# Cryptography
## Exercise Book

Giulia Giusti

**Abstract**

This is meant to be a collection of simple exercises on modern cryptography and the symbolic model. It is work in progress, and very likely will remain so in the foreseeable future. Please report any typo or mistake to `ugo.dallago@unibo.it` or `giulia.giusti7@unibo.it`.

# 1 Perfectly Secure Encryption

### Exercise 1.1

Find an example of a perfectly secure encryption scheme such that keys are *not* generated according to the uniform distribution. First, define the encryption scheme formally, then prove its correctness and perfect security. Would it be possible to design one such scheme in such a way that $|\mathcal{M}| = |\mathcal{K}|$?

### Exercise 1.2

Consider the PlayFair classic cipher[1] and formulate it as a triple of algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such that messages are strings in $\Sigma^*$ and $\Sigma$ is the English Alphabet. Prove that PlayFair cannot be perfectly secure.

### Exercise 1.3

Design a perfectly secure encryption scheme different from the OTP such that $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^n$.

### Exercise 1.4

Prove that the Caesar cipher does not have perfect secrecy.

### Exercise 1.5

In the so-called Vernam's cipher, messages, keys and ciphertexts are all taken from the same set $\{0,1\}^n$. But what if one wants to encrypt messages written in other alphabets? Given any alphabet $\Sigma$ (namely any finite set of characters), design a perfectly-secure encryption scheme $\Pi_\Sigma$ such that $\mathcal{M} = \Sigma^n$, i.e., messages are strings of length $n$ from $\Sigma$. In doing so, you are free to decide how $\mathcal{K}$ and $\mathcal{C}$ are defined. Give a proof of perfect secrecy for your construction.

[1] see `https://en.wikipedia.org/wiki/Playfair_cipher`

# 2 Private Key Encryption and Pseudorandomness

## Exercise 2.1

Prove that, whenever $f : \mathbb{N} \to \mathbb{R}$ is a negligible function, the function $g : \mathbb{N} \to \mathbb{R}$ defined as

$$g(n) = \prod_{i=0}^{n} f(i)$$

is also negligible.

## Exercise 2.2

Consider the keyed, length-preserving, function $F$ such that $F(k; x) = k \vee x$ where $\vee$ is bitwise OR. Prove that $F$ cannot be pseudorandom by giving an attack and analyze it.

## Exercise 2.3

Consider a function $CDNT$ such that there is a polynomial $p$ such that for every natural number $n$ the result $CDNT(1^n)$ is a tuple $\langle s_1, \ldots, s_{p(n)} \rangle$ where:

- $p(n) \geq 2$;
- for every $i \in \{1, \ldots, p(n)\}$ it holds that $s_i \in \{0,1\}^n$;
- $s_i = s_j$ implies $i = j$.

Given a length-preserving function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, define $F_{CDNT}$ as the function on binary strings defined as follows:

$$F_{CDNT}(r) = F(r, s_1) \cdot \ldots \cdot F(r, s_{p(|r|)})$$

where $CDNT(1^{|r|}) = \langle s_1, \ldots, s_{p(|r|)} \rangle$ and $\cdot$ is string concatenation. Prove that if $F$ is a PRF and $CDNT$ computable in deterministic polynomial time, then $F_{CDNT}$ is a pseudorandom generator.

## Exercise 2.4

First, we need some auxiliary definitions. A function $H : \{0,1\}^* \to \{0,1\}^*$ is said to be *length-preserving* iff for every $s \in \{0,1\}^*$ it holds that the length $|s|$ of $s$ is equal to $|H(s)|$. Given two functions $f : A \to B$ and $g : B \to C$, we write $g \circ f$ for the composition of $f$ and $g$, i.e. the function from $A$ to $C$ such that $(g \circ f)(a) = g(f(a))$ for every $a \in A$. Now, let us come to the actual exercise. Let $G$ be any pseudorandom generator, and let $H$ be any length-preserving bijection. Under which conditions on $H$ is $H \circ G$ guaranteed to be a PRG? Conversely, if $H \circ G$ is pseudorandom, is $G$ itself a PRG? Prove all your claims.

## Exercise 2.5

Suppose that the experiment $\mathsf{PrivK}^{eav}$ is replaced by the following variation of it, that we call $\mathsf{PrivKRour}^{eav}$:

$\mathsf{PrivKFour}^{eav}_{\mathcal{A},\Pi}(n)$ :
- $k \leftarrow Gen(1^n)$
- $(m_0, m_1, m_2, m_3) \leftarrow \mathcal{A}(1^n)$
- if $|m_i| \neq |m_j|$ for some $i \neq j$ :
  - **return 0**
- $b \leftarrow \{0, 1, 2, 3\}$
- $c \leftarrow Enc_k(m_b)$
- $b^* \leftarrow \mathcal{A}(c)$
- **return** $(b \stackrel{?}{=} b^*)$

Now:

1. Formulate a notion of security for an encryption scheme, adapting the one we know to the new experiment.
2. Prove that any encryption scheme which is secure against $\mathsf{PrivK}^{eav}$ is also secure against $\mathsf{PrivKFour}^{eav}$.
3. Prove also the converse, namely that any encyrption scheme which is secure against $\mathsf{PrivKFour}^{eav}$ has to be secure against $\mathsf{PrivK}^{eav}$.

