CS 5436 /
INFO 5303

# SSL/TLS

Vitaly Shmatikov

# What Is SSL / TLS?

◆ Secure Sockets Layer and Transport Layer Security protocols
  - Same protocol design, different crypto algorithms

◆ De facto standard for Internet security
  - "The primary goal of the TLS protocol is to provide privacy and data integrity between two communicating applications"

◆ Deployed in every Web browser; also VoIP, payment systems, distributed systems, etc.

# SSL / TLS Guarantees

◆ End-to-end secure communications in the presence of a <span style="color:red">network attacker</span>

- Attacker completely 0wns the network: controls Wi-Fi, DNS, routers, his own websites, can listen to any packet, modify packets in transit, inject his own packets into the network

◆ Scenario: you are reading your email from an Internet café connected via a r00ted Wi-Fi access point to a dodgy ISP in a hostile authoritarian country

# History of the Protocol

◆ SSL 1.0 – internal Netscape design, early 1994?

- Lost in the mists of time

◆ SSL 2.0 – Netscape, Nov 1994

- Several weaknesses

◆ SSL 3.0 – Netscape and Paul Kocher, Nov 1996

◆ TLS 1.0 – Internet standard, Jan 1999

- Based on SSL 3.0, but not interoperable (uses different cryptographic algorithms)

◆ TLS 1.1 – Apr 2006

◆ TLS 1.2 – Aug 2008

# SSL Basics

◆ SSL consists of two protocols

◆ Handshake protocol

- Uses public-key cryptography to establish several shared secret keys between the client and the server

◆ Record protocol

- Uses the secret keys established in the handshake protocol to protect confidentiality, integrity, and authenticity of data exchange between the client and the server
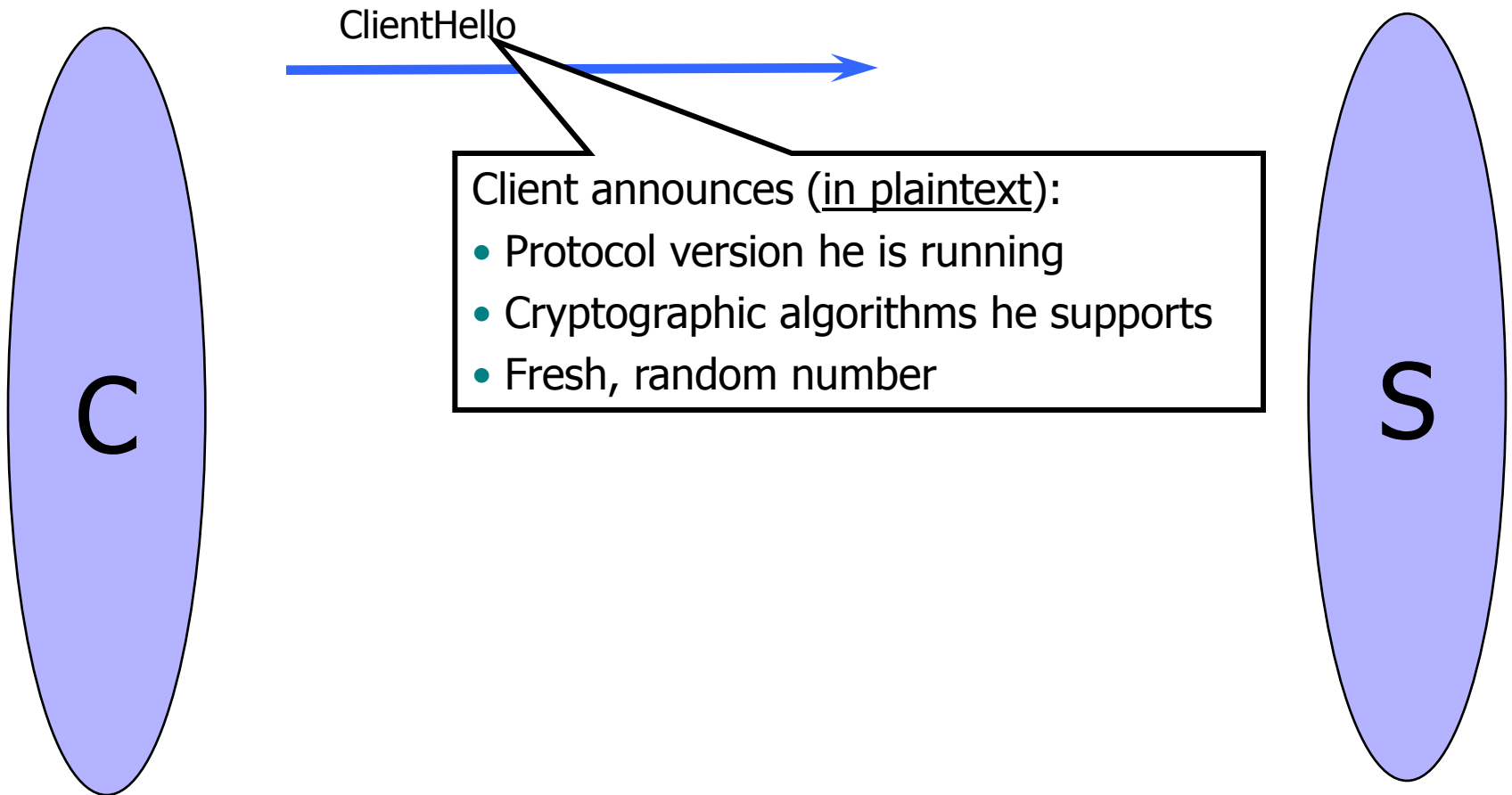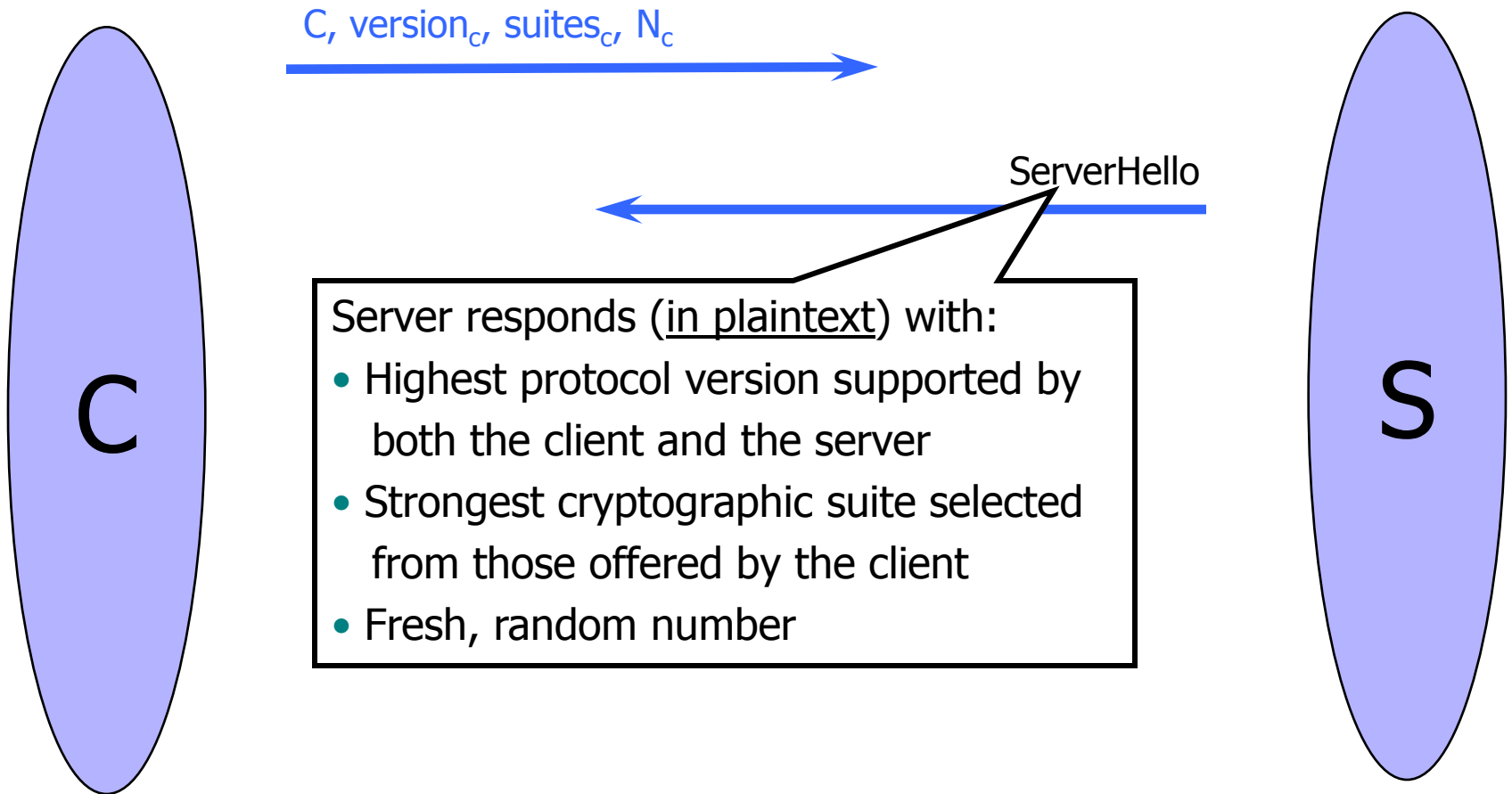
# SSL Handshake Protocol

◆ Runs between a client and a server

- For example, client = Web browser, server = website

◆ Negotiate version of the protocol and the set of cryptographic algorithms to be used

- Interoperability between different implementations

◆ Authenticate server and client (optional)

- Use digital certificates to learn each other's public keys and verify each other's identity
- Often only the server is authenticated

