C&O 487/687: Assignment #2

Due date: October 10, 2024 (Thursday)

Note: Some questions use randomization to customize to you specifically. Please include your max-8-character UW user id (`b54khan`) at the beginning of your answer so we can look up your custom solution.

1. [10 marks] **Linear cryptanalysis**

   Please include your max-8-character UW user id (`b54khan`) at the beginning of your answer so we can look up your custom solution.

   This problem uses the simplified cipher described in Section 2 of "A Tutorial on Linear and Differential Cryptanalysis" by Howard M. Heys, available at
   `http://www.engr.mun.ca/~howard/Research/Papers/ldc_tutorial.html`
   We refer to this cipher as the "Heys cipher".

   For the purposes of this problem, each student has a fixed, unknown 80-bit key. You will be carrying out a known-plaintext attack against the Heys cipher using linear cryptanalysis, using a set of 20,000 distinct random plaintext-ciphertext pairs. You can download your plaintexts and ciphertexts (unique to you) at the following addresses:
   `https://www.math.uwaterloo.ca/~dstebila/as/as.cgi/download/co487_f24/a2q1plaintexts.txt`
   `https://www.math.uwaterloo.ca/~dstebila/as/as.cgi/download/co487_f24/a2q1ciphertexts.txt`

   The format of the files is that the nth line of the ciphertext file equals the encryption of the nth line of the plaintext file under your secret key.

   For the programming aspects of questions 1.a.i, 1.b, and 1.d.ii, you may work with a partner to do the programming and you may submit the same computer program source code, but you must submit your own write-up and explanations for the non-programming parts of those questions. Please indicate in your submission who you worked with.

   (a) [2 marks] Using the linear approximation $U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 \cong 0$, Carol guesses that the target partial subkey $[K_{5,5}, K_{5,6}, K_{5,7}, K_{5,8}, K_{5,13}, K_{5,14}, K_{5,15}, K_{5,16}]$ has the value $[0, 1, 1, 1, 0, 1, 1, 0]$. Note that Carol's guess is not necessarily correct! Do one of the following:

      i. Determine the magnitude of the bias for this partial subkey value over your twenty thousand plaintext/ciphertext pairs, using a computer program, and provide the source code for your program, or:

      ii. Determine the magnitude of the bias for this partial subkey value over your first *ten* plaintext/ciphertext pairs, without using a computer program, and show your work.

   (b) [3 marks] Find the value of the partial subkey $[K_{5,5}, K_{5,6}, K_{5,7}, K_{5,8}, K_{5,13}, K_{5,14}, K_{5,15}, K_{5,16}]$ for your key, by calculating the target partial subkey which yields the largest magnitude of bias over your 20,000 plaintext/ciphertext pairs. You will almost certainly need a computer program for this task; provide the source listing for any computer code that you or your collaborators write.

   (c) [2 marks] By using Table 4 in the tutorial, compute the bias in each of the following individual S-box linear approximations:

   $$S_{11} : X_1 \oplus X_4 \cong Y_1$$
   $$S_{13} : X_1 \oplus X_4 \cong Y_1$$
   $$S_{21} : X_1 \oplus X_3 \cong Y_2$$
   $$S_{32} : X_1 \cong Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4$$

   Then, combine these to find a linear approximation of the first three rounds of the Heys cipher, and calculate the theoretical magnitude of its bias.

(d) [3 marks] Do *one* of the following:

i. Using the linear approximation from part (c), determine the entire subkey K_5. You will almost certainly need a computer program for this task; provide the source listing for any computer code that you or your collaborators write, or:

ii. Using the (incorrect) guess $K_5 = [1100011111100110]$ for the fifth subkey, determine the magnitude of the bias of the approximation from part (c) for this subkey over your first ten plaintext/ciphertext pairs, without using a computer program, and show your work.

(e) [2 bonus marks] A small amount of extra credit is available if you can determine any additional key bits.

2. [12 marks] **Penguin or not a penguin? (Block cipher modes of operation)**

This question is meant to illustrate the importance of proper implementation. The Jupyter notebook `Penguins.ipynb` (available on LEARN) contains Python implementations of four different block cipher modes of operation: ECB, CBC, OFB, and CTR. Some have been implemented correctly, and some have not. You have also been provided with four encrypted image files, which you can download from LEARN:

- `image1.ppm`, which was encrypted using ECB mode
- `image2.ppm`, which was encrypted using CBC mode and the IV specified in the Jupyter notebook
- `image3.ppm`, which was encrypted using OFB mode and the IV specified in the Jupyter notebook
- `image4.ppm`, which was encrypted using CTR mode and the IV specified in the Jupyter notebook

Some of the above files are the encryptions of images of penguins, and some are not. By analyzing the encrypted images and the code in `Penguins.ipynb` that was used to encrypt them, your task is to determine which of the images are penguins, and which are not.

For each image/mode, you should explain:

- Whether there is a flaw in the implementation of the mode of operation, and what the flaw is.
- Whether the plaintext image is:
  (a) DEFINITELY A PENGUIN
  (b) DEFINITELY NOT A PENGUIN
  (c) CANNOT BE CERTAIN WHETHER OR NOT IT IS A PENGUIN

In each case, you must fully justify your answer. (Note: "Because it looks like a penguin" is not a fully justified answer. However this doesn't mean you have to fully decrypt every image.)

