Assignment 2

Due Friday, October 26th at 11:59:59pm

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

Place each answer in a clearly named file (*e.g.,* `q1.txt`). Answers must be in `txt`, `pdf` or `doc/docx`. Place all files, including answers and any code attachments in a `.zip` file and submit via OWL by the due date. Email submissions will not be accepted. As per the course late policy, **assignments will not be accepted more than 48 hours past the due date.**

1. [5 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) executables: `assignment-good` and `assignment-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, `assignment-good` prints:

SE 4472/ECE 9064 is a GOOD course!

(b) When executed, `assignment-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 assignment-good and assignment-evil
- Compute the SHA-1 hashes of assignment-good and assignment-evil

Step 3. Submit your answers to the following questions:

(a) [1 marks] What do the MD5 and SHA1 hashes respectively suggest about these two files?

(b) [1 marks] Thinking about the answers to the previous two questions, which hash function is right? Which one is wrong? How can you tell?

(c) [1 marks] What security property is being violated here?

(d) [1 marks] On average, how many operations (*i.e.,* invocations of a hash function) should an attacker have to have to make on average to pull of this attack on MD5 if it behaved like a random oracle?

(e) [1 marks] Thinking about your answer to the previous question, justify how you think it was possible for Prof. Essex to create `assignment-good` and `assignment-evil`.

2. [1 marks] Block Cipher Padding

Suppose you wished to encrypt the message "BLOCKCHAIN" using AES-256 with PKCS7 padding and UTF-8 encoding. Using the following UTF-8 encoding table, give the corresponding *padded* paintext in hexidecimal bytes.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A

3. [2 marks] MAC Attack

Let MAC be a message authentication code defined as shown in Figure 1 where E_k is a block cipher. MAC accepts a key k, and a message $m = m1||m2||\dots||mx$. The result is an authentication tag t.

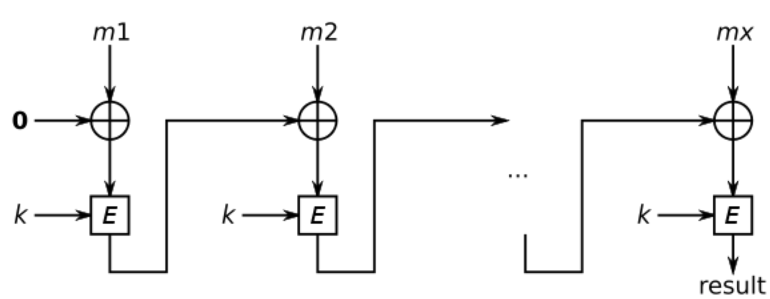

Figure 1: $t = \text{MAC}_k(m)$

(a) [1 marks] Suppose Eve obtains two valid MACs m_1, t_1 and m_2, t_2. Show how Eve could produce a third valid MAC m_3, t_3 without knowing the key.

(b) [1 marks] Without requiring any additional information (e.g., extra keys, etc), suggest a way to make MAC secure against this attack.

4. [2 marks] AES-GCM FTW

Prove that AES-GCM is secure against an adaptive chosen-ciphertext attack, i.e., is IND-CCA2 secure. Assume that:

- Alice may choose any messages m_0, m_1 she wishes to submit to Bob, so long as $m_0 \neq m_1$ and $|m_0| = |m_1|$,
- Alice may make encryption *and* decryption queries to Bob, both before and after she receives the challenge ciphertext from Bob,
- Bob will not decrypt the challenge ciphertext, or any other ciphertext with an invalid MAC tag.
- Alice cannot forge AES-GCM MAC tags in less than brute force.