Applied Cryptography, Monsoon 2022
Homework 2

This is a programming assignment. Question 1 is compulsory. You can choose to do any two of the other four questions. Your submission should be a zip file containing the implementation files. Your implementation can be in any programming language of your choice; however, we encourage using C/C++. All questions carry the same marks.

1. Using the programming language of your choice implement the RSA algorithm. Your task is to construct three functions `Keys`, `Enc` and `Dec`. On calling the `Keys` function, it should output the public and the private keys. The `Enc` function, on input a message m as a bit string of size 32 bits, should output the ciphertext c encrypted using the public key given by `Keys`. The `Dec` function on the other hand should output the original message m on giving as input the ciphertext c.

2. Recall from Homework 1 that TRIAD is a lightweight cipher. The specification of the cipher can be found here. Implement the TRIAD-MAC function. The function should take as input a message M of arbitrary length and output a tag T corresponding to the message M.

3. Implement the TRIAD-HASH function. The hash function should take as input a message M of arbitrary length and output a 32-byte hash value corresponding to M.

4. One of the faster randomized primality testing algorithms is the Solovay-Strassen (SS) algorithm. You can find a description of the algorithm here. If we use certain properties of Jacobi symbols, we can use the SS algorithm to test for primality in polynomial time. Construct a function that on input a natural number $N < 2^{32}$, uses the SS algorithm and with probability at least 0.9 correctly returns whether N is a prime number or not. The SS algorithm should run in time polynomial in the size of the input.

5. Signatures are the public key counterparts of the message authentication codes in private key cryptosystems. Signatures are used to verify the authenticity of the party claiming to be the holder of the secret key corresponding to the public key available to a verifier. You will learn more about the signatures in the subsequent lectures. Schnorr's signature scheme, based on Schnorr's identification protocol, is one such signature scheme. Details on the scheme can be found here. Your task is to implement three functions `KeyGen`, `Sign` and `Verify`. Details of the functions are given in Figure 22.1 of the specified document.