### Exercise 2.6

A banking company is offered to use an encryption scheme secure against passive attacks to encrypt data related to the transactions of its customers. The officials of this company, however, are not so convinced that the definition based on the $\mathsf{PrivK}^{eav}$ experiment is satisfactory. Convince the company otherwise by showing that every encryption scheme which is secure against passive attacks $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is such that no PPT adversary $\mathcal{A}$ is capable, starting from $c = \mathsf{Enc}_k(m)$, to rebuild the value of the last $\lfloor |m|/10 \rfloor$ bit of the message $m$ (where $|m|$ is the length of the message $m$).

### Exercise 2.7

Suppose $\Pi = (Gen_\Pi, Enc_\Pi, Dec_\Pi)$ and $\Theta = (Gen_\Theta, Enc_\Theta, Dec_\Theta)$ are two private-key encryption schemes. Define their *juxtaposition* as another private-key encryption scheme $\Pi\#\Theta = (Gen_{\Pi\#\Theta}, Enc_{\Pi\#\Theta}, Dec_{\Pi\#\Theta})$ such that

$$\begin{aligned} Gen_{\Pi\#\Theta}(1^n) &= \langle Gen_\Pi(1^n), Gen_\Theta(1^n)\rangle; \\ Enc_{\Pi\#\Theta}(\langle k_\Pi, k_\Theta\rangle, \langle m_\Pi, m_\Theta\rangle) &= \langle Enc_\Pi(k_\Pi, m_\Pi), Enc_\Theta(k_\Theta, m_\Theta)\rangle; \\ Dec_{\Pi\#\Theta}(\langle k_\Pi, k_\Theta\rangle, \langle c_\Pi, c_\Theta\rangle) &= \langle Dec_\Pi(k_\Pi, c_\Pi), Dec_\Theta(k_\Theta, c_\Theta)\rangle; \end{aligned}$$

where $\langle\cdot,\cdot\rangle$ is an operator for string concatenation. First of all, prove that $\Pi\#\Theta$ is correct whenever $\Pi$ and $\Theta$ are correct. Then, prove that if $\Pi\#\Theta$ is secure against passive adversaries, than both $\Pi$ and $\Theta$ are. What can we say about the reverse implication?

## 3 Message Authentication Codes (MAC)

### Exercise 3.1

A banking company is offered to use a secure authentication scheme to send data relating to bank transfers made to customers. The officials of this company, however, are not so convinced that the MacForge experiment is satisfactory. In particular, officials would like to be sure that no adversary, after seeing $n$ bank transfers $b_1, ..., b_n$ (where $n$ is the safety parameter) and the related tags $t_1, ..., t_n$ pass through the channel, can build a valid tag of a new transfer $b$ different from each one of the $b_i$. Prove that every secure MAC is such that such an attack, if PPT, cannot have a more than negligible probability of success. Assume that, by the Kerchoff principle, the adversary has access to a way, given an amount $a$, to generate in polynomial time in the length of $a$, a transfer of an amount equal to $a$.

### Exercise 3.2

Given a pseudorandom function $F$, consider the MAC $\Pi^2_F = (Gen, Mac, Vrfy)$ where:

- $Gen$ returns a binary string of length $n+1$ whenever invoked in $1^n$.
- $Mac_k$ is only defined on messages of length $2|k| - 2$ and returns on input $m$ the string $F_k(k_1||m_0)||F_k((\neg k_1)||m_1)$, where $m = m_0||m_1$, it holds that $|m_0| = |m_1| = |k| - 1$, $k_1$ is the first bit of $k$, and $||$ is the concatenation operator.
- $Vrfy$ is defined in a natural way.

Please discuss about the security of $\Pi^2_F$ under the hypothesis that $F$ is indeed pseudorandom.

### Exercise 3.3

Many of the MACs we have studied are such that the $\mathsf{Vrfy}$ function simply proceeds by checking whether the input tag is identical to the one produced by $\mathsf{Mac}$. We call such MACs *canonical*. Still talking about MACs, an apparently stronger definition of security than the one we have given consists in allowing the adversary to not only have access to an oracle for $\mathsf{Mac}_k$, but also to an oracle for $\mathsf{Vrfy}_k$. Prove that for every canonical MAC the aforementioned security definition is equivalent to the one we saw during the course.

## 4 Assumptions from Number Theory and Algebra

### Exercise 4.1

Consider the three assumptions: CDH, DDH, and discrete logarithm. We said, without proving it, that there are logical implications between them. Choose two of the three assumptions and formally prove the implication between them by means of a proof by reduction.

### Exercise 4.2

Study the DDH, CDH and discrete logarithm assumptions when the underlying algorithm $\mathsf{GenCG}$ on input $1^n$ produces in output a triple $(\mathbb{Z}_n, n, g)$ where the operation underlying $\mathbb{Z}_n$ is taken to be addition modulo $n$. Is there any constraint on $g$? Is there any hope that the three aforementioned assumptions indeed hold?

# Answers to Selected Exercises

## Answer to Exercise 1.1

Consider the following variation of the Vernam's cipher. Let $\Pi = (Gen, Enc, Dec)$, where:

- The message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$ are $\{0,1\}$ whereas the key space $\mathcal{K}$ is given by $\{0,1\}^2$.
- The algorithm *Gen* chooses a string $k = k_1k_2$ from $\mathcal{K}$, sampling $k_1$ according to uniform distribution, while $k_2$ is sampled not uniformly (e.g. by always outputting 1).
- The encryption algorithm *Enc* works as follows: given a key $k_1k_2 \in \{0,1\}^2$ and a message $m \in \{0,1\}$, *Enc* outputs $c := k_1 \oplus m$
- The decryption algorithm *Dec* works as follows: given a key $k = k_1k_2 \in \{0,1\}^2$ and a ciphertext $c \in \{0,1\}$, output $m := k_1 \oplus c$.

