Exercises solutions of Cryptography
Q7 - LMAT2450

Benoît Legat   Guillaume Gheysen   Olivier Leblanc   Luis Tascon Gutierrez

Jean-Martin Vlaeminck

Compilation: 06/01/2022 (11:56)

Last modification: 26/01/2020 (16:14) `7c6097bd`

Contents

1 **3**

1.1 Exercise 1 (Vigenère) 3
1.2 Exercise 2 (Perfect secrecy.) 3
1.3 Exercise 4 (Negligible functions.) 3
1.4 Exercise 5 (Efficiency.) 3

2 **4**

2.1 Exercise 1 (Security model.) 4
2.2 Exercise 2 (Pseudorandomness.) 4
2.3 Exercise 3 (Reduction.) 5
2.4 Exercise 4 (Reduction and/or attacks.) 5

3 **6**

3.1 Exercise 0 (Simple attacks) 6
3.2 Exercise 1 (Fixed-length MAC) 6
3.3 Exercise 2 (MAC length extension) 6
3.4 Exercise 3 (Authenticated encryption and sPRP) 7
3.5 Exercise 4 (Authenticated encryption) 7
3.6 Exercise 5 7
3.7 Exercise 6 8
3.8 Exercise 7 (Blue-ray security) 8
3.9 Exercise 8 (Authenticated encryption, or not) 9
3.10 Exercise 9 (Authenticated encryption) 9
3.11 Exercise X (Mode of operation) 9

4 **10**

4.1 Exercise 1 (Authenticated encryption, August 2019 exam) 10
4.2 Exercise 2 (Authenticated Encryption and sPRP) 11
4.3 Exercise 3 (Hash functions from. . . hash functions) 11
4.4 Exercise 4 (Block-cipher based hash function) 11
4.5 Exercise 5 (Authenticated encryption, or not) 11
4.6 Exercise 6 (Variable-length MAC) 12
4.7 Exercise 7 (Hash-MAC) 12
4.8 Exercise 8 (Blue-ray security) 12

5 **13**

5.1 Exercise 0 (Group order) 13
5.2 Exercise 3 (Euclidean algorithm for gcd) 13
5.3 Exercise 4 13
5.4 Exercise 5 (Group order) 13

6 **14**

6.1 Exercise 1 (ElGamal Public Key Encryption and CCA Security) 14
6.2 Exercise 2 (A Variation of ElGamal in PKE) 14
6.3 Exercise 3 (Decisional Diffie-Hellman, $\mathbb{Z}_p^*$, and QR_p) 14
6.4 Exercise 4 (A variation of ElGamal in QR_p) 14
6.5 Exercise 5 (DDH PRG) 14
6.6 Exercise X (A variant of ElGamal Encryption.) 15

7 **16**
7.1 Exercise 1 (Commitment scheme) . . . 16
7.2 Exercise 2 (Commitment with DL) . . . 16
7.3 Exercise 3 (Commitment scheme and batching) . . . 16
7.4 Exercise 4 (Decisional Diffie-Hellman and $\mathbb{Z}_p^*$) . . . 16
7.5 Exercise 5 . . . 17

8 **18**
8.1 Exercise 1 (Zero knowledge Petersen) . . . 18
8.2 Exercise 2 (Schnorr ZKP with faulty PRG) . . . 18
8.3 Exercise 3 (Commitment scheme and batching) . . . 18

9 **19**
9.1 Exercise 0 (Commitment scheme and batching) . . . 19
9.2 Exercise 1 (Jan 2011 evaluation) . . . 19
9.3 Exercise 2 (RSA permutation with modulus 221) . . . 19
9.4 Exercise 3 (Derandomizing signatures) . . . 20
9.5 Exercise 4 . . . 20
9.6 Exercise X (Jan 2011 evaluation) . . . 20

10 **21**
10.1 Exercise 1 (Authenticated encryption, or not; August Exam) . . . 21
10.2 Exercise 2 (Derandomizing signatures) . . . 21
10.3 Exercise 3 (Jan 11 evaluation) . . . 21

Special thanks to the assistants Francesco Berti (francesco.berti@uclouvain.be) and Pierrick Méaux (pierrick.meaux@uclouvain.be) who gave us the LaTeX code of the statements of the exercises.

Note: I (Luis Tascon Gutierrez) had to merge the LaTeX code of the solution we have written and the code the assistants sent to me and it means that there might still be some typo errors due to commands that were not the same between the two documents.

Note: the distribution of the exercises changes from year to year. Here, the distribution of the year 2019–2020 is indicated.

APE 1

1.1: Exercise 1 (Vigenère)

You saw in the class the Vigenère encryption scheme. Write it formally as $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, with a key of length t, with $4 \leq t \leq 20$.

1.2: Exercise 2 (Perfect secrecy.)

