Hw9 2018/4/6

Yanliang 112889478

1.

sage:
n=**125953175678398351570149977764211035679420156938429586850000579961775054888014711050952194404928504160243324417202380464659083542772305519159214463831847643286738542961736012 1**

sage: R=Integers(n)

sage: p=**112889478**

sage: e=**65537**

sage: R(p)^e

108482139001536853276560886728303271558960379761298326497284830355601702495894419878068953895953297676910592506895329310961221129570474792780301452977162684488559739785998967 3

That is sage output for my student Id,

It is not safe, since the student id has a small possible plaintext space, say it the hacker can exhausting all the possible id start with 11 and then another 7 random digit and encrypt them and compare with the encrypted text that they can get, so they know what the plain text is, so it is not safe.

It is a chosen-plaintext attack, since the hacker can give u a plain text and ask you for the encryption of that plain text.

(also, it is kind of known-plaintext attack, since you always know that your id starts with 11)

So, the solution for this is enlarge the size of the plaintext by some optimization padding algorithms.

2.

Size: 256 Bytes / 2048 Bits

9C 2A 04 77 5C D8 50 91 3A 06 A3 82 E0 D8 50 48 BC 89 3F F1 19 70 1A 88 46 7E E0 8F C5 F1 89 CE 21 EE 5A FE 61 0D B7 32 44 89 A0 74 0B 53 4F 55 A4 CE 82 62 95 EE EB 59 5F C6 E1 05 80 12 C4 5E 94 3F BC 5B 48 38 F4 53 F7 24 E6 FB 91 E9 15 C4 CF F4 53 0D F4 4A FC 9F 54 DE 7D BE A0 6B 6F 87 C0 D0 50 1F 28 30 03 40 DA 08 73 51 6C 7F FF 3A 3C A7 37 06 8E BD 4B 11 04 EB 7D 24 DE E6 F9 FC 31 71 FB 94 D5 60 F3 2E 4A AF 42 D2 CB EA C4 6A 1A B2 CC 53 DD 15 4B 8B 1F C8 19 61 1F CD 9D A8 3E 63 2B 84 35 69 65 84 C8

19 C5 46 22 F8 53 95 BE E3 80 4A 10 C6 2A EC BA 97 20 11 C7 39 99 10 04 A0 F0 61 7A 95 25 8C 4E 52 75 E2 B6 ED 08 CA 14 FC CE 22 6A B3 4E CF 46 03 97 97 03 7E C0 B1 DE 7B AF 45 33 CF BA 3E 71 B7 DE F4 25 25 C2 0D 35 89 9D 9D FB 0E 11 79 89 1E 37 C5 AF 8E 72 69

RSA public key for www.google.com and the exponent is 65537.