Problem 1. Encryption with a deck of cards. Alice, Bob and Eve are playing a card game. Alice shuffles a deck of cards and deals it all out to herself and Bob (each gets half of the 52 distinct cards). Alice now wishes to send a secret message m to Bob by saying something aloud. Everybody is in the same room, and eavesdropper Eve is listening in: she hears everything Alice says (but Eve cannot see the face of Alice's and Bob's cards).

a) Suppose Alice's message m is a string of 48-bits. Describe how Alice can communicate m to Bob in such a way that Eve will have *no* information about what is m. Note: Alice and Bob are allowed to devise a public strategy together *before* the cards are dealt.

b) Now suppose that Alice's message m is 49 bits. Show that there exists no protocol that allows Alice to communicate m to Bob in such a way that Eve will have no information about m.

Problem 2. Perfect Security. A *deterministic* symmetric encryption scheme[1] $\mathcal{E}$ specifies a pair of algorithms $\mathcal{E} = (\text{Enc}, \text{Dec})$ with three associated sets: the *key space* $\mathcal{K}$, the *message space* $\mathcal{M}$ and the *ciphertext space* $\mathcal{C}$.

- The deterministic encryption algorithm $\text{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ takes as input a key $k \in \mathcal{K}$ and a message $m \in \mathcal{M}$, and produces a ciphertext $c \in \mathcal{C}$.
- The deterministic decryption algorithm $\text{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$ takes as input a key $k \in \mathcal{K}$ and a ciphertext $c \in \mathcal{C}$, and returns a message $m \in \mathcal{M}$.

We require that $\mathcal{E}$ satisfies the *decryption correctness* property, meaning that any ciphertext produced by Enc is correctly decryptable using Dec. Formally, for all keys $k \in \mathcal{K}$ and for all messages $m \in \mathcal{M}$, it holds that $\text{Dec}(k, \text{Enc}(k, m)) = m$.

We say that encryption scheme $\mathcal{E}$ is *perfectly secure* if for all $m_0, m_1 \in \mathcal{M}$, and all $c \in \mathcal{C}$, we have

$$\Pr[\text{Enc}(\mathbf{k}, m_0) = c] = \Pr[\text{Enc}(\mathbf{k}, m_1) = c],$$

where $\mathbf{k}$ is a random variable uniformly distributed over $\mathcal{K}$. Intuitively, this means that an $\mathcal{E}$ ciphertext leaks no information about the encrypted message. An alternative definition of perfect security (for one-time pads) was provided in class.

a) Devise an encryption scheme $\mathcal{E}_{90}$ such that (1) given an encryption of any message, an adversary can figure out 90% of the secret key (i.e. the ciphertext leaks this information), but (2) the scheme is still perfectly secure, despite 90% of the key being revealed. Prove that the scheme is secure and that it is correct. When constructing your encryption scheme, you should define algorithms Enc and Dec as well as the associated sets $\mathcal{K}, \mathcal{M}$ and $\mathcal{C}$.

b) Devise an encryption scheme $\mathcal{E}_{\text{broken}}$ such that (1) given an encryption of any message, an adversary learns nothing about the secret key, but (2) the scheme is completely broken (as in, given the ciphertext, an adversary can completely recover the plaintext).

c) Build an encryption scheme $\mathcal{E}_1$ such that an adversary can recover the first bit of any message $m \in \mathcal{M}$ from its encryption, but $\mathcal{E}_1$ is nonetheless perfectly secure. Now, in addition to the above, let $\mathcal{M} = \{0,1\}^n$ for any $n \geq 1$; show that any encryption scheme $\mathcal{E}$ with message space $\mathcal{M}$ is *not* perfectly secure (if the first bit of any message $m \in \mathcal{M}$ can be recovered from its encryption, as per above).

d) Let $\mathcal{E}$ be an encryption scheme such that $|\mathcal{K}| \neq 0$ and $\mathcal{C} = \{\text{Enc}(k, m) \colon k \in \mathcal{K}, m \in \mathcal{M}\}$. Show that if $\mathcal{E}$ is perfectly secure then $|\mathcal{M}| \leq |\mathcal{C}| \leq |\mathcal{K}|$. Explain why some assumptions about $\mathcal{K}$ and $\mathcal{C}$ (such as stated above) are necessary to prove this claim.

