

### Infosec homework 4

1. Mengia Caflisch and Spoerndli analyzed the algorithm that was built in teletype model and HC-570. This experiment used the attack on the ciphertext. they found a vulnerability in the algorithm. By comparing 100 ciphertexts, Mengia and Spoerndli found a pattern to decipher the text.

### Key lesson

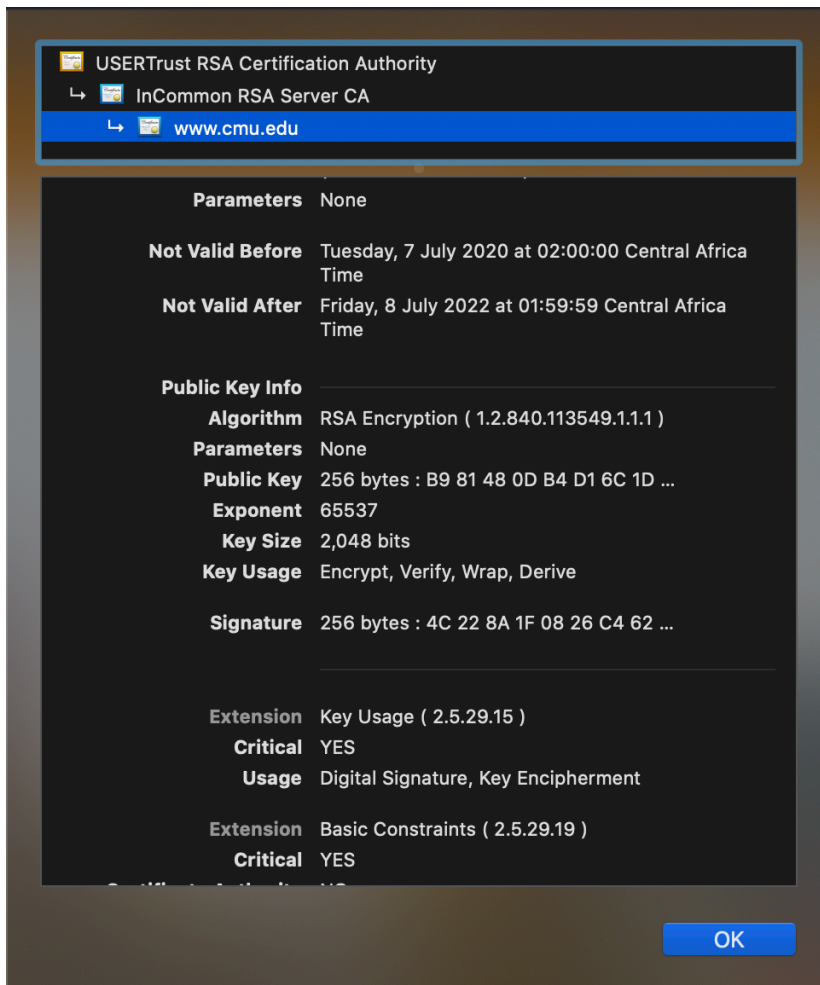
Cryptography is important and foundation of security. During wars cryptography was being used by governments as a way of ensuring the confidentiality of their data. So, cryptography is needed in industries. For example, it is used in banking industry et al.

2. Everything you need to know about Facebook, Google's app scandal
  - a. **Violation**
    - **Confidentiality:** Google and Facebook have used certificates issued by Apple for the only use of internal applications for Facebook and Google. However, those companies used the certificate to get access to the data of iPhone users.
    - **Authentication:** Facebook has authenticated their application to collect data from users by asking them to install the certificate.
    - **Signatures:** Google and Facebook have both misused the certificates signed by Apple to gain access to Apple's devices without permission.
  - b. **Facebook able to bypass HTTPS encryption:** Because Facebook asks users to trust their certificate by installing it on their devices. Once a user trust, the trusted connection is created and redirects all the data through Facebook VPN.

### 3. CMU homepage has RSA

At the beginning of the URL bar on the browser there is a small padlock => click on the padlock the pop up shows up => click on certificate (Valid) => details

The following picture shows all the component of the certificate in the browser.



Picture 1: picture shows the certificate component on the browser.

