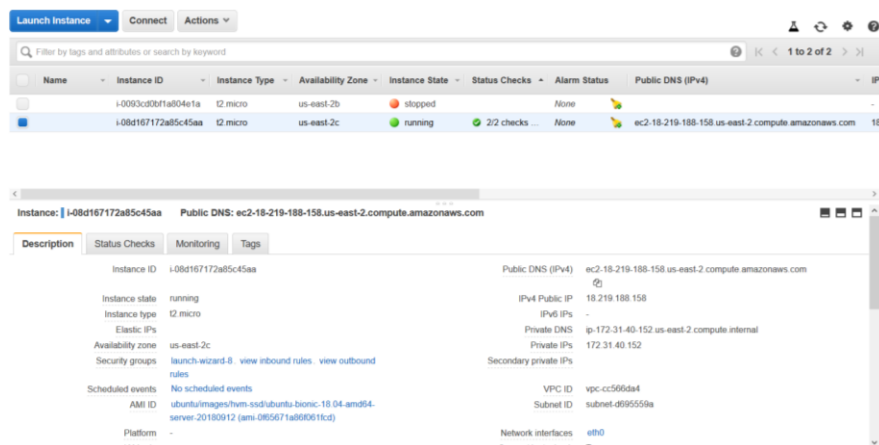


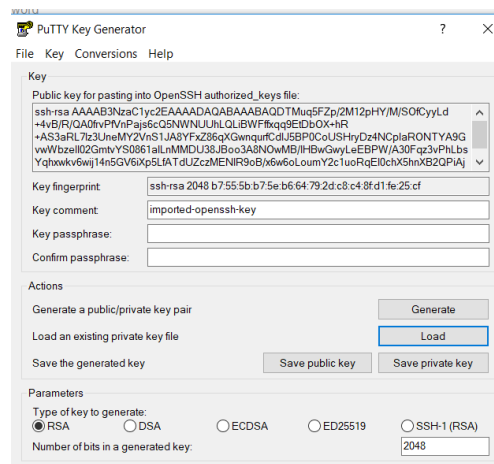
ITIS 6240
Parag Mhatre
Project 1
Oct 21st, 2018

I used a AWS instance of Ubuntu 16.04 LTS - 64-bit Linux server. Following are the steps that I undertook to perform the Project and successfully setup a Apache server with strong Cryptography.

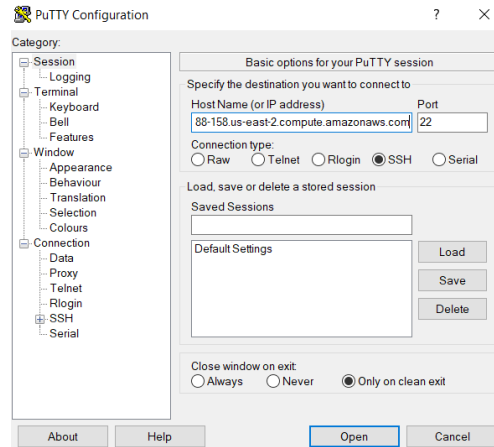
1. I spun up a Ubuntu 64-bit machine on AWS and opened up it's ports for HTTPS traffic.



2. I used PuTTYgen to convert the “keypair.pem” key that I got from AWS to Public and Private key files for the Server.



- Using the Public Address of the Instance and “.pvk” key file, I connected to it using PuTTY.



- After Connecting to the Remote Machine, the first thing I did was, to update its application libraries. To do this, I ran the command, on the instance through SSL command line:

sudo apt-get update

```
ubuntu@ip-172-31-23-146: ~  
Get cloud support with Ubuntu Advantage Cloud Guest:  
http://www.ubuntu.com/business/services/cloud  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-23-146:~$ sudo apt-get update  
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease  
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [8  
8.7 kB]  
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease  
[74.6 kB]
```

- After updating the library, I installed other packages as given in the instructions.

