Problem 3: Hidden RSA

For calculating, we used C programming + GMP library. The code we used will be included with the document. Hash of the code file is in the end of the document

We have:

$a^e = r_a \ (mod\ n)$

$b^e = r_b (mod\ n)$

$c^e = r_c (mod\ n)$

$(a.b)^e = r_{ab}\ (mod\ n)$

=> $r_a r_b = r_{ab} (mod\ n)$

=> $r_a r_b - r_{ab} = 0\ (mod\ n)$ (1)

$(a.c)^e = r_{ac}\ (mod\ n)$

=> $r_a r_c = r_{ac} (mod\ n)$

=> $r_a r_c - r_{ac} = 0\ (mod\ n)$ (2)

From (1) and (2) => $GCD(r_a r_b - r_{ab}, r_a r_c - r_{ac}) = 0\ (mod\ n)$

=> $GCD(r_a r_b - r_{ab}, r_a r_c - r_{ac}) = k.n$ (3)

Using Bob's website, we attained:

Encr(2)=501549122890393350146693397733083936426581232289658730787378
60474494117389068

Encr(3)=741771676788668065199293373666893139393000154892388645416796
30476008627210599

Encr(5)=667880511648659482237836053968696774450563522678679686402348
39015540677264876

Encr(6)=697328357118522530440751852485029707147296293733863361949277
84886349053828079

Encr(10)=361145734862708060551493342928300105044705144314793634373027
3690048446896189

Using the method (3), we calculated:

k.n=762007084434332500125013429920335715869717602189347569300586616278
67825188509

We know that n > y, and $kn/2 < y$ => k=1

(or we can use factorization to calculate k.p.q)

Then, we test 4 most common values of e:

{3, 5, 17, 257, 65537}

Turned out, e = 65537 (as expected :))

For factoring number, we used the website:
https://www.alpertron.com.ar/ECM.HTM

We factorized n into p and q.
p=232086664036792751646261018215123451301
q=328328681700354546732404725320581286809

With p and q, we can calculate $\varphi(n) = (p-1)(q-1)$
Then we calculated $d = e^{-1} \ (mod\ n)$ and $x = y^d \ (mod\ n)$
d=58041460011714671214337771652949080061981291861469879231637604933853779098273

At last, we get
x=202010181600

File included:
hiddenrsa.c
SHA256:
9dd3c82db51f794a599750838ba99964518f1da1dcf8e52748d25c460ffbbb58