IERG4210 Web Programming and Security

Course Website: https://course.ie.cuhk.edu.hk/~ierg4210/
Live FB Feedback Group: https://fb.com/groups/ierg4210.2015spring/

Transport Layer and Browser Security
Lecture 10

Dr. Adonis Fung
phfung@ie.cuhk.edu.hk

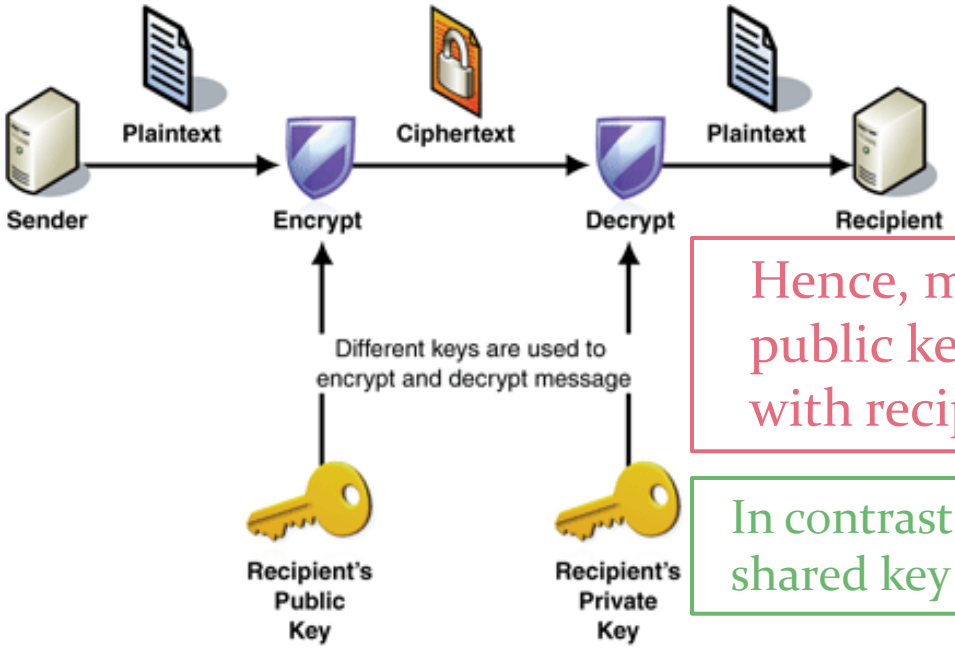
Information Engineering, CUHK
Product Security Engineering, Yahoo!

Agenda

- HTTPS and Browsers
  - Man-In-The-Middle attacks
  - Brief revision on public key cryptography
  - A high-level overview on SSL/TLS
  - Certificate Validity
- Threats and Mitigations
  - Common SSL Configuration Problems
  - A Side-channel Attack
  - SSL Stripping Attacks
  - Phishing
  - OWASP Top 10: A6-Sensitive Data Exposure, A5-Security Misconfigurations, A9-Using Components with Known Vulnerabilities

Revision on Public Key Cryptography

- A server generates 2 keys:
  - A public key – announced to the public
  - A private key – kept secret in the server
  - Using RSA algorithm (or ECC, etc), the two keys have the properties:
    - **Encryption:** $\text{Encrypt}_{\text{public-key}}(m) = c$; $\text{Decrypt}_{\text{private-key}}(c) = m$
    - **Signature:** $\text{Encrypt}_{\text{private-key}}(m) = c$; $\text{Decrypt}_{\text{public-key}}(c) = m$



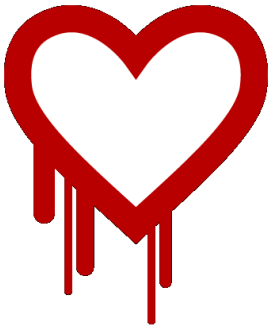
Hence, message encrypted with recipient's public key (private) can ONLY be decrypted with recipient's private (public) key

In contrast, for Symmetric Key Crypto, only one shared key is used. Algorithms: AES, 3DES, etc...

Reference: http://msdn.microsoft.com/en-us/library/ff647097.aspx

Overview of SSL/TLS

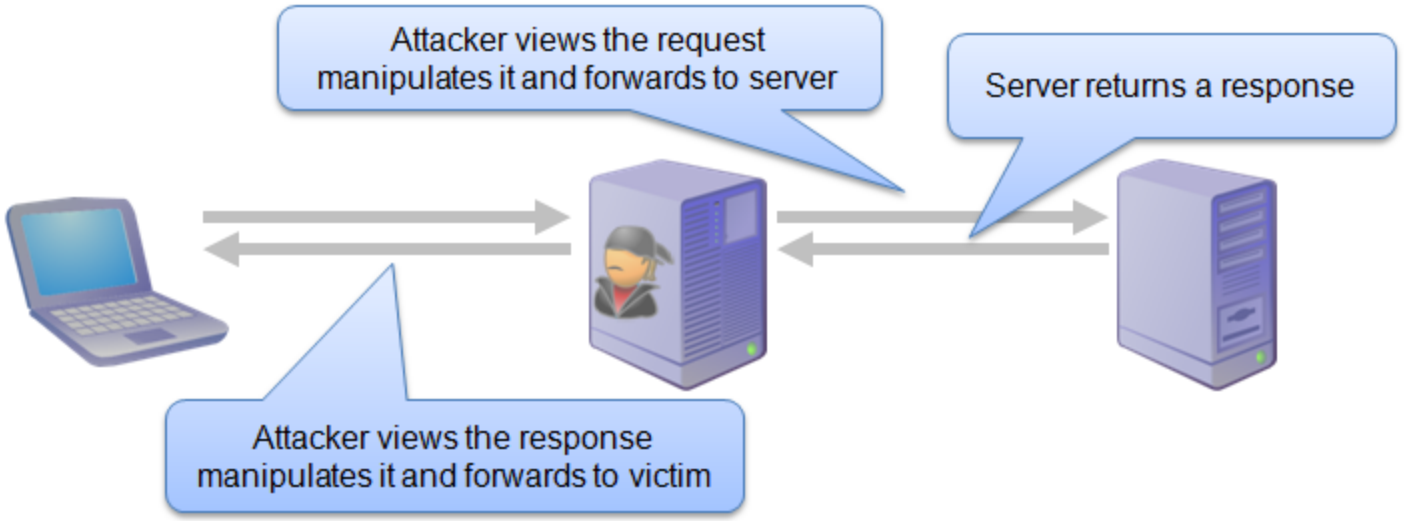
- SSL (or TLS) is a protocol to:
  - Mitigate MitM attacks
  - secure a data connection between server and client
  - using both public key and shared key cryptography
  - over an insecure network including the Internet
- Developed by Netscape in 1994
  - Latest version: v3 and later "rebranded" as TLS
  - Latest TLS version: v1.2
- Some Recent Attacks
  - HEARTBLEED
  - POODLE



Reference: http://technet.microsoft.com/en-us/library/cc767139.aspx#XSLTsection123121120120

Man-In-The-Middle (MitM) attack

- Instead of talking directly to the server,



  - Note: this is an active attacker, as he tampers content
- If no SSL is used, MitM can be launched steathily
- SSL is designed to mitigate MitM. Certificate warnings should appear to warn users

Diagram from http://blog.watchfire.com/wfblog/2009/02/active-man-in-the-middle-attacks.html

SSL Architecture

- SSL Record Protocol
- SSL Handshake Protocol
- SSL Change Cipher Spec Protocol
- SSL Alert Protocol

