Advanced Security 1 – DT211-4, DT228-4 and DT282-4

Lab Sheet 4 - (6 marks)

Part A

Using any tool create your private and public key. Make sure that you keep your private key secure. Hint: these are some of the tools you may wish to consider PuTTyKey generator or GPG4Win. However, there are a lot other tools online.

Part B

In this lab you will be required to integrate the Enigmail (http://enigmail.mozdev.org/home/index.php.html) which is a security extension to Mozilla Thunderbird (https://www.mozilla.org/projects/thunderbird/). The Enigmail enables you to write and receive email messages signed and/or encrypted with the OpenPGP standard.

In order to achieve this objective you will have to install an email client such as Mozilla Thunderbird and Enigmail. Send an encrypted email or attachment to aneelrahim263@gmail.com. You will have to create your private and public key pair, send your public key to aneelrahim263@gmail.com. Remember that I must have your public key, otherwise, I will not be able to decrypt your email or attachment.

Hint: https://securityinabox.org/en/guide/thunderbird/windows/

Part C

There are several cryptographic uses for a clock. Key management functions are often linked to deadlines. The current time can provide both a unique value and a complete ordering of events. Install either network time protocol (NTP) or precision time protocol (PTP) in your computer and check if it is working.

List and give examples of attacks that can be mounted against a system with a lock. You may wish to read the following book chapter and a paper. Write at most four paragraphs and upload the report in Webcourses.

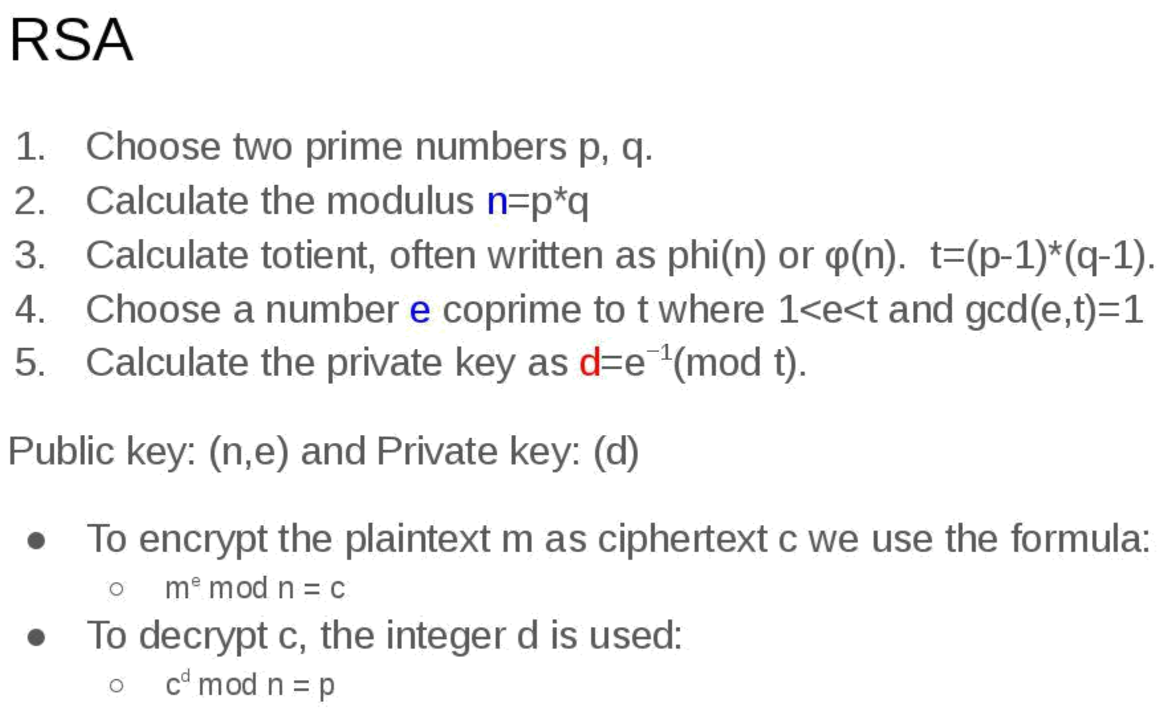
1. Niels Ferguson, Bruce Schneier and Tadayoshi Kohno, *Cryptography Engineering: Design Principles and Practical Applications*: chapter 16

2. Rick Ratzel and Rodney Greenstreet, *Toward Higher Precision: An introduction to PTP and its significance to NTP practitioners*, Communications of the ACM, Vol. 55, No. 10 October 2012.

Part D

RSA Encryption

Q2. You must write a program to implement RSA encryption and decryption. Small numbers can be used.



RSA

1. Choose two prime numbers p, q.
2. Calculate the modulus n=p*q
3. Calculate totient, often written as phi(n) or φ(n). t=(p-1)*(q-1).
4. Choose a number e coprime to t where 1<e<t and gcd(e,t)=1
5. Calculate the private key as $d=e^{-1}(\text{mod } t)$.

Public key: (n,e) and Private key: (d)

- To encrypt the plaintext m as ciphertext c we use the formula:
    - ○ $m^e \text{ mod } n = c$
- To decrypt c, the integer d is used:
    - ○ $c^d \text{ mod } n = p$

Deliverables: Please submit the code through the web course.