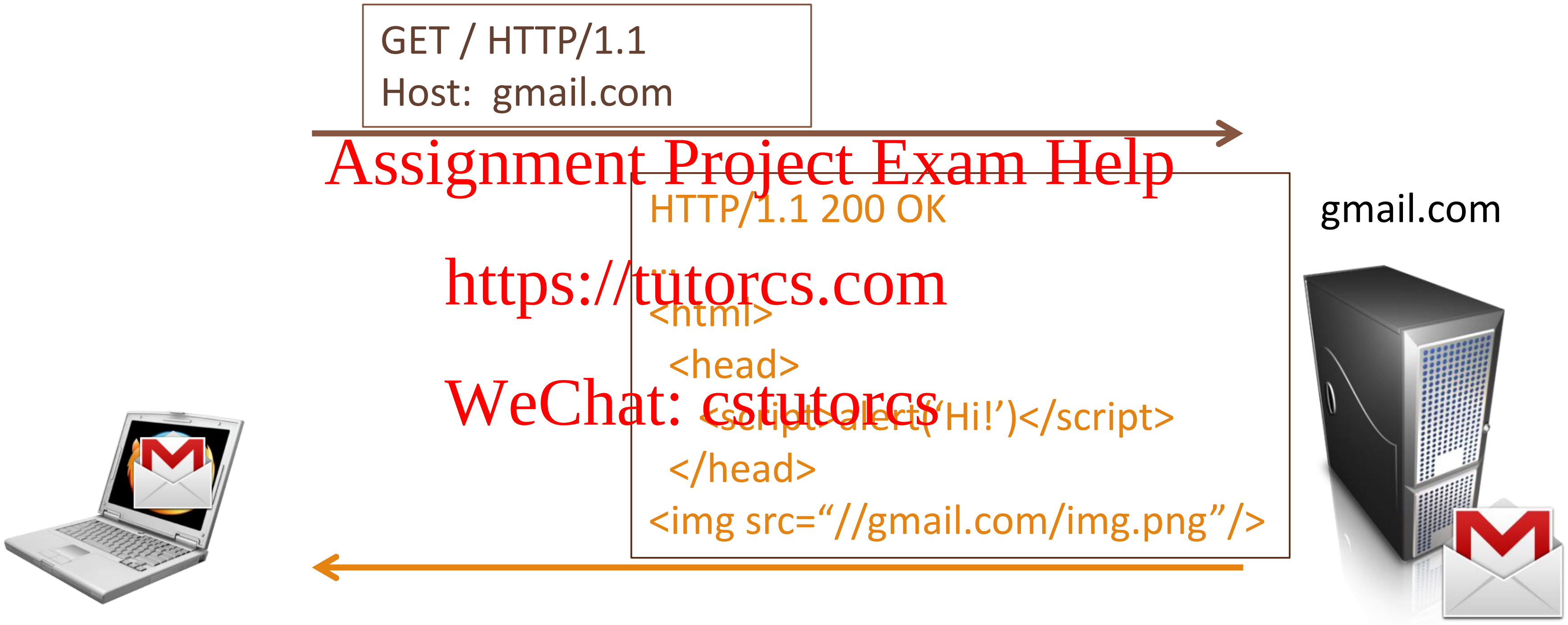
Transport Layer Security (TLS)

ECEN 4133
Feb 18, 2021

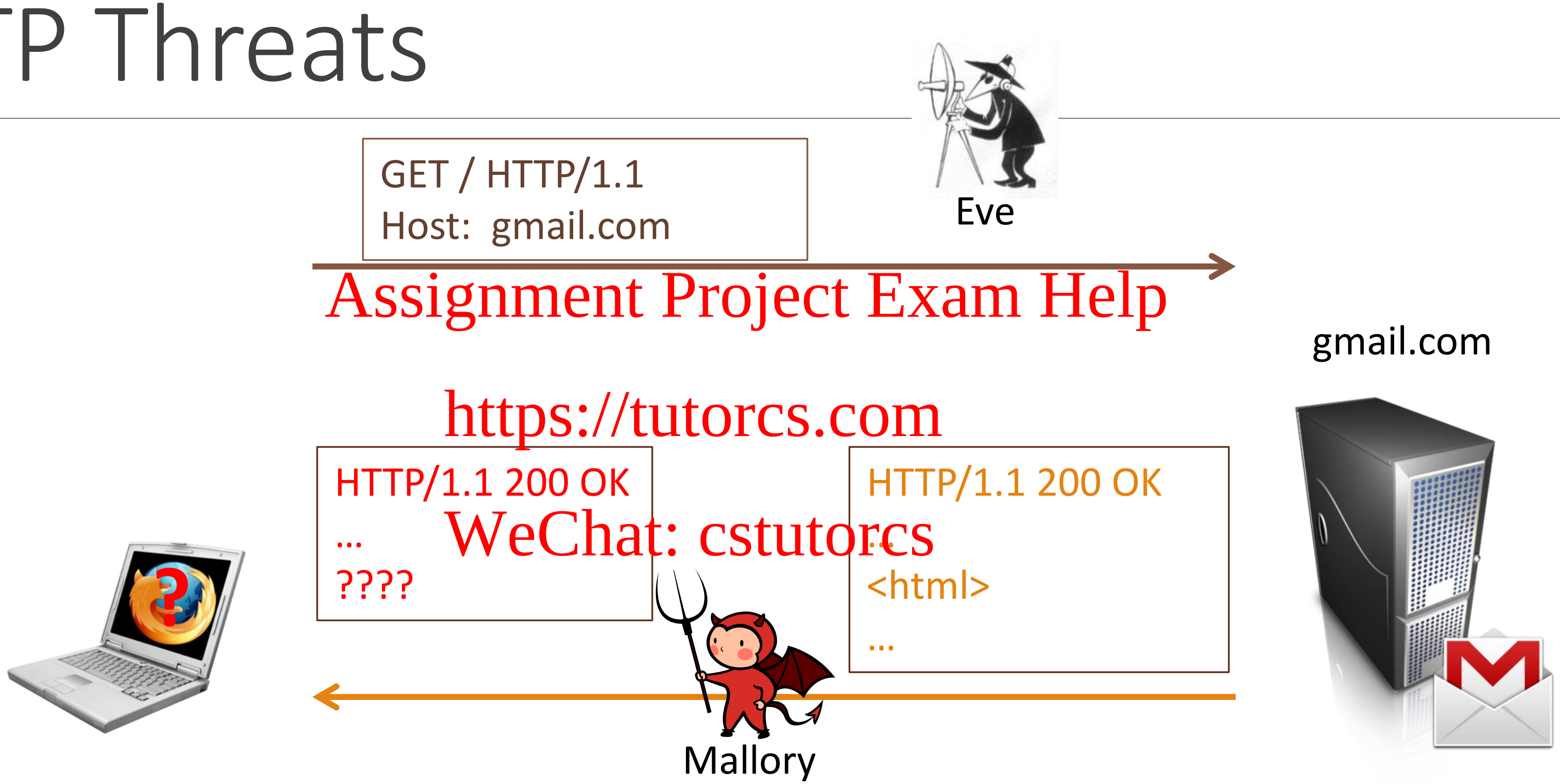
Review: HTTP



```
GET / HTTP/1.1
Host:  gmail.com
```

```
HTTP/1.1 200 OK

…
<html>
  <head>
    <script>alert('Hi!')</script>
  </head>
<img src="//gmail.com/img.png"/>
```

HTTP Threats



GET / HTTP/1.1
Host: gmail.com

HTTP/1.1 200 OK
…
????

HTTP/1.1 200 OK
…
<html>
…

HTTP Threats

Eve can observe:

- What page you are visiting (e.g. http://gmail.com/email84534)
- Server response (e.g. the content of your email)
- Cookies (Can now login as you!)
- Submitted forms (passwords, new emails, credit cards, etc)

Mallory can:

- Provide you false information (e.g. change the content of an email)
- Change what data you send (e.g. change the contents of what you post/send!)
- Insert Javascript on your page (e.g. tracking info / steal information from gmail's origin)

Solution:

- Cryptography! Confidentiality + Integrity
  - …but how?

How do we translate?

Cryptographic Primitives

Symmetric Encryption

RSA

PKI

Certificate

HMAC

Public Key

RC4

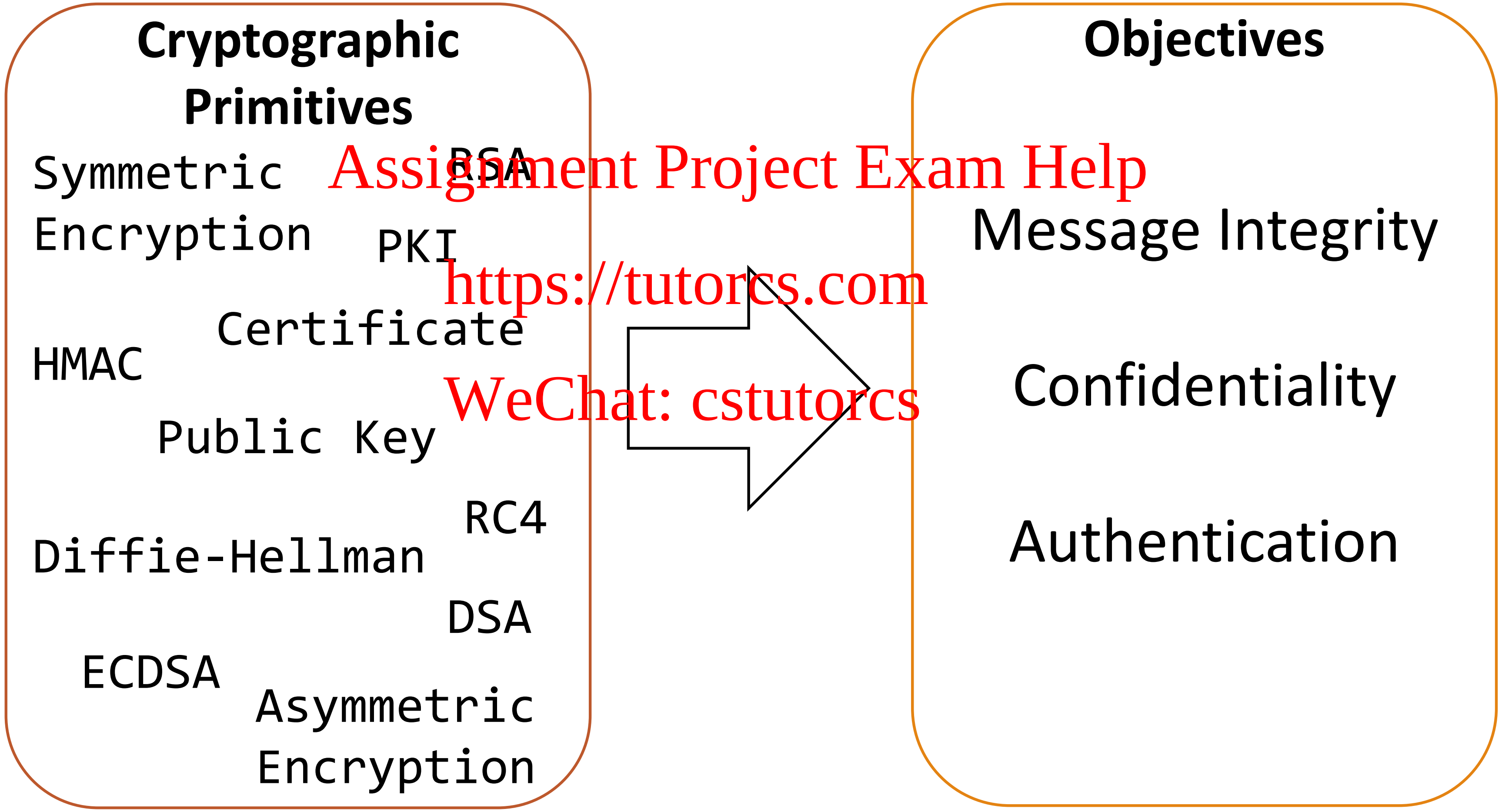
Diffie-Hellman

DSA

ECDSA

Asymmetric Encryption

How do we translate?



Cryptographic Primitives

Symmetric Encryption

RSA

PKI

Certificate

HMAC

Public Key

RC4

Diffie-Hellman

DSA

ECDSA

Asymmetric Encryption

Objectives

Message Integrity

Confidentiality

Authentication

How do we translate?

Cryptographic Primitives

Symmetric Encryption

RSA

PKI

Certificate

HMAC

Public Key

RC4

Diffie-Hellman

DSA

ECDSA

Asymmetric Encryption

Typical HTTPS Connection

HTTPS, TLS

Transport Layer Security (TLS)

- Previous versions: Secure Socket Layer (SSL) – do not use!
  - SSL 2
  - SSL 3.0
- TLS 1.0, 1.1, 1.2 – extensions/improvements to SSL 3.0
- TLS 1.3 – redesigned TLS (2018)

HTTPS – the S stands for Secure!

- HTTP over TLS

Case Study: TLS

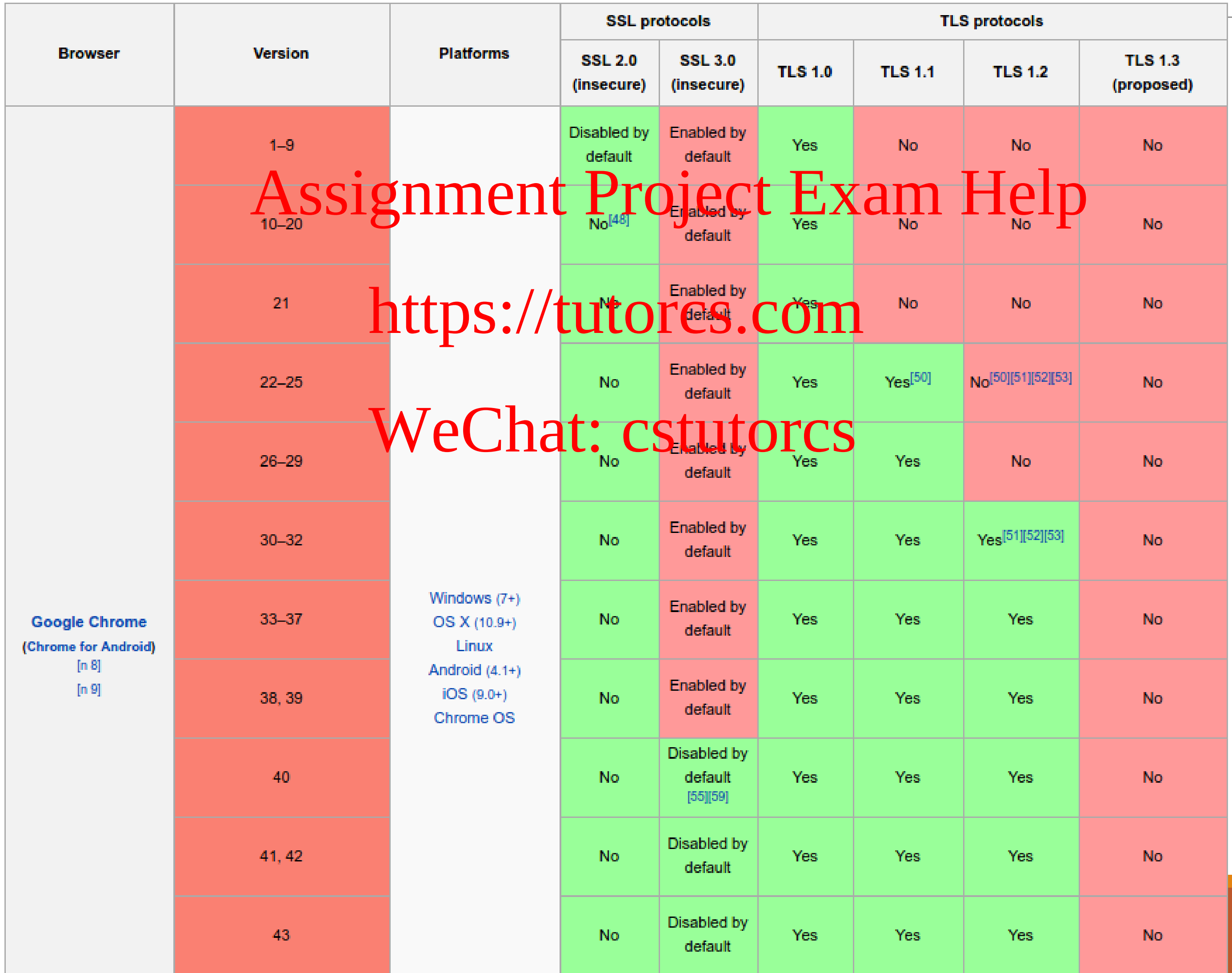
Arguably the most important (and widely used) cryptographic protocol on the Internet

