Assignment 3

Due Friday, Dec 1st at 11:59pm

Assignments are to be completed individually. We generally expect you to make an honest effort searching around online before contacting course staff with technical questions (`stackexchange.com` and `crypto.stackexchange.com` are great resources btw).

Submission Instructions

Place each answer in a clearly named file (*e.g.,* `Q1.txt`). Answers may be in `txt`, `pdf` or `doc/docx`. To avoid problems and potential mark penalties, do not use other formats. Place these answers along with all accompanying files in a `.zip` and submit via OWL by the due date. As per the course late policy, assignments can be uploaded only up to 48 hours past the due date.

1. [30 marks] Part 1. Setting up a TLS Enabled Webserver

In this question you will setup a TLS enabled webserver with a goal of serving a basic webpage over a TLS connection. To do this we're going to have to fake a couple of things since we don't want to make you pay for an actual domain name or web hosting.
Use the following setup guide to get your basic TLS enabled server up and running:
`https://whisperlab.org/information-security/assignments/server-setup.html`

Deliverables. Places all files in a directory `Q1`.

(a) A screen capture of Chrome showing your website's URL, the default webpage, and the green padlock icon. Name this file Q1.png (or Q1.jpg),

(b) Download the testssl.sh script from https://testssl.sh/. Run it on your website and pipe the output as follows:

```
./testssl.sh [your website domain] > Q1.txt
```

2. [30 marks] Part 2. Tune the TLS Configuration

Not all TLS configurations are created equal. In class we looked at good TLS configs (*e.g.,* `https://whisperlab.org` and not-so-good configs (*e.g.,* `https://www.uwo.ca`). The Apache default is more in the not-so-good direction, so in this question you will improve your website's TLS configuration. In Question 1 you may have noticed that the result of `testssl.sh` found several aspects of the configuration it described as "NOT ok."

Objective: Configure your TLS settings so `testssl.sh` no longer finds anything that it considers "NOT ok." In addition configure the settings to only offer ciphersuites that meet the following criteria (some of them may already be satisfied by the default config):

- Key exchange shall be only 256-bit ECDHE
- Signature method shall be 4096-bit RSA
- Block cipher shall be 256-bit AES-GCM or AES-CBC
- Hash shall be in the SHA family
- Does not offer any SSL versions

Deliverables. Places all files in a directory `Q2`.

(a) Re-run the testssl.sh script on your website meeting the above objectives and put the result into a file Q2.txt: `./testssl.sh [your website domain] > Q2.txt`

(b) A copy of your modified `ssl.conf` file.

3. [20 marks] Negotiate This

There are two main ways a client and server can agree on a TLS ciphersuite. One is client preference: the client has an ordered lists of ciphersuite preferences, and the server picks the ciphersuite that is *most* preferred by the client, and that is supported by the server. The other method is server preference: the server has a list of ciphersuite preferences and the picks the ciphersuite that is *most* preferred by the server, and that is supported by the client.

Deliverables. Place a file Q3 in a directory `Q3`. Thinking about ciphersuite preferences of the following clients and servers, use SSL Labs Server Test and SSL Labs Client Test to help you answer the following questions. For each of the following which ciphersuite would be selected if:

(a) Chrome 57 connected to `whisperlab.org` and the server picks

(b) Chrome 57 connected to `whisperlab.org` and the client picks

(c) Android 7.0 connected to `uwo.ca` and the client picks

(d) IE 7 / Vista connected to `uwo.ca` and the server picks

4. [20 marks] TLS Key Kraziness

As we've talked about in class a single TLS connection involves *a lot* of keys. In this question you will enumerate *all* of the keys used in a single TLS connection to *google.ca*. This includes all public, private, and secret keys exchanged or generated during a TLS handshake.

Deliverables. Place answers in a directory `Q4`. First list the ciphersuite your browser connected under. The file then should have 3 columns: a description of the key (*e.g.,* Google's public ECDHE key), the name of the entities who know the key (*e.g.,* everyone), and a sentence or two describing the purpose of the key (*e.g.,* used for public key agreement).