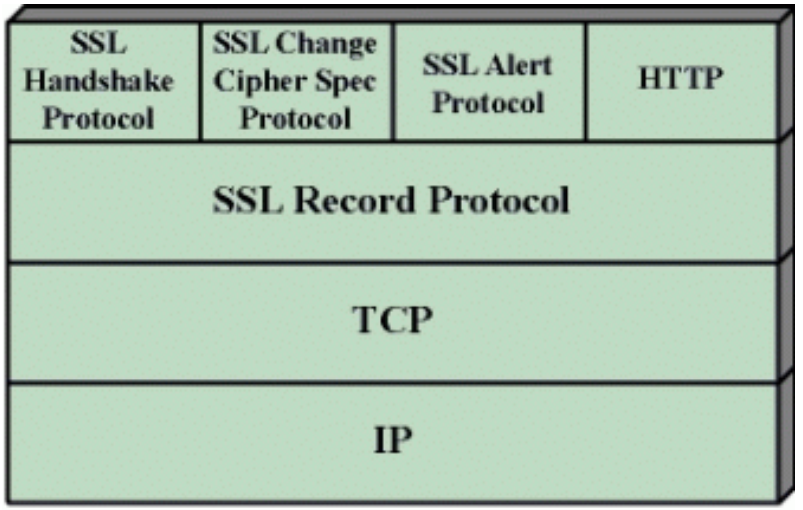


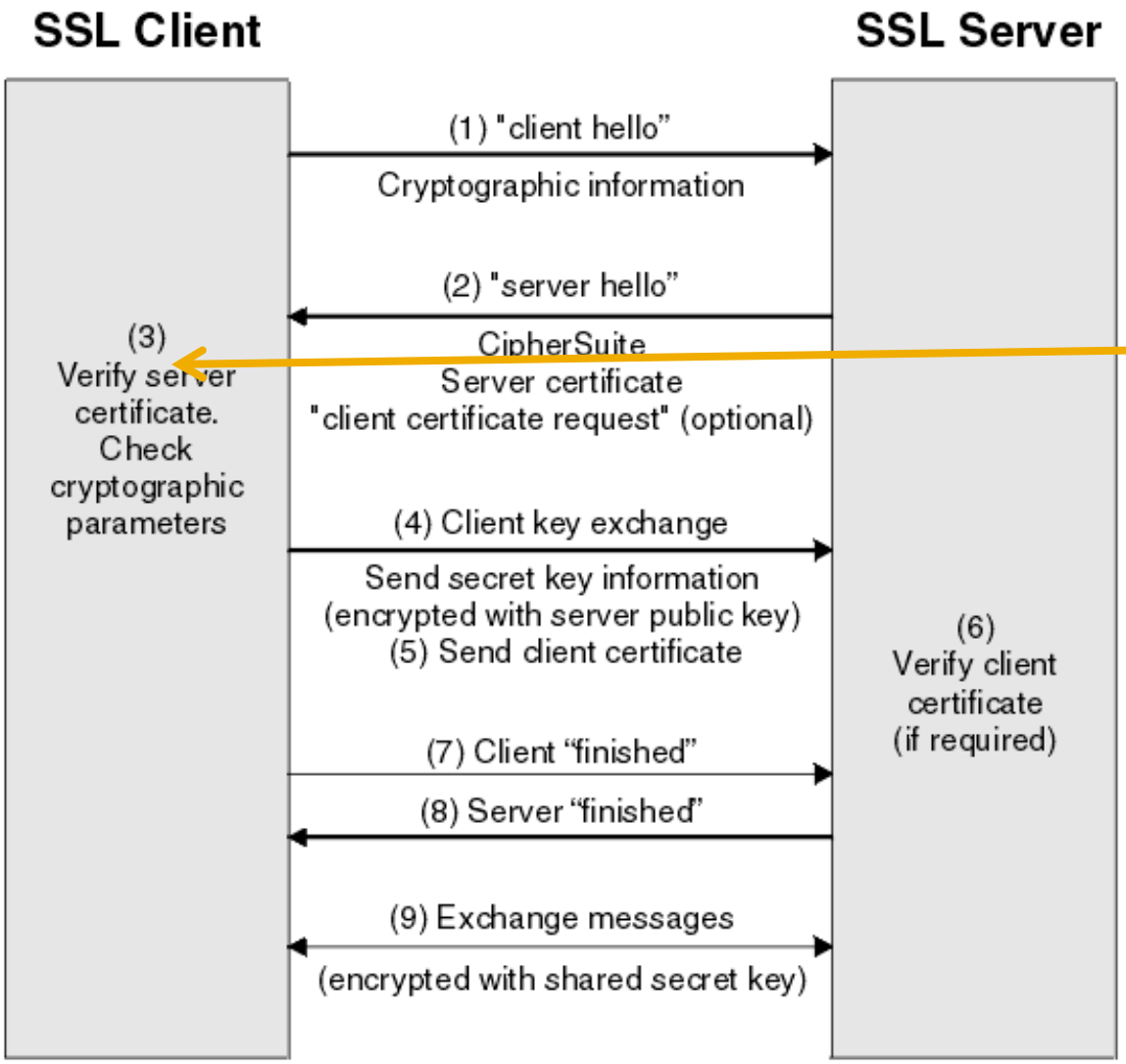
Figure 14.2 SSL Protocol Stack

- Let's see how it can ensure:
  - Authenticity
  - Confidentiality
  - Integrity

- For Full Explanation: visit here

Reference: http://technet.microsoft.com/en-us/library/cc767139.aspx#XSLTsection123121120120

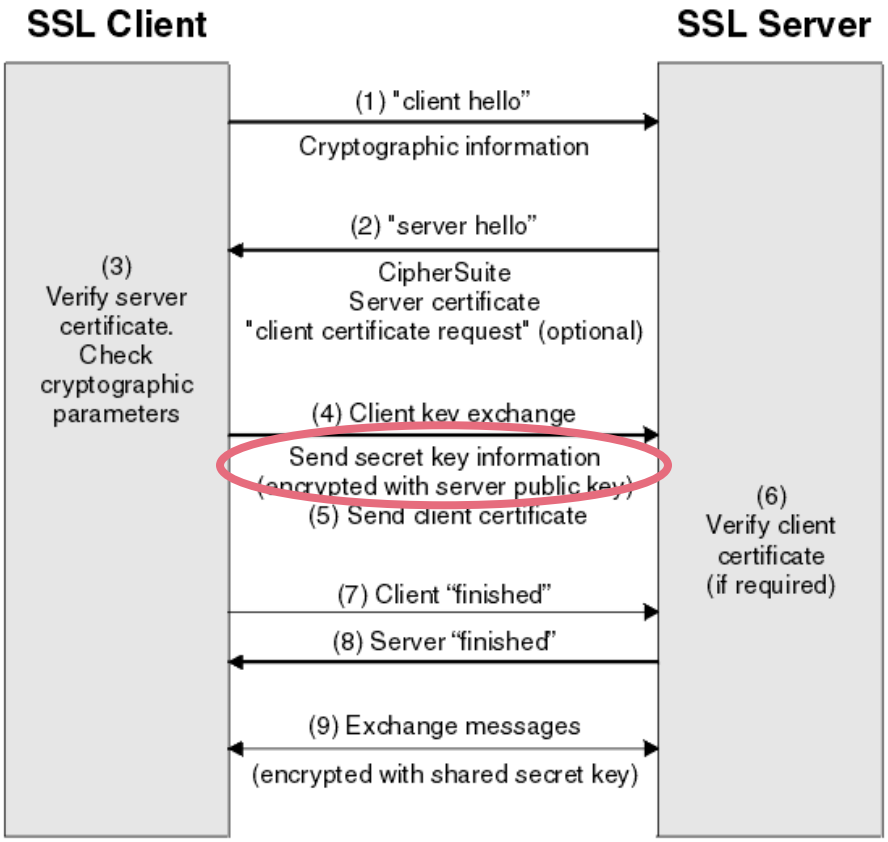
SSL Handshake Protocol



- Common to request for and verify ONLY server certificates
  - Authentication: Use public key cryptography to verify the server based on the cert
- In practice, cert revocation status may not checked

Reference: http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp?topic=%2Fcom.ibm.mq.csqzas.doc%2Fsy10660_.htm

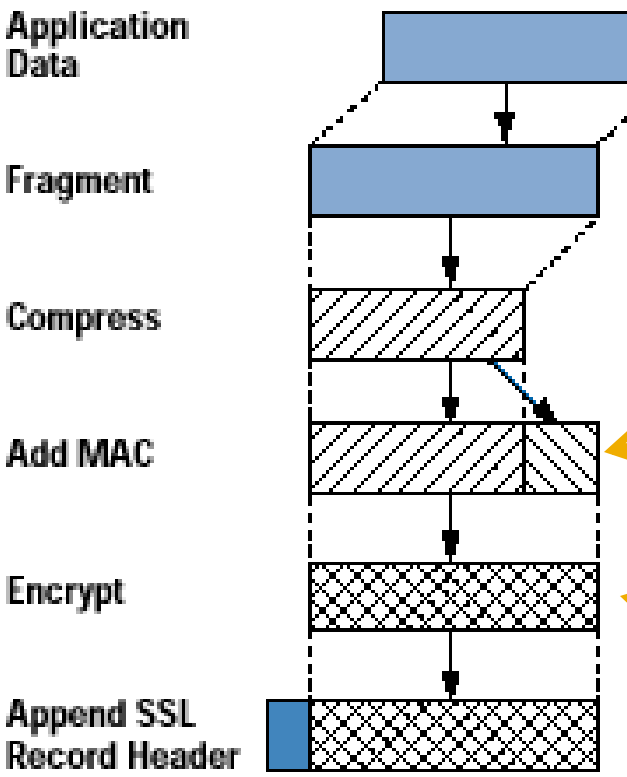
SSL Handshake Protocol



- Server Authentication:
  - Client generates a secret key info
  - Client sends the secret key info encrypted with server's public key
  - Server proves to client that it can decrypt with the *corresponding* private key
- If validated, use the secret key info to deduce a session key
  - SSL Record protocol then applies symmetric key encryption to subsequent data transmission