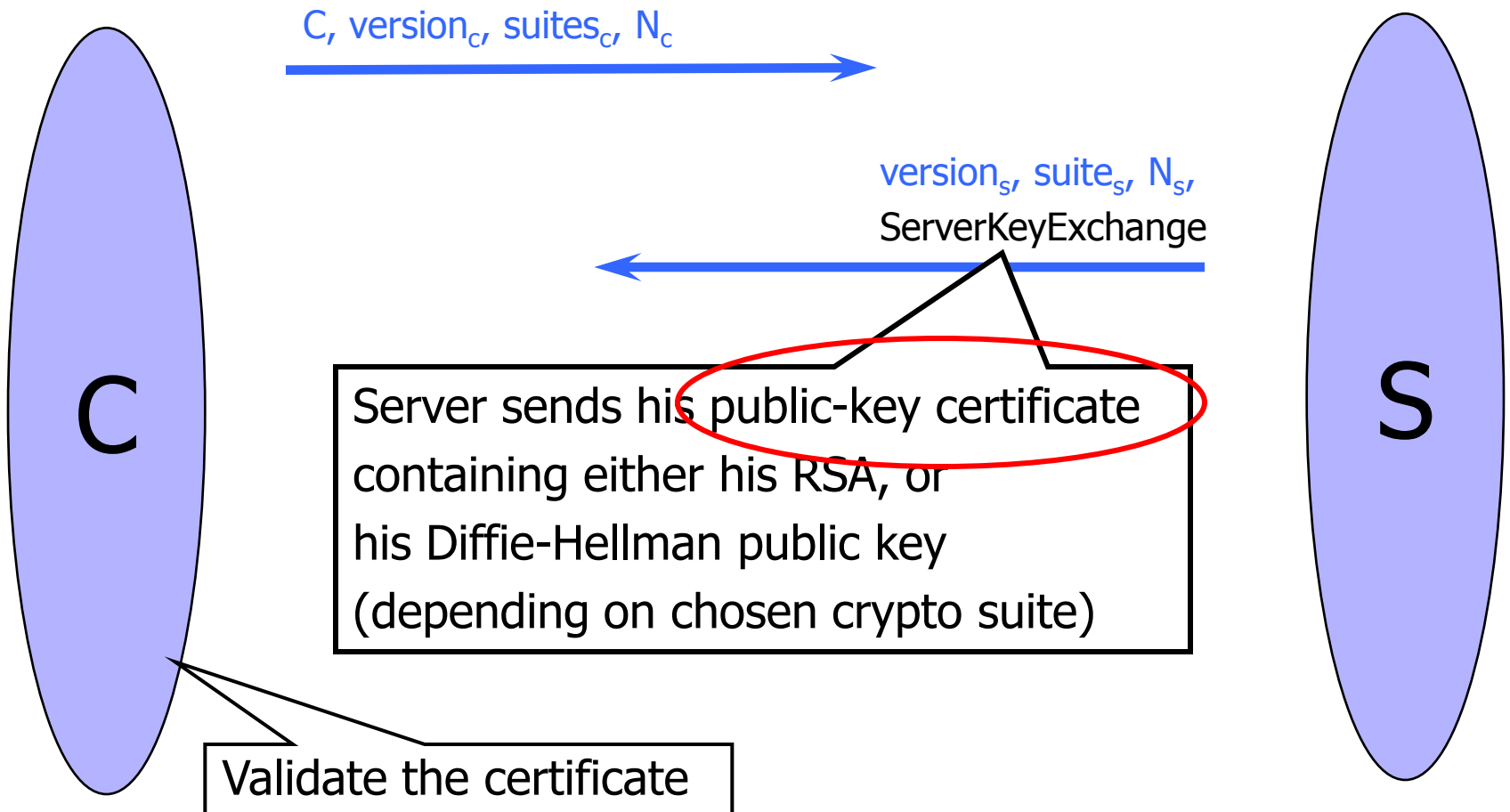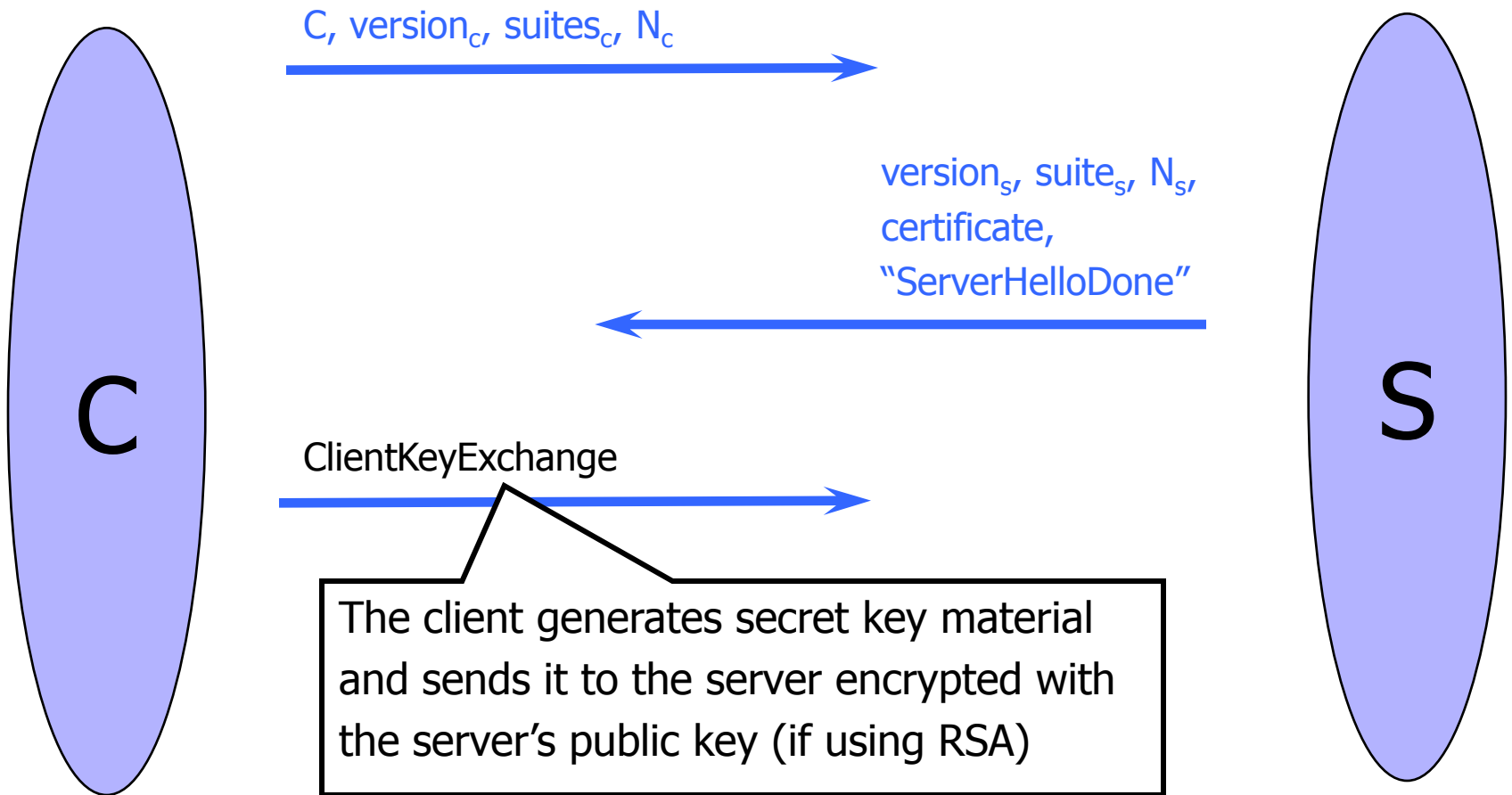◆ Use public keys to establish a shared secret

# ClientHello

ClientHello

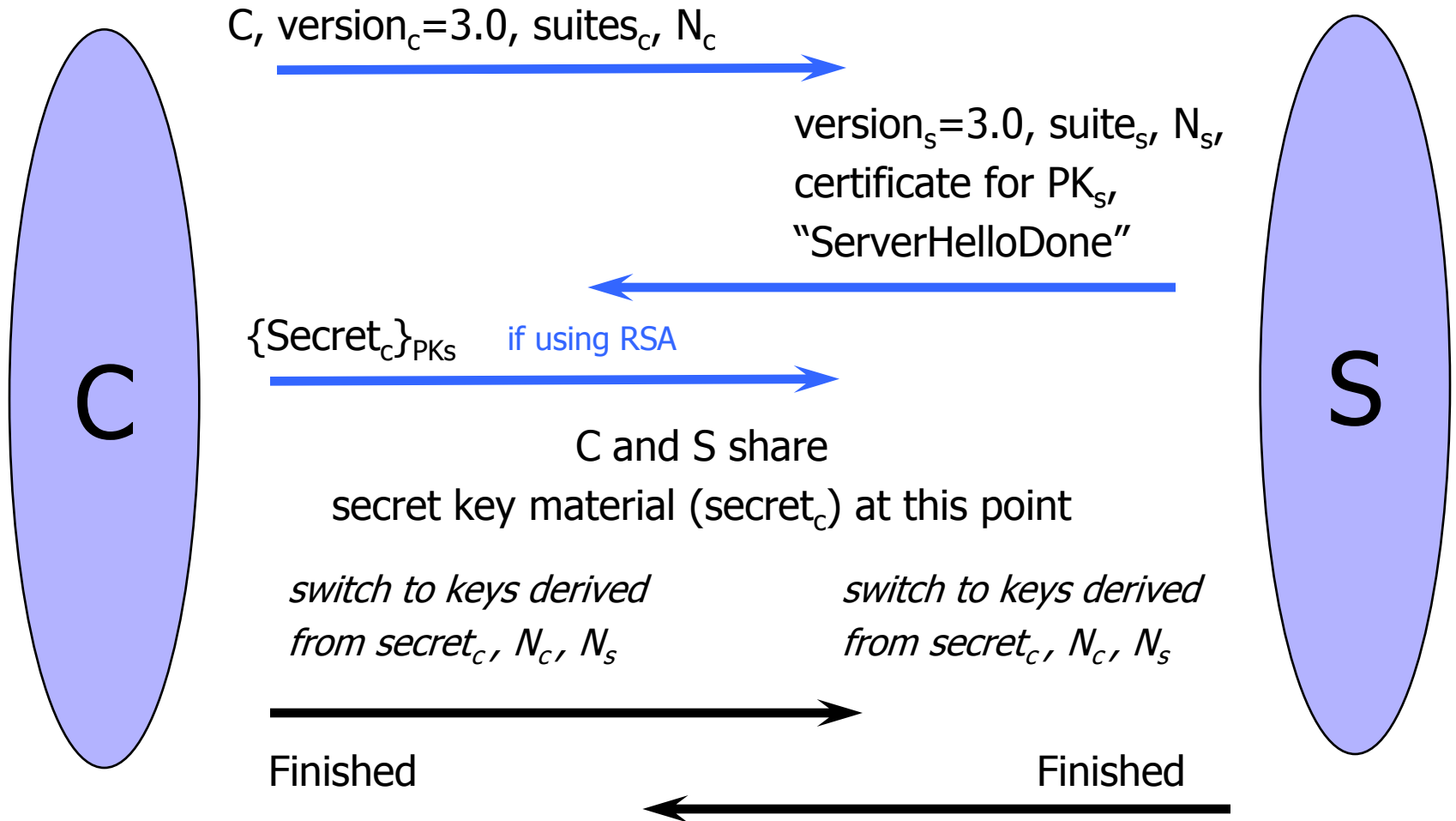Client announces (in plaintext):
- Protocol version he is running
- Cryptographic algorithms he supports
- Fresh, random number

C

S

# ServerHello

$C, version_c, suites_c, N_c$

ServerHello

Server responds (in plaintext) with:
- Highest protocol version supported by both the client and the server
- Strongest cryptographic suite selected from those offered by the client
- Fresh, random number

C

S

# ServerKeyExchange

$C$, $version_c$, $suites_c$, $N_c$ →

← $version_s$, $suite_s$, $N_s$,
ServerKeyExchange

**C**

**S**

Server sends his public-key certificate
containing either his RSA, or
his Diffie-Hellman public key
(depending on chosen crypto suite)

Validate the certificate

# ClientKeyExchange

$C$, $version_c$, $suites_c$, $N_c$

→

$version_s$, $suite_s$, $N_s$,
certificate,
"ServerHelloDone"

←

**C**

ClientKeyExchange

→

**S**

The client generates secret key material and sends it to the server encrypted with the server's public key (if using RSA)

