EECS 565 Intro Information & Computer Security

Homework 1: Cryptography

Your name: Audrey Pino

Classic ciphers.

1. A substitution cipher replaces each letter with the one at the *i*-slots to its right. Please use the key "DAWN" to decrypt the ciphertext "vealruwgwwk". Show your decryption process briefly. Assume the letter "A" is mapped to position "0". A detailed mapping is provided as follows.

Position	0	1	2	3	4	5	6	7	8	9	10	11	12
Letter	A	B	C	D	E	F	G	H	I	J	K	L	M
Position	13	14	15	16	17	18	19	20	21	22	23	24	25
Letter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

V	E	A	L	R	U	W	G	W	W	K
21	4	0	11	17	20	22	6	22	22	10

D	A	W	N	D	A	W	N	D	A	W
3	0	22	13	3	0	22	13	3	0	22

S	E	E	Y	O	U	A	T	T	W	O
21-3=18	4-0=4	0-22=4	11-13=24	17-3=14	20-0=20	22-22=0	6-13-19	22-3=19	22-0=22	10-22=14

2. What is a polyalphabetic substitution cipher? Compared with shift ciphers, discuss two major differences between the two ciphers.

A polyalphabetic substitution cipher is a technique used for encryption where each letter of a plaintext is mapped to a different letter of the ciphertext through multiple shift values of multiple alphabets. Two major differences between a polyalphabetic substitution cipher and a shift cipher are *complexity* and *security*. A polyalphabetic substitution cipher is more complex than a shift cipher as it uses both multiple alphabets as well as different shift values to encrypt. This is different from a shift cipher that only uses a single shift value and thus a single alphabet. Furthermore, a polyalphabetic substitution cipher is more secure as the frequency of each letter is not correlated to the plain text. In a shift cipher, the frequency of each letter is equal in the plaintext and cipher text, thus being crackable with frequency analysis.

3. One-time pad is used to encrypt messages. If an attacker obtains the ciphertext and the corresponding plaintext message, can he find the encryption key? Does this mean OTP is vulnerable to the known-plaintext attacks?

If an attacker obtains the ciphertext and the corresponding plaintext message in a one-time pad (OTP) system, they cannot find the encryption key. The encryption key in a OTP is used to XOR each bit of the message, producing a series of seemingly random bits, the same length as the message. Thus, an attacker may be able to obtain the key by using the XOR method, if they know the plaintext and ciphertext, however, the attacker would only be able to use the key on that specific message, hint one-time. Furthermore, the OTP is not vulnerable to known-plaintext attacks as an attacker would not be able to derive the encryption key and they cannot use this information to decrypt any other messages that were encrypted using the same key.

4. What is frequency analysis? Please use frequency analysis to crack the below ciphertext. You can use a tool to help compute the statistics, https://www.ittc.ku.edu/~fli/565/frequency_analysis.html.

o kewixn zol yg yomn wnokvpn gt o sgvfypl ek yg xggm oy bgz wofl zofy ef ofq bgz wofl zofy gvy ygfl jxoep

Hint: O=A, G=O, X=L, W=M, Y=T

PLAINTEXT MESSAGE: a simple way to take measure of a country is to look at how many want in and how many want out tony blair
B=H, E=I, F=N, G=O, I=P, J=B, K=S, L=Y, M=K, N=E, O=A, P=R, Q=D, S=C, T=F, V=U, W=M, X=I, Y=T, Z=W

Frequency analysis is a technique to decrypt ciphertext through analyzing the frequency of letters or symbols in the ciphertext. Specific letters and combinations of those letters, prove to be more common than others, thus, frequency analysis uses these patterns to break ciphertext through educated guesses about corresponding letters in the plaintext.

Secret-key cryptography.

5. What is the block size and key length in DES encryption? Can two different keys encrypt the same plaintext into the same ciphertext? Why or why not?

The block size in DES encryption is 64 bits and the key length in DES encryption is 56 bits. It is not possible for two different keys to encrypt the same plaintext into the same ciphertext because the DES encryption process uses a series of substitutions and permutations based on the specific key being used. Thus, any small changes to the key result in significant changes to the encrypted message and therefore prevents different keys from encrypting the same plaintext into the same ciphertext.

6. What is the meet-in-the-middle attack? Please briefly explain why double-DES is vulnerable to this attack, but triple-DES is not.

The meet in the middle attack is an attack that capitalizes on the vulnerabilities of block cipher encryption through the creation of two tables: (1) encryption: precomputed table of all possible values of a plaintext for a specific key and (2) decryption: precomputed table of all possible values of a ciphertext for a specific key. These two tables then "meets-in-the-middle," matching the table entries in order to find the key that was used to encrypt the ciphertext. Double-DES is vulnerable to this attack because it uses 2 rounds of encryption, so the attacker can precompute all possible values using the first key and then using the second key, resulting in a time complexity of 2^57. While this takes longer than the original single-DES, it is possible. However, when increased again to triple-DES, this attack is prevented because it uses 3 rounds of encryption which is impractical due to the myriad of possible combinations, and time it would take to attempt to match a key to the cipher text.

7. What is the key exhaustive attack? If an attacker uses this attack to break a ciphertext encrypted by AES-192-CBC. Assume he uses a computer with 4GHz CPU to crack the keys and it takes about 100 cycles to test one key. How much time **on average** does he need to find the correct encryption key?

