



TEK4500 – Introduction to Cryptography

Problem Set 2

Fall 2026

Problem 1

Read Chapter 4, 6 and 7 in [Ros] (Sections 4.3–4.5, 6.4–6.5 and 7.3–7.4 can be skipped).

Problem 2

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

- a) Suppose you have access to a really powerful computer that runs at 10 GHz and is capable of performing a full DES encryption in 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 ignores¹ every eight bit. Thus the key is effectively only 56 bits. 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 *one million* AES-128 encryptions 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.² Suppose one machine uses about 2000 kWh annually. What would it cost to run your brute-force attack? Compare with the world's yearly [GDP](#).

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

²<https://www.ssb.no/en/energi-og-industri/artikler-og-publikasjoner/very-low-electricity-price-in-2020>

$\text{Exp}_F^{\text{prf}}(\mathcal{A})$	$\mathcal{O}(X)$
1: $b \leftarrow \{0, 1\}$	1: if $b = 0$ then
2: $K \leftarrow \mathcal{K}$	2: return $F_K(X)$
3: $T \leftarrow$ empty dictionary	3: else
4: for $X \in \{0, 1\}^m$ do	4: return $T[X]$
5: $T[X] \leftarrow \{0, 1\}^n$	
6: $b' \leftarrow A^{\mathcal{O}(\cdot)}$	
7: return $b' \stackrel{?}{=} b$	
$\text{Adv}_F^{\text{prf}}(\mathcal{A}) = 2 \cdot \Pr[b' = b] - 1/2 $	

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

Problem 3

Suppose $F: \{0, 1\}^k \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ 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 $\text{Adv}_G^{\text{prf}}(\mathcal{A})$, where $\mathcal{A}$ is the adversary that runs your attack. Below " $\parallel$ " denotes string concatenation, e.g. $101\parallel 01 = 10101$.

- a) $G_K(X) = F_K(X') \parallel F_K(X'')$, where $X = X' \parallel X''$ 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 $G: \{0, 1\}^k \times \{0, 1\}^{2m} \rightarrow \{0, 1\}^{2n}$, 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 $X = X' \parallel X''$ and $X', X'' \in \{0, 1\}^m$.

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

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

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

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

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

Problem 4

This problem will provide an equivalent definition of PRF advantage that may be simpler to work with than the one we used in class. Let “ $\mathcal{A}^{\mathcal{O}_b(\cdot)} \Rightarrow b'$ ” be shorthand for the event that $\mathcal{A}$ outputs b' in the PRF experiment in Fig. 1 when the secret bit is b . Specifically, $\mathcal{A}^{\mathcal{O}_0(\cdot)}$ means that the adversary gets back $F_K(X)$ when it queries X to its oracle, while $\mathcal{A}^{\mathcal{O}_1(\cdot)}$ means that it gets back $T[X]$.

- a) Show that the definition of PRF advantage given in Fig. 1 is equivalent to the right hand side in the equation below.

$$\text{Adv}_F^{\text{prf}}(\mathcal{A}) = \left| \Pr[\mathcal{A}^{\mathcal{O}_0(\cdot)} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathcal{O}_1(\cdot)} \Rightarrow 0] \right| \quad (1)$$

In other words, $\mathcal{A}$'s PRF advantage can equivalently be measured as the *difference* in probability between $\mathcal{A}$ outputting 0 when it's interacting with the real PRF function ($b = 0$) and (wrongly) outputting 0 when it's interacting with a completely random function ($b = 1$).³

Hint: Expand $\Pr[b' = b]$ using the law of total probability, then take the complement of the case $\Pr[b' = 1 \mid b = 1]$.

- b) Calculate the PRF advantage of your attack against G in Problem 3a) using the alternative definition above.
- c) The usefulness of the alternative PRF advantage definition is that if we want to *upper bound* the PRF advantage of some adversary $\mathcal{A}$, we can simply *upper bound* the probabilities $\Pr[\mathcal{A}^{\mathcal{O}_0(\cdot)} \Rightarrow 0]$ and $\Pr[\mathcal{A}^{\mathcal{O}_1(\cdot)} \Rightarrow 0]$ directly. That is, show that if $\Pr[\mathcal{A}^{\mathcal{O}_0(\cdot)} \Rightarrow 0] < \epsilon_0$ and $\Pr[\mathcal{A}^{\mathcal{O}_1(\cdot)} \Rightarrow 0] < \epsilon_1$, then

$$\text{Adv}_F^{\text{prf}}(\mathcal{A}) < \epsilon_0 + \epsilon_1 \quad (2)$$

Hint: Prove the more general statement: if $\Pr[A] < \epsilon_0$ and $\Pr[B] < \epsilon_1$, then

$$|\Pr[A] - \Pr[B]| < \epsilon_0 + \epsilon_1$$

Hint: $|\Pr[A] - \Pr[B]| = |\Pr[A] - 0 + 0 - \Pr[B]|$.

Hint: [Triangle inequality](#).

³Note that focusing on the difference between $\mathcal{A}$ outputting a 0 in the two “worlds” is just a convention. We could equally have focused on the difference between $\mathcal{A}$ outputting a 1, i.e., defined the advantage as $|\Pr[\mathcal{A}^{\mathcal{O}_0(\cdot)} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathcal{O}_1(\cdot)} \Rightarrow 1]|$ Bonus exercise: prove that this is equivalent!

