4 problems. 30 points. 40 minutes *Closed book. Closed notes. No electronic device.* *Write your name above.*

1. [5 points] Consider DES-CBC encryption of a message of 630 bits.

a. How many block encryption operations and block XOR operations are required?

Solution [3 pts]
DES has block size of 64 bits.
So 630-bit message would be padded to 640 bits (10 blocks) [1 pt]
10 block encryptions [1 pt]
10 block XOR operations [1 pt]
End solution

b. What is the size (in bits) of the output (i.e., from which the message can be recovered knowing the key).

Solution [2 pts]
Output consists of IV (64 bit) and ciphertext blocks (640 bits).
So 64 + 640 bits, which equals 704 bits. [2 pts]
End solution

2. [5 points] Given a hash function that outputs 32-bit hashes, your goal is to find a collision (i.e., two different messages that result in the same hash) with as few hash computations as possible and unlimited memory.

a. Around how many hashes would you expect to compute. Why.

Solution [3 pts]
Hash output space is 2^{32}.
Given two messages, prob of collision is $1/(2^{32})$.
Given K messages, prob of collision between any two messages is $K/(2^{32})$. (Birthday problem) [2 pts]
So want K to be of the order of square root of 2^{32}, i.e. 2^{16}. [1 pts]
End solution

b. What size (in bits) of messages would you hash. Why.

Solution [2 pts]
Want messages to not get too long (for efficiency). [1 pt]
32-bit messages would be fine; they would definitely yield a collision. [1 pt]
20-bit messages would also be ok, since we expect a collision in 2^{16} messages. [1 pt]
Gradually increasing message size from, say 8 bits, makes good sense. [1 pt]
End solution

2. [10 points] This concerns RSA. Starting from primes $p = 11$ and $q = 13$, obtain a public key $[e, n]$ and matching private key $[d, n]$. Your e should be the smallest possible.

Solution

$p = 11$, $q = 13$.

$n = p \cdot q = 11 \cdot 13 = 143$. [1 pt]

$\phi(n) = (p-1) \cdot (q-1) = 10 \cdot 12 = 120$. [3 pts]

e should be the smallest number satisfying $gcd(e, 120) = 1$.
2, 3, 4, 5, 6, 8, 9, 10 do not satisfy $gcd(e, 120) = 1$.
7, 11 do satisfy $gcd(e, 120) = 1$.
7 is the smallest.
So $e = 7$ [3 pts]
(Lose 1 pt for choosing 11)

d should be e^{-1} mod 120. [1 pt]

$d = 103$ for $e = 7$, by Euclid or ad hoc. [2 pt]

$d = 11$ for $e = 11$.

End solution

4. [10 points] Consider the following authentication handshake between A and B, where $[pub_A, pri_A]$ is A's public-key pair, $[pub_B, pri_B]$ is B's public-key pair, each has the other's public key, and $X(txt, k)$ is the encryption of txt using key k. (Assume RSA; so $X(txt, k)$ is signing if k is a private key.)

- A generates random n_A, sends $[X(n_A, pub_B)]$.
- B receives message, generates random n_B, sends $[X([n_B, n_A], pub_A)]$, sets the session key to $n_B \oplus n_A$.
- A receives message, sends $[X(n_B, pri_A)]$, sets the session key to $n_A \oplus n_B$.

Can an attacker who sees the messages obtain the session key? Explain briefly.

Solution [2 pts]

Attacker sees messages $[X(n_A, pub_B)]$, $[X([n_B, n_A], pub_A)]$ and $[X(n_B, pri_A)]$.

Atacker has pub_A and ub_B, does not have pri_A or pri_B.

So attacker can get n_B (from $[X(n_B, pri_A)]$). [6 pts]

But attacker cannot get n_A. [4 pts]

So attacker cannot construct $n_A \oplus n_B$.

In general, you got at most 3 points if you had something absurd (e.g., brute-force search through pri_A space, brute-force search through session-key space, extracting pri_A from $[X(n_B, pri_A)]$ and pub_A, etc.)

End solution