Homework 3

Jake Maschoff
U0581770

1.

a. RSA has the limitation that the message should be less than n because then you would be able to impersonate messages with different messages. An example of this could be that $message_1 = 42$ and $message_2 = 12$ but n is 30. Then for both these messages, they would have the same signature of 12. Thus you must limit your messages to be smaller than n to not allow for impersonated signatures.

b. Yes, you do have the ability to find K_{AB}. That is because of the AND statement that is taking place with the SHA512. You are able to get the key by starting with an R_A of 1. With that one bit set, Trudy can send it to Alice and then also send R_A with the bit not set. If Alice sends back different values for when the bit is set and not set, we know that this bit must be set for K_{AB}. That is because when the bit was set for R_A, it was being bitwise anded with the K_{AB} and must be set for the SHA512 to get 2 different values. If the two values are not different after changing the bit, then we know that bit is not set for K_{AB}. Then you keep moving bit by bit until the values are no longer changing or until you get the same value that was initially sent during the first transaction. An example of this could be done using $K_{AB} = 22 = 10110_2$. Trudy first sends $R_a = 1$ and $R_a = 0$. Alice sends back the same values for both these transmissions since K_{AB} & $1 = 0$ and K_{AB} & $0 = 0$, so they have the same SHA512. We know now that the LSB is a 0, and we not iterate to the next bit. So now we send $R_a = 2$ and $R_a = 0$. Since K_{AB} & $2 = 2$ and K_{AB} & $0 = 0$, we know the values are different and thus the second bit must be a 1. You keep doing this until the values stop changing or until you have the length of the key needed to impersonate Alice.

c. Since each character is being encrypted separately, there is a known small message space. For each character, you could try a-z using the substitution table, and then exponentiate it to the power of the known public key, mod it by the known n value and see if you get a match with what was sent. If you do, you know the character and continue to decrypt all the characters in the message.

d.

1d) $(x^c \bmod n)^d \bmod n = x^{cd} \bmod n$

$x \bmod n = x - nk$

$= (x^c - nk)^d \bmod n$ ← use binomial therom to expand

← distribute

$= \left(\sum_{i=0}^{d} \binom{d}{i} (x^c)^{d-i} * (-nk)^i \right) \bmod n$

= for when $i = 0$, all other i's will have multiple of n, thus $\bmod n = 0$ and those terms are irrelevant

$\binom{d}{0} x^{cd} \cdot (-nk)^0$

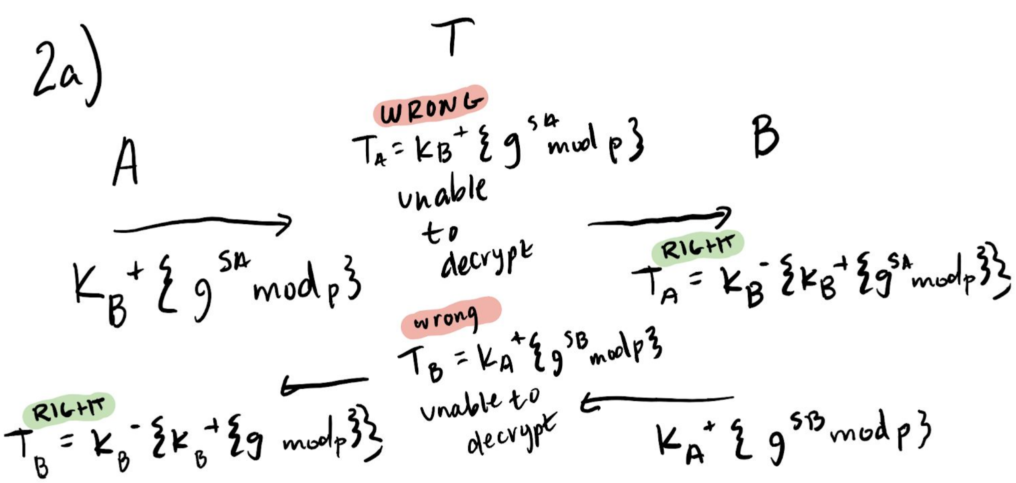
$= x^{cd} \bmod n$ ✓

2.

a.