### To obtain the certificate using openssl s\_client

The command used is **ex** for editing the output from the OpenSSL command and write the output in a file called **cmu.crt**. The following command is used to obtain the certificate and save in the file

**ex +'/BEGIN CERTIFICATE/;/END CERTIFICATE/p' <(echo | openssl s\_client -connect cmu.edu:443) -sq > cmu.crt**

```
(base) Muhizis-MacBook-Pro:test muhizi$ ex +'/BEGIN CERTIFICATE/;/END CERTIFICATE/p' <(echo | openssl s_client \
-showcerts -connect cmu.edu:443) -sq > cmu.crt
depth=2 C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority
verify return:1
depth=1 C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
verify return:1
depth=0 C = US, postalCode = 15213, ST = Pennsylvania, L = Pittsburgh, street = 5000 Forbes Avenue,
O = Carnegie Mellon University, OU = Carnegie Mellon University, CN = cmu.edu
verify return:1
DONE
(base) Muhizis-MacBook-Pro:test muhizi$
```

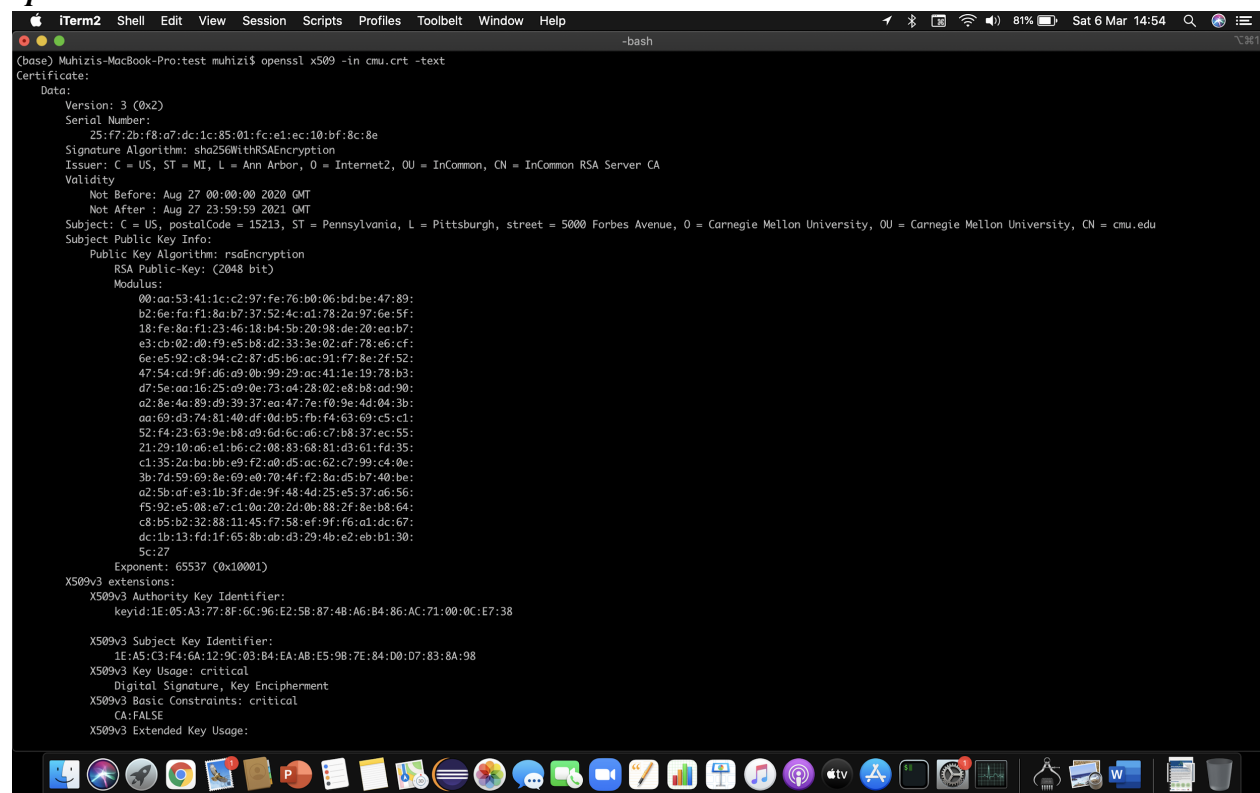
Picture 1: the output of the command that create cmu.crt

The following command is used only to view the certificate from the <https://cmu.edu> website.

