Introduction to Cryptography

TEK 4500 (Fall 2023)
Problem Set 2

Problem 1.

Read Chapter 3 and Chapter 4 (Sections 4.8–4.10 can be skipped) in [BR].

Problem 2.

Before AES was standardized in 2001, one of the most popular block ciphers was the Data Encryption Standard. Unlike AES, DES only have a key size of 56 bits and a block size of 64 bits.

a) Suppose you have access to a really powerful computer which runs at 10 GHz and is capable of performing a full DES encryption on a single clock cycle. How long would it take to brute-force a 56 bit DES key using this computer? Assume you have a number of known plaintext-ciphertext pairs available.

b) Technically, DES actually takes a 64 bit key, it's just that it ignores[1] every eight bit. Thus the key is effectively only 56 bit. Now suppose *all* of the key bits were used by DES. How long would the attack take now?

c) After an upgrade, the computer is modified to instead perform a single AES-128 encryption on every clock cycle. How long would it take to brute force an AES-128 key using this computer? Give your answer in years.

d) How old is the universe?

e) How many of these machines would you have to use to brute-force the AES-128 key within one year? The average cost of electricity in Norway in 2020 was 20.7 øre per kWh. Suppose one machine uses about 4000 kWh annually. What would it cost to run your brute-force attack? Compare with the world's yearly gross product.

[1] *Actually*, these bits are meant to be used for parity-checking of the key, but for security purposes this is equivalent to simply ignoring them.

$\underline{\mathbf{Exp}^{\mathsf{prf}}_{F}(\mathcal{A})}$

$b \leftarrow \{0,1\}$
$K \xleftarrow{\$} \mathcal{K}$
$F_0 \leftarrow F_K$
$F_1 \xleftarrow{\$} \mathsf{Func}[m,n]$
$b' \leftarrow A^{F_b(\cdot)}$
return $b' \stackrel{?}{=} b$

$$\mathbf{Adv}^{\mathsf{prf}}_{F}(\mathcal{A}) = \left| 2 \cdot \Pr[\mathbf{Exp}^{\mathsf{prf}}_{F}(\mathcal{A}) \Rightarrow \mathsf{true}] - 1 \right|$$

Figure 1: PRF security experiment for a function $F\colon \mathcal{K} \times \{0,1\}^m \to \{0,1\}^n$.

Problem 3. [Problem 6.8 in [Ros]]

Suppose $F\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$ is a secure PRF with input length m and output length n. We want to use F to construct another PRF G that has a longer *input* length than F. Below are some approaches that don't work. For each suggestion, describe a successful distinguishing attack and compute its PRF-advantage. That is, compute $\mathbf{Adv}^{\mathsf{prf}}_{G}(\mathcal{A})$, where $\mathcal{A}$ is the adversary that runs your attack. Below "$\|$" denotes string concatenation, e.g. $101\|01 = 10101$.

a) $G_K(X) = F_K(X')\|F_K(X'')$, where $X = X'\|X''$ and $X', X'' \in \{0,1\}^m$.

That is, G splits its input $X \in \{0,1\}^{2m}$ into two halves X', X'', applies F to each half separately, and concatenates the result. Note that G is a PRF of the form $G\colon \{0,1\}^k \times \{0,1\}^{2m} \to \{0,1\}^{2n}$, i.e., it has the twice the input length and twice the output length of F.

b) $G_K(X) = F_K(X') \oplus F_K(X'')$, where $X = X'\|X''$ and $X', X'' \in \{0,1\}^m$.

Note that G is of the form $G\colon \{0,1\}^k \times \{0,1\}^{2m} \to \{0,1\}^n$.

c) $G_K(X) = F_K(X') \oplus F_K(X' \oplus X'')$,where $X = X'\|X''$ and $X', X'' \in \{0,1\}^m$.

Note that G is of the form $G\colon \{0,1\}^k \times \{0,1\}^{2m} \to \{0,1\}^n$.

d) $G_K(X) = F_K(0\|X') \oplus F_K(1\|X'')$, where $X = X'\|X''$ and $X', X'' \in \{0,1\}^{m-1}$.

Note that G is of the form $G\colon \{0,1\}^k \times \{0,1\}^{2(m-1)} \to \{0,1\}^n$.

Problem 4.