Sudo apt-get -y install make wget libssl-dev libncurses5-dev gcc

```
ubuntu@ip-172-31-23-146:~$ sudo apt-get -y install make wget libssl-dev libncurses5-dev gcc  
Get:1 http://security.ubuntu.com/ubuntu bionic-security/main Sources [53.8 kB]  
Get:2 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [18  
6 kB]  
Get:3 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [72  
1.4 kB]  
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages  
[89.0 kB]  
Get:5 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en  
[48.5 kB]  
Get:6 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packag  
es [1449 B]  
Get:7 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-  
en [1894 B]  
Fetched 26.1 MB in 5s (5323 kB/s)  
Reading package lists... Done  
ubuntu@ip-172-31-23-146:~$ sudo apt-get -y install make wget libssl-dev libncurses5-dev gcc  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
wget is already the newest version (1.19.4-1ubuntu2.1).  
The following additional packages will be installed:  
binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-7 gcc-7  
binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-7 gcc-7
```

6. Then I installed Apache2 on the server. To do this, I entered,

Sudo apt-get install apache2

```
ubuntu@ip-172-31-23-146:~$ sudo apt-get install apache2
Setting up libc6-dev:amd64 (2.27-3ubuntu1) ...
Setting up libltdl:amd64 (8.2.0-1ubuntu2-18.04) ...
Setting up libisl19:amd64 (0.19-1) ...
Setting up libasan4:amd64 (7.3.0-27ubuntu1-18.04) ...
Setting up libbinutils:amd64 (2.30-21ubuntu1-18.04) ...
Setting up libcilkrts5:amd64 (7.3.0-27ubuntu1-18.04) ...
Setting up libubsan0:amd64 (7.3.0-27ubuntu1-18.04) ...
Setting up libgcc-7-dev:amd64 (7.3.0-27ubuntu1-18.04) ...
Setting up cpp-7 (7.3.0-27ubuntu1-18.04) ...
Setting up libcurses5-dev:amd64 (6.1-1ubuntu18.04) ...
Setting up binutils-x86-64-linux-gnu (2.30-21ubuntu1-18.04) ...
Setting up cpp (4:7.3.0-3ubuntu2.1) ...
Setting up binutils (2.30-21ubuntu1-18.04) ...
Setting up gcc-7 (7.3.0-27ubuntu1-18.04) ...
Setting up gcc (4:7.3.0-3ubuntu2.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
ubuntu@ip-172-31-23-146:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
```

7. Now, to create and install the SSL certificate and key, I used the following commands:

Openssl genrsa -des3 -out server.key 2048

Openssl req -new -key server.key -out server.csr

Openssl rsa -in server.key.org -out server.key

Openssl x509 -req -days 365 -in server.csr -signkey rsakey.key -out rsacert.crt

```
ubuntu@ip-172-31-40-152:~$ sudo vim /etc/apache2/sites-available/default-ssl.conf
ubuntu@ip-172-31-40-152:~$ openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
ubuntu@ip-172-31-40-152:~$ openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CLT
Locality Name (eg, city) []:NC
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNCC
Organizational Unit Name (eg, section) []:CCI
Common Name (e.g. server FQDN or YOUR name) []:ec2-18-219-188-158.us-east-2.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:utut
An optional company name []:abc
ubuntu@ip-172-31-40-152:~$ cp server.key server.key.org
ubuntu@ip-172-31-40-152:~$ openssl rsa -in server.key.org -out server.key
Enter pass phrase for server.key.org:
writing RSA key
ubuntu@ip-172-31-40-152:~$ openssl x509 -req -days 365 -in server.csr -signkey r
sakey.key -out rsacert.crt
x509: Cannot open input file rsakey.key, No such file or directory
x509: Use -help for summary.
ubuntu@ip-172-31-40-152:~$ openssl x509 -req -days 365 -in server.csr -signkey s
erver.key -out rsacert.crt
Signature ok
subject=C = US, ST = CLT, L = NC, O = UNCC, OU = CCI, CN = ec2-18-219-188-158.us
```

- After that, I ran some commands to enable the settings to load in the Apache service and restarted it with the modifications.

