Problem Set 8

We will work on the following exercises together during the work sessions on Friday, 6. October 2017.

You are strongly encouraged to work together on the exercises, including the homework. You do not have to hand in solutions to these problem sets.

Problem 1: The birthday attack

1. Assume that people's birthdays (the dates without the year) are independent and uniformly distributed among the dates except Febuary 29th. What is the smallest number of people such that the probability that two of them have the same birthday is larger than 99%?

2. Consider the birthday attack on a hash function $H : \{0,1\}^* \to \{0,1\}^n$, i.e. the attack by calculating $H(x)$ for $2^{n/2}$ random values of x and check for collisions. This attack uses $(m+n)2^{n/2}$ bits of memory, assuming we use m bit inputs. In this blog post, Randall Munroe estimates Googles total memory to be 15 exabytes. How do you have to choose the output length of your hash function to prevent a birthday attack by Google? Assume that Google uses $m \geq n$ and provide a length n along with a proof that it is sufficient.

Problem 2: Hash functions and short inputs

On some hash functions one can run a space-efficient birthday attack by using the fact that for the sequence $x_0 = 0^n$ and $x_{i+1} = H(x_i)$, if there exists a pair $i < j$ such that $x_i = x_j$, then there exists an index $i' < j$ such that $x_{i'} = x_{2i'}$[1].

1. Argue that there could be hash functions where this attack never succeeds.

[1] It is a nice exercise to prove that fact, but we don't ask for such a proof here.

2. Prove that any hash function H is literally collision free for inputs of length $O(\log n)$, i.e. $H(x) \neq H(x')$ for $x \neq x'$ for $|x| = O(\log n)$

Problem 3: HMAC?

Let $H : \{0,1\}^* \to \{0,1\}^n$ be a collision resistant hash function. Construct collision resistant hash functions such that the following MAC functions are insecure.

1. $\mathsf{MAC}_k(m) = H'(k\|m)$
2. $\mathsf{MAC}_k(m) = H'(k \oplus m_0\|m_1)$, where $m = m_0\|m_1$ and $|m_0| = |k|$.

Problem 4: Authenticate-then-encrypt

Given a CPA-secure encryption scheme $(\mathsf{Gen}_1, \mathsf{Enc}, \mathsf{Dec})$ and a MAC $(\mathsf{Gen}_2, \mathsf{MAC}, \mathsf{Vrfy})$, we construct an encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$. Gen' generates keys for both the encryption scheme and the MAC. The encryption is defined by $\mathsf{Enc}'_{k_1k_2}(m) = \mathsf{Enc}_{k_1}(m\|MAC_{k_2}(m))$, and the decryption $\mathsf{Dec}'_{k_1k_2}$ runs Dec_{k_1} and then Vrfy_{k_2} and outputs $\perp$ if the latter does. Show that this scheme is not always CCA-secure.

Hint: Consider the encryption Enc function that runs some CPA-secure encryption and then appends a random bit to the ciphertext.