Department of Electrical Engineering
Sharif University of Technology

Foundations of Blockchain
by: Dr. Mohammad A. Maddah-Ali
Fall 1399(2020)

Problem Set 1

Issued: Monday, Aban 5, 1399 **Due: Monday, Aban 12, 1399**

Question 1: If $g, h \in \mathbb{G}$ and $g^n = h$ we call g the n-th root of h. Lets say we have an element h. If w_1 is the x-th roof of h and w_2 is the y-th root of h what is the xy-th roof of h? (Assume that x and y are co-prime with each other)

Question 2: We have an airline with a list of passengers $S = \{s_1, s_2, s_3, ..., s_n\}$ and a security agency with a list of "dangerous" people $K = \{k_1, k_2, ..., k_m\}$. The security agency has to check the list of passengers to make sure that there is no "dangerous" person in the list. Both the airline and the security agency are not willing to share their plain list to each other. But they want to find the intersection of their lists. Someone suggests this approach.

1-both security agency and airline hash each of the elements of their list and publish it.
Security agency publishes $H_K = \{H(k_1), H(k_2), ..., H(k_m)\}$
Airline publishes $H_S = \{H(s_1), H(s_2), ..., H(s_n)\}$

2-Find the intersection of these two lists which leads to finding the intersection.

a) What is the problem of this method?

b) suggest a modification on this approach to solve the problem you mentioned.

Question 3: Alice and Bob want to play Rock-Paper-Scissors over Blockchain. Notice that as their decision will be published on the blockchain, they can not publish their choices in plain text. And they can not the other one can see it. What can we do in order to being able to perform this over blockchain?

Question 4: Recall the RSA method with the same notation in the slides of the class for $N, p, q, \varphi(N), e, d$.

a) (Bonus points) By having a (public-key,private-key) pair in RSA how can we efficiently factorize N? In other words how can we efficiently factorize N by having e, d, N?

b) Imagine a company wants to give each employee a different pair of keys (e, d)(N is the same for all pairs). Show that in this company any employee can decrypt any message encrypted by other employees. So there is no privacy! (**Hint:** Use part a. even if you have not solved it)

c) it is claimed that we have to keep $\varphi(N)$ secret in RSA. How can we attack RSA cryptosystem if we know $\varphi(N)$ and a public key?(public-key = (e, N))

Question 5: We have discussed about merkle tree as a way to commit to a set of files. Now we introduce another approach to commit to a set of files. A trusted party chooses two large prime values p and q and reveals $N = pq$ with a generator g in this RSA group. (It is really important that no one else knows factorization of N)
The RSA accumulator works as follows:

Assumption 1: all the values that we are going to commit are prime numbers.

- We define g as an empty accumulator. $A_0 = g$
- To commit to a value v into an accumulator A_i we set the new accumulator $A_{i+1} = A_i^v \bmod N$

Lets see a short example:
Assume we have the set $S = \{3, 5, 7\}$. The RSA accumulator for these values is $Acc(S) = g^{3 \cdot 5 \cdot 7} \bmod N$.
Now we publish the value of $Acc(S)$. (this is equivalent of publishing the Merkle root!)
Membership proof: Proving membership of an element in an accumulator requires a witness. The witness is g to the power of product of all the values in the set except the value that we are proving inclusion for.
In our previous example the proof of membership for 7 is (π_7) in which $\pi_7 = g^{3 \cdot 5} \bmod N$.
Answer these questions regarding to RSA accumulator.

- **a)** having $(Acc(S), k, \pi_k)$ (and other public parameters such as N and g) how can someone verifies if k actually belongs to the set S?

- **b)** In one or two lines compare the membership proof size of this method with the membership proof size in Merkle tree. Which proof is smaller?
- **c)** Assumption 1 mentioned that all the elements must be prime number. Otherwise it is possible to create a membership proof for an element t which is not an element in the set S. Find an example for this scenario.
- **d)** How can someone create a non-membership proof? (Hint: use Bezout theorem. Notice that if some element is not a member in S then it is co-prime with the product of all the elements in S)
- **e)** One cool feature in RSA accumulator is that you can aggregate different proofs and create a single proof proof. As an example by having a proof of membership for 3 and having a proof of membership for 7 we can create a proof for the membership for {3,7}. How can we do that?

Question 6: Someone propose a signature schemes as follows:
First all users of this signature schemes agree on a group $\mathbb{G}$ of prime order p with a generator g and a secure hash function H. The discrete log problem is believed to be hard in this group. Alice wants to sign a message.

For key generation algorithm Alice choose a private signing key $x \in Z_p^*$ and publish the public verification key $y = g^x$. Here is the algorithm to sign a message M by Alice:

- Choose a random $k \in Z_p^*$.
- Let $r = g^k$.
- Let $e = H(r||M)$. ($||$ is the concatenation of two bit strings)
- Let $s = k - xe$.

The signature of M is the pair (s, e). (Notice that y is also public)

a) Having a signature pair and the message M how can you verify if it is a correct signature from Alice?

Question 7: Someone propose another signature schemes using elliptic curves as follows:
First all users of this signature schemes agree on an elliptic curve with a generator G, a prime value n and a secure hash function H. Alice wants to sign a message M.

For key generation algorithm Alice choose a random value d between $[0, n-1]$ as her private key and calculates her public key $D = d \cdot G$. The public key can be compressed to just one of the coordinates (Lets assume $x-$coordinate). Here is the algorithm to sign a message M by Alice:

- Calculate $h = H(M)$.
- Choose a random $k \in Z_n^*$.
- Calculate $R = k \cdot G$ and take its $x-$coordinate called r.
- Calculate signature $s = k^{-1} * (h + r \cdot d) \pmod{n}$. ($k^{-1}$ is the modular inverse of k mod n. Meaning $k \cdot k^{-1} \equiv 1 \pmod{n}$)

The signature of M is the pair (s, r). (Notice that D is also public)

a) Having a signature pair and the message M how can you verify if it is a correct signature from Alice?

Question 8: We have studied RSA encryption scheme which is a public key encryption scheme. In this question we analyze another public key encryption scheme: First Alice generates a cyclic group $\mathbb{G}$ of order q with a generator g. Then Alice choose a random value $x \in Z_q^*$ (as her private key) and will compute $h = g^x$ as a part of her public key. Her public key is (G, q, g, h).

here is the encryption algorithm for encrypting a message M to her:

- Map the message M to an element m of $\mathbb{G}$ using a reversible mapping function. you can assume M is already an element in G for simplicity
- Choose a random $y \in Z_q^*$.
- Calculate $s = h^y$.
- Calculate $c_1 = g^y$.
- Calculate $c_2 = m \cdot s$
- Bob sends (c_1, c_2) to Alice.

a) Having (c_1, c_2) how can Alice decrypt this ciphertext and calculate M?