650.445/600.454: Practical Cryptographic Systems September 12, 2018

Written HW 1

Instructor: Matthew Green *Due: 11:59 pm September 19, 2018*

Name: _____

The assignment must be completed individually. You are permitted to use the Internet and any printed references. In particular, you may find the Handbook of Applied cryptography (available online) useful for several of these questions.

Please submit the completed assignment via Blackboard.

Problem 1: Short answers. (You don't need to justify your answer, but you might want to for partial credit.)

1. What is the keyspace (total number of possible key settings) of a four-rotor Naval Enigma (M4)?
2. The Lucky13 attack relies on changing the size of the input to HMAC so that it crosses a 64-byte boundary. Why?
3. Describe the complementation property of the DES cipher.
4. What is the danger of using a "two time pad". More concretely, what can happen if I produce two equal-length ciphertexts $C_1 = M_1 \oplus K$ and $C_2 = M_2 \oplus K$ where M_1, M_2 are different messages but K is identical for both ciphertexts. Why is this bad?
5. The KRACK paper relied on forcing nonce re-use for an encryption scheme. Explain the possible implications of this for security. How does this relate to your previous answer?

Problem 2: Longer answer

1. **Double Encipherment**. Imagine I want to make a stronger version of the DES cipher that uses two 56-bit keys (thus doubling the key size to 112 bits). My approach is to simply run the DES encipherment twice in a row (in what's called a *cascade*):

$$\mathsf{SuperDESEnc}(K_1 \| K_2, M) = \mathsf{DESEnc}_{K_2}(\mathsf{DESEnc}_{K_1}(M))$$

   My hope is that a brute-force search of the keyspace for this cipher will require around 2^{112} decryption attempts (operations) in the worst case. Because we live in the real world, we'll assume that an attacker can ask any encryptor to generate the encryption of a *chosen* plaintext M under her (unknown to the attacker) keys K_1, K_2 (as many times as the attacker would like):

Describe a strategy that allows the attacker to recover the victim's keys K_1, K_2 in *significantly* fewer than 2^{112} operations. You can assume you have a very big hard disk. **Specify the worst-case number of encryption/decryption operations and total amount of storage your attack requires.**[1]

[1]Hint: your first step is to build a big table (of size 2^{56} entries). Then you'll ask the victim to encipher some chosen plaintext under her keys. Think about this for a while before you go diving for the HAC or the Internet, it will serve you well on the exam.