Sudo a2enmod ssl

Sudo a2enmod headers

Systemctl restart apache2

```
ubuntu@ip-172-31-40-152:~$ sudo service apache2 restart
ubuntu@ip-172-31-40-152:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
ubuntu@ip-172-31-40-152:~$ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
systemctl restart apache2
```

- Now, I modified the default-ssl.conf file located in /etc/apache2/sites-available/ and added some lines to the text.

Sudo vim /etc/apache2/sites-available/default-ssl.conf

ServerName ec2-18-219-188-158.us-east-2.compute.amazonaws.com

SSLCertificateFile rsacert.crt

SSLCertificateKeyFile rsakey.key

```
ubuntu@ip-172-31-40-152:~$ sudo vim /etc/apache2/sites-available/default-ssl.conf
<VirtualHost default :443>
    ServerAdmin webmaster@localhost
    ServerName ec2-18-219-188-158.us-east-2.compute.amazonaws.com
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice,
    warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For exampl
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # A self-signed (snakeoil) certificate can be created by insta
    # the ssl-cert package. See
    # /usr/share/doc/apache2/README.Debian.gz for more info.
    # If both key and certificate are stored in the same file, only
    # the
    # SSLCertificateFile directive is needed.
    SSLCertificateFile rsacert.crt
    SSLCertificateKeyFile rsakey.key

    # Server Certificate Chain:
    # Point SSLCertificateChainFile at a file containing the
    # concatenation of PEM encoded CA certificates which form the
    1,1 Top
```