Problem 3. PRGs. Let $\ell, n \in \mathbb{N}$ such that $\ell < n$. Let $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$ be a secure PRG (pseudorandom generator). For each of the new constructions below, determine if it is also a secure PRG. If you believe that a derived generator is not secure then provide an attack. If you believe that a derived generator is secure then try to justify that as formally as possible (if you have seen proofs

[1] We will define and discuss symmetric and asymmetric encryption schemes in detail later in class. For this exercise you only need the definition provided here.

by reduction, that is a useful technique here). In the following, s, s_1 and s_2 are strings in $\{0,1\}^\ell$, and $\|$ denotes string concatenation. The bit-wise XOR operation is denoted by $\oplus$. The bit-wise AND operation is denoted by $\wedge$. Some of the derived generators have domains or ranges that differ from that of G.

a) $\mathsf{G}_1(s) := \mathsf{G}(s) \oplus 1^n$, where 1^n is the bit-string consisting of n 1s, for example $1^4 = 1111$.
b) $\mathsf{G}_2(s) := \mathsf{G}(s)[0 \ldots n-2]$. Here we treat the output string $\mathsf{G}(s)$ as an array and use vector notation to indicate that we truncate the result by removing the last bit. For example $abcd[0 \ldots 2] = abc$.
c) $\mathsf{G}_3(s) := \mathsf{G}(s) \,\|\, \mathsf{G}(s)$. Note that the range of G_3 is $\{0,1\}^{2n}$.
d) $\mathsf{G}_4(s_1 \,\|\, s_2) := s_1 \,\|\, \mathsf{G}(s_2)$. Note that G_4 is a function from $\{0,1\}^{2\ell}$ to $\{0,1\}^{\ell+n}$.
e) $\mathsf{G}_5(s) := \mathsf{G}(s) \,\|\, \mathsf{G}(\mathsf{G}(s)[0 \ldots \ell-1])$. Note that the range of G_5 is $\{0,1\}^{\ell+n}$.
f) $\mathsf{G}_6(s_1 \,\|\, s_2) := \mathsf{G}(s_1) \wedge \mathsf{G}(s_2)$. Note that the domain of G_6 is $\{0,1\}^{2\ell}$.
g) $\mathsf{G}_7(s_1 \,\|\, s_2) := \mathsf{G}(s_1) \oplus \mathsf{G}(s_2)$. Note that the domain of G_7 is $\{0,1\}^{2\ell}$.

Problem 4. Block Ciphers. Consider the following definition of a block cipher. This definition is equivalent to the one in the lecture slides.

Definition 1. A function $\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is called a *block cipher* if

(1) $\mathcal{X} = \mathcal{Y}$ and

(2) for all $\mathsf{K} \in \mathcal{K}$, $\mathsf{E}_\mathsf{K} : \mathcal{X} \to \mathcal{X}$ is an efficiently computable permutation on the set $\mathcal{X}$.

Here $\mathsf{E}_\mathsf{K}(x) = \mathsf{E}(\mathsf{K}, x)$ for all $x \in \mathcal{X}$, which is a common shorthand notation.

Task: Let $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher.

a) Let the function $\mathsf{F}_1 : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be defined by

$$\mathsf{F}_1(\mathsf{K}, x) = \mathsf{E}(\mathsf{K}, x) \oplus x.$$

Is F_1 a block cipher? Prove your answer.

b) Let the function $\mathsf{F}_2 : (\{0,1\}^k \times \{0,1\}^n) \times \{0,1\}^n \to \{0,1\}^n$ be defined by

$$\mathsf{F}_2((\mathsf{K}_1, \mathsf{K}_2), x) = \mathsf{E}(\mathsf{K}_1, \mathsf{K}_2 \oplus x).$$

The keyspace of F_2 is $\{0,1\}^k \times \{0,1\}^n$. Show that F_2 is a block cipher and that it is PRP-secure assuming that E is PRP-secure.