You are unable to because Trudy doesn't have the secret keys to decrypt the DH values correctly, as seen from this diagram. However, Bob and Alice can decrypt the correct DH values.

2a)



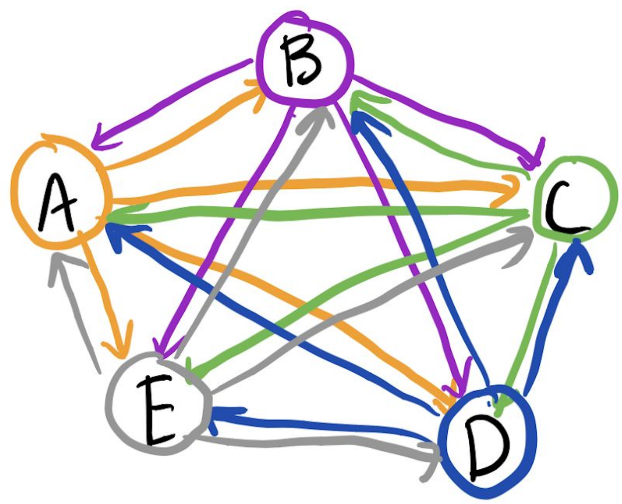
b.

Now that Alice has Bob's DH key T_b, she must send Bob her DH key T_a which is g^{Sa}modp. Once Bob has that value, they are able to share secret messages by creating the secret key. Alice creates her secret key with $K = T_b^{Sa}$ mod p. She can now send a secret message using K{m}.

c.

For every group member to have a secret key with each other member, it requires that every group member shares their key with everyone else in the group (as shown in the diagram below). That means there must be n broadcast messages sent, or (n-1) • n total messages.

2c)



3.

a.

The NP-complete problem used in this ZKP is the 3-partition problem (https://en.wikipedia.org/wiki/3-partition_problem). Alice creates these sets of numbers where there exists the triplet either equals T_a or T_b, and that there are 2 other sets that either equal T_a and T_b. Bob is then able to ask Alice to report the 3 sets of the number needed to equal either T_a or T_b. The scheme is shown below. However, it is infeasible for Trudy to actually compute the three sets for either T_a or T_b since they don't know the actual pairs located within each of the $S_{1...n}$ that add to T_a or T_b, whereas Alice does before sending.

3a)

$T_A = 30$ $T_B = 291$

A

B

$S_1 = \{121, -33, \dots, 81\}$

$S_2 = \{1, 2, \dots 3\}$

$\vdots$

$S_n = \{4, 5, \dots, 7\}$

T_A for S_1

T_A for S_2

$\vdots$

T_B for S_n

responds w/ correct triplet pairs that add up to either $T_A | T_B$ (whichever is specified)

b.

To create this into a signature, you sum every value from each set altogether. You then concatenate that value with the message being sent using an MD: $MD(m, sum(sum_{s1}, sum_{s2}, \dots, sum_{sn}))$. This results in some binary value. T_a is used for a set if its bit position is set as a 1, and T_b is used for a set if its bit position is set as a 0. You then include the triplet values for the T_a or T_b value for each of the sets in the signature depending on if the bit is set or not for that set's position. The signature then becomes:

$<m, S_1, S_2, S_3, \dots, S_n, T_a triplet(S_1), T_b triplet(S_1), T_b triplet(S_1), \dots, T_a triplet(S_1)>$

This is secure because any change in message causes a different message digest, and thus resulting in a different binary sequence and different T_a and T_b values for each set.

4.

PROGRAMMING ASSESSMENT

5.

a.

The paper suggests to get the bits for the secret from the 'spatial and temporal variations of the reciprocal wireless channel.' With is a less expensive and flexible solution to ensure that the secret bits are unique to two endpoints.

b.

There is no authentification used in either the DH scheme or this scheme. Along with that, both the DH and this scheme are vulnerable to active adversaries. If active adversaries act in the middle, they can pull off a 'man-in-the-middle' attack.

c.

This scheme does provide perfect forward secrecy. They do this via privacy amplification, use of universal hashes from a known set, and through hash-lemmas.

d.

Due to the lack of variations in a channel, the bits used can have a low entropy making these bits easily guessable and not suitable for key generation. Also, an adversary can inject known variations into the channel allowing for Alice and Bob to use a predictable key.