***openssl s\_client -connect cmu.edu:443***

The following command is used to view the certificate from the saved in the file **cmu.crt**

***openssl x509 -in cmu.crt -text***



```
(base) Muhizis-MacBook-Pro:test muhizi$ openssl x509 -in cmu.crt -text
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      25:f7:2b:f8:a7:dc:1c:85:01:fc:e1:ec:10:bf:8c:8e
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
    Validity
      Not Before: Aug 27 00:00:00 2020 GMT
      Not After : Aug 27 23:59:59 2021 GMT
    Subject: C = US, postalCode = 15213, ST = Pennsylvania, L = Pittsburgh, street = 5800 Forbes Avenue, O = Carnegie Mellon University, OU = Carnegie Mellon University, CN = cmu.edu
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public-Key: (2048 bit)
      Modulus:
        00:aa:53:41:1c:c2:97:fe:76:b0:06:bd:be:47:89:
        b2:6e:fa:f1:8a:b7:37:52:4c:a1:78:2a:97:6e:5f:
        18:fa:8a:f1:23:46:18:b4:5b:20:9d:de:20:ea:b7:
        e3:cb:02:d0:f0:e5:b8:d2:33:3e:02:af:78:a6:cf:
        6e:e5:92:c8:94:c2:87:d5:b6:ac:91:f7:8e:2f:52:
        47:54:cd:9f:d6:a9:0b:99:29:ac:41:1e:19:78:b3:
        d7:5e:aa:16:25:a9:0e:73:a4:28:02:e8:b8:ad:90:
        a2:8e:4a:89:d9:39:37:ea:47:7e:f0:9e:4d:04:3b:
        aa:69:d3:74:81:40:df:0d:b5:fb:f4:63:69:c5:c1:
        52:f4:23:63:9e:b8:a9:6d:6c:a6:c7:b8:37:ec:55:
        21:29:10:a6:e1:b6:c2:08:83:68:81:d3:61:fd:35:
        c1:35:2a:ba:bb:e9:f2:a0:d5:ac:62:c7:99:c4:0e:
        3b:7d:59:69:8e:69:e0:70:4f:f2:8a:d5:b7:40:be:
        a2:5b:af:e3:1b:3f:de:9f:48:4d:25:e5:37:a6:56:
        f5:92:e5:08:e7:c1:0a:20:2d:0b:88:2f:8e:b8:64:
        c8:b5:b2:32:88:11:45:f7:58:ef:9f:f6:a1:dc:67:
        dca1b13:fd:1f:65:8b:ab:d3:29:4b:e2:eb:b1:30:
        5c:27
      Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Authority Key Identifier:
      keyid:1E:05:A3:77:8F:6C:96:E2:58:87:4B:A6:B4:86:AC:71:00:0C:E7:38

    X509v3 Subject Key Identifier:
      1E:A5:C3:F4:6A:12:9C:03:B4:EA:AB:E5:9B:7E:84:D0:D7:83:8A:98
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment
    X509v3 Basic Constraints: critical
      CA:FALSE
    X509v3 Extended Key Usage:
```

Picture 2: the output of the command to view the certificate.

4. **To produce the smallest encrypted file:** I would first compress the text and after encrypting the output from the compressed file. Because compression depends on some pattern in the text. One example of those patterns is that compression uses the repetitive character in the text. Therefore, those patterns might be destroyed along the encryption due to the noise the encryption created if the text is encrypted first.

5. The strategy of hashing the secret key is secure due to the advantages of the hash algorithm. Hash functions are one way which means that it is irreversible. Therefore, the attacker would not be able to reverse the key. The only technique that could be used is “Brute force attack”. So, in the worst case, it will take an attacker to try more than  $4.714723635E+284$ , which is extremely discouraging.

In addition, if an attacker manages to use brute force attack and luckily get the hashed key. The next key will be completely different due to the increment of the secret counter because hash function create a completely different value.

6. **RSA with small numbers:**

$$p = 17$$

$$q = 11$$

$$n = pq \Leftrightarrow 17 \times 11 = 187$$

$$\Phi = (P-1) \times (Q-1) = 16 \times 10 = 160$$

e has to be a prime number between 1 and 160 ( $1 < e < 160$ ).

e should never be a factor of  $\Phi$  (160).

Prime number 2, 3, 5, 7, 11, 13, 17, 19, 23

e = 3, because is the smallest and not factor of 160.

Finding of d using Euler method.

$$(exd) \bmod 160 = 1 \Leftrightarrow 3xd \bmod 160 = 1.$$

**Computing d using the Euler method.**

160	160
-3x53	-1x53
=1	=107

Therefore,  $d = 107$ .

**Encrypting message**

Message  $m = 5$

$$\text{Ciphertext} = m^{\text{pow}(e)} \bmod n \Leftrightarrow 5^{\text{pow}(3)} \bmod 187 = 125$$

**Decrypting message**

$$m = 125^{\text{power}(107)} \bmod 187 = 5$$