c) Let the function $\mathsf{F}_3 : (\{0,1\}^k \times \{0,1\}^n) \times \{0,1\}^n \to \{0,1\}^n$ be defined by

$$\mathsf{F}_3((\mathsf{K}_1, \mathsf{K}_2), x) = \mathsf{E}(\mathsf{K}_1, \mathsf{K}_2) \oplus x.$$

Show that F_3 is a block cipher. Define a distinguisher $\mathcal{A}$ attacking the PRP-security of F_3, such that $\mathcal{A}$ makes exactly two *distinct* queries to its challenger (we call this a 2-query adversary) and achieves $\mathbf{Adv}^{\mathrm{PRP}}_{\mathsf{F}_3}(\mathcal{A}) = \frac{1}{2} \cdot (1 - \frac{1}{2^n - 1})$. (This is essentially the highest possible advantage.) Note that despite the similarities between F_2 and F_3, one is secure while the other is not. Think about how misplaced parentheses can have important consequences for security!

d) Let the function $\mathsf{F}_4 : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be defined by

$$\mathsf{F}_4(\mathsf{K}, x) = \mathsf{E}(\mathsf{K}, x) \oplus \mathsf{E}(\mathsf{K}, \mathsf{K}).$$

Prove that F_4 is a block cipher. Prove that F_4 does *not* have strong-PRP security. Provide an intuition explaining why F_4 should be a secure PRP assuming that E is PRP-secure. *Note:* Formally proving this is difficult.

Problem 5. IND-CPA security. Suppose SE = (KGen, Enc, Dec) is an IND-CPA secure encryption scheme with key space $\mathcal{K}$ and message space $\mathcal{M}$, such that $\mathcal{K} = \mathcal{M} = \{0,1\}^n$ for some even integer n. You can assume that messages of the same length have equally-sized ciphertexts (if not stated otherwise). Which of the following encryption algorithms are guaranteed to represent correct encryption schemes with IND-CPA security?

a) $\text{Enc}_a(\mathsf{K}, m) = \text{Enc}(\mathsf{K}, (m, r))$. Here, the message space for Enc_a is $\{0,1\}^{n/2}$, r is a random $n/2$-bit string, and (m, r) is the concatenation of m and r.
b) $\text{Enc}_b(\mathsf{K}, m) = \text{Enc}(\mathsf{K}, m) \oplus \text{Enc}(\mathsf{K}, 0^n)$.
c) $\text{Enc}_c(\mathsf{K}, m) = (\text{Enc}(\mathsf{K}, m), m[1])$. Here, $m[1]$ is the first bit of m.
d) $\text{Enc}_d(\mathsf{K}, m) = (\text{Enc}(\mathsf{K}, m), \text{Enc}(\mathsf{K}, m \oplus 1^n))$. That is, encrypt m, and then encrypt the bitwise complement of m.
e) $\text{Enc}_e(\mathsf{K}, m) = \{\text{Enc}(\mathsf{K}, m), \text{Enc}(\mathsf{K}, m \oplus 1^n)\}$. Here, $\{a, b\}$ denotes an unordered set containing elements a and b.
f) $\text{Enc}_f(\mathsf{K}, m) = (\text{Enc}(\mathsf{K}, m), \mathsf{K})$.
g) $\text{Enc}_g(\mathsf{K}, m) = (\text{Enc}(\mathsf{K}, m), \text{Enc}(\mathsf{K}, m))$. Here, the two calls to Enc each independently samples its own random coins.
h) $\text{Enc}_h(\mathsf{K}, m) = (c, c)$ for $c \leftarrow\!\!\$\ \text{Enc}(\mathsf{K}, m)$.

There are several cases:

(1) The new scheme is guaranteed to be secure, no matter how SE works (as long as it is IND-CPA secure). In this case, prove your claim.
(2) The new scheme is always insecure, no matter what SE does. In this case, show an attack that works no matter what.
(3) It may be the case that the new scheme is not a correct encryption scheme for some choices of SE. In this case explain why. Also explain: does the new encryption scheme leak any information about encrypted messages?

You do not need to explain how to decrypt.

