FACULTY OF COMPUTER SCIENCE
CS4355: Cryptanalysis an DB Security
Professor: Dr. Rongxing Lu
Office: GE 114
Email: rlu1@unb.ca
Phone: 451-6966
Fall 2019 Tutorial Time: T 2:30-3:20

Tutorial 1

1) Briefly define essential computer and network security requirements including Accountability, Availability, Authenticity, Integrity, Confidentiality.
   - Answer:
     - Accountability: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
     - Availability: Assures that systems work promptly and service is not denied to authorized users.
     - Authenticity: The property of being genuine and being able to be verified and trusted; confidence in the validly of a transmission, a message, or a message originator.
     - Data integrity: assures that information and programs are changed only in a specified authorized manner system integrity: assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.
     - Data confidentiality: assures that private or confidential information is not made available or disclosed to unauthorized individuals; Privacy: assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.
2) Briefly define the Caesar cipher.
   - Answer: The Caesar cipher involves replacing each letter of the alphabet with the letter standing k places further down the alphabet, for k in the range 1 through 25.
3) What is the difference between passive and active security attacks?
   - Answer: Passive attacks have to do with eavesdropping on, or monitoring, transmissions. Electronic mail, file transfers, and client/server exchanges are examples of transmissions that can be monitored. Active attacks include the modification of transmitted data and attempts to gain unauthorized access to computer systems.

     Passive attacks: release of message contents and traffic analysis. Active attacks: masquerade, replay, modification of messages, and denial of service.
4) What is the Denial of Service (DoS) attack?
   - Answer: Denial of Service (DoS): prevents the normal use or management of communications facilities. DoS attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of DoS is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.
5) What is the non-repudiation?
   - Answer: Nonrepudiation: Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.
6) A generalization of the Caesar cipher, known as the affine Caesar cipher, has the following form: For each plaintext letter p, substitute the ciphertext letter C:

   $$C = E([a, b], p) = (ap + b) \bmod 26$$

   A basic requirement of any encryption algorithm is that it be one-to-one. That is, if $p \neq q$, then $E(k, p) \neq E(k, q)$. Otherwise, decryption is impossible, because more than one plaintext character maps into the same ciphertext character. The affine Caesar cipher is not one-to-one for all values of a. For example, for $a = 2$ and $b = 3$, then $E([a, b], 0) = E([a, b], 13) = 3$.
   - Are there any limitations on the value of b? Explain why or why not.
     - Answer: No. A change in the value of b shifts the relationship between plaintext letters and ciphertext letters to the left or right uniformly, so that if the mapping is one-to-one it remains one-to-one.
   - Determine which values of a are not allowed.
     - Answer: $2, 4, 6, 8, 10, 12, 13, 14, 16, 18, 20, 22, 24$. Any value of a larger than 25 is equivalent to $a \bmod 26$.

- Provide a general statement of which values of a are and are not allowed. Justify your statement.
  - Answer: The values of a and 26 must have no common positive integer factor other than 1. This is equivalent to saying that a and 26 are relatively prime, or that the greatest common divisor of a and 26 is 1. To see this, first note that $E(a, p) = E(a, q)$ $(0 \leq p \leq q < 26)$ if and only if $a(p - q)$ is divisible by 26.
    1. Suppose that a and 26 are relatively prime. Then, $a(p - q)$ is not divisible by 26, because there is no way to reduce the fraction $a/26$, and $(p - q)$ is less than 26.
    2. Suppose that a and 26 have a common factor $k > 1$. Then $E(a, p) = E(a, q)$, if $q = p + m/k \neq p$ where $m = 26$.

7) A ciphertext has been generated with an affine cipher. The most frequent letter of the ciphertext is “B”, and the second most frequent letter of the ciphertext is “U”. Break this code.
   - Answer: Assume that the most frequent plaintext letter is e and the second most frequent letter is t. Note that the numerical values are $e = 4; B = 1; t = 19; U = 20$. Then we have the following equations:

$$1 = (4a + b) \bmod 26, \qquad 20 = (19a + b) \bmod 26$$

   Thus, $19 = 15a \bmod 26$. By trial and error, we solve: $a = 3$. Then $1 = (12 + b) \bmod 26$. By observation, $b = 15$.

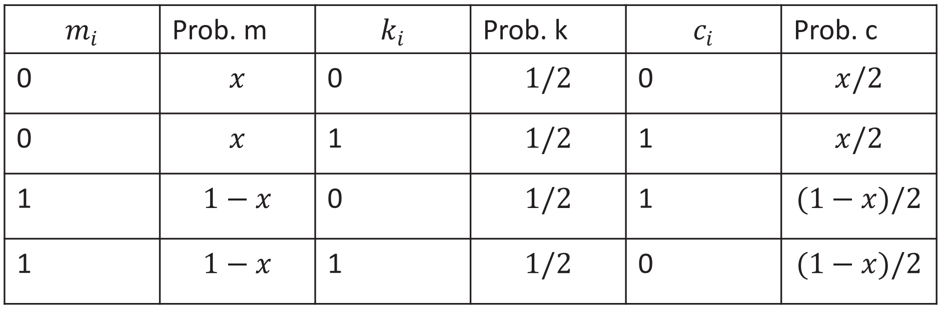
8) Using the Vigenre cipher, encrypt the word “explanation” using the key “leg”.
   - Answer:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plain text	e	x	p	l	a	n	a	t	i	o	n
key	l	e	g	l	e	g	l	e	g	l	e
Cipher text	p	b	v	w	e	t	l	x	o	z	r

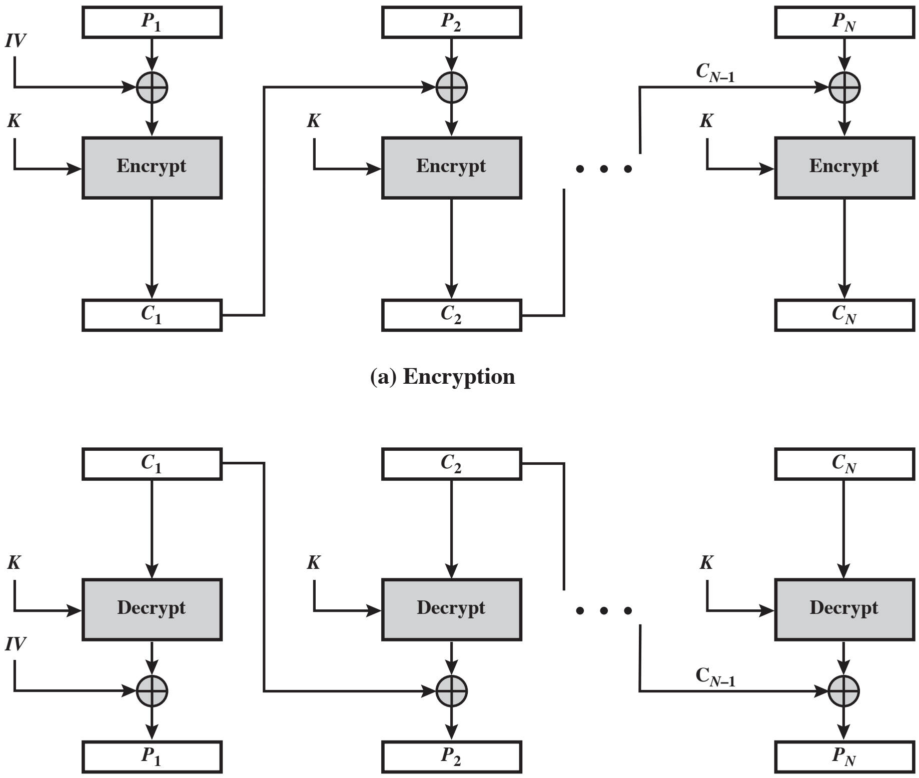
Tutorial 2

1) What is the difference between a block cipher and a stream cipher?
   - Answer: A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
2) What is the difference between diffusion and confusion?
   - Answer: In diffusion, the statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext. This is achieved by having each plaintext digit affect the value of many ciphertext digits, which is equivalent to saying that each ciphertext digit is affected by many plaintext digits. Confusion seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key. Thus, even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key. This is achieved by the use of a complex substitution algorithm.
3) Explain the avalanche effect.
   - Answer: The avalanche effect is a property of any encryption algorithm such that a small change in either the plaintext or the key produces a significant change in the ciphertext.
4) Prove One-Time Padding is unconditional secure.
   - Answer: The security depends on the randomness of the key, but it is hard to define randomness. In cryptographic context, we seek two fundamental properties in a binary random key sequence: **Unpredictability:** Independent of the number of the bits of a sequence observed, the probability of guessing the next bit is not better than 1/2. Therefore, the probability of a certain bit being 1 or 0 is exactly equal to 1/2. **Balanced (Equal Distribution):** The number of 1 and 0 should be equal.



m_i	Prob. m	k_i	Prob. k	c_i	Prob. c
0	x	0	$1/2$	0	$x/2$
0	x	1	$1/2$	1	$x/2$
1	$1-x$	0	$1/2$	1	$(1-x)/2$
1	$1-x$	1	$1/2$	0	$(1-x)/2$

   The probability of a key bit being 1 or 0 is exactly equal to $1/2$; The plaintext bits are not balanced. Let the probability of 0 be x and then the probability of 1 turns out to be $1-x$; We can calculate the probability of ciphertext bits. We find out the probability of a ciphertext bit being 1 or 0 is equal to $1/2 \cdot x + 1/2 \cdot (1-x) = 1/2$, and the ciphertext looks like a random sequence.
5) With the ECB mode, if there is an error in a block of the transmitted ciphertext, only the corresponding plaintext block is affected. However, in the CBC mode, this error propagates. For example, an error in the transmitted C_1 obviously corrupts P_1 and P_2.
   - Are any blocks beyond P_2 affected?
     - Answer: No. For example, suppose C_1 is corrupted. The output block P_3 depends only on the input blocks C_2 and C_3.
   - Suppose that there is a bit error in the source version of P_1. Through how many ciphertext blocks is this error propagated? What is the effect at the receiver?



