MIT 6.875J/18.425J and Berkeley CS276
Foundations of Cryptography (Fall 2020)

Problem Set 4: Released October 20, Due November 3

The problem set is due on **Tuesday November 3 at 10pm ET/7pm PT**. Please make sure to upload to the Gradescope course webpage by the deadline (all registered students should have access to this webpage on Gradescope). Be sure to mark on Gradescope where each problem's solution starts. Typed solutions using LaTeX are strongly encouraged (template provided on the course webpage).

Submit solutions to 4 out of the five problems. If you submit solutions to all 5 problems, your grade will be calculated from the top 4 highest scores.

Collaboration is permitted; however, you must write up your own solutions individually and acknowledge all of your collaborators.

Problem 1. Proof of Knowledge of RSA Inversion

A *proof of knowledge* is a stronger notion than a proof: the prover must convince the verifier not only that the statement is correct, but also that the prover "knows" a witness testifying to its correctness. This is formalized through the notion of an *extractor*, an algorithm that can extract the witness through interaction with the prover. Formally, the definition is as follows.

Definition 1 (Honest-verifier Zero-knowledge Proof of Knowledge)**.** *Let $R = \{(x, w)\}$ be a NP relation and the associated NP language $L = \{x | \exists w, R(x, w) = 1\}$. An honest-verifier zero-knowledge proof of knowledge with knowledge error k for relation R is an interactive proof protocol with a prover P and a verifier V with the following three properties:*

1. *(Correctness) $\forall (x, w) \in R, \Pr[\langle P(w), V)\rangle(x) = 1] = 1$.*
2. *(Honest-verifier Zero-knowledge) $\exists$ PPT simulator S, $\forall (x, w) \in L, \mathsf{View}_{P(w),V}(x) \approx_c S(x)$.*
3. *(Proof of knowledge) There is a polynomial-time extractor E such that for any $x \in L$, for any (possibly unbounded) P^* for which $p^*_x = \Pr[\langle P^*, V\rangle(x) = 1] > k$, we have*

$$\Pr[w \leftarrow E^{P^*}(x) : (x, w) \in R] \geq poly(p^*_x - k)$$

Note that soundness is implicit.

Let $N = pq$ and e be prime. Let $y \in \mathbb{Z}^*_N$ and $s = y^{1/e} \bmod N$. Let the prover P and the verifier V be given N, e, and y. Consider the following protocol for proving knowledge of s:

1. P picks r in $\mathbb{Z}^*_N$ randomly and sends $t = r^e \bmod N$ to V.
2. V picks $c \in \{0, 1\}$ randomly and sends c to P.
3. P computes $w = s^c \cdot r \bmod N$ and sends w to V.
4. V accepts if and only if $w^e = y^c \cdot t$.

1.1 Prove that the protocol above is an honest-verifier ZkPoK with knowledge error $\frac{1}{2}$. Remember to prove completeness and honest-verifier zero-knowledge, and to show proof of knowledge by demonstrating an extractor.

1.2 Does this mean that the prover necessarily knows a value d (that is, a value d can be extracted) such that $ed \equiv 1 \bmod \phi(N)$?

1.3 (Extra credit) Show that we can modify the protocol by choosing c from $\{0, 1, \ldots, B-1\}$ instead of $\{0, 1\}$, to get a lower knowledge error.

Problem 2. ZK and factoring

Suppose we have a number n, which we know is either a product of two distinct primes, or a product of three distinct primes. Alice wishes to prove to Bob (in zero knowledge) that it is a product of three primes.

Consider the following protocol:

1. Bob picks three elements at random (a, b, r) at random from $\mathbb{Z}_n^*$. He computes the three numbers a, b, and $c = ar^2$. He then sends the three numbers a, b, c to Alice in a random order.

2. Alice receives (x_1, x_2, x_3), finds two entries that are of the form $x_i = x_j r^2$ for some r (if there are multiple, pick one arbitrarily), and tells Bob $\{1, 2, 3\} \setminus \{i, j\}$. That is, Alice tells Bob which index is not i or j (this can also be viewed as Alice telling Bob which entry corresponds to b).

3. The above is repeated $100 \log n$ times. Each time in step 2, if the index sent to Bob corresponds to b, Bob gives Alice a point. If the total number of points Alice gets is more than $90 \log n$, then Bob accepts. Otherwise, he rejects.

2.1 Prove that for all large enough n, if n is a product of three primes, then if Alice and Bob follow the protocol above, with probability at least 2/3 Bob will Accept.

Hint: Think of the total number of quadratic residues that exist mod an n which is a product of two primes vs an n which is a product of three primes.

2.2 Prove that if n is a product of two primes, Bob will accept with probability at most 1/3 (even if Alice does not follow the protocol).

2.3 Show that this protocol is not honest-verifier zero knowledge. You may assume that there is no polynomial time algorithm A such that: there is some constant c greater than 3/4 such that on all inputs n where n is a product of three primes, with probability c the algorithm A on input n outputs a quadratic nonresidue r mod n such that the Jacobi symbol of r is $+1$.

Problem 3. NIZK for NP

Recall the ZK interactive proof for 3-coloring a graph with m edges. We will show a way to turn it into an NIZK interactive proof, given a certain setup described below.

As the setup, a third (trusted by everyone) party will create m public key encryption pairs of keys: $(s_1, p_1), (s_2, p_2), \ldots, (s_m, p_m)$. He will publish all of the p_i's (publicly for everyone to see), and then pick a random j, and send the verifier s_j (the prover does not know the value of j). For all $k \neq j$, the secret key s_k is destroyed.

For the NIZK proof, the input is a graph G with m edges. We describe the message sent by the prover to the verifier in the below.