Problem 6. Building hash functions. Let $h_1 : \{0,1\}^{2m} \to \{0,1\}^m$ be a hash function.

1. Let $h_2 : \{0,1\}^{4m} \to \{0,1\}^m$ be defined as follows:

   - For $x \in \{0,1\}^{4m}$ define $x_1, x_2 \in \{0,1\}^{2m}$ such that $x = x_1 || x_2$.
   - Define $h_2(x) = h_1\left(h_1(x_1) || h_1(x_2)\right)$.

   Show that finding a collision in h_2 leads to finding a collision in h_1.

2. For every $i > 2$ let $h_i : \{0,1\}^{2^i m} \to \{0,1\}^m$ be defined recursively from h_{i-1} as follows:

   - For $x \in \{0,1\}^{2^i m}$ define $x_1, x_2 \in \{0,1\}^{2^{i-1} m}$ such that $x = x_1 || x_2$.
   - Define $h_i(x) = h_1\left(h_{i-1}(x_1) || h_{i-1}(x_2)\right)$.

   Show that finding a collision in h_i leads to finding a collision in h_1.

3. Let $h_2' : \{0,1\}^{4m} \to \{0,1\}^m$ be any hash function.

   $$h'(x) = h_2(x) || h_2'(x).$$

   Show that finding a collision in h' leads to finding a collision in h_2 or h_2'.

Problem 7. Building a secure MAC. In this course, you have heard a lot about different cryptographic primitives and their constructions. Not rarely, their security depends on a parameter that changes daily, e.g. computational power.

To obtain a construction that is more resilient to changes, one sometimes uses two different constructions $\mathcal{I}_1, \mathcal{I}_2$ of the same primitive (e.g. MAC) to build a new primitive of the same type. If combined properly, the resulting construction might remain secure even if one of the underlying constructions becomes insecure.

In this problem, we define two such constructions for a MAC scheme $\mathcal{I}$. Your assignment is to prove that they remain secure even if one of the underlying MAC schemes, $\mathcal{I}_1 = (\mathrm{KGen}_1, \mathrm{Tag}_1, \mathrm{Vrf}_1)$ or $\mathcal{I}_2 = (\mathrm{KGen}_2, \mathrm{Tag}_2, \mathrm{Vrf}_2)$, becomes insecure. [For proof, you should use SUF-CMA security model with no verification queries.] Suppose that $\mathcal{I}_1$ and $\mathcal{I}_2$ are deterministic MAC schemes.

a) Let $\mathcal{I} = (\mathrm{KGen}, \mathrm{Tag}, \mathrm{Vrf})$ be a deterministic MAC scheme such that

$$\mathrm{KGen} := (\mathrm{KGen}_1, \mathrm{KGen}_2)$$

and

$$\mathrm{Tag}((k_1, k_2), m) := (\mathrm{Tag}_1(k_1, m), \mathrm{Tag}_2(k_2, m))$$

and

$$\mathrm{Vfy}((k_1, k_2), m, (\tau_1, \tau_2)) := \text{If } \mathrm{Vfy}_1(k_1, m, \tau_1) \wedge \mathrm{Vfy}_2(k_2, m, \tau_2) \text{ then return 1 else return 0.}$$

Show that $\mathcal{I}$ is secure if $\mathcal{I}_1$ or $\mathcal{I}_2$ is secure.

b) Let $\mathcal{I} = (\mathrm{KGen}, \mathrm{Tag}, \mathrm{Vrf})$ be a deterministic MAC scheme such that

$$\mathrm{KGen} := (\mathrm{KGen}_1, \mathrm{KGen}_2)$$

and

$$\mathrm{Tag}((k_1, k_2), m) := \mathrm{Tag}_1(k_1, m) \oplus \mathrm{Tag}_2(k_2, m)$$

and

$$\mathrm{Vfy}((k_1, k_2), m, \tau) := \text{ If } \tau = \mathrm{Tag}_1(k_1, m) \oplus \mathrm{Tag}_2(k_2, m) \text{ then return 1 else return 0.}$$

Show that $\mathcal{I}$ is secure if $\mathcal{I}_1$ or $\mathcal{I}_2$ is secure.