Almost all encrypted protocols (minus SSH) uses TLS for transport encryption

HTTPS, POP3, IMAP, SMTP, FTP, NNTP, XMPP (Jabber), OpenVPN, SIP (VoIP), …

source: http://en.wikipedia.org/wiki/Transport_Layer_Security

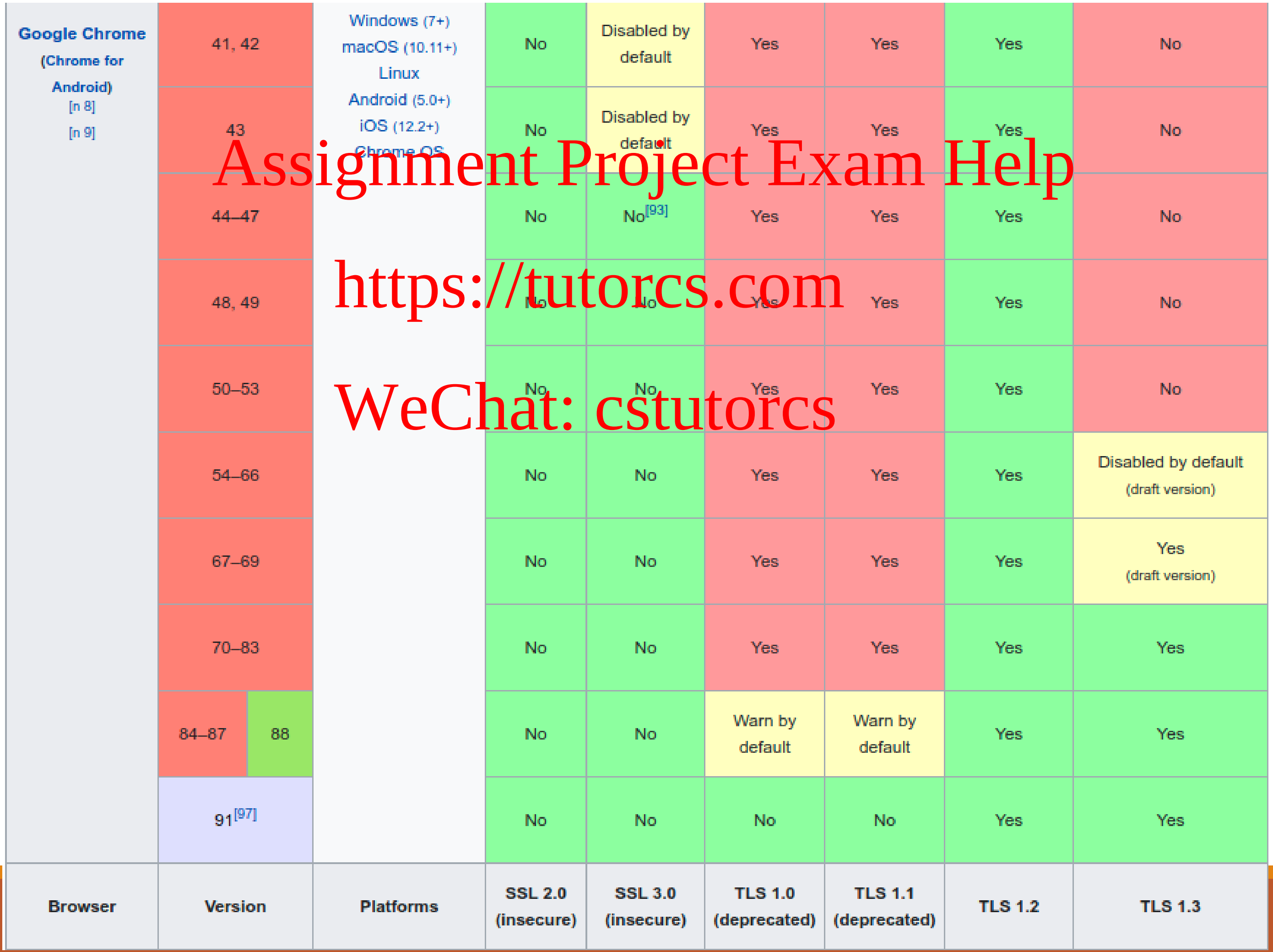
Browser TLS Support



Browser	Version	Platforms	SSL protocols		TLS protocols			
			SSL 2.0 (insecure)	SSL 3.0 (insecure)	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3 (proposed)
Google Chrome (Chrome for Android) [n 8] [n 9]	1–9	Windows (7+) OS X (10.9+) Linux Android (4.1+) iOS (9.0+) Chrome OS	Disabled by default	Enabled by default	Yes	No	No	No
	10–20		No[48]	Enabled by default	Yes	No	No	No
	21		No	Enabled by default	Yes	No	No	No
	22–25		No	Enabled by default	Yes	Yes[50]	No[50][51][52][53]	No
	26–29		No	Enabled by default	Yes	Yes	No	No
	30–32		No	Enabled by default	Yes	Yes	Yes[51][52][53]	No
	33–37		No	Enabled by default	Yes	Yes	Yes	No
	38, 39		No	Enabled by default	Yes	Yes	Yes	No
	40		No	Disabled by default [55][59]	Yes	Yes	Yes	No
	41, 42		No	Disabled by default	Yes	Yes	Yes	No
	43		No	Disabled by default	Yes	Yes	Yes	No

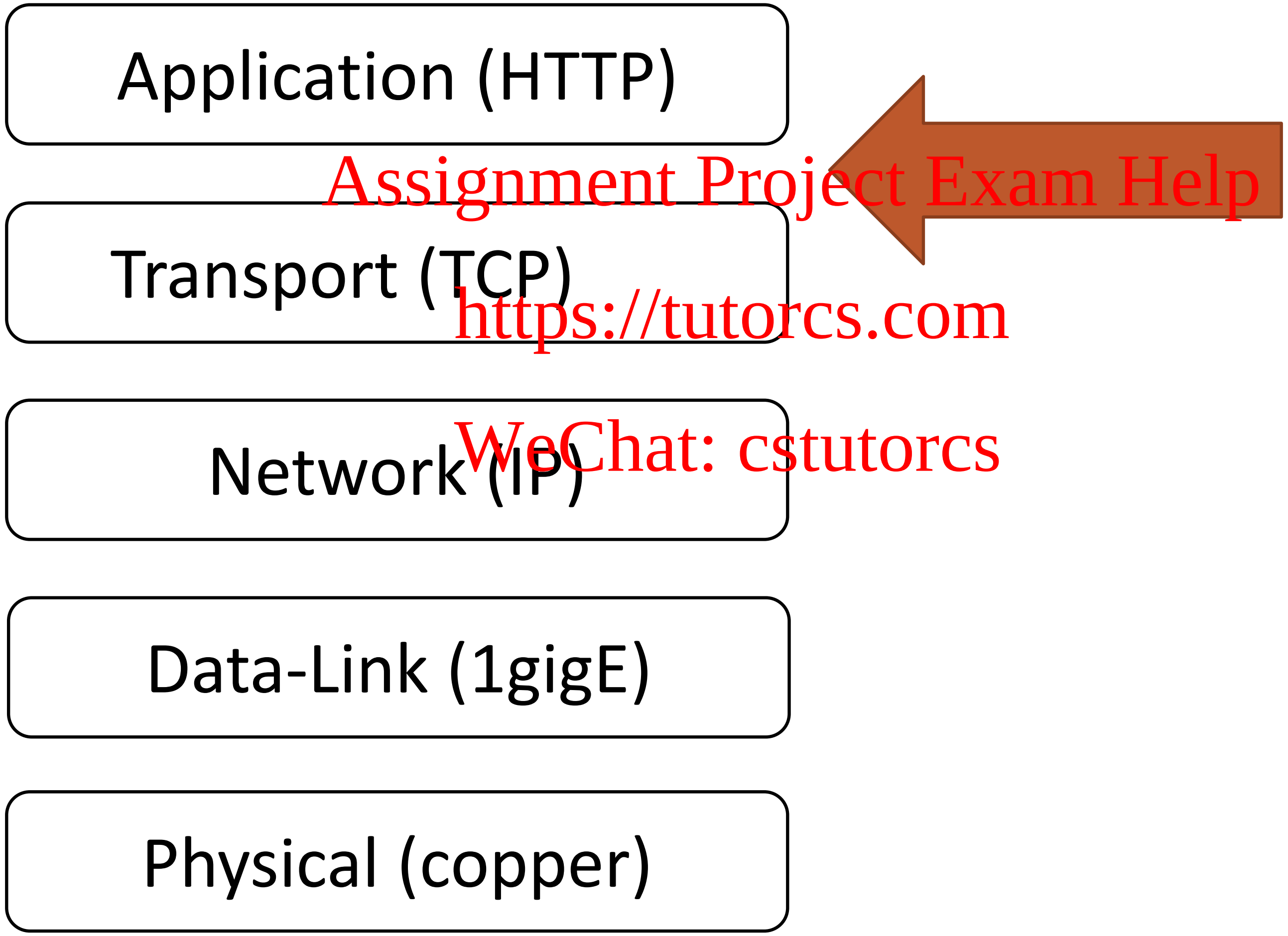
source: http://en.wikipedia.org/wiki/Transport_Layer_Security

Browser TLS support



Browser	Version	Platforms	SSL 2.0 (insecure)	SSL 3.0 (insecure)	TLS 1.0 (deprecated)	TLS 1.1 (deprecated)	TLS 1.2	TLS 1.3
Google Chrome (Chrome for Android) [n 8] [n 9]	41, 42	Windows (7+) macOS (10.11+) Linux Android (5.0+) iOS (12.2+) Chrome OS	No	Disabled by default	Yes	Yes	Yes	No
	43		No	Disabled by default	Yes	Yes	Yes	No
	44–47		No	No[93]	Yes	Yes	Yes	No
	48, 49		No	No	Yes	Yes	Yes	No
	50–53		No	No	Yes	Yes	Yes	No
	54–66		No	No	Yes	Yes	Yes	Disabled by default (draft version)
	67–69		No	No	Yes	Yes	Yes	Yes (draft version)
	70–83		No	No	Yes	Yes	Yes	Yes
	84–87 88		No	No	Warn by default	Warn by default	Yes	Yes
	91[97]		No	No	No	No	Yes	Yes

Where does TLS live?



Application (HTTP)

Transport (TCP)

Network (IP)

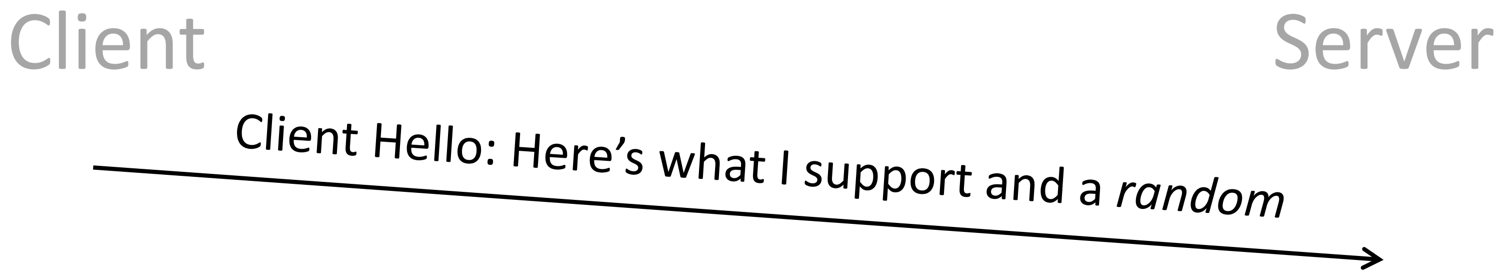
Data-Link (1gigE)

Physical (copper)

Client

Server

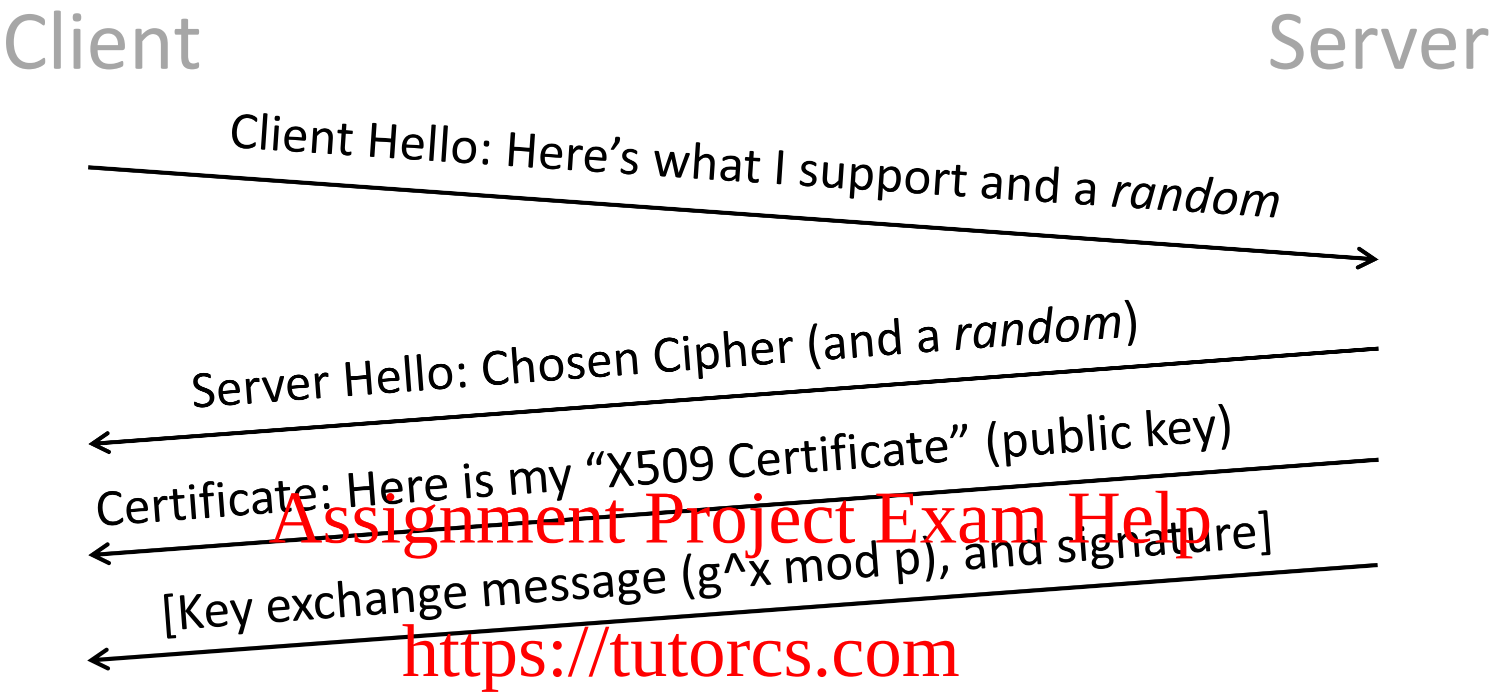
“the handshake”



Client

Server

Client Hello: Here’s what I support and a *random*



Client

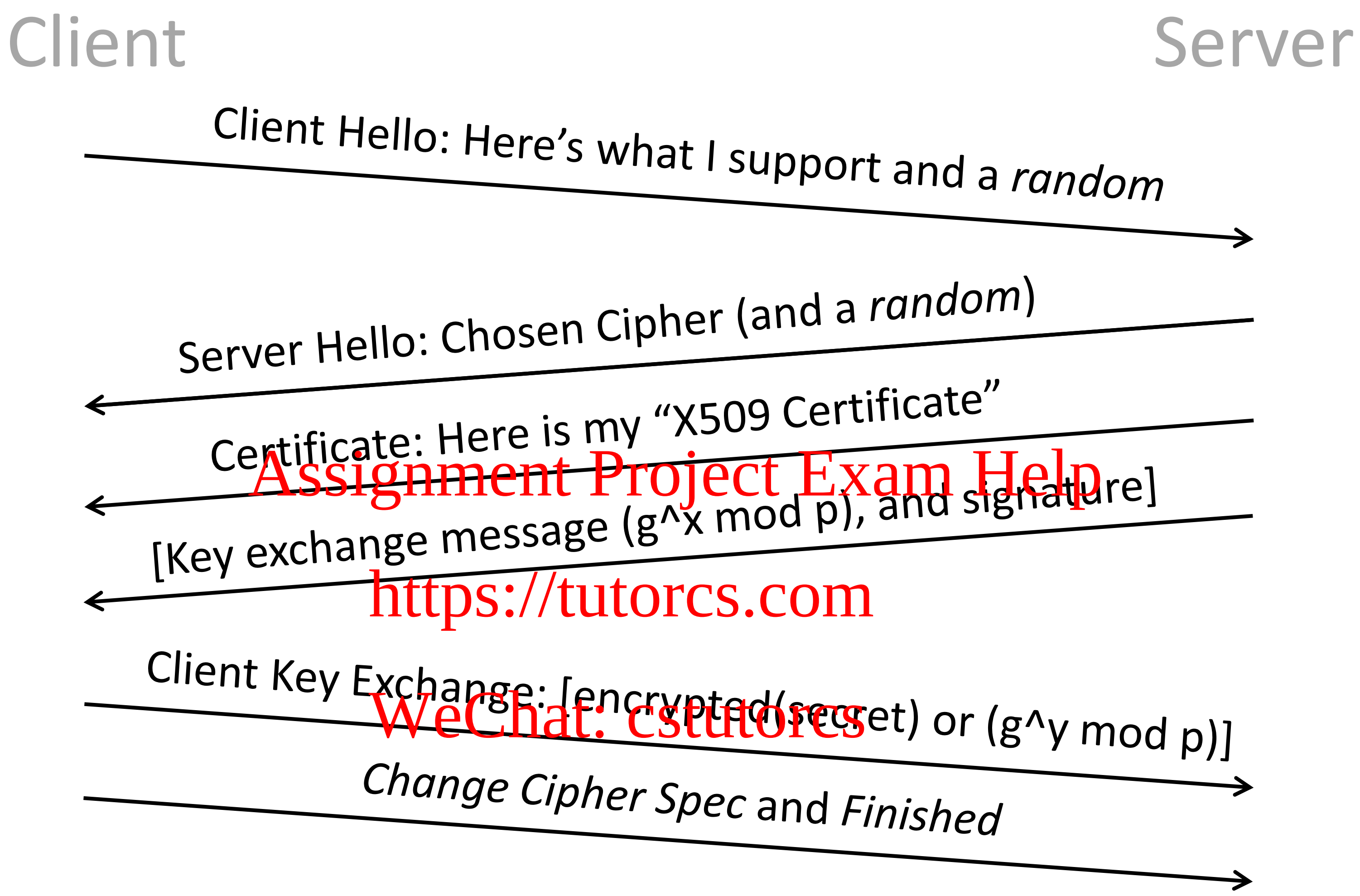
Server

Client Hello: Here’s what I support and a *random*

Server Hello: Chosen Cipher (and a *random*)

Certificate: Here is my “X509 Certificate” (public key)

[Key exchange message (g^x mod p), and signature]



