Institut for Matematik og Datalogi
Syddansk Universitet

September 27, 2019
JFB

Cryptology – 2019 – Lecture 8

Announcement

All lectures on Thursdays will start at 16:10, instead of 16:15.

Lecture, September 26

We covered the rest of chapter 13, except for subsection 13.5, which we will cover later.

Lecture, October 1

We will begin on chapter 14, probably covering up through section 14.4.

Lecture, October 7

We will finish chapter 14 and begin on section 11.4.

Problem session October 8

1. Let p be an odd prime and g_0 and g_1 be generators of $\mathbb{Z}_p^*$. Consider the following two functions: $f_0(x) = g_0^x \pmod{p}$ and $f_1(x) = g_1^x \pmod{p}$. Use these two functions to create a hash function which will hash an arbitrary length message down to a value in $\mathbb{Z}_p^*$. Can you make it secure under the assumption that the discrete log problem is infeasible? (See page 51 in the textbook.)

2. Consider the following proposal for a hash function, where $E(K, M)$ is encryption of the 128 bits of M using a 128-bit key K in Rijndael (AES). Let IV be a 128-bit random string. Pad the document to be

hashed such that the number of bits is divisible by 128. Let the resulting document be $M = m_1||m_2||...||m_r$, where each block m_i contains exactly 128 bits and the operation $||$ is concatenation.

$$\begin{aligned} H_0 &\leftarrow IV \\ H_1 &\leftarrow E(m_1, H_0) \\ H_2 &\leftarrow E(m_2, H_1) \\ &\vdots \\ H_r &\leftarrow E(m_r, H_{r-1}) \\ H &\leftarrow E(m_1 \oplus m_2 \oplus ... \oplus m_r, H_r) \end{aligned}$$

The operation $\oplus$ is bit-wise exclusive-or, and the output of the hash function is (H_0, H). (Note that if you use the Method 0 for padding, a message which has zeros at the end will hash to the same value as that message with the zeros truncated (removed). Thus, finding collisions is trivial, so assume one of the other methods is used.)

a. Using the Birthday Paradox, define and analyze (how many calls to Rijndael) an algorithm for finding a collision.

b. The above hash function would be much less secure if the steps with the ByteSub transformations were simply removed from Rijndael. If that was done, which is the hardest of the three problems Preimage Resistance, Second Preimage Resistance, and Collision Resistance; which could now be solved efficiently? How?

3. In ECB mode, what is the problem with the five padding schemes, if the plaintext space is very small? What about the other modes of operation?

4. I may lecture at the end.