# "Core" SSL Handshake

$C$, $version_c=3.0$, $suites_c$, $N_c$
→

$version_s=3.0$, $suite_s$, $N_s$,
certificate for $PK_s$,
"ServerHelloDone"
←

$\{Secret_c\}_{PKs}$    *if using RSA*
→

C and S share
secret key material ($secret_c$) at this point

*switch to keys derived*
*from $secret_c$, $N_c$, $N_s$*

*switch to keys derived*
*from $secret_c$, $N_c$, $N_s$*

→

Finished                                    Finished
←

**C**                                         **S**

# SSL/TLS Record Protection



Application Data

Fragment

Compress

Add MAC

Encrypt

Append SSL
Record Header

Use symmetric keys established in the handshake protocol
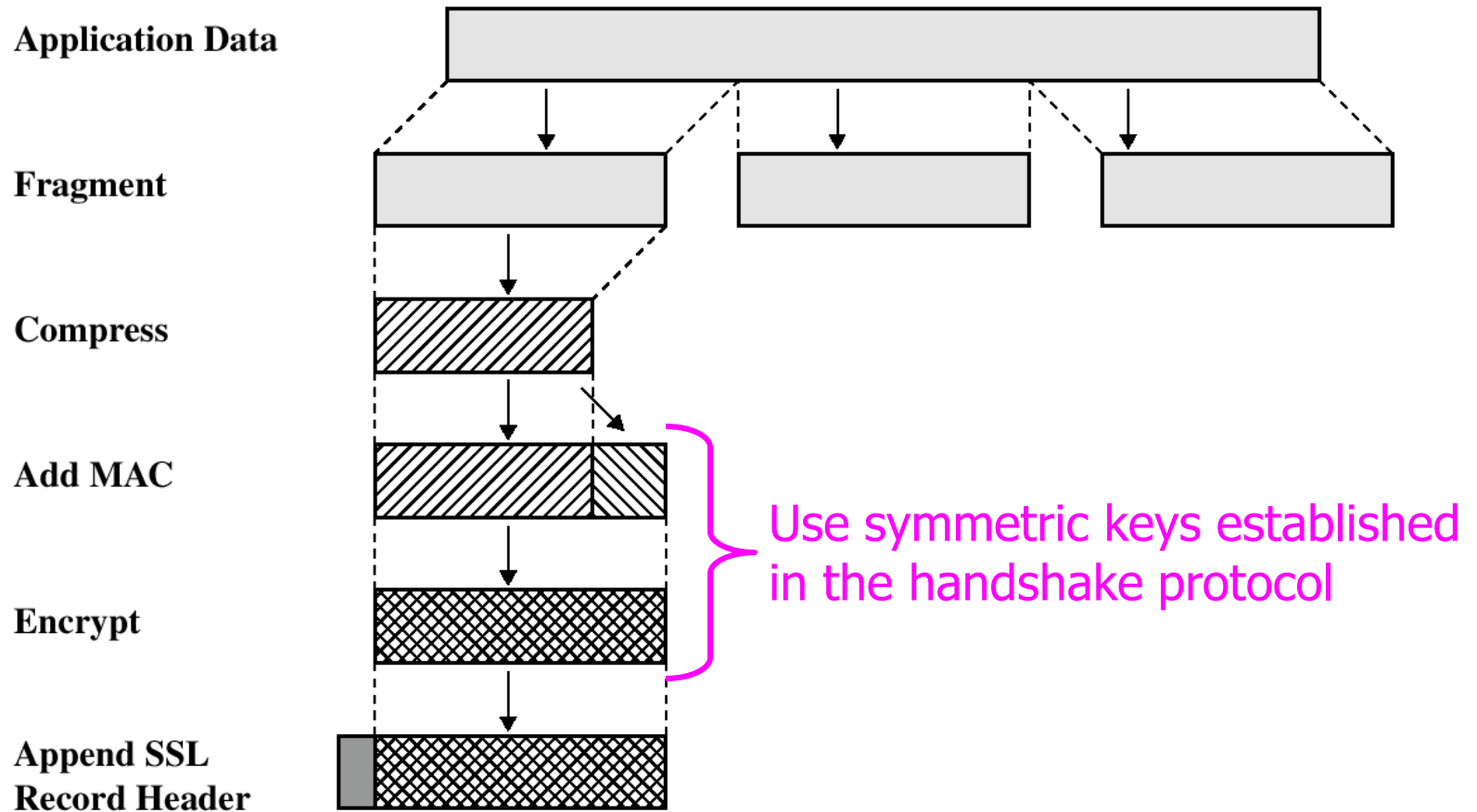
# TLS Heartbeat

A way to keep TLS connection alive
without constantly transferring data

C

If you are alive, send me
this 5-letter word: "xyzzy"

"xyzzy"

S
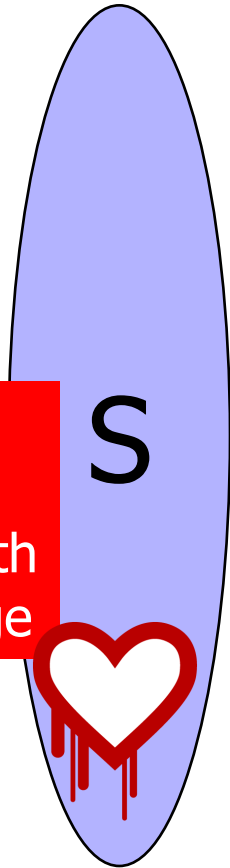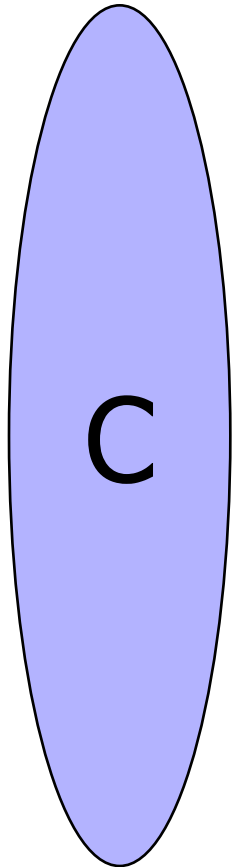
Per RFC 6520:

```
struct {
HeartbeatMessageType type;
uint16 payload_length;
opaque payload[HeartbeatMessage.payload_length];
opaque padding[padding_length];
} HeartbeatMessage;
```

OpenSSL omitted to
check that this value
matches the actual length
of the heartbeat message

# Heartbleed Consequences

◆ Attacker can obtain chunks of server memory

- Passwords, contents of other users' communications, even the server's private RSA key
- Why is the RSA key still in memory?  Long story:

  https://www.lightbluetouchpaper.org/2014/04/25/heartbleed-and-rsa-private-keys/

◆ Assisted by a custom allocator that does not zero out malloc'd memory (for "performance," natch!)

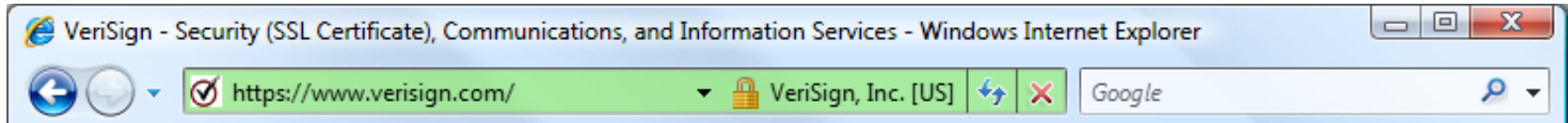# Most Common Use of SSL/TLS

# HTTPS and Its Adversary Model

◆ HTTPS: end-to-end secure protocol for Web

◆ Designed to be secure against network attackers, including man-in-the-middle (MITM) attacks

browser        proxy                    Internet                    HTTPS server

HTTPS tunnel
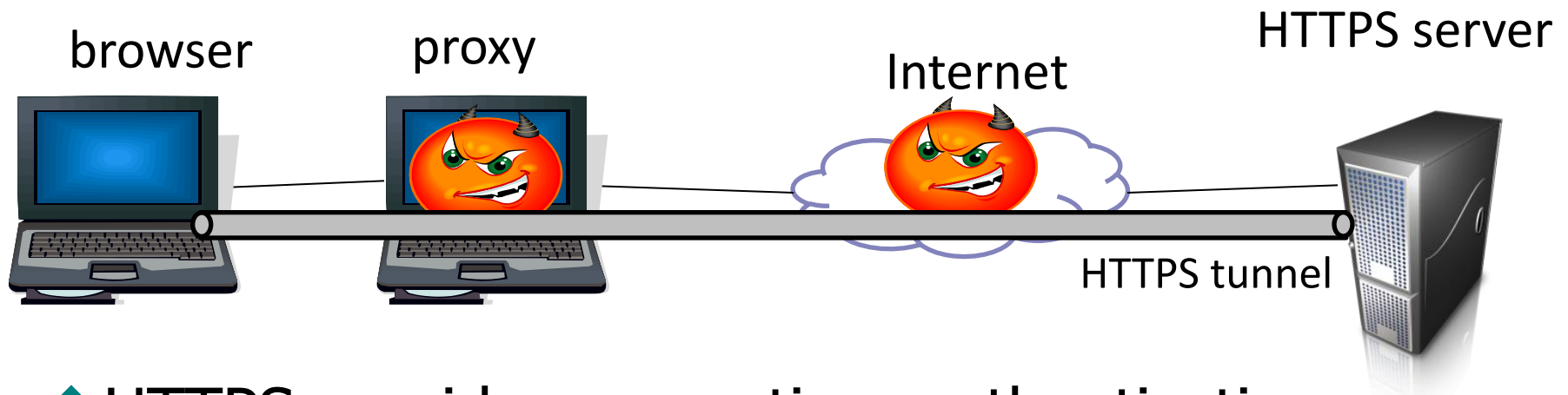
◆ HTTPS provides encryption, authentication (usually for server only), and integrity checking

# The Lock Icon



VeriSign - Security (SSL Certificate), Communications, and Information Services - Windows Internet Explorer
https://www.verisign.com/    VeriSign, Inc. [US]    Google

◆ Goal: identify secure connection

- SSL/TLS is being used between client and server to protect against active network attacker

◆ Lock icon should only be shown when the page is secure against network attacker

- Semantics subtle and not widely understood by users
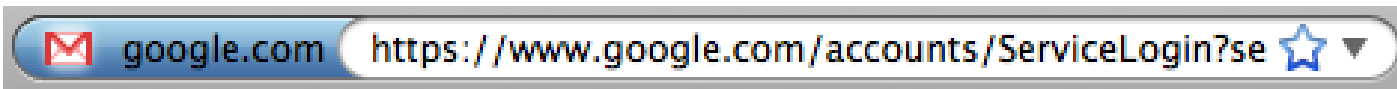- Problem in user interface design

# HTTPS Security Guarantees
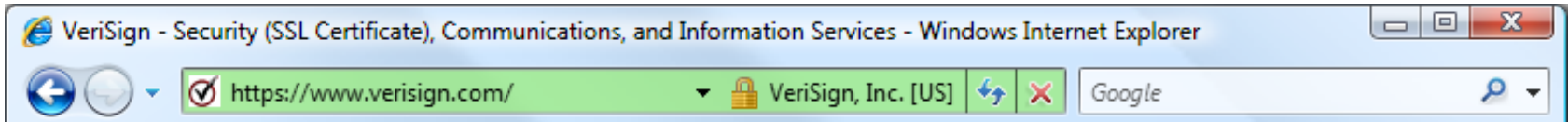


◆ The origin of the page is what it says in the address bar

- User must interpret what he sees

◆ Contents of the page have not been viewed or modified by a network attacker

# Evolution of the Lock in Firefox

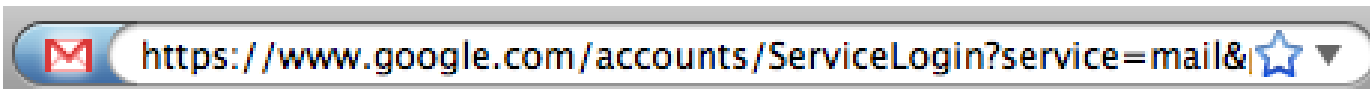How about Firefox 4?

**Firefox 3.6**

M google.com  https://www.google.com/accounts/ServiceLogin?se

*bottom-right corner of browser window (status bar):*

**Firefox 3.0**

M  https://www.google.com/accounts/ServiceLogin?service=mail&

*bottom-right corner of browser window:*  www.google.com

**Firefox 2.0**

M https://www.google.com/accounts/ServiceLogin?service=mail&p:
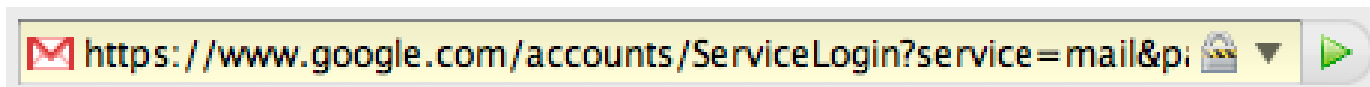
*bottom-right corner of browser window:*  www.google.com

# Combining HTTPS and HTTP

◆ Page served over HTTPS but contains HTTP
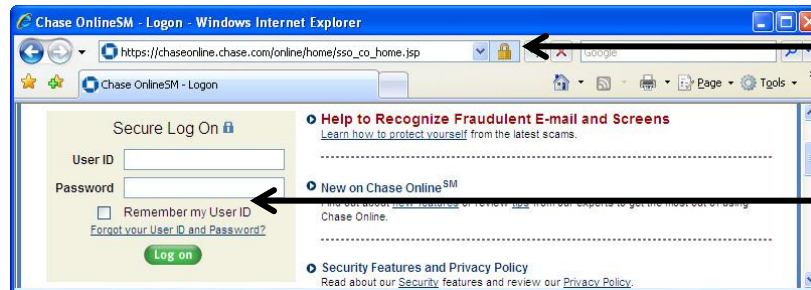
- IE 7: no lock, "mixed content" warning
- Firefox: "!" over lock, no warning by default
- Safari: does not detect mixed content

Lock icon

Flash file served over HTTP

Can script embedding page!

