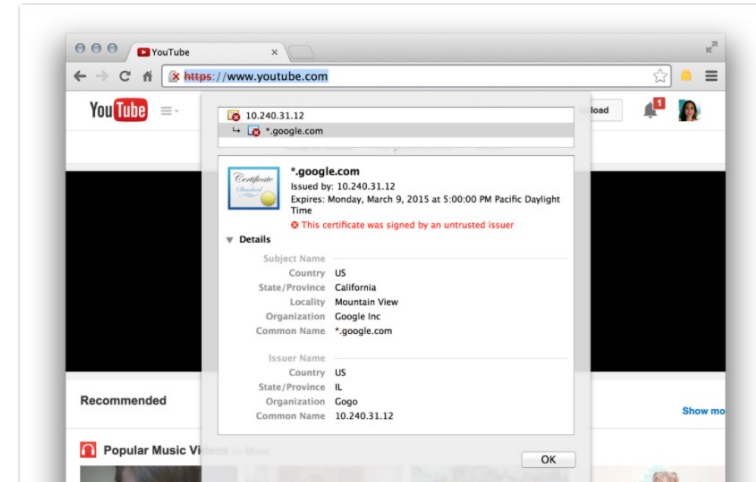
DAN GOODIN - 2/20/2015, 12:36 AM



Adrienne Porter Felt
@_apf_

Follow

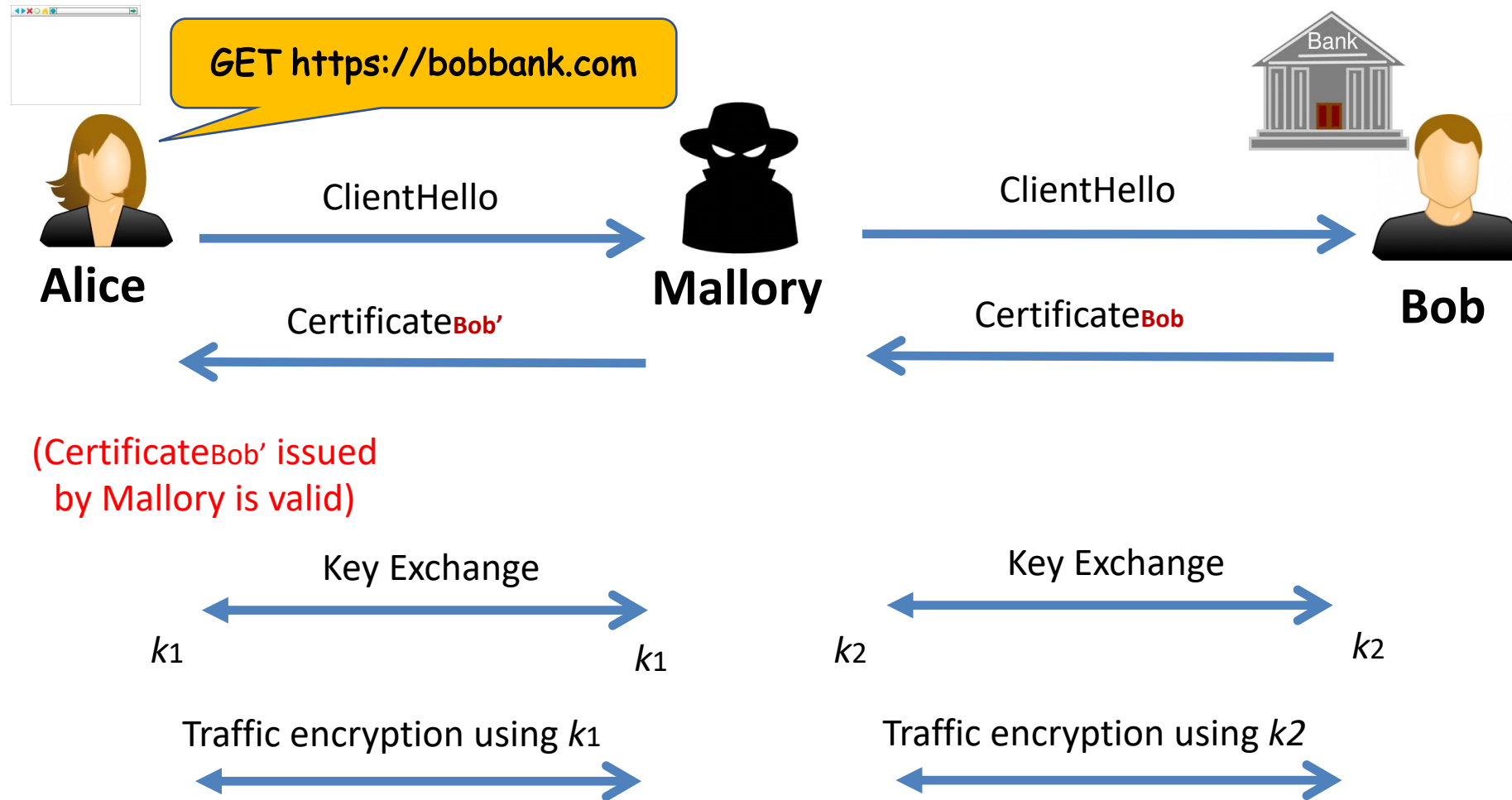
hey @Gogo, why are you issuing
*.google.com certificates on your planes?