3. [3 marks] **OFB mode**

We generally assume that the adversary does not know the secret key, but does know the initialization vector. Suppose that, for OFB mode, we kept the IV secret as well. Show that this does not make an exhaustive key search any more expensive – in other words, show how to perform a brute-force attack on OFB with an unknown key and an unknown IV. What requirements have you made on the plaintext and ciphertext?

4. [7 marks] **Hash function properties**

Trits are similar to bits but have three possible values $\{0, 1, 2\}$ instead of two. As such, they can be seen as elements in $\mathbb{Z}_3$ – integers modulo 3. They are sometimes used in algorithms where the objects we are working with have three possible states, such as walks on 3-regular graphs.

For two strings of trits with same length $(t_1, \ldots, t_n)$ and $(t'_1, \ldots, t'_n)$, we can define the subtraction operation as:

$$(t_1, \ldots, t_n) - (t'_1, \ldots, t'_n) := (t_1 - t'_1, \ldots, t_n - t'_n)$$

where each tritwise operation is done modulo 3.

Let $f : \{0,1,2\}^m \to \{0,1,2\}^m$ be a preimage-resistant bijection.

For $x \in \{0,1,2\}^{2m}$ write $x = x' \| x''$ (in other words, split the $2m$-trit string x into two m-trit halves x' and x''). Define a new hash function $H : \{0,1,2\}^{2m} \to \{0,1,2\}^m$ as

$$H(x) = f(x' - x'')$$

(a) [6 marks] For sufficiently large values of m (e.g., $m = 256$), does H have each of the following desired properties for a cryptographic hash function? If yes, justify your answer with a contrapositive argument. If no, justify your answer by showing how to break the property.

i. Preimage resistance

ii. Second preimage resistance

iii. Collision resistance

(b) [1 mark] Suppose $m = 128$, so that the input space for H is $\{0,1,2\}^{256}$ and the output space is $\{0,1,2\}^{128}$. If you sample the inputs uniformly at random, what is the expected number of steps before you find a collision in H? It is okay to use an approximation here, as long you state, and justify that approximation.

5. [7 marks] **Sponge construction**

The sponge construction is a key part in multiple cryptographic schemes, most notably the SHA-3 hash function. The construction has the following properties:

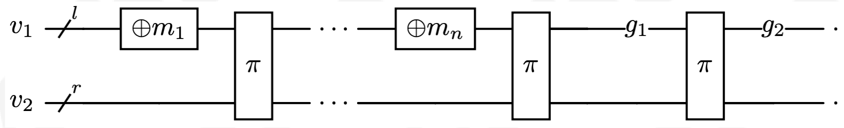
- A message m of length $n\ell$ is separated into n blocks $(m_1, \ldots, m_n)$ each of length ℓ.
- The initialization vector v is of length $\ell + r$, and is separated into a two potions $v = v_1 \| v_2$, where v_1 is ℓ bits in length and v_2 is r bits in length.
- $\pi : \{0,1\}^{\ell+r} \to \{0,1\}^{\ell+r}$ is a hard-to-invert permutation.

Here is the pseudocode for a sponge construction based on π that takes an input message m of length $n\ell$, and produces an output g of length $s\ell$.

```
H(m)
$h_1 \| h_2 \leftarrow v_1 \| v_2$
for $i = 1, \ldots, n$ :
    $h_1 \leftarrow h_1 \oplus m_i$
    $h_1 \| h_2 \leftarrow \pi(h_1 \| h_2)$
for $j = 1, \ldots, s$ :
    $g_j \leftarrow h_1$
    $h_1 \| h_2 \leftarrow \pi(h_1 \| h_2)$
return $g_1 \| g_2 \| \ldots \| g_s$
```

The first loop is called the *absorbing phase* and the second is the *squeezing phase.*

Here is a diagram showing the operation of the sponge construction:



For the sponge construction to be a secure cryptographic hash function, we need that the permutation π is hard to invert.

Note that usually the permutation π is a permutation on the entire space of all $2^{\ell+r}$ bitstrings, not a permutation on the $\ell + r$ positions of the bitstrings. This question will explore what goes wrong if that is not the case.

Suppose that π was simply the function shifting each bit one position to the right (and looping around), so that π maps $(b_1 b_2 \dots b_n)$ to $(b_n b_1 \dots b_{n-1})$). Your task is to justify why using this π would make H insecure.

(a) Prove that, if π is the shifting function as described above, then the sponge construction is not preimage resistant. For your attack, you can fix a number of input blocks $n \geq 1$ and a number of squeezing rounds $s \geq 1$ that makes it easier.

(b) Prove that, if π is the shifting function as described above, then the sponge construction is not second preimage resistant. Recall that the second preimage does not necessarily need to be of the same length.

(c) Prove that, if π is the shifting function as described above, then the sponge construction is not collision resistant.

Academic integrity rules

You should make an effort to solve all the problems on your own. You are also welcome to collaborate on assignments with other students in this course. However, solutions must be written up by yourself. If you do collaborate, please acknowledge your collaborators in the write-up for each problem. *If you obtain a solution with help from a book, paper, a website, or any other source, please acknowledge your source. You are not permitted to solicit help from other online bulletin boards, chat groups, newsgroups, or solutions from previous offerings of the course.*

Due date

The assignment is due via Crowdmark by 11:59:59pm on October 10, 2024.

If you are requesting an extension or accommodation because of a verification of illness form or a self-declared student absence, see the "Missed assignment policy and forms" instructions on the CO 487/687 LEARN site.

Changelog

- Mon. Sep. 30: Fix wording about permutation π in A-2-5 parts (b) and (c).
- Thu. Oct. 3: Fix a typo in Q4.