Client

Server

Client Hello: Here's what I support and a *random*

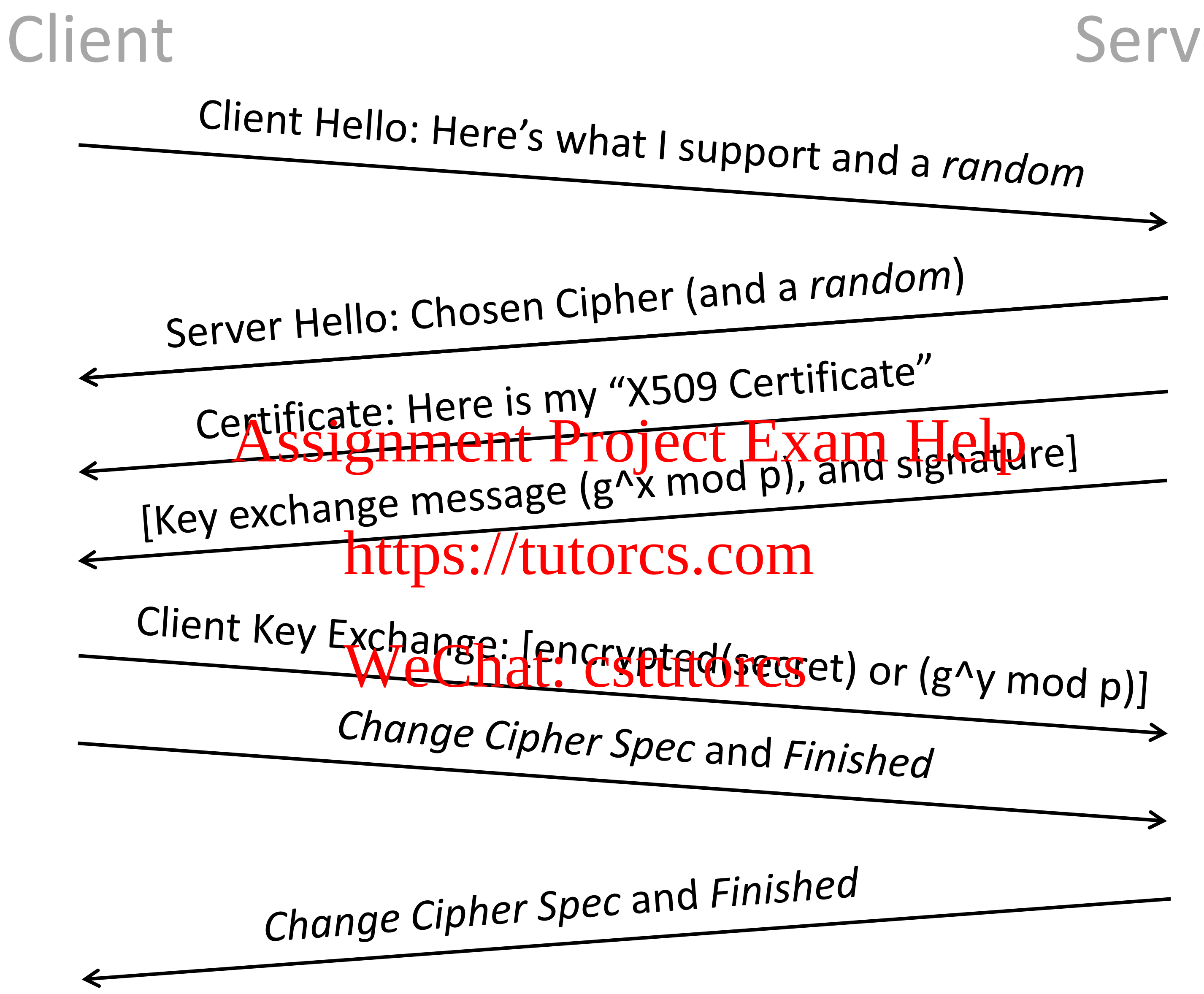
Server Hello: Chosen Cipher (and a *random*)

Certificate: Here is my "X509 Certificate"

[Key exchange message (g^x mod p), and signature]

Client Key Exchange: [encrypted(secret) or (g^y mod p)]

Change Cipher Spec and *Finished*



Client

Server

Client Hello: Here's what I support and a *random*

Server Hello: Chosen Cipher (and a *random*)

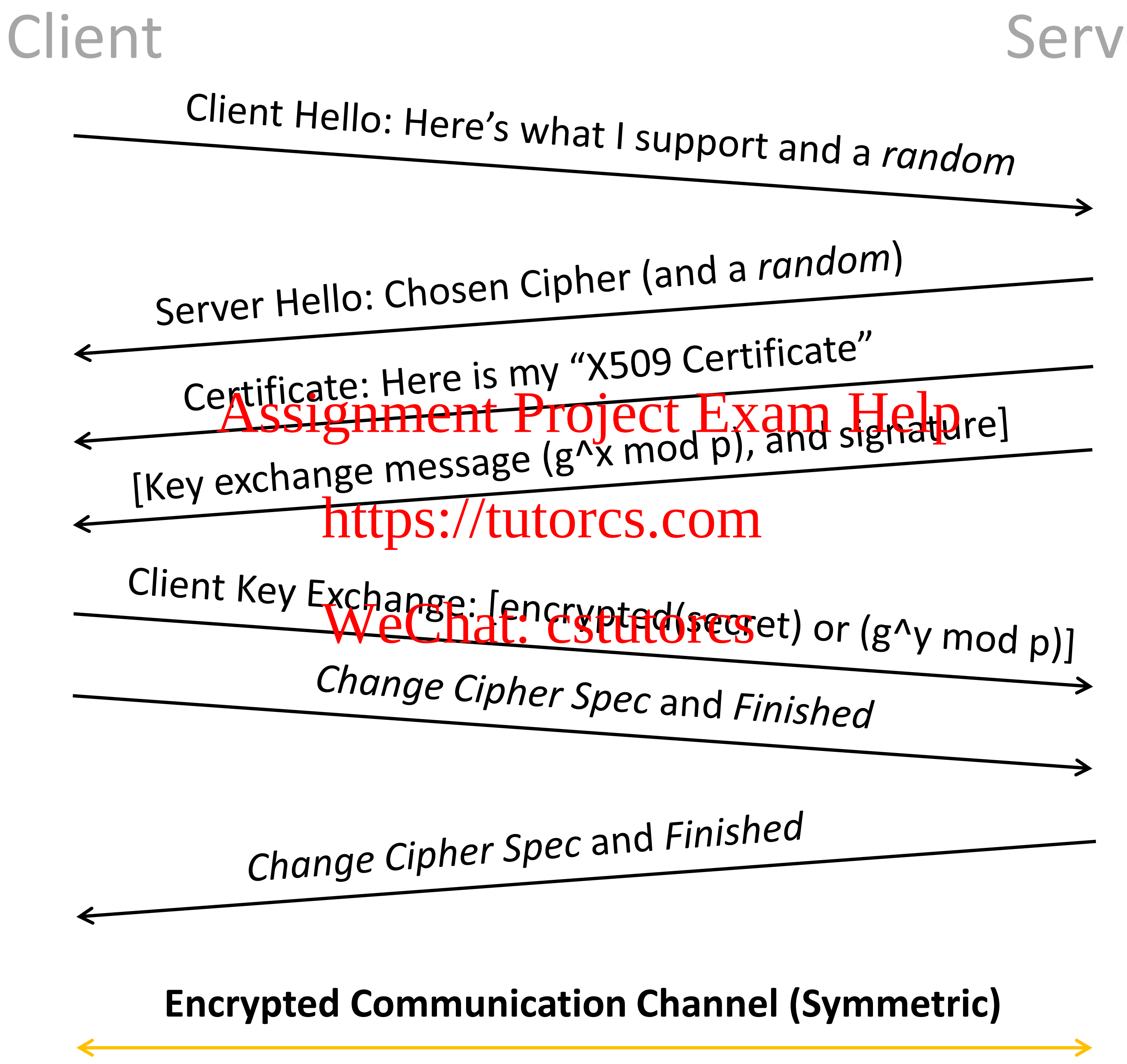
Certificate: Here is my "X509 Certificate"

[Key exchange message (g^x mod p), and signature]

Client Key Exchange: [encrypted(secret) or (g^y mod p)]

Change Cipher Spec and *Finished*

Change Cipher Spec and *Finished*



Client

Server

Client Hello: Here's what I support and a *random*

Server Hello: Chosen Cipher (and a *random*)

Certificate: Here is my "X509 Certificate"

[Key exchange message (g^x mod p), and signature]

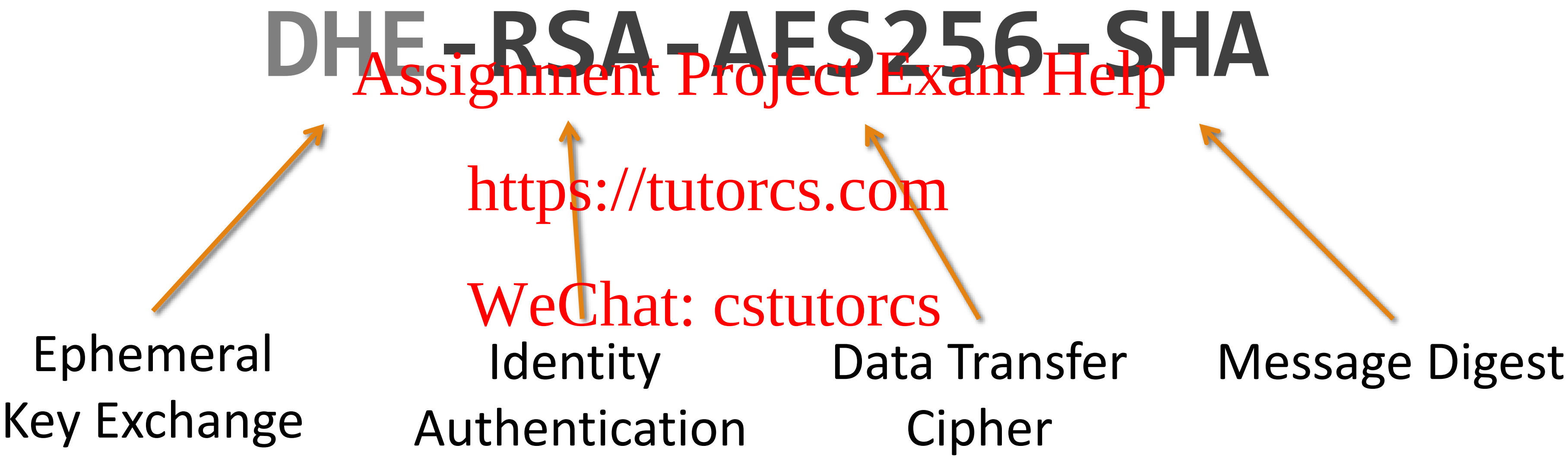
Client Key Exchange: [encrypted(secret) or (g^y mod p)]

Change Cipher Spec and *Finished*

Change Cipher Spec and *Finished*

Encrypted Communication Channel (Symmetric)

Cipher Suites



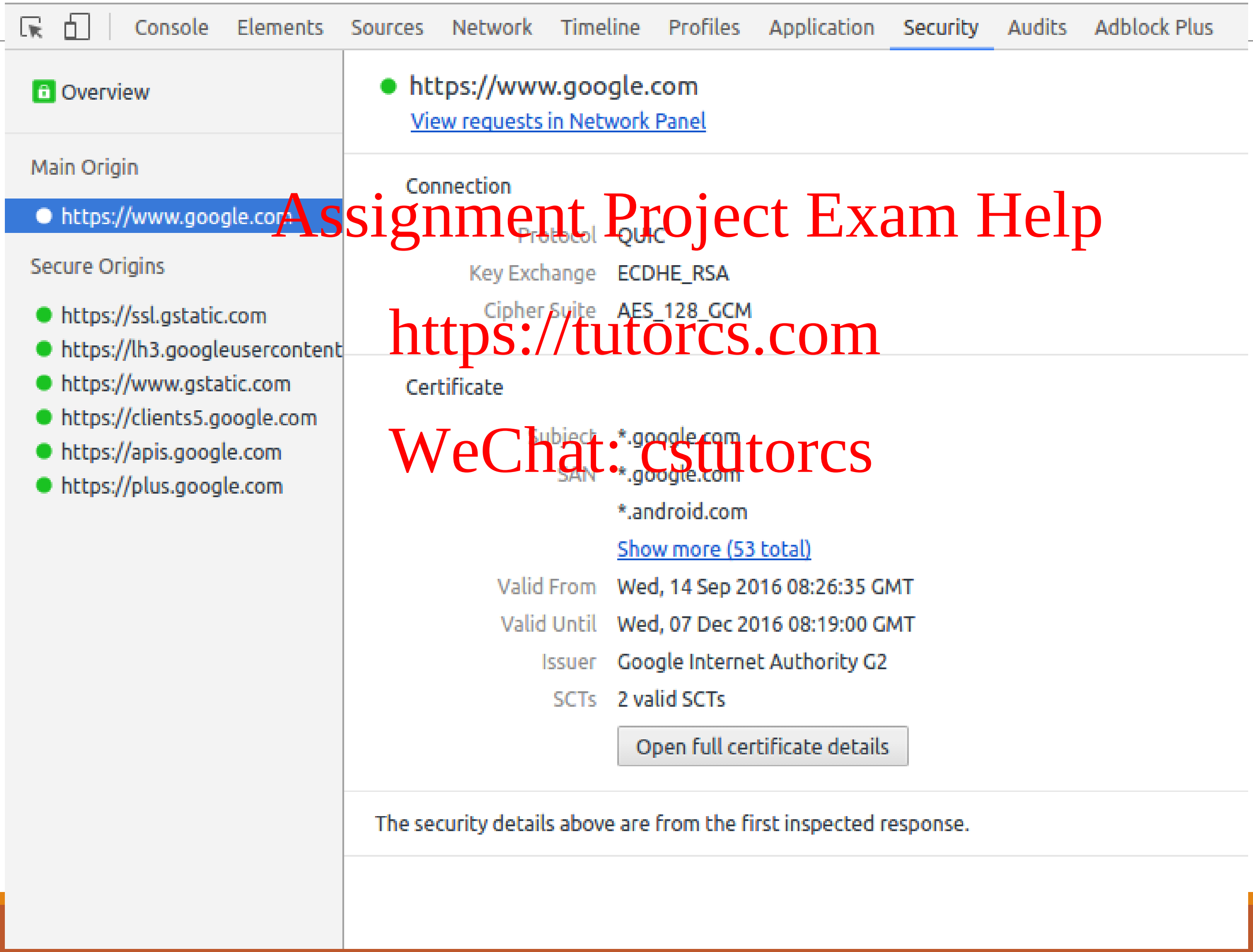
DHE-RSA-AES256-SHA

Ephemeral Key Exchange

Identity Authentication

Data Transfer Cipher

Message Digest



Main Origin

- https://www.google.com

Secure Origins

- https://ssl.gstatic.com
- https://lh3.googleusercontent
- https://www.gstatic.com
- https://clients5.google.com
- https://apis.google.com
- https://plus.google.com

https://www.google.com

Connection

Protocol	QUIC
Key Exchange	ECDHE_RSA
Cipher Suite	AES_128_GCM

Certificate

Subject	*.google.com
SAN	*.google.com *.android.com Show more (53 total)
Valid From	Wed, 14 Sep 2016 08:26:35 GMT
Valid Until	Wed, 07 Dec 2016 08:19:00 GMT
Issuer	Google Internet Authority G2
SCTs	2 valid SCTs

The security details above are from the first inspected response.

Goals

✓ Confidentiality

✓ Message Integrity

✗ Authentication

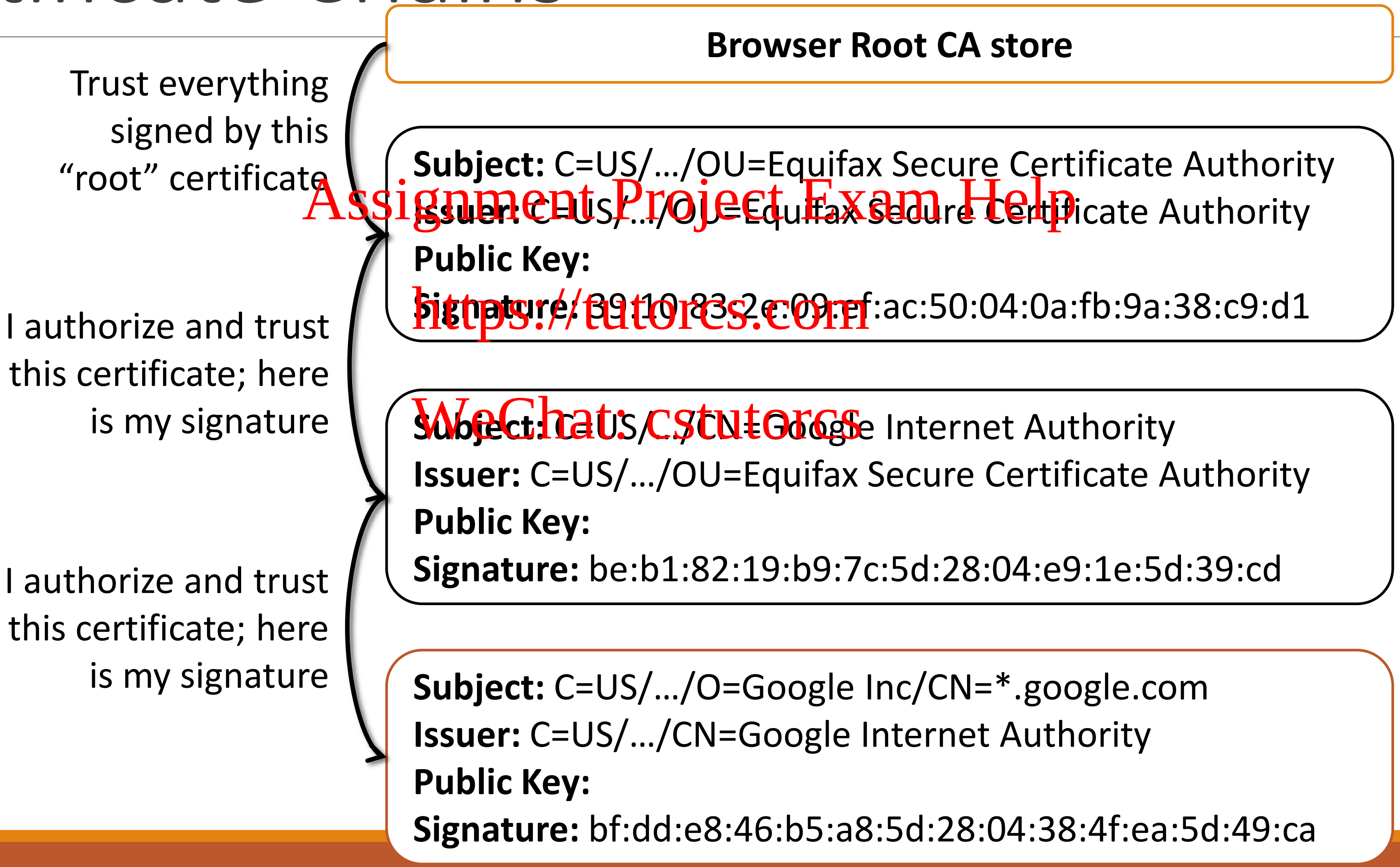
X509 Certificates

Subject: C=US/O=Google Inc/CN=www.google.com
Issuer: C=US/O=Google Inc/CN=Google Internet Authority
Serial Number: 01:b1:04:17:be:22:48:b4:8e:1e:8b:a0:73:c9:ac:83
Expiration Period: Jul 12 2010 - Jul 19 2012
Public Key Algorithm: rsaEncryption
Public Key: 43:1d:53:2e:09:ef:dc:50:54:0a:fb:9a:f0:fa:14:58:ad:a0:81:b0:3d
7c:be:b1:82:19:b9:7c3:8:04:e9:1e5d:b5:80:af:d4:a0:81:b0:b0:68:5b:a4:a4
:ff:b5:8a:3a:a2:29:e2:6c:7c3:8:04:e9:1e5d:b5:7c3:8:04:e9:39:23:46

