COMP 4901W – Homework 1

Name: Hangyul Son
SID: 20537267
Email: hson@connect.ust.hk

Exercise 1

1. Design a secure protocol for playing rock-paper-scissors. Formally specify the steps, the time between the steps and the cryptographic primitives used in the protocol

   Make use of cryptographic hash functions as given in the 'Simple auction' commitment scheme example during the lecture.

   a) Alice make her choice with a nonce and create a hash of finite length. Alice then digitally signs the hash with her private key and send the message over to Bob.

      Hiding property: Given value h(x)=y, we cannot find the value x;
      At most 1 minute delay,

   b) Bob receives lots of messages not only from Alice but also from malicious actors. Bob can try to decode messages using Alice's public key to verify the digitally signed messaged from Alice.

   c) Bob makes his choice with his unique nonce and create a hash of finite length. Bob then encodes the hash with his private key and send the message over to Alice.

      At most 1 minute delay,

   d) Alice also receives lots of messages from various actors over the internet. She uses Bob's public key to verify Bob's digitally signed message.

   e) If both Alice and Bob has each other's hash result, Alice and Bob sends choice from rock, paper, scissors, along with her unique choice of nonce to each other. It is also signed using their private key.

      At most 1 minute delay,

   f) Alice and Bob verifies the result and the nonce has been sent from each other. Then they hash the result and nonce they received to see if the hash result is no different from hash received from step b) and d).

      - Collision resistant: We cannot find two inputs $x, y \epsilon \sum *$ such that h(x) = h(y)

   g) If the new hash from the received nonce and hash is different from announced hash, the game is null. If the same, the game has settled.

2. Can either party cheat in this protocol? If so, can the cheating be provably detected? If it cannot be provably detected, go back to Step 1 and design a better protocol,

    a) The either party can cheat in this protocol, but the cheating will be probably detected. As cryptographic hash functions are deterministic, the earlier received hash must be a result of a choice from rock, paper, scissors with a nonce. In order to prove that one cheat, the recipient can show that hashing the later received nonce and the choice is different from what one announced earlier.

3. Prove that your protocol is immune to tampering by Alice, Bob or any third party on the network. In your proof, mention explicitly which properties of each primitive are being used in each part of the argument.

    Using proof by contradiction,
    a) My protocol is not immune to tampering by Alice or Bob
    b) Alice and Bob can change his or her choice of rock, paper or scissors after each other has announced the choice and the nonce to cheat and win the game.
    c) This can be done by finding a nonce that creates the same hash result with the winning choice. The newly found nonce and the winning choice can be announced and the game would be tampered without being detected.
    d) However due to the property of 'Collision Resistant' it is impossible find an two different input that creates the same hash result. Therefore c) contradicts with the property of Collision Resistant property which proves that my protocol is immune to tampering by Alice, Bob or any third party on the network.

    Using proof by contradiction,
    a) My protocol is not immune to tampering by Alice or Bob
    b) When Alice or Bob sends his or her hash to each other, one can find the input that created the hashed result which returns the choice and the nonce.
    c) Alice or Bob can choose what to pick from rock, paper, scissors which will definitely win the game and tamper the game as a whole.
    d) However due to 'Hiding' property of cryptographic hash functions, it is impossible to find the input when given the hashed result. The function is irreversible and has no or very little relationship with the input. So a very small change of the input can make the hashed result entirely different. Therefore, this property contradicts with b) as Bob or Alice cannot find the input to the hash function to tamper the game.

    Using proof by contradiction
    a) My protocol is not immune to tampering by third party on the network.
    b) Third party can impersonate Alice or Bob to tamper the game result by finding out the private keys given the public key. Alice or Bob using each other's public key to verify identity will be no use, and a malicious entity can tamper the game result.

c) However, as it is ‘Hard to factor large integers’, malicious entity cannot feasibly calculate the private key from the public key and therefore this contradicts b), and malicious entities cannot tamper the game.

Exercise 2

1. Is this a valid symmetric encryption scheme?

   Yes

2. Is this scheme secure for a one-time communication? If not, how can we make it secure? Prove that the original method or your variant discloses no information about message to an eavesdropper Eve.
   a) Yes, the number of possible outcome of the encryption function p as 'm' can be any number between 1 to p and k is a number between 0 to p-1.
   b) The chance of Eve guessing the right 'm' is to guess the correct k, and perform $(e - k) mod\ p$. However if p is a very large prime number, and if k is a random number between 0 to p-1, the change guessing the correct 'k' is 1/p which ensure to be a secure scheme.

3. Is this scheme secure for multiple rounds of communication? If not, how can we make it secure? Prove that the original method or your variant discloses no information about message to an eavesdropper Eve.

   a) It is not secure for a multiple rounds of communication.
   b) Given that two different messages are sent, $e_1 = (m_1 + k) mod\ p$, $e_2 = (m_2 + k) mod\ p$, Eve can simply subtract the $e_1 - e_2$ to get $(m_1 - m_2) mod\ p$. This shows the relationship between the first message and the second message. As the multiple rounds of communication proceeds, the relationship between each message can have a pattern vivid enough for Eve to induce the messages.
   c) Therefore, the key must be renewed after every of communication. The safety of this variant is shown in question 2 as the answer proves that this symmetric encryption scheme is secure for a one-time communication.

Exercise 3

1. For any probability bound $pr \in [0, 1)$, design a protocol that convinces Bob with probability at least pr that Alice knows such a value k.

    a) Given that m, g, and p from $m = g^k mod\ p$ is known to both Bob and Alice. Alice randomly selects a number between $r\epsilon[0, p-2]$ and calculate $C = g^r mod\ p$ and transfer the value 'C' to Bob.
    b) Alice must prepare two different values which are 'r' and '$(x + r)mod(p-1)$' and Bob requires Alice to send one of the value from the two.
    c) Bob verifies the sent value by calculating the validity of
    '$(C * y)mod\ p \equiv g^{(x+r)mod(p-1)}$' or '$C \equiv g^r mod\ p$'
    d) Assuming that Alice does not know the value of 'k'. The probability of Alice being able to cheat in steps from a), b), and c) is 50% because Alice can prepare one of the two values. If Alice only prepares 'r' and 'C' from $C \equiv g^r mod\ p$, then she can respond to Bob's challenge. However, if Bob requests '$(x + r)mod(p-1)$' Alice would fail Bob's challenge. On the other hand, Alice can prepare a random value 'r'' and send Bob the value $C' = g^{r'} * (g^x)^{-1} mod\ p$. If Bob requests Alice $(x + r)mod(p-1)$, then Alice can send the value 'r''. Bob can verify $C' * y \equiv g^{r'} mod\ p$ because C' contains multiplicative inverse of $g^x mod\ p$. And therefore, Alice without the knowledge of 'k', can pass the Bob's challenge with the probability of 0.5.
    e) However, by executing a large enough number of rounds the provability that Alice can cheat consistently is extremely low.

2. Is it possible for Bob to obtain k based on the information disclosed to him in your protocol? If not, formally prove this. If yes, go back to Step 1.

    a) Bob cannot obtain 'k' because the random number is $r\epsilon[0, p-2]$. If 'p' is a very large prime number, this won't leak information about k.
    b) $(x + r)mod(p-1)$ can be seen as the encrypted value of $x\ mod\ (p-1)$. And therefore following the property of 'one-time-pad' this does not reveal any information about 'x'.