Reference: http://publib.boulder.ibm.com/infocenter/wmqv6/v6ro/index.jsp?topic=%2Fcom.ibm.mq.csqzas.doc%2Fsy10660_.htm

SSL Record Protocol



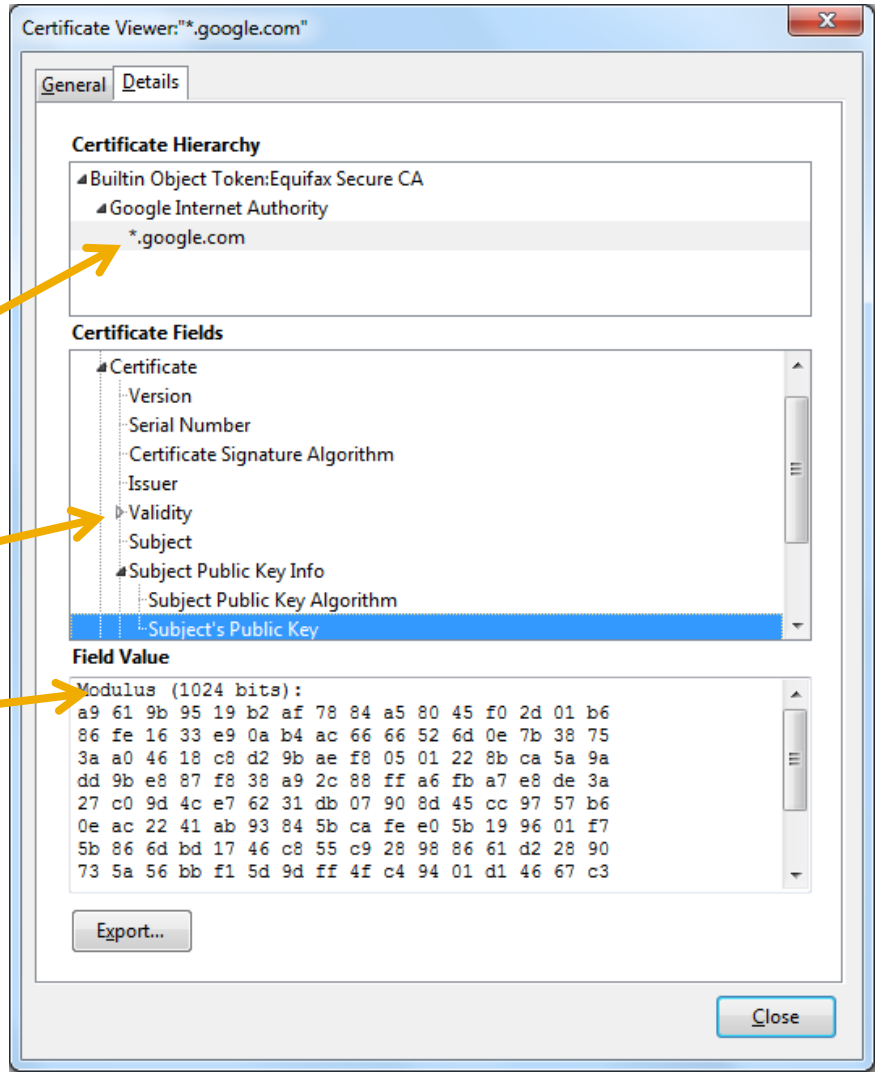
- To provide two security goals:
  - Integrity:
    Computing message digest with one-way hash function to prevent active attacks (i.e. data tampering)
  - Confidentiality:
    Encrypting data using symmetric cryptography to prevent passive attacks (i.e. eavesdropping)
- Used by other SSL sub-protocols and application protocols

Reference: http://technet.microsoft.com/en-us/library/cc767139.aspx#XSLTsection123121120120

Certs in Public Key Infrastructure (PKI)

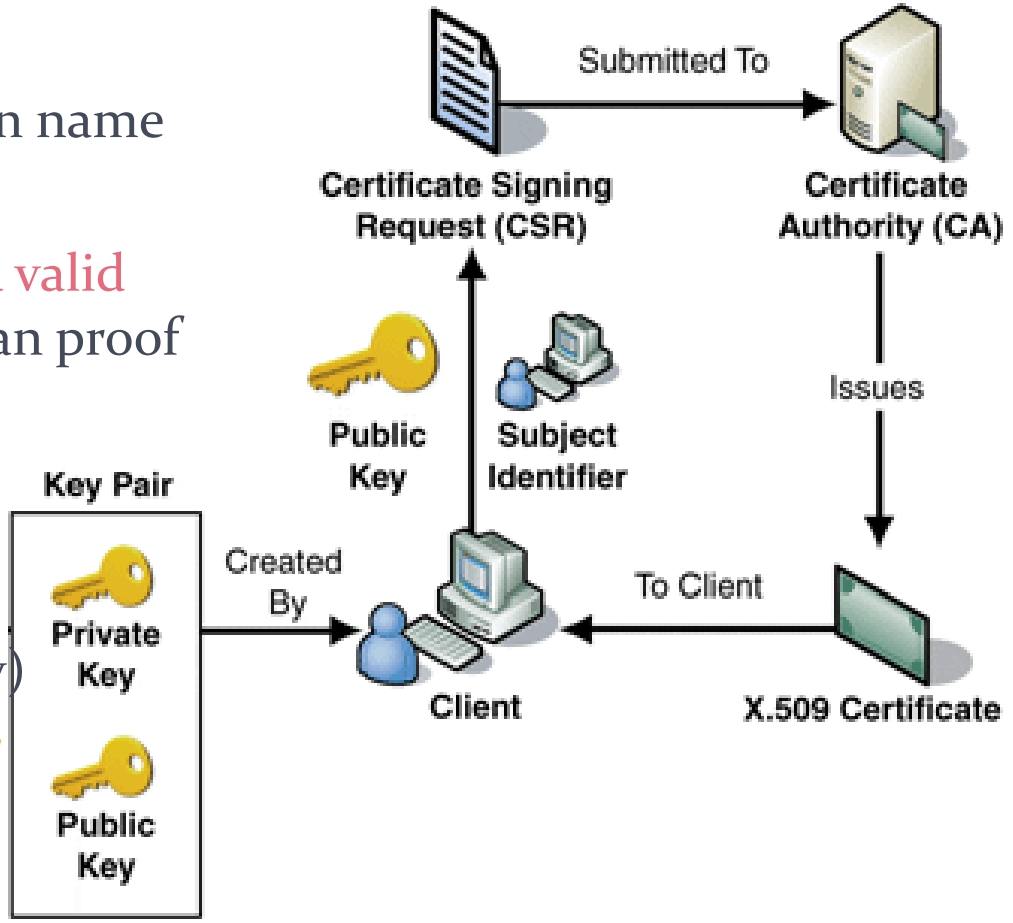
- PKI defines standards of Digital Certificates (certificate) and Certificate Authorities (CA), etc

- Important fields of a certificate:
  - Subject identifier
    aka Common Name or CN
    (domain name for server certs)
  - Validity period
  - CA-signed Public Key
  - etc…



CA in Public Key Infrastructure (PKI)

- To apply a server certificate from CA (as done in tutorial 7):
  - Generate a Key Pair with Subject equals the domain name
  - Produce a CSR to the CA
  - CA validates that applicant is a valid domain name holder and/or can proof his identity
  - If validated, CA certifies a cert by signing on among others, the public key and CN in CSR (i.e. encrypt with CA's priv. key)
  - Install the issued cert to server