Weak Browser Trust Model


- Browser trust model:
 - A pre-loaded list of widely-used root CAs compiled by browsers
 - An form of Certificate Trust List (CTL) approach:
a list of CAs' certificates are compiled by a trusted authority
- Security issue:
 - Trust anchor: the *union* of all root CAs
 - Question: which root CA is the one used from the root-CA list?
 - Certification is only as *strong* as the *weakest* root CA!
- Real-world analysis:
 - Eckersley and Burns, “An observatory for the SSLiverse”, Defcon 18, 2010
 - Eckersley and Burns, “Is the SSLiverse a Safe Place?”, 27th Chaos Communication Congress (CCC), 2010

MITM Attack Using Rouge Certificate



Mallory performs a proxy re-encryption
He can see all traffic, and also modify data!

Defenses Against Compromised Certs

- How to Detect If Being Served Bad Cert
 - Notaries (e.g. Convergence):
see “[SSL And The Future Of Authenticity](#)”
 - Certificate Revocation
 - Certificate Pinning
 -  Certificate Transparency
 - Perfect Forward Secrecy
- Basic idea: Deploy some extra measures to help protect the users despite bad certs

Certificate Revocation

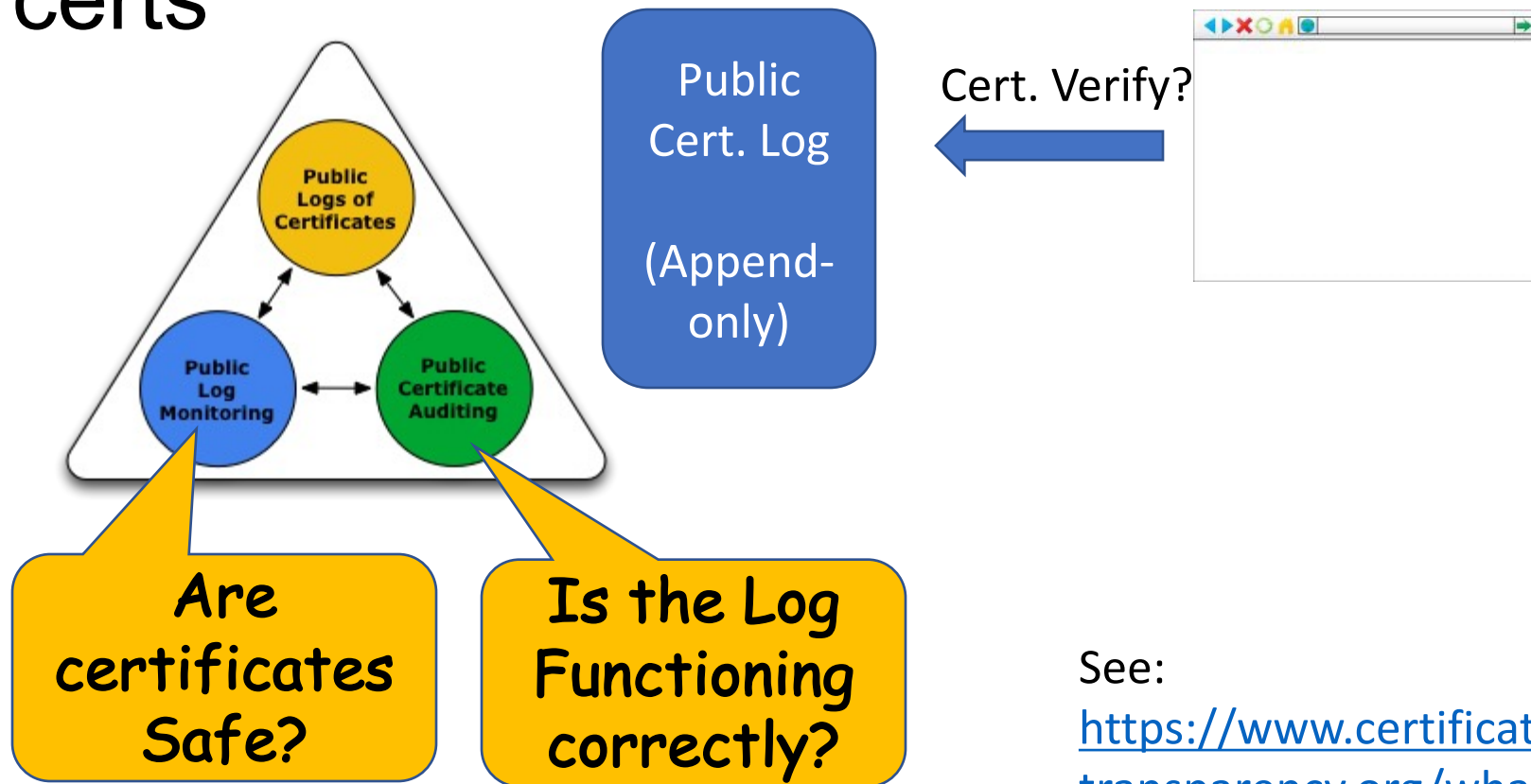
- Idea: CA can revoke compromised certs
- Supported by CRL and OCSP
 - CRL: CA signs a revocation list
 - OCPS: OCSP Responder validates a cert in question
- OCSP problems?
 - Time windows after compromise
 - Privacy: OCPS Responder knows certs you are validating
 - Soft-fail validation: Some browsers proceed in the event of no reply to an OCSP request (no reply *is* a good reply)