- Flash does not trigger warning in IE7 and FF

◆ Network attacker can now inject scripts, hijack session

# Mixed Content and Network Attacks

◆ Banks: after login, all content served over HTTPS

◆ Developer error: somewhere on bank site write

`<script src=http://www.site.com/script.js> </script>`

- Active network attacker can now inject scripts

◆ Better way to include content:
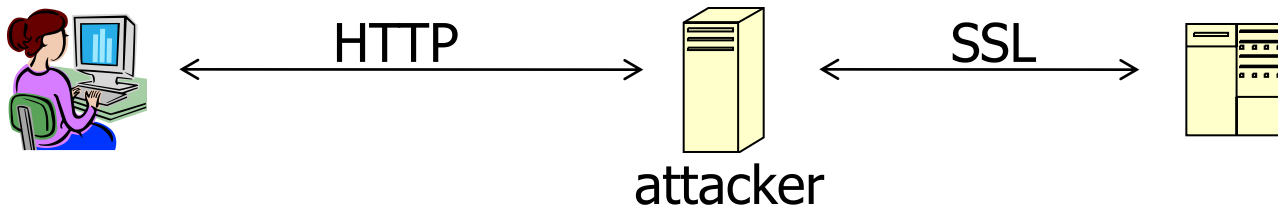
`<script src=//www.site.com/script.js> </script>`

- Served over the same protocol as embedding page

# HTTP → HTTPS and Back

◆ Typical pattern: HTTPS upgrade

- Come to site over HTTP, redirect to HTTPS for login
- Browse site over HTTP, redirect to HTTPS for checkout

◆ sslstrip: network attacker downgrades connection



- Rewrite <a href=https://…>  to  <a href=http://…>
- Redirect Location: https://…  to  Location: http://…
- Rewrite <form action=https://… >
  to <form action=http://…>

Can the server detect this attack?

# Will You Notice?
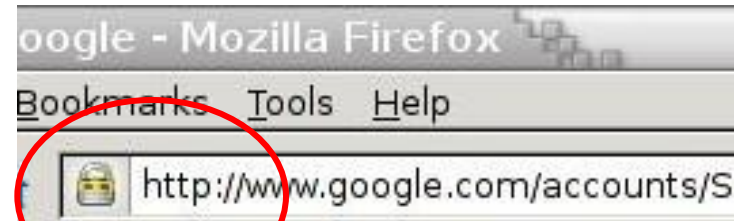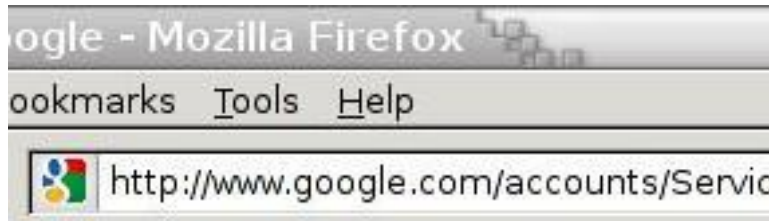
Clever favicon inserted by network attacker

# Motivation

https://



Whose public key is used to establish the secure session?

# Distribution of Public Keys

◆ Public announcement or public directory

- Risks: forgery and tampering

◆ Public-key certificate

- Signed statement specifying the key and identity
  - $sig_{Alice}$("Bob", $PK_B$)

◆ Common approach: certificate authority (CA)

- An agency responsible for certifying public keys
- Browsers are pre-configured with 100+ of trusted CAs
- A public key for any website in the world will be accepted by the browser if certified by one of these CAs

# Trusted Certificate Authorities
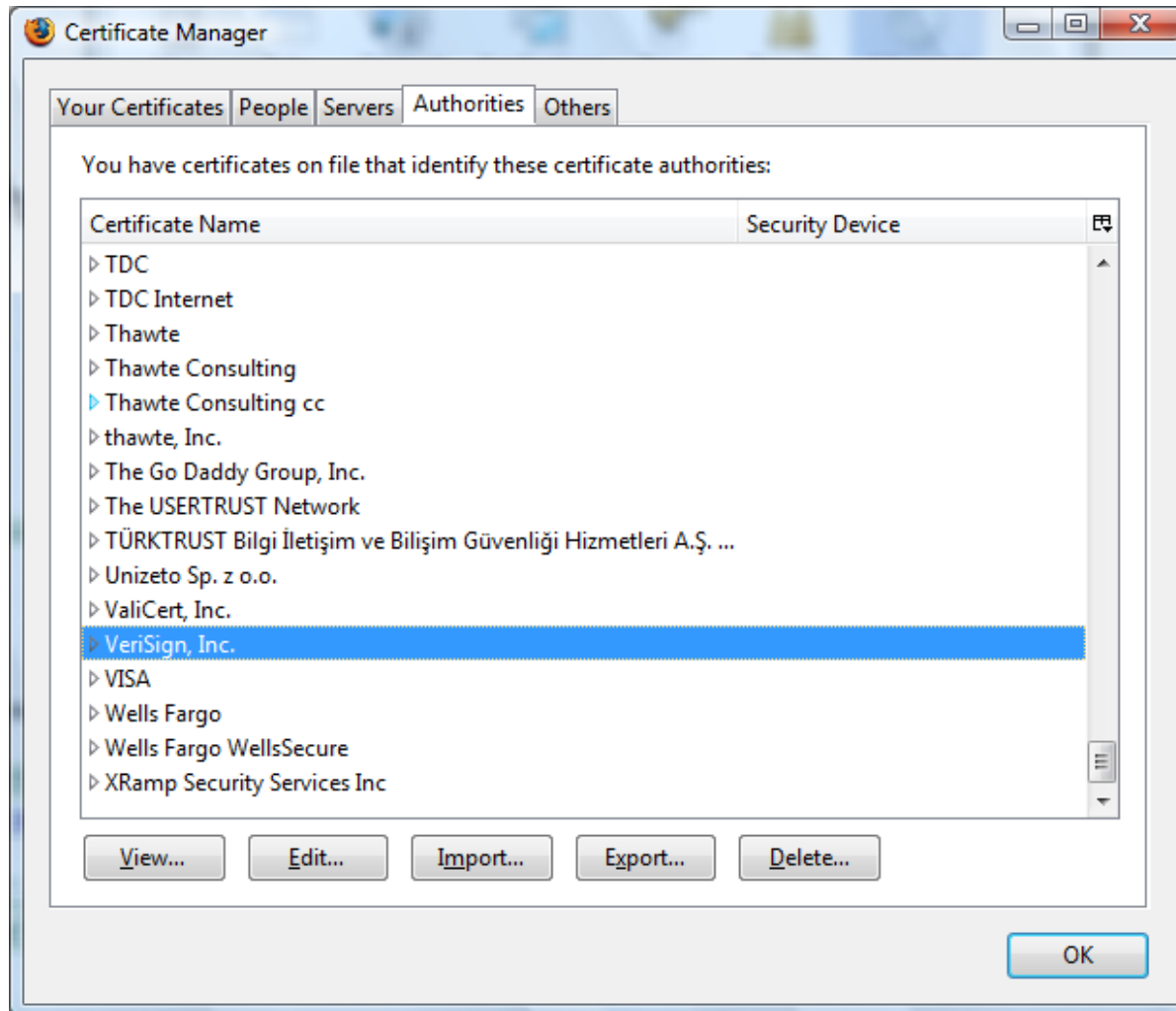
# Example of a Certificate

## Important fields

Certificate Signature Algorithm
Issuer
▲ Validity
    Not Before
    Not After
Subject
▲ Subject Public Key Info
    Subject Public Key Algorithm
    Subject's Public Key
▲ Extensions
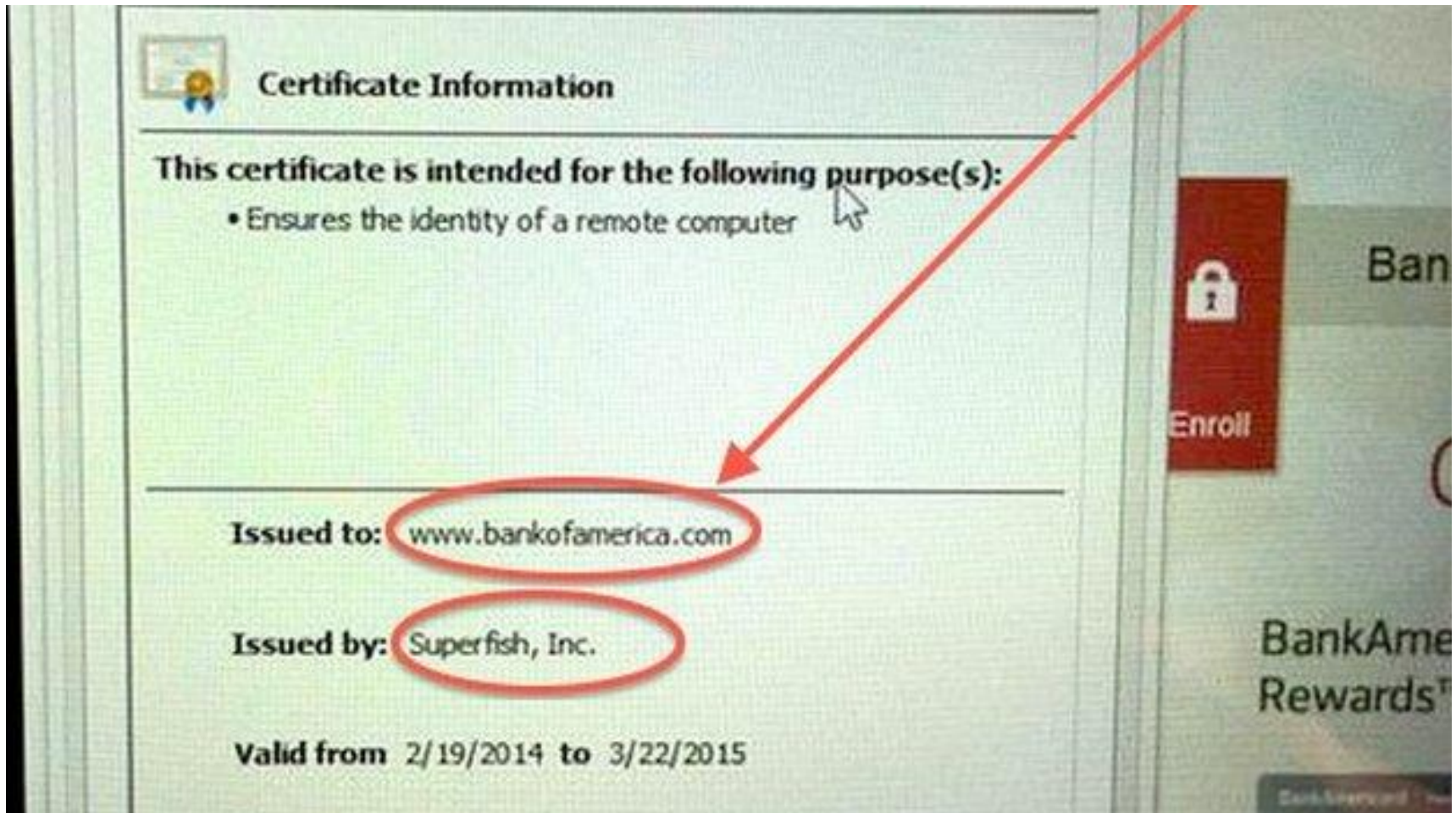
**Field Value**

```
Modulus (1024 bits):
ac 73 14 97 b4 10 a3 aa f4 c1 15 ed cf 92 f3 9a
97 26 9a cf 1b e4 1b dc d2 c9 37 2f d2 e6 07 1d
ad b2 3e f7 8c 2f fa a1 b7 9e e3 54 40 34 3f b9
e2 1c 12 8a 30 6b 0c fa 30 6a 01 61 e9 7c b1 98
2d 0d c6 38 03 b4 55 33 7f 10 40 45 c5 c3 e4 d6
6b 9c 0d d0 8e 4f 39 0d 2b d2 e9 88 cb 2d 21 a3
f1 84 61 3c 3a aa 80 18 27 e6 7e f7 b8 6a 0a 75
e1 bb 14 72 95 cb 64 78 06 84 81 eb 7b 07 8d 49
```

Certificate Viewer:"*.gmail.com"

**General** | Details

This certificate has been verified for the following uses:

SSL Server Certificate

**Issued To**
Common Name (CN)        *.gmail.com
Organization (O)          Google Inc
Organizational Unit (OU)  <Not Part Of Certificate>
Serial Number            65:F8:33:2D:6B:CB:67:BC:AD:3A:B0:A9:98:80:28:49

**Issued By**
Common Name (CN)        Thawte Premium Server CA
Organization (O)          Thawte Consulting cc
Organizational Unit (OU)  Certification Services Division

**Validity**
Issued On                9/25/2008
Expires On               9/25/2010

**Fingerprints**
SHA1 Fingerprint         B7:A7:89:34:54:5D:C9:6F:41:FD:A9:3E:41:AF:2B:1D:13:C8:CC:AD
MD5 Fingerprint          55:5F:09:17:24:03:F7:80:2B:B6:90:26:3B:0B:E3:3B

# Another Example of a Certificate

# Root Certificates in Lenovo



In the news

**Lenovo hit by lawsuit over Superfish adware**
CNET - 3 days ago
Sarah Tew/CNET. **Lenovo** may find itself in a courtroom over its **Superfish adware** fiasco.