$\Pi$ is correct. Indeed, we have that $Dec_k(Enc_k(m)) = (k_1 \oplus (k_1 \oplus m)) = m$. $\Pi$ is also perfectly secure. As for the Vernam's cipher, we can prove that $Pr(C = c \mid M = m)$ does not depend on $m$:

$$\begin{aligned}\Pr(C = c \mid M = m) &= \Pr(M \oplus K_1 = c \mid M = m) = \Pr(m \oplus K_1 = c) \\ &= \Pr(m \oplus (m \oplus K_1) = m \oplus c) = \Pr(K_1 = m \oplus c) = \frac{1}{2}\end{aligned}$$

We cannot define a perfectly secure scheme such that keys are not generated according to the uniform distribution and such that it also holds that $|\mathcal{M}| = |\mathcal{K}|$. We can indeed verify that for every perfectly secure scheme such that $|\mathcal{M}| = |\mathcal{K}|$, all keys are chosen with the same probability. First of all, if the encryption scheme is secure and $|\mathcal{M}| = |\mathcal{K}|$, then for every $c \in \mathcal{C}$ there is a bijection $\sigma_c : \mathcal{K} \to \mathcal{M}$ such that $\sigma_c(k) = Dec_k(c)$. Indeed, if

$$Dec_{\mathcal{K}}(c) := \{m \in \mathcal{M} \mid \exists k \in \mathcal{K}.\ Dec_k(c) = m\}$$

Then $Dec_{\mathcal{K}}(c) = \mathcal{M}$. Indeed, if there were $m \in \mathcal{M}$ is such that $m \notin Dec_{\mathcal{K}}(c)$, then, for any $m' = Dec_k(c)$ with $Pr(K = k) > 0$, it should be $Pr(C = c \mid M = m) \neq Pr(C = c \mid M = m')$, contradicting perfect security. Then, by defintion of $\sigma_c$, we have that $\sigma_c(\mathcal{K}) = \mathcal{M}$. So, $\sigma_c$ is a surjective function between two (finite) sets having same cardinality, i.e. it is a bijection. As a consequence, for every $k$ it holds that

$$Pr(K = k) = Pr(C = c \mid M = \sigma_c(k))$$

since, if other keys send $\sigma_c(k)$ to $c$, then we would not have a bijection. And for every $k, h \in \mathcal{K}$, by perfect secrecy,

$$Pr(K = k) = Pr(C = c \mid M = \sigma_c(k)) = Pr(C = c \mid M = \sigma_c(h)) = Pr(K = h)$$

i.e. all keys are chosen with the same probability.

## Answer to Exercise 1.2

PlayFair can be formalized as an encryption scheme composed of three spaces $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and a triple $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ where:

$$\mathsf{Gen} : 1 \to \mathcal{K} \qquad \mathsf{Enc} : \mathcal{M} \text{ x } \mathcal{K} \to \mathcal{C} \qquad \mathsf{Dec} : \mathcal{C} \text{ x } \mathcal{K} \to \mathcal{M}$$

Let us consider $\Sigma$ as the English alphabet and $\Delta$ as the English alphabet - $\{J\}$ because we use the convention in which the letter $J$ is not used in the key table.
The three spaces are defined as follows:

- $\mathcal{K} \in$ 5x5 table containing the elements of $\Delta$, the filling of this table depends on the generation of a pair $< keyword, convention\_filling >$ where $keyword \in \Delta^*$ and $convention\_filling \in$ set of all possible strategies for placing the non-duplicated letters of *keyword* in order in a 5x5 table.

- $\mathcal{M} \in \Sigma^*$
- $\mathcal{C} \in \Delta^*$

The three algorithms for PlayFair are defined as follows:

- Gen is an algorithm that outputs a key $k$ in the form of a 5x5 table generated as follows:

  $\mathsf{Gen}(1)$ :
  $keyword \leftarrow$ random generation of a string with alphabet $\Sigma$
  $convention\_filling \leftarrow$ random choice of a convention
  $k \leftarrow generate\_table(convention\_filling, keyword)$
  **return** $k$

- Enc is an algorithm that takes as input a message $m =< m_1, ..., m_n >$ where the $m_i$ are the characters of the plaintext message and a key $k$ and returns as output a ciphertext $c$ calculated in the following way:

  $\mathsf{Enc}(m, k)$ :
  **for** $n$=1 **to** $|m|$ **do**
  $$(l_1, l_2) = \begin{cases} [m_{2n-1}, X] & \text{if } 2n > |m| \ \vee \ m_{2n-1} = m_{2n} \\ [m_{2n-1}, m_{2n}] & \text{otherwise} \end{cases}$$
  $[i_1, j_1] \leftarrow$ position of $l_1$ in $k$
  $[i_2, j_2] \leftarrow$ position of $l_2$ in $k$
  $$(c_{2n-1}, c_{2n}) = \begin{cases} (k[(i_1+1) \ mod \ 5, \ j_1], k[(i_2+1) \ mod \ 5, \ j_2]) & \text{if } j_1 = j_2 \\ (k[i_1, \ (j_1+1) \ mod \ 5], k[i_2, \ (j_2+1) \ mod \ 5]) & \text{if } i_1 = i_2 \\ (k[i_1, \ j_2], k[i_2, \ j_1]) & \text{otherwise} \end{cases}$$
  **return** $c$

  Note that using $mod$ 5 for each operation is necessary to respect the cyclicality of the rows and columns of the table.