Certificate Pinning

- HTTP Public Key Pinning (HPKP)
- Delivered via an HTTP header (over HTTPS):
`Public-Key-Pins:`
`pin-sha256="cUPcTAZWKaASuYWhhneDttWpY3oBAkE3h2+soZS7sWs=";`
`pin-sha256="M8HztCzM3elUxkcjR2S5P4hhYBNf6lHkmjAHKhpGPWE=";`
`max-age=5184000; includeSubDomains;`
`report-uri="https://www.example.org/hpkp-report"`
- Associates a host with their expected X509 public keys
- Browser should only use the pinned public keys for connections to the domain (during the set pinning period; see `max-age`)

Certificate Transparency

- Idea: An open framework for publicly auditing all SSL certs



See:
<https://www.certificate-transparency.org/what-is-ct>

Certificate Transparency

- **Goals:**
 - Make it hard for a CA to issue a cert for a domain without the cert being visible to the domain owner
 - Provide an open auditing & monitoring system that lets any domain owner or CA determine whether certs have been mistakenly or maliciously issued
 - Protect users from being duped by mistakenly/maliciously issued certs
- **Main components/entities:**
 - **Logs:** maintains cryptographically assured, publicly auditable, append-only records of certificates
 - **Monitors** (usually CAs): identifies suspicious certs, and issues alerts
 - **Auditors** (usually browsers): identify misbehaving logs

Certificate Transparency: Public Log

- Idea: Using Merkle Trees
- Let you verify quickly
 - If a cert is in the log?
 - The log is append-only
 - Not tampered

