CSCI262 – System Security

Client Puzzle

Sionggo Japit

sjapit@uow.edu.au

17 November 2020

Question 1 …1

1) One of the client puzzles we considered contained the statement $h(C, Ns, Nc, Y) = 000 \ldots 000X$.
   a. Describe each of the components in the expression above.
   b. How much work is required to “solve” the puzzle, in the context of this statement?
   c. What is the purpose of such a puzzle?
   d. Describe how we could modify this to generate sub-puzzles.
   e. What advantage do we obtain by using many sub-puzzles rather than just one single large puzzle?

Question 1 …2

$$h(C, Ns, Nc, Y) = 000\ldots000X.$$

a) Describe each of the components in the expression above.

Question 1 …2

$$h(C, Ns, Nc, Y) = 000 \ldots 000X.$$

a) Describe each of the components in the expression above.

- h: a cryptographic hash function (e.g., MD5 or SHA)
- C: the client identity
- N_s: the server's nonce
- N_c: the client's nonce
- Y: the solution of the puzzle
- $000 \ldots 000$: the k first bits of the hash value; must be zero. The reasonable values of k lie between 0 and 64.
- X: the rest of the hash value; may be anything

Question 1 …3

b) How much work is required to “solve” the puzzle, in the context of this statement?

Question 1 …3

b) How much work is required to “solve” the puzzle, in the context of this statement?

The cost of solving the puzzle depends exponentially on the required number of k of zero bits in the beginning of the hash. If $k = 0$, no work is required. If $k = 64$, then in the worst case, it would be 2^k. In such a puzzle, the reasonable values of k lie between 0 and 64.

Question 1 …4

c) What is the purpose of such a puzzle?

Question 1 ...4

c) What is the purpose of such a puzzle?

The purpose of such a puzzle is to ensure that the client should always commit its resources to the authentication protocol first and the server should be able to verify the client commitment before allocating its own resources.

Question 1 …5

d) Describe how we could modify this to generate sub-puzzles.

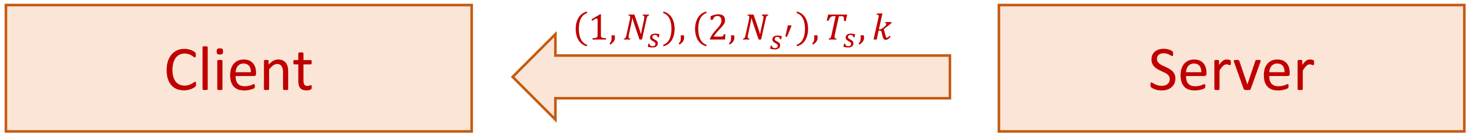
Question 1 ...5

d) Describe how we could modify this to generate sub-puzzles.

The modification can be done as follow:

The Server:

- Server determines k.
- Server generates two nonce - N_s and $N_{s'}$, and a timestamp T_s.
- Server sends the two puzzles with sequence number - $(1, N_s)$ and $(2, N_{s'})$, the timestamp T_s and the puzzle difficulty level k to client.



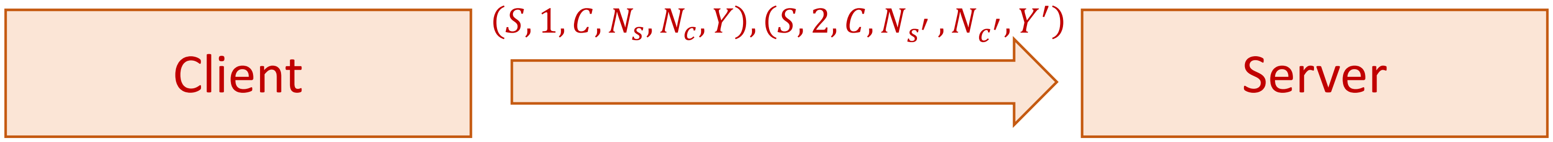
Question 1 …6

The Client:

- Client receive the puzzles and commits its resources into solving the puzzle.
- Client verifies the timestamp T_s.
- Client generates two nonce $N_c\ and\ N_{c'}$
- Client finds Y such that $h(1, C, N_s, N_c, Y) = 000 \ldots 000X$, where $000 \ldots 000$ denotes $k\ 0's$, and X can be any value.
- Client finds the second solution Y' such that $h(2, C, N_{s'}, N_{c'}, Y') = 000 \ldots 000X$, where $000 \ldots 000$ denotes $k\ 0's$, and X can be any value.

Question 1 …7

- Client sends the solutions (solved puzzles) $(S, 1, C, N_s, N_c, Y)$ *and* $(S, 2, C, N_{s'}, N_{c'}, Y')$ to server.



Question 1 …8

The Server:

- verifies that $N_s \ and \ N_{s'}$ are recent.
- checks that $C, N_s, N_{s'}, N_c, and \ N_{c'}$ have not been used before.
- checks if there are $k \ 0's$ in each solution, that is, the server checks if $h(1, C, N_s, N_c, Y) = 000 \ldots 000X$ and $h(2, C, N_{s'}, N_{c'}, Y') = 000 \ldots 000X$ are correct.
- If they do, the server commits the resources, stores $(C, N_s, N_{s'}, N_c, N_{c'})$ and sends $(S, C, N_c, N_{c'})$ to the client.
- The operation can now continue.

Question 1 …9

e) What advantage do we obtain by using many sub-puzzles rather than just one single large puzzle?

Question 1 …9

e) What advantage do we obtain by using many sub-puzzles rather than just one single large puzzle?

If we use a single big puzzle instead of sub puzzles, then the difficulty level is hard to adjust. This is because one bit of change in *k* could require a much longer time to solve the puzzle. Using sub puzzles we can fine tune the difficulty level.