Cryptanalysis 2021 Homework 1

Ling Song, Hosein Hadipour

March 10, 2021

Question 1.

This concerns using memory in favor of speed in implementing the encryption algorithms. Let S be the AES S-box. Let MC be the mix-column, i.e., it takes as input a column consisting of 4 bytes and outputs such a column. Note that MC is linear, i.e., we have for any two columns C_1, C_2 that:

$$MC(C_1 \oplus C_2) = MC(C_1) \oplus MC(C_2).$$

Define a function T_0 as follows: it takes as input a byte b, and the output $T_0(b)$ is a 4-byte column computed as follows: you first form a column $C(b)$ by placing $S(b)$ in the top byte and all-0 bytes in the lower three positions. Then set $T_0(b) = MC(C(b))$. We also define functions T_1, T_2, T_3. They are similar to T_0, except that when we form $C(b)$, we place $S(b)$ in the second, third and fourth entry from the top respectively, and put in 0's elsewhere.

Now consider the state of AES encryption algorithm at the start of some round. Name the bytes in this state a_{ij} as in ? and let R be the state after we have done `SubBytes`, `ShiftRow` and `MixColumn`. So R is a 4 by 4 matrix of bytes.

Show that the first column of R is

$$T_0(a_{00}) \oplus T_1(a_{11}) \oplus T_2(a_{22}) \oplus T_3(a_{33}).$$

Give similar expressions for the other 3 columns of R.

Sketch how this result can be used to implement AES based only on table look-up and XOR, instead of explicitly computing the operations. How much memory would you need for this?

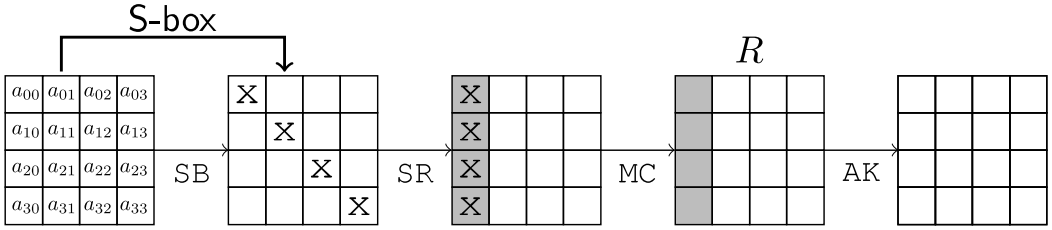

Figure 1: A Round of AES

Question 2.

Consider the following structure where ENC and DEC represent the encryption and decryption via the `DES` algorithm with 56-bit key respectively. Note that, the first and last encryption blocks use the same 56-bit key k_1, whereas the middle one utilizes k_2 which is not necessarily the same as k_1. Does it provide the 112-bit security level? If not so, provide a cryptographic attack with time complexity of strictly less than 2^{112} `DES` encryptions. Please explain what model is your attack

classified in (known-plaintext, chosen-plaintext, ...). Besides, the amount of **time** and **memory** in your attack should be specified in detail.

A known plaintext attack has more points in comparison to a chosen plaintext attack.

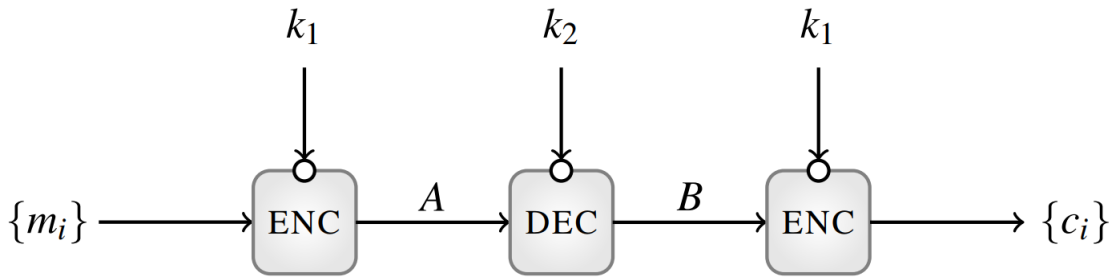

Figure 2: Two-Key Triple Encryption

Question 3.

It is usually much easier to find matching objects than it is to find a particular object!

(a) What is the probability that someone has the same birthday as you?

(b) What is the probability that at least two people share the same birthday?

(c) In a group of 23 strangers, what is the probability that at least two of them have the same birthday? How about if there are 40 strangers?

(d) In a group of 200 strangers, what is the probability that one of them has the same birthday as your birthday?

Question 4.

An urn contains N balls, of which n are red and $N-n$ are blue. Bob randomly selects a ball from the urn, replaces it in the urn, randomly selects a second ball, replaces it, and so on. He does this until he has looked at a total of m balls.

(a) Prove that Bob selects at least one red ball with the following probability:

$$\Pr(\text{at least one red}) = 1 - \left(1 - \frac{n}{N}\right)^m.$$

(b) Prove that a lower bound for the above probability is:

$$\Pr(\text{at least one red}) \geq 1 - e^{-\frac{m \cdot n}{N}}.$$

(c) Prove that if N is large and if m and n are not too much larger than $\sqrt{N}$ (e.g., $m, n \leq 10\sqrt{N}$), then the two sides of the above inequality are almost the same.

