Marco CAVANI

ECU  10570027

Cryptographic CONCEPT’S part 2

Contents

[Intro 3](#_Toc103166463)

[Define RSA modulus (n) 3](#_Toc103166464)

[Define (phi) 3](#_Toc103166465)

[Find RSA public exponent (e) 3](#_Toc103166466)

[Make the public key (n, e) 3](#_Toc103166467)

[Make the private key (d) 3](#_Toc103166468)

[RSA encryption formula 4](#_Toc103166469)

[RSA decryption formula 4](#_Toc103166470)

[Hash function 4](#_Toc103166471)

[Hash Function Design 4](#_Toc103166472)

[Digital signature 5](#_Toc103166473)

[Digital Signature formula 5](#_Toc103166474)

[Cryptosystem 5](#_Toc103166475)

[Lesson learned 6](#_Toc103166476)

## Intro

In this second part I will demonstrate how we can take an advantage of an asymmetric cryptosystem and more precisely how RSA can be used to provide message nonrepudiation, integrity and authentication.

In portfolio 1 we used a block cipher to encrypt the message sent by Bob to Alice . However, the symmetric cryptosystem could not guarantee the integrity of the plaintext such as oscar could have easily manipulate or decrypt the message by intercepting the key during the exchange. Additionally, non-repudiation and authentication were not provided. Authentication as Alice could not verify that Bob actually sent her the message. Nonrepudiation such as Bob could deny having sent her the message.

Using RSA will enable us to cut out the key exchange which could be risky and address the nonrepudiation and authentication factors.

In this example we are going to generate the public and the private key for Alice to demonstrate encryption and decryption of the key used in portfolio 1 with RSA.

The first steps to generate Alice’s public and private keys:

### Define RSA modulus (n)

* Choose two prime number (p) and (q)
* Calculate n = p \* q for solid encryption is best to choose large primes

### Define (phi)

* Phi is the result of (p - 1) \* (q - 1)

### Find RSA public exponent (e)

* (e) can be any number coprime of (phi) between 1 and (phi)

### Make the public key (n, e)

* The public key is form by the pair of the public exponent (e) and the public modulus (n)

### Make the private key (d)

* The private key (d) is formed with (e) and (phi)
* (d) Is the number that multiplied by (e) is the equivalent of 1 modular (phi)
* (d) = (de) ≡ 1 mod (phi)
* to define (d) we multiply the public exponent (e) to each number on the left column all modular (phi) until we find 1 on the right column. once we find 1, we pick the number on the left column as (d) = (de) ≡ 1 mod (phi) as per look up table in the excel implementation

now we can use Alice’s RSA public exponent to encrypt the key.

### RSA encryption formula

* y = xe mod n
* the RSA strength is based on the difficulty of calculating the root of modular arithmetic.
* Even if (n) and (e) are publicly available is difficult for Oscar to decrypt the message as y = x5 ≡ mod 221 is not computably easy.
* Is considered one-way function

### RSA decryption formula

* x = yd mod n

in the second part we are going to hash the plaintext than we can use the output to create the digital signature.

## Hash function

* the hash method is a function that accepts an input of any chosen size and returns a fixed-size value.

### Hash Function Design

* conversion of the plaintext to Ascii numbers
* all number within the Ascii values are added together
* the result is modularized to a fixed size number

however, the function can accept an input of any length and the output should be short. Additionally, it should be one way and computably easy function. One way function provides security such as its impractical to reverse. Moreover, the function should be sensible to all bits in a way that if an input is swapped with another the output should be different. Therefore, the hash function design can be improved as if Oscar attempting to interchange values within the message the hash output will remain the same.

As you can see in the example provided, if we swop a number to another the output will be the same.

This is not ideally what we want from it, as the integrity of the message could not be verified. That is why I have implemented the multiplication raw. The multiplication raw multiplies each number to the next within the Ascii raw. then, add together the multiplication values. Therefore, If Oscar attempt to interchange the values within the Ascii raw the output will be completely different.

### Digital signature

In this example I have created Bob’s pair of keys and used the message digest to generate the signature. As previously mentioned, the digital signature is to provide nonrepudiation, integrity and authentication. In contrast with the encryption used in symmetric encryption where the key exchange is often a problem, the RSA system uses a pair of private and public key for each user which can be utilized to verify the sender through the digital signature.

### Digital Signature formula

The signature can be generated using the public exponent and verified with the private key.

S = xe mod n

V = yd mod n

## Cryptosystem

The cryptosystem design is taking advantages of both symmetric and asymmetric methods. The following components ensure the efficiency and the security of the cipher:

* The block ciphers to encrypt the plaintext
* A fixed-size key block
* The RSA private and public keys for each user
* The digital signature to verify the users

The block cipher offers a secure symmetric encryption method as we add permutation confusion diffusion and key schedule rounds to encrypt the message. it is also fast and computably easy. Therefore, we pick the block cipher over the stream cipher as it comes with a fix-size key block that allows message encryption of any size. The RSA Asymmetric system is used as followed:

* Encrypt and decrypt the key block with a pair of public and private key
* generate the digital signature which provides nonrepudiation, authentication and integrity of the communication.

## Lesson learned

What have you learnt from constructing this portfolio?

At the beginning I did not know the is different between symmetric and asymmetric cryptography and many other related cryptographic principles like:

* How Cryptography secures the information?
* How cryptography can be used effectively Against classic attacks social attacks and reverse engineering threats?
* How some math formulas prevent cryptoanalysis from breaking a cipher even if the attackers know exactly the cryptographic system?
* And more in general what are the criteria to take in consideration in order to build an efficient and secure cryptographic system.

Now that we are almost at the conclusion of the unit, I have learnt that symmetric and asymmetric systems can be used combined together efficiently to protect the confidentiality, integrity and availability of the information. And, how to prevents classic threat like brutal force attacks by using strong and unique key combinations also how to ensure the security even if the attackers know the cryptographic system. for example, the use of public and private exponents seen in the demo, or the one-way function seen in the hash method. Moreover, I learnt what are the characteristics that a cryptosystem should have in order to be efficient and secure. In conclusion, cryptography is an essential part of cybersecurity and I wish to learn more about cryptanalysis in the future as the more I know about how to break the code the better I am going to be at design the system.