Problem 5

Suppose $F : \{0, 1\}^{128} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ 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 6

In this problem we'll look at a way of turning a *PRF* into a *PRP*. That is, how can we make an *invertible* function from a (possibly) non-invertible function? The construction is called a *Feistel network* (after Horst Feistel) and is shown in Fig. 2. In detail, a Feistel network converts a PRF $F : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ into a PRP $E : \{0, 1\}^{r \cdot k} \times \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$, 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. Refer to Sections 7.3 and 7.4 in [Ros] for more background on the Feistel construction.

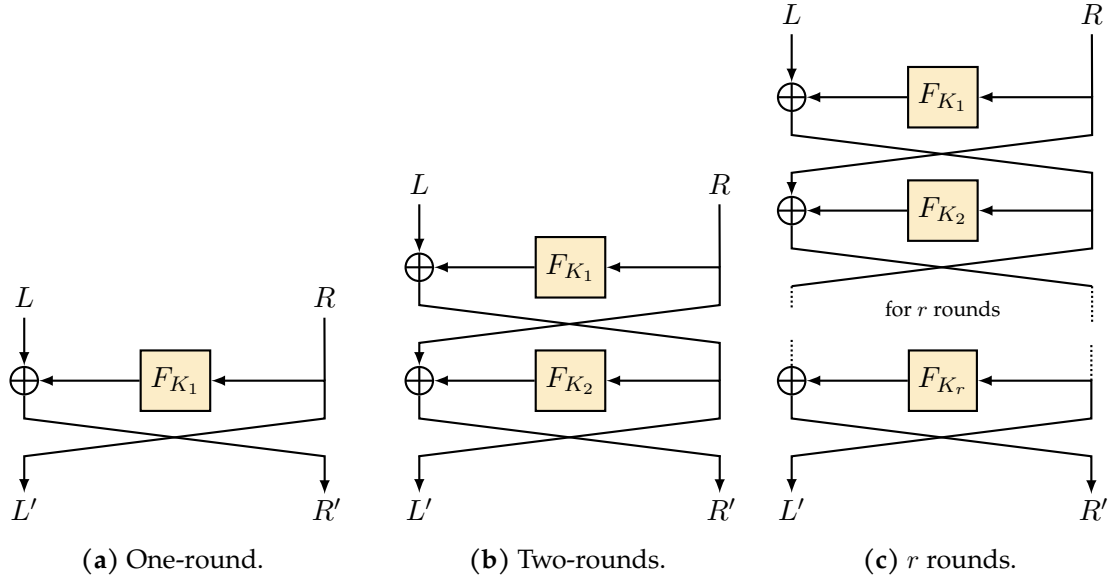


Figure 2: Feistel network.

- 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 $E_K: \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$ is *invertible*.
- b) Let $E^{(1)}: \{0, 1\}^{128} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ denote the block cipher defined by the one-round Feistel network shown in Fig. 2a, where $F: \{0, 1\}^{128} \times \{0, 1\}^{64} \rightarrow \{0, 1\}^{64}$ 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 $\text{Adv}_{E^{(1)}}^{\text{prf}}(\mathcal{A})$, where $\mathcal{A}$ is the adversary that runs your attack?

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

- c) Let $E^{(2)}: \{0, 1\}^{256} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ 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 $\text{Exp}_{E^{(2)}}^{\text{prf}}(\mathcal{A})$.

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

Problem 7

Recall that the AES round function consists of the following operations:

$$X \mapsto \text{SubBytes} \mapsto \text{ShiftRows} \mapsto \text{MixColumns} \mapsto \text{AddRoundKey} \mapsto Y$$

Since the operations ShiftRows and MixColumns are *linear*⁴ they can be represented by a matrix A . Consequently, if we think of the 4×4 AES state as a collection of four 4-byte column vectors $\mathbf{x}_i$, and use $s[\mathbf{x}_i]$ to denote applying the S-box (SubBytes) to each byte in $\mathbf{x}_i$, then the whole AES round function can be summarized as

$$\mathbf{y}_i \leftarrow A s[\mathbf{x}_i] + \mathbf{k}_i, \quad i = 1, 2, 3, 4 \quad (3)$$

where $\mathbf{k}_i$ is the corresponding 4-byte column vector of the current round key, and $\mathbf{y}_i$ is the resulting 4-byte column vector after the round function.

Note: If you are not familiar with matrix-vector multiplication you can simply pretend that A , $\mathbf{k}_i$, and $\mathbf{y}_i$ are ordinary integers, and that the multiplication and addition in (3) is standard integer multiplication and addition.

a) Now suppose AES did *not* have an S-box. Then (3) would simply become:

$$\mathbf{y}_i \leftarrow A \mathbf{x}_i + \mathbf{k}_i \quad (4)$$

Using (4), write an equation that summarizes the *whole* AES function for a given 4-byte vector $\mathbf{x}_i$. Your equation should include the 11 AES-round keys $\mathbf{k}_{i,0}, \mathbf{k}_{i,1}, \mathbf{k}_{i,2}, \dots, \mathbf{k}_{i,10}$, where $\mathbf{k}_{i,0}$ is the initial round key added to $\mathbf{x}_i$ before round 1.

Hint: after 1 round the equation looks like $\mathbf{y}_i = A(\mathbf{x}_i + \mathbf{k}_{i,0}) + \mathbf{k}_{i,1}$.

b) Show that the no-S-box variant of AES is not a secure block cipher.

References

[Ros] Mike Rosulek. *The Joy of Cryptography*. <https://joyofcryptography.com>.

⁴A function F is linear if $F(A + B) = F(A) + F(B)$ for all inputs A, B .