Downgrade-1.0 force-response-1.0

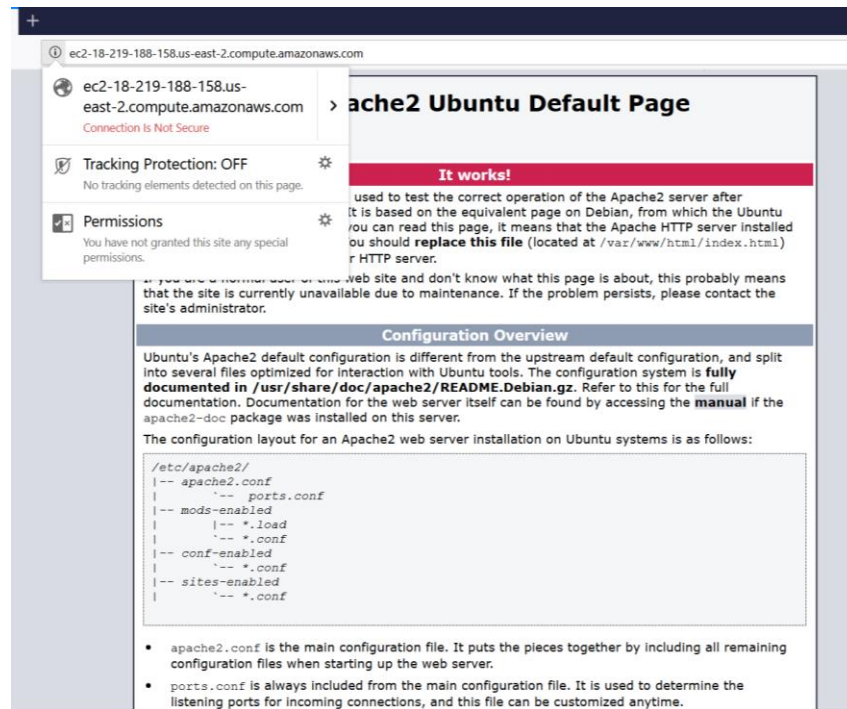
[illegible]

```
Sudo systemctl restart apache2
```

```
ubuntu@ip-172-31-23-146:/
```

```
/system/apache-htcacheclean.service.  
Processing triggers for libc-bin (2.27-3ubuntu1) ...  
Processing triggers for ureadahead (0.100.0-20) ...  
Processing triggers for systemd (237-3ubuntu0.3) ...  
Processing triggers for ufw (0.35-5) ...  
ubuntu@ip-172-31-23-146:~$ sudo vim /etc/apache2/sites-available/default-ssl  
ubuntu@ip-172-31-23-146:~$ sudo vim /etc/apache2/sites-available/ssl-default  
ubuntu@ip-172-31-23-146:~$ sudo vim /etc/apache2/sites-available/default-ssl  
ubuntu@ip-172-31-23-146:~$ cd /etc/apache2/sites-available/  
ubuntu@ip-172-31-23-146:/etc/apache2/sites-available$ ls  
000-default.conf  default-ssl.conf  
ubuntu@ip-172-31-23-146:/etc/apache2/sites-available$ cd ..  
ubuntu@ip-172-31-23-146:/etc/apache2$ cd ..  
ubuntu@ip-172-31-23-146:/etc$ cd ..  
ubuntu@ip-172-31-23-146:~$ sudo vim /etc/apache2/sites-available/default-ssl.conf  
ubuntu@ip-172-31-23-146:~$ sudo vim /etc/apache2/sites-available/default-ssl.conf  
ubuntu@ip-172-31-23-146:~$ sudo a2enmod ssl  
Considering dependency setenvif for ssl:  
Module setenvif already enabled  
Considering dependency mime for ssl:  
Module mime already enabled  
Considering dependency socache_shmcb for ssl:  
Enabling module socache_shmcb.  
Enabling module ssl.  
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.  
To activate the new configuration, you need to run:  
systemctl restart apache2  
ubuntu@ip-172-31-23-146:~$ sudo a2enmod headers  
Enabling module headers.  
To activate the new configuration, you need to run:  
systemctl restart apache2  
ubuntu@ip-172-31-23-146:~$ sudo a2ensite default-ssl  
Enabling site default-ssl.  
To activate the new configuration, you need to run:  
systemctl reload apache2  
ubuntu@ip-172-31-23-146:~$ sudo systemctl restart apache2
```

11. Now after that process, I checked if the server was accessible through the browser. I opened it up using Mozilla Firefox



12. Now to change the SSL Configuration, I accessed the “ssl.conf” file from /etc/apache2/mods-available/ssl.conf and I added the following lines and restarted the “apache2” service.

SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:!DH

SSLHonorCipherORDER on

SSLProtocol +TLSv1.2 -TLSv1 -TLSv1.1 -SSLv3


```
ubuntu@ip-172-31-40-152: ~$ \
# options.
# Enable only secure ciphers:
SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:DH
SSLHonorCipherOrder on

# SSL server cipher order preference:
# Use server priorities for cipher algorithm choice.
# Clients may prefer lower grade encryption. You should enable this
# option if you want to enforce stronger encryption, and can afford
# the CPU cost, and did not override SSLCipherSuite in a way that puts
# insecure ciphers first.
# Default: Off
SSLHonorCipherOrder on

# The protocols to enable.
# Available values: all, SSLv3, TLSv1, TLSv1.1, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol +SSLv1.2 -SSLv3 -TLSv1 -TLSv1.1

# Allow insecure renegotiation with clients which do not yet support it
# secure renegotiation protocol. Default: Off
SSLInsecureRenegotiation on
```

13. Now, I ran tests on DigiCert.com and the screenshots of the results are attached below:

 **SSL** **PKI** **IoT** **Solutions** **About** **Support**

DigiCert® SSL Installation Diagnostics Tool

SSL Certificate Checker

If you are having a problem with your SSL certificate installation, please enter the name of your server. Our installation diagnostics tool will help you locate the problem and verify your SSL Certificate installation.