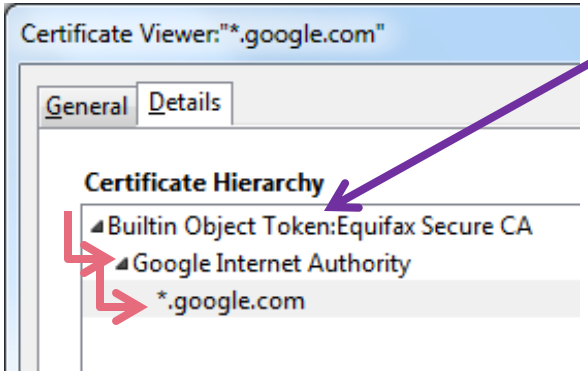
SSL Ecosystem Summarized

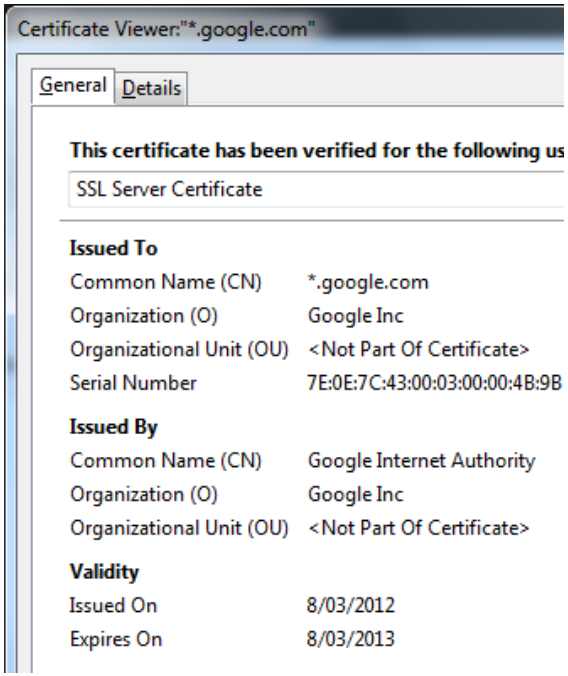
- Protocol designers
  (IETF TLS Working Group)
- Library developers
  (Microsoft, OpenSSL, NSS by Mozilla, …)
- Software vendors
  - Server vendors (IIS, mod_ssl)
  - Browser vendors (IE, Firefox, Chrome, …)
- Certificate Authorities and resellers
  (Verisign, Godaddy)
- Server administrators
- End users

Reference: I. Ristic and W. Kandek, "SSL and Browsers: The Pillars of Broken Security", RSA Conference, Feb 27-Mar 2, 2012

CERTIFICATE VALIDITY

Valid Certificates in Browsers



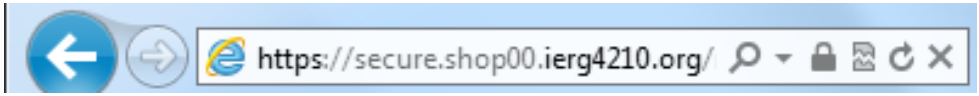


- Browsers/OS preinstalled some CA certs
  - All CA certs are self-signed (no issuer)
  - Implicitly trust on the CAs
- A certificate is considered valid if:
  - Not Expired: within validity period
  - Valid Issuer: verifies CA's signature using a preinstalled CA's cert, i.e. tests if cert info decrypted with CA's public key equal to what was signed on
    - if intermediate CA (e.g. the 2nd one on LHS) is present, verifies along the chain of certificates
  - CN matches domain name: checks if the common name of the final cert matches with the domain name of the current website

Reference: http://publib.boulder.ibm.com/infocenter/wmqv6/v6ro/index.jsp?topic=%2Fcom.ibm.mq.csqzas.doc%2Fsy10660_.htm

Browser UI: SSL Indicators for Valid Certs

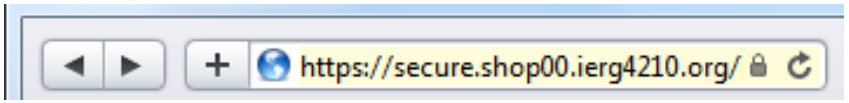
- The padlock changes location in every new browser version
  - Internet Explorer 9



  - Firefox does not use padlock anymore

  - Google Chrome

  - Safari 5



  - Opera 11

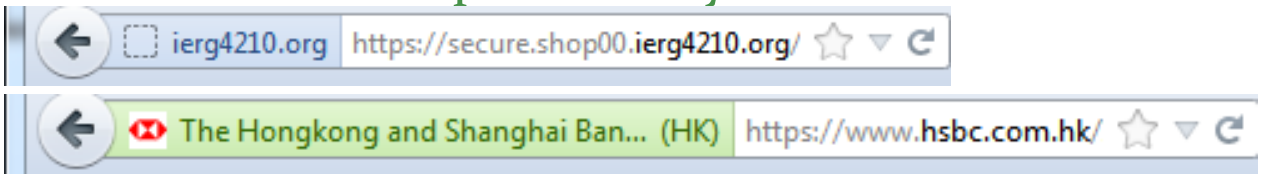
  - How about mobile browsers?? It even disappears after loading

Can you tell whether SSL is used from the location bar?

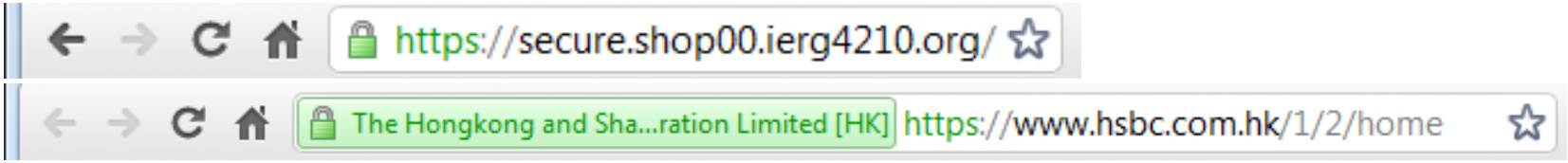
Browser UI: SSL Indicators for Valid EV Certs (1/2)

- Extended Validation Certs is issued ONLY to those who pay more and can provide a proof of real business identity; BUT technically, they're the same as ordinary certs
  - Internet Explorer 9

  - Firefox does not use padlock anymore



  - Google Chrome



Browser UI: SSL Indicators for Valid EV Certs (2/2)

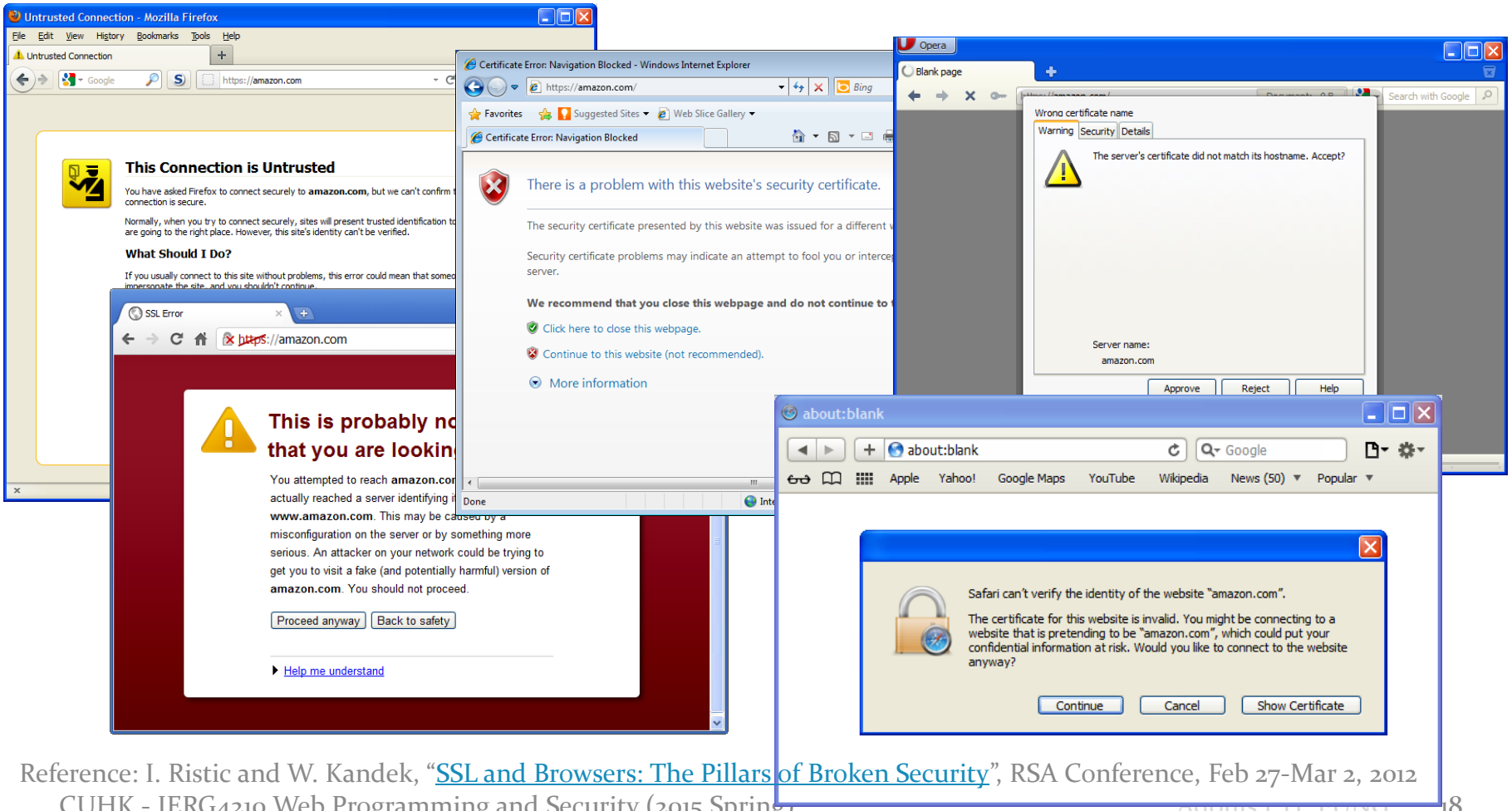
- Safari 5

- Opera 11

- Are EV-certified sites FREE from OWASP Top 10 attacks? **NO!**
- False sense of security!!
  - Relying on UI to tell security may not be a good solution

