INDIAN STATISTICAL INSTITUTE, KOLKATA

Assignment for Cryptology Internship 2014

Due : 9 June 2014

There are five problems in total, and attempting all problems is mandatory. Each of the following sections contain one problem, and some related references to motivate the same. Submit the solutions in the desired format, as specified individually for each problem.

MATHEMATICS

Prime numbers are probably the most mysterious elements that we encounter in the realm of mathematics. Centuries back, Euclid proved that there exist infinitely many prime numbers. However, it still remains an open problem to prove that there exist infinitely many pairs of consecutive prime numbers $(p, p+2)$. The (in)famous and unproved *twin prime conjecture* claims that "there are infinitely many primes p such that $p+2$ is also a prime".

If a prime p is such that both $p-2$ and $p+2$ are composites, then it is called a '*lone prime*'. Otherwise, it belongs to a twin-prime pair. In this context, solve the following problem.

Problem 1.

Is the prime $p = 7150309735704750361$ a lone prime, or does it belong to a twin prime pair? If it does belong to a twin prime pair, find its twin as well.

Submission: If you are proving the result theoretically, submit a complete proof (typed or handwritten) in soft/hard copy. If you are writing a code to solve the problem, submit the code and a detailed README file so that we may execute the code properly, and also submit a theoretical justification for the design and development of the algorithm that you have coded.

BLOCK CIPHERS

Data Encryption Standard (DES) is quite famous a block cipher, constructed on the basis of the Feistel network structure. Let us denote the DES encryption formally as a function $DES : K \times M \to C$ such that $c = DES(k, m)$ for a random key $k \in K$, plaintext $m \in M$ and the resultant ciphertext $c \in C$.

Consider the standard instantiation of DES with 56-bit key k, 64-bit plaintext m and 64-bit ciphertext c, that is, $DES : \{0,1\}^{56} \times \{0,1\}^{64} \to \{0,1\}^{64}$. Let us denote the bitwise complement of any bit-string x as $x^{(c)}$. Solve the following problem in this context.

Problem 2.

Show that for every key $k \in K$ and every plaintext $m \in M$, the function DES satisfies the following complementarity property

$$DES(k, m)^{(c)} = DES\left(k^{(c)}, m^{(c)}\right).$$

if an adversary may use either of ciphertext-only, known-plaintext, chosen-plaintext or chosen-ciphertext attack, is it possible to use the above property of DES to mount a key-recovery attack on the cipher that recovers the 56-bit key k with probability 1 in time 2^{55}?

Submission: Solve the problem theoretically, and submit in soft/hard copy (typed/written).

STREAM CIPHERS

The theoretical model for stream ciphers is the pseudo-random generator (PRG), which claims to produce a *long* pseudo-random or 'random-looking' stream of bits (or words) from an input of a *short* truly random key. Thus, an efficient observation of any non-random behavior of the output stream from a stream cipher is sufficient to distinguish it from a pseudo-random generator.

RC4 is probably the simplest stream cipher still in practice, with a 256-byte random key $k \in K$ and a 256-byte internal state S in its generic setting. The output of RC4 is a stream of bytes, and for each output byte z, we may consider the probability distribution $\Pr_K(z = v)$, where $v = 0, 1, \ldots, 255$, and the probability is calculated over the keyspace K. In this context, solve the following problem.

Problem 3.

Compute an approximate probability distribution (with a *small* random sample of keys from K) for the first 16 bytes of the RC4 output stream, that is, approximately compute

$$\Pr_K(z_i = v) \qquad \text{for } i = 1, 2, \ldots, 16, \quad \text{and } v = 0, 1, \ldots, 255.$$

Identify and try to prove any non-random behavior that you can spot in these distributions.

Submission: For the experimental part of the problem, submit the code and a detailed README file so that we may execute the code properly. For the theoretical portion where you attempt the proofs, submit your solution in soft/hard copy (typed/written).

INTRACTABLE PROBLEMS

One of the most famous intractable problems in Cryptology is the problem of *integer factorization*. If we construct a composite N by multiplying two random primes p, q, it is believed to be computationally hard (theoretically) to find the factors p, q from the composite N.

However, if the choice of the primes p, q are *poor*, it may be possible to factorize N within a computationally feasible time-frame, even if the primes are *large*. One such *poor* choice of primes p, q is attained when the primes are too close to one another. In this context, solve the following problem.

Problem 4.

Factorize $N = 5112692931651420847690077826024274104 3$ when it is given that $N = pq$ for primes p, q which are close to each other, satisfying $|p - q| < 2N^{0.25}$. What is the estimated running-time (in 2^x terms) for your factorization algorithm in this specific case?

Submission: If you are solving the problem theoretically, submit a complete proof (typed or handwritten) in soft/hard copy. If you are writing a code to solve the problem, submit the code and a detailed README file so that we may execute the code properly, and also submit a theoretical justification for the design and development of the algorithm that you have coded.

PUBLIC KEY CRYPTOGRAPHY

The El-Gamal encryption scheme is quite popular in modern public key cryptography. One of the main notions for security in case of a public key encryption scheme relies on the property of having indistinguishable encryptions under a chosen-plaintext attack (known as IND-CPA security).

To prove that an encryption scheme is IND-CPA secure, we generally require an associated hard problem assumption in its basis for construction. In this context, solve the following problem.

Problem 5.

To prove that El-Gamal encryption scheme is IND-CPA secure, what hard problem assumption do we require in its construction? Prove that if this problem is hard, then El-Gamal encryption scheme is IND-CPA secure in its generic setting.

Submission: Solve the problem theoretically, and submit in soft/hard copy (typed/written).