- Dec is an algorithm that takes as input a ciphertext $c =< c_1, ..., c_n >$ where the $c_i$ are the characters of the ciphertext and a key $k$ and returns as output a clear message $m$ calculated in the following way:

  $\mathsf{Dec}(c, k)$ :
  **for** $n$=1 **to** $|c|$ **do**
  $(l_1, l_2) = [c_{2n-1}, c_{2n}]$
  $[i_1, j_1] \leftarrow$ position of $l_1$ in $k$
  $[i_2, j_2] \leftarrow$ position of $l_2$ in $k$
  $$(m_{2n-1}, m_{2n}) = \begin{cases} (k[(i_1-1) \ mod \ 5, \ j_1], k[(i_2-1) \ mod \ 5, \ j_2]) & \text{if } j_1 = j_2 \\ (k[i_1, \ (j_1-1) \ mod \ 5], k[i_2, \ (j_2-1) \ mod \ 5]) & \text{if } i_1 = i_2 \\ (k[i_1, \ j_2], k[i_2, \ j_1]) & \text{otherwise} \end{cases}$$
  **return** $m$

Now we want to prove that PlayFair cannot be perfectly secure. By contradiction, we assume that PlayFair is perfectly secure. By Shannon Theorem we know that if an encryption scheme is perfectly secure on $\mathcal{M}, \mathcal{K}, \mathcal{C}$ then $|\mathcal{K}| \geq |\mathcal{M}|$.
In PlayFair we have that:

- $\mathcal{K}$ contains all possible ways to create a 5x5 table by filling it with 25 letters of the English alphabet $\Delta$, so $|\mathcal{K}| = 25!$.
- $\mathcal{M}$ contains all possible strings of the English alphabet $\Sigma$ with arbitrary length, , so $|\mathcal{M}| = 26^n$ where $n$ is the message length which is not fixed a priori.

Consequently we obtain that $|\mathcal{K}| < |\mathcal{M}|$ and we can conclude that PlayFair cannot be perfectly secure.

## Answer to Exercise 2.1

Let $f : \mathbb{N} \to \mathbb{R}$ a negligible function and $g(n) = \prod_{i=0}^{n} f(i)$. Since $f$ is negligible, there is $N$ such that for every $n \geq N$, $f(n) \leq 1$ (just consider the constant polynomial $constone(n) = 1$). There is also $M > N$ such that for every $n > M$, $f(n) \leq \frac{1}{\prod_{i=0}^{N} f(i)}$. Consider now any polynomial $p$. Then there is $N_p$ such that for every $n \geq N_p$, $f(n) \leq \frac{1}{p(n)}$. Now, for every $n \geq max\{M+2, N_p\}$ we have that

$$\begin{aligned} g(n) &= \prod_{i=0}^{n} f(i) = f(n) \cdot \prod_{i=0}^{n-1} f(i) \leq f(n) \cdot \prod_{i=0}^{M+1} f(i) \\ &= f(n) \cdot \left( \prod_{i=0}^{M} f(i) \right) \cdot f(M+1) \\ &\leq \frac{1}{p(n)} \cdot \left( \prod_{i=0}^{M} f(i) \right) \cdot \frac{1}{\prod_{i=0}^{N} f(i)} \\ &\leq \frac{1}{p(n)} \cdot \left( \prod_{i=0}^{N} f(i) \right) \cdot \frac{1}{\prod_{i=0}^{N} f(i)} = \frac{1}{p(n)} \end{aligned}$$

In conclusion, $g$ is negligible.

## Answer to Exercise 2.2

Let us recall the definition of a pseudorandom function

**Definition 1** *Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed function. $F$ is a pseudorandom function if for all probabilistic polynomial-time distinguishers $D$, there is a negligible function $\varepsilon$ such that:*

$$|Pr(D^{F_k(\cdot)}(1^n) = 1) - Pr(D^{f(\cdot)}(1^n) = 1)| \leq \varepsilon(n) \tag{1}$$

*where $k$ is chosen randomly from the set of all strings of length $n$ and $f(\cdot)$ is a random function from $\{0,1\}^n$ to $\{0,1\}^n$.*

Let $n = |k|$. In order to prove that the function $F$ described in the exercise is not a pseudorandom function, we define a distinguisher $D'$ such that $|Pr(D'^{F_k(\cdot)}(1^n) = 1) - Pr(D'^{f(\cdot)}(1^n) = 1)|$ is not negligible. The distinguisher $D'$ is defined as follows

Distirnguisher $D'$:
$D'$ is given input $1^n$and access to an oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$.
1) Run $\mathcal{O}(1^n)$, obtaining the string $z$.
2) Return 1 if $z = 1^n$, and 0 otherwise.

First, we observe that if the oracle $\mathcal{O}$ of $D'$ is a random function $f$ then we have that

$$Pr(D'^{f(\cdot)}(1^n) = 1) = \frac{1}{2^n}$$

Second, we observe that if the oracle $\mathcal{O}$ of $D'$ is the function $F$ described in the exercise then, for all $k$, we have that

$$Pr(D'^{F_k(\cdot)}(1^n) = 1) = 1$$

because $k \vee 1^n$ is always equal to $1^n$.
We can conclude that $F$ is not a pseudorandom function because

$$|Pr(D'^{F_k(\cdot)}(1^n) = 1) - Pr(D'^{f(\cdot)}(1^n) = 1)| = 1 - \frac{1}{2^n}$$

and $1 - \frac{1}{2^n}$ is not negligible.

