COMP-4541 – HomeWork 1

Farshid Adavi - 4021305965007

Before start

- Notation list:

  Red: Anything written in red means that it is private.
  d: Private Key that use for Decription.
  s: Signature Key that use for Sign.

  Blue: Anything written in blue means that it is public.
  e: Public Key that use for encryption.
  v: Verification Key that use for verify.
  n: A poblic value that all calculation are madular of n.
  h: Hash Function.

  Green: Anything written in green means that it is private at first but it can be Publish for athenticatin.
  nonce: A long random string (256 bits), used to prevent brute force attacks against the *hiding* feature provided by hash function.

- Hash Functon:

  (I) A function that takes a string of any length as input and produces a string of fixed length as output.

  (II) Let's assume that SAH256 serves as our hash function for this assignment.

  (III) Collision-resistance:
  Is one of our requirements in the goal of creating a hash function.
  We know that if SHA256 were our hash function, it would be extremely difficult to find any two distinct strings x and y such that $h(x) = h(y)$.// Extremely difficult in this context means that it is practically impossible to find such x and y or that finding them would take on the order of 10^{30} years.

  (IV) Hiding:
  Hiding is one of our requirements in the goal of creating a hash function.
  We know that if y is the output of a SHA256 hash function, we can't find any string x such that $h(x) = y$.

- Key Generation Algorithm:

  (I) Choose two large primes p, q.

(II) $n = p \cdot q$

(III) Choose a random $d \in \{1, 2, \ldots, n-1\}$

(IV) Cumpute $e = d^{-1}$ *mod* $lcm(p-1, q-1)$.
If gcd $(e, n) \neq 1$, Choose a different e, n.

(V) Publish n, e

- RSA assumption:

  Given n, e, m^e *mod* n without knowing m, p, q and d it is hard to find any of the private values.

Exercise 1

(i) No one see other players choice before making their choice.

(ii) No one can change their choice.

(iii) Transparency at the end (everyone can verify).

(iv) Design a secure protocol to play the game over the internet.

(v) Strong Security Property. It is impossible to find any information, such as player selection, winners, losers, etc., through network eavesdropping.

Step0:

- We presume that the hash function and key generation algorithm are readily accessible on the Internet.
- We presume that that two players from (Alice, Bob, and Carol) do not coordinate with each other in any way.
  Or, the two players in the game are actually not two accounts of one player.

Step1:

- Alice, Bob and Carol by using Key Generation Algorithm make their own public, private, signature and verification keys.
  Each player (Alice, Bob and Carol) must follows these steps twice after selecting p and q and calculating n in the first and second steps of the key Generation algorithm. This involves performing steps 3, 4, and 5 twice to create two separate pairs of keys. So all computations are modular n.

- Now each player (Alice, Bob and Carol) must specify her choice between 0 and 1 as C and also select a *nonce*.

- So far each player has this information:
  Alice: $\{e_{Alice}, v_{Alice}, n_{Alice}, d_{Alice}, s_{Alice}, p_{Alice}, q_{Alice}, C_{Alice}, nonce_{Alice}\}$
  Bob: $\{e_{Bob}, v_{Bob}, n_{Bob}, d_{Bob}, s_{Bob}, p_{Bob}, q_{Bob}, C_{Bob}, nonce_{Bob}\}$
  Carol: $\{e_{Carol}, v_{Carol}, n_{Carol}, d_{Carol}, s_{Carol}, p_{Carol}, q_{Carol}, C_{Carol}, nonce_{Carol}\}$

- Alice, Bob and Carol Publish their poblic values, $h(C||nonce)$ and $Sgns(h(C||nonce))$.

  Poblic values: $\{e_{Alice}, v_{Alice}, n_{Alice}, h(C_{Alice}||nonce_{Alice}), Sgns_{Alice}(h(C_{Alice}||nonce_{Alice})),$
  $e_{Bob}, v_{Bob}, n_{Bob}, h(C_{Bob}||nonce_{Bob}), Sgns_{Bob}(h(C_{Bob}||nonce_{Bob})),$
  $e_{Carol}, v_{Carol}, n_{Carol}, h(C_{Carol}||nonce_{Carol}), Sgns_{Carol}(h(C_{Carol}||nonce_{Carol}))\}$