Signature Algorithm: sha1WithRSAEncryption

Signature: 39:10:83:2e:09:ef:ac:50:04:0a:fb:9a:f0:fa:14:58:ad:a0:81:b0:3d
7c:be:b1:82:19:b9:7c3:8:04:e9:1e5d:b5:80:af:d4:a0:81:b0:b0:68:5b:a4:a4
:ff:b5:8a:3a:a2:29:e2:6c:7c3:8:04:e9:1e5d:b5:7c3:8:04:e9:1e:5d:b5

Certificate Chains



Browser Root CA store

Trust everything signed by this “root” certificate

Subject: C=US/…/OU=Equifax Secure Certificate Authority
Issuer: C=US/…/OU=Equifax Secure Certificate Authority
Public Key:
Signature: 39:10:83:2e:09:ef:ac:50:04:0a:fb:9a:38:c9:d1

I authorize and trust this certificate; here is my signature

Subject: C=US/…/CN=Google Internet Authority
Issuer: C=US/…/OU=Equifax Secure Certificate Authority
Public Key:
Signature: be:b1:82:19:b9:7c:5d:28:04:e9:1e:5d:39:cd

I authorize and trust this certificate; here is my signature

Subject: C=US/…/O=Google Inc/CN=*.google.com
Issuer: C=US/…/CN=Google Internet Authority
Public Key:
Signature: bf:dd:e8:46:b5:a8:5d:28:04:38:4f:ea:5d:49:ca

Goals

✓ Confidentiality (Symmetric Crypto)

✓ Message Integrity (HMACs)

✓ Authentication (Public Key Crypto)

Certificate Authority Ecosystem

Each browser trusts a set of CAs

- CAs can sign certificates for new CAs
- CAs can sign certificates for any web site

If a single CA is compromised, then the entire system is compromised

We ultimately place our complete trust of the Internet in the weakest CA

Immediate Concerns

Nobody has any idea who these CAs are…

1,500+ known browser trusted CAs

History of CAs being hacked (e.g. Diginotar)

Oooops, Korea gave every elementary school, library, and agency a CA certificate (1,324)

- Luckily invalid due to a higher-up constraint

Getting a Certificate

Certificates are free and easy to get!

https://letsencrypt.org/

Identity validated via e-mail in whois, or proving control over a certain webpage on the domain

- What can go wrong?

Setting up TLS manually is hard. People are terrible at it!

DigiNotar

DigiNotar ***was*** a Dutch Certificate Authority

On June 10, 2011, **`*.google.com`** cert was issued to an attacker and subsequently used to orchestrate MITM attacks in Iran

Nobody noticed the attack until someone found the certificate in the wild… and posted to *pastebin*

DigiNotar Contd.

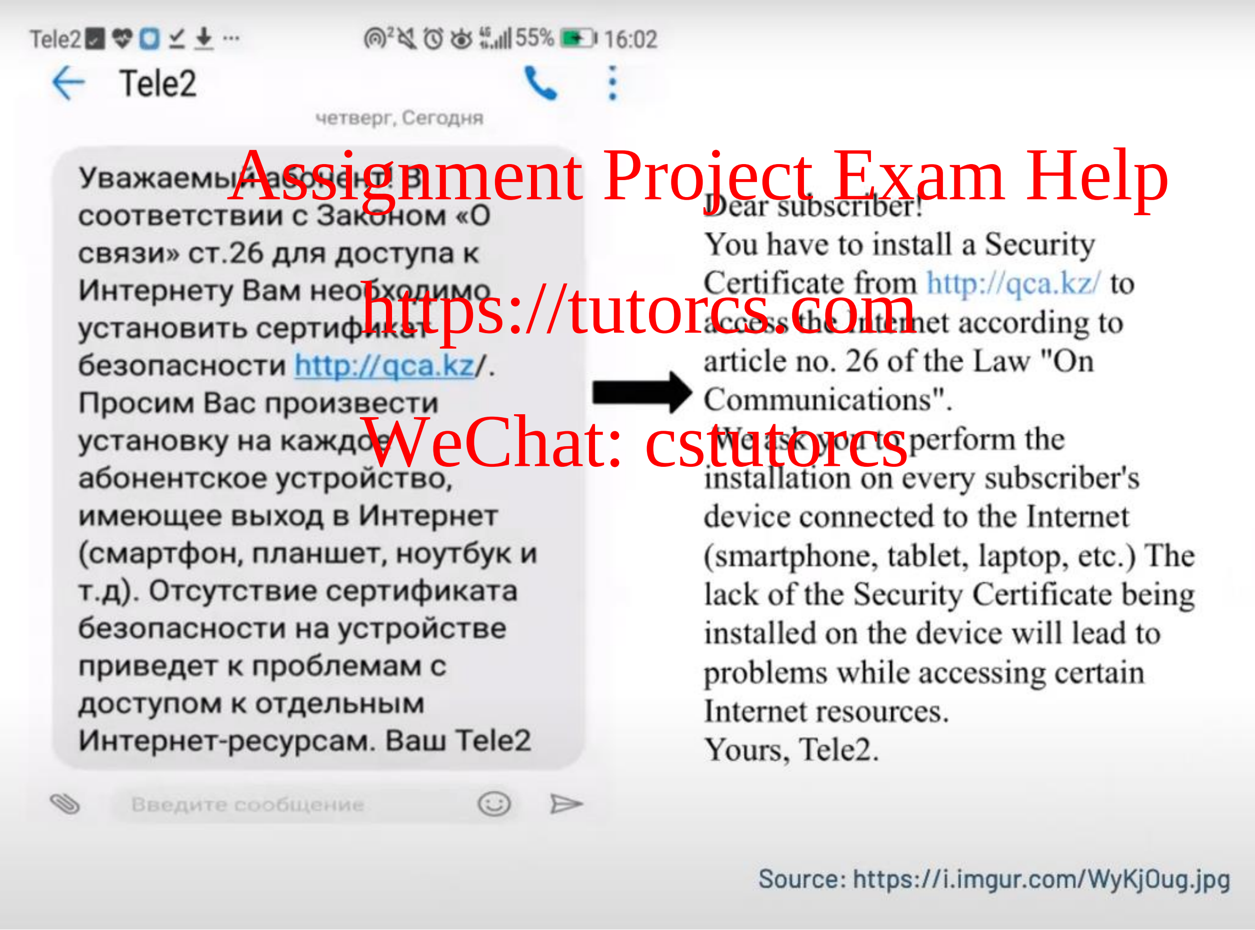
DigiNotar later admitted that dozens of fraudulent certificates were created

Google, Microsoft, Apple and Mozilla all revoked the root Diginotar certificate

Dutch Government took over Diginotar

Diginotar went bankrupt and died

Kazakhstan TLS MITM

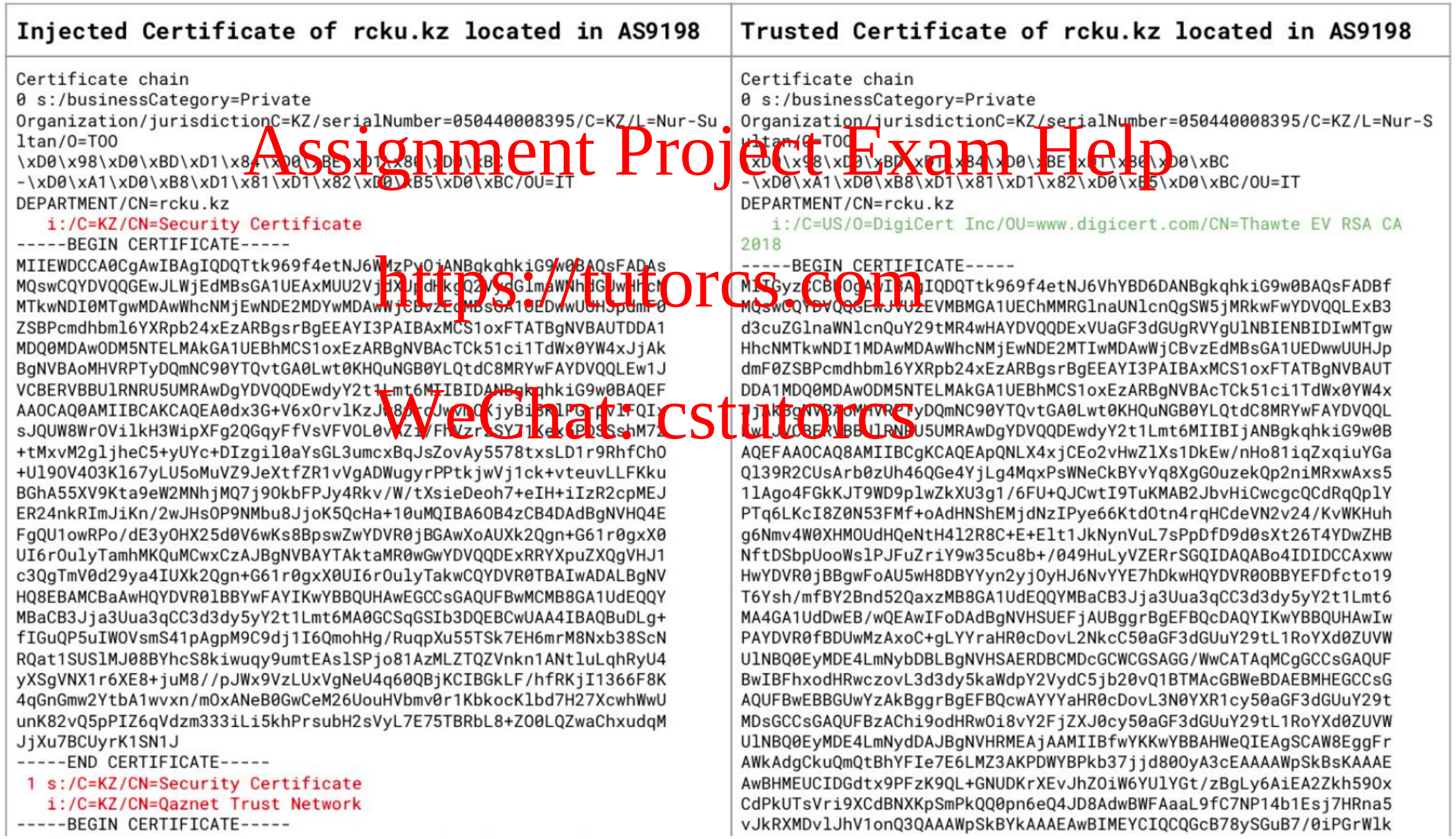


Уважаемый абонент! В соответствии с Законом «О связи» ст.26 для доступа к Интернету Вам необходимо установить сертификат безопасности http://qca.kz/. Просим Вас произвести установку на каждое абонентское устройство, имеющее выход в Интернет (смартфон, планшет, ноутбук и т.д). Отсутствие сертификата безопасности на устройстве приведет к проблемам с доступом к отдельным Интернет-ресурсам. Ваш Tele2

Dear subscriber!
You have to install a Security Certificate from http://qca.kz/ to access the Internet according to article no. 26 of the Law "On Communications".
We ask you to perform the installation on every subscriber's device connected to the Internet (smartphone, tablet, laptop, etc.) The lack of the Security Certificate being installed on the device will lead to problems while accessing certain Internet resources.
Yours, Tele2.

Source: https://i.imgur.com/WyKjOug.jpg

Kazakhstan TLS MITM



Injected Certificate of rcku.kz located in AS9198

```
Certificate chain
0 s:/businessCategory=Private
Organization/jurisdictionC=KZ/serialNumber=050440008395/C=KZ/L=Nur-Su
ltan/O=TOO
\xD0\x98\xD0\xBD\xD1\x84\xD0\xBE\xD1\x80\xD0\xBC
-\xD0\xA1\xD0\xB8\xD1\x81\xD1\x82\xD0\xB5\xD0\xBC/OU=IT
DEPARTMENT/CN=rcku.kz
   i:/C=KZ/CN=Security Certificate
-----BEGIN CERTIFICATE-----
MIIEWDCCA0CgAwIBAgIQDQTtk969f4etNJ6WMzPyOjANBgkqhkiG9w0BAQsFADAs
MQswCQYDVQQGEwJLWjEdMBsGA1UEAxMUU2VjdXJpdHkgQ2VydGlmaWNhdGUwHhcN
MTkwNDI0MTgwMDAwWhcNMjEwNDE2MDYwMDAwWjCBvzEdMBsGA1UEDwwUUHJpdmF0
ZSBPcmdhbml6YXRpb24xEzARBgsrBgEEAYI3PAIBAxMCS1oxFTATBgNVBAUTDDA1
MDQ0MDAwODM5NTELMAkGA1UEBhMCS1oxEzARBgNVBAcTCk51ci1TdWx0YW4xJjAk
BgNVBAoMHVRPTyDQmNC90YTQvtGA0Lwt0KHQuNGB0YLQtdC8MRYwFAYDVQQLEw1J
VCBERVBBUlRNRU5UMRAwDgYDVQQDEwdyY2t1Lmt6MIIBIDANBgkqhkiG9w0BAQEF
AAOCAQ0AMIIBCAKCAQEA0dx3G+V6xOrvlKzJw8prdJwvnOKjyBi6RlPGrpV1FQIx
sJQUW8WrOVilkH3WipXFg2QGqyFfVsVFVOL0vXZi/FHVzrZSY71KekPQSShM72
+tMxvM2gljheC5+yUYc+DIzgil0aYsGL3umcxBqJsZovAy5578txsLD1r9RhfChO
+Ul9OV4O3Kl67yLU5oMuVZ9JeXtfZR1vVgADWugyrPPtkjwVj1ck+vteuvLLFKku
BGhA55XV9Kta9eW2MNhjMQ7j9OkbFPJy4Rkv/W/tXsieDeoh7+eIH+iIzR2cpMEJ
ER24nkRImJiKn/2wJHsOP9NMbu8JjoK5QcHa+10uMQIBA6OB4zCB4DAdBgNVHQ4E
FgQU1owRPo/dE3yOHX25d0V6wKs8BpswZwYDVR0jBGAwXoAUXk2Qgn+G61r0gxX0
UI6rOulyTamhMKQuMCwxCzAJBgNVBAYTAktaMR0wGwYDVQQDExRRYXpuZXQgVHJ1
c3QgTmV0d29ya4IUXk2Qgn+G61r0gxX0UI6rOulyTakwCQYDVR0TBAIwADALBgNV
HQ8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMB8GA1UdEQQY
MBaCB3Jja3Uua3qCC3d3dy5yY2t1Lmt6MA0GCSqGSIb3DQEBCwUAA4IBAQBuDLg+
fIGuQP5uIWOVsmS41pAgpM9C9dj1I6QmohHg/RuqpXu55TSk7EH6mrM8Nxb38ScN
RQat1SUSlMJ08BYhcS8kiwuqy9umtEAslSPjo81AzMLZTQZVnkn1ANtluLqhRyU4
yXSgVNX1r6XE8+juM8//pJWx9VzLUxVgNeU4q60QBjKCIBGkLF/hfRKjI1366F8K
4qGnGmw2YtbA1wvxn/mOxANeB0GwCeM26UouHVbmv0r1KbkocKlbd7H27XcwhWwU
unK82vQ5pPIZ6qVdzm333iLi5khPrsubH2sVyL7E75TBRbL8+ZO0LQZwaChxudqM
JjXu7BCUyrK1SN1J
-----END CERTIFICATE-----
 1 s:/C=KZ/CN=Security Certificate
   i:/C=KZ/CN=Qaznet Trust Network
-----BEGIN CERTIFICATE-----
```