(d) Bob has a box that contains N numbers. he chooses n distinct numbers from the box and puts them in a list. he then makes a second list by choosing m (not necessarily distinct) numbers from the box. Prove that if n and m are each slightly larger than $\sqrt{N}$, then it is very likely ($\Pr \geq 0.5$) that the two lists contain a common elements.

Hint: In the previous question, assume that all balls are blue at first. Then, Bob selects n balls one at a time to construct the first list. Next, he repaints those n balls with the red paint and return them to the box. The second list is constructed by drawing m balls out of the urn one after another, noting its number and color, and then replacing it.

(e) Assume that $H : \{0,1\} \rightarrow \{0,1\}^n$ is a hash function, where $n \in \mathbb{N}$. Prove that, regardless of the internal structure of H, a collision can be found with high probability in $\mathcal{O}(2^{\frac{n}{2}})$ steps, where $\mathcal{O}(2^{\frac{n}{2}})$ words of memory are used.

Question 5.

Assume that f is a function from $S = \{1, \ldots, N\}$ to itself where $N \in \mathbb{N}$. In addition, f is easy to evaluate but hard to invert. We are given a value $y \in S$ and are asked to find its preimage x under f, i.e., $y = f(x)$. Let $x_1^0, \ldots, x_m^0$ be some elements of S. Build the following array of chains:

$$
\begin{array}{ll}
x_1^0 \longrightarrow x_1^1 = f(x_1^0) \longrightarrow & \cdots \longrightarrow x_1^t = f(x_1^{t-1}), \\
x_2^0 \longrightarrow x_2^1 = f(x_1^0) \longrightarrow & \cdots \longrightarrow x_1^t = f(x_2^{t-1}), \\
\cdots & \cdots \\
x_m^0 \longrightarrow x_m^1 = f(x_1^0) \longrightarrow & \cdots \longrightarrow x_m^t = f(x_m^{t-1}),
\end{array}
$$

where $t \in \mathbb{N}$.

(a) Given a $y \in S$, how would you check the existence of x (such that $f(x) = y$) among the first t columns of the above array if you are merely given the first and the last columns of the above array?

(b) Assuming that all elements inside the 0'th through $t-1$'st columns of the above array are different, compute the probability of appearing x among the first t columns.

(c) By the birthday paradox, prove that if we add new a new row to the above array, the additional row have likely no common point with the previous ones as long as $t \cdot mt \leq N$. Hence, choosing m and t such that $mt^2 > N$ will not be a good choice.

(d) If we choose m and t such that the relation $mt^2 = N$ is satisfied, then a single array like above covers only a fraction $\frac{mt}{N} = \frac{1}{t}$ of S. On the other hand, constructing additional tables using the same strategy as above, increases the probability of reappearing the same points in different tables which causes us to waste a lot of space! So, what is your solution to cover the whole S?

Question 6.

This concerns a trick that is very useful to find a cycle in a sequence of iterated function values. Let S be any finite set, f be any function from S to itself, and x_0 be any element of S. For any $i > 0$, let $x_i = f(x_{i-1})$. Let μ be the smallest index such that the value x_μ reappears infinitely often within the sequence of values x_i, and let λ be the smallest positive integer such that $x_\mu = x_{\lambda+\mu}$.

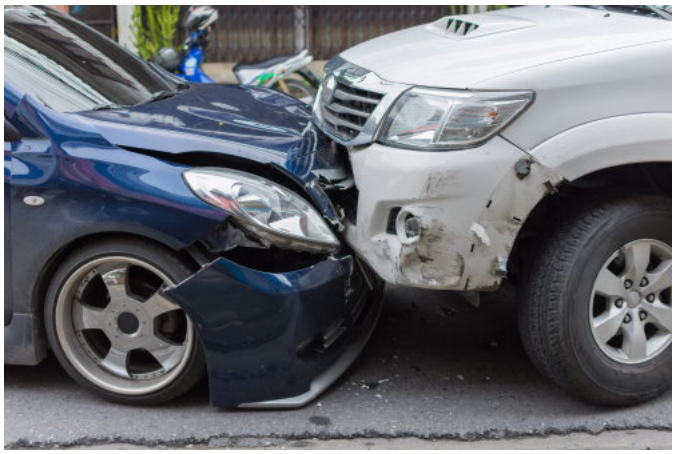

Figure 3: Collisions for Truncated `SHA3`

(a) Prove that $i = k\lambda \geq \mu$ for some k if and only if $x_i = x_{2i}$.

(b) Based on the above fact, propose an algorithm to find μ and λ, given f and x_0.

(c) Using the proposed algorithm in the previous part, find a 64-bit collision for `SHA3-512`.

Hint: Study about the cycle detection algorithms in `https://en.wikipedia.org/wiki/Cycle_detection`.

To compute the `SHA3-512` using the `Python` language you can use the following commands:

```
In [1]: import hashlib
In [2]: st = "Hello World!"
In [3]: digest = hashlib.sha3_512(st.encode())
In [4]: digest.hexdigest()
Out[4]: '32400b5e89822de254e8d5d94252c52bdcb27a3562ca593e980364d9848b8041
b98eabe16c1a6797484941d2376864a1b0e248b0f7af8b1555a778c336a5bf48'
```