Suppose $F : \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$ is a secure PRF. For each of the following constructions of a *new* PRF from F, decide whether it is also a secure PRF. If you think it's not, describe an attack, else, indicate why the new construction is also secure.

a) $G_K(X) = \begin{cases} 0^{128}, & \text{if } K = 0^{128} \\ F_K(X), & \text{otherwise} \end{cases}$

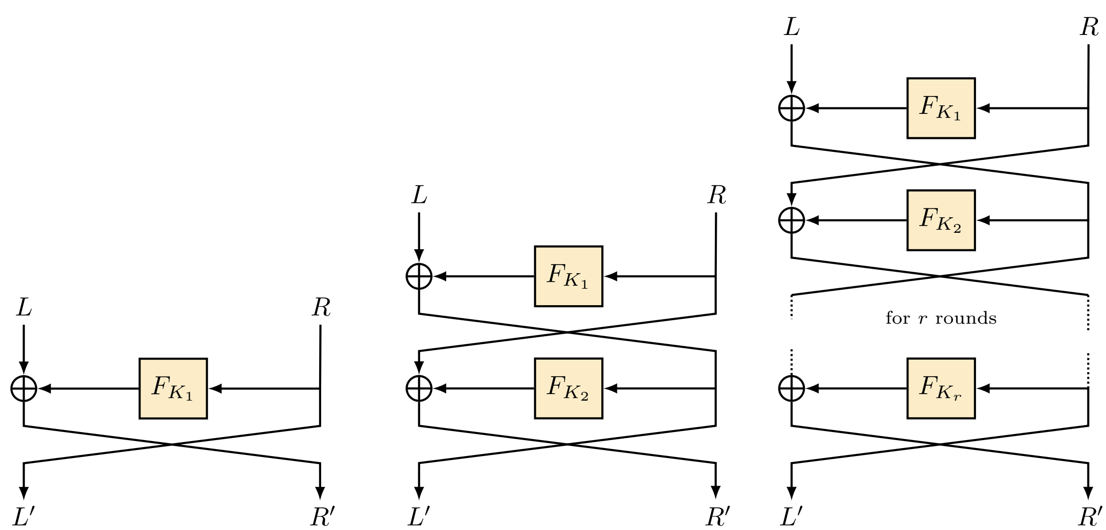
b) $G_K(X) = \begin{cases} 0^{128}, & \text{if } X = 0^{128} \\ F_K(X), & \text{otherwise} \end{cases}$

c) $G_K(X) = F_K(X) \oplus 1^{128}$

d) $G_K(X) = F_K(X) \oplus C$, where $C \in \{0,1\}^{128}$ is a *fixed* and *public* (and thus known to the adversary) hard-coded string of some arbitrary value.

Problem 5.

In this problem we'll look at a way of turning *PRFs* into *PRPs*. The construction is called a *Feistel network* (after Horst Feistel) and is shown in Fig. 2. In detail, a Feistel network converts a PRF $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ into a PRP $E : \{0,1\}^{r \cdot k} \times \{0,1\}^{2n} \to \{0,1\}^{2n}$, having double the block length and r times longer key, by applying F in r rounds where the first k bits of the key are used in the first round, the next k bits are used in the second round, and so on.



(**a**) One-round. (**b**) Two-rounds. (**c**) r rounds.

Figure 2: Feistel network.

a) Show that the Feistel network turns *any* PRF F into a PRP E. That is, for all keys $K \in \{0,1\}^{r \cdot k}$, show that the function $E_K \colon \{0,1\}^{2n} \to \{0,1\}^{2n}$ is *invertible*.

b) Let $E^{(1)} : \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$ denote the block cipher defined by the one-round Feistel network shown in Fig. 2a, where $F : \{0,1\}^{128} \times \{0,1\}^{64} \to \{0,1\}^{64}$ is the internal round function. Show that $E^{(1)}$ is *not* a secure PRF by demonstrating an attack. What is the PRF-advantage of your attack? That is, what is $\mathbf{Adv}^{\mathsf{prf}}_{E^{(1)}}(\mathcal{A})$, where $\mathcal{A}$ is the adversary that runs your attack?

Hint: What is $E^{(1)}(K_1, 0^{128})$?

c) Let $E^{(2)} : \{0,1\}^{256} \times \{0,1\}^{128} \to \{0,1\}^{128}$ denote the block cipher defined by the two-round Feistel network shown in Fig. 2b. Show that $E^{(2)}$ is *not* a secure PRF by demonstrating an attack. What is the PRF-advantage of your attack?

