

(Applied) Cryptography

Tutorial #2

Bernardo Portela (bernardo.portela@fc.up.pt) Rogério Reis (rvreis@fc.up.pt)

MSI/MCC/MERSI – 2023/2024

Recall that a probability distribution D over a set \mathcal{S} can be seen as a deterministic function mapping random coins C sampled uniformly at random from a set \mathcal{C} to \mathcal{S} . In this case, the probability mass function is defined, for all $S' \in \mathcal{S}$, as:

$$\Pr[S = S' : S \leftarrow \$ D] = \Pr[S = S' : C \leftarrow \$ \mathcal{C}; S \leftarrow D(C)] = \frac{\#\{C : D(C) = S'\}}{|\mathcal{C}|}$$

We abbreviate this, when clear from the context, to $\Pr[S']$.

Recall also that the entropy of such a distribution is given by:

$$\sum_{S' \in \mathcal{S}} -\Pr[S'] \cdot \log_2(\Pr[S'])$$

For example, the entropy associated with a perfect coin flip is $-\frac{1}{2} \cdot \log_2(\frac{1}{2}) + (-\frac{1}{2} \cdot \log_2(\frac{1}{2})) = 1$.

Answer the following questions

1 - Consider \mathcal{S} the set of integers in the range $0..250$ and note $p = 251$ is a prime number. Take \mathcal{C} to be the set of all bit strings of length 8. Let the distribution D to be defined by the function $D(C) := C \pmod{p}$, i.e. takes the remainder of coins C divided by p .

- Calculate the probability of each value in \mathcal{S} to be produced by D .
- Repeat the above considering now the set \mathcal{C} to be the set of all bit strings of length 64.
- Are these distributions uniform? If not, can you think of a way to quantify how distant they are from uniform?

2 - Repeat question #1 but take $p = 2^8$, i.e., a power of 2.

3 - Use Sage to compute the entropy of the two distributions referred in questions #1 and #2. Compute also the entropy of the uniform distribution over \mathcal{S} .

4 - Generalize the computations from question #3 in Sage to compute the entropy of distribution D when \mathcal{C} is the set of bit strings of length k . Check (approximately) what is the smallest k for which the entropy computed in Sage for D matches the entropy of the uniform distribution over \mathcal{S}

5 - `hexdump` can be used to extract randomness from `/dev/urandom`. Explain what the following command is doing.

```
$ hexdump -n 32 -e '1/4 "%0X" 1 "\n"' /dev/urandom
```

Implement an alternative command that uses `/dev/urandom` to create a file with random bytes.

- HINT: use the shell `dd` command.

Use `openssl` to do exactly the same.

- HINT: look at command `rand`.

6 - Use `openssl` to generate a key pair where private key is protected with a password.

```
openssl genrsa 4096
```

See what happens when you increase/decrease the key size.

Investigate how openssl converts the passphrase into a cryptography key for encryption/wrapping.

7 - Use openssl to generate random Diffie-Hellman parameters.

```
openssl dhparam 2048
```

See what happens when you increase/decrease the key size. Compare to the previous case.