Interview with Lenovo's CTO will scare anyone still thinking of buying a Lenovo product
BGR - 2 days ago

Lenovo's Chief Technology Officer Discusses the Superfish Adware Fiasco - NYTimes.com
Bits - The New York Times - 3 days ago

More news for lenovo superfish adware

**Lenovo Sued Over Superfish Adware : NPR**
www.npr.org › News › Business    NPR
2 days ago - Renee Montagne talks to Jordan Robertson of Bloomberg News about computer maker **Lenovo**, which allowed controversial spyware to be ...

**Lenovo users lawyer up over hole-filled, HTTPS-breaking ...**
arstechnica.com/.../**lenovo**-users-lawyer-up-over-hole-filled...    Ars Technica
4 days ago - In the wake of last week's **Lenovo's Superfish** debacle, at least one ... and that **Superfish adware** "does not present a security risk," despite ...

**Lenovo's Chief Technology Officer Discusses the Superfish ...**
bits.blogs.nytimes.com/.../**lenovos**-chief-technology-officer-discusses-the...
3 days ago - The **adware** was intended to serve **Lenovo** users targeted ads, but the company **Lenovo** partnered with to do this, **Superfish**, did so by hijacking ...

**Lenovo Sued Over Superfish Adware | News & Opinion ...**
www.pcmag.com › Reviews › Software › Security    PC Magazine
4 days ago - Not surprisingly, the controversy over **Lenovo** installing **Superfish adware** into its consumer PCs has resulted in a lawsuit. According to the suit, ...

# CA Hierarchy

◆ Browsers, operating systems, etc. have trusted root certificate authorities

- Firefox 3 includes certificates of 135 trusted root CAs

◆ A Root CA signs certificates for intermediate CAs, they sign certificates for lower-level CAs, etc.

- Certificate "chain of trust"

    – $sig_{Verisign}$("UT Austin", $PK_{UT}$), $sig_{UT}$("Vitaly S.", $PK_{Vitaly}$)

◆ CA is responsible for verifying the identities of certificate requestors, domain ownership

# Certificate Hierarchy



root CA

What power do they have?

intermediate CA's

Who trusts their certificates?

users
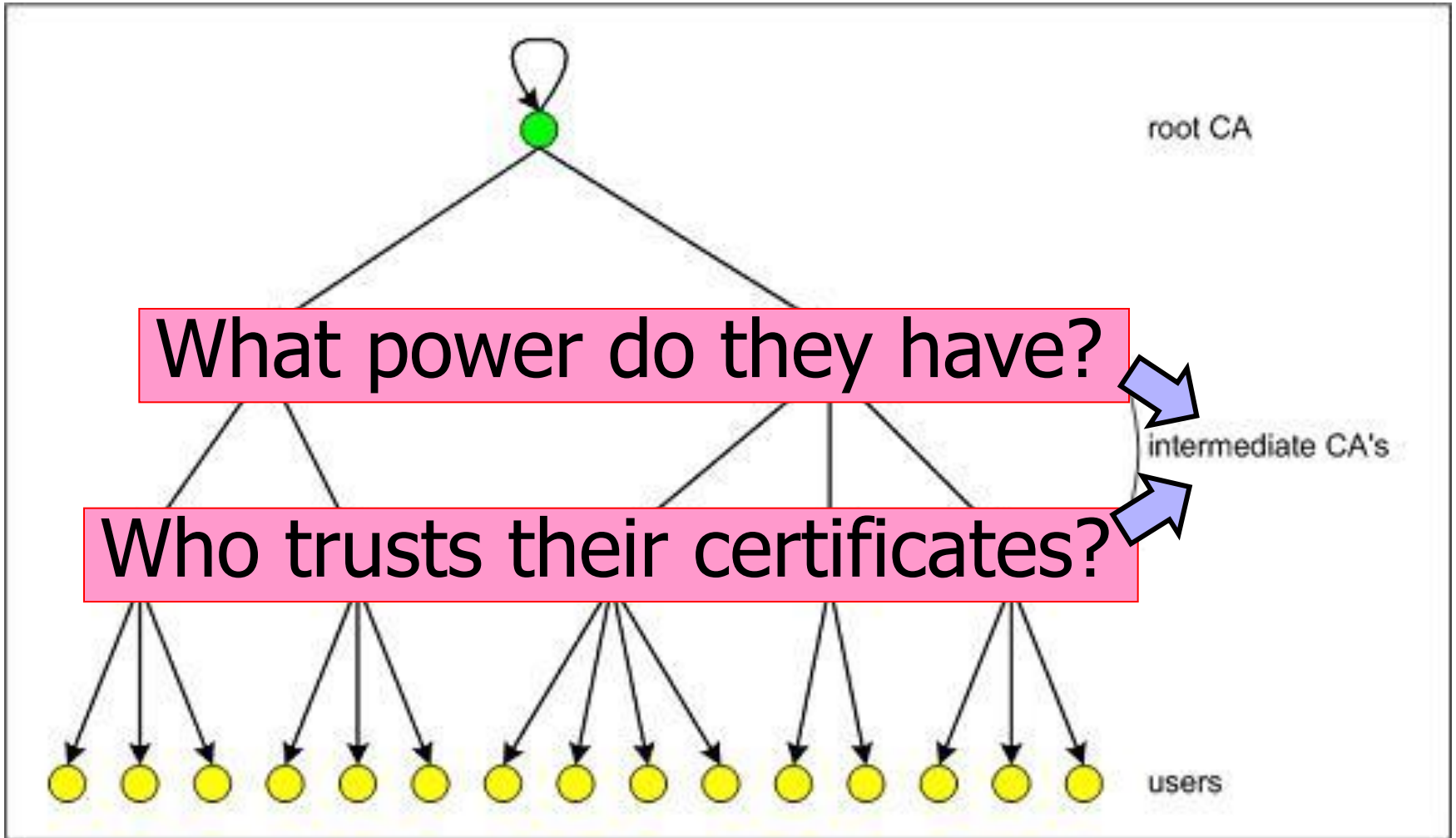
# Common Name

◆ Explicit name: www.foo.com

◆ Wildcard: *.foo.com or www*.foo.com

◆ Matching rules

- Firefox 3: * matches anything
- Internet Explorer 7: * must occur in the leftmost component, does not match '.'
  - *.foo.com matches a.foo.com, but not a.b.foo.com

# International Domain Names

◆Rendered using international character set

◆Chinese character set contains characters that look like / ? = .

- What could go wrong?

◆Can buy a certificate for *.foo.cn, create any number of domain names that look like

www.bank.com/accounts/login.php?q=me.foo.cn

- What does the user see?
- *.foo.cn certificate works for all of them!

# Example

[Moxie Marlinspike]

# Meaning of Color

**Internet Explorer 9**

http://www.google.com/

https://mail.google.com/mail/?shva=1#inbox

https://www.bankofamerica.com/p    Bank of America Corporati...

**Firefox 4**

http://www.google.com/

google.com  https://mail.google.com/mail/?shva=1#inbox

Bank of America Corporation (US)  https://www.bankofamerica.com

**Chrome 8**

www.google.com

https://mail.google.com/mail/?shva=1#inbox

Bank of America Corporation [US]  https://www.bankofamerica.com/privacy

**Safari 4**

http://www.google.com/

https://mail.google.com/mail/?shva=1#inbox    RSS

top right corner:

https://www.bankofamerica.com/    Bank of America Cor...

top right corner:

## What is the difference?

Domain Validation (DV) certificate

vs.

Extended Validation (EV) certificate

## Means what?

slide 35

# Extended Validation (EV) Certificates

◆ Certificate request must be approved by a human lawyer at the certificate authority
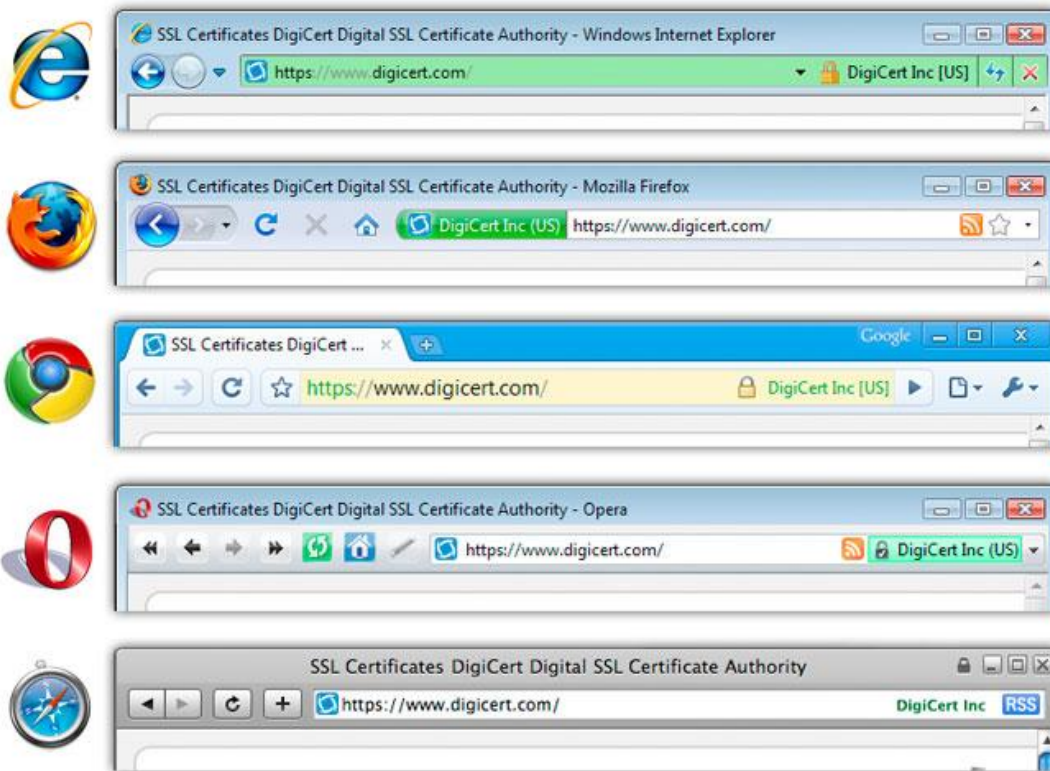
# Questions about EV Certificates

◆ What does EV certificate mean?

◆ What is the difference between an HTTPS connection that uses a regular certificate and an HTTPS connection that uses an EV certificate?

◆ If an attacker has somehow obtained a non-EV certificate for bank.com, can he inject a script into https://bank.com content?

- What is the origin of the script? Can it access or modify content that arrived from actual bank.com via HTTPS?

◆ What would the browser show – blue or green?