(a) Encryption

(b) Decryption

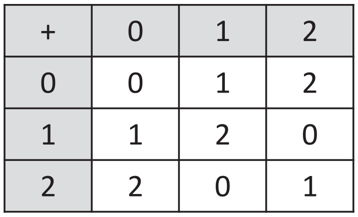
- Answer: An error in P_1 affects C_1. But since C_1 is input to the calculation of C_2, C_2 is affected. This effect carries through indefinitely, so that all ciphertext blocks are affected. However, at the receiving end, the decryption algorithm restores the correct plaintext for blocks except the one in error. You can show this by writing out the equations for the decryption. Therefore, the error only effects the corresponding decrypted plaintext block.

6) Is it possible to perform encryption operations in parallel on multiple blocks of plaintext in CBC mode? How about decryption?
   - Answer: In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In CBC decryption, however, the input blocks for the inverse cipher function (i.e., the ciphertext blocks) are immediately available, so that multiple inverse cipher operations can be performed in parallel.

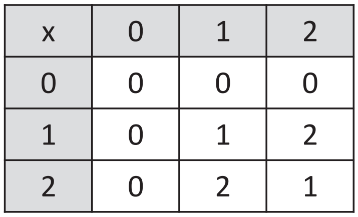
7) CBC-Pad is a block cipher mode of operation used in the RC5 block cipher, but it could be used in any block cipher. CBC-Pad handles plaintext of any length. The ciphertext is longer then the plaintext by at most the size of a single block. Padding is used to assure that the plaintext input is a multiple of the block length. It is assumed that the original plaintext is an integer number of bytes. This plaintext is padded at the end by from 1 to bb bytes, where bb equals the block size in bytes. The pad bytes are all the same and set to a byte that represents the number of bytes of padding. For example, if there are 8 bytes of padding, each byte has the bit pattern 00001000. Why not allow zero bytes of padding? That is, if the original plaintext is an integer multiple of the block size, why not refrain from padding?
   - Answer: After decryption, the last byte of the last block is used to determine the amount of padding that must be stripped off. Therefore there must be at least one byte of padding.

Tutorial 3

1) Does the set of residue classes (mod 3) form a group
   - with respect to modular addition?
     - Answer: Here are the addition and multiplication tables



+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1



x	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Yes. The identity element is 0, and the inverses of 0, 1, 2 are respectively 0, 2, 1.

   - with respect to modular multiplication?
     - No. The identity element is 1, but 0 has no inverse.

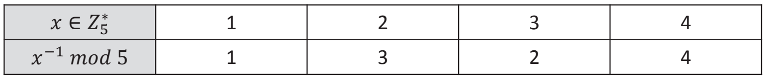
2) Consider the set $S = \{a, b\}$ with addition and multiplication defined by the following tables. Is S a ring? Justify your answer.

+	a	b
a	a	b
b	b	a

$\times$	a	b
a	a	a
b	a	b

   - Answer: S is a ring. We show it by using the axioms
     - (A1) Closure: The sum of any two elements in S is also in S.
     - (A2) Associative: S is associative under addition, by observation.
     - (A3) Identity element: a is the additive identity element for addition.
     - (A4) Inverse element: The additive inverses of a and b are a and b, respectively.
     - (A5) Commutative: S is commutative under addition, by observation.
     - (M1) Closure: The product of any two elements in S is also in S.
     - (M2) Associative: S is associative under multiplication, by observation.
     - (M3) Distributive laws: S is distributive with respect to the two operations, by observation.

3) Find the multiplicative inverse of each nonzero element in Z_5.
   - Answer:



$x \in Z_5^*$	1	2	3	4
x^{-1} *mod* 5	1	3	2	4

4) Let p be a prime number, and $2^m \not\equiv 1 \bmod p$. Please prove

$$1^m + 2^m + \cdots + (p-1)^m \equiv 0 \bmod p$$

   - Answer: Let $A = \{A_1 = 1, A_2 = 2, \cdots, A_{p-1} = p-1\}$ be one set. We can construct another set B, where $B = \{B_1 = 2 \cdot 1 \bmod p, B_2 = 2 \cdot 2 \bmod p, \cdots, B_{p-1} = 2 \cdot (p-1) \bmod p\}$. Since $\gcd(2, p) = 1$, we can see $|A| = |B|$ and two sets A and B are identical. Therefore, we have

$$\sum_{i=1}^{p-1} A_i^m = \sum_{i=1}^{p-1} B_i^m \Rightarrow \sum_{i=1}^{p-1} A_i^m \equiv \sum_{i=1}^{p-1} B_i^m \bmod p$$

Then, we have

$$1^m + 2^m + \cdots + (p-1)^m \equiv (2 \cdot 1)^m + (2 \cdot 2)^m + \cdots + (2 \cdot (p-1))^m \bmod p$$

$$1^m + 2^m + \cdots + (p-1)^m \equiv 2^m \cdot (1^m + 2^m + \cdots + (p-1)^m) \bmod p$$

$$(2^m - 1) \cdot (1^m + 2^m + \cdots + (p-1)^m) \equiv 0 \bmod p$$

Because $2^m \not\equiv 1 \bmod p$, we have

$$1^m + 2^m + \cdots + (p-1)^m \equiv 0 \bmod p$$

5) Let $p > 3$ be a prime number. Please prove that, for any integers a, b, we will have

$$ab^p - ba^p \equiv 0 \bmod 6p$$

- Answer:

  $$ab^p - ba^p \equiv 0 \bmod 6p \Rightarrow ab^p - ab - (ba^p - ab) \equiv 0 \bmod 6p$$

  We first prove $ab^p - ab \equiv 0 \bmod 6p$, and then prove $ba^p - ab \equiv 0 \bmod 6p$.
  For $ab^p - ab \equiv 0 \bmod 6p$, we actually need to prove $6p|(ab^p - ab)$.
  Because $b^p - b = b(b^{p-1} - 1) = b[(b^2)^{\frac{p-1}{2}} - 1] = b(b^2 - 1)[(b^2)^{\frac{p-1}{2}-1} + (b^2)^{\frac{p-1}{2}-2} \cdots + 1]$, we know

  $$b(b^2 - 1)|b^p - b$$

  It is easy to see

  $$6|b(b^2 - 1)$$

  (we can see $b(b^2 - 1)$ always has a factor 2 and a factor 3, so we have $6|b(b^2 - 1)$.) Therefore, we have

  $$6|b^p - b$$

  From the Fermat Little Theorem, we have

  $$p|b^p - b$$

  Because $\gcd(6, p) = 1$, we have

  $$6p|b^p - b$$

  Then, $6p|(ab^p - ab)$. Similarly, we can prove $6p|(ba^p - ab)$.
  Finally, we have $ab^p - ab - (ba^p - ab) \equiv 0 \bmod 6p$, that is, $ab^p - ba^p \equiv 0 \bmod 6p$.

Tutorial 4

1) What are the principal elements of a public-key cryptosystem?
   - Answer: *Plaintext:* This is the readable message or data that is fed into the algorithm as input. *Encryption algorithm:* The encryption algorithm performs various transformations on the plaintext. *Public and private keys:* This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input. *Ciphertext:* This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts. *Decryption algorithm:* This algorithm accepts the ciphertext and the matching key and produces the original plaintext.
2) What are the roles of the public and private key?
   - Answer: A user's private key is kept private and known only to the user. The user's public key is made available to others to use. The private key can be used to encrypt a signature that can be verified by anyone with the public key. Or the public key can be used to encrypt information that can only be decrypted by the possessor of the private key.
3) In a public-key system using RSA, you intercept the ciphertext $C = 10$ sent to a user whose public key is $e = 5$, $n = 35$. What is the plaintext M?
   - Answer: $M = 5$
4) In the RSA public-key encryption scheme, each user has a public key, e, and a private key, d. Suppose Bob leaks his private key. Rather than generating a new modulus, he decides to generate a new public and a new private key. Is this safe?
   - Answer: No, it is not safe. Once Bob leaks his private key, Alice can use this to factor his modulus, N. Then Alice can crack any message that Bob sends.

     Here is one way to factor the modulus:

     Let $k = ed - 1$. Then k is congruent to $0 \bmod \phi(N)$ (where 'ϕ' is the Euler totient function). Select a random x in the multiplicative group Z_N^*. Then $x^k \equiv 1 \bmod N$, which implies that $x^{k/2}$ is a square root of $1 \bmod N$. With 50% probability, this is a nontrivial square root of N, so that

     $$\gcd(x^{k/2} - 1, N)$$

     will yield a prime factor of N.

     If $x^{k/2} = 1 \bmod N$, then try $x^{k/2}, x^{k/4}$, etc...

     This will fail if and only if $x^{k/2i} = -1 \bmod N$ for some i. If it fails, then choose a new x.

     This will factor N in expected polynomial time.
5) "I want to tell you, Holmes," Dr. Watson's voice was enthusiastic, "that your recent activities in network security have increased my interest in cryptography. And just yesterday I found a way to make one-time pad encryption practical."

   "Oh, really?" Holmes' face lost its sleepy look.

   "Yes, Holmes. The idea is quite simple. For a given one-way function F, I generate a long pseudorandom sequence of elements by applying F to some standard sequence of arguments. The cryptanalyst is assumed to know F and the general nature of the sequence, which may be as simple as $S, S+1, S+2, \cdots$, but not secret S. And due to the one-way nature of F, no one is able to extract S given $F(S+i)$ for some i, thus even if he somehow obtains a certain segment of the sequence, he will not be able to determine the rest."