## Answer to Exercise 2.5

We will consider the two following experiments:

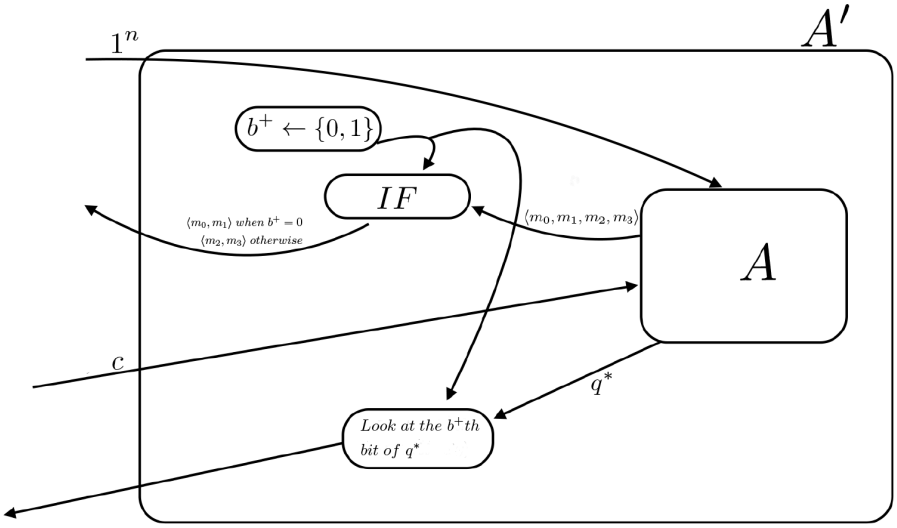
$\mathsf{PrivK}^{eav}_{A,\Pi}(n)$:
$(m_0, m_1) \leftarrow A(1^n)$;
**if** $|m_0| \neq |m_1|$ **then**
  **Result:** 0
$k \leftarrow Gen(1^n)$; $b \leftarrow \{0,1\}$;
$c \leftarrow Enc_k(m_b)$;
$b^* \leftarrow A(c)$;
**Result:** $b^* \stackrel{?}{=} b$

$\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)$:
$(m_0, m_1, m_2, m_3) \leftarrow A(1^n)$;
**if** $|m_0| \neq |m_1|$ **then**
  **Result:** 0
$k \leftarrow Gen(1^n)$; $q \leftarrow \{0,1,2,3\}$;
$c \leftarrow Enc_k(m_q)$;
$q^* \leftarrow A(c)$;
**Result:** $q^* \stackrel{?}{=} q$

1. We already know the notion of perfect security with respect to $\mathsf{PrivK}^{eav}$. We give now the notion of perfect security with respect to $\mathsf{PrivKFour}^{eav}$.
   **Definition 2** *An encryption scheme* $\Pi$ *is said to be secure with respect to* $\mathsf{PrivKFour}^{eav}$ *iff for every PPT adversary* $A$ *there exists a negligible function* $\varepsilon$ *such that:*

   $$Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1) \leq \frac{1}{4} + \varepsilon(n)$$

   So, an encryption scheme is secure iff the success probability of any PPT adversary is at most negligibly greater than $\frac{1}{4}$. Indeed, it is easy to succeed with probability $\frac{1}{4}$ by just outputting the outcome of two coin flips. The challenge is to do better than this.
2. We prove now that any encryption scheme which is secure against $\mathsf{PrivK}^{eav}$ is also secure against $\mathsf{PrivKFour}^{eav}$ . We prove this by reduction, i.e we consider an adversary $A$ which wins over $\mathsf{PrivKFour}^{eav}$, and upon $A$ we construct an adversary $A'$ that wins over $\mathsf{PrivK}^{eav}$. The adversary $A'$ can be built from $A$ as follows:



Let us be a little more specific about how the final output of $A'$ is produced. At the end of the experiment, $A$ will output an element $q^* \in \{0,1,2,3\}$. Then, the output of $A'$ will be the bit in position $b^+$ in the binary representation of $q^*$, call it $b^*$. For example, suppose that A outputs $q^* = 2$, whose binary representation is 10, and that $b^+ = 1$. Then, $b^* = 1$, because the bit in position 1 in the string 10 is 1. Now, how can we relate the two probabilities of success, namely

$$\Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1) \qquad \Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1)$$

We know that

$$Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1) = \frac{1}{4} + \varepsilon(n)$$

where $\varepsilon$ is *not* negligible. We also have that

$$\begin{aligned}\Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1) &= \Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1) \mid b^+ = 0) \cdot \Pr(b^+ = 0) \\ &\quad + \Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1) \mid b^+ = 1) \cdot \Pr(b^+ = 1) \\ &= \frac{1}{2} \cdot \Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1 \mid b^+ = 0) \\ &\quad + \frac{1}{2} \cdot \Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1 \mid b^+ = 1)\end{aligned}$$

We can play exactly the same trick, but now with the value of $b$, namely the random bit drawn by the $\mathsf{PrivK}^{eav}$ experiment, and conclude that

$$\Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1) = \frac{1}{4} \cdot \sum_{t=00}^{11} Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1 \mid bb^+ = t)$$