# X.509 Authentication Service

◆Internet standard (1988-2000)

◆Specifies certificate format

- X.509 certificates are used in IPsec and SSL/TLS

◆Specifies certificate directory service

- For retrieving other users' CA-certified public keys

◆Specifies a set of authentication protocols

- For proving identity using public-key signatures

◆Can use with any digital signature scheme and hash function, but must hash before signing

# X.509 Certificate
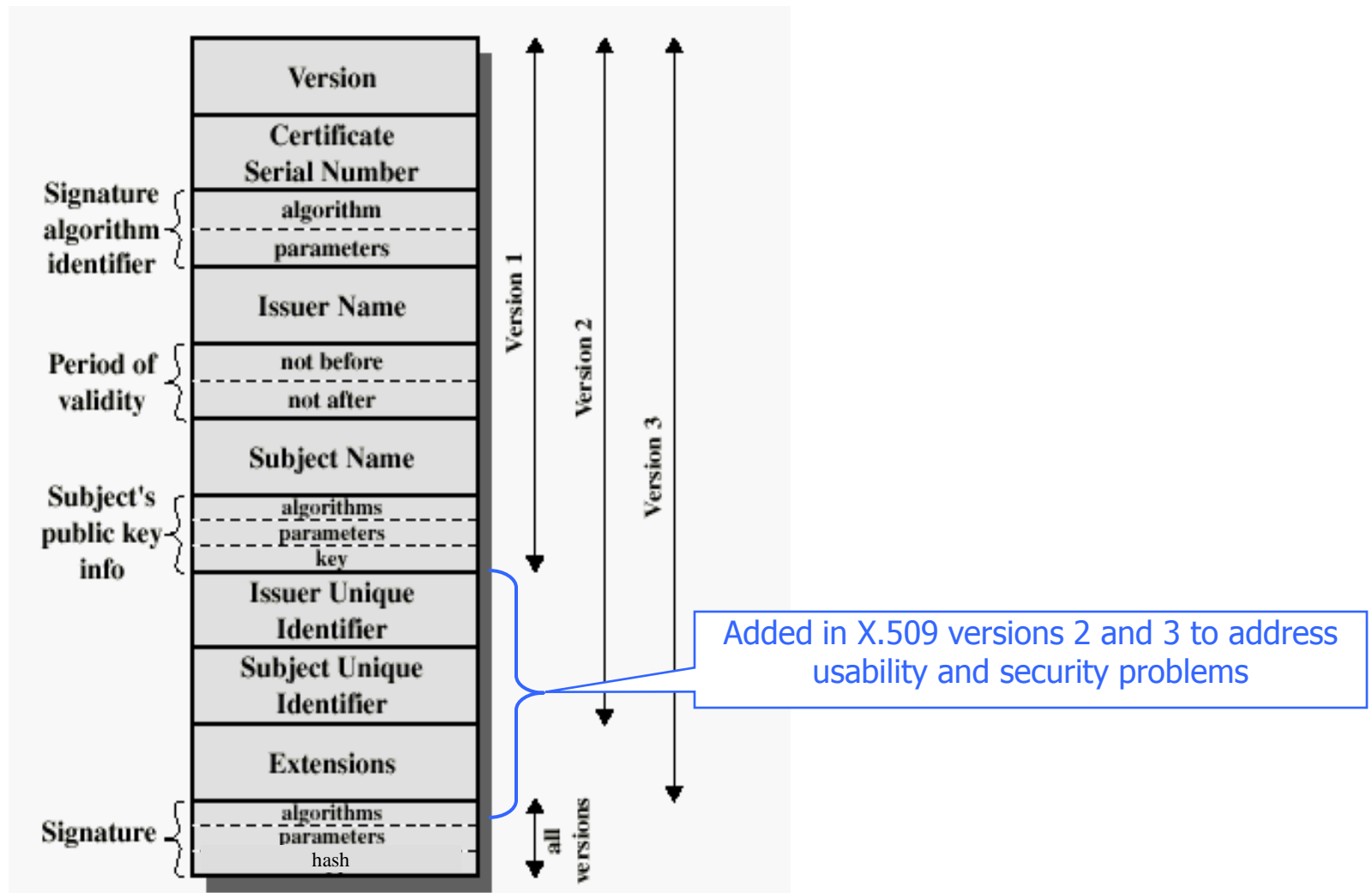


Added in X.509 versions 2 and 3 to address usability and security problems

# Back in 2008

◆ Many CAs still used MD5

- RapidSSL, FreeSSL, TrustCenter, RSA Data Security, Thawte, verisign.co.jp

◆ Sotirov et al. collected 30,000 website certificates

◆ 9,000 of them were signed using MD5 hash

◆ 97% of those were issued by RapidSSL

# Colliding Certificates

| set by the CA | | chosen prefix (difference) | |
|---|---|---|---|
| | serial number | | serial number |
| | validity period | | validity period |
| | real cert domain name | | rogue cert domain name + key |
| | real cert RSA key | Hash to the same MD5 value! | ??? |
| | | collision bits (computed) | |
| Valid for both certificates! | X.509 extensions | identical bytes (copied from real cert) | X.509 extensions |
| | signature | | signature |

slide 42
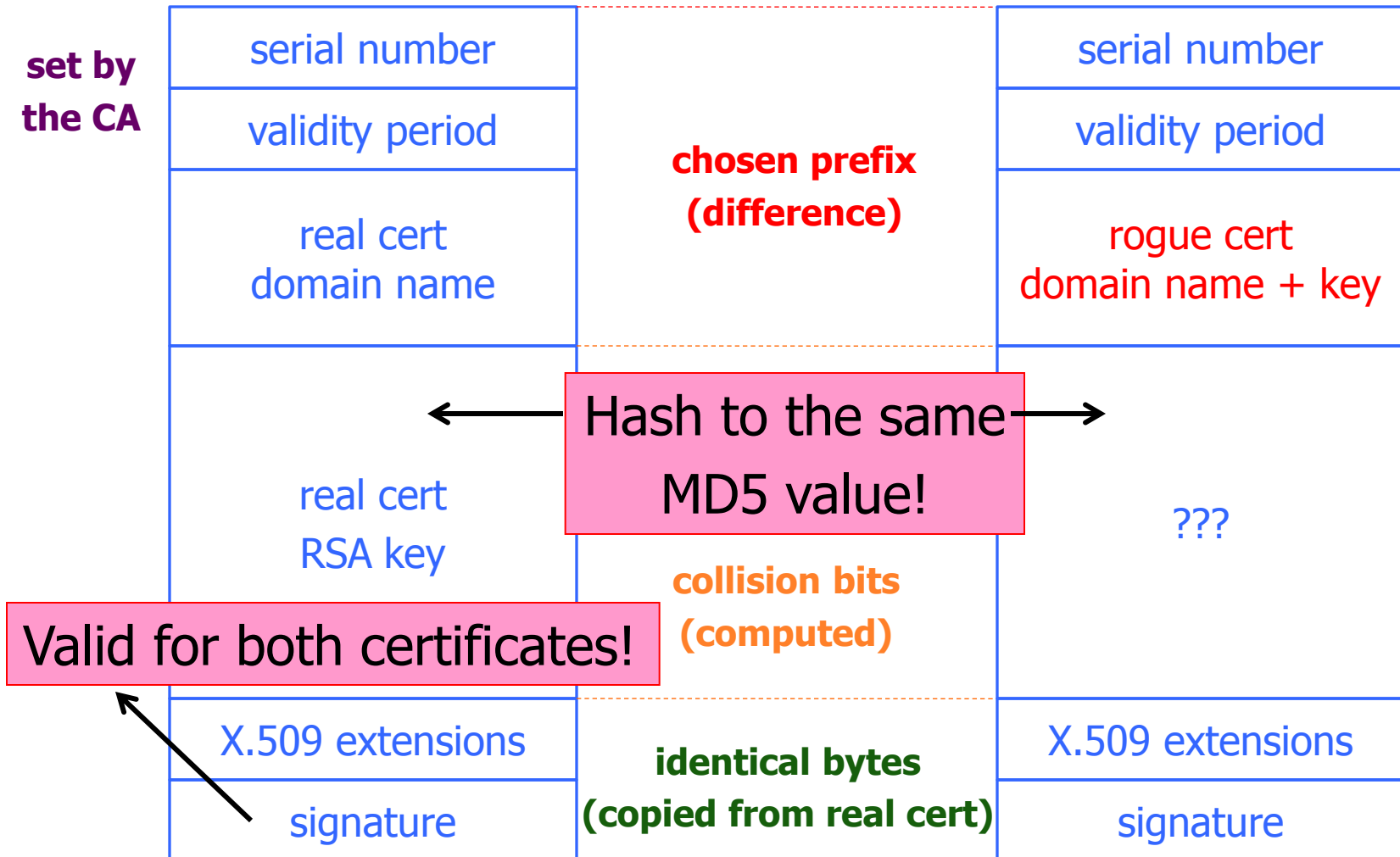
# Generating Collisions

1-2 days on a cluster of 200 PlayStation 3's

Equivalent to 8000 desktop CPU cores or $20,000 on Amazon EC2

# Generating Colliding Certificates

◆ RapidSSL uses a fully automated system

- $69 for a certificate, issued in 6 seconds
- Sequential serial numbers

◆ Technique for generating colliding certificates

- Get a certificate with serial number S
- Predict time T when RapidSSL's counter goes to S+1000
- Generate the collision part of the certificate
- Shortly before time T buy enough (non-colliding) certificates to increment the counter to S+999
- Send colliding request at time T and get serial number S+1000

# Creating a Fake Intermediate CA

[Sotirov et al.]

| real cert | chosen prefix (difference) | rogue CA cert |
|---|---|---|
| serial number | | |
| validity period | | rogue CA cert |
| real cert domain name | | rogue CA RSA key |
| | | rogue CA X.509 extensions |
| real cert RSA key | collision bits (computed) | Netscape Comment Extension (contents ignored by browsers) |
| X.509 extensions | identical bytes (copied from real cert) | |
| signature | | signature |

← **CA bit!**

We are now an intermediate CA. W00T!

# Result: Perfect Man-in-the-Middle

◆This is a "skeleton key" certificate: it can issue fully trusted certificates for <u>any</u> site (why?)

Certificate

| General | Details | Certification Path |

Certification path

Equifax Secure Global eBusiness CA-1
    MD5 Collisions Inc. (http://www.phreedom.org/md5)
        127.0.0.1
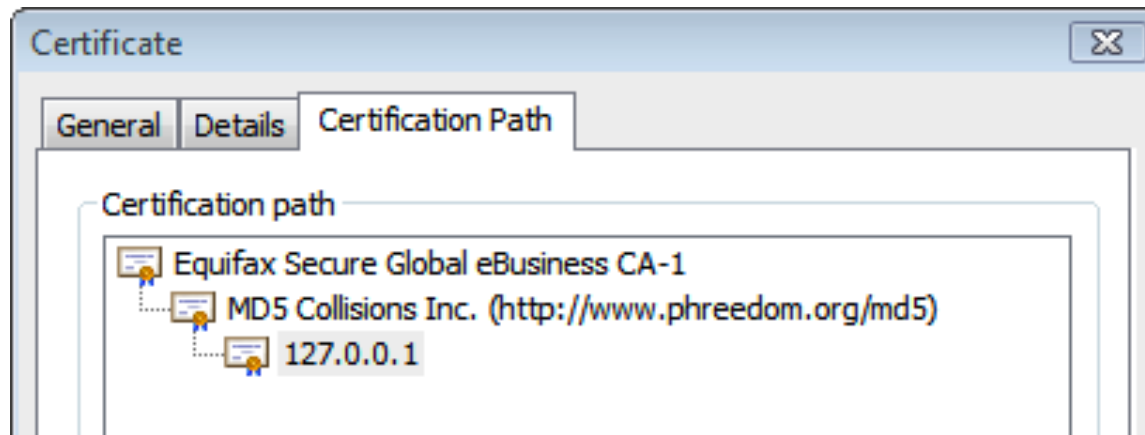
◆To take advantage, need a network attack
- Insecure wireless, DNS poisoning, proxy auto-discovery, hacked routers, etc.