1. the prover chooses a valid coloring, and creates a computationally hiding and information-theoretically binding commitment to the coloring of each vertex. Let C_v be the commitment corresponding to the vertex v.

2. For each edge $e_i = (u, v)$ of the graph, open the commitment C_u and C_v of colors of the two vertices of the edge to get M_i (that is, M_i contains the colors c_u and c_v of u and v, as well as the proof that c_u and c_v are indeed the colors that were commited to in C_v and C_u).

3. Encrypt M_i using public key p_i to create E_i

4. Send to Alice all of the C_vs, as well as all of the E_is.

To verify, the verifier will then use his secret key to determine the colors of the endpoints of edge e_j, and verify that they are distinct.

3.1 Prove that if the graph is indeed 3-colorable, the prover can make the verifier accept with probability 1.

3.2 Prove that if the graph is not 3-colorable, the verifier will reject with probability at least $1/m$.

3.3 Find (and prove) a new non-interactive proof in which if the graph is not 3-colorable, the verifier will reject with all but negligible probability. Additionally, if the graph is indeed 3-colorable, the prover can make the verifier accept with probability 1.

3.4 Is your interactive proof from part 3.3 zero knowledge? If so, prove that it is. If not, find a new protocol that is zero knowledge (and satisfies the conditions of part 3.3), and prove it is zero knowledge.

Notice that the model here is different from the model described in class. Specifically, the verifier is given some secret information (namely, s_j). In your simulation argument to show that this problem is zero knowledge, you must also make sure that the verfier can simulate s_j.

Problem 4. ZK and DDH

Let p be a prime. Consider the problem where the input is a 4-tuple $(a \bmod p, b \bmod p, c \bmod p, d \bmod p)$ of elements of $\mathbb{Z}_p^*$. Let L be the set of 4-tuples that can be represented in the form $(g_1 \bmod p, g_2 \bmod p, g_1^x \bmod p, g_2^x \bmod p)$ for some x, g_1, and g_2, and otherwise the output is 0. That is, if there is some x such that the third element is the first element raised to the x power, and the fourth element is the second element raised to the x power, then the 4-tuple is in L.

Consider the computational problem of determining whether an element is in L or not.

4.1 Show that if there is a polynomial time algorithm for distinguishing if quadruples are in L or not, then there is also a polynomial time algorithm for DDH.[1]

4.2 Find an honest-verifier 2-round perfect zero-knowledge (interactive) proof for proving a 4-tuple is an element of L.

Problem 5. CCA Encryption from IBE

In this problem, we will introduce the concept of IBE (Identity Based Encryption), and show that it can be used to achieve CCA-security.

We define IBE below. Intuitively speaking, the idea of IBE is that there is a party called the PKG (public key generator). Anyone else has some ID, and they can go to the PKG, who can use their ID (along with a master secret key, which only the PKG knows), to generate a private key for the user with that ID. Anyone can, given an ID r, encrypt a message so that only the private key corresponding to r can decrypt the message.

Formally, IBE is four algorithms ($\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt}$) which satisfy:

[1] Recall that in the DDH problem, the input is a prime p, generator g of $\mathbb{Z}_p^*$, and a triple $(g^x \bmod p, g^y \bmod p, g^z \bmod p)$ of elements of $\mathbb{Z}_p^*$. This triple is either sampled by (1) picking x, y and z uniformly at random from $\mathbb{Z}_p^*$, or by (2) picking x and y uniformly at random from $\mathbb{Z}_p^*$, and then setting $z = xy$. The DDH problem is to distinguish between situations (1) and (2) with non-negligible probability.

- Setup: This algorithm is run by the PKG one time for creating the whole IBE environment. The master key is kept secret and used to derive users' private keys, while the system parameters are made public. It accepts a security parameter k (i.e. binary length of key material) and outputs:
  1. A set $\mathcal{P}$ of system parameters, including the message space and ciphertext space $\mathcal{M}$ and $\mathcal{C}$,
  2. A master key K_m.
- Extract: This algorithm is run by the PKG when a user requests his private key. Note that the verification of the authenticity of the requestor and the secure transport of d are problems with which IBE protocols do not try to deal. It takes as input $\mathcal{P}$, K_m and an identifier $ID \in \{0,1\}^*$ and returns the private key d for user ID.
- Encrypt: Takes $\mathcal{P}$, a message $m \in \mathcal{M}$ and $ID \in \{0,1\}^*$ and outputs the encryption $c \in \mathcal{C}$.
- Decrypt: Accepts d, $\mathcal{P}$ and $c \in \mathcal{C}$ and returns $m \in \mathcal{M}$.

Assume we have a semantically-secure IBE scheme ($\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt}$). We want to use it to construct a CCA-secure (non-identity-based) public-key encryption scheme ($\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$). We define it as follows:

- $\mathsf{Gen}(1^n)$: Run $\mathsf{Setup}(1^n)$ to obtain public parameters PP and master secret key msk. Set the public key $\mathsf{pk} = \mathsf{PP}$ and the secret key $\mathsf{sk} = \mathsf{msk}$.
- $\mathsf{Enc}(\ \mathsf{pk}\ , m)$:
  - Choose random $r \in \{0,1\}^n$.
  - Let $c = \mathsf{Encrypt}(\mathsf{PP}, r, m)$. That is, we encrypt m to the "identity" r using the IBE scheme.
  - Output (c, r) as the encryption.

5.1 Give the corresponding decryption algorithm Dec. Argue correctness (that is, show that the decryption of a valid encryption will be the original message).

5.2 Prove that this scheme is secure against CCA1 ("lunchtime") attacks.