Trusted Certificate of rcku.kz located in AS9198

```
Certificate chain
0 s:/businessCategory=Private
Organization/jurisdictionC=KZ/serialNumber=050440008395/C=KZ/L=Nur-S
ultan/O=TOO
\xD0\x98\xD0\xBD\xD1\x84\xD0\xBE\xD1\x80\xD0\xBC
-\xD0\xA1\xD0\xB8\xD1\x81\xD1\x82\xD0\xB5\xD0\xBC/OU=IT
DEPARTMENT/CN=rcku.kz
   i:/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=Thawte EV RSA CA
2018
-----BEGIN CERTIFICATE-----
MIIGyzCCBbOgAwIBAgIQDQTtk969f4etNJ6VhYBD6DANBgkqhkiG9w0BAQsFADBf
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMR4wHAYDVQQDExVUaGF3dGUgRVYgUlNBIENBIDIwMTgw
HhcNMTkwNDI1MDAwMDAwWhcNMjEwNDE2MTIwMDAwWjCBvzEdMBsGA1UEDwwUUHJp
dmF0ZSBPcmdhbml6YXRpb24xEzARBgsrBgEEAYI3PAIBAxMCS1oxFTATBgNVBAUT
DDA1MDQ0MDAwODM5NTELMAkGA1UEBhMCS1oxEzARBgNVBAcTCk51ci1TdWx0YW4x
JjAkBgNVBAoMHVRPTyDQmNC90YTQvtGA0Lwt0KHQuNGB0YLQtdC8MRYwFAYDVQQL
Ew1JVCBERVBBUlRNRU5UMRAwDgYDVQQDEwdyY2t1Lmt6MIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEApQNLX4xjCEo2vHwZlXs1DkEw/nHo81iqZxqiuYGa
Ql39R2CUsArb0zUh46QGe4YjLg4MqxPsWNeCkBYvYq8XgGOuzekQp2niMRxwAxs5
1lAgo4FGkKJT9WD9plwZkXU3g1/6FU+QJCwtI9TuKMAB2JbvHiCwcgcQCdRqQplY
PTq6LKcI8Z0N53FMf+oAdHNShEMjdNzIPye66KtdOtn4rqHCdeVN2v24/KvWKHuh
g6Nmv4W0XHMOUdHQeNtH4l2R8C+E+Elt1JkNynVuL7sPpDfD9d0sXt26T4YDwZHB
NftDSbpUooWslPJFuZriY9w35cu8b+/049HuLyVZERrSGQIDAQABo4IDIDCCAxww
HwYDVR0jBBgwFoAU5wH8DBYYyn2yjOyHJ6NvYYE7hDkwHQYDVR0OBBYEFDfcto19
T6Ysh/mfBY2Bnd52QaxzMB8GA1UdEQQYMBaCB3Jja3Uua3qCC3d3dy5yY2t1Lmt6
MA4GA1UdDwEB/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIw
PAYDVR0fBDUwMzAxoC+gLYYraHR0cDovL2NkcC50aGF3dGUuY29tL1RoYXd0ZUVW
UlNBQ0EyMDE4LmNybDBLBgNVHSAERDBCMDcGCWCGSAGG/WwCATAqMCgGCCsGAQUF
BwIBFhxodHRwczovL3d3dy5kaWdpY2VydC5jb20vQ1BTMAcGBWeBDAEBMHEGCCsG
AQUFBwEBBGUwYzAkBggrBgEFBQcwAYYYaHR0cDovL3N0YXR1cy50aGF3dGUuY29t
MDsGCCsGAQUFBzAChi9odHRwOi8vY2FjZXJ0cy50aGF3dGUuY29tL1RoYXd0ZUVW
UlNBQ0EyMDE4LmNydDAJBgNVHRMEAjAAMIIBfwYKKwYBBAHWeQIEAgSCAW8EggFr
AWkAdgCkuQmQtBhYFIe7E6LMZ3AKPDWYBPkb37jjd80OyA3cEAAAAWpSkBsKAAAE
AwBHMEUCIDGdtx9PFzK9QL+GNUDKrXEvJhZOiW6YUlYGt/zBgLy6AiEA2Zkh59Ox
CdPkUTsVri9XCdBNXKpSmPkQQ0pn6eQ4JD8AdwBWFAaaL9fC7NP14b1Esj7HRna5
vJkRXMDvlJhV1onQ3QAAAWpSkBYkAAAEAwBIMEYCIQCQGcB78ySGuB7/0iPGrWlk
```

Kazakhstan TLS MITM

Domains impacted:

allo.google.com, android.com, cdninstagram.com, dns.google.com, docs.google.com, encrypted.google.com, facebook.com, goo.gl, google.com, groups.google.com, hangouts.google.com, instagram.com, mail.google.com, mail.ru, messages.android.com, messenger.com, news.google.com, ok.ru, picasa.google.com, plus.google.com, rukoeb.com, sites.google.com, sosalkino.tv, tamtam.chat, translate.google.com, twitter.com, video.google.com, vk.com, vk.me, vkuseraudio.net, vkuservideo.net, www.facebook.com, www.google.com, www.instagram.com, www.messenger.com, www.youtube.com, youtube.com

Browser response:

- Remove KZ root cert *even if user explicitly added it!*

Attack Vectors

Attack the weakest Certificate Authority

Attack browser implementations

Magically notice a bug in a key generation library that leads you to discovering all the private keys on the Internet

Attack the cryptographic primitives

- Math is hard, let’s go shopping!

TLS Attacks

User concerns

- Deploying site leaks private key
- Client users ignore HTTPS errors!

Attack (weakest) CA

- DigiNotar, Comodo, WoSign/Startcom

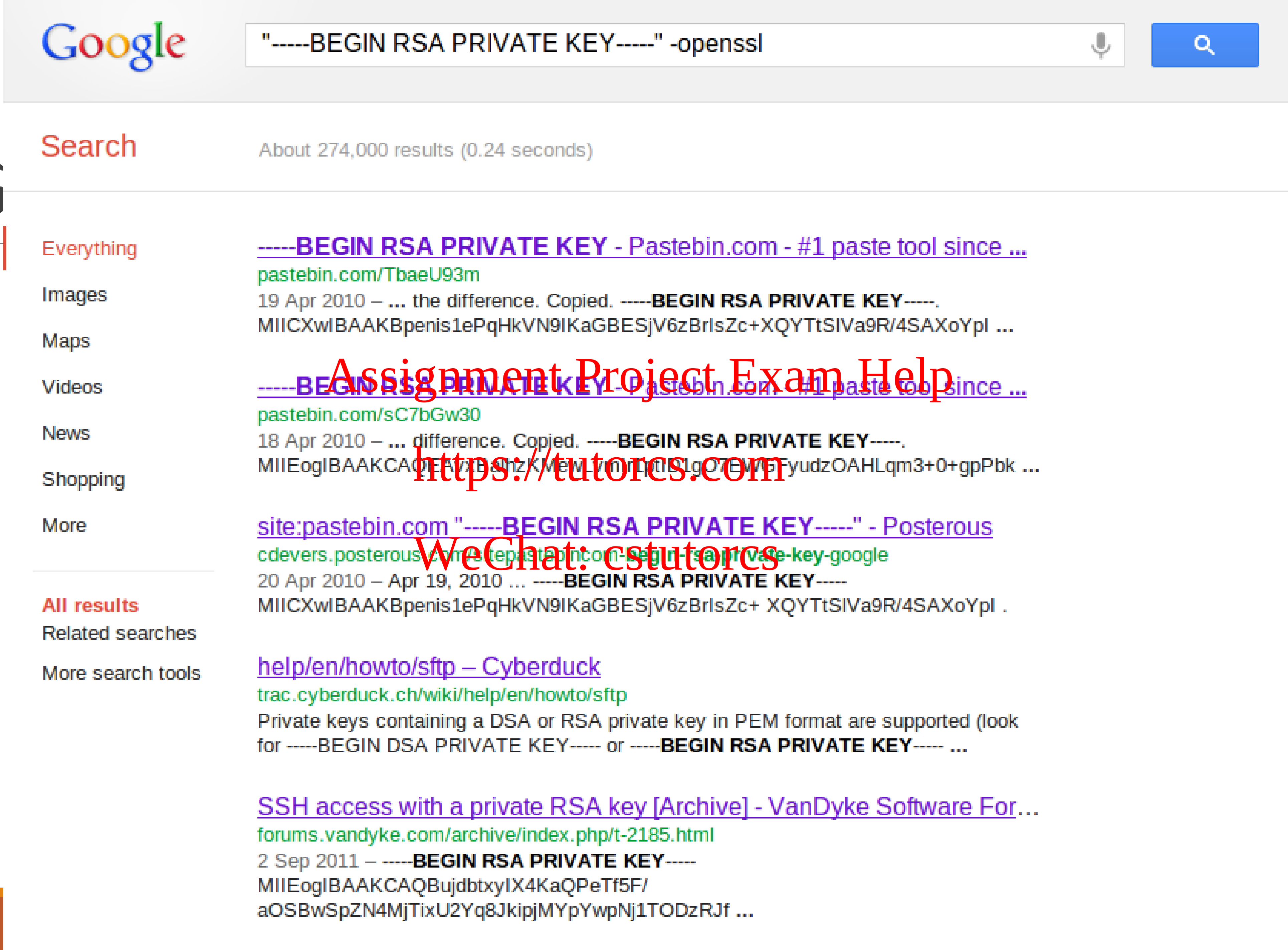
Attack Browser

- SSL Strip, Null Prefix, Padding Oracle, BEAST, CRIME, goto fail, POODLE, FREAK, LogJam, DROWN, …

Attack Server

- Heartbleed

G



"-----BEGIN RSA PRIVATE KEY-----" -openssl

About 274,000 results (0.24 seconds)

-----**BEGIN RSA PRIVATE KEY** - Pastebin.com - #1 paste tool since ...
pastebin.com/TbaeU93m
19 Apr 2010 – **...** the difference. Copied. -----**BEGIN RSA PRIVATE KEY**-----.
MIICXwIBAAKBpenis1ePqHkVN9IKaGBESjV6zBrlsZc+XQYTtSIVa9R/4SAXoYpl **...**

-----**BEGIN RSA PRIVATE KEY** - Pastebin.com - #1 paste tool since ...
pastebin.com/sC7bGw30
18 Apr 2010 – **...** difference. Copied. -----**BEGIN RSA PRIVATE KEY**-----.
MIIEogIBAAKCAQEA[illegible]FyudzOAHLqm3+0+gpPbk **...**

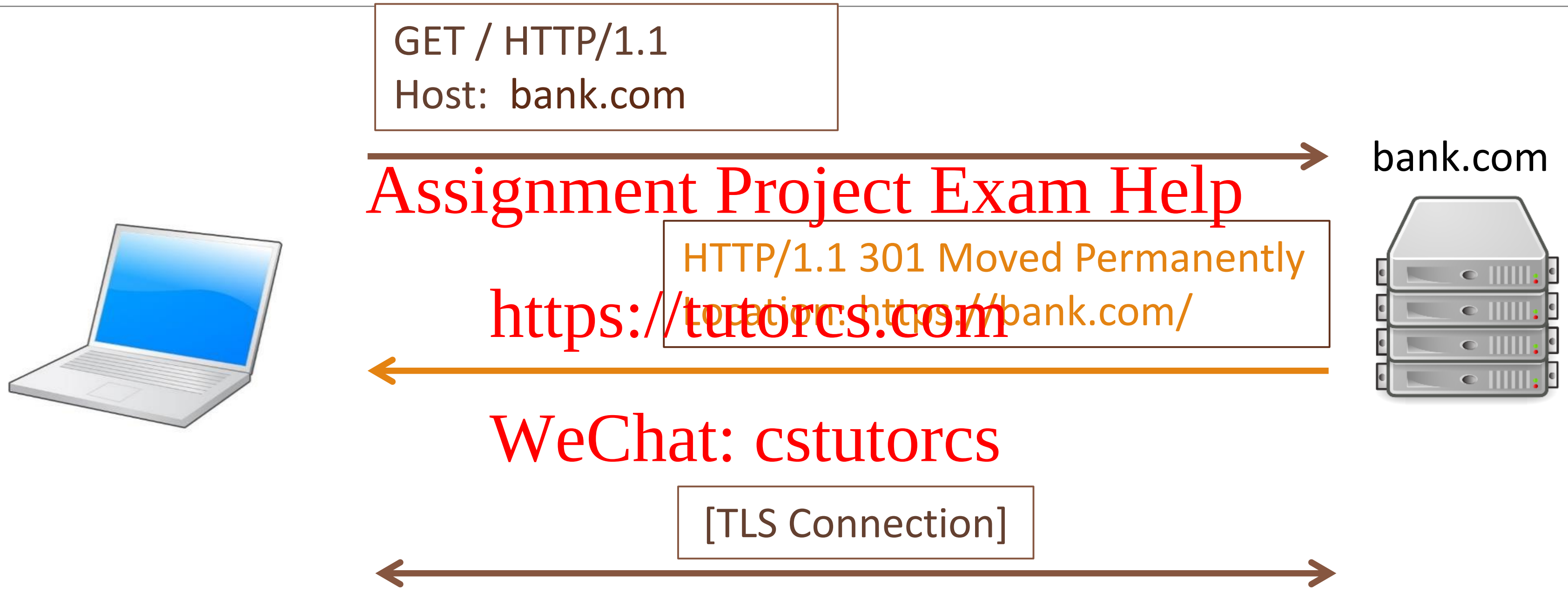
site:pastebin.com "-----**BEGIN RSA PRIVATE KEY**-----" - Posterous
cdevers.posterous.com/sitepastebincom-begin-rsa-private-key-google
20 Apr 2010 – Apr 19, 2010 ... -----**BEGIN RSA PRIVATE KEY**-----
MIICXwIBAAKBpenis1ePqHkVN9IKaGBESjV6zBrlsZc+ XQYTtSIVa9R/4SAXoYpl .

help/en/howto/sftp – Cyberduck
trac.cyberduck.ch/wiki/help/en/howto/sftp
Private keys containing a DSA or RSA private key in PEM format are supported (look for -----BEGIN DSA PRIVATE KEY----- or -----**BEGIN RSA PRIVATE KEY**----- **...**

SSH access with a private RSA key [Archive] - VanDyke Software For...
forums.vandyke.com/archive/index.php/t-2185.html
2 Sep 2011 – -----**BEGIN RSA PRIVATE KEY**-----
MIIEogIBAAKCAQBujdbtxyIX4KaQPeTf5F/
aOSBwSpZN4MjTixU2Yq8JkipjMYpYwpNj1TODzRJf **...**

SSL Strip

Discovered by Moxie Marlinspike, 2009



GET / HTTP/1.1
Host: bank.com

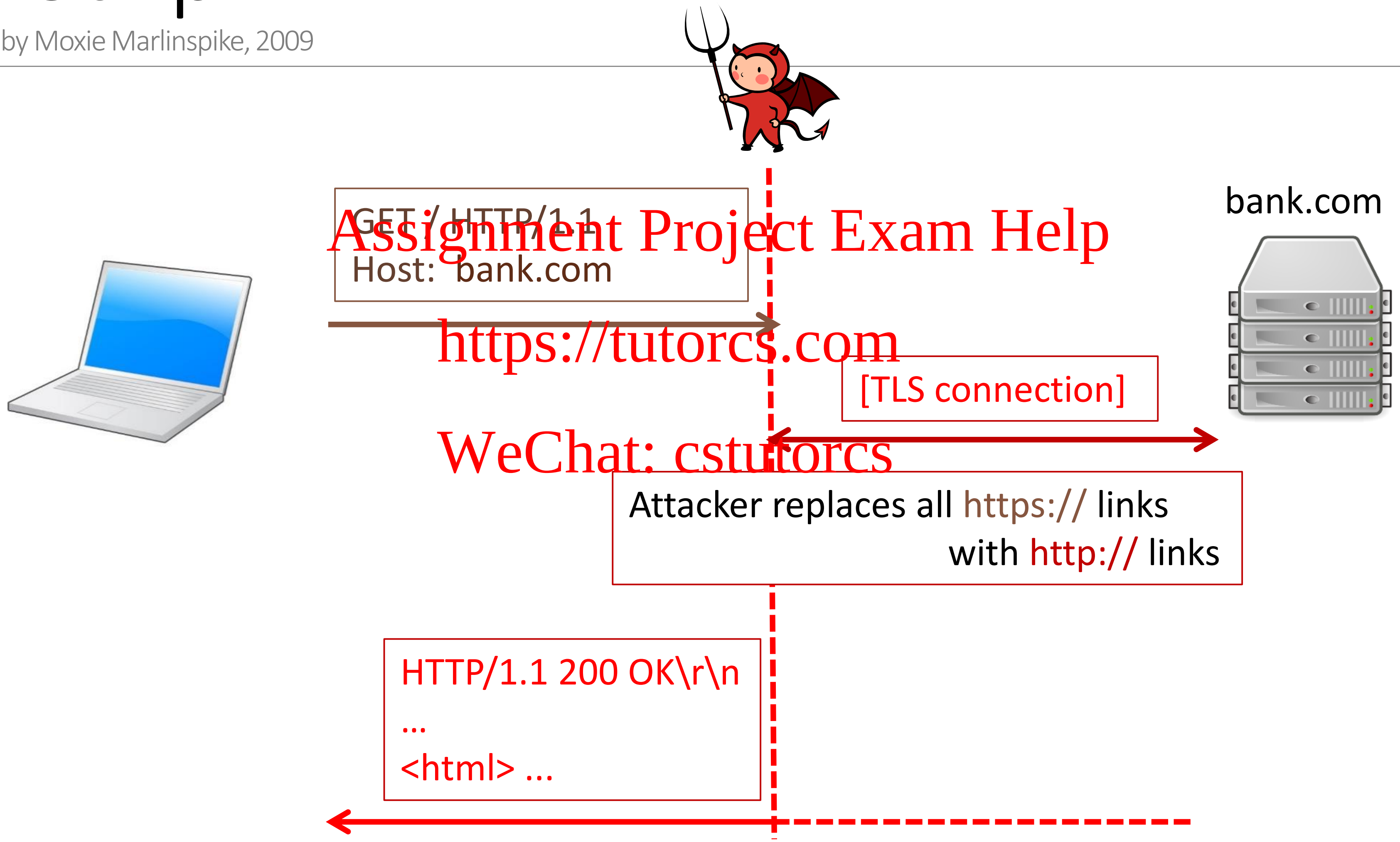
bank.com

HTTP/1.1 301 Moved Permanently
Location: https://bank.com/

[TLS Connection]

SSL Strip

Discovered by Moxie Marlinspike, 2009



GET / HTTP/1.1
Host: bank.com

[TLS connection]

Attacker replaces all https:// links
with http:// links

HTTP/1.1 200 OK\r\n
…
<html> …

Null Termination Attack

Discovered by Moxie Marlinspike, 2009

ASN.1 utilizes Pascal-style strings

Web browsers utilize use C-style strings

`gmail.com.evil.com` ✓

`gmail.com\0.evil.com` ✓

```
strcmp(“gmail.com\0.evil.com”, “gmail.com”) == 0
```

BEAST attack

Discovered by Thai Duong and Juliano Rizzo, 2011

“Browser Exploit Against SSL/TLS”