   "I am afraid, Watson, that your proposal isn't without flaws and at least it needs some additional conditions to be satisfied by F. Let's consider, for instance, the RSA encryption function, that is $F(M) = M^K \bmod N$, K is secret. This function is believed to be one-way, but I wouldn't recommend its use, for example, on the sequence $M = 2, 3, 4, 5, 6, \cdots$

   'But why, Holmes?" Dr. Watson apparently didn't understand. "Why do you think that the resulting sequence $2^K \bmod N$, $3^K \bmod N$, $4^K \bmod N$, $\cdots$ is not appropriate for one-time pad encryption if K is kept secret?"

   "Because it is at least partially predictable, dear Watson, even if K is kept secret. You have said that the cryptanalyst is assumed to know F and the general nature of the sequence. Now let's assume that he will obtain somehow a short segment of the output sequence. In crypto circles, this assumption is generally considered to be a viable one. And for this output sequence, knowledge of just the first two elements will allow him to predict quite a lot of the next elements of the sequence, even if not all of them, thus this sequence can't be considered to be cryptographically strong. And with the knowledge of a longer segment he could predict even more of the next elements of the sequence. Look, knowing the general nature of the sequence and its first two elements $2^K \bmod N$ and $3^K \bmod N$, you can easily compute its following elements."

   Show how this can be done.
   - Answer: 3rd element, because it equals to the 1st squared, 5th element, because it equals to the product of 1st and 2nd, 7th element, because it equals to the cube of 1st, etc.

Tutorial 5

1) The example used by Sun-Tsu to illustrate the Chinese Remainder Theorem (CRT) was

$$\begin{cases} x & \equiv & 2 \bmod 3 \\ x & \equiv & 3 \bmod 5 \\ x & \equiv & 2 \bmod 7 \end{cases}$$

Solve for x.

- Answer: Let $m_1 = 3, m_2 = 5, m_3 = 7$. $a_1 = 2, a_2 = 3, a_3 = 2$. We have $M = m_1 \cdot m_2 \cdot m_3 = 3 \times 5 \times 7 = 105$, $M_1 = M/m_1 = 35$, $M_2 = M/m_2 = 21$, $M_3 = M/m_3 = 15$.
  $\alpha_1 = M_1^{-1} \bmod m_1 = 35^{-1} \bmod 3 = 2$, $\alpha_2 = M_2^{-1} \bmod m_2 = 21^{-1} \bmod 5 = 1$, $\alpha_3 = M_3^{-1} \bmod m_3 = 15^{-1} \bmod 7 = 1$
  Therefore,

  $$x = a_1 \cdot \alpha_1 \cdot M_1 + a_2 \cdot \alpha_2 \cdot M_2 + a_3 \cdot \alpha_3 \cdot M_3 \bmod M = 2 \times 2 \times 35 + 3 \times 1 \times 21 + 2 \times 1 \times 15 \bmod M = 23$$

2) Consider a Diffie-Hellman scheme with a common prime $q = 11$ and a primitive root $\alpha = 2$.
- Show that 2 is a primitive root of 11.
  - Answer: $\phi(11) = 10$, $2^{10} = 1024 = 1 \bmod 11$. If you check 2^n for $n < 10$, you will find that none of the values is $1 \bmod 11$.
- If user A has public key $Y_A = 9$, what is A's private key X_A?
  - Answer: 6, because $2^6 \bmod 11 = 9$.
- If user B has public key $Y_B = 3$, what is the secret key K shared with A?
  - Answer: $K = 3^6 \bmod 11 = 3$

3) Consider an ElGamal encryption scheme with a common prime $q = 11$ and a primitive root $\alpha = 2$. If B has public key $Y_B = 3$ and A choose the random integer $k = 2$, what is the ciphertext of $M = 8$?
- Answer: $(4, 6)$. Because $C_1 = \alpha^k = 2^2 = 4 \bmod 11 = 4$, $C_2 = M \cdot Y_B^k = 8 \times 3^2 = 6 \bmod 11 = 6$

4) The lecture note describes a man-in-the-middle attack on the Diffie-Hellman key exchange protocol in which the adversary generates two public-private key pairs for the attack. Could the same attack be accomplished with one pair? Explain.
- Answer:
  - Darth prepares for the attack by generating a random private key X_D and then computing the corresponding public key Y_D.
  - Alice transmits Y_A to Bob.
  - Darth intercepts Y_A and transmits Y_D to Bob. Darth also calculates $K_2 = (Y_A)^{X_D} \bmod q$.
  - Bob receives Y_D and calculates $K_1 = (Y_D)^{X_B} \bmod q$.
  - Bob transmits X_A to Alice.
  - Darth intercepts X_A and transmits Y_D to Alice. Darth calculates $K_1 = (Y_B)^{X_D} \bmod q$.
  - Alice receives Y_D and calculates $K_2 = (Y_D)^{X_A} \bmod q$.

5) What are the negatives of the following elliptic curve points over Z_{17}? $P = (5, 8)$, $Q = (3, 0)$, $R = (0, 6)$.
- Answer: The negative of a point $P = (x_P, y_P)$ is the point $-P = (x_P, -y_P \bmod p)$. Thus $-P = (5, 9)$; $-Q = (3, 0)$; $-R = (0, 11)$.

6) Consider the elliptic curve $E_{11}(1, 6)$, that is, the curve is defined by $y^2 = x^3 + x + 6$ with a modulus of $p = 11$. For some point $G = (2, 7)$. Compute the multiples of G from $2G$ through $4G$.
- Answer: We follow the rules of addition described in lecture notes, $2G = (2, 7) + (2, 7)$, we first compute

  $$\lambda = \frac{3 \times 2^2 + 1}{2 \times 7} \bmod 11 = \frac{13}{14} \bmod 11 = 2/3 \bmod 11 = 8$$

  Then,

  $$x_3 = 8^2 - 2 - 2 \bmod 11 = 5, \quad y_3 = 8(2 - 5) - 7 \bmod 11 = 2 \qquad \Rightarrow 2G = (5, 2)$$

  Similarly, $3G = 2G + G = (8, 3)$, $4G = 3G + G = (10, 2)$.

Tutorial 6

1) Let us consider using an encryption algorithm to construct a one-way hash function. Consider using RSA with a known key. Then process a message consisting of a sequence of blocks as follows: Encrypt the first block, XOR the result with the second block and encrypt again, etc. Show that this scheme is not secure by solving the following problem. Given a two-block message B_1, B_2, and its hash $RSAH(B_1, B_2) = RSA(RSA(B_1) \oplus B_2)$.
   Given an arbitrary block C_1, choose C_2 so that $RSAH(C_1, C_2) = RSAH(B_1, B_2)$. Thus, the hash function does not satisfy weak collision resistance.
   - Answer: The opponent has the two-block message B_1, B_2 and its hash $RSAH(B_1, B_2)$. The following attack will work. Choose an arbitrary C_1 and choose C_2 such that:

     $$C_2 = RSA(C_1) \oplus RSA(B_1) \oplus B_2$$

     Then,

     $$RSA(C_1) \oplus C_2 = RSA(C_1) \oplus RSA(C_1) \oplus RSA(B_1) \oplus B_2 = RSA(B_1) \oplus B_2$$

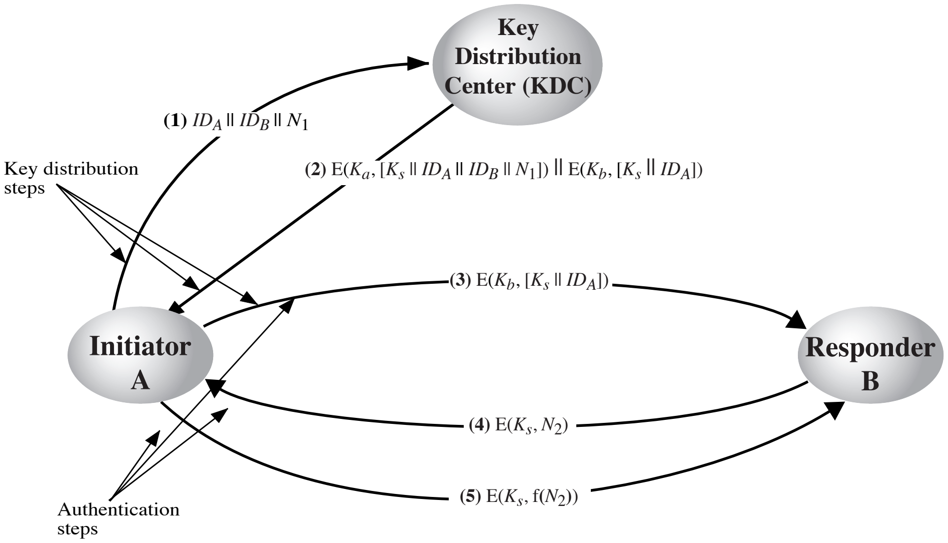
     So

     $$RSAH(C_1, C_2) = RSA(RSA(C_1) \oplus C_2) = RSA(RSA(B_1) \oplus B_2) = RSAH(B_1, B_2)$$

2) It is tempting to try to develop a variation on Diffie-Hellman that could be used as a digital signature. Here is one that is simpler than DSA and that does not require a secret random number in addition to the private key.
   Public elements: q prime number, α, $\alpha < q$ and α is a primitive root of q
   Private key: X, $X < q$
   Public key: $Y = \alpha^X \bmod q$
   To sign a message M, compute $h = H(M)$, which is the hash code of the message. We require that $\gcd(h, q-1) = 1$. If not, append the hash to the message and calculate a new hash. Continue this process until a hash code is produced that is relatively prime to $(q-1)$. Then calculate Z to satisfy $Z \times h \equiv X (\text{mod} q - 1)$. The signature of the message is α^Z. To verify the signature, a user verifies that $Y = (\alpha^Z)^h = \alpha^X \bmod q$.
   - Show that this scheme works. That is, show that the verification process produces an equality if the signature is valid.
     - Answer: To verify the signature, the user verifies that $(g^Z)^h = g^X \bmod q$.
   - Show that the scheme is unacceptable by describing a simple technique for forging a user's signature on an arbitrary message.
     - Answer: To forge the signature of a message, we first find its hash h. Then we calculate Z to satisfy $Z \cdot h = 1 \bmod (q-1)$. Now $g^{Zh} = g$, so $g^{XZh} = g^X \bmod q$. Hence (h, g^{XY}) is a valid signature and the opponent can calculate g^{XY} as $(g^X)^Y$.