- (i) Hiding: No one can compute C based on $h(C||nonce)$.
  (ii) Collision-resistance: No one can change their choice after Published.
  (iv) RSA assumption: The players choice is not tampered on the way, because only the sender has the signature key.
  (v) Hiding: Strong Security Property. We had assumed that any message sent over the Internet would be delivered in at most 1 minute.
  We had assumed everyone commits soon. so after a minute at most we can go for next step:

Step2:

- At the beginning of this step, Alice, Bob and Carol must have all the public information.
  So each player check $verif_v(Sgns(h(C||nonce))) = h(C||nonce)$ for the other two players by they verification keys and ensured that sender is true. Each of Alice, Bob and Carol select two new $nonce$.

- Now:
  Alice send $Enc_{e_{Bob}}(C_{Alice}||nonce_{Alice}||nonce_{AliceForBob}, Sgn_{s_{Alice}}(h(nonce_{AliceForBob})))$ to Bob
  and $Enc_{e_{Carol}}(C_{Alice}||nonce_{Alice}||nonce_{AliceForCarol}, Sgn_{s_{Alice}}(h(nonce_{AliceForCarol})))$ to Carol.
  Bob send $Enc_{e_{Alice}}(C_{Bob}||nonce_{Bob}||nonce_{BobForAlice}, Sgn_{s_{Bob}}(h(nonce_{BobForAlice})))$ to Alice
  and $Enc_{e_{Carol}}(C_{Bob}||nonce_{Bob}||nonce_{BobForCarol}, Sgn_{s_{Bob}}(h(nonce_{BobForCarol})))$ to Carol.
  Carol send $Enc_{e_{Alice}}(C_{Carol}||nonce_{Carol}||nonce_{CarolForAlice}, Sgn_{s_{Carol}}(h(nonce_{CarolForAlice})))$ to Alice
  and $Enc_{e_{Bob}}(C_{Carol}||nonce_{Carol}||nonce_{CarolForBob}, Sgn_{s_{Carol}}(h(nonce_{CarolForBob})))$ to Bob.

- (iv) RSA assumption: The players choice is not tampered on the way, because only the sender has the signature key.
  (v) RSA assumption: Strong Security Property.

  After a minute at most:

step 3:

- Now each player: $(Dec_d(Enc_e(C||nonce||nonce_{newNonce}, Sgn_{s_{Carol}}(h(nonce_{CarolForBob}))))$
  Check: $h(nonce_{newNonce}) = verif_v(Sgn_{s_{Carol}}(h(nonce_{CarolForBob}))))$.
  And check: $h(C||nonce)$ be equal to $h(C||nonce)$ that published by the player at step1.

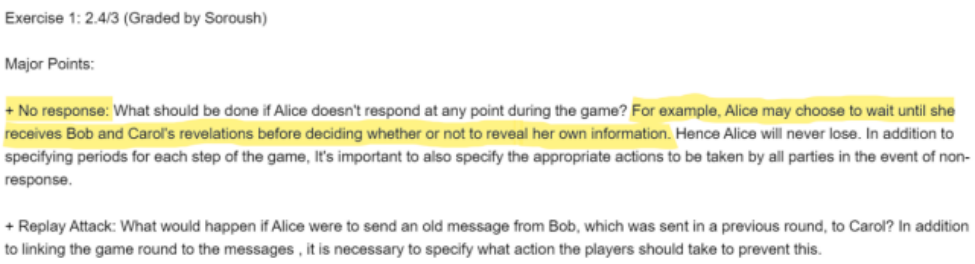
- (iii) We have transparency at the end because each player can verify others by calculat Dec_deveryone can verify.
  (vi) A secure protocol was designed. In the second part, we see the reason for some choices. (v) RSA assumption: Strong Security Property.

None of the parties can cheat in this protocol.

In step1 each player Published $h(C||nonce), Sgn_s(h(C||nonce))$ While they could have done the same in step2 and send $Dec_d(h(C_{Alice}||nonce_{Alice}), Sgn_s(h(C_{Alice}||nonce_{Alice})))$.
In this scenario, one player could send different choices to the other two players.