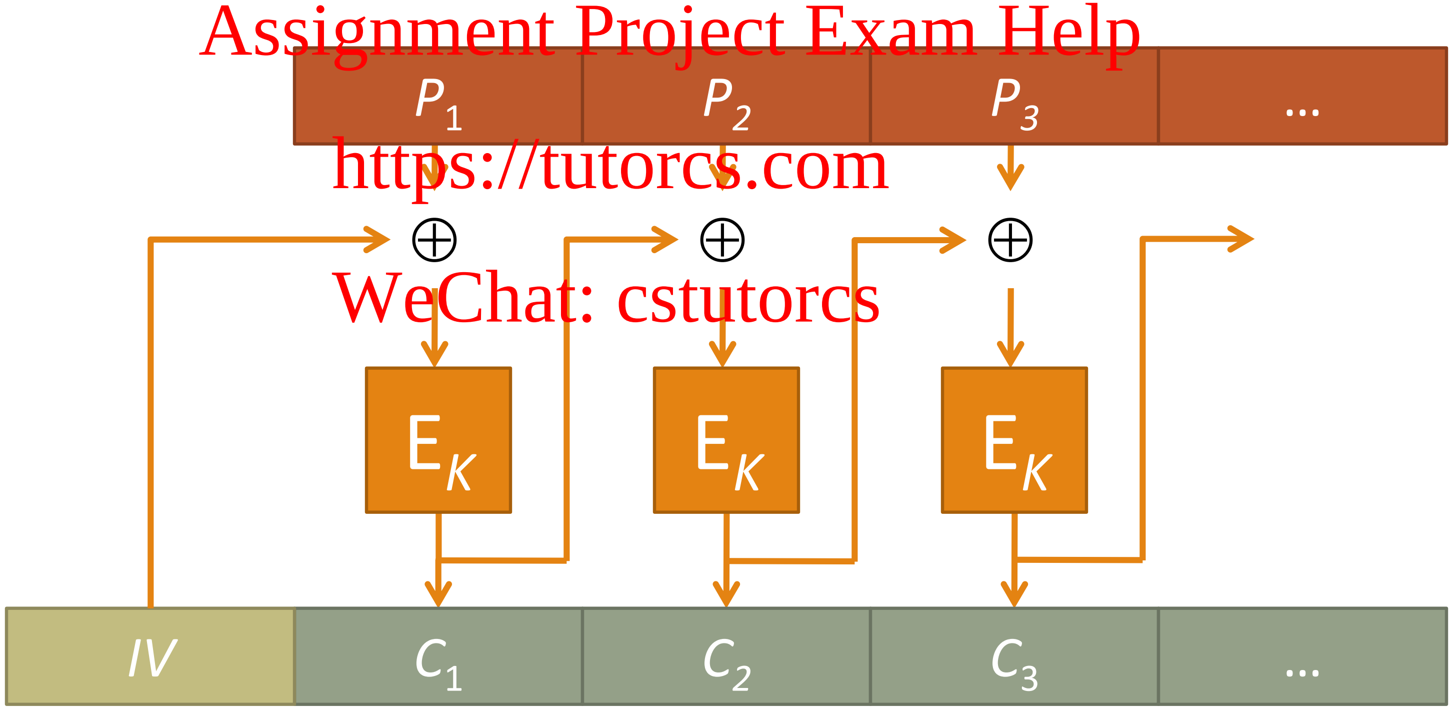
Chosen Plaintext attack against CBC-mode

Attacker can:

- Observe Alice’s Ciphertext
- Make Alice to send **secret plaintext** P over TLS
    - E.g. HTTP Cookie
- Make Alice to send **arbitrary plaintext** over same TLS session

CBC: Cipher-Block Chaining Mode

$C_i := E(K, P_i \oplus C_{i-1})$ for $i = 1, \ldots, n$



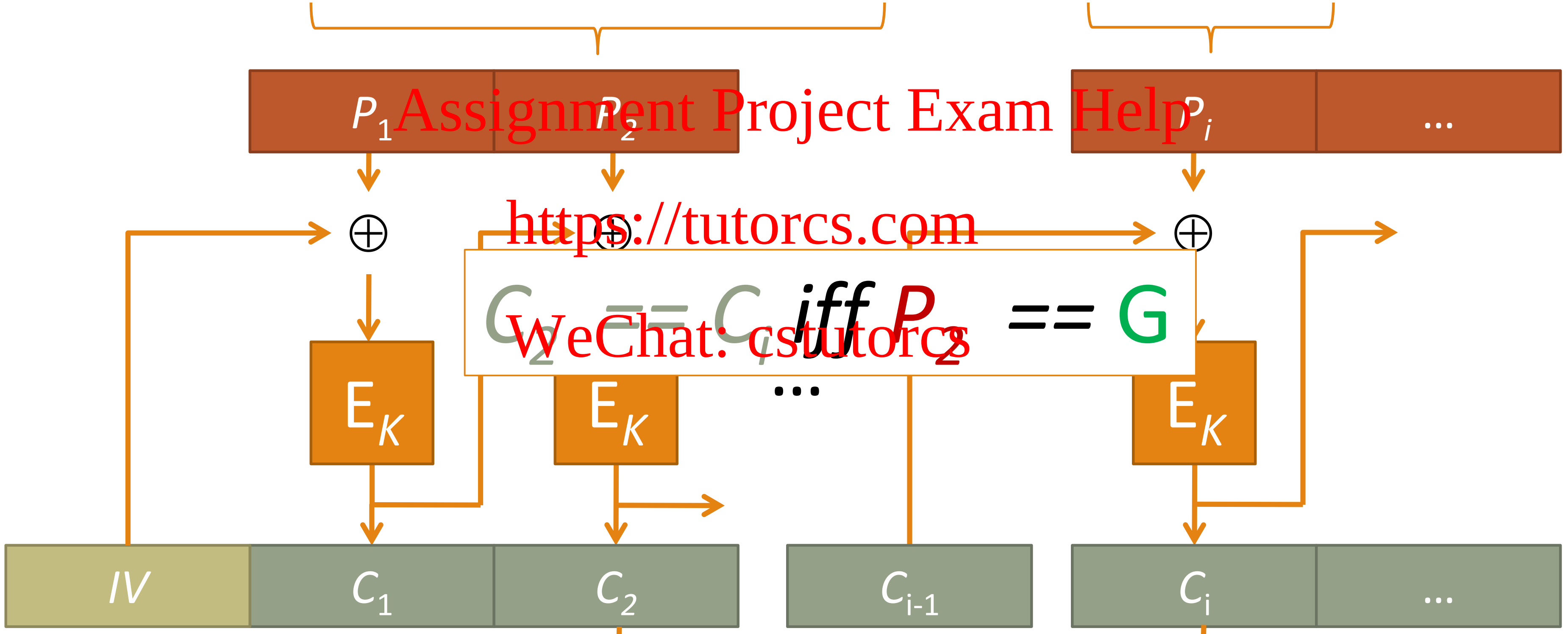
BEAST attack

Secret plaintext:

`Cookie: secret=a26b3f8e…`

Attacker-chosen plaintext:

$(C_{i-1} \oplus C_1 \oplus G)$



$C_2 == C_i$ *iff* $P_2 == G$

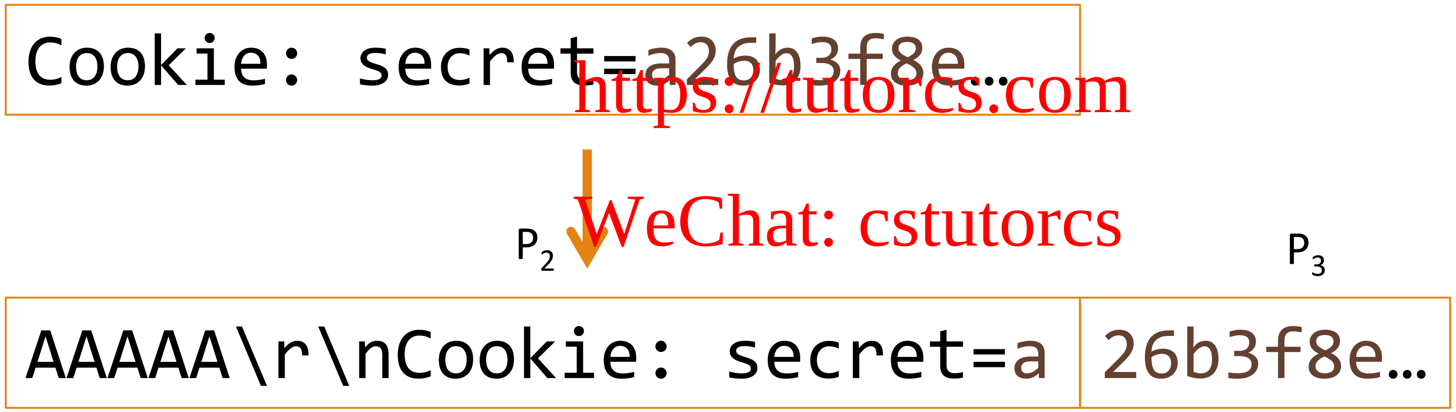
$C_2 = E_K(C_1 \oplus P_2)$

$C_i = E_K(C_{i-1} \oplus C_{i-1} \oplus C_1 \oplus G) = E_K(C_1 \oplus G)$

BEAST attack

Problem: Attacker has to guess G entirely

Solution: force part of P_2 to be known padding!



```
Cookie: secret=a26b3f8e…
```

P_2 P_3

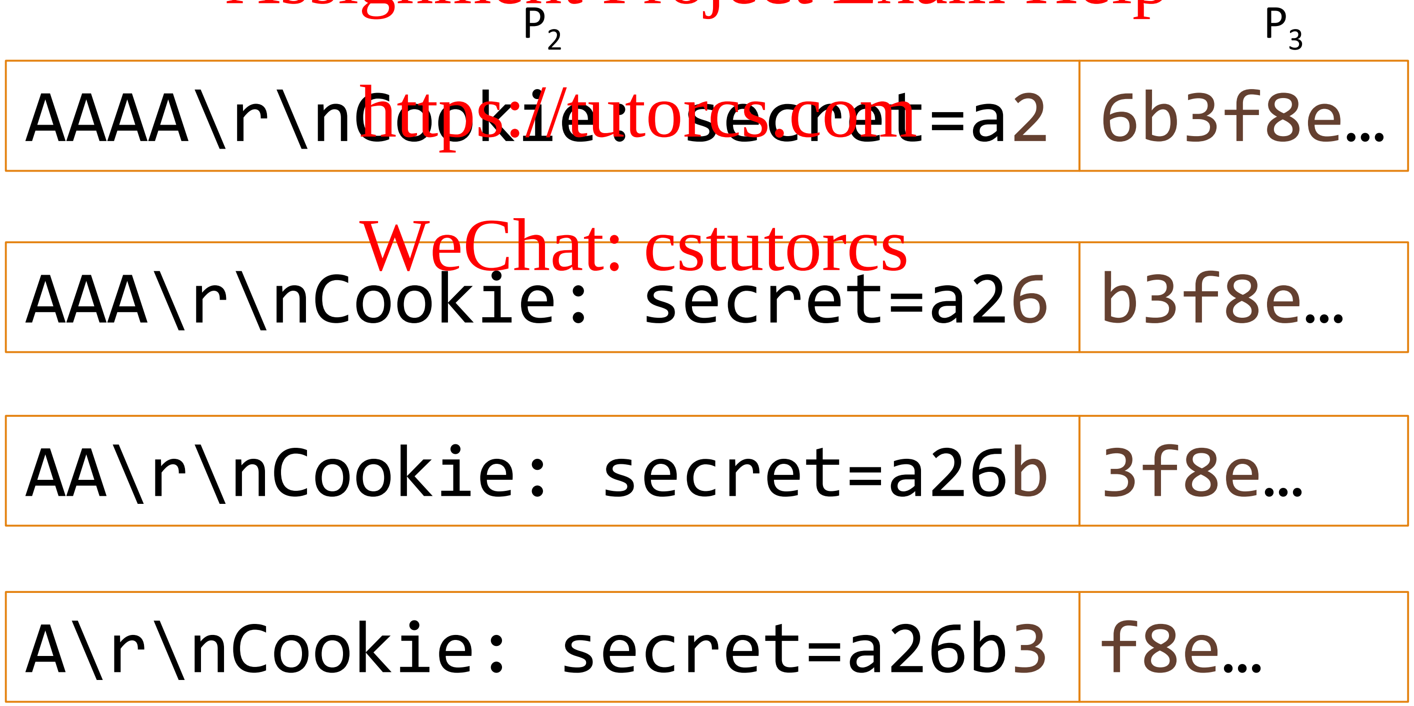
```
AAAAA\r\nCookie: secret=a 26b3f8e…
```

Only have to guess 1-byte now!

- 256 guesses and we’re sure to get it

BEAST attack

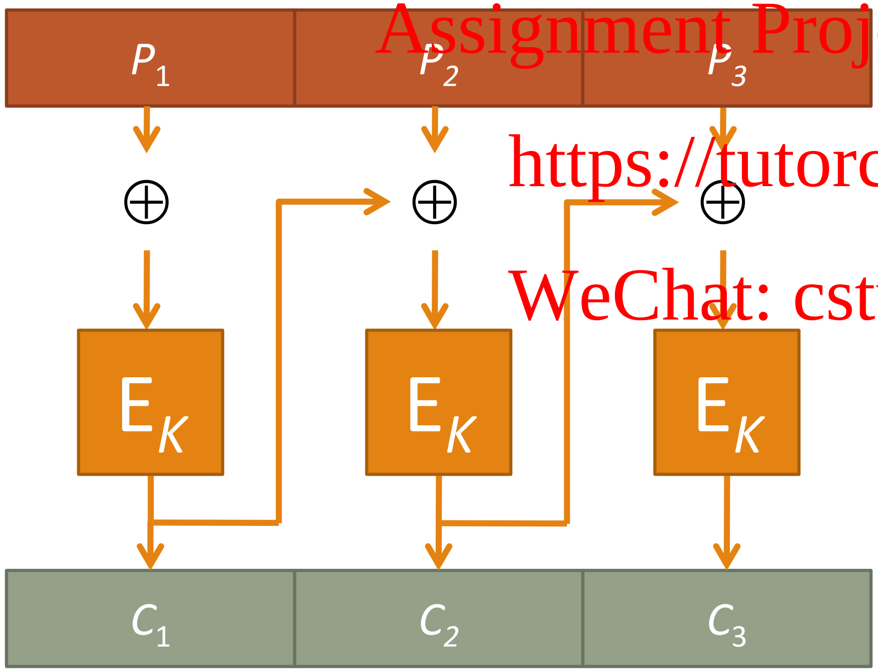
Once we guess a, we can redo the attack, with less padding:



P_2	P_3
AAAA\r\nCookie: secret=a2	6b3f8e…
AAA\r\nCookie: secret=a26	b3f8e…
AA\r\nCookie: secret=a26b	3f8e…
A\r\nCookie: secret=a26b3	f8e…

Padding oracle attack

Discovered by Serge Vaudenay, 2003



$D(C_3)$	5b	d8	99	ee
C_2	34	da	9b	ed
P_3	6f	02	02	01

MAC ERROR

$$P_1 = D(C1) \oplus IV$$
$$P_2 = D(C2) \oplus C1$$
$$P_3 = D(C3) \oplus C2$$

CRIME attack

Discovered by Thai Duong and Juliano Rizzo, 2012

Compression Ratio Info-leak Made Easy

Client compresses HTTP header

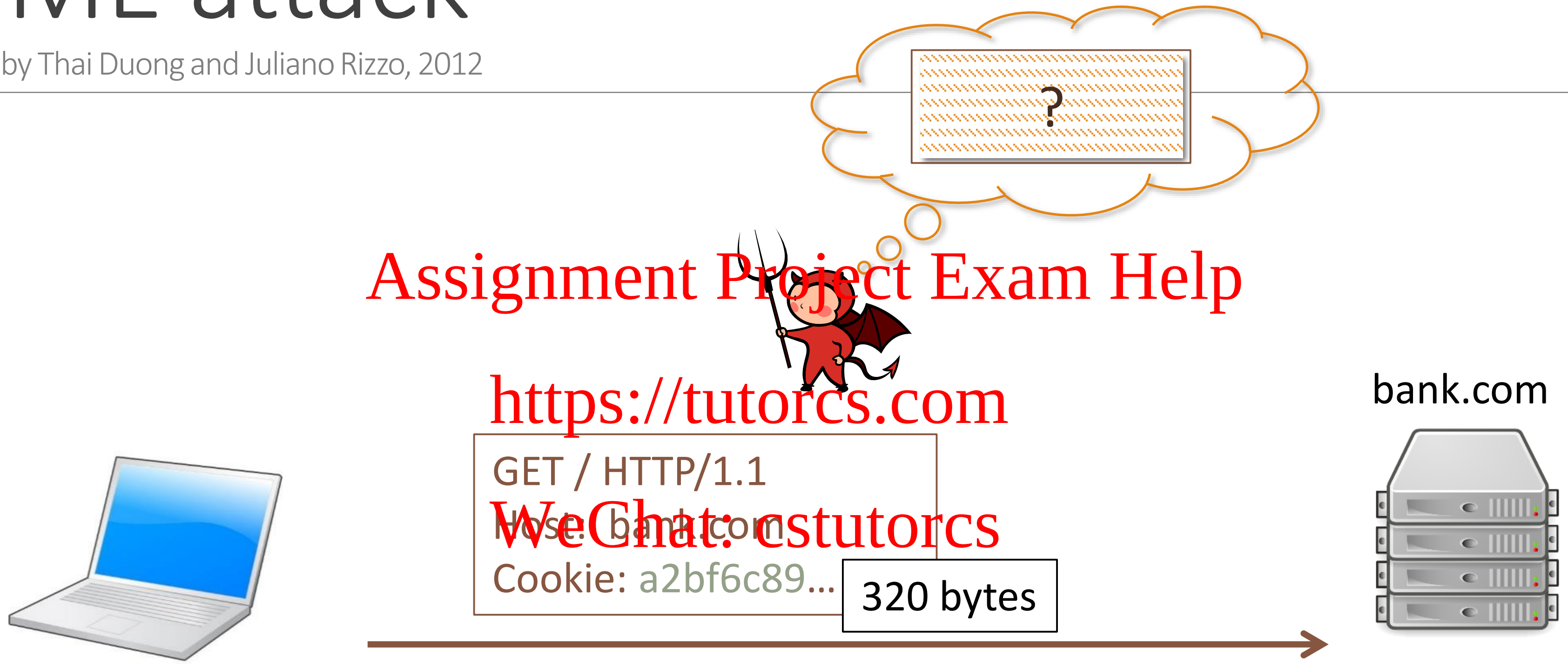
- Contains attacker controlled AND secret data!!

