Introduction to Cryptography, Spring 2024

Homework 5

Due: 5/21/2024 (Tuesday)

Notes:

(1) For Part A, submit a "hardcopy" right after the class on the due day.

(2) TAs will run plagiarism check on your submitted programs. Write your own code and do not copy from others or anywhere.

Part A: Written Problems

1. Consider the elliptic curve $E_{17}(1,2)$. Compute the following values.
   a. All points on the curve.
   b. – (3, 10)
   c. (1, 2)+(3, 10)
   d. 2(3, 10)
2. Consider to use the above cure for EC-ElGamal encryption. Let G=(3,10). Assume that the private key is $n_A = 5$.
   a. What is the public key?
   b. What is the ciphertext of message $P_m = (1, 15)$ when k=4?
   c. Decrypt the above ciphertext and verify its correctness.
3. Compute the signature of M = "Welcome!" using the specified methods. The hash value of a string x is the last 4 bits of of SHA256(*x*). If a number is put into the hash function, convert it to a string by its ASCII code, such as, 25 → 0x3035.
   a. RSA: private key (d, n) = (247, 323)
   b. ElGamal: private key (q, α, X_A) = (103, 11, 37)
   c. Schnorr: private key (p, q, a, s) = (103, 17, 72, 10)
   d. DSA: private key (p, q, g, *x*) = (103, 17, 72, 7)
4. Compute the public keys of the above problem and verify correctness of the signatures.
5. Consider to use RSA with a fixed key for constructing a hash function RSAH as follows. Let message M be partitioned into blocks $B_1B_2\ldots B_n$, where each block is significantly smaller than the modulus of RSA key. The function RSAH is: $RSAH(B_1)=RSA(B_1)$ and $RSAH(B_1B_2\ldots B_n)=RSA(RSAH(B_1B_2\ldots B_{n-1})\oplus B_n)$ for $n\geq 2$. Show that RSAH is not weak collision-resistant.

Part 2: Programming Problem

This programming problem is to simulate the bitcoin mining and build a blockchain. Note that this is not the real bitcoin mining. It only verifies the difficulty of finding hash values with many leading zeros. Use Crypto++ for computing sha256.

1. Consider the following example:

   a. Initial message: "Bitcoin", where its hash value is:
   `B4056DF6691F8DC72E56302DDAD345D65FEAD3EAD9299609A826E2344EB63AA4`

   b. `Build the blockchain as follows:`

# of leading zeros	Preimage = Previous hash (in Hex)+ Nonce (32 bits, in Hex)	Hash value (in Hex), with the specified leading zeros (in Hex)
0	**B4056DF6691F8DC72E56302DDAD345D65FEAD3EAD9299609A826E2344EB63AA4** 00000000	**2767667C2AF3BE01EFAC4FB387EC27C10B9D3BEE9C5D48CFF4CFB9F523560B24**
1	**2767667C2AF3BE01EFAC4FB387EC27C10B9D3BEE9C5D48CFF4CFB9F523560B24** 0000000A	**0DE32E85C2AC9D96659D42C8A3EA3D2C05FDE384B468E6EFE062B6E21288CBCA**
2	?	?
3	?	?
…	?	?

   c. `The blockchain is specified as:`

0
B4056DF6691F8DC72E56302DDAD345D65FEAD3EAD9299609A826E2344EB63AA4
00000000
2767667C2AF3BE01EFAC4FB387EC27C10B9D3BEE9C5D48CFF4CFB9F523560B24
1
2767667C2AF3BE01EFAC4FB387EC27C10B9D3BEE9C5D48CFF4CFB9F523560B24
0000000A
0DE32E85C2AC9D96659D42C8A3EA3D2C05FDE384B468E6EFE062B6E21288CBCA
…

2. **Your program**

   a) Initial message: your ID in ASCII.

   b) Output: a blockchain like 1.(c), to file out.txt

   c) A block contains 4 separate lines. For example, the above has two blocks.

   d) Your program will be run for verifying your output.

3. **Submission**

   a. due: 5:00pm, 5/21/2024 (Tuesday)

   b. submit two files "blockchain.cpp" and "out.txt" to E3.

4. Grading: the more leading zeros your hash values have, the higher your grade is.

5. There is no on-site test for this programming problem.