Homework 2

Jake Maschoff
U0581770

1.

a)

Reflection attack:

When a third party (let's say Trudy) is able to open a connection with the server (let's say Bob) and the user connecting authenticating with the user (let's say Alice) is able to authenticate themselves as Alice even though they are actually Trudy.

Man-in-the-middle attack:

When someone in the 'middle' is eavesdropping on transmissions and is able to gain access to keys by impersonating both Alice and Bob and then able to read all messages sent between Alice and Bob once they have access to each others keys.

b)

Even if you do find the key, this key has nothing to do with the pattern in the following bytes. Since each byte encrypted independently of each other it doesn't matter that there was a duplicate key. That is because each encrypted byte is the char XOR key.

c)

You should use a **block cipher** because with RC4 you are likely to have a repeating key, and once a repeated key is used, you would be able to decrypt the message due to the fact that the adversary would know the {plaintext, cipertext} of the previous message and allowing them to solve for the repeated key. With AES, though, it is very hard to figure out the key even with the message and cipher block pairs.

2.

a)

Only 2 blocks of plaintext would be lost since the plaintext block is only dependent on the previous cipher-text block and the current cipher-text block.

b)

Since XOR is commutative after the two are switched you are still able to continue decrypting and encrypting because the cipher blocks still contain all the correct terms even if

they weren't switched. That is because you only need all of the previous values in order to keep the chain going (in order or out of order) due to XOR's commutative property.

c)

If you lose a ciphertext block, that means you also lose a message block. With losing this message block you are then unable to continue encrypting and decrypting. That is because NCBC uses previous messages in its chaining algorithm. So once this happens you are out of sync and all the continued messages are no longer able to be encrypted and decrypted correctly.

d)

Even after n and n+2 ciphertext blocks are switched, you are still **in sync** after due to XOR's commutative property and because you still have received all the prior information.

e)

Even after any n permutations of ciphertext blocks are switched, you are still **in sync** after due to XOR's commutative property and because you still have received all the prior information.

3.

a)

Assuming that the likelihood of the message digest of outputting *d* for both message lengths is the same, then they would theoretically have to test the **same amount** of messages before finding a message that outputs *d*. That is because you need to examine 2^{256} messages before finding an output of d.

b)

$n(n-1)/2K = 0.8$
$n(n-1) = 1.6K$
$n^2-n-1.6K = 0$
Given the quadratic form, we can apply the quadratic formula:
$n = (1 \pm \text{sqrt}(1 + 4(1.6K))) / 2$

4.

PROGRAMMING EXERCISE

5.

a)

Replaces the TCP, and IP layer and replaces it with a data chunks layer and security layer. The data chunks layer consists of 2 types of packets, interest, and data. An interest packet is a packet used to make a request (similar to an HTTP GET request) and a data packet is used to respond to interest packets with the requested data. When forwarding interest packets, the CCN model broadcasts the interest packet across all available connections. The data packet response is then sent only to the connection that requested the data.

b)

With this architecture, only the number of data packets requested will be sent or forwarded, which incorporates flow balance. The paper even talks about even if an interest packet is traveling across multiple networks in search of Data, that after the data is found only 1 Data packet will be sent back in response. Reducing the flow of packets and thus not making DDoS attacks possible. Also since a content name doesn't directly map to a specific endpoint, you are able to distribute the load dynamically by having all the requests map to all different endpoints which contain the data. With the use of the content store, too, many other devices are able to hold the data that is being requested by the interest packets. Thus removing a large amount of load from the device which contains the original data.

c)

The paper states that publishers are able to bind names to content. That way a content provider can sign the content for a specific name, allows you to certify content for a specific user. However, a publisher isn't really able to dictate where their packets go. Once they send the packet to their connected interfaces, those interfaces are able to send it wherever they please.

d)

I do think that spam could be prevented using this CCN architecture. That is because of the trust model that is also incorporated into the CCN architecture. With the ability to have key relationships between publishers and content consumers, you could block all incoming traffic that has been deemed spam traffic. That is because they would sign their content with the domain. So if www.spam.com signs their content, you could learn that their content is spam and drop all packets that use this certificate.