In step 2, each player generated two new nonces and signed the hash of them, sending the hashes separately to the other two players. Why did we create new nonces? Because in step 1, we published $Sgn_s(h(C||nonce))$ without encryption. So when sending $h(C||nonce)$ it is meaningless to sign it or not because the unique sender will not be identified, players and malicious people have this message.
On the other hand, not signing the message may also lead to problems. For instance, if Alice and Carol's messages reach Bob first and Bob realizes he is the loser, before Alice's message reaches Carol and Carol's message reaches Alice, Bob may alter their messages and present Carol's message to Alice as his own message, and vice versa for Alice. This could result in confusion and cheating, with no means for Carol and Alice to detect it. Therefore, it is preferable to generate two new nonces, hash them, sign both, encrypt one with the public key of the first opponent, and the other with the public key of the second opponent, before sending them.

- Recently added:
  + No response:



Exercise 1: 2.4/3 (Graded by Soroush)

Major Points:

+ No response: What should be done if Alice doesn't respond at any point during the game? For example, Alice may choose to wait until she receives Bob and Carol's revelations before deciding whether or not to reveal her own information. Hence Alice will never lose. In addition to specifying periods for each step of the game, It's important to also specify the appropriate actions to be taken by all parties in the event of non-response.

+ Replay Attack: What would happen if Alice were to send an old message from Bob, which was sent in a previous round, to Carol? In addition to linking the game round to the messages , it is necessary to specify what action the players should take to prevent this.

According to these highlights in the text of exercise 1:

Exercise 1

This exercise accounts for 3% of your total grade.

Alice, Bob and Carol are playing a game in which one of them is selected as a loser and has to pay some money to the others. The game is quite simple. They each choose either 0 or 1. The player whose choice is in the minority loses, i.e. one loses if they choose 0 and both other players choose 1 or vice versa. It is possible that there is no loser, e.g. if all players choose 1. In that case, the game will be restarted and played for another round.

Unfortunately, Alice, Bob and Carol are in different countries and they have to play the game over the internet. Assume that any message sent over the internet will be delivered in at most 1 minute. However, there might be malicious actors on the network who try to impersonate Alice, Bob or Carol or otherwise tamper with the result of the game. These malicious actors can read the messages and also change them, but they cannot stop the messages. Moreover, they are discreet and will not change a message if this can be detected in your protocol. Also, none of the players (Alice, Bob or Carol) are trustworthy and they might want to cheat in order to win the game or at least decrease their chance of losing.

1. Design a secure protocol to play this game over the internet. Formally specify the steps, the time between the steps and the cryptographic primitives used in the protocol.
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.
3. Prove that your protocol is immune to tampering by Alice, Bob, Carol or any third party on the network. In your proof, mention explicitly which properties of each cryptographic primitive are being used in each part of the argument.

We have by default that no one can stop sending messages and the maximum time for sending messages is 1 minute. So there is no justification for not responding (assuming that the Internet of all players is strong and does not disconnect).

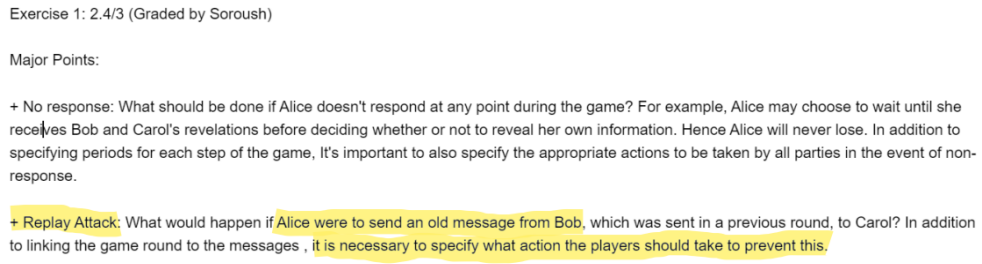
signature key.
(v) Hiding: Strong Security Property. We had assumed that any message sent over the Internet would be delivered in at most 1 minute.
We had assumed everyone commits soon. so after a minute at most we can go for next step:

Step2:

- At the beginning of this step, Alice, Bob and Carol must have all the public information.
  So each player check $verif_v(Sgn_s(h(C||nonce))) = h(C||nonce)$ for the other two players by they verification keys and ensured that sender is true. Each of Alice, Bob and Carol select two new $nonce$.
- Now:
  Alice send $Enc_{e_{Bob}}(C_{Alice}||nonce_{Alice}||nonce_{AliceForBob}, Sgn_{s_{Alice}}(h(nonce_{AliceForBob})))$ to Bob and $Enc_{e_{Carol}}(C_{Alice}||nonce_{Alice}||nonce_{AliceForCarol}, Sgn_{s_{Alice}}(h(nonce_{AliceForCarol})))$ to Carol.
  Bob send $Enc_{e_{Alice}}(C_{Bob}||nonce_{Bob}||nonce_{BobForAlice}, Sgn_{s_{Bob}}(h(nonce_{BobForAlice})))$ to Alice and $Enc_{\cdot}(C_{\cdot}||nonce_{\cdot}||nonce_{\cdot}, Sgn_{\cdot}(h(nonce_{\cdot})))$ to Carol

Due to the default, no one can stop sending messages, in the previous answer to this exercise; at end of step 1 I assumed that everyone sends messages quickly and at the beginning of step 2 I said all parties in the

game must resived the other's messages.
I will add here that with all this, if someone avoids responding, what is the appropriate actions to be taken by all parties.
At the end of step 1, if a player doesn't send any response, we don't need to be too strict. After one minute at the beginning of step 2 it is enough to expel him from the game and replace him with another player. But at the end of step 2, if someone does not send any response, After one minute at the beginning of step 3 we will declare him the loser and go to the next round. + Replay Attack:



Exercise 1: 2.4/3 (Graded by Soroush)

Major Points:

+ No response: What should be done if Alice doesn't respond at any point during the game? For example, Alice may choose to wait until she receives Bob and Carol's revelations before deciding whether or not to reveal her own information. Hence Alice will never lose. In addition to specifying periods for each step of the game, It's important to also specify the appropriate actions to be taken by all parties in the event of non-response.

+ Replay Attack: What would happen if Alice were to send an old message from Bob, which was sent in a previous round, to Carol? In addition to linking the game round to the messages , it is necessary to specify what action the players should take to prevent this.

In order to prevent Replay Attack, we also concate the game round with other message information such as player selection and nonce so that if the message of the previous rounds is used, the manipulation of the message can be recognized, as a result, the malicious will not do this manipulation.

Exercise 2

Now that exercise one has been solved in detail above and exercise two is almost the same. I will explain exercise two a little more briefly.
Since the public keys are available in the text of the exercise, we assume that we do not have a signature key.

1.

- Buyer ith makes his bid as b_i, and generate a nonce as $nonce_i$ then send to Alice and all other buyer(j use for recevers) $Enc_{Pubk_j}(h(b_i||nonce_i), Sgn_{Prik_i}(h(h(b_i||nonce_i))))$.
  Hiding: No one can compute b_i based on $h(b_i||nonce)$.
  Collision-resistance: No one can change their choice.
  No buyer can send bid instead of other buyers because the requests are signed with the private key of the buyers.
- Buyer ith send to Alice and all other buyer(j use for receivers) $Enc_{Pk_j}(b_i||nonce_i, Sgn_{Prik_i}(h(h(h(b_i||nonce_i)))))$.
  Collision-resistance: No one can change their choice.

- Now Alice announces the first and second buyer and sells the lecture notes. Alice cannot lie because everyone knows all the bid amounts.

2.

- Buyer ith makes his bid as b_i, and generate a nonce as $nonce_i$ then send to Alice and all other buyer(j use for recevers) $Enc_{Pubk_j}(h(b_i||nonce_i), Sgn_{Prik_i}(h(h(b_i||nonce_i))))$ Hiding: No one can compute b_i based on $h(b_i||nonce)$.
  Collision-resistance: No one can change their choice.
  No buyer can send bid instead of other buyers because the requests are signed with the private key of the buyers.
- Buyer ith send to Alice $Enc_{Pk_A}(b_i||nonce_i, Sgn_{Prik_i}(h(b_i||nonce_i)))$ Collision-resistance: No one can change their choice.

- Now Alice announces the first and second buyer b_i and $nonce_i$ and sells the lecture notes.
  If one buyer wants to cheat and send different amounts to Alice and other buyers, he will fail because his signature is on the amount he sent to Alice.
  On the other hand, if Alice wants to cheat, there are two situations. First, Alice does not think about her own profit and wants to sell the lecture notes to a specific buyer. In this case, the buyer, whose amount is higher than the second and maybe even the first person, protests and submits his b_i and $nonce_i$, and to reject this claim, Alice only has to submit another request with the buyer private key, and as a result, she is exposed.
  On the other hand, if Alice wants to falsely place a large amount as the first or second amount in order to sell the lecture note more expensively, everyone can get the hash of that amount and find out that it was not in the requests and ask Alice for the signed message of this request.