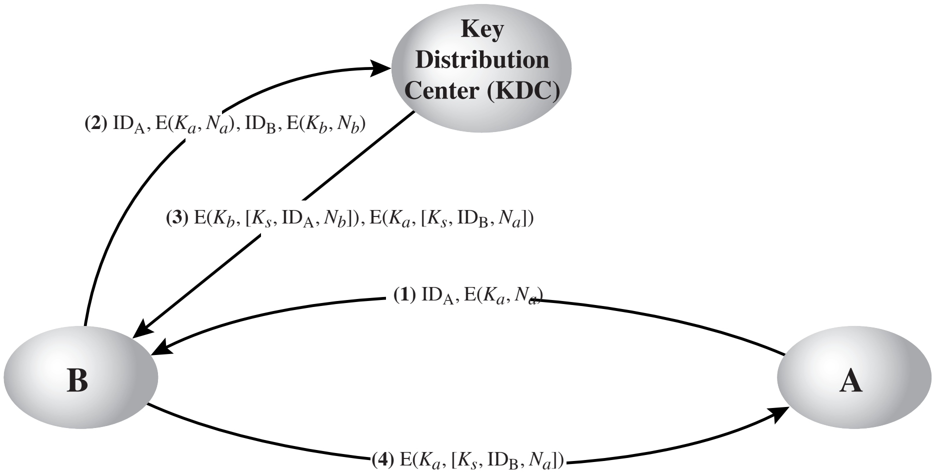
Tutorial 7

1) What is the difference between a session key and a master key?
   - Answer: A session key is a temporary encryption key used between two principals. A master key is a long-lasting key that is used between a key distribution center and a principal for the purpose of encoding the transmission of session keys. Typically, the master keys are distributed by non-cryptographic means.
2) What is a nonce?
   - Answer: A nonce is a value that is used only once, such as a timestamp, a counter, or a random number; the minimum requirement is that it differs with each transaction.
3) List four general categories of schemes for the distribution of public keys.
   - Answer:
     – public announcement: users distribute public keys to recipients or broadcast to community at large
     – publicly available directory: can obtain greater security by registering keys with a public directory
     – public-key authority: improve security by tightening control over distribution of keys from directory
     – public-key certificates: certificates allow key exchange without real-time access to public-key authority
4) One local area network vendor provides a key distribution facility, as illustrated in the figure. Describe the scheme.



Key Distribution Center (KDC)
(1) $ID_A \| ID_B \| N_1$
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$
Key distribution steps
(3) $E(K_b, [K_s \| ID_A])$
Initiator A
Responder B
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps

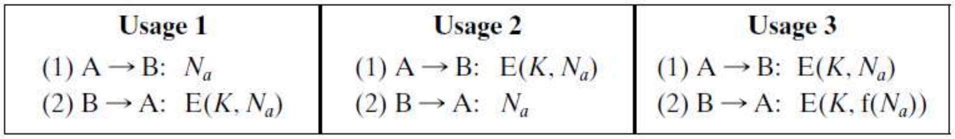
   - Answer:
     a) Before A wants to connect with B, A first send ID_A, ID_B, and a nonce N_1 to the KDC.
     b) KDC returns $E(K_a, [K_s||ID_A||ID_B||N_1])$ and $E(K_b, [K_s||ID_A])$ to A. (K_A, K_B are master keys, K_s is a session key.)
     c) A forwards $E(K_b, [K_s||ID_A])$ to B.
     d) B uses $E(K_s, N_2)$ to authenticate the session key K_s.
     e) A returns $E(K_s, f(N_2))$ for authentication.
5) Describe the scheme in the following figure, and compare the scheme to that in the previous figure, what are the pros and cons?


Key Distribution Center (KDC)
(2) ID_A, E(K_a, N_a), ID_B, E(K_b, N_b)
(3) E(K_b, [K_s, ID_A, N_b]), E(K_a, [K_s, ID_B, N_a])
(1) ID_A, E(K_a, N_a)
B
A
(4) E(K_a, [K_s, ID_B, N_a])

- Answer: A sends a connection request to B, with an event marker or nonce (Na) encrypted with the key that A shares with the KDC. If B is prepared to accept the connection, it sends a request to the KDC for a session key, including A's encrypted nonce plus a nonce generated by B (Nb) and encrypted with the key that B shares with the KDC. The KDC returns two encrypted blocks to B. One block is intended for B and includes the session key, A's identifier, and B's nonce. A similar block is prepared for A and passed from the KDC to B and then to A. A and B have now securely obtained the session key and, because of the nonces, are assured that the other is authentic.
  The proposed scheme appears to provide the same degree of security as that of in previous figure. One advantage of the proposed scheme is that the, in the event that B rejects a connection, the overhead of an interaction with the KDC is avoided.

Tutorial 8

1) There are three typical ways to use nonces as challenges in user authentication. Suppose N_a is a nonce generated by A, A and B share key K, and $f()$ is a function (such as an increment). The three usages are



Usage 1	Usage 2	Usage 3
(1) A → B: N_a	(1) A → B: $E(K, N_a)$	(1) A → B: $E(K, N_a)$
(2) B → A: $E(K, N_a)$	(2) B → A: N_a	(2) B → A: $E(K, f(N_a))$

Describe situations for which each usage is appropriate.

- Answer: All three really serve the same purpose. The difference is in the vulnerability. In Usage 1, an attacker could breach security by inflating N_a and withholding an answer from B for future replay attack, a form of suppress-replay attack. The attacker could attempt to predict a plausible reply in Usage 2, but this will not succeed if the nonces are random. In both Usage 1 and 2, the messages work in either direction. That is, if N is sent in either direction, the response is $E[K, N]$. In Usage 3, the message is encrypted in both directions; the purpose of function f is to assure that messages 1 and 2 are not identical. Thus, Usage 3 is more secure.

2) In addition to providing a standard for public-key certificate formats, X.509 specifies an authentication protocol. The original version of X.509 contains a security flaw. The essence of the protocol is as follows.

$$\begin{aligned} A \to B &: \quad A\{t_A, r_A, ID_B\} \\ B \to A &: \quad B\{t_B, r_B, ID_A, r_A\} \\ A \to B &: \quad A\{r_B\} \end{aligned}$$

where t_A and t_B are timestamps, r_A and r_B are nonces and the notation $X\{Y\}$ indicates that the message Y is transmitted, encrypted, and signed by X.

The text of X.509 states that checking timestamps t_A and t_B is optional for three-way authentication. But consider the following example: Suppose A and B have used the preceding protocol on some previous occasion, and that opponent C has intercepted the preceding three messages. In addition, suppose that timestamps are not used and are all set to 0. Finally, suppose C wishes to impersonate A to B. C initially sends the first captured message to B:

$$C \to B : A\{0, r_A, ID_B\}$$

B responds, thinking it is talking to A but is actually talking to C:

$$B \to C : B\{0, r'_B, ID_A, r_A\}$$

C meanwhile causes A to initiate authentication with C by some means. As a result, A sends C the following:

$$A \to C : A\{0, r'_A, ID_C\}$$

C responds to A using the same nonce provided to C by B:

$$C \to A : C\{0, r'_B, ID_A, r'_A\}$$

A responds with

$$A \to C : A\{r'_B\}$$

This is exactly what C needs to convince B that it is talking to A, so C now repeats the incoming message back out to B.

$$C \to B : A\{r'_B\}$$

So B will believe it is talking to A whereas it is actually talking to C. Suggest a simple solution to this problem that does not involve the use of timestamps.

- Answer: The problem has a simple fix, namely the inclusion of the name of B in the signed information for the third message, so that the third message now reads:

$$A \to B : \qquad A\{r_B, B\}$$

3) Consider a one-way authentication technique based on asymmetric encryption:

case 1:

$$\begin{aligned} A \rightarrow B &: ID_A \\ B \rightarrow A &: R_1 \\ A \rightarrow B &: E(PR_a, R_1) \end{aligned}$$

case 2:

$$\begin{aligned} A \rightarrow B &: ID_A \\ B \rightarrow A &: E(PU_a, R_2) \\ A \rightarrow B &: R_2 \end{aligned}$$

where R_1, R_2 are random numbers. PU_a, PR_a are the public key and private key of A.

- Explain the protocol in each case.
  - Answer:
    - ∗ This is a means of authenticating A to B. R_1 serves as a challenge, and only A is able to encrypt R_1 so that it can be decrypted with A's public key.
    - ∗ This is a means of authenticating A to B. Only A can decrypt the second message, to recover R_2.
- In each case, what type of attack is the protocol susceptible to?
  - Answer:
    - ∗ Someone (e.g., C) can use this mechanism to get A to sign a message. Then, C will present this signature to D along with the message, claiming it was sent by A. This is a problem if A uses its public/private key for both authentication, signatures, etc.
    - ∗ Someone (e.g. C) can use this mechanism to get A to decrypt a message (i.e., send that message as R_2) that it has eavesdropped from the network (originally sent to A).

