1845479-HW07

Carmignani Federico

December 29, 2021

Abstract

The goal of this homework is to define a secure protocol for playing ´K six-sided dices'.
The solution has to be sent to cns@diag.uniroma1.it from the institutional email.
The deadline is December 30^{th} at 11:59 pm.

1 What is the challenge

The challenge consists in creating a protocol for playing ´K six-sided dices'.
The goal is to define a protocol for letting Alice and Bob to play dice using K six-sided dices imagining that they act remotely and, without violating protocol rules, will try to cheat to win.
A possible "Man in the Middle" (MITM) attack is not considered, otherwise a form of authentication is needed, like a digital signature, in the communication; now the protocol just is concentrating on Alice and Bob so that they do not cheat.
It should allow to play games and secure the operations, bearing in mind that for each game, the goal of the players is to get the highest score.
The dice game is famous and a single game consists in simultaneous throwing of all the dices by each of the two players, who will sum all the results and will compare the sums; the winner of a single game is who gets the highest score.
A match is instead a sequence of fixed-length games.

Security requirement: Prevent cheatings of the two players in the game.

2 How to solve the challenge

The challenge is based on proposing a protocol that makes this game secure.
This is a problem known as **multiparty computation**, where different parties have to make a computation, in this case it is a game, without trusting each other.
A third party is not an acceptable solution, otherwise it becomes too much easy to create a protocol to reach te security requirement.
The protocol proposed is based on existing protocols, the **DH protocol for key-exchange**, and algorithms, like the **SHA-256** used as cryptographic hashing function, considered secure nowadays.
Besides, it is characterised by a **public key cryptographic system**.

2.1 The protocol

The protocol for a game is based on these steps:

- Alice and Bob exchange their public keys: PK_A and PK_B.
- Alice and Bob exchange two public parameters: g, p parameters of DH protocol.
- Alice generates a random salt a and computes $A = g^a \ mod(p)$.
- Alice sends {A, sign}PK_B.
- Bob generates a random salt b.

Figure 1: Six-sided dices.

- Bob sends {b, sign}PK_A.
- Alice sends {a, sign}PK_B.
- Bob verifies that $g^a \ mod(p)$ is equal to A received before.
- Alice and Bob convert b to a base-$[7 + 6(K - 1)]$ integer, representing Bob's dices throw sum.
- Alice and Bob convert a to a base-$[7 + 6(K - 1)]$ integer, representing Alice's dices throw sum.

The two salts created by the two parties are representing in integer representation the sum of the K dices throw obtained by each one, so it is including all possible number of dices.
The messages are encrypted with public keys, thus it is only possible for the other party to decrypt the message using their private keys and therefore ensuring confidentiality.
Furthermore when a party signs the message, the other is able to verify that the message has not been changed, therefore ensuring authenticity. It is possible using an hashing function like SHA-256, which is considered secure for its high output size; so the birthday bound is high and the bruteforce attack is not possible in practice, to find collisions.
This protocol exploits a technique based on a **"locked box"**: when Alice sends the first message to Bob, he cannot retrieve a since the function is a One-Way-Function, not invertible, but in any case it is like if a locked box containing the choice of Alice is kept by Bob; Bob will send his choice to Alice, he cannot cheat since he does not know Alice's choice in that moment, then Alice will send her choice to Bob directly. If Alice cheats, Bob will know it computing the first message sent by Alice, when she did not know about Bob's choice.
The use of random salts a and b in with an high upper bound will permit to reduce the probability of finding collisions in Bob's computation.
They convert their chosen salts in a number representing the sum of the throws of K dices, converting them to a base-$[7 + 6(K - 1)]$ integers. Using the technique of the locked box, they cannot cheat, because they communicate their choices before knowing that of the other party; Alice will be forced to send his choice after receiving Bob's one by a timer set after Bob's message arrives, if Alice does not respond in this time the game ends and Bob will win. The numbers computed in the last steps are the choices of the two players and they will know both their choices, so they cannot cheat about the winning player of the game. Finally, since it is possible to play a match, defined as a sequence of games, and a match is secure through this protocol, they cannot cheat about the winning of the match: the two player keep in memory all choices done in every game and the winner of each game, at the end of the match they will communicate to the other if she/he is a winner or not; they cannot cheat since both the players know all the numbers of all the single games.
A similar protocol could be be extended to all remote games between two parties that do not trust each other.