The key exhaustive attack (i.e. brute-force attack) is a cryptographic attack where an attacker tries every possible key until the correct one is found. To calculate the time required to perform a key exhaustive attack on AES-192-CBC encryption with a 4GHz CPU, we first calculate the possible keys and then divide it by the amount of keys the computer can test per second. The key size for AES-192 is 192 bits, which means there are 2^192 possible keys. Further, on a computer with a 4GHz CPU and taking 100 cycles to test one key, the number of keys that can be tested per second is (4 GHz / 100cycles)/key = 40 million keys/second. Then, it can be calculated that the average time required to test all possible keys would be 2^192 / 40 million keys/second = **7.6 x 10^23 seconds**.

8. Errors in one block will propagate to other blocks when the CBC mode is used in block ciphers.
    a. Suppose an error occurs during transmission. One bit of the first ciphertext block is wrong. When the receiver tries to recover the message, how many plaintext blocks cannot be decrypted correctly?

If an error occurs in the first ciphertext block in CBC mode, then both the corresponding plaintext block and the next plaintext block will be unable to be decrypted correctly. The incorrect ciphertext block will be decrypted into an incorrect plaintext block, and the incorrect plaintext block will be XORed with the initialization vector), thus producing an incorrect value for the next ciphertext block.

b. Suppose a one-bit error occurs in the first block of the plaintext message. After encrypting the message, how many ciphertext blocks will have error bits? When the receiver recovers the message, how many plaintext blocks cannot be decrypted correctly?

If an error occurs in the first block of the plaintext message, then the first ciphertext block will have error bits but the second ciphertext block will not have any error bits. When the receiver recovers the message, only the first plaintext block cannot be decrypted correctly, but the rest of the message can be recovered correctly as long as the error is corrected or the corrupted block is discarded.

Public-key cryptography

9. Use RSA to encrypt the message "EECS". Assume p = 3 and q=11, and e=7. Please show the encryption steps (assume A=1). What is the security problem with textbook RSA encryption?

- n = p*q = 3*11 = 33.
- phi(n) = (p-1)(q-1) = 210 = 20.
- e=7 is the public key exponent
    - d = e^(-1) mod phi(n),
- (7*d) mod 20 = 1.
    - Using the Extended Euclidean Algorithm: d=3.
- A = 1, B =2, …
    - E E C S: 5 5 3 19

- E = 5 = 5^7 mod 33 = 14
  E = 5 = 5^7 mod 33 = 14
  C = 3 = 3^7 mod 33 = 9
  S = 19 = 19^7 mod 33 = 13

- **ENCRYPTED MESSAGE: 14 14 9 13.**

The security problem with textbook RSA encryption is due to the fact that there is no variation in the produced ciphertext when the same key is used to encrypt the same message. Moreover, if the exponent is small and fixed, it is vulnerable to attacks that exploit the structure of the ciphertext to recover the plaintext or the private key.

10. The Diffie-Hellman key negotiation protocol is vulnerable to the man-in-the-middle attack. Please explain the attack process and the mitigation methods.

The man in the middle attack in respect to the Diffie-Hellman key negotiation protocol occurs when the attacker intercepts the message being sent and chooses their own secret integer and further modifies the shared secret key, giving the attacker the ability to decrypt and modify the messages being sent without detection. For example, if Bob and Alice are attempting to establish a shared secret key, Bob

may choose b for his secret integer while Alice chooses a. Bob then sends B = g^b mod p while Alice sends A = g^a mod p, with shared secret keys of K = A^b mod p and K = B^a mod p, respectively. The attacker however can intercept the message and modify the secret integers and shared secret keys to be C = g^c mod p with Bob and D = g^d mod p with Alice in order to decrypt and re-encrypt messages sent between Bob and Alice without detection. This type of attack can be mitigated two ways: (1) use published DH numbers and (2) authenticated DH exchange. Further, using published DH numbers such as a common base, selecting permanent public and private numbers, and publishing public numbers through a trusted channel can mitigate this attack by removing the attackers ability to change the secret keys without Bob and Alice knowing. Moreover, using an authenticated DH exchange can further mitigate this attack by establishing a secure connection (i.e. confirming Bob is communicating with Alice, vice versa) while also confirming that the message had not been modified.

11. SHA-256 is commonly used as the signing algorithm on SSL certificates. Which hash properties are desired in this use case? To successfully generate a collision (i.e., two certificates with the same signature), how many attempts **on average** should the attacker try (assume the desired collision probability is greater than 50%)?

In respect to SSL certificates, the hash properties desired in a signing algorithm like SHA-256 include preimage resistance and collision resistance. Preimage resistance removes the attacker's ability to generate a fake certificate with the same hash value as the legitimate certificate. In other words, given a randomly selected hash value x, it is hard to find any message x’ such that h(x)=h(x’). Moreover, collision resistance removes the attacker's ability to create two certificates with the same hash value; it is hard to find any two messages x ≠ x’ such that h(x)=h(x’). To successfully generate a collision in SHA-256, an attacker should try an average of 2^128 attempts, assuming the desired collision probability is greater than 50%.

12. What is HMAC? Find one use case of HMAC in real-world applications. Which hash property/properties is utilized by this application?

HMAC ("keyed-Hash Message Authentication Code") is a process that uses the equation hmac(k, M) = h(k2||h(k1||M)), where M is the message and k is the secret key, to ensure a message was not tampered with during transmission) and that it came from a trusted sender. In other words, the message was not involved in a man-in-the-middle attack. One use case of HMAC in real-world applications is web API’s. Web API’s from companies such as Spotify, Twitter, and Google use HMAC to authenticate API requests by giving the client and server both a shared secret key and further ensuring that client has an HMAC value in each request. Thus, a server can validate the request, and know it was not tampered with, if the client's HMAC value matches the server's expected HMAC value. The hash property utilized by HMAC in this application is collision resistance as it removes the attacker's ability to create two certificates with the same hash value.