Modern Cryptography

Bachelor Computer Science, University of Amsterdam, 2017/18

teacher: Christian Schaffner, TA: Jan Czajkowski, Christian Majenz

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 checking 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 Google's 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 to prevent an attack.

Problem 2: Hash functions and short inputs

Section 5.4.2 in [KL] describes a variant of the birthday attack that use only a small amount of memory. For a hash function with output length n, it traverses the space of n bit strings by computing a sequence $x, H(x), H(H(x)), H(H(H(x)))\ldots$ in a clever way.

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

2. Prove that any collision resistant 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. Use H to construct a collision-resistant hash functions H' such that the MAC function given as $\mathsf{MAC}_k(m) = H'(k \oplus m_0 \| m_1)$, where $m = m_0 \| m_1$ and $|m_0| = |k|$ is insecure.

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 independent keys k_1 and k_2 for both the encryption scheme and the MAC. The encryption is defined by $\mathsf{Enc}'_{k_1 k_2}(m) = \mathsf{Enc}_{k_1}(m \| MAC_{k_2}(m))$, and the decryption $\mathsf{Dec}'_{k_1 k_2}$ 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.