Question 1 (1 mark)

What is a one-way function?

Suggested answer:

A **one-way function** is a function that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy whereas the calculation of the inverse is infeasible:

$Y = f(X)$ easy
$X = f^{-1}(Y)$ infeasible

Question 2 (1 mark)

What is message authentication code (MAC), and how is it differing from a digital signature scheme?

Suggested answer:
A message authentication code (MAC) is an authentication tag known as a checksum that is generated by applying an authentication scheme, together with a secret key, to a message. This tag is then appended to the original message. Unlike digital signatures, MACs are computed and verified with the same key, so that they can only be verified by the intended recipient. As for a digital signature scheme, the digital signature can be verified with the public key of the party that signed the message.

Question 3 (1 mark)

If a bit error occurs in the transmission of a ciphertext character in 4-bit CFB mode, how far does the error propagate?

Suggested answer:

The error will affect 5 plaintext characters. The plaintext character corresponding to the ciphertext character is obviously altered. In addition, the altered ciphertext character enters the shift register and is not removed until next four characters are processed.

Question 4 (2 marks)

The Lehman's primality test can determine if a number is a prime number or a composite number. Describe Lehman's test algorithm.

Suggested answer:

Lehman's test is a primality test. It determines whether a given integer is a prime number or a composite number. Lehman's theorem states that if n is odd, a group $G=\{1, n-1\}$ exist if and only if n is prime. The following four steps describe the algorithm:

1) Pick a random integer a, such that $1 \leq a < n$,

   where n is an odd number to be tested for primality.

2) Let x be $a^{\frac{n-1}{2}} mod\ n$.

3) If x is either $1\ or\ (-1\ mod\ n)$, then n might be prime (or a prime_witness exist), otherwise, n is a composite number.

To be more certain, try with a few more different random values of a. If for a given n, the test returns prime_witness for 100 randomly chosen a, then the probability of n not being not prime is less than 2^{-100}.

Question 5 (4 marks)

Compute the following by demonstrating the step-by-step calculation correctly.

(i) Compute $21^{221} mod\ 123$ using fast exponentiation algorithm discussed in lecture and/or tutorial. Show all steps.
(ii) Compute $3037^{-1} mod\ 4051$
(iii) Does 337 have a multiplicative inverse modulo 1394? If yes, what is it?
(iv) What does it mean that g is a generator in Z_p^*?

Suggested answer:

(i) $21^{221} \bmod 123 = 21 \bmod 123 = 21$.

The student is expected to use fast exponentiation to compute.

221 (in decimal) = 11011101 (in binary)

	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
221 =	1	1	0	1	1	1	0	1
21^{221} mod 123	21^{128} mod 123	21^{64} mod 123		21^{16} mod 123	21^{8} mod 123	21^{4} mod 123		21^{1} mod 123
	78 mod 123	**18 mod 123**	51 mod 123	**57 mod 123**	**78 mod 123**	**18 mod 123**	72 mod 123	**21 mod 123**
	$19^{221} \bmod 123 = 78 \times 18 \times 57 \times 78 \times 18 \times 21 \bmod 123 =$ **21 mod 123**							

$Thus\ 21^{221} mod\ 123 = 21\ mod\ 123 = 21.$

(ii) Compute $3221^{-1} \bmod 4019$.

Suggested answer:

$3037^{-1} mod\ 4051 = -811\ mod\ 4051 = 3240$

The student is expected to use Extended Euclidean Algorithm to compute as follow:

n1	n2	r	q	a1	b1	a2	b2	a2'	b2'
4051	3037	1014	1	1	0	0	1	0	1
3037	1014	1009	2	0	1	1	-1	1	4050
1014	1009	5	1	1	-1	-2	3	4049	3
1009	5	4	201	-2	3	3	-4	3	4047
5	4	1	1	3	-4	-605	807	3446	807
4	1	0	4	-605	807	608	-811	608	3240

(iii) The student is expected to use Extended Euclidean Algorithm to compute as follow:

n1	n2	r	q	a1	b1	a2	b2
1394	337	46	4	1	0	0	1
337	46	15	7	0	1	1	-4
46	15	1	3	1	-4	-7	29
15	1	0	15	-7	29	22	-91

Since gcd(1394, 337) = 1, 337 has a multiplicative inverse modulo 1394; the multiplicative inverse is 337^{-1} mod 1394 = -91 mod 1394 = 1303.

(iv) Suggested solution:

G is a generator for Z_p^* if g^n mod p, where n = 1, 2, …,p-1, gives all elements of Z_p^*.

$$\forall y \in [1; p-1] : \exists k; y = g^k \bmod p$$

Question 6 (2 marks)

What is factorization problem? Show an example of a signature scheme that relies on the security of factorization problem.

Suggested answer:
Factorization refers to splitting of an integer number into a set of factors (a smaller set of numbers) which when multiplied together will get back the original integer. All integer numbers may be prime-factorized; i.e., expressed as a product of many prime numbers. When one has an integer number and wants to find the factors of these prime numbers, that can produce back the integer number, is difficult. This problem is known as factorization problem. Many public-key cryptosystems base on this factorization problem, including the RSA cryptosystem as well as RSA digital signature system.

RSA Digital Signature

Key generation:

- The key generation algorithm of RSA digital signature system is the same as the one employed by the RSA cryptosystem.
- Every user will generate his/her public key pair *(e, n)* and private key pair *(d, n)*. The user chooses two prime numbers *p* and *q* and compute the modulus $n = p \times q$.
- The user next chooses two more numbers, *e* and *d*. The number *e* is coprime (relatively prime) to *(p-1)(q-1)*. The number *d* is chosen such that $((e \times d) - 1)$ is divisible by *(p-1)(q-1)*.
- The *(e, n)* pair is the public key, and the *(d, n)* pair is private key.

