CS/ECE407 Cryptography – Homework 5
Public Key Cryptography

Due: Wednesday, May 6, 11:59PM CT

Remember, you are free to collaborate with up to one classmate. See the course webpage for more details. You are expected to write out and submit your own solutions! Your collaboration is for discussing problems at a high level, not plagiarizing answers.

Your homework should be typed or carefully handwritten. We provide a LaTeX template for this document, if you would like to use it as a starting point. **If we cannot read your handwritten answers, they will not receive credit.**

Your typed solutions should be submitted through gradescope (see course webpage). Your hand-written solutions should be scanned and turned in through gradescope.

Problem 1 (Certificates)**.** Recall that a certificate is a document signed by some 'certificate authority' (CA) that associates an identity with a public key. For instance, let sk_C be the secret key of the CA. The following is a certificate associating Bob with some key pk_B:

$$Cert_{C \to B} = Sign(sk_C, \text{“Bob's key is } pk_B\text{”})$$

This problem involves *certificate revocation*, where, for example, a CA might maintain a list of certificates that should no longer be trusted. In the following, you may assume that the CA wishes to revoke any certificate for which the secret key has been compromised.

1. **(2 points)** Suppose Bob's secret key is leaked in a data breach, and so now he would like the CA to revoke this certificate, so that others cannot forge messages on his behalf. Bob sends the following message to the CA:

   $$m = \text{“please revoke } Cert_{C \to B}\text{”}$$

   The CA calls Bob on the phone to verify Bob's identity using non-cryptographic mechanisms. Why does the CA need to verify Bob's identity before revoking his certificate?

2. **(2 points)** Now, in addition to sending m, suppose that Bob also signs m and sends the following signature to the CA:

   $$\sigma = Sign(sk_B, m)$$

   Does the CA still need to call and verify Bob's identity before revoking his certificate? Why/why not?

Problem 2 (RSA)**.** Suppose Enc', Dec' is a symmetric-key encryption scheme with CPA security where keys are uniformly sampled from $\{0,1\}^\lambda$. Recall that the RSA key generation algorithm outputs a triple (N, e, d) such that:

$$e \cdot d \equiv 1 \mod \phi(N)$$

Let $H : \mathbb{Z}_N^* \to \{0,1\}^\lambda$ be a random oracle, and consider the following public-key encryption algorithm that uses RSA keys:

$Enc((N, e), m)$:
  $r \leftarrow \mathbb{Z}_N^*$
  $k = H(r)$
  return $(r^e \bmod N, Enc'(k, m))$

1. **(2 points)** Specify a corresponding decryption algorithm:

   $Dec((N, d), c)$:
     ...

2. **(2 points)** Prove that your decryption algorithm is correct.

3. **(2 points)** Argue that your public key scheme is CPA secure, under the RSA assumption. Namely, argue why it is that an adversary that can win the CPA game must be able to break either the CPA security of the symmetric-key scheme, or the RSA assumption.

Problem 3 (CCA Security for Public-Key Schemes)**.** In class we studied public-key schemes with CPA security. It is useful to also define *CCA security* for public-key encryption schemes. Recall that in the symmetric-key setting, CCA security is related to *malleability* – namely the ability for the adversary to induce a change in a message by changing the encryption of that message. The same is true for public-key schemes, and the definition of CCA security is similar:

Definition 1 (Public Key Encryption Security against Chosen Ciphertext Attacks)**.** Let $\Pi = (KeyGen, Enc, Dec)$ denote a public-key encryption scheme. We define two helper procedures:

$$Enc_0(pk, m_0, m_1) = \{\ c_0 \mid c_0 \leftarrow Enc(pk, m_0)\ \}$$
$$Enc_1(pk, m_0, m_1) = \{\ c_1 \mid c_1 \leftarrow Enc(pk, m_1)\ \}$$

We say Π achieves CCA security if the following ensembles are indistinguishable to any polytime adversary issuing *legal queries* (discussed next):

$$\left\{\ (pk, Dec(sk, \cdot), Enc_0(pk, \cdot, \cdot)) \;\middle|\; (sk, pk) \leftarrow KeyGen(1^\lambda)\ \right\} \approx$$
$$\left\{\ (pk, Dec(sk, \cdot), Enc_1(pk, \cdot, \cdot)) \;\middle|\; (sk, pk) \leftarrow KeyGen(1^\lambda)\ \right\}$$

That is, the adversary is given the public key pk, as well as oracle access to decryption, and oracle access either to Enc_0 or Enc_1. The *legal query* restriction states that the adversary may not pass any output obtained from its encryption oracle as input to its decryption oracle.

1. **(3 points)** Recall the ElGamal encryption scheme, which we discussed was CPA secure under the DDH assumption. Show that ElGamal is *not* CCA secure.