1. Let $G : K \to \{0,1\}^n$ be a secure PRG. Define $G'(k_1, k_2) = G(k_1) \wedge G(k_2)$ where $\wedge$ is the bit-wise AND function. Consider the following statistical test A on $\{0,1\}^n$: $A(x)$ outputs $LSB(x)$, the least significant bit of x.

What is $Adv_{PRG}[A, G']$?

You may assume that $LSB(G(k))$ is 0 for exactly half the seeds k in K.

2. Assume that you have a block cipher $E(k, m) = c$ for which the only attack exists has complexity of 2^{64}. You consider to double the key size by either:
$E'_1(k_1, k_2) := k_2 \oplus E(k_1, m)$ or
$E'_2(k_1, k_2) := E(k_1, k_2 \oplus m)$. Analyze which approach is better. What is the complexity of attacks on both new schemes?

3. Continuing with the previous problem. Find an attack which has the complexity smaller than 2^{192} on the following construction:
$E'_3(k_1, k_2, k_3) := k_1 \oplus E(k_2, k_3 \oplus m)$, where each key has 64 bits.

4. Nonce-based CBC. If one wants to use CBC encryption with a non-random unique nonce then the nonce must first be encrypted with an independent PRP key and the result then used as the CBC IV. Let's see what goes wrong if one encrypts the nonce with the same PRP key as the key used for CBC encryption.

Let $F : K \times \{0,1\}^l \to \{0,1\}^l$ be a secure PRP with, say, $l = 128$. Let n be a nonce and suppose one encrypts a message m by first computing $IV = F(k, n)$ and then using this IV in CBC encryption using $F(k, \cdot)$. Note that the same key k is used for computing the IV and for CBC encryption. Show that the resulting system is not nonce-based CPA secure.

The attacker begins by asking for the encryption of the two block message $m = (0^l, 0^l)$ with nonce $n = 0^l$. It receives back a two block ciphertext (c_0, c_1).

Observe that by definition of CBC we know that $c_1 = F(k, c_0)$. Next, the attacker asks for the encryption of the one block message $m_1 = c_0 \oplus c_1$ with nonce $n = c_0$. It receives back a one block ciphertext c'_0.

What relation holds between c_0, c_1, c'_0? Explain how these relations can let an adversary win the CPA game.

Hint: consider: $c_1 = c'_0$, $c_1 = 0^l$, $c_0 = c_1 \oplus c'_0$, $c_0 = c'_0$.

5. An administrator comes up with the following key management scheme: he generates an RSA modulus N and an element s in Z_N^*. He then gives to the ith user the secret key $s_i = s^{r_i} \bmod N$, where r_i is the i'th prime (i.e. 2 is the first prime, 3 is the second, and so on).

Now, the administrator encrypts a file that is accssible to users i, j and t with the key $k = s^{r_i r_j r_t} \bmod N$. It is easy to see that each of the three users can compute k. For example, user i computes k as $k = (s_i)^{r_j r_t}$. The administrator hopes that other than users i, j and t, no other user can compute k and access the file.

Show how any two colluding users can combine their secret keys to recover the master secret s and then access all files on the system.

6. Let G be a finite cyclic group of order p and let $pk = (g, h = g^a)$ and $sk = (g, a)$ be Bob's ElGamal public/secret key pair in G. To encrypt a message m, a random number r is selected and a ciphertext is set to: $\mathsf{Enc}(m) := \langle g^r, mh^r \rangle$.

Alice uses an PRNG which generates outputs: r_i and encrypts messages $m_1, \ldots, m_k$ obtaining ciphertexts $c_i = \langle g^{r_i}, m_i h^{r_i} \rangle$ which are then sent to Bob. Every message is intercepted by Eve. Some time later, Eve finds out that $r_{2i} = 2r_i$. Moreover she convinces Bob to reveal message m_j $(1 \leq j \leq k)$.

Can she learn a content of any other message? How she can do that? If Eve is able to pick j, how many messages she can read? Which one?

7. RSA function is defined over Z_N^*, where $N = pq$ with p, q primes. A public key is a pair $\langle N, e \rangle$ and a private key is $\langle N, d \rangle$ where $d = e^{-1} \bmod \phi(N)$.

Now: assume that RSA function is defined over Z_t^* where t is prime (instead of t being composite). Show how one can compute d from a public key $\langle t, e \rangle$.

8 Let O_n be an oracle that on input x returns a square root of x mod n, if one exists, and $\perp$ otherwise. Prove that there exists a probabilistic polynomialÂ-time algorithm that on input an integer n and access to O_n outputs n's factorization.

9. Consider the following signature scheme. Assume that the discrete logarithm problem in this (G) group is hard.

A Prover chooses g, an element of order p in G. Picks a private key: x and a public key: $S = g^x$.

To prove that a Prover knows x, a Prover and a Verifier perform the following protocol:

1. The Prover generates r, r', computes $C = g^r$ and $C' = g^{r'}(g^x)^{-1} \bmod p$ and sends $\langle C, C' \rangle$ to the Verifier.
2. The Verifier chooses a bit b and sends it to the Prover.
3. The Prover:
   if $b = 0$ then: sends r
   if $b = 1$ then: sends $t = x + r \mod p - 1$ together with r'.
4. The verifier checks if $C = g^r \bmod p$ (if $b = 0$), and in the case when $b = 1$ checks if $C'S = g^{r'} \bmod p$ and $S \cdot C = g^t \bmod p$.

Prove that the protocol is a zero-knowledge proof of knowledge (*i.e.,* show its: completeness, soundness and zero-knowledge).

How one can transfor the protocol into a signature scheme?

10. Let ISO denotes the language of all pairs of graphs (G, H) such that G is isomorphic to H.

Let $\langle GEN, COM, VER \rangle$ be a perfectly hiding commitment scheme. Prove that the following protocol is, in fact, a zeroÂknowledge proof system for ISO (with negligible soundness error).

1. The prover selects $g \leftarrow GEN(1^k)$ and sends g to the verifier.

2. The verifier chooses a k-Âbit random string r, selects $(c, d) \leftarrow COM(g, r)$ and sends c to the prover.

3. The prover randomly selects k graphs $C_1, \ldots, C_k$ such that each C_i is isomorphic to G and sends $C_1, \ldots, C_k$ to the verifier.

4. The verifier sends d and r to the prover.

5. If $r = VER(g, c, d)$ then for each graph C_i the prover sends the verifier a random isomorphism mapping G to C_i if the i-th bit of r is 0 and a random isomorphism mapping H to C_i if the i-th bit of r is 1.