Certificate Warnings

- Invalid certificates trigger browsers' certificate warnings
  - SSL is to alert certificate warnings during man-in-the-middle attacks in which attackers cannot produce a valid cert for other domains



Reference: I. Ristic and W. Kandek, "SSL and Browsers: The Pillars of Broken Security", RSA Conference, Feb 27-Mar 2, 2012

THREATS AND MITIGATIONS

Reasons prohibiting SSL usage

- Slower than having no encryption
  - Google introduced SPDY, Will be part of HTTP/2.0
  - ECDSA is generally faster than RSA
- Prevent caching in Internet proxies
  - With proper configurations, caching public content is still possible
- CA-signed Certificate is expensive
  - Approx. US$12/year for a domain only
- Incompatible with virtual hosting
  - 1 IP can only associate w/1 cert
  - Latest standard has an extension to relax this. Modern browsers support.

Sensitive Data Exposure

- Some Common Problems
  - Missing Secure Flag for Cookies
  - No SSL at all or using invalid cert
  - Supporting insecure/week protocols and ciphers
  - Contain Mixed Content
  - Transition from HTTP to HTTPS
- Categorization Example
  - A5-Security Misconfigurations
  - A6-Sensitive Data Exposure
  - A9-Using Components with Known Vulnerabilities

Internet-wide Scan Results

Scan Date Completed	EFF [14] 2010-8	Ps & Qs [16] 2011-10	First 2012-6-10	Representative 2013-3-22	Latest 2013-8-4	Total Unique
Hosts with port 443 Open	16,200,000	28,923,800	31,847,635	33,078,971	36,033,088	(unknown)
Hosts serving HTTPS	7,704,837	12,828,613	18,978,040	21,427,059	24,442,824	108,801,503
Unique Certificates	4,021,766	5,758,254	7,770,385	8,387,200	9,031,798	42,382,241
Unique Trusted Certificates	1,455,391	1,956,267	2,948,397	3,230,359	3,341,637	6,931,223
Alexa Top 1 Mil. Certificates	(unknown)	89,953	116,061	141,231	143,149	261,250
Extd. Validation Certificates	33,916	71,066	89,190	103,170	104,167	186,159

Table 1: Internet-wide Scan Results — Between June 6, 2012 and August 4, 2013, we completed 110 scans of the IPv4 address space on port 443 and collected HTTPS certificates from responsive hosts.

Status	Hosts	
Expired	595,168	(5.80%)
Not Yet Valid	1,966	(0.02%)
Revoked	28,033	(0.27%)
No Trust Chain	654,667	(6.30%)
Misordered Chain	25,667	(0.24%)
Incorrect Chain	11,761	(0.14%)
Unnecessary Root	4,365,321	(42.2%)
Optimally Configured	4,657,133	(45.0%)

Table 11: Common Server Certificate Problems — We evaluate hosts serving browser-trusted certificates and classify common certificate and server configuration errors. The number of misconfigured hosts indicates that procuring certificates and correctly configuring them on servers remains a challenge for many users.

Reference:
https://jhalderm.com/pub/papers/https-imc13.pdf

OWASP Top 10 Application Security Risks

2010	2013
A1-Injection	A1-Injection
A2-Cross Site Scripting (XSS)	A2-Broken Authentication and Session Management
A3-Broken Authentication and Session Management	A3-Cross-Site Scripting (XSS)
A4-Insecure Direct Object References	A4-Insecure Direct Object References
A5-Cross Site Request Forgery (CSRF)	**A5-Security Misconfiguration**
A6-Security Misconfiguration	**A6-Sensitive Data Exposure**
A7-Insecure Cryptographic Storage	A7-Missing Function Level Access Control
A8-Failure to Restrict URL Access	A8-Cross-Site Request Forgery (CSRF)
A9-Insufficient Transport Layer Protection	**A9-Using Components with Known Vulnerabilities**
A10-Unvalidated Redirects and Forwards	A10-Unvalidated Redirects and Forwards

- References: https://www.owasp.org/index.php/Top_10_2010-Main https://www.owasp.org/index.php/Top_10_2013

To apply sufficient SSL protection

- We apply the following:
  1. All session cookies have their "secure" flag set (covered)
  2. Use valid certificate
  3. Support Strong Algorithms and Secure Cipher Suites
  4. No Mixed Content within the same page
  5. Tackling transition from HTTP to HTTPS
  - We will discuss number 2 to 5

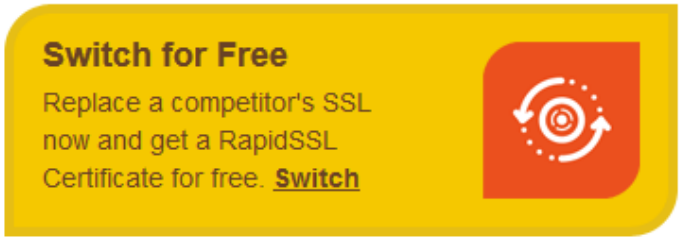
- More best practises can be found in https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices.pdf

Reference: https://www.owasp.org/index.php/Top_10_2010-A9

To apply sufficient SSL protection (2.1/5)

2. Use valid certificate (no more cert warnings)
   - Using a valid cert, hopefully users won't click "yes" in cert warnings when get man-in-the-middle (MITM) attacked one day
     - Pay GoDaddy US$12.99 for 1-year cert

     - Then extends 5 years more for FREE at rapidSSL (i.e. $12.99/6yrs=$2.16/yr), e.g. https://secure.ie.cuhk.edu.hk



   - In addition, remember to renew certificates before expiry!

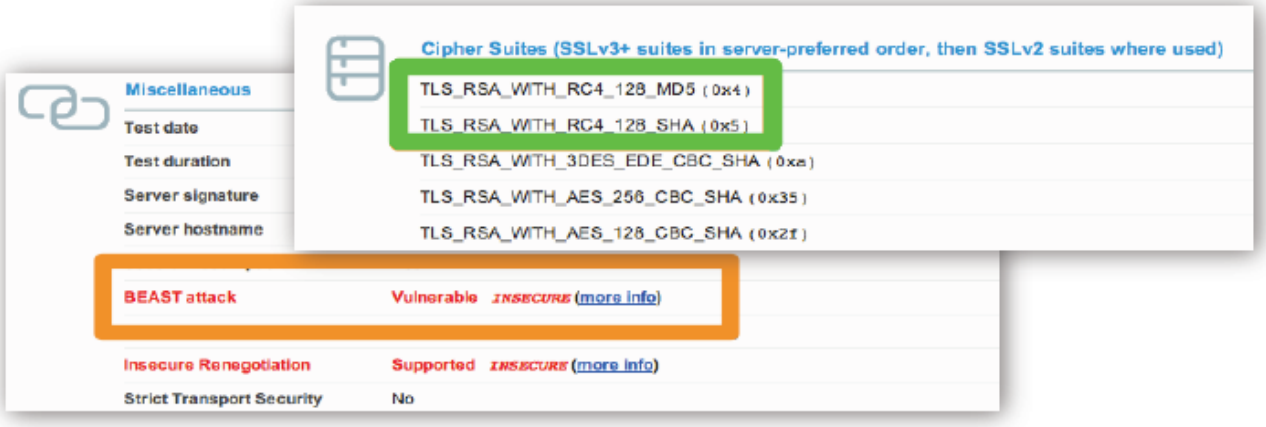
To apply sufficient SSL protection(2.2/5)

2. Use valid certificate (no more cert warnings)
   - IF INVALID (or self-signed) cert is used, users are forced to click "yes"
     - e.g. https://webmail.cse.cuhk.edu.hk
     - e.g. https://www2.cuhk.edu.hk/
   - During MITM, attacker's cert also triggers cert warning
     - How to differentiate a valid visit from a compromised one?
   - Is US$2.16/yr too expensive for CUHK departments?