Tutorial 9

1) Provide a brief definition of network access control.
   - Answer: Network access control (NAC) is an umbrella term for managing access to a network. NAC authenticates users logging into the network and determines what data they can access and actions they can perform. NAC also examines the health of the user's computer or mobile device (the endpoints).
2) What is an EAP?
   - Answer: The Extensible Authentication Protocol (EAP) acts as a framework for network access and authentication protocols. EAP provides a set of protocol messages that can encapsulate various authentication methods to be used between a client and an authentication server. EAP can operate over a variety of network and link level facilities, including point-to-point links, LANs, and other networks, and can accommodate the authentication needs of the various links and networks.
3) List and briefly define four EAP authentication methods.
   - Answer:
     - EAP-TLS (EAP-Transport Layer Security): EAP-TLS (RFC 5216) defines how the TLS protocol can be encapsulated in EAP messages.
     - EAP-TTLS (EAP-Tunneled TLS) is similar to EAP-TLS except only the server has a certificate to authenticate itself to the client first.
     - EAP-GPSK (EAP Generalized Pre-Shared Key) is an EAP method for mutual authentication and session key derivation using a Pre-Shared Key (PSK).
     - EAP-GPSK specifies an EAP method based on pre-shared keys and employs secret key-based cryptographic algorithms. EAP-IKEv2 supports mutual authentication and session key establishment using a variety of methods.
4) Define cloud computing.
   - Answer: NIST defines cloud computing as follows: A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.
5) Describe some of the main cloud-specific security threats.
   - Answer:
     - *Abuse and nefarious use of cloud computing:* For many cloud providers (CPs), it is relatively easy to register and begin using cloud services, some even offering free limited trial periods. This enables attackers to get inside the cloud to conduct various attacks, such as spamming, malicious code attacks, and denial of service.
     - *Insecure interfaces and APIs:* CPs expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. The security and availability of general cloud services is dependent upon the security of these basic APIs.
     - *Malicious insiders:* Under the cloud computing paradigm, an organization relinquishes direct control over many aspects of security and, in doing so, confers an unprecedented level of trust onto the CP. One grave concern is the risk of malicious insider activity. Cloud architectures necessitate certain roles that are extremely high-risk. Examples include CP system administrators and managed security service providers.
     - *Shared technology issues:* IaaS vendors deliver their services in a scalable way by sharing infrastructure. Often, the underlying components that make up this infrastructure (CPU caches, GPUs, etc.) were not designed to offer strong isolation properties for a multi-tenant architecture.
     - *Data loss or leakage:* For many clients, the most devastating impact from a security breach is the loss or leakage of data.
     - *Account or service hijacking:* With stolen credentials, attackers can often access critical areas of deployed cloud computing services, allowing them to compromise the confidentiality, integrity, and availability of those services.
     - *Unknown risk profile:* In using cloud infrastructures, the client necessarily cedes control to the cloud provider on a number of issues that may affect security.

Tutorial 10

1) Why does PGP generate a signature before applying compression?

   - Answer: It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification. If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required. b. Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and, as a result, produce different compressed forms. However, these different compression algorithms are interoperable because any version of the algorithm can correctly decompress the output of any other version. Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm.

2) Why is R64 conversion useful for an e-mail application?

   - Answer: R64 converts a raw 8-bit binary stream to a stream of printable ASCII characters. Each group of three octets of binary data is mapped into four ASCII characters.

3) What is the basic difference between X.509 and PGP in terms of key hierarchies and key trust?

   - Answer: We trust this owner, but that does not necessarily mean that we can trust that we are in possession of that owner's public key.

4) Consider radix-64 conversion as a form of encryption. In this case, there is no key. But suppose that an opponent knew only that some form of substitution algorithm was being used to encrypt English text and did not guess that it was R64. How effective would this algorithm be against cryptanalysis?

   - Answer: It certainly provides more security than a monoalphabetic substitution. Because we are treating the plaintext as a string of bits and encrypting 6 bits at a time, we are not encrypting individual characters. Therefore, the frequency information is lost, or at least significantly obscured.

CS4355/6355 Theory Assignment 1 --- Solutions

Question 1.

Part a (Question). List and briefly define categories of passive and active security attacks.

Part a (Answer).

Active attack is about modification of data stream or creation of false stream and therefore it attempts to alter or change the system resources or affects their operations.

- **Masquerade** happens when an entity pretends to be a different entity.
- **Modification of a message** occurs when portion of a legitimate message is altered, or that messages reordered or delayed.
- **Denial of service** is about preventing the normal use and availability of machine or service.
- **Replay** happens when an entity copies a message and replays it later.

Passive attack is about eavesdropping on or monitoring of the traffic and transmitted data. It attempts to gather information from the system but does not affect system resources or operations.

- **Release of message contents**
- **Traffic analysis** occurs when an opponent try to determine the frequency and length of messages.

Part b (Question). List and briefly define the basic security requirements in computer and network security.

Part b (Answer).

The security requirements fall in two main categories:

- **Functional requirements**
- **Assurance requirements**

By the way, essential computer and network security requirements can be enumerated as bellow:

- **Accountability:** the traceability of actions performed on a system to a specific system entity (user, process, and device).
- **Availability:** Assures that systems work promptly and service is not denied to authorized users.
- **Authenticity:** The property of being genuine and being able to be verified and trusted.
- **Integrity (Data and System Integrity):** Assures that information are changed only in a specific and authorized manner and the system performs its intended function in an unimpaired (strong and stable) manner.
- **Confidentiality (Data confidentiality and privacy):** assure that private or confidential information is not made available or disclosed to unauthorized individuals.

And network security services can be listed as follows:

- **Authentication**: is concerned with assuring that a communication is authentic.
- **Access Control**: is the ability to limit and control the access to host systems and applications via communications links.
- **Data Confidentiality**: is the protection of transmitted data from passive attacks.
- **Data Integrity**: is the protection of transmitted data from active attacks.
- **Non-Repudiation**: is the prevention of either sender or receiver denying a transmitted message.
- **Availability Service**: is the property of a system that being accessible and usable upon demand by an authorized system entity.

Part c (Question). Describe the Kerckhoffs's Principles.

Part c (Answer).

Based on Kerckhoffs's principles, in designing a cryptosystem we need to assume that the opponents know it in detail and details of the system could be revealed without being worry, except the key.

1. The system must be practically, if not mathematically, indecipherable;
2. It should not require secrecy, and it should not be a problem if it falls into enemy hands;
3. It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will;
4. It must be applicable to telegraph communications;
5. It must be portable, and should not require several persons to handle or operate;

6. Lastly, given the circumstances in which it is to be used, the system must be easy to use and should not be stressful to use or require its users to know and comply with a long list of rules.

Part d (Question). Describe the functions of confusion and diffusion in symmetric ciphers.

Part d (Answer).

- **Confusion**: Process of substituting characters or symbols to make the relationship between ciphertext and key as complex as possible.
- **Diffusion**: Process of spreading effect of plaintext or key as widely as possible over ciphertext and a little change in input stream or key causes a big change in output.

Part e (Question). Describe the Strict Avalanche Conditions in symmetric ciphers.

Part e (Answer).

Each m*n S-Box is a basic component in Symmetric-key algorithms and transforms the input bits (m bit) into output bits (n bit) by an implemented lookup table and the substitution algorithm should have good avalanche properties.

Strict Avalanche Criterion (SAC): States that any output bit j of and S-Box should change with probability ½ when any single input bit I in inverted for all i, j.

Bit Independence Criterion (BIC): States that output bits j and k should change independently when any single input bit I in inverted for all i, j, and k.

Part f. (Question). Describe the key management problem in conventional cryptosystems.

Part f (Answer).

A cryptosystem has at least five important entities: 1) Plaintext, 2) Secret Key, 3) Ciphertext, 4) Encryption Algorithm and 5) Decryption Algorithm. Critical part of it is successful key management and distribution in both symmetric and asymmetric cryptosystems. In a symmetric key algorithm the keys involved are identical for both encrypting and decrypting a message and in an asymmetric key algorithm, in contrast are two distinct keys that are mathematically related to each other. Key management typically consists of three steps for carefully dealing with the generation, exchange, storage, use, crypto-shredding (destruction) and replacement of keys securely:

- **Key Exchange**: prior to making a secure communication, key must be exchanged between communication entities (sender and receiver)
- **Key Storage**: keys must be saved and stored securely.
- **Key Lifetime**: period of the time that a key is to be used and frequency of key replacement.

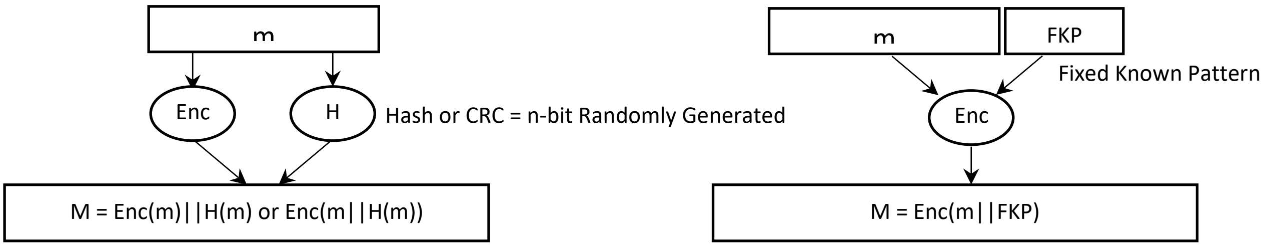
Question 2.

