Modern Cryptology | Spring 2022

Problem Set 3

Ron Rothblum | Due: 14/6/2022

Reminder: please type your answers (LaTeX is encouraged but not mandatory).

1 Public-key Encryption from QR

In this question, we will build a public-key encryption scheme from the quadratic residuosity problem. Throughout this question, assume that P and Q are primes and $N = P \cdot Q$.

1.1 QR given factorization

Show a polynomial-time algorithm that given two primes P and Q and $x \in \mathbb{Z}_N^*$, decides if x is a QR modulo $N = P \cdot Q$.

1.2 Generating QR

Let x be a quadratic *non* residue (QNR) modulo N with Jacobi Symbol 1, i.e., $x \in QNR(N)$.[1] Prove that if $y \overset{\$}{\sim} \mathbb{Z}_N^*$ then $(y^2 \cdot x) \overset{\$}{\sim} QNR(N)$.

1.3 Public-key Encryption

Show a PKE scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ for encrypting single bits that is CPA secure under the QRP assumption.

Guideline 1.1. *For simplicity, you may assume that* $-1 \mod N$ *is always a QNR, e.g. this assumption holds for* Blum Integers

Remark 1.2. *For this question you may assume a (strict) probabilistic polynomial-time algorithm for sampling uniformly random* n*-bit prime numbers.*

1.4 Malleability

Prove that given encryptions of two bits (that were encrypted using the same public key) using the PKE that you suggested, you can generate an encryption of their XOR (which can be decrypted using the same secret key).

1.5 Refresh

Construct a PPT algorithm $\mathsf{Refresh}(pk, c)$ that outputs a "fresh" random encryption of the same plaintext of c. Formally, for any message $m \in \{0,1\}$, $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$ and $c \leftarrow \mathsf{Enc}(pk, m)$ it holds that the distributions $\mathsf{Refresh}(pk, c), \mathsf{Enc}(pk, m)$ are identical.

[1] As this set is defined in the tutorial.

2 Statistically Hiding Commitments

Denote by $\langle a, b\rangle = \sum_i a_i \cdot b_i \pmod 2$.

2.1 Inner Product with Random String

Let $b \in \{0,1\}^n$ be a non-zero vector. Show that $\Pr_{a\in\{0,1\}^n}[\langle a, b\rangle = 0] = \Pr_{a\in\{0,1\}^n}[\langle a, b\rangle = 1] = 1/2$.

2.2 Inner Product is Universal

For every $a \in \{0,1\}^n$ define $h_a : \{0,1\}^n \to \{0,1\}$ as the function $h_a(b) = \langle a, b\rangle$. Show that the collection $\{h_a\}_{a\in\{0,1\}^n}$ is a universal hash function family.

2.3 Purifying Randomness

Let $S \subseteq \{0,1\}^n$. Show that the following two distributions are $O(\sqrt{1/|S|})$ close to uniform (in statistical distance):

- $r, \langle r, s\rangle$; and
- r, b,

where r is uniform in $\{0,1\}^n$, s is uniform in S and b is a uniformly distributed bit.

2.4 Commitments

Assume the existence of a collision resistant hash function mapping strings of length n to outputs of length $n/2$. Show that there exists a statistically hiding computationally binding commitment scheme. You may assume for simplicity that the hash function is regular (i.e., every element in the range has the same number of preimages).

Guideline 2.1. *Consider $C(b) = (h(s), r, \langle r, s\rangle \oplus b$, where $s, r \in_R \{0,1\}^n$.*

Remark 2.2. *For this question you may consider commitments in the "common reference string" (CRS) model. In this model, a trusted party first generates some string (under some distribution) to which all of the parties (both honest and dishonest) have access to. The definition of a commitment scheme is adapted to this notion in the natural way (i.e., by letting all parties have access to the CRS and requiring all security properties to hold even given the CRS).*

3 Is Factoring NP complete?

In this question you will show that under a widely believed complexity-theoretic assumption the factoring problem is not NP complete (thereby answering a question that was raised in class). To formalize this we first introduce a *decision* problem (i.e., a YES/NO question) which is computationally equivalent to factoring (so that it even makes sense to talk about NP, which is a class of decision problems).

Define the language $L = \{(N, M) \in \mathbb{N} \times \mathbb{N} : N$ has a prime factor larger than $M\}$.

3.1 Equivalence to Factoring

Show that L can be solved in polynomial-time if and only if factoring can be solved in polynomial-time.

3.2 coNP

Show that $L \in NP \cap coNP$. Deduce that if L is NP-complete than $NP \subseteq coNP$.[2]

[2]Note that it is widely believed that $NP \nsubseteq coNP$ (c.f. https://tinyurl.com/bdzey65a)