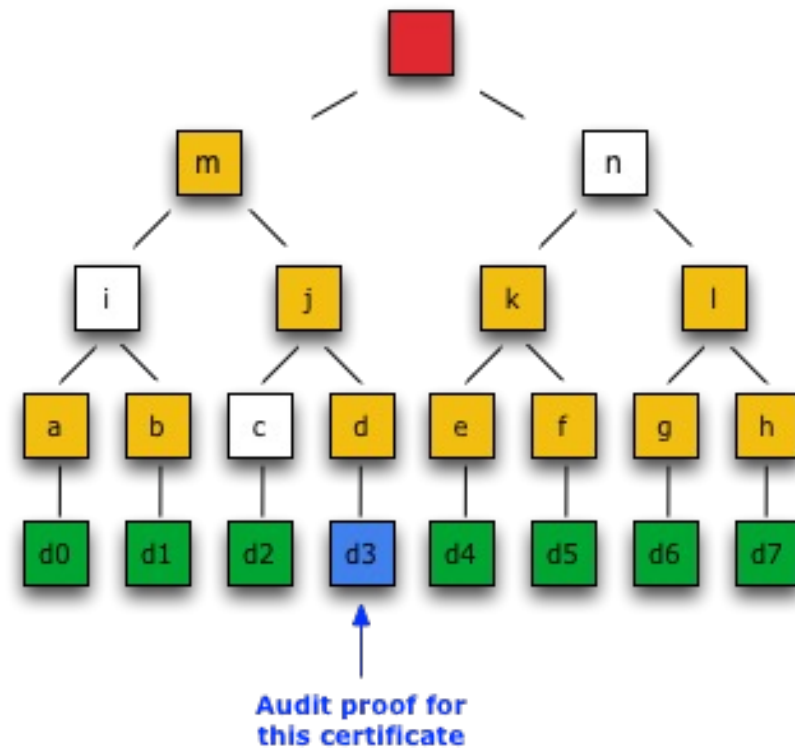


Figure 5

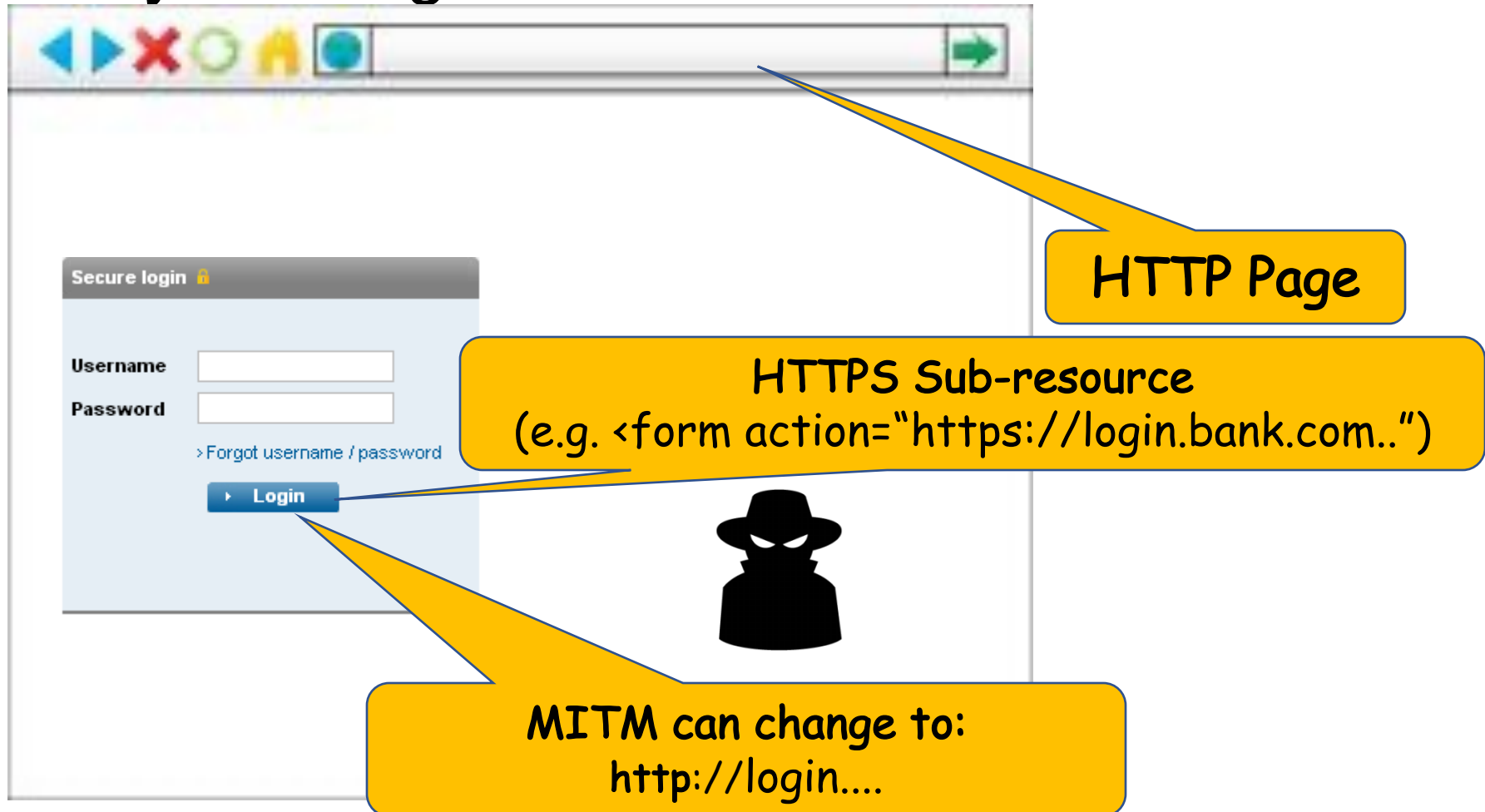
Certificate Transparency

- Benefits:
 - Early detection of misissued/malicious certs and rogue CAs
 - Faster mitigation after suspect certs or CAs are detected