Now consider $q \in \{0,1,2,3\}$ as in $\mathsf{PrivKFour}^{eav}$ experiment's description and $s_q$ its binary encoding. We have then

$$\begin{aligned}\frac{1}{4} + \varepsilon(n) &= Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1) \\ &= \frac{1}{4} \cdot (\Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1 \mid s_q = 00) \\ &\quad + \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1 \mid s_q = 01) \\ &\quad + \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1 \mid s_q = 10) \\ &\quad + \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1 \mid s_q = 11))\end{aligned}$$

So,

$$\sum_{t=00}^{11} \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1 \mid s_q = t) = 1 + \varepsilon^*(n) \tag{2}$$

where $\varepsilon^*(n)$ is not negligible. We now have to consider terms in the form

$$\Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1 \mid bb^+ = t)$$

for $t \in \{00, 01, 10, 11\}$, and relate them to the terms in the sum (2). Let us for example consider the case $t = 00$. For $A'$ to be able to win in this situation, it must be that the bit in position 0 (namely the value of $b^+$) of the output of $A$ is 0 (namely the value of $b$). This happens precisely when $A'$ wins, given that $s_q = 00$ or $s_q = 10$. In other words

$$\begin{aligned}\Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n) = 1 \mid bb^+ = 00) &= \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1 \mid s_q = 00) \\ &\quad + \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n) = 1 \mid s_q = 10)\end{aligned}$$

With a very similar kind of reasoning, we can reach

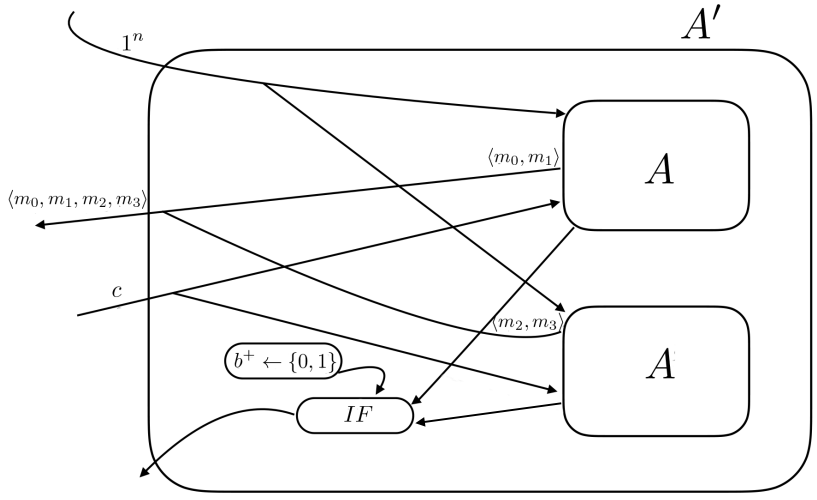
$$\begin{aligned}
\Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n)=1 \mid bb^+=01) &= \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)=1 \mid s_q=00)\\
&\quad + \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)=1 \mid s_q=01)\\
\Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n)=1 \mid bb^+=10) &= \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)=1 \mid s_q=01)\\
&\quad + \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)=1 \mid s_q=11)\\
\Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n)=1 \mid bb^+=11) &= \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)=1 \mid s_q=11)\\
&\quad + \Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)=1 \mid s_q=10)
\end{aligned}$$

As a consequence:

$$\begin{aligned}
Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n)=1) &= \frac{1}{4}\cdot\sum_{t=00}^{11} Pr(\mathsf{PrivK}^{eav}_{A',\Pi}(n)=1 \mid bb^+=t)\\
&= \frac{1}{4}\cdot\left(2\cdot\sum_{t=00}^{11}\Pr(\mathsf{PrivKFour}^{eav}_{A,\Pi}(n)=1 \mid s_q=t)\right)\\
&= \frac{1}{4}(2+2\varepsilon^*(n)) = \frac{1}{2}+\frac{\varepsilon^*(n)}{2}
\end{aligned}$$

which is negligible.

3. We prove now that any encryption scheme which is secure against $\mathsf{PrivKFour}^{eav}$ is also secure against $\mathsf{PrivK}^{eav}$. Again, we prove this by reduction, i.e we consider an adversary $A$ which wins over $\mathsf{PrivK}^{eav}$, and upon $A$ we construct an adversary $A'$ that wins over $PrivKFour^{eav}$. For every adversary for $\mathsf{PrivK}^{eav}$, we can build an aversary for $\mathsf{PrivKFour}^{eav}$ as follows:




Now, consider $q$ as in $\mathsf{PrivKFour}^{eav}$ experiment's description and $b^+$ the random bit that determines which of the two bits produced by the two copies of $A$ will be outputted by $A'$ (see the figure above). First, let us observe that

$$\Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n)=1 \mid q\in\{2,3\}\wedge b^+=1) = \Pr(\mathsf{PrivK}^{eav}_{A,\Pi}(n)=1)$$

Indeed, if $q\in\{2,3\}$ and $b^+=1$ then the outcome of $\mathsf{PrivKFour}^{eav}_{A',\Pi}(n)$ directly depends on $\mathsf{PrivK}^{eav}_{A,\Pi}(n)$. On the other hand, we cannot say much about, e.g.

$$\Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n)=1 \mid q\in\{2,3\}\wedge b^+=0)$$