Message signing:

- To sign a message, the sender signs the message using his/her private key; i.e.,
  $S=m^d \bmod n$
  Where
  - S is the signature,
  - m is the message,
  - (d, n) the sender's private key.

- The sender sends the message *m* and the signature *S* to the recipient (receiver).

Signature verification:

- To verify that the message is indeed signed by the sender, the receiver verifies the message authentication using the sender's public key; i.e.,

  $m' = S^e \bmod n$
  where
  - m' is the message recovered by decrypting the digital signature.
  - S is the sender's digital signature
  - (e, n) pair is the sender's public key.

- If the message received m is the same as the recovered message m', the receiver can be assured that the messages sent are authentic.

Example,
Suppose Alice wants to send a message *m* to Bob in such a way that Bob is assured the message is both authentic, has not been tampered with, and from Alice. Alice creates a digital signature *s* by exponentiation: $s=m^d \bmod n$, here *d* and *n* are Alice's private key. She sends *m* and *s* to Bob. To verify the signature, Bob exponentiates and checks that the message *m* is recovered: $m = s^e \bmod n$, where *e* and *n* are Alice's public key.

(The student can either describe the algorithm, or give example as above.)

Question 7 (2 marks)

A message was encrypted using Affine transformation cipher. You have the ciphertext, and it is KRQWL. You also happen to know the plaintext starts with a letter P and ends with a letter Y; in other words, the plaintext letter P is encrypted to K, and the plaintext letter Y is encrypted to L. Decrypt the ciphertext KRQWL. You can assume the Affice cipher used in this ecnryption uses only 26 alphabetic characters, and the mapping of the alphabets to its numerical equivalent is as follow:

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

Suggested solution:

Since Affine transformation cipher is done using $C = aX + b \pmod{26}$, and we know that a plaintext character P is transformed into a ciphertext character K, and the other plaintext characer Y is transformed into a ciphertext character L, we can establish the following two congruences:

1. $\mathrm{K} = \mathrm{aP} + \mathrm{b} \pmod{26}$ *or equivalently* $10 = 15\,a + b \pmod{26}$ $\cdots$ *congruence* 1
2. $\mathrm{L} = \mathrm{aY} + \mathrm{b} \pmod{26}$ *or equivalently* $11 = 24\,a + b \pmod{26}$ $\cdots$ *congruence* 2

Subtract congruence 1 (1) from congruence 2 (2), and we have:

$$
\begin{aligned}
11 &= 24\,a + b \pmod{26} \qquad (2)\\
\underline{10} &\underline{= 15\,a + b \pmod{26}} \quad \underline{(-} \qquad \underline{(1)}\\
1 &= 9\,a \pmod{26}\\
a &= 1 \times 9^{-1} \pmod{26}\\
&= 1 \times 3 \pmod{26}\\
&= 3 \bmod 26
\end{aligned}
$$

Substituting the value of a = 3 into the first congruence, we obtain the value of b as follow:

$$
\begin{aligned}
10 &= 15\,a + b \pmod{26}\\
10 &= (15)(3) + b \pmod{26}\\
10 &= 45 + b \pmod{26}\\
b &= 10 - 45 \pmod{26}\\
b &= 17
\end{aligned}
$$

Having calculated the value of a and b, we can now decrypt the rest of the ciphertext as follow:

$$
\begin{aligned}
C &= aX + b \pmod{26}\\
X &= (C - b) \times a^{-1} \pmod{26}
\end{aligned}
$$

If $a = 3$, then inverse a (a^{-1}) $c$an be calculated to be $a^{-1} = 9$.

Hence, for the rest of the ciphertext characters (R, Q, W) can be decrypted as:

R: $X = (C - b) \times a^{-1} \pmod{26}$
$= (17 - 17) \times 9 \pmod{26}$
$= 0 \bmod 26$
$X = A$

Q: $X = (C - b) \times a^{-1} \pmod{26}$
$= (16 - 17) \times 9 \pmod{26}$
$= -9 \pmod{26}$
$= 17 \bmod 26$
$X = R$

W: $X = (C - b) \times a^{-1} \pmod{26}$
$= (22 - 17) \times 9 \pmod{26}$
$= 45 \pmod{26}$
$= 19 \bmod 26$
$X = T$

Hence the plaintext message is PARTY.

Question 8 (2 marks)

In El Gamal encryption, the private key x in an element in Z_p^*. The public key is given by $y = g^x mod\ p$, where g is a primitive element in Z_p^*. To encrypt message m, one generates a random key $k < p - 1$, sets $c_1 = g^k\ mod\ p$, and $c_2 = m \times y^k\ mod\ p$, and output the ciphertext $c = c_1, c_2$. Describe the decryption process and show/prove that it always returns the encrypted message m.

Suggested answer:
To decrypt (c_1, c_2) with private key x, the recipient compute $m = \frac{c_2}{(c_1)^x} mod\ p$ in three steps:

(i) Compute $c_t = c_1^x\ mod\ p$ using fast exponentiation.
(ii) Compute the inverse $invC_t = c_t^{-1}\ mod\ p$ using extented Euclidean algorithm.
(iii) Compute $m = c_2 \times invC_t$ mod p.

The computation described above will always return the encrypted message m because

$m = \frac{c_2}{(c_1)^x} mod\ p$

$= \frac{m \times y^k}{(g^k)^x}\ mod\ p$

$= \frac{m \times y^k}{(g^x)^k}\ mod\ p \qquad since\ y = g^k mod\ p, hence$

$= \frac{m \times y^k}{y^k}\ mod\ p$

$m = m\ mod\ p$

$m = m.$

END OF TEST