1 Birthday Paradox

1.1

A deck of 52 cards is shuffled and the top eight cards are turned over.

- What is the probability that the king of hearts is visible?
- A second deck is shuffled and its top eight cards are turned over. What is the probability that any visible card from the first deck matches a visible card from the second deck?

Proof. [SOLUTION: Question is from "An Introduction to Mathematical Cryptography"] □

1.2

Suppose there are k people in a room, and assume that there are 365 possible birthdays, equally likely and independent for each person.

Use the approximation $\frac{k(k-1)}{2n}$ the chance that at least two of them share a birthday for the following values of k:

- $k = 5$
- $k = 10$
- $k = 15$
- $k = 23$
- $k = 40$
- $k = 100$

Now calculate the exact probabilities (using a calculator / WolframAlpha). When does the estimate become unreliable?

Proof. [SOLUTION:

$$\begin{pmatrix} 2 & 0.00273973 & 0.00273973 \\ 5 & 0.0271356 & 0.0273973 \\ 10 & 0.116948 & 0.123288 \\ 15 & 0.252901 & 0.287671 \\ 23 & 0.507297 & 0.693151 \\ 40 & 0.891232 & 2.13699 \\ 100 & 1. & 13.5616 \\ 360 & 1. & 177.041 \end{pmatrix}$$

] □

1.3

Suppose you are using 64-bit random numbers. Roughly how many numbers can you generate before there is likely to be a collision?

What if you generate 128-bit numbers instead? 32-bit numbers?

Proof. [SOLUTION: $2^{32}, 2^{64}, 2^{16}$] □

1.4

Suppose you are generating k keys, and need them all to be different. Roughly how large does the key space need to be for the following values of k?

- $k = 100$
- $k = 10$
- $k = 2$

Proof. [SOLUTION: $n > k^2/2p$] □

1.5

Suppose everyone uses RSA with two primes p and q, each selected uniformly at random from all the primes with 256 bits.

Assuming you are only worried about collisions, about how many keys can be generated so that the probability that none of the keys share any primes is $< 1/2^{40}$ (one in a trillion)?

You'll need to use the fact that there are about $\frac{n}{2\log_e n}$ n-bit primes. Keep in mind that every RSA key contains *2* primes.

Proof. [SOLUTION: Solve

$$\frac{(2k)^2}{2 \cdot 2^{256}/2\log_e 2^{256}} = 1/2^{40}$$

for k:

$$k \approx 2.43 \cdot 10^{31}$$

(Note the answer on the previous line is in base 10.] □

2 Baby-Step Giant Step

Recall the baby-step giant-step algorithm for finding x given $(g, h = g^x)$

- $Let m = \lceil \sqrt{n} \rceil$
- Make a lookup table:
  - $h \cdot g^0 \to k = 0$
  - $h \cdot g^1 \to k = 1$
  - ...
  - $h \cdot g^m \to k = m$
- For $i = 0, m, 2m, ..., p$:
  - If g^i is in the table as $h \cdot g^k$, output $i - k$.

Solve the following discrete log problem instances.

- $G = \mathbb{Z}_{29}^*$, $g = 3$, $g^x = 7$ **Proof.** [SOLUTION: 8] □
- $G = \mathbb{Z}_{67}^*$, $g = 2$, $g^x = 53$ **Proof.** [SOLUTION: 21] □
- $G = \mathbb{Z}_{257}^*$, $g = 27$, $g^x = 57$ **Proof.** [SOLUTION: 42] □

How many steps did each one take you? How long would it have taken the naive way?

2.1

Suppose $g \in \mathbb{Z}_p^*$. If x is a random exponent, how many step on average will it take to solve the discrete log problem using baby-step giant-step.

Proof. [SOLUTION: $1.5 \cdot \sqrt{p}$] □

3 Hash Tables

Ask Lucas about how to make baby-step giant-step actually run fast.

4 Malleability

Use the homomorphic properties of these systems to create and modify ciphertexts/signatures:

- RSA: $E(m) \to E(2m)$
- ElGamal Encryption: $E(m) \to E(2m)$

Can you create any forgeries for ElGamal Signatures?

5 ElGamal Signatures

Verify the following ElGamal signatures:

- $(p = 13, g = 2, h = 8)$, $m = 24, s = 6$
- $(p = 29, g = 2, h = 11)$, $m = 24, s = 2$
- $(p = 137, g = 3, h = 37)$, $m = 100, s = 42$

5.1

Pick a partner; ask them to send you a signature. Verify it.