Finally, we have that

$$\Pr(\mathsf{PrivK}^{eav}_{A,\Pi}(n) = 1) = \frac{1}{2} + \varepsilon^*(n)$$

where $\varepsilon^*$ is not negligible. Then we have that:

$$\begin{aligned}
&Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n) = 1) \\
&= Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n) = 1 \mid q \in \{0,1\}) \cdot Pr(q \in \{0,1\}) \\
&+ Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n) = 1 \mid q \in \{2,3\}) \cdot Pr(q \in \{2,3\}) \\
&= \frac{1}{2} \cdot [Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n) = 1 \mid q \in \{0,1\} \wedge b^+ = 0) \cdot Pr(b^+ = 0) \\
&\qquad + Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n) = 1 \mid q \in \{0,1\} \wedge b^+ = 1) \cdot Pr(b^+ = 1)] \\
&+ \frac{1}{2} \cdot [Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n) = 1 \mid q \in \{2,3\} \wedge b^+ = 0) \cdot Pr(b^+ = 0) \\
&\qquad + Pr(\mathsf{PrivKFour}^{eav}_{A',\Pi}(n) = 1 \mid q \in \{2,3\} \wedge b^+ = 1) \cdot Pr(b^+ = 1)] \\
&\geq \frac{1}{2} \cdot \left[\frac{1}{2} \cdot Pr(\mathsf{PrivK}^{eav}_{A,\Pi}(n) = 1) + 0\right] + \left[0 + \frac{1}{2} \cdot Pr(\mathsf{PrivK}^{eav}_{A,\Pi}(n) = 1)\right] \\
&\geq \frac{1}{2} \cdot \left(\frac{1}{4} + \frac{\varepsilon^*(n)}{2}\right) + \frac{1}{2} \cdot \left(\frac{1}{4} + \frac{\varepsilon^*(n)}{2}\right) \\
&= \frac{1}{4} + \frac{\varepsilon^*(n)}{4}
\end{aligned}$$

which is the thesis.

## Answer to Exercise 3.1

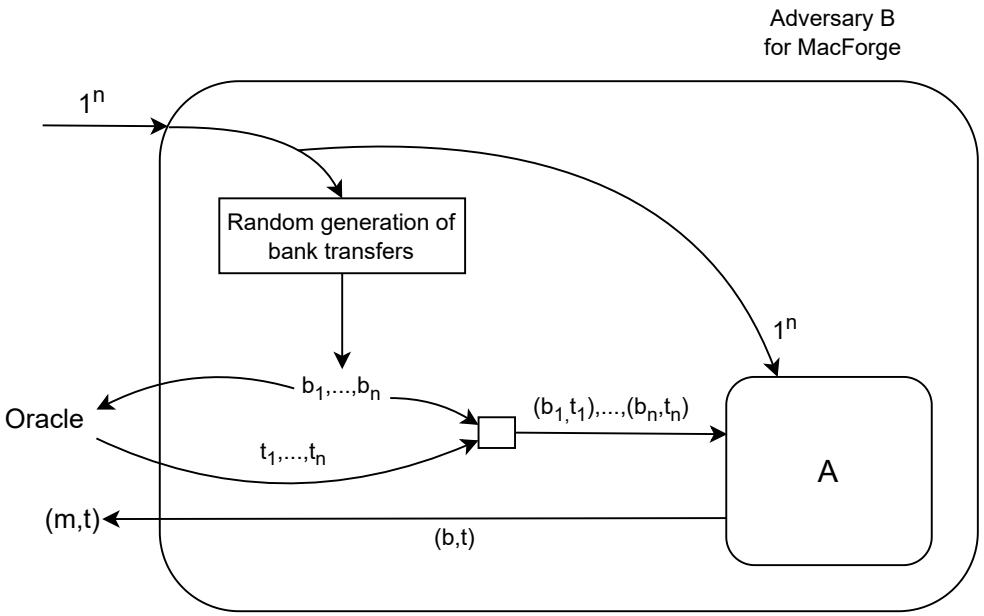
Given a secure authentication scheme $\Pi$, we want to show that an adversary like the one described in the exercise cannot have more than negligible chance of success. The adversary A described in the exercise has access to:

- $n$ pairs of in the form $(b_i, t_i)$, where $b_i$ are meaningful messages (bank transfers) and $t_i$ are valid tags for the aforementioned messages.
- An algorithm able to generate, starting from a value $a$, a valid bank transfer of amount $a$ in polynomial time. Note that thanks to this we know that the adversary described by the exercise is PPT, this is crucial.

Let us proceed by reduction, defining an adversary B for MacForge that uses the adversary A described in the exercise. More precisely, the adversary B is defined as follows

Adversary B:
B is given input $1^n$and access to an oracle $\mathcal{O}$.
1) Randomly generate $n$ bank transfers, denoted by $b_1, ..., b_n$.
2) Query $n$ times the oracle $\mathcal{O}$ to get the valid tags (denoted by $t_1, ..., t_n$) related to the bank transfers generated in the previous step.
3) Run $A(1^n, (b_1, t_1), ..., (b_n, t_n))$ which returns a pair $(b, t)$ containing a bank transfer $b \notin \{b_1, ..., b_n\}$ and the related tag $t$.
3) Return $(b, t)$ as output.

The behavior of adversary B can be summarized graphically as follows




By this reduction we have that

$$Pr(MacForge_{B,\Pi}(n) = 1) = Pr(A(n, (b_1, t_1), ..., (b_n, t_n)) = (b, t))$$

By hypothesis we know Π is a secure authentication scheme, consequently, by definition, the first term is negligible and we can conclude that the second term is negligible, which is the thesis.

