COM 402 exercises 2019, session 3: Crypto and Trust in the Internet

Exercise 3.1

- Why can't you use a message authentication code (e.g. HMAC-SHA2) to sign a contract between a buyer and a seller?

Exercise 3.2

- If asymmetric crypto is really more useful than symmetric, why are we still using AES ?

Exercise 3.3

- Explain why using the same initialization vector (IV) multiple times with a stream cipher is more dangerous than with a block cipher.

Exercise 3.4

- Explain why AEAD in not malleable.

Exercise 3.5

- How can you find out all cipher suites supported by a TLS server?

Exercise 3.6

- Why is perfect forward secrecy important?

Exercise 3.7

To be sure that your customers connect to your website with https instead of http, you configure your web server to answer requests on the http port with a redirection to the https port.

- Why does this not guarantee that all customers will end up using https?
- Why does closing the http ports still not guarantee that customers will use https?
- What would be a working solution?

Exercise 3.8

Most mobile e-banking applications use certificate pinning to validate the certificates of the servers they connect to.

- Describe an attack that can be prevented by using a pinned certificate.

Exercise 3.9

- Which are the two certificate authorities that were used this summer and fall by the `com402.epfl.ch` web server.

Solutions to the Exercises

Solution 3.1

The MAC is based on a symmetric key that both parties need to know. Any party could modify the contract, replace the MAC and pretend it is authentic.

Solution 3.2

Symmetric crypto algorithms are usually much faster than asymmetric ones.

(Symmetric algorithms usually do simple bit manipulations and permutations, whereas asymmetric crypto typically implies mathematical operations on large numbers)

Solution 3.3

Stream cipher: If M_1 and M_2 are encrypted with the same key and IV then the xor of the ciphertexts is equal to the xor of the plaintext. Thus if you know parts of one plaintext you can decipher the same part of other ciphertext. $M_2 = M_1 \oplus C_1 \oplus C_2$. This not the case for block ciphers.

(If two blocks of a block cipher are encrypted with the same key and IV, then you can only detect if the two cleartexts are identical or not.)

Solution 3.4

The block cipher mode

Auhtenticated Encrpytion with Additional Data includes the calculation of an authentication code (called Tag). The tag is transmitted together with the ciphertext and used by the receiving party to verify that the ciphertext (and the tag) was not modified.

Solution 3.5

In TLS, the client sends a list of supported ciphersuites and the server chooses which one to use. To detect all supported suites, you need to connect multiple times and propose a single ciphersuite each time.

Solution 3.6

Without PFS, an attacker who records all encrpyted communications and key exchanges would be able to decrypt them in future if they are able compromise one communicating party (e.g. get the private key of a TLS server).

(With PFS encryption keys are based on freshly generated random information. Once the keys are discarded at the end of a communication, there is no way to decrypt any recorded communications anymore.)

Solution 3.7

- An attacker in a man-in-the-middle (MITM) position could modify the response of your http server to hide the redirection. The customer would continue to talk http tho the attacker who can forward the traffic with https to the real server. They can see and modify all the data.
- Even if the real server has no open http port, the MITM can fake the responses of an HTTP server and run the same attack as above.
- If your website is registered in the http Strict Transport Security (HSTS) pre-load list, then the browsers know that they should only use https to connect to your site.

A simple solution would be to change the default behavior of web browsers to use https by default when user do not write `http://` or `https://` in front of then name of websites they want to visit.

Solution 3.8

Without certificate pinning, the client has to trust a list of know trusted root certification authorities (CAs) when connecting to a server. If one of these CA was hacked or had an interest in spying on the connections, the application might not detect when it connects to a MITM instead of the real server.

Solution 3.9

Certificate transparency logs store and publish all certificates that were issued by participating CAs. If you got to `crt.sh` and search for `com402.epfl.ch`, you'll find the in August 2019 the site used a certificate issued by Let's Encrypt and since September it is using a certificate issued by Quo Vadis.