Attacker can:

- Make Alice send HTTPS requests with some data controlled by the attacker, some data secret
- Observe encrypted data (length)

CRIME attack

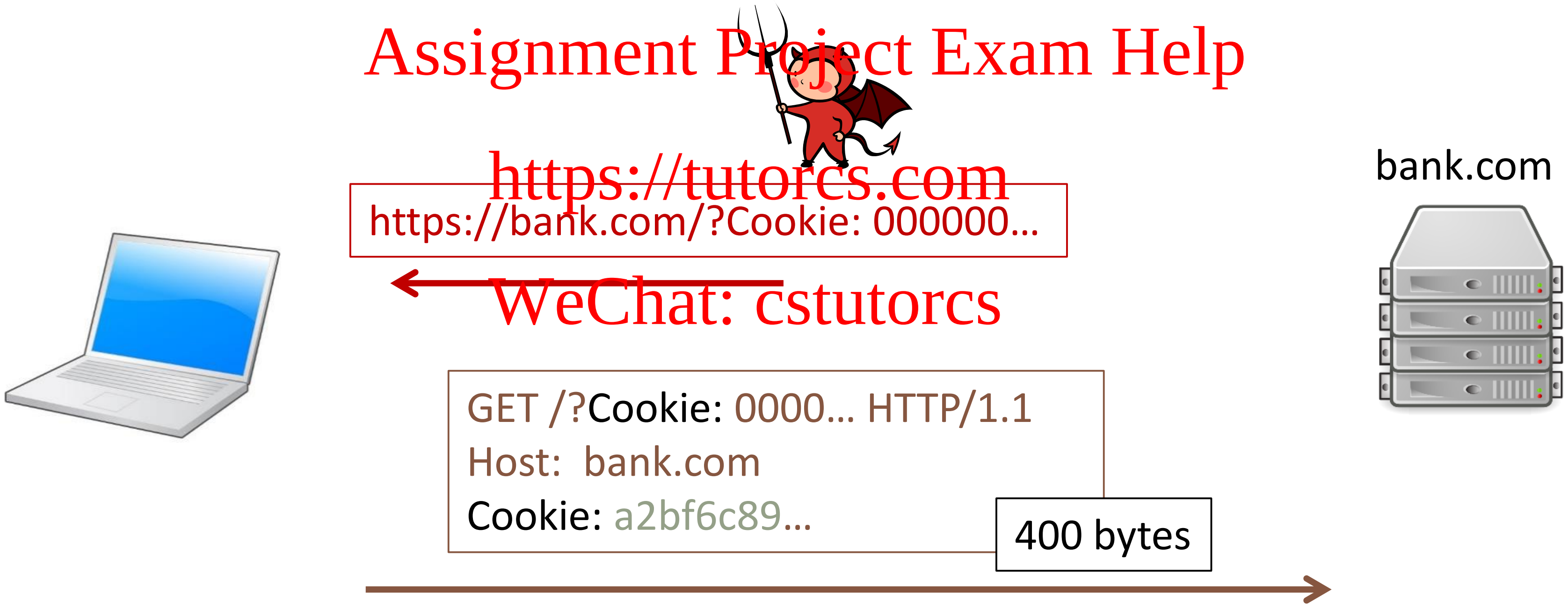
Discovered by Thai Duong and Juliano Rizzo, 2012



```
GET / HTTP/1.1
Host: bank.com
Cookie: a2bf6c89...
```

CRIME attack

Discovered by Thai Duong and Juliano Rizzo, 2012



bank.com

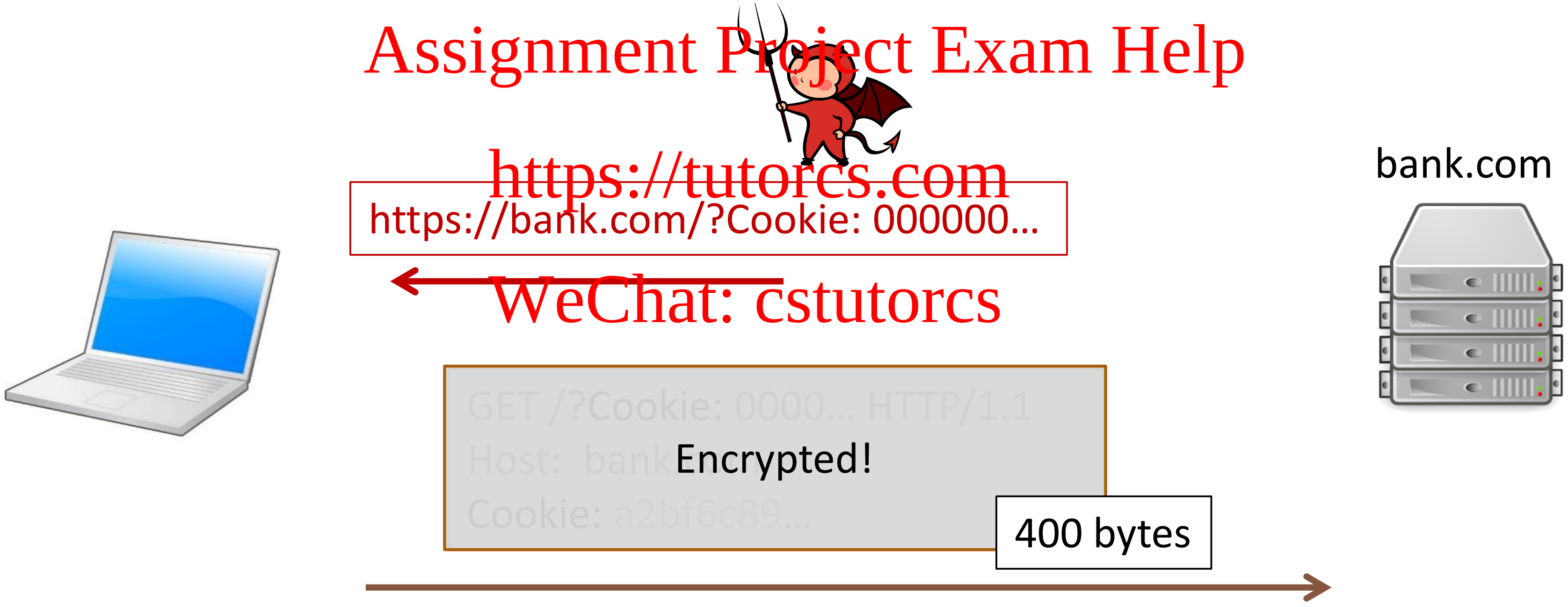
https://bank.com/?Cookie: 000000…

```
GET /?Cookie: 0000… HTTP/1.1
Host:  bank.com
Cookie: a2bf6c89…
```

400 bytes

CRIME attack

Discovered by Thai Duong and Juliano Rizzo, 2012

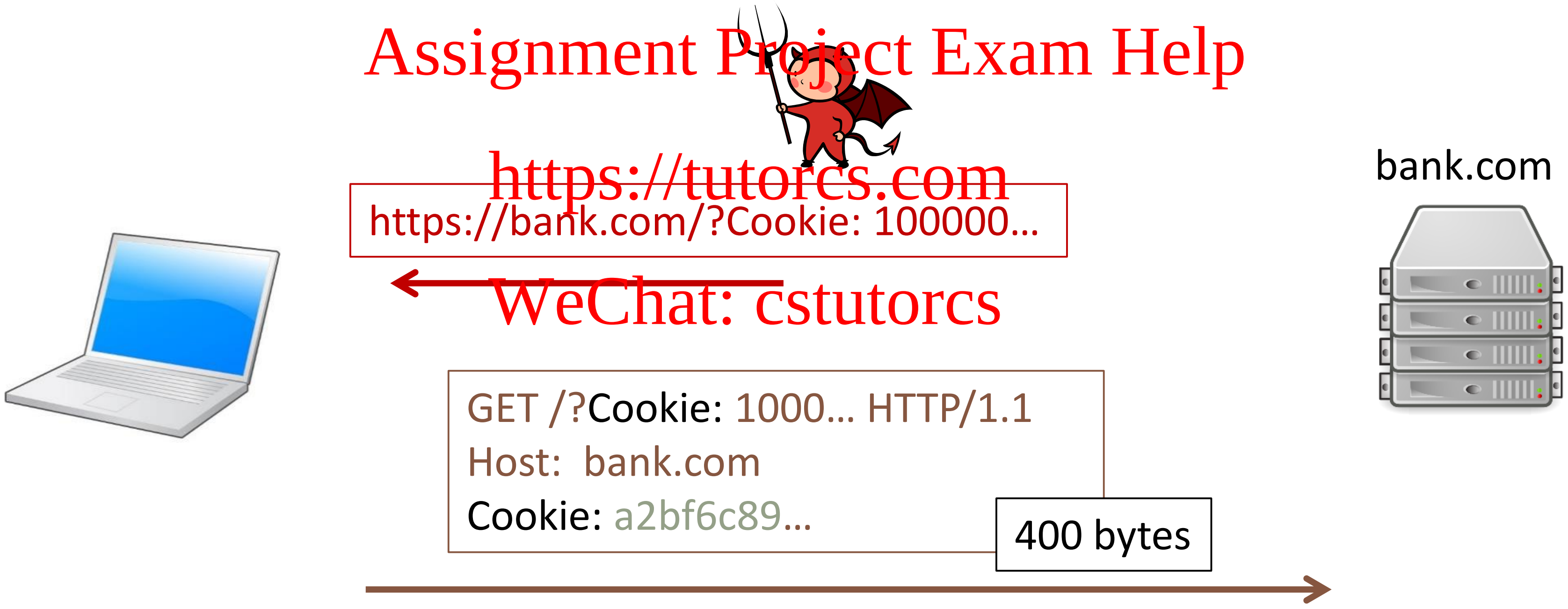


https://bank.com/?Cookie: 000000…

```
GET /?Cookie: 0000… HTTP/1.1
Host: bank…
Cookie: a2bf6c89…
```

CRIME attack

Discovered by Thai Duong and Juliano Rizzo, 2012



bank.com

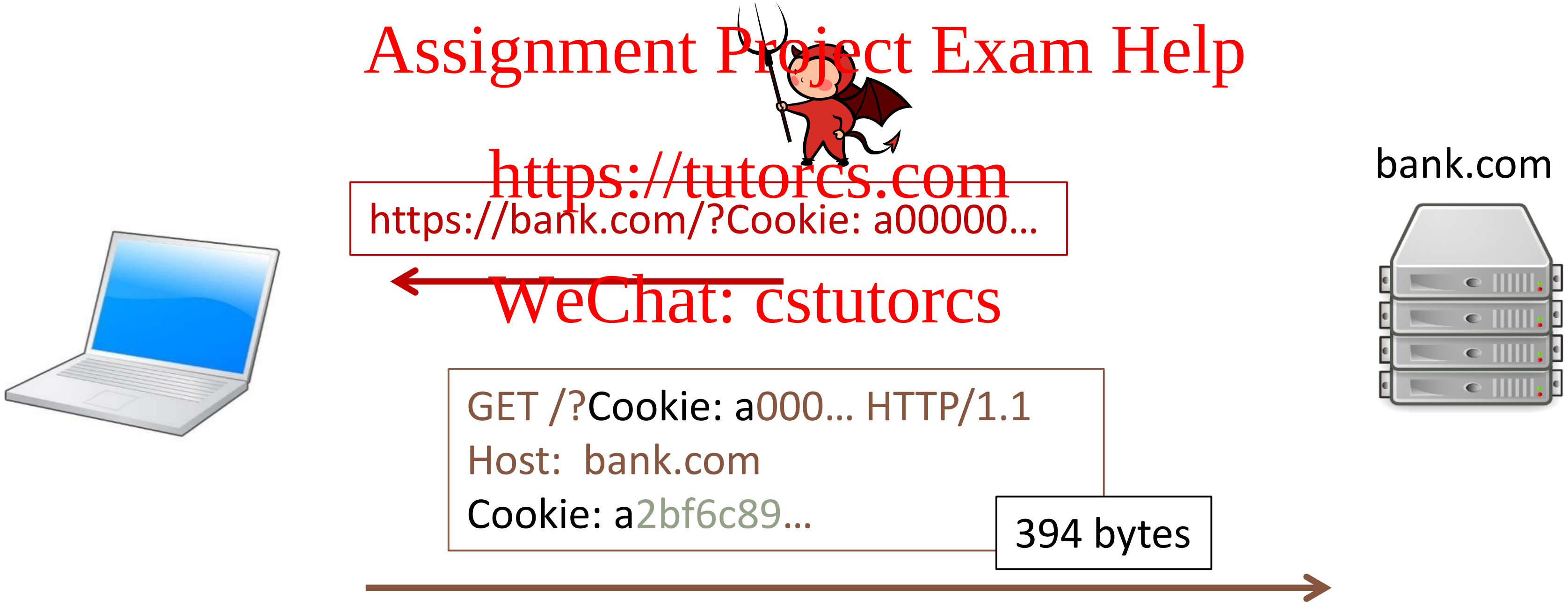
https://bank.com/?Cookie: 100000…

GET /?Cookie: 1000… HTTP/1.1
Host: bank.com
Cookie: a2bf6c89…

400 bytes

CRIME attack

Discovered by Thai Duong and Juliano Rizzo, 2012



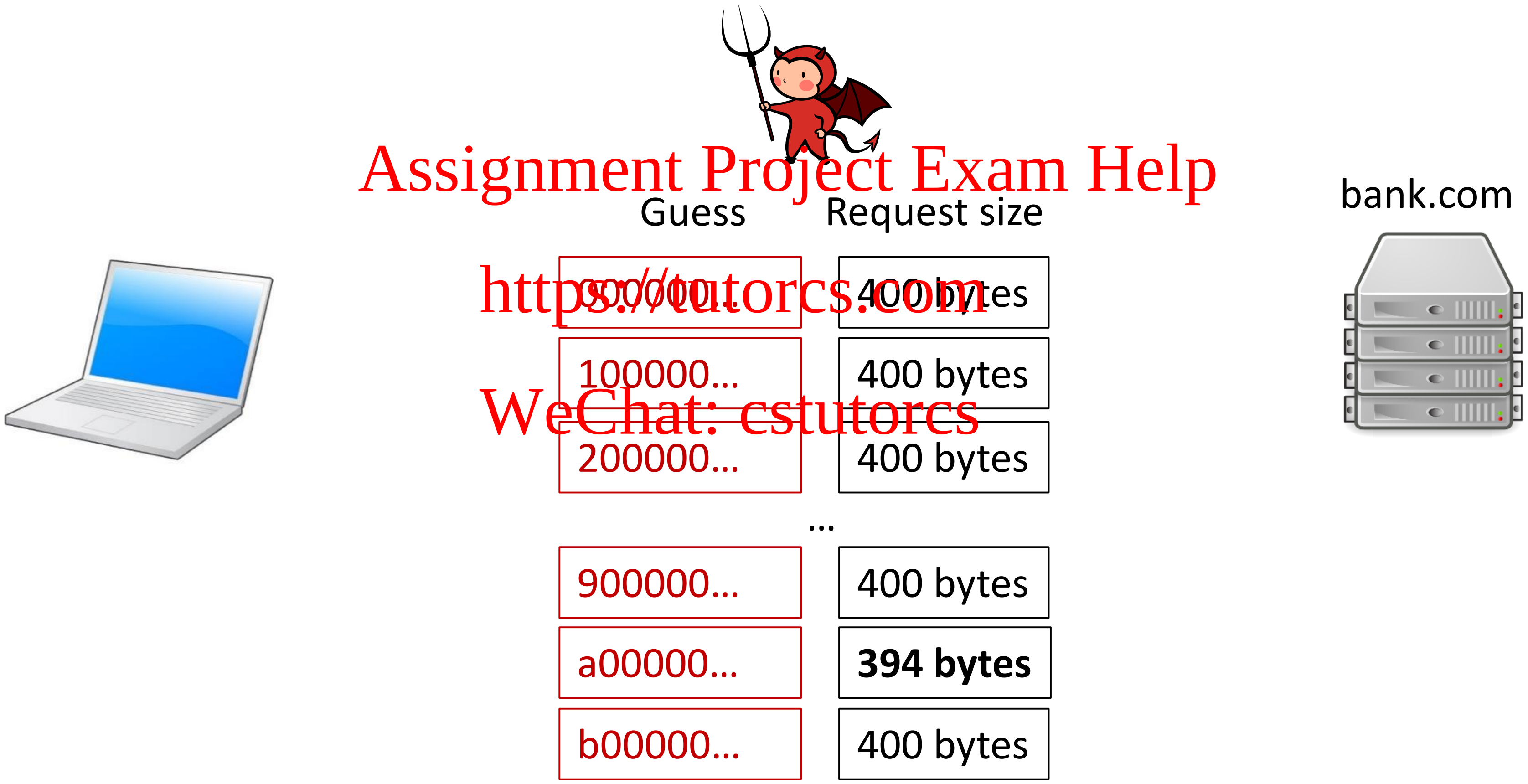
https://bank.com/?Cookie: a00000…

```
GET /?Cookie: a000… HTTP/1.1
Host:  bank.com
Cookie: a2bf6c89…
```

394 bytes

CRIME attack

Discovered by Thai Duong and Juliano Rizzo, 2012



Guess	Request size
000000…	400 bytes
100000…	400 bytes
200000…	400 bytes
…	
900000…	400 bytes
a00000…	**394 bytes**
b00000…	400 bytes

goto fail;