A fundamental cryptographic principle states that all messages must have redundancy. But we also know that redundancy helps an intruder tell if a guessed key is correct. Consider two forms of redundancy. First, the initial n bits of the plaintext contain a known pattern. Second, the final n bits of the message contain a hash over the message. From a security point of view, are these two equivalent? Discuss your answer.

Answer.

There are two fundamental cryptographic principles:

- **Redundancy** (Message must contain some redundancy to prevent intruders from sending garbage message and tricking the receiver).
- **Freshness** (Some method is needed to foil replay attacks) are fundamental cryptographic principles.


m
Enc
H
Hash or CRC = n-bit Randomly Generated
M = Enc(m)||H(m) or Enc(m||H(m))
m
FKP
Fixed Known Pattern
Enc
M = Enc(m||FKP)

In this question two different approaches have been proposed to satisfy redundancy. If we do not have any detail of the cryptosystem and try only by generating different permutation based on the given plaintext both of them are hard to recover plaintext m or guess the key, meaning they are equivalent.
In the second approach intruder faces with two different algorithms, encryption/decryption algorithm and hash algorithm. Thus, it will take more time.

Question 3.

Suppose that a message has been encrypted using DES in ciphertext block chaining mode. One bit of ciphertext in block C i is accidentally transformed from a 0 to a 1 during transmission. How much plaintext will be garbled as a result?

Answer.

Consider 5 ciphertext C_1, C_2, … and C_5. If we have had a problem in C2 only P2 and P3 would be garbled.
$P_1 = Dec_K(C_1)$ xor IV ;C0=Initialization Vector (IV).
$P_2 = Dec_K(C_2)$ xor C_1 ;C_2, with transmission error, will affect P_2.
$P_3 = Dec_K(C_3)$ xor C_2 ;C_2, with transmission error, will affect P_3.
$P_4 = Dec_K(C_4)$ xor C_3
$P_5 = Dec_K(C_5)$ xor C_4
Therefore, existing error in C_k will only affect P_k and P_{k+1}.

Question 4.

The following is a ciphertext with Caesar Cipher, please analyze it, and give the corresponding plaintext and the used key.
DRO MSDI LBSWC GSDR CEWWOB'C NOVSQRDC, GSDR MYVYBPEV ZBYNEMO SX DRO WKBUOD CDKXNC KXN RKGKSSKX WECSM CZSVVSXQ YXDY LOKMROC.

Answer.

We can create different substitution of alphabet letters by shifting 0 to 25.
For example first word of ciphertext (**DRO**) have a 26 different status by n-shifting algorithm.
0DRO,1ESP,2FTQ,3GUR,4HVS,5IWT,6JXU,7KYV,8LZW,9MAX,10NBY,11OCZ,12PDA,13QEB,14RFC,15SGD,**16THE**,17UIF,18VJG,19WKH,20XLI,21YMJ,22ZNK,23AOL,24BPM,25CQN
This Caesar cipher has been made by shift 26-16+1=11. (A→K, B→L, C→M, D→N, E→O, F→P . . .)
Or: Based on the relative frequency, e is the most popular, we can assume o→e, then use the relation to check others.
Plaintext: **THE CITY BRIMS WITH SUMMER'S DELIGHTS, WITH COLORFUL PRODUCE IN THE MARKET STANDS AND HAWAIIAN MUSIC SPILLING ONTO BEACHES.**

Question 5.

Please complete the following two tables, and describe why Z_{11} and Z_{11}^* are abelian groups.

Answer.

x + y		x										
mod 11		0	1	2	3	4	5	6	7	8	9	10
y	0	0	1	2	3	4	5	6	7	8	9	10
	1	1	2	3	4	5	6	7	8	9	10	0
	2	2	3	4	5	6	7	8	9	10	0	1
	3	3	4	5	6	7	8	9	10	0	1	2
	4	4	5	6	7	8	9	10	0	1	2	3
	5	5	6	7	8	9	10	0	1	2	3	4
	6	6	7	8	9	10	0	1	2	3	4	5
	7	7	8	9	10	0	1	2	3	4	5	6
	8	8	9	10	0	1	2	3	4	5	6	7
	9	9	10	0	1	2	3	4	5	6	7	8
	10	10	0	1	2	3	4	5	6	7	8	9

- **Closure:**
result of operation (+ mod 11) is in set Z_{11}.
- **Associativity:**
a+(b+c) mod 11 = (a+b)+c mod 11
- **Existence of identity:**
e+a=a+e=a mod 11 for each a in Z_{11}, and e=0.
- **Existence of inverse:**
a+a'=a'+a=0. For each a in Z_{11}; a'=-a mod 11=11-a mod 11.
- **Commutativity:**
a+b mod 11=b+a mod 11.

x * y mod 11		x									
		1	2	3	4	5	6	7	8	9	10
y	1	1	2	3	4	5	6	7	8	9	10
	2	2	4	6	8	10	1	3	5	7	9
	3	3	6	9	1	4	7	10	2	5	8
	4	4	8	1	5	9	2	6	10	3	7
	5	5	10	4	9	3	8	2	7	1	6
	6	6	1	7	2	8	3	9	4	10	5
	7	7	3	10	6	2	9	5	1	8	4
	8	8	5	2	10	7	4	1	9	6	3
	9	9	7	5	3	1	10	8	6	4	2
	10	10	9	8	7	6	5	4	3	2	1

- **Closure:**
result of operation (* mod 11) is in set Z_{11}^*.
- **Associativity:**
a*(b*c) mod 11 = (a*b)*c mod 11
- **Existence of identity:**
e*a=a*e=a mod 11 for each a in Z_{11}^*, and e=1.
- **Existence of inverse:**
For each a in Z_{11}^*; a has an inverse which is shadowed in table.
- **Commutativity:**
a*b mod 11=b*a mod 11.

Question 6.

Prove the following:

(a) [(a mod n) + (b mod n)] mod n = (a + b) mod n

(b) [(a mod n) × (b mod n)] mod n = (a × b) mod n

Answer.

Part a)

Let assume, a mod n = p → a = nk_1 + p & b mod n = q → b = nk_2 + q

Left side: [(a mod n) + (b mod n)] mod n = p + q mod n

Right side: a + b mod n = [(nk_1 + p) + (nk_2 + q)] mod n = [n(k_1 + k_2) + p + q] mod n = p + q mod n

Part b)

Let assume, a mod n = p → a = nk_1 + p & b mod n = q → b = nk_2 + q

Left side: [(a mod n) * (b mod n)] mod n = p * q mod n

Right side: a * b mod n = [(nk_1 + p) * (nk_2 + q)] mod n = [n(k_1k_2+k_1q+k_2p) + p * q] mod n = p * q mod

Question 7.

Prove the following:

a) Prove the One-time padding is provably secure.

b) Prove the Fermat's Little Theorem $a^{p-1} \equiv 1$ mod p, where p is prime and gcd(a, p) = 1.

c) Prove that there are infinitely many primes.

Answer.

Part a)

The probability of plaintext bits are not equal, P(bitP=0) = x and P(bitP=1) = 1-x

The probability of a key bit being 0 or 1 is equal, P(bitK=0)=p(bitK=1) = ½.

Probability of chipertext can be calculated as bellow.

Plaintext		Key		Ciphertext (XOR operation)	
p values	P(p)	k values	P(k)	c values	P(c)
0	x	0	½	0	x*½
0	x	1	½	1	x*½
1	1-x	0	½	1	(1-x)*½
1	1-x	1	½	0	(1-x)*½
P(c=0 or c=1) = 0*(x*½)+1*(x*½)+0*((1-x)*½)+0*((1-x)*½) = x*½ +(1-x)*½ = ½					

Part b)

Let assume A={1, 2, 3, …, p-1} and B={a*1, a*2, 3*a, …, a*(p-1)}, we need to proof |A|=|B| or there is not redundant element in B, so the p-1 multiples of a in B are distinct and nonzero.

By contradiction, i≠j → a*i = a*j mod p

If a*i=a*j mod p → a*(i-j) mod p = 0 → a mod p=0 or i-j mod p=0.

As we know that gcd(a,p)=1, therefore a mod p≠0 and i-j mod p=0 → i=j.

Now we know that A and B have a same number of elements, and try to calculate the multiplication of element in A and B.

∏(a_i in A) = ∏(b_i in B) mod p

∏(a_i in A) = 1*2*3*…*(p-1) mod p

∏(b_i in B) = (a*1)*(a*2)*(a*3)*…*(a*(p-1)) mod p = a^{p-1}*(p-1) mod p; Assume that 1*2*3*…*(p-1) = α.

→ α mod p = a^{p-1}*α mod p → (a^{p-1}-1)*α=0 mod p → a^{p-1} = 1 mod p.

Part c)

Assume that the primes are finite, and we can list them as L={p_1, p_2, p_3, …, p_r}

Let p be any common multiple of these primes plus one, i.e., P=1+p_1*p_2*p_3*..p_r. Then, P is either a prime or not.

If P is a prime, then P is a new prime that was not in L and therefore we cannot say L is finite.

If P is not prime, then P is devisable by some prime call α.

α|P and as we assume that L is infinite α is in L and therefor αlp_1*p_2*p_3*..p_r.

α|1+p_1*p_2*p_3*..p_r and α|p_1*p_2*p_3*..p_r and we know that if a|m and a|n then a|m-n →

α|(1+p_1*p_2*p_3*..p_r) – (p_1*p_2*p_3*..p_r) → α|1 and it is impossible. So α cannot divide P and therefore P is a new prime that was not in L.

Question 8.

Using the extended Euclidean algorithm, find the multiplicative inverse of

a) 1234 mod 4321

b) 550 mod 1769

Answer.

