Being Tre Assignment

Section 1 : Diffie Hellman

$g = 5$ $P = 103$

Alice $A = 10$, Bob

$B = 71$

- Let a and b be the secret keys of Alice and Bob respectively
- A and B is computed using:

$A = g^a \bmod P$ $B = g^b \bmod P$

$A = 5^a \bmod 103$ $B = 5^b \bmod 103$

- The shared secret is $S = g^{ab} \bmod 103 = A^b \bmod 103 = B^a \bmod 103$
- Using the Baby-Step Giant-Step algorithm,

$m = \sqrt{n}$, n = group order.

- for 103, the group order $= 103 - 1 = 102$
- So $m = \lceil \sqrt{102} \rceil = 11$

Baby steps

Computing $5^i \bmod 103$ for $i = 0 \dots 10$

i	0	1	2	3	4	5	6	7	8	9	10
$5^i \bmod 103$	1	5	25	22	7	35	72	51	49	39	92

- Computing g^{-m}

$g^{-m} = 5^{-11} = (5^{11})^{-1}$

$5^{11} \bmod 103 = 15$

$\Rightarrow (5^{11})^{-1} \equiv 15^{-1} \bmod 103 \equiv 88 \bmod 103$

- finding a (Grant steps)

$A = 10$

Computing $10 \cdot (88)^j \bmod 103$ for $j = 0 \dots 10$

j	0	1	2	3	4	5	6	7	8	9	10
$10 \cdot (88)^j \bmod 103$	10	56	87	34	5	28	95	17	54	14	99

- 5 is found at position $i = 1$ and $j = 4$ in tables 1 & 2

So $a = i + jm$

$= 1 + 4 \cdot 11$

$\boxed{a = 45}$

Finding b (Giant steps)

$B = 71$

Computing $71 \cdot (58)^k \bmod 103$ for $k = 0 \cdots 10$

k	0	1	2	3	4	5	6	7	8	9	10
$71(58)^k \bmod 103$	71	68	10	56	87	34	5	28	95	17	59

5 appears at position $i = 1$ and $k = 6$.

So $b = i + km$

$= 1 + 6 \cdot 11$

$\boxed{b = 67}$

Checking confirms that $5^{45} \bmod 103 = 10$ and $5^{67} \bmod 103 = 71$

Finding shared secret

$S = 5^{45 \cdot 67} \bmod 103$

$\boxed{S = 31}$

Where this fails for larger values of the integers

If the value of P is very large, it would take a lot of time to compute the Baby steps and giant steps that produce the tables above. We would need about $m \cdot \lceil \sqrt{P-1} \rceil$ steps, which can be very long if the value of P is very very large.

Section 2 : RSA

$(e_{Bob}, n_{Bob}) = (17, 266473)$.

↳ For each number x in the original message, Alice computes $x^{e_{Bob}} \bmod n_{Bob}$ to encrypt the message and send it to Bob.

↳ When Bob gets the encrypted message from Alice, he takes each integer y of the text and compute $y^{d_{Bob}} \bmod n_{Bob}$ to decrypt the message.

Getting d_{Bob}

↳ We want d_{Bob} such that $e_{Bob} d_{Bob} \bmod \lambda(n_{Bob}) = 1$.

↳ First, we need to compute $\lambda(n_{Bob})$

↳ Let p_B and q_B be two prime numbers such that $n_{Bob} = p_B q_B$.

↳ $\lambda(n_{Bob}) = lcm(p_B - 1, q_B - 1)$

↳ We have $n_{Bob} = 266473$

↳ Two prime numbers whose product give n_{Bob} are

$p_B = 439$ and $q_B = 607$.

$\lambda(n_{Bob}) = lcm(438, 606)$

$= 44238$

So we want d_{Bob} such that:

$17 \cdot d_{Bob} \bmod 44238 = 1$

$\Rightarrow 17 \cdot d_{Bob} - 1 = 44238\, t, \quad t \in \mathbb{Z}$

$\rightarrow 17 d_{Bob} - 44238\, t = 1$

↳ have:

$44238 = 17 \cdot 2602 + 4$

$17 = 4 \cdot 4 + 1$

Back subsituting:

$1 = 17 - 4 \cdot 4$

$1 = 17 - (44238 - 17 \cdot 2602) \cdot 4$

$1 = 17 - 4 \cdot 44238 + 17 \cdot 10408$

$1 = 17 \cdot 10409 - 44238 \cdot 4$

Comparing it $1 = 17 d_{Bob} - 44238 t$,

$d_{Bob} = 10409 \quad t = 4$

Hence $d_{Bob} = 10409 \bmod 44238$

We then use the method discussed earlier to decipher the text.

↳ The numbers in the deciphered text do not seem to represent individual ASCII characters, so do the numbers in the encrypted message sent by Alice.

↳ Alice encoded the message by pairing 2 characters' ASCII codes together in one block. So each block of numbers in the deciphered text correspond to two ASCII characters. She

↳ Alice used the following way:

encoded block = ASCII 1 × 256 + ASCII 2.

↳ So to get the first and second characters respectively we have:

First char = encoded block ÷ 256

Second char = encoded block / 256

↳ She then ciphered the numbers as discussed earlier.

↳ If n_{Bob} is very large it would be more difficult to find the values of P_B and q_B. It would take a lot of time to look at all the possible numbers that could correspond to P_B and q_B such that $n_{Bob} = P_B q_B$.

OAEP

↳ Alice's encoding would still be insecure because she only used 2 characters packed into a block. That makes it so that each block is at most 2^{16} plaintext values, which an attacker could encrypt with public key and see which ciphertext matches.

Ciphered text:
[42750, 225049, 67011, 9062, 263924, 83744, 10951, 156009, 174373, 125655, 207173, 200947, 227576, 183598, 148747, 211083, 225049, 218587, 191754, 164498, 225049, 171200, 193625, 99766, 94020, 223044, 38895, 74666, 48846, 219950, 139957, 77545, 171672, 165278, 150326, 262673, 164498, 142355, 77545, 171672, 255299, 5768, 264753, 75667, 261607, 31371, 164498, 140654, 244325, 140696, 40948, 179472, 168428, 34824, 32543, 30633, 104926, 190298, 148747, 132510, 42607, 232272, 42721, 188452, 239228, 50536, 216512, 139240, 78779, 166647, 100152, 261607, 121165]

Deciphered text:
[18533, 31008, 17007, 25132, 8296, 25970, 25895, 29472, 29551, 28005, 8291, 29305, 28788, 28519, 29281, 28776, 31008, 26729, 29556, 28530, 31008, 26223, 29216, 31087, 29984, 10344, 29812, 28787, 14895, 12133, 28206, 30569, 27497, 28773, 25705, 24878, 28530, 26415, 30569, 27497, 12116, 26725, 24397, 24935, 26979, 24407, 28530, 25715, 24417, 29285, 24403, 29045, 25953, 28009, 29544, 24399, 29555, 26982, 29281, 26469, 10542, 8264, 24944, 28793, 8294, 24931, 29807, 29289, 28263, 11296, 16748, 26979, 25902]

Unpacking and converting to ASCII we get:
"Hey Bob, here's some cryptography history for you (https://en.wikipedia.org/wiki/The_Magic_Words_are_Squeamish_Ossifrage). Happy factoring, Alice."