2014 Apple TLS library – `SSLVerifySignedServerKeyExchange()`

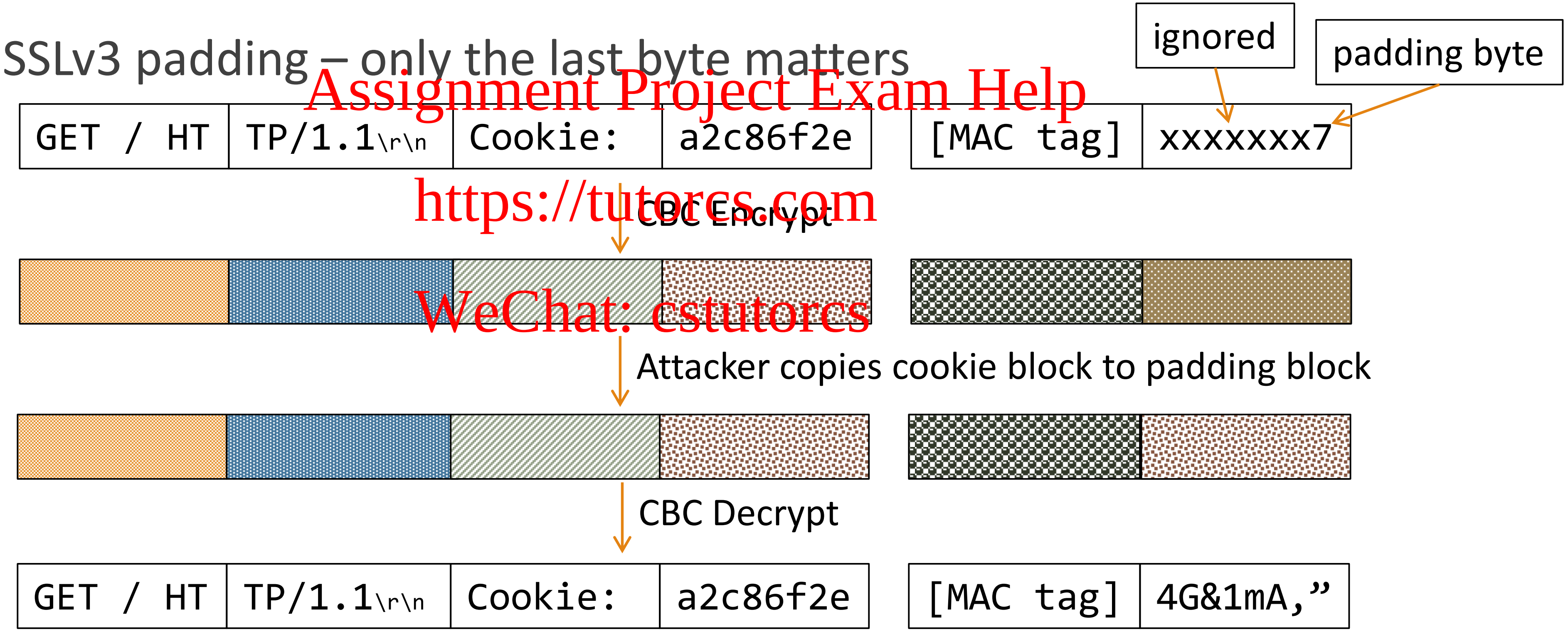
```
hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
hashOut.length = SSL_SHA1_DIGEST_LEN;

if ((err = SSLFreeBuffer(&hashCtx)) != 0)
    goto fail;

if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
    goto fail;

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(...);
fail:
    // Cleanup buffers, etc. Return err
    return err;
```

POODLE

Discvoered by Bodo Möller, Thai Duong and Krzysztof Kotowicz, 2014

Padding Oracle On Downgraded Legacy Encryption

SSLv3 padding – only the last byte matters



ignored

padding byte

GET / HT	TP/1.1\r\n	Cookie:	a2c86f2e	[MAC tag]	xxxxxxx7

CBC Encrypt

Attacker copies cookie block to padding block

CBC Decrypt

GET / HT	TP/1.1\r\n	Cookie:	a2c86f2e	[MAC tag]	4G&1mA,”

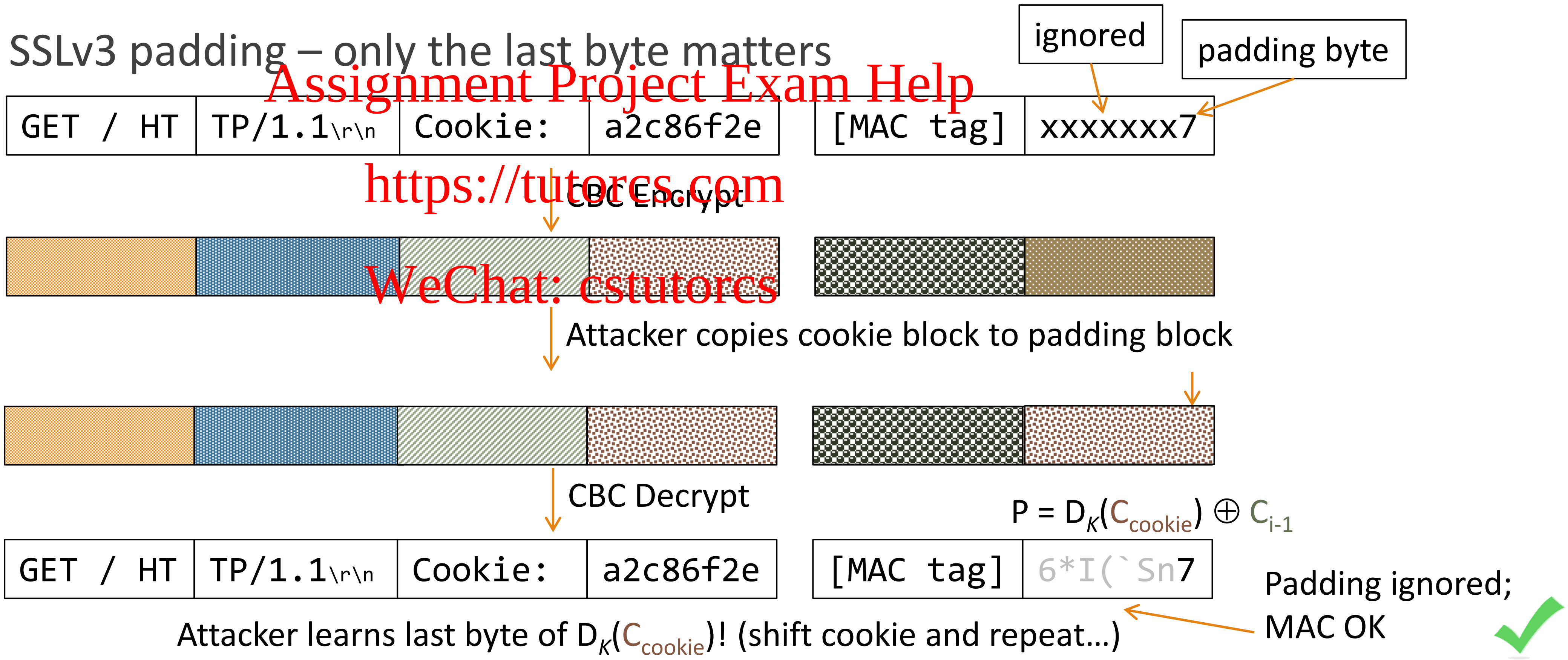
✗ BAD PADDING OR MAC

POODLE

Discvoered by Bodo Möller, Thai Duong and Krzysztof Kotowicz, 2014

Padding Oracle On Downgraded Legacy Encryption

SSLv3 padding – only the last byte matters



Attacker copies cookie block to padding block

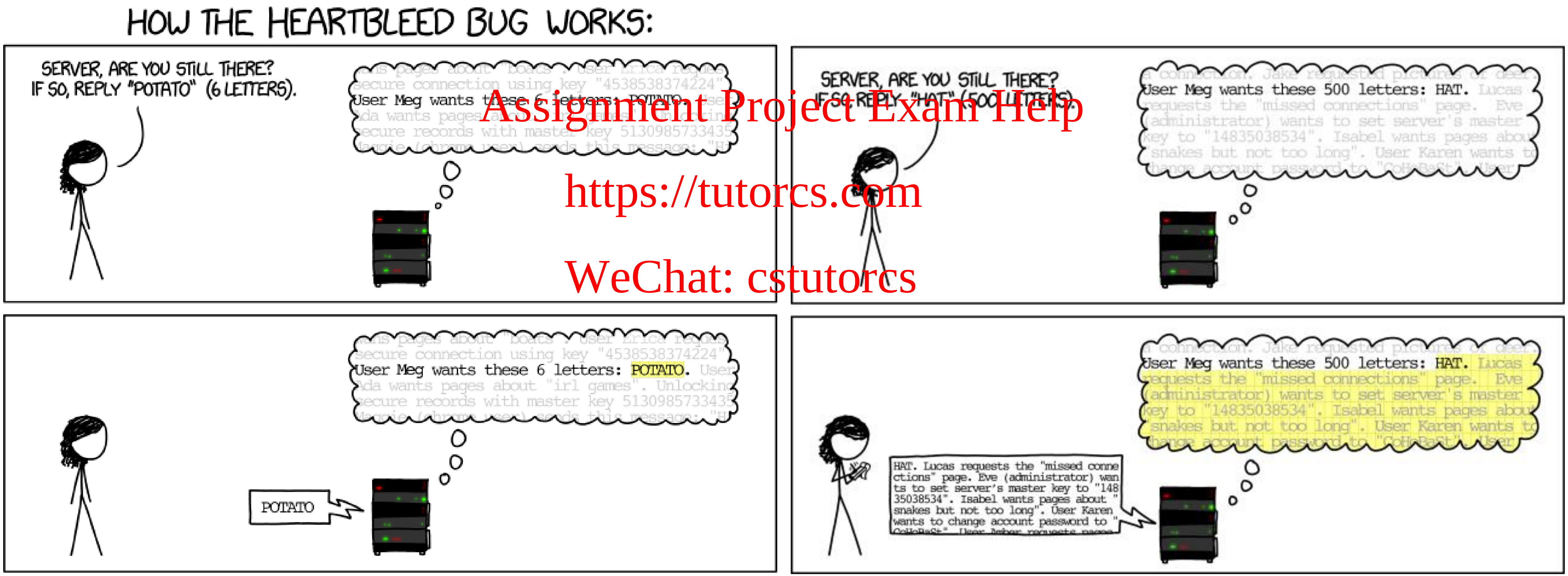
$P = D_K(C_{cookie}) \oplus C_{i-1}$

Padding ignored;
MAC OK

Attacker learns last byte of $D_K(C_{cookie})$! (shift cookie and repeat…)

Heartbleed

Heartbleed

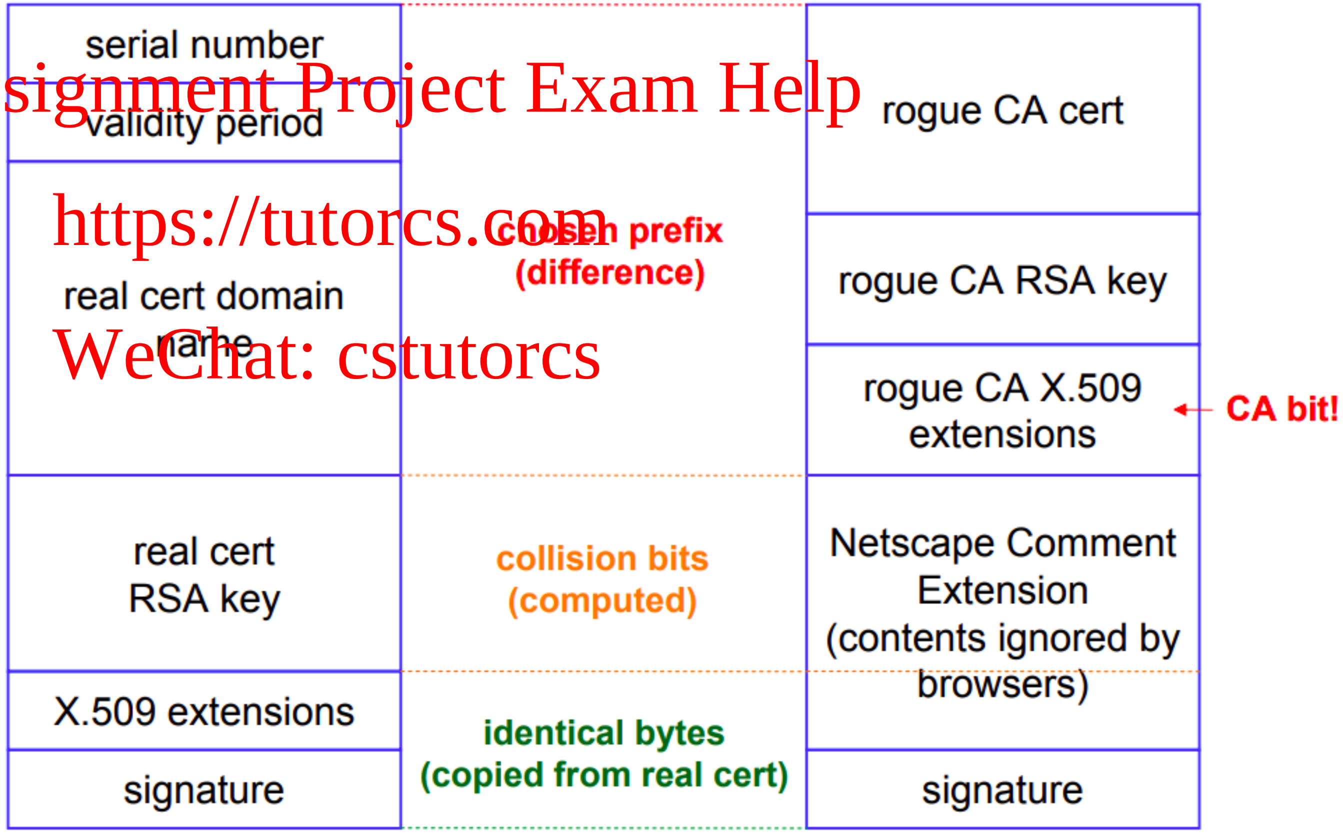

HOW THE HEARTBLEED BUG WORKS:
SERVER, ARE YOU STILL THERE? IF SO, REPLY "POTATO" (6 LETTERS).
ants pages about "boats". User Erica requests a secure connection using key "4538538374224". User Meg wants these 6 letters: POTATO. User Ada wants pages about "irl games". Unlocking secure records with master key 5130985733435. Maggie (chrome user) sends this message: "H
SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
a connection. Jake requested pictures of deer. User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User
POTATO
HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User Amber requests pages

https://xkcd.com/1354/

MD5 Considered Harmful Today

Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger

In 2008 (at CCC), a group of researchers showed that they could create a rogue CA certificate using an MD5 collision



serial number

validity period

real cert domain name

chosen prefix (difference)

rogue CA cert

rogue CA RSA key

rogue CA X.509 extensions

CA bit!

real cert RSA key

collision bits (computed)

Netscape Comment Extension (contents ignored by browsers)

X.509 extensions

signature

identical bytes (copied from real cert)

signature

https://win.tue.nl/hashclash/rogue-ca/downloads/md5-collisions-1.0.pdf

MD5 Considered Harmful Today

Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger

This kind of md5 collisions takes a bit more processing than `fastcoll` from the crypto project…

- So researchers used a cluster of 200 PS3s for ~2 days:
- Took 4 attempts (CA signatures)

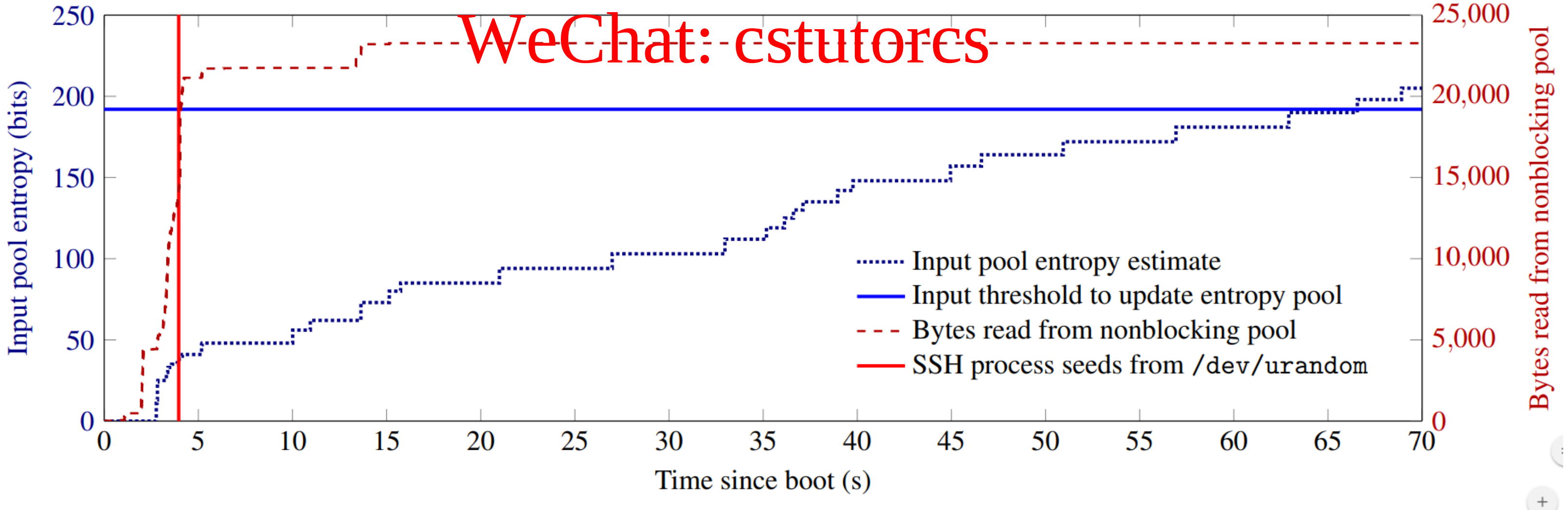
https://win.tue.nl/hashclash/rogue-ca/downloads/md5-collisions-1.0.pdf

“Mining Your Ps and Qs”

Nadia Heninger, Zakir Durumeric, Eric Wustrow , and J. Alex Halderman

In 2012, a team of researchers performed a global analysis of SSL/TLS and SSH keys

- 5.6% of TLS and 9.6% of SSH hosts shared cryptographic keys in a vulnerable manner
- Calculated the private keys for 0.5% of TLS hosts and 1.06% of SSH hosts
  - What if two RSA servers generate the same p but different q? $N_1 = pq_1$ and $N_2 = pq_2$ **[Find p given N_1 and N_2?]**
- Uncovered vulnerabilities in Linux’s Random Number Generator (**`/dev/urandom`**)


Input pool entropy (bits)
Bytes read from nonblocking pool
Time since boot (s)
Input pool entropy estimate
Input threshold to update entropy pool
Bytes read from nonblocking pool
SSH process seeds from /dev/urandom