## Answer to Exercise 3.2

The MAC $\Pi_F^2$, defined in the exercise, is not a secure authentication scheme because it is possible to define an adversary $A$ in the sense of the MacForge experiment which has a non-negligible probability of success. The adversary $A$ can be built as follows:

Adversary A:
A is given input $1^n$and access to an oracle $\mathcal{O} = Mac_k(\cdot)$.
1) Query the oracle $\mathcal{O}$ on input $0^{2n}$, obtaining the string
$y = y_1||y_2 = F_k(k_1||0^n)||F_k((\neg k_1)||0^n)$
2) Query the oracle $\mathcal{O}$ on input $1^{2n}$, obtaining the string
$z = z_1||z_2 = F_k(k_1||1^n)||F_k((\neg k_1)||1^n)$
3) Return $(m, t) = (0^n||1^n, y_1||z_2)$ as output.

We can observe that

$$Pr(MacForge_{A,\Pi_F^2}(n) = 1) = 1$$

because the message $m = 0^n||1^n$ has not been used by $A$ for any oracle queries ($m \notin \{0^{2n}, 1^{2n}\}$) and $Vrfy(k, m, t) = Vrfy(k, 0^n||1^n, y_1||z_2) = 1$, so we can conclude that $\Pi_F^2$ is not a secure authentication scheme.

## Answer to Exercise 4.1

Consider the following implication: if the $CDH$ assumption holds with respect to $GenCG$ then the discrete logarithm assumption also holds with respect to $GenCG$. We will consider the following two experiments related to the $CDH$ assumption and the discrete logarithm assumption:

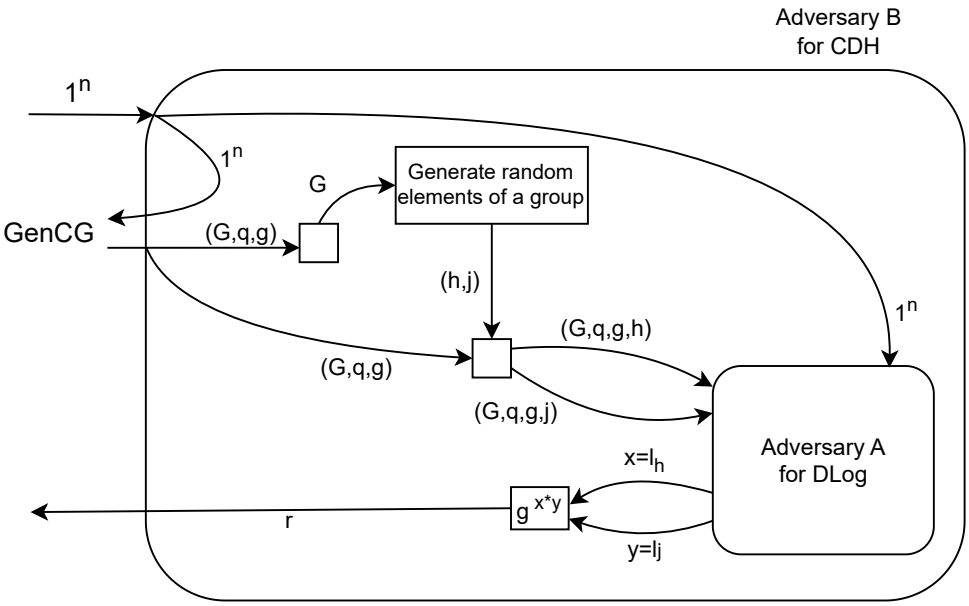
$\mathsf{CDH}_{A,GenCG}(n)$:
$(\mathbb{G}, q, g) \leftarrow GenCG(1^n)$;
$h \leftarrow \mathbb{G}$;
$j \leftarrow \mathbb{G}$;
$r \leftarrow A(\mathbb{G}, q, g, h, j)$;
**Result:** $DH_g(h, j) \stackrel{?}{=} r$

$\mathsf{DLog}_{A,GenCG}(n)$:
$(\mathbb{G}, q, g) \leftarrow GenCG(1^n)$;
$h \leftarrow \mathbb{G}$;
$x \leftarrow A(\mathbb{G}, q, g, h)$;
**Result:** $g^x \stackrel{?}{=} h$

We prove this by reduction, i.e we build an adversary $B$ in the sense of $\mathsf{CDH}_{B,GenCG}$ starting from an adversary $A$ in the sense of $\mathsf{DLog}_{A,GenCG}$. The adversary $B$ can be built from $A$ as follows:

Adversary B:
B is given input $1^n$and access to a routine $GenCG$.
1) Run $GenCG(1^n)$, obtaining$(\mathbb{G}, q, g)$.
2) Generate two random elements of the group $\mathbb{G}$, denoted by $h$ and $j$.
3) Run $A(\mathbb{G}, q, g, h)$ which returns the discrete logarithm of h, denoted by $l_h$.
4) Run $A(\mathbb{G}, q, g, j)$ which returns the discrete logarithm of j, denoted by $l_j$.
5) Compute $r$ as $g^{l_h \cdot l_j}$.
6) Return $r$ as output.

The behavior of adversary B can be summarized graphically as follows




By this reduction we have that

$$Pr(\mathsf{CDH}_{B,GenCG}(n) = 1) = Pr(\mathsf{DLog}_{A,GenCG}(n) = 1)$$

and from this equality we can observe that if adversary $B$ wins against the experiment $\mathsf{CDH}$ then adversary $A$ also wins against the experiment $\mathsf{DLog}$. We can conclude that every efficient algorithm for the discrete logarithm induces an efficient algorithm for $CDH$. In other words, this means that the $CDH$ assumption implies the discrete logarithm assumption, which is the thesis.