   - Usability Studies find that users click "yes" very often
     - For IE7, 53% in 2007 and 95% in 2009;
     - For Firefox 3, 58% in 2009 (4 clicks to say "yes")
     - An incident: a bank not renewing cert discouraged only 1 out of 300 visitors

To apply sufficient SSL protection (3.1/5)

3. Support Strong Algorithms and Secure Cipher Suites
   - Example Flaw: BEAST attack against CBC Cipher Suites (Q4, 2011)
     - Vulnerability in SSL 3.0 and TLS 1.0
     - Decrypts small parts of traffic (e.g., cookies)
     - **Fixed a long time ago in TLS 1.1 (2006)**
     - **But TLS 1.1+ ignored by majority ("Attack not practical")**



   - and many other vulnerabilities…

Reference: I. Ristic and W. Kandek, "SSL and Browsers: The Pillars of Broken Security", RSA Conference, Feb 27-Mar 2, 2012

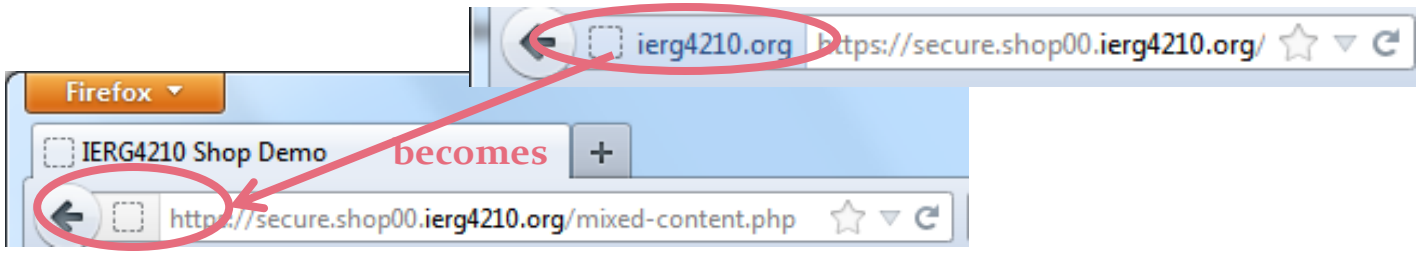
To apply sufficient SSL protection (3.2/5)

3. Support Strong Algorithms and Secure Cipher Suites
   - Example Flaw: SSL v2 and v3 are some insecure protocols
   - POODLE attack can force fallback to insecure protocols
     - Drawbacks of banning SSLv3: terminating old browsers' support
   - Statistics on SSLv3 and POODLE:
     - https://zmap.io/sslv3/
   - Example Flaw: ciphers below 128 bits generally weak

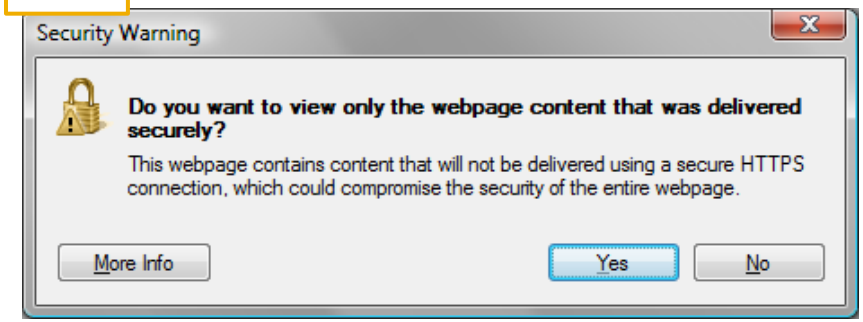
- Mitigation
  - Check using https://ssllabs.com
  - Apply the recommended algorithms and ciphers

Mixed Content (or mixed SSL)

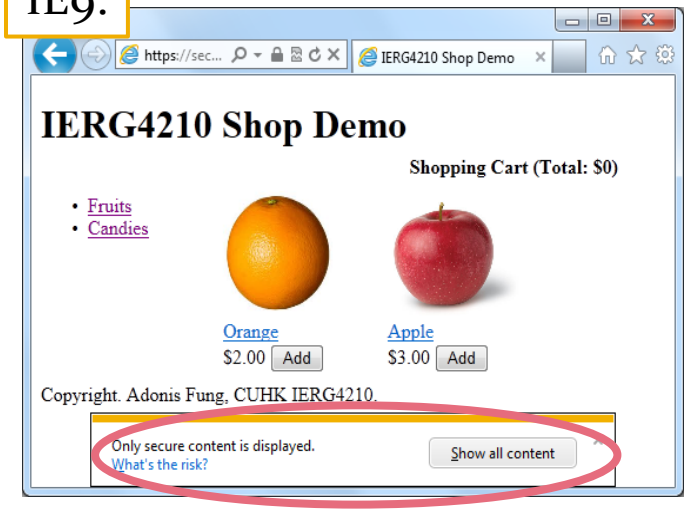
- When a HTTPS page embeds HTTP content
- Some browsers behave differently:



IE8:



IE9:

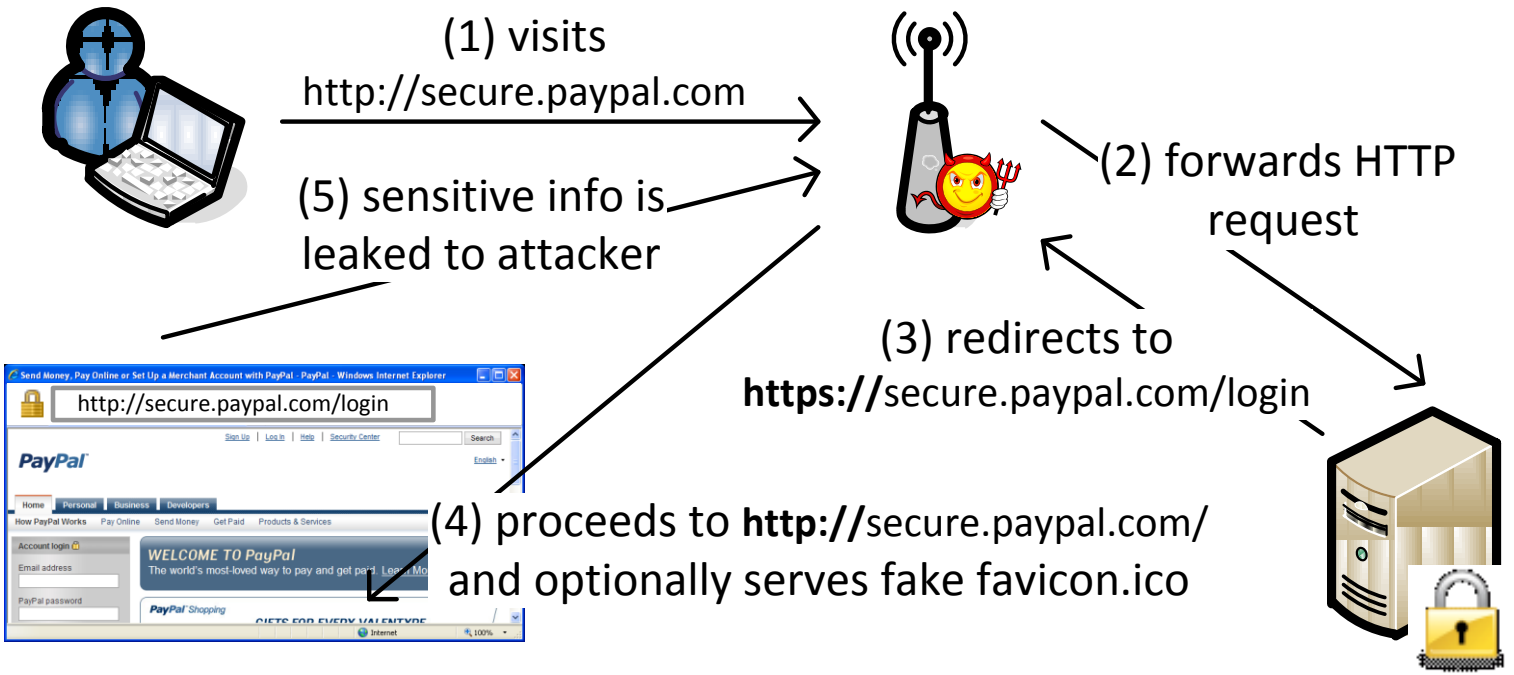


To apply sufficient SSL protection (4/5)