 - Better oversight of the entire TLS/SSL system
- Success story: Symantec incident
 - It issued a cert for google.com, certs for non-existing domains
 - Some issued “testing certs”
- Google now requires the incorporation of Certificate Transparency for EV certs: certs are valid only if they are published on log servers

Defeating HTTPS (V): Protocol Implementation Flaws (Optional Material)

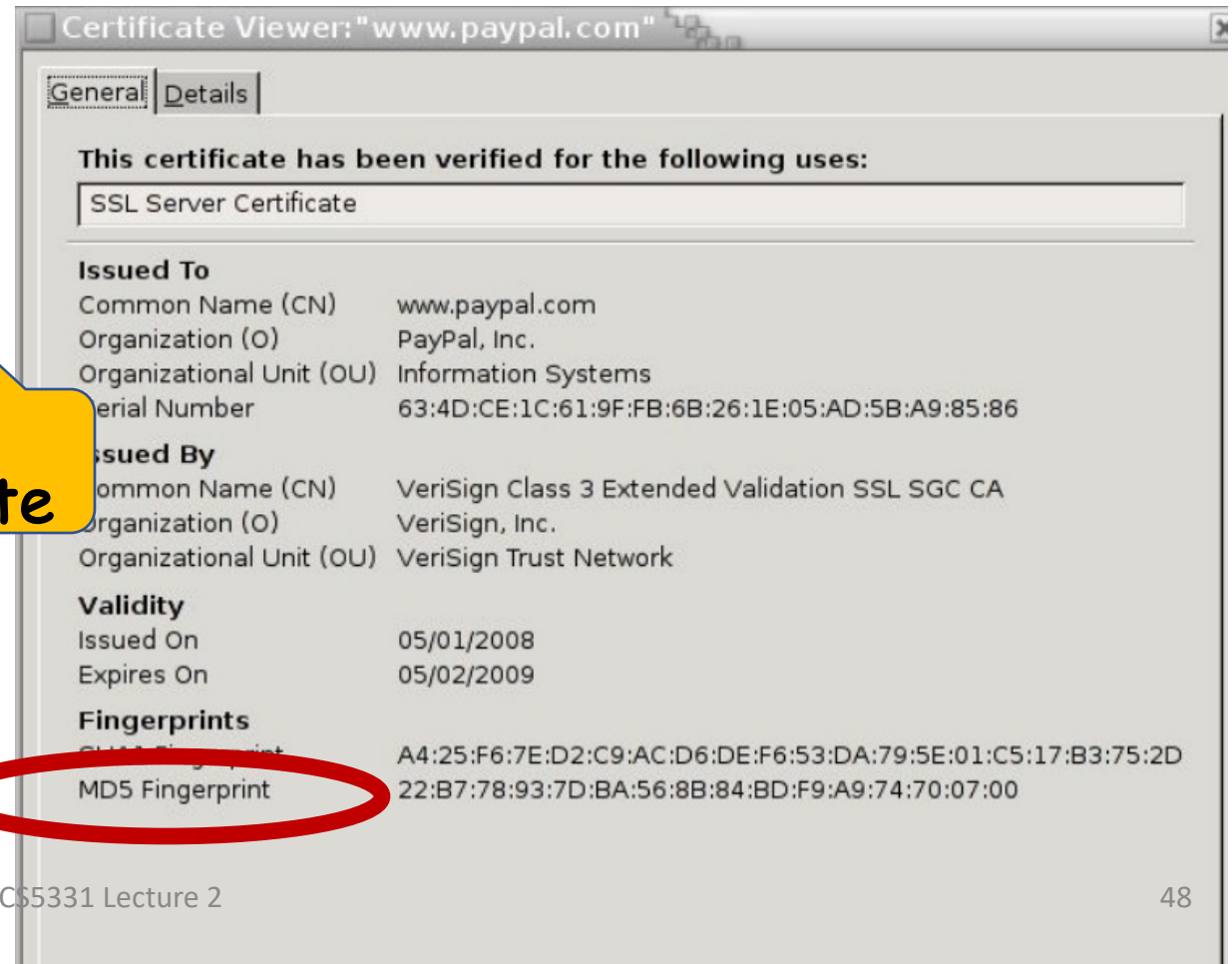
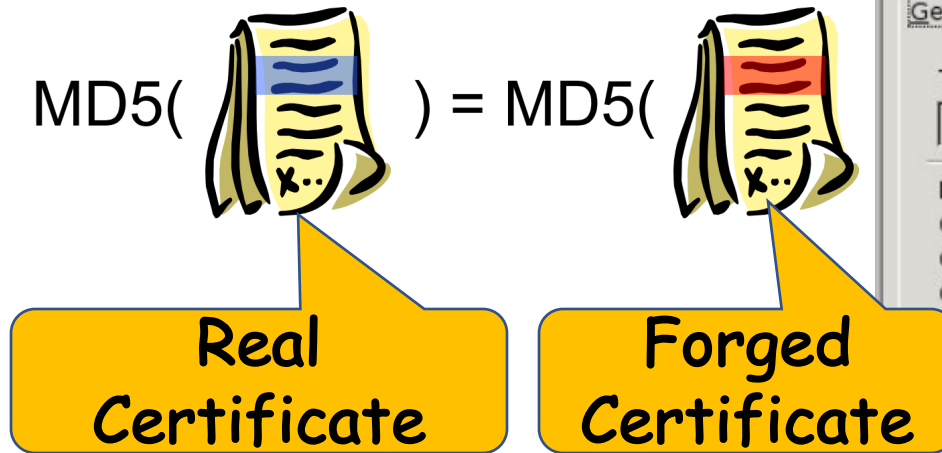
Implementation Flaws: Sub-resource Hijacking