# A Rogue Certificate
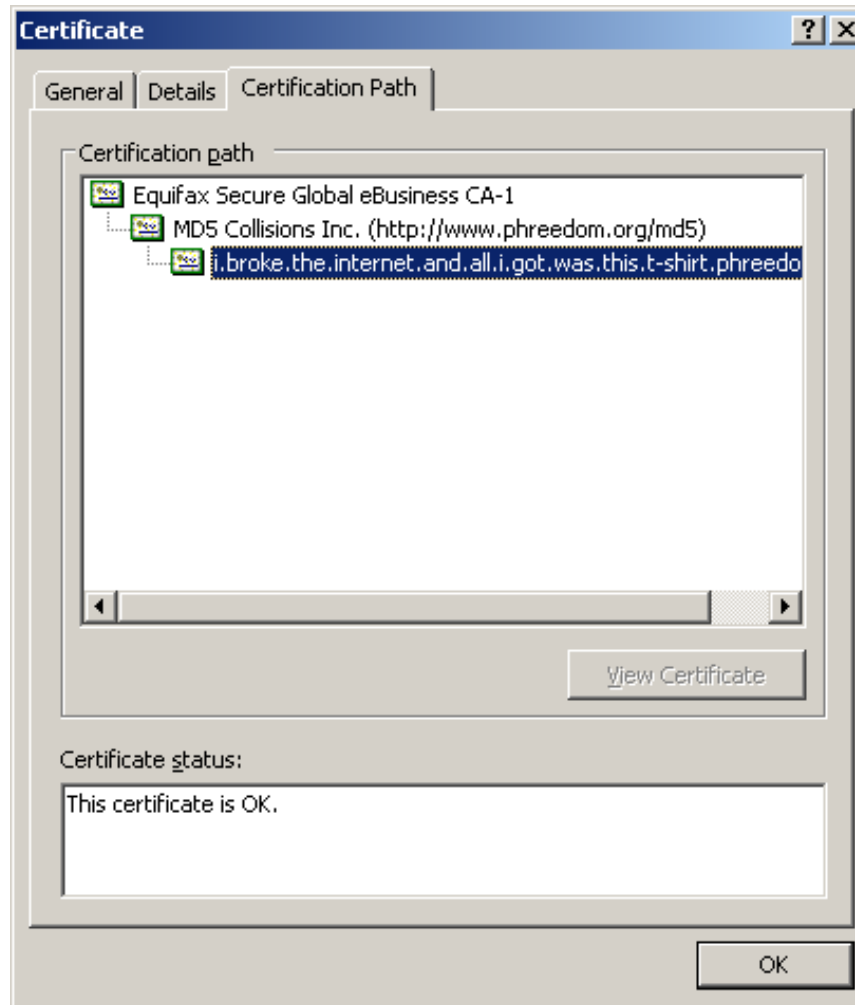
# Flame

◆ Cyber-espionage virus (2010-2012)

◆ Signed with a fake intermediate CA certificate accepted by any Windows Update service

- Fake intermediate CA certificate was created using an MD5 chosen-prefix collision against an obscure Microsoft Terminal Server Licensing Service certificate that was enabled for code signing and still used MD5

◆ MD5 collision technique possibly pre-dates Sotirov et al.'s work

- Evidence of state-level cryptanalysis?
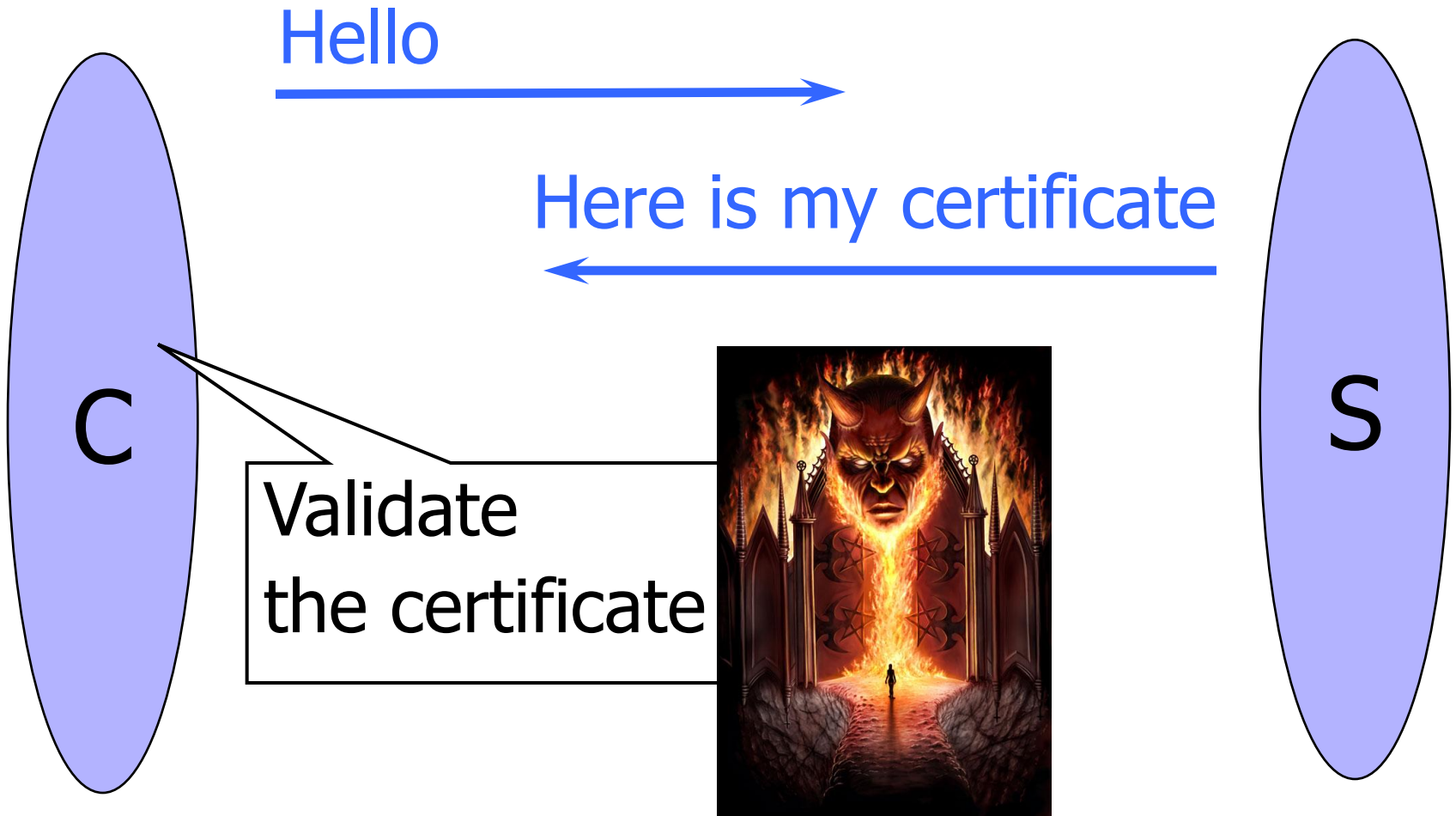
# SSL/TLS Handshake

Hello →

Here is my certificate

C

S

Validate the certificate

# SSL/TLS Handshake

# Failing to Check Hostname
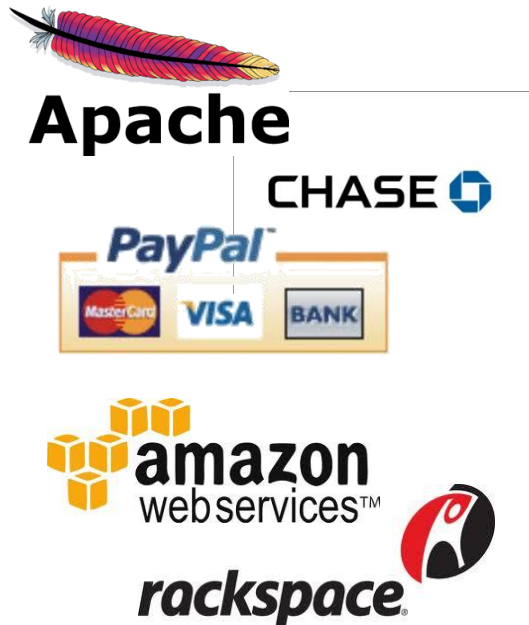
"Researchers at the University of Texas at Austin and Stanford University have discovered that poorly designed APIs used in SSL implementations are to blame for vulnerabilities in many critical non-browser software packages. Serious security vulnerabilities were found in programs such as Amazon's EC2 Java library, Amazon's and PayPal's merchant SDKs, Trillian and AIM instant messaging software, popular integrated shopping cart software packages, Chase mobile banking software, and several Android applications and libraries. SSL connections from these programs and many others are vulnerable to a man in the middle attack…"

- Threatpost (Oct 2012)

Major payment processing gateways, client software for cloud computing, integrated e-commerce software, etc.

# What Happens After Validation?

Hello

I am PayPal.com

(or whoever you want me to be)

Here is PayPal's certificate for

its RSA signing key

And here is my signed Diffie-Hellman value

Validate the certificate

... then verify the signature on the DH value using the public key from the certificate
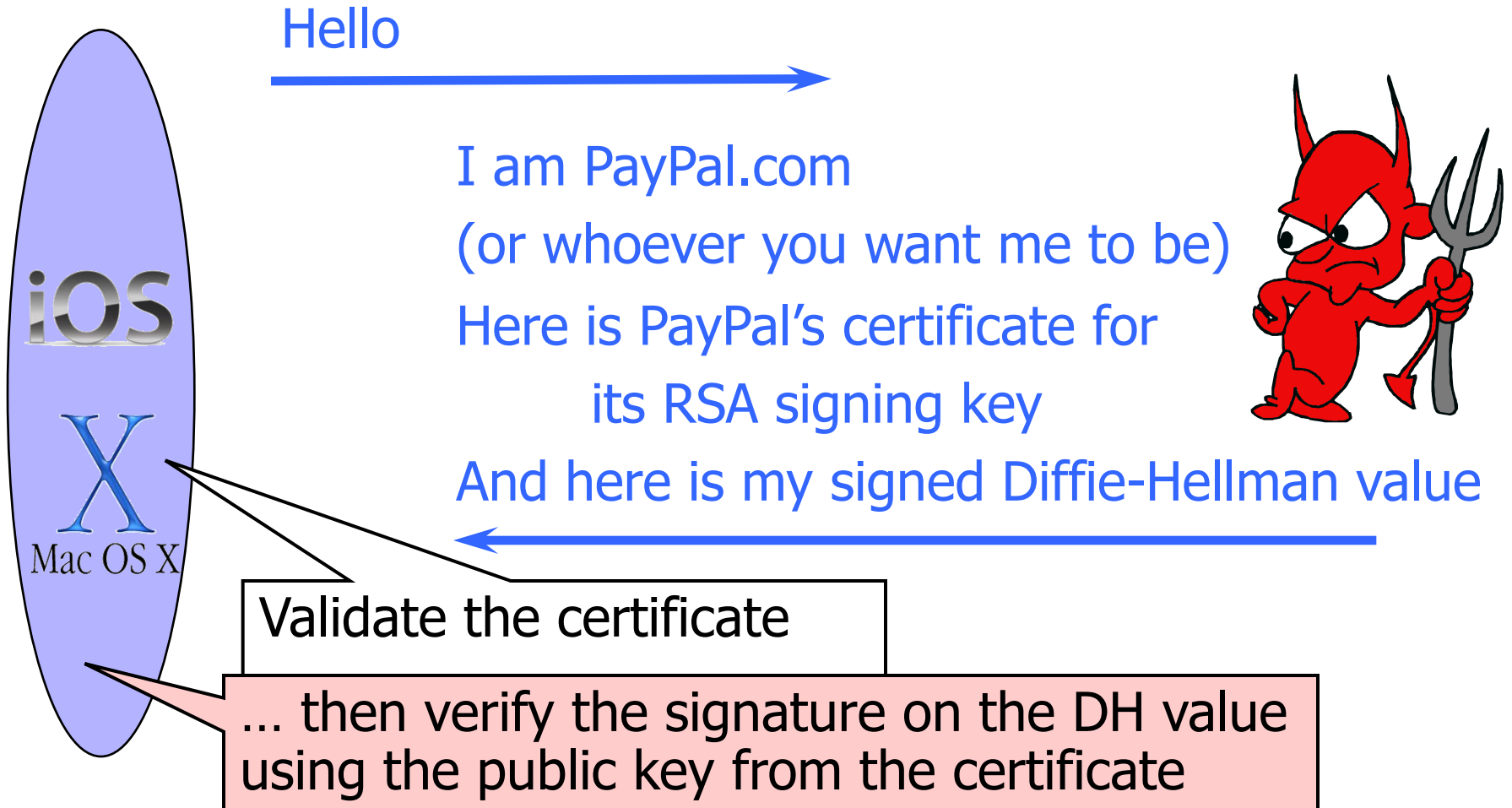
# Goto Fail

Here is PayPal's certificate
And here is my signed Diffie-Hellman value

… verify the signature on the DH value using the public key from the certificate
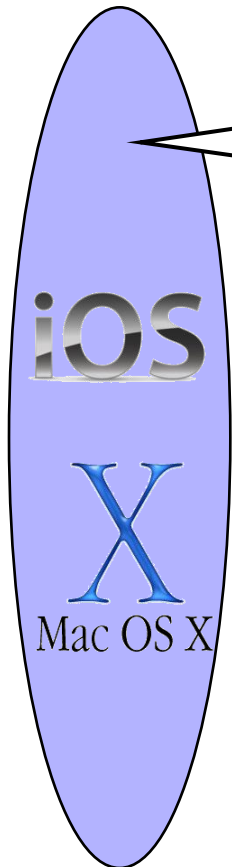
```
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;      ??? 
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail; …
err = sslRawVerify(…);        Signature is verified here

…

fail: … return err …
```

# Complete Fail Against MITM

◆ Discovered in February 2014

◆ All OS X and iOS software vulnerable to man-in-the-middle attacks

- Broken TLS implementation provides no protection against the very attack it was supposed to prevent

◆ What does this tell you about quality control for security-critical software?

```
goto fail;
goto fail;
```

# Certificate Revocation

◆ Revocation is <u>very</u> important

◆ Many valid reasons to revoke a certificate

- Private key corresponding to the certified public key has been compromised
- User stopped paying his certification fee to the CA and the CA no longer wishes to certify him
- CA has been compromised

◆ Expiration is a form of revocation, too

- Many deployed systems don't bother with revocation
- Re-issuance of certificates is a big revenue source for certificate authorities

# Certificate Revocation Mechanisms

◆ Online revocation service

- When a certificate is presented, recipient goes to a special online service to verify whether it is still valid

◆ Certificate revocation list (CRL)

- CA periodically issues a signed list of revoked certificates
- Can issue a "delta CRL" containing only updates

Q: Does revocation protect against forged certificates?

