SE 4472a / ECE 9064
Information Security

Fall 2017
Prof. Essex

Assignment 1

Due Friday, October 6th, 2017 at 23:59

Assignments are to be completed individually. We generally expect you to make an honest effort searching around online before contacting course staff with technical questions (`stackexchange.com` and `crypto.stackexchange.com` are great resources btw).

Submission Instructions

Please use common document file formats such as `.txt`, `.pdf`, `.doc`, etc. Place your answers into separate files and/or subfolders for each of the 4 questions. Submit a `.zip` of the assignment files via OWL. The submission form in OWL will be available only up to 48 hours after the assignment due date. Email submissions are not accepted. Be sure to consult the course outline regarding the late policy.

1. [20 marks] Encryption Basics

For each of the following, state whether the given ciphertext is a valid encryption under the given cipher. Your answer should be of the form: *Yes*, *No*, or *Insufficient information*, followed by a one or two sentence justification. .

(a) [5 marks] Is `VKGORNVTGBA` a valid Enigma encryption of the plaintext `SUPERSECURE`?

(b) [5 marks] Is `ABCDEFGHIJKLMONPQR` a valid one-time pad encryption of `WESTERNENGINEERING`?

(c) [5 marks] Let `a={0xF2, 0x14, 0x72, 0xA9, 0x30, 0x4b, 0x03 }` be a vector of bytes. Is `a` a valid DES ciphertext?[1]

(d) [5 marks] Let `b={0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00}` be a vector of bytes. Is `b` a valid AES ciphertext? Why or why not?

2. [24 marks] Security Games

For each of the following, use the appropriate security game to prove the given cipher is insecure under the stated security notion.

(a) [8 marks] Prove that Enigma is not IND-EAV secure.

[1] Note: The prefix `0x...` denotes a hexidecimal value.

(b) [8 marks] Prove that AES in CBC mode is not IND-CPA secure if the IV can be predicted by the adversary.

(c) [8 marks] Prove that AES in CBC mode in not IND-CCA2 secure, even if the adversary cannot predict the IV.

Note: You don't have to be formal. The goal here is to convince us that you understand the mechanics of these security games, can come up with a strategy for the adversary to follow, and then justify why that strategy gives them a non-negligible advantage.

3. [20 marks] Encryption in Practice

For this question you will explore encryption in practice. This is a somewhat open-ended problem and you will be graded according to your ability to make security-conscious choices.

The plaintext shall consist of your full name and student number. Encrypt it using 128-bit AES in CBC mode. Use any method (*i.e.,* programming language or software program) you wish. Encode characters using UTF-8 (e.g., the character '0' is encoded as the byte `0x30`, 'A' as `0x41`, etc).

Submit the following:

(a) A text file containing the source code or commands you used. In the comments, specify your name, student number, what programming language, library, or software program you used,

(b) A text file containing the exact plaintext message (i.e., name and student number),

(c) The encryption key used, written as a list of hexidecimal bytes (e.g., `00, 01, 02, ...EE, FF`),

(d) Any initialization vector (IV) used, written as a list of hexidecimal bytes,

(e) The resulting ciphertext, written as a list of hexidecimal bytes.

4. [36 marks] Fun with Hashing

In this question we will explore the security properties of hash functions in the context of file hashing.

Step 1. Download the following assignment files. This zip file contains to (Linux) programs: `assignment1-good` and `assignment1-evil`.

Note: You do *not* need to run these programs (and you may not even be able to depending on your OS). We only use them to compute their hash values.

(a) When executed, `assignment1-good` prints:

SE 4472/ECE 9064 is a GOOD course!

(b) When executed, `assignment1-evil` prints:

SE 4472/ECE 9064 is an EVIL course. MOO HA HA!

Step 2. Use a command like tool (e.g., openssl) or library (e.g., Python's hashlib) to compute the following hashes:

- Compute the MD5 hashes of assignment1-good and assignment1-evil
- Compute the SHA-1 hashes of assignment1-good and assignment1-evil

Step 3. Submit the following:

1. [8 marks] The hashes computed in **Step 2**.
2. [28 marks] An analysis of the results from the previous step. For each of the following, answer the question in one or two sentences:

   (a) [4 marks] What do the MD5 hashes suggest about these two files?

   (b) [4 marks] What do the SHA1 hashes suggest about these two files?

   (c) [4 marks] Thinking about the answers to the previous two questions, which hash function is right? Which one is wrong? How can you tell?

   (d) [4 marks] What security property is being violated here?

   (e) [4 marks] Suggest a possible (evil) application of the ability to violate this security property?

   (f) [4 marks] On average, how many operations (*i.e.,* invocations of a hash function) would an attacker have to have to make (on average) to pull of this attack on a well-designed k-bit hash function?

   (g) [4 marks] Do you think Prof. Essex actually performed that much computation in order to create these files for this assignment? Feel free to be creative/speculative, but justify your answer.