Part a)

Dividend	Divisor	Quotient	Reminder
4321	1234	3	619
1234	619	1	615
619	615	1	4
615	4	153	3
4	3	1	1
3	1	3	0

1) 1 = 1
2) 1 = 4 – (3 * 1) = 4 – 3 * 1
3) 1 = 4 – (615 - 4 * 153) * 1 = (4 * 154) – 615
4) 1 = (619 – 615 * 1) * 154 – 615 = (619 * 154) – (615 * 155)
5) 1 = (619 * 154) - (1234 – 619 * 1) * 155 = (619 * 309) – (155 * 1234)
6) 1= ((4321 – 1234 * 3) * 309) – (155 * 1234) = (4321 * 3090) – (1234 * 1082)

➔ 4321 * (309) + 1234 * (-1082) = 1 mod 4321

➔ 1234 * (-1082) = 1 mod 4321

➔ -1082 mod 4321 = 4321 – 1082 = 3239 → $(1234)^{-1}$ mod 4321 = **3239**

Part b)

Dividend	Divisor	Quotient	Reminder
1769	550	3	119
550	119	4	74
119	74	1	45
74	45	1	29
45	29	1	16
29	16	1	13
16	13	1	3
13	3	4	1
3	1	3	0

1) 1 = 1
2) 1 = 13 – (3 * 4) = 13 – (16 – 13 * 1) * 4 = (13 * 5) – (16 * 4)
3) 1 = (29 – 16 * 1) * 5 – (16 * 4) = (29 * 5) – (16 * 9)
4) 1 = (29 * 5) – ((45 – 29 * 1) * 9) = (29 * 14) – (45 * 9)
5) 1 = ((74 – 45 * 1) * 14) – (45 * 9) = (74 * 14) – (23 * 45)
6) 1 = (74 * 14) – (23 * (119 – 74 * 1)) = (37 * 74) – (119 * 23)
7) 1 = (37 * (550 – 119 * 4)) – (119 * 23) = (37 * 550) – (171 * 119)
8) 1= (37 * 550) – (171 * (1769 – 550 * 3)) = (550 * 550) – (171 * 1769)

➔ 1769 * (-171) + 550 * (550) = 1 mod 1769

➔ 550 * (550) = 1 mod 1769

➔ +550 mod 1769 = 550 → $(550)^{-1}$ mod 1769 = **550**

Question 9.

Suppose Alice and Bob shared the common modulus n=p*q=35263, but have different public-private key pairs (e_1=17, d1) and (e_2=23, d2). If David wants to send a message M to Alice and Bob, he first computes the cipher text C1=M^{e1} mod n for Alice, the value of C1 is 28657, and also compute the cipher text C_2=M^{e2} mod n for Bob, the value of C_2 is 22520. Finally, David send (C_1, C_2) to Alice and Bob, respectively. Now, suppose a passive adversary A eavesdrops the cipher-texts (C_1, C_2). Can the adversary A recover message M just from (C_1, C_2) and then public keys (n, e_1, e_2)? If the adversary A can. Please show what strategy that the adversary A would apply, and give the value of message M as well.

Answer.

As per you mentioned in the class, this situation is one where the RSA could be unsecure, and the attacker could be able to recover the message M.

We know $C_1 = (m)^{e_1}\ mod\ n$ and$C_2 = (m)^{e_2}\ mod\ n$; by assuming $e_1u + e_2v = 1$, try to calculate$C_1{}^u * C_2{}^v = ((m)^{e_1})^u * ((m)^{e_2})^v = m^{e_1u+e_2v} = m\ mod\ n$. Hence, we only need to solve $e_1u + e_2v = 1$, by applying Extended Euclidian algorithm.

$$Sk_1 = d1\ and\ Pk_1 = \ (e1, n) = (17, 35263) \rightarrow\ C_1 = 28657 = \ (m)^{17}\ mod\ 35263$$

$$Sk_2 = d2\ and\ Pk_2 = (e2, n) = (23, 35263) \rightarrow\ C_2 = 22520 = \ (m)^{23}\ mod\ 35263$$

$$\mathrm{C}_1{}^u * \mathrm{C}_2{}^v = ((m^{17})^u) * ((m^{23})^v) = m^{17u+23v}\ mod\ 35263 \xrightarrow{17u+23v=1} = m$$

$$17u + 23v = 1 \xrightarrow{Extended\ Euclidian\ Alg.} 17(-4) + 23(3) = -68 + 69 = 1 \rightarrow u = -4\ and\ v = 3$$

$$C_1{}^{-4} * C_2{}^3 = (C_1{}^{-1})^4 * \left(C_2{}^3\right) = (28657^{-1})^4 * (22520)^3\ mod\ 35263 \xrightarrow{(28657)^{-1} mod\ 35263=34884}$$

$$34884^4 * 22520^3 mod\ 35263 = \ 168 \rightarrow m = 168$$

CS4355/6355 Theory Assignment 1 --- Solutions

Question 1.

Use the Chinses Reminder Theorem (CRT) to solve x, where

$$\begin{cases} x \equiv 1 \quad mod\ 2 \\ x \equiv 1 \quad mod\ 3 \\ x \equiv 6 \quad mod\ 7 \end{cases}$$

Answer.

$a_1 = 1 \quad a_2 = 1 \quad a_3 = 6$

$m_1 = 2 \quad m_2 = 3 \quad m_3 = 7$ and $M = m_1 * m_2 * m_3 = 2*3*7 = 42$

$M_1 = M/m_1 = (m_1*m_2*m_3)/m_1 = m_2*m_3 = 3*7 = 21$

$M_2 = M/m_2 = (m_1*m_2*m_3)/m_2 = m_1*m_3 = 2*7 = 14$

$M_3 = M/m_3 = (m_1*m_2*m_3)/m_3 = m_1*m_2 = 2*3 = 6$

$$x = \left(\sum_{i=1}^{3} a_i * M_i * \left({M_i}^{-1}\ mod\ m_i\right)\right) mod\ M$$

$$= \left(\left(a_1 * M_1 * ({M_1}^{-1} mod\ m_1)\right) + \left(a_2 * M_2 * ({M_2}^{-1} mod\ m_2)\right) + \left(a_3 * M_3 * ({M_3}^{-1} mod\ m_3)\right)\right) mod\ M$$

$$= \left((1 * 21 * (21^{-1} mod\ 2)) + (1 * 14 * (14^{-1} mod\ 3)) + (6 * 6 * (6^{-1} mod\ 7))\right) mod\ M$$

- Modular Multiplicative Inverse 21 mod 2 = 21^{-1} mod 2 = 1
- Modular Multiplicative Inverse 14 mod 3 = 14^{-1} mod 3 = 2
- Modular Multiplicative Inverse 6 mod 7 = 6^{-1} mod 7 = 6

$$= \left((1 * 21 * 1) + (1 * 14 * 2) + (6 * 6 * 6)\right) mod\ 42 = 21 + 28 + 216\ mod\ 42 = 13$$

Question 2.

Consider an ElGamal encryption scheme with a common prime P = 71 and a primitive root α = 7.

a) If B has public key $Y_B = 3$ and A choose the random integer r = 2, what is the ciphertext of M = 30?

b) If A now chooses a different value of r so that the encoding of M = 30 is C = (59, C_2), what is the integer C_2 ?

Answer.

Public Parameters = (P, α)

P is a large prime number

α is a primitive root (generator) of Z_P^*

A
Public Key (PK_A) = $\alpha^{XA} = Y_A$ Private Key (SK_A) = X_A
A has Y_B

B
Public Key (PK_B) = $\alpha^{XB} = Y_B$ Private Key (SK_B) = X_B
B has Y_A

If A wants to send m to B (A → B)
A chooses a random 0 < r < P-1 and computes
C = (C_1, C_2) = (α^r mod P, m * $(Y_B)^r$ mod P)

If B wants to send m to A (A ← B)
B chooses a random 0 < r < P-1 and computes
D = (D_1, D_2) = (α^r mod P, m * $(Y_A)^r$ mod P)

If A wants to recover D
m = (D_2 / (D_1^{XA})) mod P = (D_2 * ($(D_1)^{-1}$ mod P $)^{XA}$) mod P

If B wants to recover C
m = (C_2 / (C_1^{XB})) mod P = (C_2 * ($(C_1)^{-1}$ mod P $)^{XB}$) mod P

Part a)

Public parameters (P, α) = (71, 7) and A has $Y_B = 3$. For m = 20 and random value r = 2

$C = (C_1, C_2) = (\alpha^r \bmod P\ ,\ m * (Y_B)^r \bmod P)$

$C_1 = \alpha^r \bmod P = 7^2 \bmod 71 = 49$

$C_2 = m * (Y_B)^r \bmod P = 30 * (3)^2 \bmod 71 = 270 \bmod 71 = 57$

$C = (C_1, C_2) = (49, 57)$

Part b)

$C = (C_1, C_2) = (\alpha^r \bmod P, m * (Y_B)^r \bmod P) = (59, ?)$

Since the possible value of r changes between 1 and P-1, for small prime values, it would be possible to calculate r (discrete logarithm).

$1 < r < 70$, $C_1 = 7^r \bmod 71 = 59$ → **r = 3**

$C_2 = m * (Y_B)^r \bmod P = 30 * (3)^3 \bmod 71$ → **$C_2 = 29$**

r = 3 and C_2 =29

Question 3.

Secure Hash Algorithm (SHA) is one kind of popular hash function, where SHA-256, SHA-384, and SHA-512 algorithms can produce the hash values with 256, 384, and 512 bits in length, respectively. Please explain why we usually say SHA-256, SHA-384, and SHA-512 algorithms are designed to match the security of AES with 128, 192, and 256 bits, respectively?

Answer.

