Crypto Engineering
Discrete probability

2019-09-26

In all of the following exercises, we let $\mathcal{S}$ be an arbitrary finite set of size N, and we denote by $X \xleftarrow{\$} \mathcal{S}$ the process of drawing X from $\mathcal{S}$ uniformly at random, and independently of any other process.

Exercise 1: (multi-)collisions

Let $X \xleftarrow{\$} \mathcal{S}$, $Y \xleftarrow{\$} \mathcal{S}$, $Z \xleftarrow{\$} \mathcal{S}$.

1. Compute $\Pr[(X = x) \wedge (Y = y)]$ for any $x, y \in \mathcal{S}$.
2. Compute $\Pr[X = Y]$.
3. Compute $\Pr[X = Y = Z]$.

Exercise 2: (non-)uniform masks

Let X and Y be two independent random variables drawn from $\mathbb{F}_2$ with a uniform law for X and an unknown arbitrary law for Y.

1. What is the distribution of $X + Y$? (That is, compute $\Pr[X + Y = 0]$)

We now draw X and Y from a finite group $(\mathbb{G}, +)$ of size N.

2. What is (again) the distribution of $X + Y$?

Remark. This result is essential in cryptography, and is used to justify the security of many constructions.

We go back to X and Y being drawn over $\mathbb{F}_2$, but consider this time arbitrary laws for both of them. We write c_X the *correlation bias* of X defined as $c_X = |2\Pr[X = 0] - 1|$, and the same for c_Y.

3. Compute c_{X+Y}, the correlation bias of $X + Y$.
4. By indudction, give a formula for the correlation bias of the sum $X_1 + \cdots X_N$ of N variables of correlation biases $c_1, \ldots, c_N$.

Remark. This last result is known in cryptography as the *piling-up lemma*.

Exercise 3: (close-to) uniform permutations

We consider the following algorithm to generate a random permutation of $[\![1, N]\!]$ (or more generally, of N arbitrary elements): 1) build a list of N pairs (r_i, i), where $r_i \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$; 2) sort the list according to the first element of the pairs; 3) return the list of the second element of the pairs in the sorted order.

Q.1 : Compute the number of sorted lists of N elements of $\mathbb{Z}/q\mathbb{Z}$.

Hint: Map all such possible lists to paths from $(0, 1)$ to (N, q) in the 2-dimensional discrete grid, where only horizontal and vertical steps are allowed.

Q.2 :

1. For every possible permutation generated by the algorithm, compute the *maximum* number of drawings for $(r_1, \ldots, r_N)$ that lead to it.
2. What is then the maximum probability of occurence of any permutation?
3. Express this probability as $\delta/N!$ for δ of the form $\prod_{i=1}^{N-1}(1 + x_i/q)$.
4. For a fixed N, give an approximative criterion on q for δ to be close to 1.

We now consider a variant of the algorithm, where one is interested in drawing a random combination of weight w. This is done as follows: 1) build a list of N pairs $(r_i, i \texttt{ <= w ? } 1 \texttt{ : } 0)$, where $r_i \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$; 2) sort the list according to the first element of the pairs; 3) return the list of the second element of the pairs in the sorted order.

Q.3 :

1. For every possible combination generated by the algorithm, compute the *maximum* number of drawings for $(r_1, \ldots, r_N)$ that lead to it.
2. What is then the maximum probability of occurence of any combination?
3. Express this probability as $\delta/\binom{N}{w}$ for δ of the form $\prod_{i=1}^{N-1}(1 + x_i/q)$.
4. How could this have been found directly by using the result of **Q.2**?

Remark. Generating (close-to) uniform permutations and combinations is an important step in code- and lattice-based cryptosystems. This exercise is based on: https://ntruprime.cr.yp.to/divergence-20180430.pdf.