# Comodo



◆ Comodo is one of the trusted root CAs

- Its certificates for any website in the world are accepted by every browser

◆ Comodo accepts certificate orders submitted through resellers

- Reseller uses a program to authenticate to Comodo and submit an order with a domain name and public key, Comodo automatically issues a certificate for this site

# Comodo Break-In

- ◆ An Iranian hacker broke into instantSSL.it and globalTrust.it resellers, decompiled their certificate issuance program, learned the credentials of their reseller account and how to use Comodo API
  - username: gtadmin, password: globaltrust
- ◆ Wrote his own program for submitting orders and obtaining Comodo certificates
- ◆ On March 15, 2011, got Comodo to issue 9 rogue certificates for popular sites
  - mail.google.com, login.live.com, login.yahoo.com, login.skype.com, addons.mozilla.org, "global trustee"

# Consequences

◆ Attacker needs to first divert users to an attacker-controlled site instead of Google, Yahoo, Skype, but then…

- For example, use DNS to poison the mapping of mail.yahoo.com to an IP address

◆ … "authenticate" as the real site

◆ … decrypt all data sent by users

- Email, phone conversations, Web browsing

Q: Does HTTPS help?  How about EV certificates?

# Message from the Attacker

http://pastebin.com/74KXCaEZ

I'm single hacker with experience of 1000 hacker,  I'm single programmer with experience of 1000 programmer, I'm single planner/project manager with experience of 1000 project managers …

When USA and Isarel could read my emails in Yahoo, Hotmail, Skype, Gmail, etc. without any simple little problem, when they can spy using Echelon, I can do anything I can. It's a simple rule. You do, I do, that's all. You stop, I stop. It's rule #1 …

Rule#2: So why all the world got worried, internet shocked and all writers write about it, but nobody writes about Stuxnet anymore?... So nobody should write about SSL certificates.

Rule#3: I won't let anyone inside Iran, harm people of Iran, harm my country's Nuclear Scientists, harm my Leader (which nobody can), harm my President, as I live, you won't be able to do so. as I live, you don't have privacy in internet, you don't have security in digital world, just wait and see...

# DigiNotar Break-In

◆ In June 2011, the same "ComodoHacker" broke into a Dutch certificate authority, DigiNotar

- Message found in scripts used to generate fake certificates:

  "THERE IS NO ANY HARDWARE OR SOFTWARE IN THIS WORLD EXISTS WHICH COULD STOP MY HEAVY ATTACKS MY BRAIN OR MY SKILLS OR MY WILL OR MY EXPERTISE"

◆ Security of DigiNotar servers

- All core certificate servers in a single Windows domain, controlled by a single admin password (Pr0d@dm1n)
- Software on public-facing servers out of date, unpatched
- Tools used in the attack would have been easily detected by an antivirus… if it had been present

# Consequences of DigiNotar Hack

◆ Break-in not detected for a month

◆ Rogue certificates issued for *.google.com, Skype, Facebook, www.cia.gov, and 527 other domains

◆ 99% of revocation lookups for these certificates originated from Iran

- Evidence that rogue certificates were being used, most likely by Iranian government or Iranian ISPs to intercept encrypted communications
    - Textbook man-in-the-middle attack
- 300,000 users were served rogue certificates

# Another Message from the Attacker

Most sophisticated hack of all time … I'm really sharp, powerful, dangerous and smart!

My country should have control over Google, Skype, Yahoo, etc. […] I'm breaking all encryption algorithms and giving power to my country to control all of them.

You only heards Comodo (successfully issued 9 certs for me -thanks by the way-), DigiNotar (successfully generated 500+ code signing and SSL certs for me -thanks again-), StartCOM (got connection to HSM, was generating for twitter, google, etc. CEO was lucky enough, but I have ALL emails, database backups, customer data which I'll publish all via cryptome in near future), GlobalSign (I have access to their entire server, got DB backups, their linux / tar gzipped and downloaded, I even have private key of their OWN globalsign.com domain, hahahaa)…. BUT YOU HAVE TO HEAR SO MUCH MORE! SO MUCH MORE! At least 3 more. AT LEAST!

# TurkTrust

◆ In Jan 2013, a rogue *.google.com certificate was issued by an intermediate CA that gained its authority from the Turkish root CA TurkTrust

- TurkTrust accidentally issued intermediate CA certs to customers who requested regular certificates
- Ankara transit authority used its certificate to issue a fake *.google.com certificate in order to intercept and filter SSL traffic from its network

◆ This rogue *.google.com certificate was trusted by every browser in the world

# TrustWave

- In Feb 2012, admitted issuing an intermediate CA certificate to a corporate customer
  - Purpose: "re-sign" certificates for "data loss prevention"
  - Translation: forge certificates of third-party sites in order to spy on employees' encrypted communications with the outside world
- Customer can now forge certificates for any site in world… and they will be accepted by any browser!
  - What if a "re-signed" certificate leaks out?
- Do other CAs do this?

# Komodia

◆ Israeli startup

◆ From their website: "Our advanced SSL hijacker SDK is a brand new technology that allows you to access data that was encrypted using SSL and perform on the fly SSL decryption."

- Installs its own root certificate
- Goal: re-sign SSL certificates, proxy/MITM connections

◆ Same private key on all machines, easily extracted

- Anyone can issue fake Komodia certificates, do man-in-the-middle attacks on any machine with Komodia

# It Gets Worse

https://blog.filippo.io/komodia-superfish-ssl-validation-is-broken/

◆ What happens if a MITM attacker serves a self-signed certificate to a Komodia client?

◆ Komodia re-signs and turns it into a trusted certificate

  • But it will also change the name in the certificate, which won't match what the browser is expecting and user will see a warning  -  maybe not so bad

◆ But if attacker puts target domain into "alternate name" field, Komodia won't touch it and browser will think the certificate is completely valid

# Complete SSL Fail

https://blog.filippo.io/komodia-superfish-ssl-validation-is-broken/
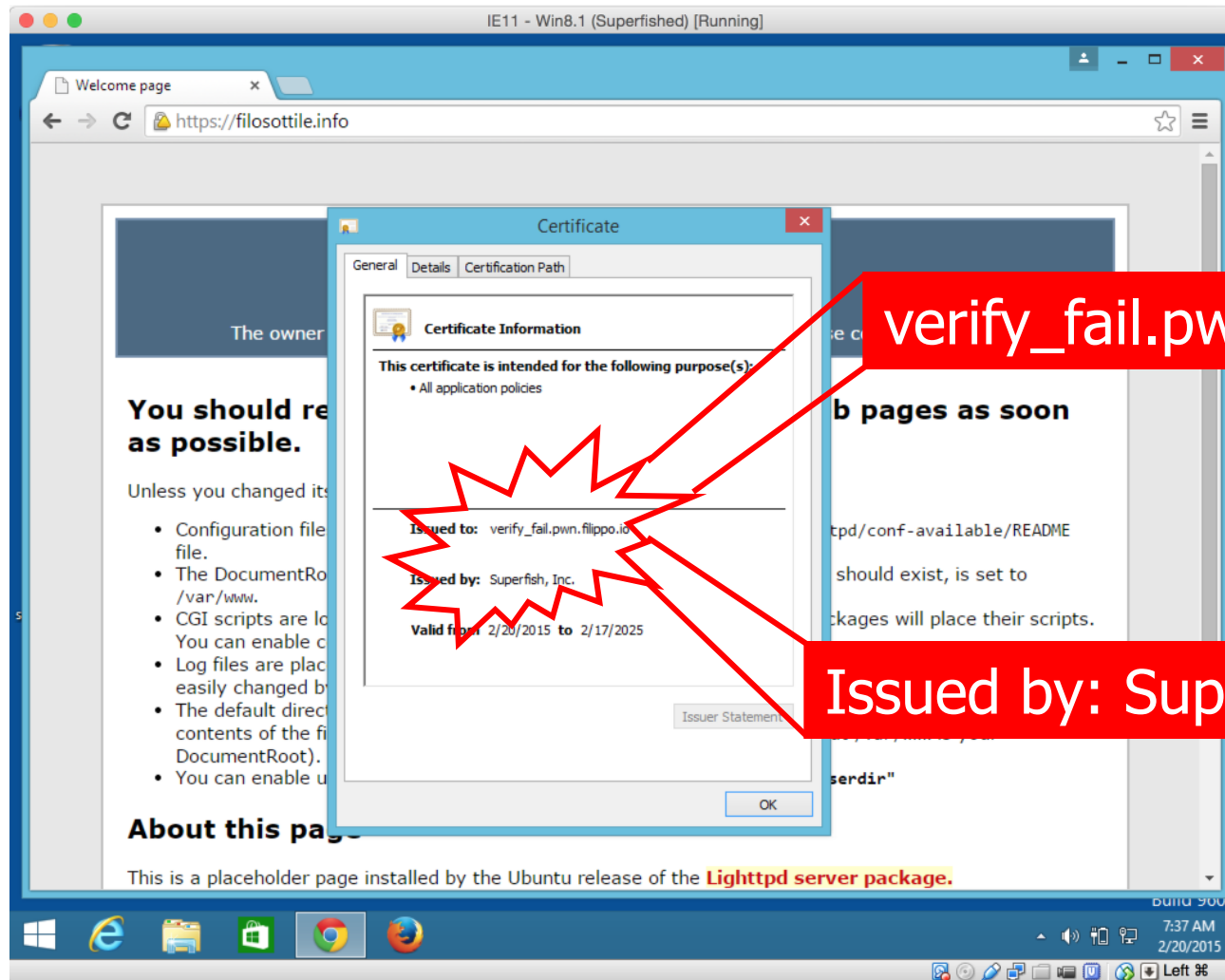
# Software based on Komodia SDK

- Superfish
- CartCrunch Israel LTD
- WiredTools LTD
- Say Media Group LTD
- Over the Rainbow Tech
- System Alerts
- ArcadeGiant
- Objectify Media Inc
- Catalytix Web Services
- OptimizerMonitor

# Statement from Superfish CEO

There has been significant misinformation circulating about Superfish software that was pre-installed on certain Lenovo laptops. The software shipped on a limited number of computers in 2014 in an effort to enhance the online shopping experience for Lenovo customers. Superfish's software utilizes visual search technology to help users achieve more relevant search results based on images of products they have browsed.

Despite the false and misleading statements made by some media commentators and bloggers, the Superfish software does not present a security risk. In no way does Superfish store personal data or share such data with anyone. Unfortunately, in this situation a vulnerability was introduced unintentionally by a 3rd party. Both Lenovo and Superfish did extensive testing of the solution but this issue wasn't identified before some laptops shipped. Fortunately, our partnership with Lenovo was limited in scale. We were able to address the issue quickly. The software was disabled on the server side (i.e., Superfish's search engine) in January 2015.

# Not Just Komodia

◆ PrivDog

- "Your privacy is under attack!"

◆ Provides "private Web browsing"

- Translation: replaces ads on webpages with other ads from "trusted sources"

◆ Re-signs certificates to MITM SSL connections

◆ Accepts self-signed certificates and turns them into trusted certificates

◆ Founded by the CEO of Comodo CA