Hint: it is possible to obtain a very high PRF-advantage by making two oracle queries in the PRF experiment $\mathbf{Exp}^{\mathsf{prf}}_{E^{(2)}}(\mathcal{A})$.

d) **Bonus:** What can you say about the 3-round Feistel network $E^{(3)}$?

Problem 6. (*Hard*):

The DES block cipher introduced in Problem 2 is based on the Feistel network. Suppose DES was only using a *single* round and suppose you have access to two plaintext-ciphertext pairs (X, Y), (X', Y') (in particular, $X = L_0 \| R_0$ and $Y = L_1 \| R_1$; $X' = L'_0 \| R'_0$ and $Y' = L'_1 \| R'_1$). Explain how you can recover the *key* $K \in \{0,1\}^{48}$ of this one-round version of DES. For simplicity, assume that K is used directly in the round function F without any key expansion first.

Hint 1: Unlike in Problem 5b), you should now exploit the *concrete* round function $F : \{0,1\}^{48} \times \{0,1\}^{32} \to \{0,1\}^{32}$ used inside DES. The following amount of detail about the DES round function is sufficient to answer this question (refer to Fig. 3):

- E expands 32 bits to 48 bits by copying the 32 input bits to 32 different positions in the output, and then duplicating certain bits of the input in the remaining 16 positions of the output.
- $S_1, \ldots, S_8$ are the DES S-boxes. By design, each S-box is a 4-to-1 function[2] that maps 6 bits to 4 bits.
- P shuffles the 32 input bits around.

[2] That is, for any output B of an S-box S_i there are exactly *four* inputs A_1, A_2, A_3, A_4 such that $S_i[A_j] = B$.

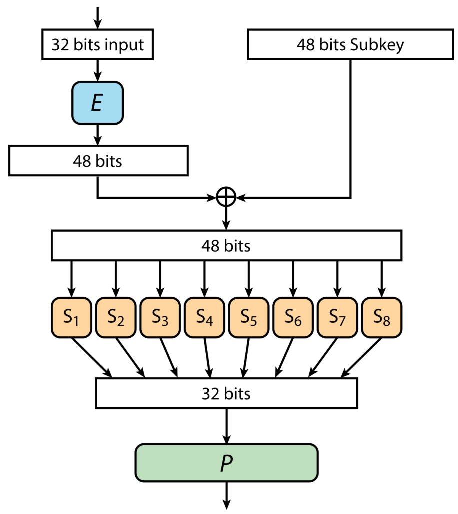


Figure 3: DES round function F.

Hint 2: From the equation $\mathrm{DES}^{(1)}(K, X) = Y$ work your way forward from $X = L_0 \| R_1$ till the input of the S-boxes, and backwards from $Y = L_1 \| R_1$ till the output of the S-boxes. For each S-box, what is the relationship between the input and the output?

Hint 3: Some trial-and-error of candidate keys is necessary. However, it should be possible to obtain K by trying about $4^8 = 2^{16}$ candidate keys. Notice that this is much less than the possibly 2^{48} keys you would have to try by brute-force.

Problem 7.

A crucial component of round functions used in DES and AES is their S-boxes. The S-boxes are the only non-linear parts of DES and AES. Recall that a function F is linear if $F(A + B) = F(A) + F(B)$ for all inputs A, B. In this exercise you are asked to validate that the first S-box of DES, S_1, is indeed non-linear by computing the output values for a set of input values. In particular, show that $S_1(X_1) \oplus S_1(X_2) \neq S_1(X_1 \oplus X_2)$ for:

a) $X_1 = 000000, X_2 = 000001$

b) $X_1 = 111111, X_2 = 100000$

c) $X_1 = 101010, X_2 = 010101$

The definition of the S_1 S-box can be found here.

Extra: Write a script (e.g. in Python) that checks whether S_1 is non-linear for *all* inputs. Do the same for the other DES S-boxes and the AES S-box. Values for the DES and AES S-boxes can be found online, e.g., here (DES) and here (AES).

References

[BR] Mihir Bellare and Phillip Rogaway. *Introduction to Modern Cryptography*. https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf.

[PP] Christof Paar and Jan Pelzl. *Understanding Cryptography - A Textbook for Students and Practitioners*. Springer, 2010.

[Ros] Mike Rosulek. *The Joy of Cryptography*, (draft Feb 6, 2020). https://web.engr.oregonstate.edu/~rosulekm/crypto/crypto.pdf.