4. No Mixed Content within the same page
   - Attack: active attackers can modify Javascript served over HTTP
     - XSS can be launched in HTTPS origin due to origin inheritance
     - e.g. In `https://secure.shop00.ierg4210.org/`,
       `<script src="http://www.shop00.ierg4210.org/ui.js"/>`
     - Note: even if you expect a page to serve over HTTP, the attacker can still force a HTTPS connection to your site if web server (e.g. apache) allows it
   - Defense:
     - NEVER put http:// when specifying paths; Use either:
       - Relative URL: e.g. `/incl/prod/1.jpg`
       - Protocol-less URL: e.g. `//www.jquery.com/jquery.js`
         - » The protocol will be determined by the embedding page
       - Fix it as `https://` even if the embedding page is served over HTTP

To apply sufficient SSL protection (5.1/5)

5. Tackling transition from HTTP to HTTPS
   - SSLStrip Attack: To prevent a page from redirecting to HTTPS
     - Users seldom type https:// in location bar
     - Victim always stay in HTTP and the data can be tampered
     - No certificate warning will ever be triggered

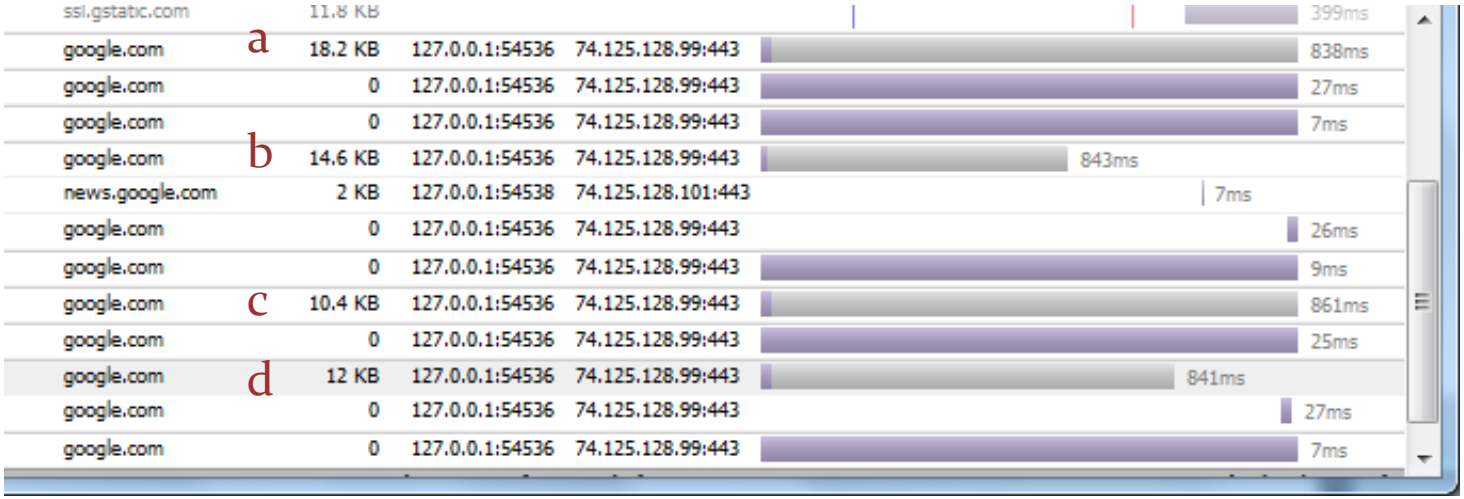


To apply sufficient SSL protection (5.2/5)

5. Tackling transition from HTTP to HTTPS
   - Defense 1: Apply HSTS if you're using valid certs
     - To apply HTTP Strict Transport Security (HSTS), insert a header in apache: `Header always set Strict-Transport-Security "max-age=600; includeSubDomains"`
     - Within 600 seconds, browsers remember the settings and convert automatically any HTTP URLs to into HTTPS
     - Valid Cert is a must; otherwise, cert warnings will have no button to bypass
     - e.g. before accessing the server http://example.com/some/page/ will be modified to https://example.com/some/page/
     - Major Limitation: Your browser must have visited the legitimate site once
   - Defense 2: Certifcate Pining
     - Hardcode the certificate signature for a particular in browsers
       - Chrome hardcodes google.com to use only certain certs
       - Updates through frequent browser update
     - Or similarly, signalled through a first legit visit (draft:websec-key-pinning)

A Side-channel Attack

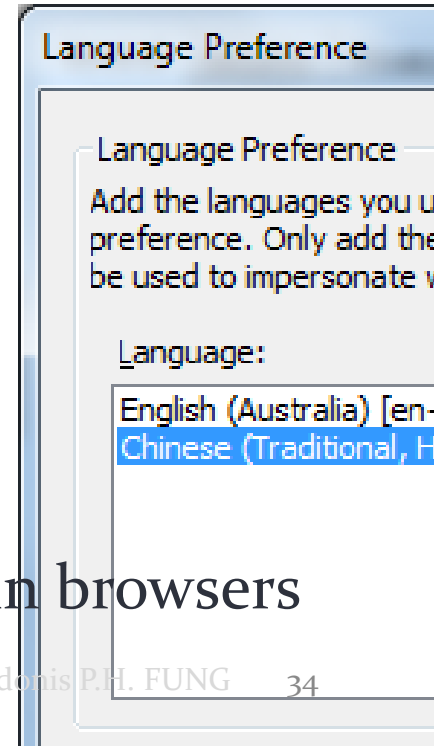
- Rather than attacking cryptography itself, recover encrypted information by gathering side-channel leaks
  - Given a finite set of data, if their sizes are distant and reproducible,
    - Monitoring only the size of ciphertext can uncover the original data
  - Demonstrated feasible over SSL and WiFi by S. Chen et al in 2010
  - For example: when you type in Google Suggest



Phishing

- Imitate the look-and-feel of a legitimate site
  - Copy the same HTML and images
  - If you like, copy also some "secure" seals
  - Lure/MITM victims to enter fake sites
    - Steal their passwords and credit cards, etc

- Except MITM, the only difference to tell apart is the URL
  - Look-alike domain names
    - e.g. west.example.com v.s. vvest.example.com
    - e.g. example.com v.s. examp1e.com
    - Attackers can apply certificates for the latter domains
  - Look-alike URLs or even IDN
    - Chinese / instead of / and ？ instead of ?
    - Nowadays, only works in IE
      with Chinese charset enabled

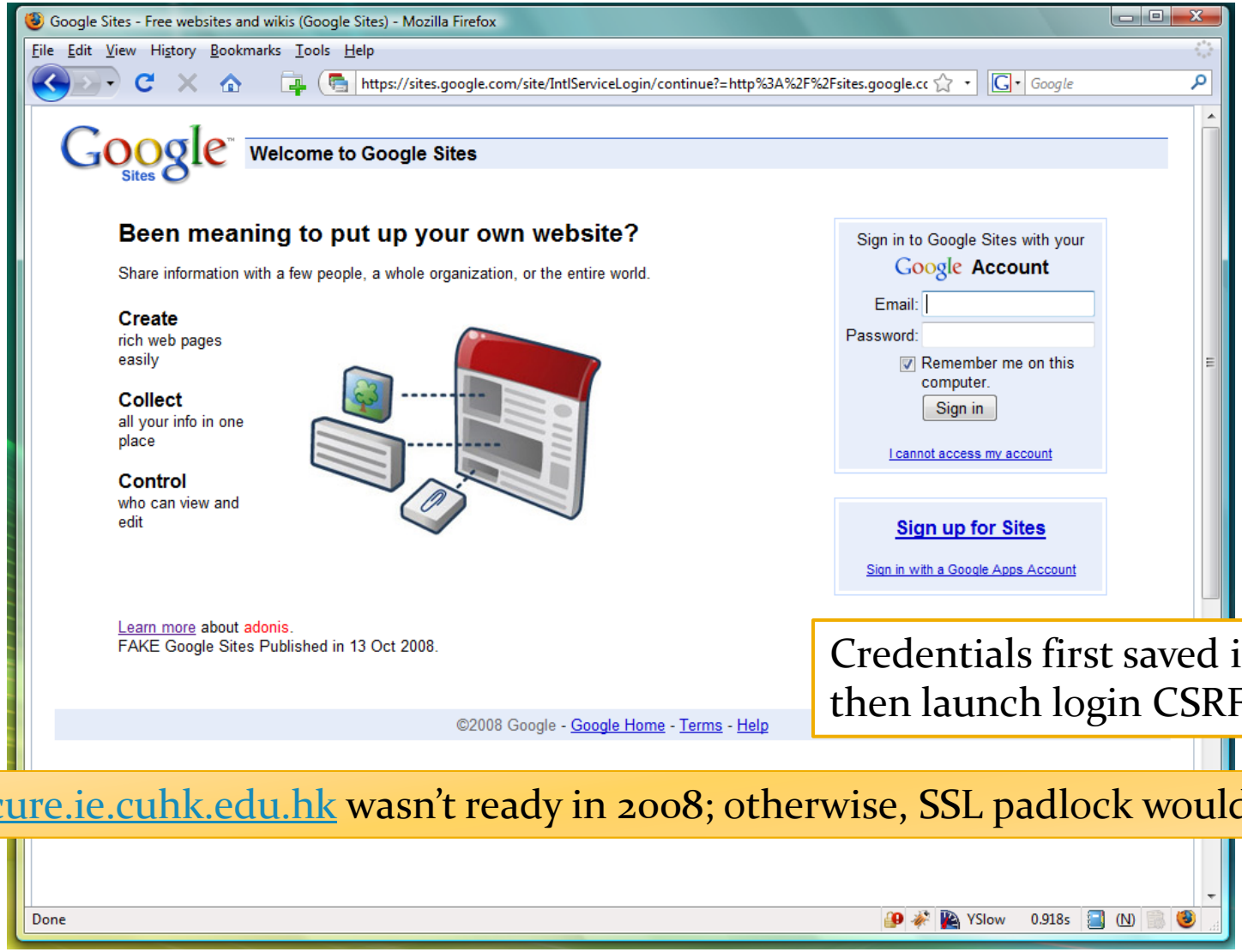


- Defense: Anti-phishing URL filters are deployed in browsers

Phishing with Google Sites (1/2)

