Homework 4

1. **Factorizing the RSA modulus.** Let N be the product of two random n-bit prime numbers p and q. Recall that $\varphi(N)$ is the size of $\mathbb{Z}_N^*$, and we have $\varphi(N) = (p-1)(q-1)$. Construct an efficient algorithm that takes as input N and $\varphi(N)$, and outputs the prime factors of N.

2. **Sophie-Germain Primes.** Recall that the Prime Number Theorem states that there are roughly $\frac{N}{\log N}$ prime numbers $< N$. To generate a random n-bit prime number, recall that, we followed the following two steps

   - First, we counted the number of n-bit primes, and
   - Finally, we generated T random numbers and one of them turned out to be a prime number.

   We chose T such that the probability of finding an n-bit prime number in these T attempts is $\geqslant (1 - 2^{-t})$, for a parameter t.

   Now, we want to do this for the Sophie-Germain primes. We shall rely on the conjecture that there are $\frac{N}{\log^2 N}$ Sophie-Germain primes $< N$.

   (a) How many Sophie-Germain primes need n-bits in their binary representation?

   (b) Construct an algorithm that that as input (n, t) and outputs a random n-bit Sophie-Germain prime with probability $\geqslant (1 - 2^{-t})$.

3. **Encryption along with Signature.** Recall that in RSA-based public-key encryption, if Bob announces his public-key $\mathsf{pk}_B = (N_B, e_B)$ then other parties can encrypt and send messages to Bob that he can decrypt (using the trapdoor d_B that he keeps with himself).

   Recall that in RSA-based signatures, if Alice announces her public-key $\mathsf{pk}_A = (N_A, e_A)$ then she can sign messages that other people can verify that Alice has generated the signature (because Alice holds the trapdoor d_A).

   How can Alice encrypt a message m of her choice and send it to Bob so that only Bob can recover the message, and Bob is guaranteed that it is indeed Alice who sent the ciphertext?