1st Programming Project
Cryptography
2020-1

In this first programming project, you and your team will implement a program in which you compare the efficiency of the algorithms shown below. Remember that, on this occasion, you do not need to implement the algorithms, you only use the implementations that the language you choose has. To do this, you need to use the testing vectors provided on the following table. These will allow you to measure the time your computer takes executing each algorithm.

Algorithm	Size	Testing Vectors
RC4	Key length: 256 bits	https://tools.ietf.org/html/rfc6229
DES	Key size: 64 bits Block size: 64 bits	https://www.cosic.esat.kuleuven.be/nessie/testvectors/bc/des/Des-64-64.test-vectors
AES	Key size: 256 bits Block size: 256 bits	https://www.cosic.esat.kuleuven.be/nessie/testvectors/bc/rijndael/Rijndael-256-256.unverified.test-vectors
MD5	Hash size: 128 bits	https://www.cosic.esat.kuleuven.be/nessie/testvectors/hash/md5/Md5-128.unverified.test-vectors
SHA-1	Hash size: 160 bits	https://www.cosic.esat.kuleuven.be/nessie/testvectors/hash/sha/Sha-1-160.test-vectors
SHA-2	Hash size: 256 bits	https://www.cosic.esat.kuleuven.be/nessie/testvectors/hash/sha/Sha-2-256.unverified.test-vectors
RSA-OAEP	Key size: 1024 bits Block size: 2048 bits Encryption salt size: 160 bits Decryption salt size: 0 bits	https://www.cosic.esat.kuleuven.be/nessie/testvectors/asymenc/rsa-oaep/RSA-OAEP-1024.unverified.test-vectors
RSA-PSS	Signature size: 1024 bits Signature generation salt size: 160 bits Signature verification salt size: 0 bits	https://www.cosic.esat.kuleuven.be/nessie/testvectors/sign/rsa-pss/RSA-PSS-1024.unverified.test-vectors
DSA	1024 Bits	https://www.rfc-editor.org/rfc/rfc6979.txt

Each algorithm is used for some particular goal, therefore, you need to compare only the ones that share such a goal. For example, if you want to compare hashing algorithms you compare the efficiency of MD5, SHA-1 and SHA-2 by using the same input testing vector.

Following this idea, you need to create a table comparing the efficiency of these algorithms for the following operations:

- Encryption
- Decryption
- Hashing
- Signing
- Verifying

After the execution of the program, you should show a table showing the results for each operation. Coming back to the hashing example, after the execution of all hashing algorithms with all the hashing vectors, you should show a table similar to the following:

```
Hashing Operations

           MD5                     SHA-1                   SHA-2

 Vector 1  <time taken to hash     <time taken to hash     <time taken to hash
           vector 1 with MD5>      vector 1 with SHA-1>    vector 1 with SHA-2>

 Vector 2  <time taken to hash     <time taken to hash     <time taken to hash
           vector 2 with MD5>      vector 2 with SHA-1>    vector 2 with SHA-2>

 ...       ...                     ...                     ...

 Vector n  <time taken to hash     <time taken to hash     <time taken to hash
           vector n with MD5>      vector n with SHA-1>    vector n with SHA-2>
```

Remember that the program should be implemented in the language that you choose in one of the previous activities.

References

- NIST Official Site for testing Vectors http://csrc.nist.gov/groups/STM/cavp/
- European Project for Security Testing https://www.cosic.esat.kuleuven.be/nessie/testvectors/
- Files of testing vectors are attached to this assigment