Phishing with Google Sites (2/2)

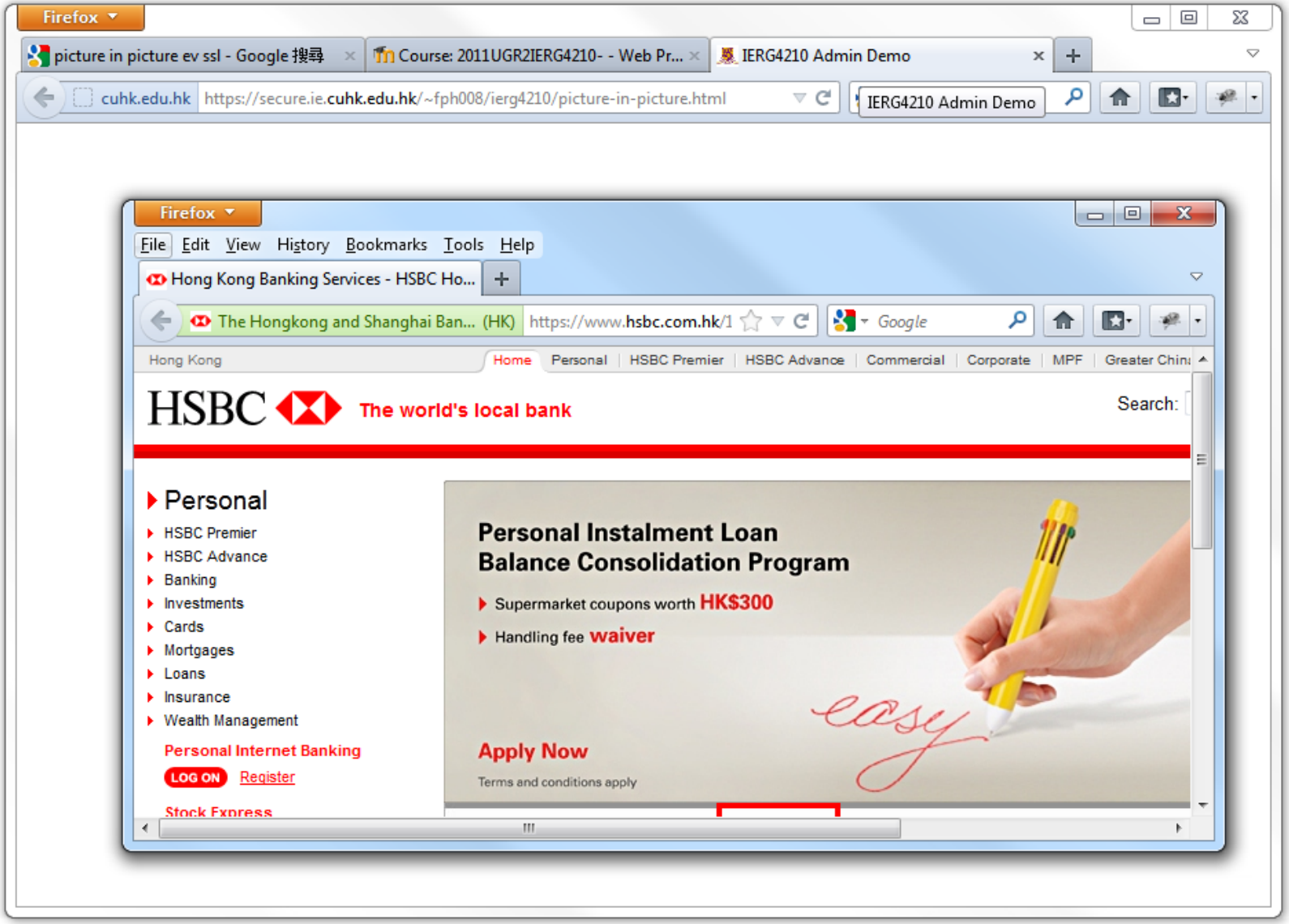
Problem fixed.



Credentials first saved in my DB, then launch login CSRF to google!

https://secure.ie.cuhk.edu.hk wasn't ready in 2008; otherwise, SSL padlock would look fine

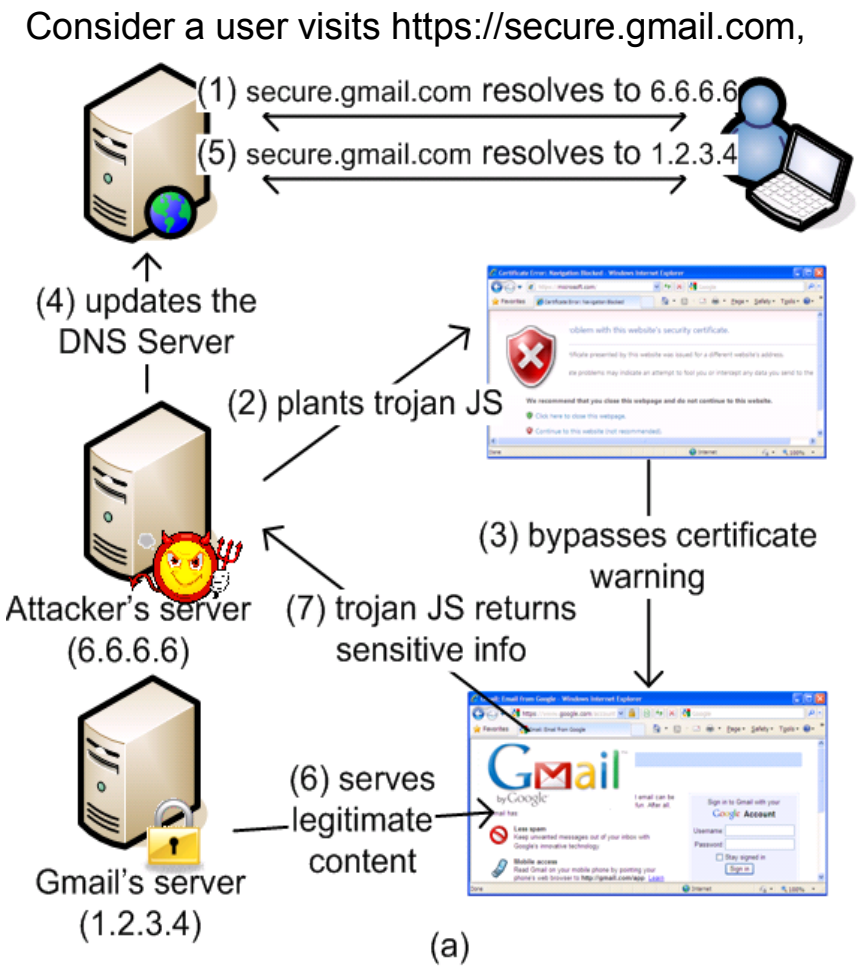
Picture in Picture Phishing Attack



DNS Rebinding Attacks (Time-permitting)

- A DNS is resolved to another host after a short TTL
  - Bypass SOP by DNS Rebinding
  - Cert warning is triggered but may be easily bypassed by users

- Defenses:
  - Deploy SSL and HSTS
  - Browsers prevent resolving to local IPs



Other Browser Security (time-permitting)

- XSS related
  - XSS Audits
  - Content Security Policy
- Man-in-the-Browser
  - Browser Extension Security: Adware/malware
  - E.g., Superfish installs a root CA cert, and its priv. key was easy to extract
- Two factor authentication
  - Duo Mobile
  - Google Authenticator