3.

- Buyer ith makes his bid as b_i, and generate a nonce as $nonce_i$ then send to Alice $Enc_{Pubk_A}(h(b_i||nonce_i), Sgn_{Prik_i}(h(h$
  Hiding: No one can compute b_i based on $h(b_i||nonce)$.
  Collision-resistance: No one can change their choice.
  No buyer can send bid instead of other buyers because the requests are signed with the private key of the buyers.
- Now Alice must publish the hashes before Buyers send to her $Enc_{Pk_A}(b_i||nonce_i, Sgn_{Prik_i}(h(b_i||nonce_i)))$..
  Hiding: No one can compute b_i based on $h(b_i||nonce)$.
  Alice still does not know the offers, so she cannot cheat and add a fake offer to the hashes. Because there is very little chance for that amount to be exactly the second purchase offer in the future. If it is the first institution that will be revealed. If it is third or more suggestions, it will not help.

- Buyer ith send to Alice $Enc_{Pk_A}(b_i||nonce_i, Sgn_{Prik_i}(h(b_i||nonce_i)))$. Collision-resistance: No one can change their choice.

- Now Alice announces the first and second buyer b_i and the first and second buyer $nonce_i$ and sells the lecture notes.

If one buyer wants to cheat and send different amounts to Alice and other buyers, he will fail because his signature is on the amount he sent to Alice.
On the other hand, if Alice wants to cheat, there are two situations. First, Alice does not think about her own profit and wants to sell the lecture notes to a specific buyer. In this case, the buyer, whose amount is higher than the second and maybe even the first person, protests and submits his b_i and $nonce_i$, and to reject this claim, Alice only has to submit another request with the buyer private key, and as a result, she is exposed.
On the other hand, if Alice wants to falsely place a large amount as the first amount in order to sell the lecture note more expensively, everyone can get the hash of that amount and find out that it was not in the requests.

- Recently added:
  + Replay Attack:

  In order to prevent Replay Attack, we assume that each auction has a new specific serial number. We also concate this serial number with other message information such as bidder's bid and nonce.

Exercise 3

1. Yes, why not. When the final verification of the transaction requires the bank's signature and the transaction contains a pointer to the last transaction of the chain of transactions, before signing and confirming, it is enough for the bank to take the traces of the coins in the transaction awaiting verification in the chain of transactions to go to the first transaction. where these coins came from and which is a transaction created by the bank itself because only the bank creates coins. Here, if it includes that particular coin, the bank will not sign it. It is as if that coin is frozen and cannot be spent.

2. No, we saw that the ownership of coins changed only with the transaction and the signature(by private key) of the owner (public key) of the coin is essential for every transaction.

3. No, when the bank signs a transaction, it is sent to everyone and added to the chain of transactions. Now reversing this transaction is equivalent to changing the ownership of a number of coins, which we have seen is not possible. If that block is to be removed from the chain of transactions, then all subsequent blocks will need to be changed because all the hashes will be destroyed.

4. No, if the sum of the incoming coins is more than the outgoing coins and the bank is not the originator of this transaction, no one will publish it and the bank itself will not sign it.

5. If the central bank has the public key of this particular person. Yes, why not. It can blacklist it and not sign transactions that contain this key. But in general, the bank does not have a key list of people's names and their public keys.

6. If the central bank has the public key of this particular person. Yes, why not. It can whitelist it and sign transactions that contain this key. But in general, the bank does not have a key list of people's names and their public keys.

7. No, it cannot because the identity of each person in the network is only one or more public keys related to her accounts.

8. Yes, why not, we prepare a white list of people's names and public keys, and everyone who gives us their name and public key by presenting documents, we enter this white list.

9. Yes, the bank can do this; it is enough for the head to consider a number of valid transactions, then connect them with the hash pointer of the previous transaction and sign them. But any node in the network can check the incompatibility of the two chains and complain to the court. We are aware that the bank's identity is not hidden unlike other members.