Birth day attack (square root attacks) can be used to find collisions in hash functions. Consequently, if the output of the hash function is not sufficiently large, then the possibility of collision would be increased. For example, an n-bit hash function H could be able to provide $N = 2^n$ different possible outputs and there is a good chance of having two similar hash value in a list of length$\sqrt{N} = \sqrt{2^n} = 2^{\frac{n}{2}}$.

Therefore,

- in SHA-256 (n=256 and $N=2^{256}$ →Equivalent security algorithm is AES-128; $\sqrt{2^{256}} = 2^{\frac{256}{2}} = 2^{128}$
- in SHA-384 (n=384 and $N=2^{384}$ →Equivalent security algorithm is AES-192; $\sqrt{2^{384}} = 2^{\frac{384}{2}} = 2^{192}$
- in SHA-256 (n=512 and $N=2^{512}$ →Equivalent security algorithm is AES-256; $\sqrt{2^{512}} = 2^{\frac{512}{2}} = 2^{256}$

Question 4.

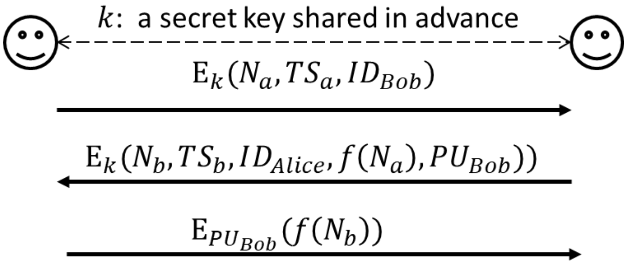
Suppose that Bob and Alice already share a secret key, but Alice still wants Bob's public key. Is there now a way to get it securely? If so, how?

Answer.

One of the possible solutions, as shown in the following figure, is:

1. Alice encrypts a generated random value (nounce), Na, and Timestamp TSa, ID_Bob with the shared key and sends it to Bob.
2. Bob decrypts the incoming message and checks its validity, then calculates F(Na) and encrypts the F(Na), Timestamp TSb, a new nounce Nb, and his public key PU_Bob, and sends the result to Alice.
3. Alice decrypt the message, checks its validity and then sends back the encrypted message, including F(Nb), with Bob's public key and send it to Bob.
4. Bob verifies the message F(Nb)'s validity by decrypting the new incoming message with his private key.

- *f() is a common shared function between Alice and Bob, e.g., F(x)=x+1.*
- *Na, Nb are nounces, random numbers*
- *TSa, TSb are the timestamps.*
- *PU_Bob is a public key of Bob.*
- *E provides an Encryption Function.*
- *k is a shared secret key.*

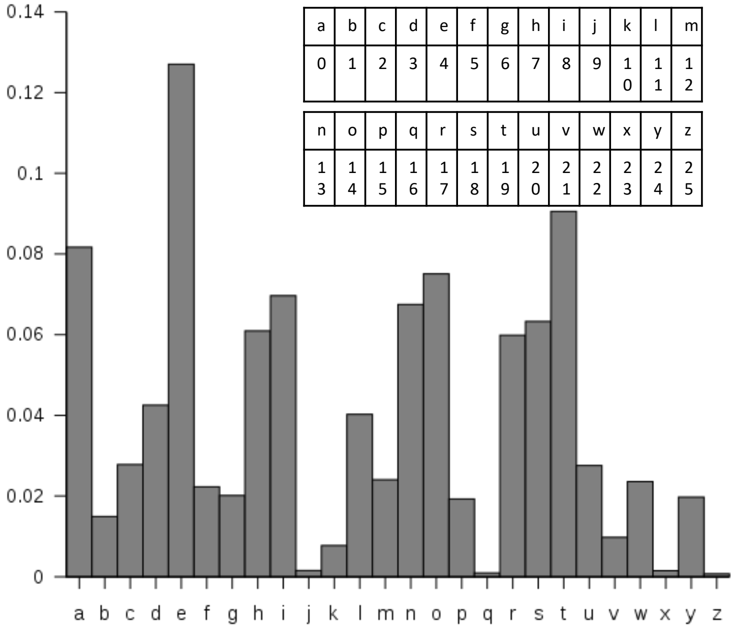

k: a secret key shared in advance
$E_k(N_a, TS_a, ID_{Bob})$
$E_k(N_b, TS_b, ID_{Alice}, f(N_a), PU_{Bob}))$
$E_{PU_{Bob}}(f(N_b))$

University of New Brunswick
Faculty of Computer Science

CS4355/6355 Cryptanalysis and Database Security

October 29th, 2019; Time Allowed: 80 miniutes

Instructions
This paper contains 6 questions and comprises 2 pages.
Answer ALL questions.
This is a closed-book examination, a calculator (not a smart phone) is allowed.
The marking scheme is shown in the left margin and [100] constitutes full marks.
The relative frequencies of letters in text and their encoded-numbers are shown in the following figure, which may be needed in this examination.



a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

[**30**] 1. **Multiple choice questions:** read each question carefully and choose the correct answer: A, B, C or D. Make sure you only choose one answer for each question, and write it down on the answer booklet.

[5] (1) Which of the following symmetric encryption algorithms does not satisfy the property of "Diffusion" ________.

A. DES B. AES

C. Triple-DES D. One-Time Padding

[5] (2) When a message "hello" is encrypted into a ciphertext by using Caesar cipher, which of the following messages is impossible to be the corresponding ciphertext of "hello" ________.

A. "jgnnq" B. "olssv" C. "miqqt" D. "khoor"

[5] (3) Assume the public key of one RSA cryptosystem is $(n = 77, e = 13)$, which of the following numbers is the corresponding private key d ________.

A. 6 B. 37 C. 23 D. 19

[5] (4) Assume a is one element in group Z_{31}^*, which of the following numbers is impossible to be the order of a ________ .

A. 5 B. 6 C. 8 D. 10

[5] (5) Suppose that a message has been encrypted using AES in ciphertext block chaining (CBC) mode. One bit of ciphertext in block C_i is accidentally transformed from a 0 to a 1 during transmission. How much plaintext will be garbled as a result?

A. 1 block B. 2 blocks C. 3 blocks D. all blocks

[5] (6) Which of the following is the correct result of computing 5^{31^3} mod 31 ________ .

A. 1 B. 5 C. 25 D. 125

[10] 2. A ciphertext has been generated with an affine cipher. The most frequent letter of the ciphertext is "b", and the second most frequent letter of the ciphertext is "u". Break this code.

[10] 3. Is it possible to perform encryption operations in parallel on multiple blocks of plaintext in CBC mode? How about decryption?

[20] 4. Prove the following:

[10] (a) $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$

[10] (b) $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

[10] 5. Let n be a positive odd integer. Please prove that $3|(2^n + 1)$.

[20] 6. In the RSA public-key encryption scheme, each user has a public key, e, and a private key, d. Suppose Alice leaks her private key. Rather than generating a new modulus, she decides to generate a new public key and a new private key. Is this safe?

END OF PAPER

Solutions.

1. D, C, B, C, B, B

2. Answer: Assume that the most frequent plaintext letter is e and the second most frequent letter is t. Note that the numerical values are $e = 4; B = 1; t = 19; U = 20$. Then we have the following equations:

$$1 = (4a + b) \bmod 26, \qquad 20 = (19a + b) \bmod 26$$

Thus, $19 = 15a \bmod 26$. By trial and error, we solve: $a = 3$. Then $1 = (12 + b) \bmod 26$. By observation, $b = 15$.

3. Answer: In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In CBC decryption, however, the input blocks for the inverse cipher function (i.e., the ciphertext blocks) are immediately available, so that multiple inverse cipher operations can be performed in parallel.

4. Since $a' = \underline{a \bmod n} < n$, $b' = \underline{b \bmod n} < n$, we have $a = a' + k_1 n$, $b = b' + k_2 n$,

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a' + b') \bmod n$$

$$(a + b) \bmod n = (a' + k_1 n + b' + k_2 n) \bmod n = (a' + b') \bmod n$$

Therefore, $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$.

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a' \times b') \bmod n$$

$$(a \times b) \bmod n = [(a' + k_1 n) \times (b' + k_2 n)] \bmod n = (a' \times b' + \Delta \cdot n) \bmod n = (a' \times b') \bmod n$$

where $\Delta = a'k_2 + b'k_1 + k_1 k_2 n$. Therefore, $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$.

5. From $2 + 1 \equiv 0 \bmod 3$, we have $2 \equiv -1 \bmod 3$ and $2^n \equiv (-1)^n \bmod 3$. Because n is odd, we have $2^n + 1 \equiv 0 \bmod 3$, which means $3 | (2^n + 1)$.

6. No, it is not safe. Once Alice leaks her private key, Bob can use this to factor the modulus, N. Then Bob can crack any message that Alice sends.

Here is one way to factor the modulus:

First, given $x^2 \equiv 1 \bmod N$, we know we have four solutions, namely $x_1 = 1$, $x_2 = -1$, $x_3 = 1 + k_3 p$, $x_4 = 1 + k_4 q$. Then, if we have x_3, then $\gcd(x_3 - 1, N) = p$. Therefore,

Let $k = ed - 1$. Then k is congruent to $0 \bmod \phi(N)$ (where 'ϕ' is the Euler totient function). Select a random x in the multiplicative group Z_N^*. Then $x^k \equiv 1 \bmod N$, which implies that $x^{k/2}$ is a square root of $1 \bmod N$. With 50% probability, this is a nontrivial square root of N, so that

$$\gcd(x^{k/2} - 1, N)$$

will yield a prime factor of N.

If $x^{k/2} = 1 \bmod N$, then try $x^{k/2}, x^{k/4}$, etc...

This will fail if and only if $x^{k/2i} = -1 \bmod N$ for some i. If it fails, then choose a new x.

This will factor N in an expected polynomial time.