Server Address: (Ex: www.digicert.com)

☐ Check for common vulnerabilities

CHECK SERVER

✔ **DNS resolves ec2-18-219-188-158.us-east-2.compute.amazonaws.com to 18.219.188.158**

HTTP Server Header: Apache/2.4.29 (Ubuntu)

✔ **SSL certificate**


Common Name = ec2-18-219-188-158.us-east-2.compute.amazonaws.com
Issuer = ec2-18-219-188-158.us-east-2.compute.amazonaws.com
Serial Number = AD494C81725F6A9D
SHA1 Thumbprint = 724B36085A3FD52F574501A9352F28E6BD77B24A
Key Length = 2048
Signature algorithm = SHA256 + RSA (excellent)
Secure Renegotiation: Supported

✔ **SSL Certificate has not been revoked**

LIVE CHAT

Get Help Now!
Click here for live help with your SSL installation.

CHAT NOW

 **SSL** **PKI** **IoT** **Solutions** **About** **Support**

Signature algorithm = SHA256 + RSA (excellent)
Secure Renegotiation: Supported


✔ **SSL Certificate has not been revoked**

OCSP Staple: Not Enabled
OCSP Origin: Not Enabled
CRL Status: Not Enabled

✔ **SSL Certificate expiration**

The certificate expires October 20, 2019 (364 days from today)

✔ **Certificate Name matches ec2-18-219-188-158.us-east-2.compute.amazonaws.com**

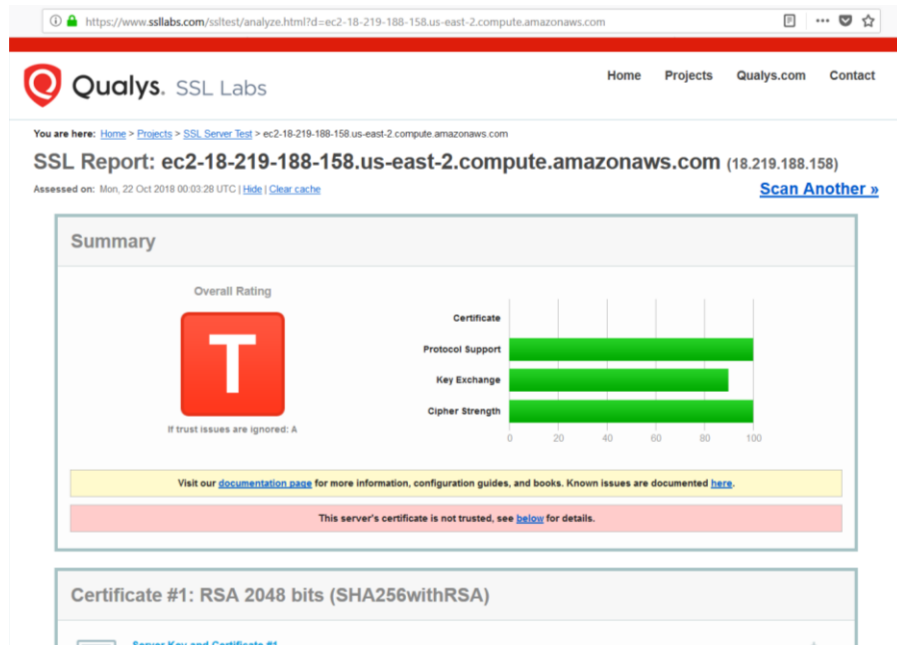


Subject ec2-18-219-188-158.us-east-2.compute.amazonaws.com
Valid from 20/Oct/2018 to 20/Oct/2019
Issuer ec2-18-219-188-158.us-east-2.compute.amazonaws.com

✗ **SSL Certificate is not trusted**

The certificate is not signed by a trusted authority (checking against Mozilla's root store). If you bought the certificate from a trusted authority, you probably just need to install one or more intermediate certificates. Contact your certificate provider for assistance doing this for your server platform.

