Cryptology complementary
Final Examination

2019-05-16

Instructions

The duration of this examination is one hour and a half. Answers to the questions must be detailed and complete to get maximum credit.

Exercise 1: A random sequence

Let $\mathcal{S}$ be a set of N arbitrary elements; let $(u_n)_{n\in\mathbb{N}}$ be a sequence whose elements are drawn independently and uniformly at random from $\mathcal{S}$, i.e. for all i, $u_i \xleftarrow{\$} \mathcal{S}$. Suppose that you do not initially know $\mathcal{S}$, nor N.

Q. 1:

1. Give an efficient algorithm that takes as input a finite number of elements of (u_n) and that returns an approximation of N.
2. What is the time and memory complexity of your algorithm?

Exercise 2: Block cipher key search

We consider a block cipher (i.e. a family of permutations) of signature $E : \{0,1\}^{256} \times \{0,1\}^{128} \to \{0,1\}^{128}$. We denote by $\mathcal{P}$ the set of all permutations over $\{0,1\}^{128}$ and by I the identity mapping over $\{0,1\}^{128}$.

Q. 1:

1. What is the size of $\mathcal{P}$?
2. How many permutations does E define? Are they all necessarily distinct?
3. Assuming for simplicity that $\forall\, k \in \{0,1\}^{256}$, $E(k,\cdot) \xleftarrow{\$} \mathcal{P}$, compute $\Pr[\exists\, k |\, E(k,\cdot) = I]$, where the probability is taken over the previously mentioned random drawings (you don't need to simplify the expression of this probability).

We now consider the following setting: a challenger picks a key $k \xleftarrow{\$} \{0,1\}^{256}$, and an adversary is provided with some plaintext/ciphertext pairs for $E(k,\cdot)$, i.e. it is given some pairs of the form $(p, c := E(k,p))$. The goal of the adversary is to recover k; we model it as an algorithm that takes the plaintext/ciphertext pairs as input and outputs a key candidate k'. We say that the adversary is successful if $k' = k$.

Q. 2: We first consider a case where $\forall\, k \in \{0,1\}^{256}$, $E(k,\cdot) = I$.

1. Compute the probability that *any* adversary is successful when provided with $N = 1337$ plaintext/ciphertext pairs.
2. Does this probability depend on N?
3. Give an algorithm with minimum time complexity that reaches the above probability.

Q. 3: We now assume a simplified model where $\forall\, k \in \{0,1\}^{256}$, $E(k,\cdot) \xleftarrow{\$} \mathcal{P}$.

1. We first consider a single plaintext/ciphertext pair (p,c). What is the expected size X of $\{k' | c = E(k',p)\}$?
2. Give an algorithm for an adversary that is successful with probability $1/X$. What is its expected time complexity, where the unit is the number of computations of E?
3. How many plaintext/ciphertext pairs are required to have $X \approx 1$ (give a general formula in function of the key size κ and the block size n of a block cipher. You are not required to compute the *exact* expected size; a reasonably good approximation is sufficient). What is the expected time complexity of the algorithm in this case?

Q. 4: We now assume that E possesses a complementation property similar to the one exhibited by the DES block cipher, that is $\forall\, k, x\ E(k,x) = \overline{E(\overline{k},\overline{x})}$, where $\overline{x}$ denotes x with all its bits flipped.

1. Give a partition of $\{0,1\}^{256}$ into two sets $\mathcal{A}$, $\mathcal{B}$ s.t. $x \in \mathcal{A} \Leftrightarrow \overline{x} \in \mathcal{B}$. What are the sizes of $\mathcal{A}$ and $\mathcal{B}$?
2. Consider an adversary given the following plaintext/ciphertext pairs as input: $(p_1, c_1 = E(k,p_1))$, (p_2,c_2), (p_3,c_3), $(\overline{p_1}, c_4 = E(k,\overline{p_1}))$, $(\overline{p_2},c_5)$, $(\overline{p_3},c_6)$. Modify your algorithm of the previous question to enumerate keys based on the above partition. Does this allow you to decrease the expected time complexity?

Q. 5: We finally consider the case where $E(k,\cdot)$ is linear: $\forall\, k\ \exists\, \boldsymbol{K} \in \mathrm{GL}(128, \mathbb{F}_2)$ s.t. $E(k,x) = \boldsymbol{x}\boldsymbol{K}$, where $\boldsymbol{x}$ is the canonical embedding of x into $\mathbb{F}_2^{128}$.

1. Give an algorithm that recovers $\boldsymbol{K}$ from a few well-chosen plaintext/ciphertext pairs.
2. Does your algorithm always allow to recover k? If not, does it mean that it is not an attack on E? Propose a security model for block ciphers under which your algorithm is clearly an efficient attack.