SIL765 - PROGRAMMING ASSIGNMENT

PROJECT 1

Submitted By:
Mehak
2018MCS2143

PROBLEM STATEMENT

You are required to build clients A and B that wish to send messages from B to A (only B to A) but encrypted using Elgamal Cryptosystem, but only after having exchanged messages that finally result in computation of one‑time keys using the Diffie Hellman protocol.

The Elgamal Cryptosystem uses the parameters (q, a). These are known to each other using "other means". To ensure that man‑in‑the‑middle attack is not possible, the two clients A and B send initial messages to each other by adding to each message a MAC (or an "integrity check") that itself is based on HMAC algorithm and uses a shared "secret".

To do so, you will need to:

- ensure that clients already (somehow) know the shared "secret" to be used with HMAC,
- assume that Diffie‑Hellman parameters (q, a) are already fixed, and known to them,
- (once they know how to generate or compute one‑time passwords) messages sent from B to A are encrypted using Elgamal cryptosystem with parameters, (q, a).

Once the one‑time keys have been computed (or can be computed on the fly), B should encrypt and send messages "hello 1", "hello 2", "hello 3".

INTRODUCTION

ElGamal encryption is an public-key cryptosystem. It uses asymmetric key encryption for communicating between two parties and encrypting the message. This cryptosystem is based on the difficulty of finding discrete logarithm in a cyclic group that is even if we know g^a and g^k, it is extremely difficult to compute g^{ak}.

SYSTEM SPECIFICATION

Language used : Python

Assumptions

- ensure that clients already (somehow) know the shared "secret" to be used with HMAC
- assume that Diffie-Hellman parameters (q, a) are already fixed, and known to them

PROCEDURE

Suppose Alice wants to communicate to Bob.

1. Bob generates public and private key :
   - Bob chooses a very large number q and a cyclic group F_q.
   - From the cyclic group F_q, he choose any element g and an element a such that $gcd(a, q) = 1$.
   - Then he computes $h = g^a$.
   - Bob publishes F, $h = g^a$, q and g as his public key and retains a as private key.
2. Alice encrypts data using Bob's public key :
   - Alice selects an element k from cyclic group F such that $gcd(k, q) = 1$.
   - Then she computes $p = g^k$ and $s = h^k = g^{ak}$ mod q
   - She multiples s with M.
   - Then she sends $(p, M*s) = (g^k, M*s)$.
3. Bob decrypts the message :
   - Bob calculates $s^{'} = p^a = g^{ak}$ mod q.
   - He divides $M*s$ by $s^{'}$ to obtain M as $s = s^{'}$

IMPLEMENTATION

Client B:


```
meh@meh-Len:~/Documents/nss/ass4$ python clientB.py
secret  3a9402ce5a7b14c4312ace99fc517cb70345332fd37deee7635fd245f6a19418
q  6242741714182298041782933352952436862011268983782
alpha  695014620412076759967480322061432183457765550790 3
Message received from A 373974408443907134466764849515555194772735512906 81||350ec89638593ae488f0caeec3d666345c066abc1a68884ee44fb3c
b4b7016ea
a^XA mod q =  37397440844390713446676484951555519477273551290681


Enter q to quit


a^XB mod q =  3858629551042761102459061086209873246962303876 85
a^(XA*XB) mod q =  277595510278023226250548859421947085416583244 33639
B: Hello1
Message sent by B (original):  Hello1||3858629551042761102459061086 20987324696230387685||48db62443db270ee13b758818877b5485d910001e3
91c3f4b40895470f97c528
Message sent by B (encrypted):   1998687674001767229003951787838019 014999399359222008#2803714653808034585130543480161665562707490767
797539#29980315110026508435059276817570285224990990388330 12#299803151100265084350592768175702852249909903883301 2#308131016408605781
13810923395836126481240740121339 29#1360218000362313808627689411167540718541257897248311||3858629551042761102459061086209873246962 30
387685||48db62443db270ee13b758818877b5485d910001e391c3f4b40895470f97c528


a^XB mod q =  1716737791094558438540250227061416732111726352 2509
a^(XA*XB) mod q =  208769451145514455387441387762956553131871200 15361
B: Hello2
Message sent by B (original):  Hello2||1716737791094558438540250227 0614167321117263522509||d6b289caaa8f787252ed30ef4013aca7b1e73c20
d7d5c21f904c1835f1f2a442
Message sent by B (encrypted):   1503140048247704078789577991893287 182549472641105992#210857145656969599941315801640586118663189912 1
551461#225471007237155611818436698783993077382420896165898 8#2254710072371556118184366987839930773824208961658988#231734090771521045
480059940416881773976377032170507 1#1043847255727572276937206938814782765659356000768050||1716737791094558438540250227061416732111 72
63522509||d6b289caaa8f787252ed30ef4013aca7b1e73c20d7d5c21f904c1835f1f2a442
```

Client A:


```
meh@meh-Len:~/Documents/nss/ass4$ python clientA.py
secret  3a9402ce5a7b14c4312ace99fc517cb70345332fd37deee7635fd245f6a19418
q  6242741714182298041782933352952436862011268983782
alpha  695014620412076759967480322061432183457765550790 3
a^XA mod q =  37397440844390713446676484951555519477273551290681
Message sent to B 37397440844390713446676484951555519477273551290681||350ec89638593ae488f0caeec3d666345c066abc1a68884ee44fb3cb4b701
6ea


Message received from B 1998687674001767229003951787838019014999399359222008#2803714653808034585130543480161665562707490767797539#2
99803151100265084350592768175702852249909903883301 2#29980315110026508435059276817570285224990990388330 12#3081310164086057811381 0923
3958361264812407401213392 9#1360218000362313808627689411167540718541257897248311||3858629551042761102459061086209873246962 30387685||
48db62443db270ee13b758818877b5485d910001e391c3f4b40895470f97c528
a^XB mod q =  38586295510427611024590610862098732469623 0387685
YB  3858629551042761102459061086209873246962303876 85
a^(XA*XB) mod q =  277595510278023226250548859421947085416583244 33639
Message received from B (decrypted):  Hello1


Message received from B 1503140048247704078789577991893287182549472641105992#2108571456569695999413158016405861186631899121551461#2
25471007237155611818436698783993077382420896165898 8#2254710072371556118184366987839930773824208961658988#231734090771521045480059 94
0416881773976377032170507 1#1043847255727572276937206938814782765659356000768050||1716737791094558438540250227061416732111726352250 9
||d6b289caaa8f787252ed30ef4013aca7b1e73c20d7d5c21f904c1835f1f2a442
a^XB mod q =  1716737791094558438540250227061416732111726352 2509
YB  17167377910945584385402502270614167321117263522509
a^(XA*XB) mod q =  208769451145514455387441387762956553131871200 15361
Message received from B (decrypted):  Hello2
```