14. Furthermore, I ran tests on SSLabs.com and the results are given below(including the list of ciphers on the server):



Server Key and Certificate #1	
Subject	ec2-18-219-188-158.us-east-2.compute.amazonaws.com
Fingerprint SHA256:	a5845164c6b9429c1c850513d2d55b0a7e89742e4b37fae0eb8f6ee5b401
Pin SHA256:	PT+UQ65TygazZTLhPbGueG0Bckzw+KDDIXAWQOPne=
Common names	ec2-18-219-188-158.us-east-2.compute.amazonaws.com
Alternative names	- INVALID
Serial Number	00ad494c81729f6a9d
Valid from	Sat, 20 Oct 2018 08:16:43 UTC
Valid until	Sun, 20 Oct 2019 08:16:43 UTC (expires in 11 months and 28 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	ec2-18-219-188-158.us-east-2.compute.amazonaws.com Self-signed
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	No
OCSP Must Staple	No
Revocation information	None
DNS CAA	No (more info)
Trusted	No NOT TRUSTED (Why?) Mozilla Apple Android Java Windows

Additional Certificates (if supplied)	
Certificates provided	1 (902 bytes)
Chain issues	None

Certification Paths	
---------------------	--

https://www.ssllabs.com/ssltest/analyze.html?d=ec2-18-219-188-158.us-east-2.compute.amazonaws.com

Click here to expand

Configuration

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

For TLS 1.3 tests, we only support RFC 8446.

Cipher Suites

TLS 1.2 (we could not determine if the server has a preference)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (bwe030)	ECDH x25519 (eq. 3072 bits RSA)	FS	256
--	---------------------------------	----	-----

Handshake Simulation

Android 4.4.2	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Android 5.0.0	Server sent fatal alert: handshake_failure			
Android 6.0	Server sent fatal alert: handshake_failure			
Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS

https://www.ssllabs.com/ssltest/analyze.html?d=ec2-18-219-188-158.us-east-2.compute.amazonaws.com

Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Brave/Preview Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Chrome 49 / XP SP3	Server sent fatal alert: handshake_failure			
Chrome 69 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Chrome 70 / Win 10	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Firefox 31.3.0 ESR / Win 7	Server sent fatal alert: handshake_failure			
Firefox 47 / Win 7 R	Server sent fatal alert: handshake_failure			
Firefox 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Googlebot Feb 2018	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
IE 11 / Win 7 R	Server sent fatal alert: handshake_failure			
IE 11 / Win 8.1 R	Server sent fatal alert: handshake_failure			
IE 11 / Win Phone 8.1 R	Server sent fatal alert: handshake_failure			
IE 11 / Win Phone 8.1 Update R	Server sent fatal alert: handshake_failure			
IE 11 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 13 / Win Phone 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Java Bv161	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.0.1f R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.0.2e R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 6 / iOS 6.0.1	Server sent fatal alert: handshake_failure			
Safari 7 / iOS 7.1 R	Server sent fatal alert: handshake_failure			
Safari 7 / OS X 10.9 R	Server sent fatal alert: handshake_failure			
Safari 8 / iOS 8.4 R	Server sent fatal alert: handshake_failure			
Safari 8 / OS X 10.10 R	Server sent fatal alert: handshake_failure			
Safari 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS

https://www.ssllabs.com/sslltest/analyze.html?id=ec2-18-219-188-158.us-east-2.compute.amazonaws.com

ECDH public server param reuse	No
Supported Named Groups	x25519, secp256r1, secp521r1, secp384r1 (server preferred order)
SSL 2 handshake compatibility	No

HTTP Requests

1 https://ec2-18-219-188-158.us-east-2.compute.amazonaws.com/ (HTTP/1.1 200 OK)

Miscellaneous

Test date	Mon, 22 Oct 2018 00:02:58 UTC
Test duration	30.514 seconds
HTTP status code	200
HTTP server signature	Apache/2.4.29 (Ubuntu)
Server hostname	ec2-18-219-188-158.us-east-2.compute.amazonaws.com