1. ① Denning's modification protocols

Adv: use timestamp to ensure that the opponent does not perform replay attack, each message is sent in a fixed time slot.

Dis: sender's clock and receiver's clock needs to synchronize, so it is a hard problem to solve; and easy to be attacked by suppress-attack.

② Neuman 93 modification

Adv. use nounce instead of timestamp, so it does not need to synchronize clock.

Dis: the protocol is relatively complex than other protocols.

2. (a). When key length is minimum: i.e. m_1, the number of non-trivial key is $2^{9^{m_1}}-1$.

When key length is maximum: ie. m_2, the number of non-trivial key is $2^{9^{m_2}}-1$.

Thus: the number of non-trivial key is between $2^{9^{m_1}}-1$ and $2^{9^{m_2}}-1$.

(b). The number of non-trival key is the permutation of keylength minus 1, which is $m_3!-1$.

(c). The number of non-trival key is between $(2^{9^{m_1}}-1)(m_3!-1)$ and $(2^{9^{m_2}}-1)(m_3!-1)$.

3. (a) $a^{p-1}+(p-1)^a \bmod p$

$=\left[(a^{p-1}) \bmod p + (p-1)^a \bmod p\right] \bmod p$

$=\left[1 + (p-1)^a \bmod p\right] \bmod p$ Format's Theorem.

$=(1+p-1) \bmod p$ ..... Since $(p-1)^a \bmod p = (-1)^a \bmod p$ and a is an odd number, $(p-1)^a \bmod p = -1 \bmod p = p-1$

$=0$

(b). $3x^{14}+4x^{10}+6x-18 \bmod 5 \equiv 0$

$=3x^{5\cdot 2}\cdot x^5\cdot x^4 + 4\cdot x^5\cdot x^5 + 6x-18 \bmod 5 \equiv 0$

$=3x^6+4x^2+6x-18 \bmod 5 \equiv 0$

$=3x^2+4x^2+6x-18 \bmod 5 \equiv 0$

$=7x^2+6x-18 \bmod 5 \equiv 0$

$=(2x^2+x-3) \bmod 5 = 0$

$(2x+3)(x-1) \bmod 5 = 0$

$x = 1+5j$, where $j \in \mathbb{Z}$

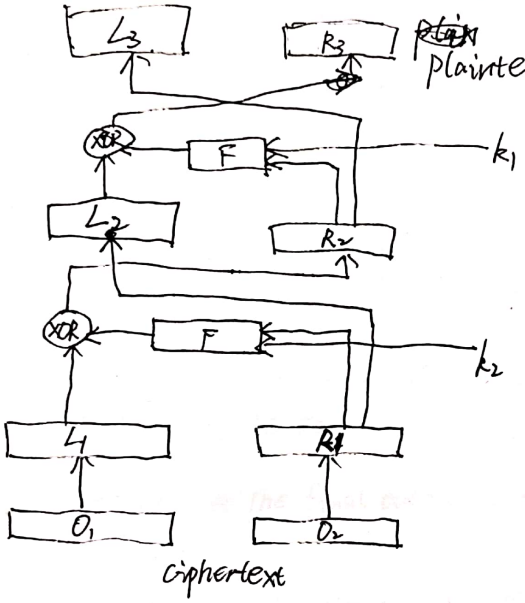
4. (a) ① key length is large enough

② ~~the~~ substitution and permutation blocks are used one after another to increase diffusion and confusion

③ the number of rounds are large enough.

④ easy to compute to increase efficiency.

(b).



5. $n = p \cdot q = 1189$

$\phi(n) = (p-1)(q-1) = 1120$.

Because we need a pair of e, d that is relatively prime to $\phi(n)$,

d can take all prime numbers 2, 3, 5, 7, ...

After eliminating all numbers that is not prime to 1120,

the three smallest possible values for d is 3, 11, 13.

6. (a).

i	Elements: x^i	As polynomials
$-\infty$	0	0
0	1	1
1	x	x
2	x^2	x^2
3	x^3	$1+x^2$
4	x^4	$1+x+x^2$
5	x^5	$1+x$
6	x^6	$x+x^2$
7	x^7	1

(b) x^2+x represents x^6, the multiplicative inverse is x.

7. (a) (i) It satisfies. We can take any-length of message, since we can split them into blocks.

(ii). It satisfies. The final output is performed by modulo n, which has a fixed size.

(iii). It satisfies, if RSA encryption is efficent to compute. XOR operation is easy to perform in modern computer.

(iv). It satisfies. If the length of a block is large enough, determine $M_1, M_2, \ldots M_m$ is equal to solve RSA problem which is hard.

(v). It does not satisfy. We can easy to find an alternative message such th

$M = M_1 \oplus M_2 \oplus \cdots \oplus M_m \oplus 0$, where 0 is a block of all 0.

(vi) It does not satisfy. Since it does not satisfy second pre-image resistant, the collision resistant is also fail to satisfy. The example is the same as the previous one.

8. (1). If we want the signature of $m_3 m_4^3 + m_5$,

what we need: $(m_3 m_4^3 + m_5)^d \bmod n$.

$$= \left[(m_3 m_4^3)^d \bmod n + m_5^d \bmod n \right] \bmod n$$

$$= \left[m_3^d (m_4^d)^3 \bmod n + S_5 \right] \bmod n$$

$$= \left[m_3^d \bmod n \cdot (m_4^d)^3 \bmod n + S_5 \right] \bmod n$$

$$= (S_3 \cdot S_4^3 + S_5) \bmod n.$$

(2). If we want the signature of $m_1^3 m_2^4 m_4^{513}$

what we need: $(m_1^3 m_2^4 m_4^{513})^d \bmod n$

$$= \left[(m_1^d{}^3 \bmod n) \cdot (m_2^{d4} \bmod n) \cdot (m_4^{d513} \bmod n) \right] \bmod n$$

$$= S_1^3 S_2^4 S_4^{513} \bmod n.$$

(3). If we want the signature of $m_1 + 19897 m_3 + 23987 m_5$,

what we need: $(m_1 + 19897 m_3 + 23987 m_5)^d \bmod n$

$$= \left[m_1^d \bmod n + 19897^d \cdot m_3^d \bmod n + 23987^d \cdot m_5^d \bmod n \right] \bmod n$$

$$= \left[S_1 + 19897^d S_3 \bmod n + 23987^d S_5 \bmod n \right] \bmod n$$

This signature needs to compute $19897^d \bmod n$ and $23987^d \bmod n$

Since we do not know private key d. we cannot forge this message.

9. No. This protocol is not vulnerable to man-in-the-middle attack, but it is still an insecure protocol. This protocol is susceptible to replay attack. Suppose there is an intruder Z, which recorded the old session key in step 3. Z can simply replay this old message to B, because B has discarded all previous session keys and thought I am communicating with A, but actually Z. When B replies a message, Z can intercept it and starts a communication with B. So the identity of A is impersonated by Z, which indicates this protocol is not insecure.

10. (1) If the masterkey is replaced by one-time pad key, it can provide the system perfect secrecy if one-time pad key is only use once. In this case, Opponent Z cannot replay message in step 3, since each communication's one-time pad key is unique. So replay attack can be addressed.

(2). If the session key is also replace by one-time pad key, the whole session will use the same one-time pad key, which is a threat to this system. Since using the same session key will leak the information, and if the opponent knows one message, he can easily decrypt any other messages.

(3) If the session key is distributed as one-time pad key for each communicating it will be inefficient to the system. The property of one-time pad key is the length of key is the same as that of message. If our message is long, our key must also be long, which is not a good encryption algorithm in practical purpose.