- No security warning for sub-resource loads

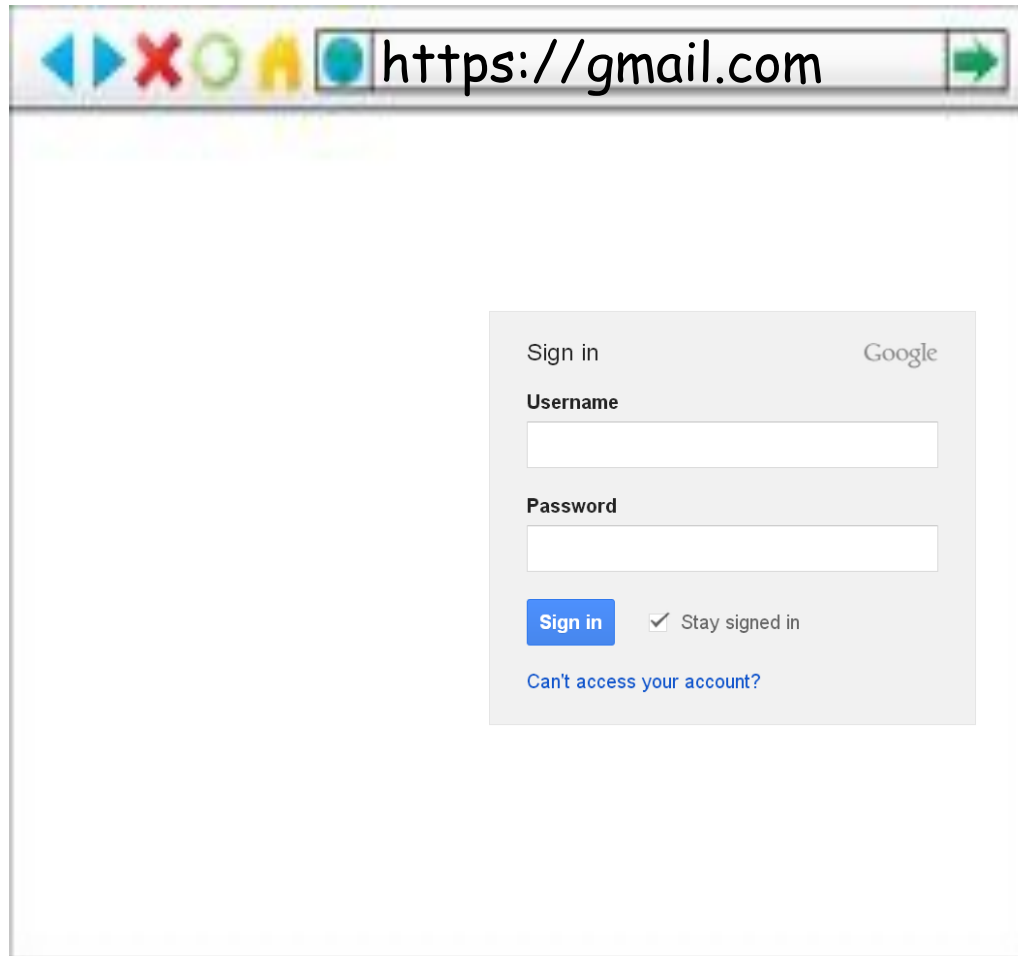


Implementation Flaws:(II):Forged Certs.

- Can attack the cryptographic signing [Sotirov et al.]
- MD5 can have collisions [2004, 2007], SHA-1 too



Implementation Flaws (III): Forged Certs.



gmail.com\0.evil.com

“Null/termination byte injection attack” on a cert’s Common Name

Implementation Flaws (III): Forged Certs.

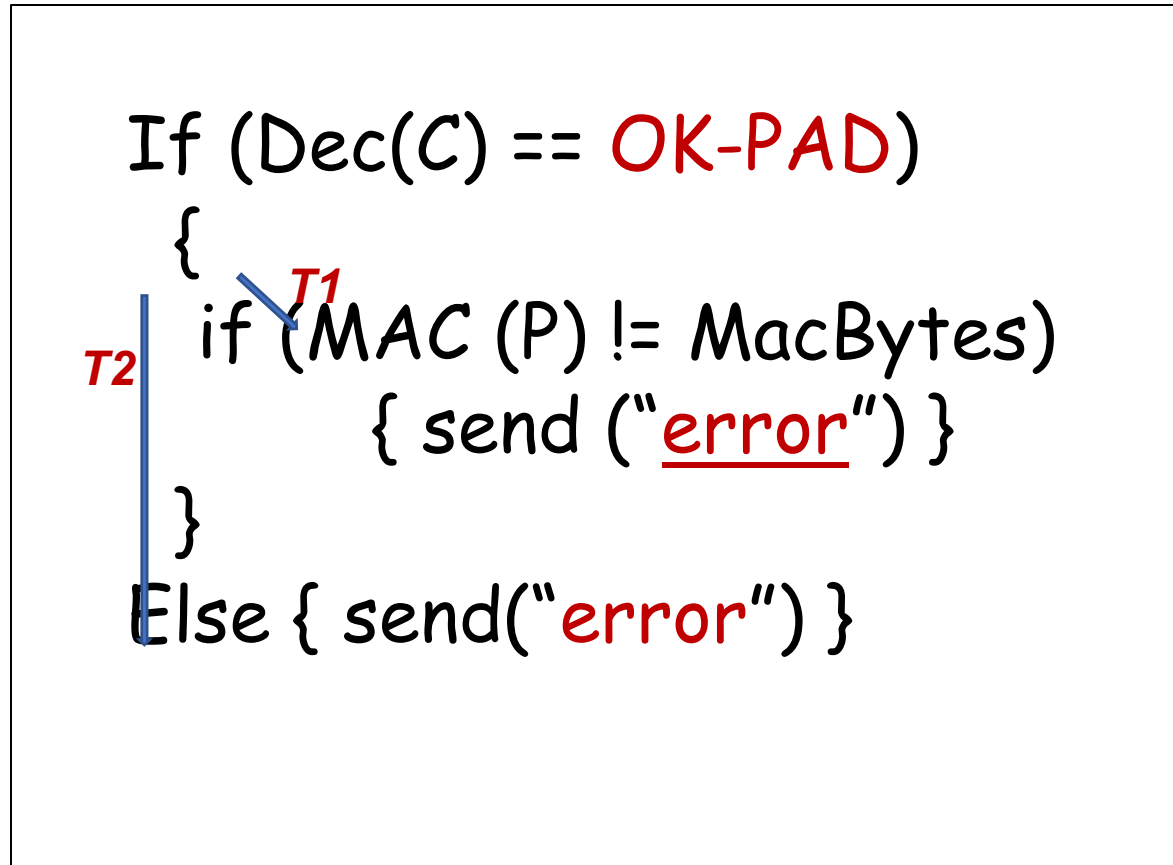


Registered
Cert for:

p#1072;ypal.com

“Domain encoding attack” on
a cert’s Common Name

Side-Channels: Timing Attacks on TLS Implementation

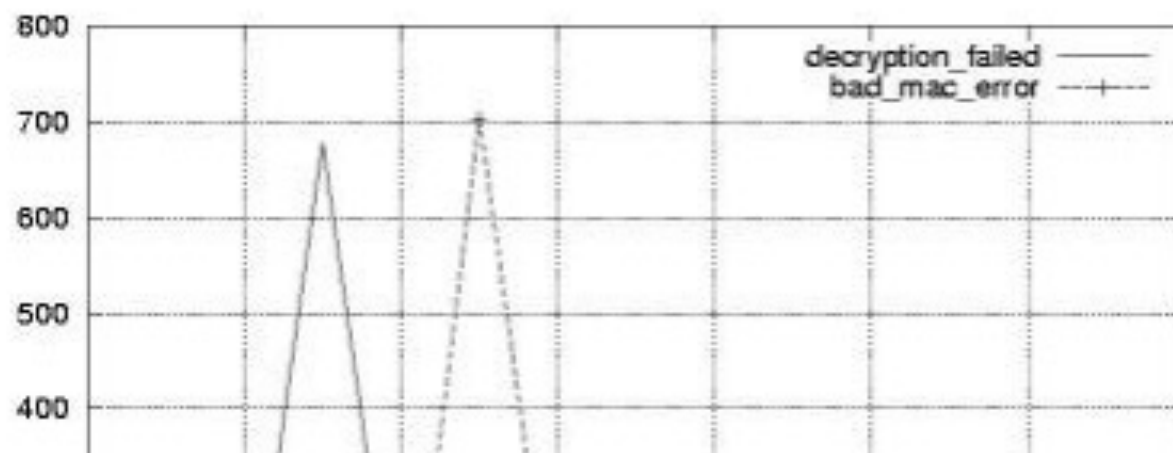


There is another problem with this code ...



Can distinguish execution times **T1** and **T2**

Side-Channels: Timing Attacks on TLS implementation



“Lucky Thirteen” attack snarfs cookies protected by SSL encryption

Exploit is the latest to subvert crypto used to secure Web transactions.

by **Dan Goodin** - Feb 4 2013, 10:14pm MPST

HACKING PRIVACY

Other Implementation Flaws

- Timing side-channels:
 - Vulnerable RSA PKCS#1 v1.5 [1998]
 - Compression
 - CRIME [2012], new one – BREACH [Aug 2013]
- Renegotiation attacks [Rescola 2009]
- IV in CBC mode incorrect => BEAST [2011]
- Uses RC4 (no padding needed)
 - RC4 is totally broken! (gives biased stream)
- Downgrading attacks from 'strong' to 'export-grade' crypto algorithms: FREAK [2015], Logjam [2015]
- Check:
https://en.wikipedia.org/wiki/Transport_Layer_Security#Attacks_against_TLS/SSL

Summary

- Network technology and attacks
- Secure channels: HTTPS
 - Conceptually secure, but
 - Numerous practical issues in getting it right
- Several useful techniques:
 - HSTS and HSTS preloaded list
 - Secure cookies
 - Avoid mixed contents, and additionally set CSP's upgrade-insecure-requests
 - OCSP stapling and OCSP Must-Staple
 - Certificate Pinning
 - Crypto suite that provides Perfect Forward Secrecy
 - EV Certificate with Certificate Transparency