We define the following encryption scheme for messages, keys and ciphertexts in $\mathbb{Z}_n$, where $\mathbb{Z}_n$ is essentially the integers in the interval $[0, n[$ (in fact $(\mathbb{Z}_n, +)$ forms a group):

- Gen outputs a key $k \in \mathcal{K}$ selected uniformly at random.
- $\mathsf{Enc}_k(m) \coloneqq k + m \mod n$
- $\mathsf{Dec}_k(c) \coloneqq c - k \mod n$

Suppose messages are drawn from $\mathcal{M}$ according to the binomial distribution. More precisely $M \sim \mathrm{Bi}(n-1, p)$ for some probability p which means that $\forall m \in \mathcal{M} : \Pr[M = m] = \binom{n-1}{m} p^m (1-p)^{n-1-m}$.

1. Show that the encryption scheme above is perfectly secret.
2. Evaluate $\Pr[C = c]$ for every $c \in \mathcal{C}$.
3. Evaluate $\Pr[K = k | C = c]$ for every $k \in \mathcal{K}$ and $c \in \mathcal{C}$.

1.3: Exercise 4 (Negligible functions.)

1. Let f be a negligible function in n. Show that $g : n \mapsto 1000 \cdot f(n)$ is negligible too.
2. Show that the function $n \mapsto n^{-\log(n)}$ is negligible in n.

1.4: Exercise 5 (Efficiency.)

Explain why the function that maps n on a sequence of "1" of length $\lfloor \sqrt{n} \rfloor$ cannot be evaluated by any efficient algorithm.

An example of such algorithm is given in Algorithm 1.

```
Require: $n \geq 0$
Ensure: A sequence of $\sqrt{n}$ "1"
  for $i = 0$ to $\lfloor \sqrt{n} \rfloor$ do
    Print '1'
  end for
```

Algorithm 1: example of algorithm

Hint: see n as a power of 2.

APE 2

2.1: Exercise 1 (Security model.)

Let ϵ denote a negligible function. Remember that $\Pi := \langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ has *indistinguishable multiple encryption in the presence of eavesdroppers* if $\forall$ PPT $\mathcal{A}$, $\exists\ \epsilon$:

$$\Pr[\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)\,,$$

where $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}(n)$ is defined as follows.

1. $\mathcal{A}$ outputs $M_0 = (m_0^1, \ldots, m_0^t), M_1 = (m_1^1, \ldots, m_1^t)$
2. Choose $k \leftarrow \mathcal{G}(1^n)$ and $b \leftarrow \{0,1\}$, and send $(\mathsf{Enc}_k(m_b^1), \ldots, \mathsf{Enc}_k(m_b^t))$ to $\mathcal{A}$
3. $\mathcal{A}$ outputs b'
4. Define $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}(n) := 1$ iff $b = b'$

Also remember that $\Pi := \langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ has *indistinguishable encryption under a chosen-plaintext attack* if $\forall$ PPT $\mathcal{A}$, $\exists\ \epsilon$:

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)\,,$$

where $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$ is defined as follows.

1. Choose $k \leftarrow \mathsf{Gen}(1^n)$
2. **$\mathcal{A}$ is given oracle access to $\mathsf{Enc}_k(\cdot)$**
3. $\mathcal{A}$ outputs $m_0, m_1 \in \mathcal{M}$
4. Choose $b \leftarrow \{0,1\}$ and send $\mathsf{Enc}_k(m_b)$ to $\mathcal{A}$
5. **$\mathcal{A}$ is again given oracle access to $\mathsf{Enc}_k(\cdot)$**
6. $\mathcal{A}$ outputs b'
7. Define $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) := 1$ iff $b = b'$

Define the concept of indistinguishable *multiple* encryption under a chosen-plaintext attack.

2.2: Exercise 2 (Pseudorandomness.)

Let $F\colon \{0,1\}^* \times \{0,1\}^* \mapsto \{0,1\}^*$ be a (length-preserving) pseudorandom function, that is, if k is selected uniformly at random in $\{0,1\}^n$, then $F_k(\cdot)$ is computationnaly indistinguishable from a function f selected randomly in the set of functions from $\{0,1\}^n$ to $\{0,1\}^n$. More formally, $\forall$ PPT $\mathcal{D}$, $\exists$ negl. ϵ:

$$\left|\Pr[\mathcal{D}^{F_k(\cdot)}(1^n) = 1] - \Pr[\mathcal{D}^{f(\cdot)}(1^n) = 1]\right| \leq \epsilon(n)$$

Show that F cannot seem random in front of an adversary who has an unbounded computational power, in the sense that she can distinguish it from a random function.

2.3: Exercise 3 (Reduction.)

Let $\Pi := \langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be an encryption scheme having indistinguishable encryption under a chosen plaintext attack. Suppose we define a new scheme $\Pi' := \langle \mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}' \rangle$ as follows.

- $\mathsf{Gen}' := \mathsf{Gen}$
- $\mathsf{Enc}'_k(m) := \mathsf{Enc}_k(m)||1$ (i.e. a '1' bit is appended to the ciphertext)
- $\mathsf{Dec}'_k(c) := \mathsf{Dec}_k(c_1)$, where c_1 is obtained by discarding the last bit of c.

Is Π' also a CPA secure encryption scheme? Provide either an (efficient) attack/adversary or a (polynomial) reduction, depending on your claim.

2.4: Exercise 4 (Reduction and/or attacks.)

Let $\Pi_1 = \langle \mathsf{Gen}^1, \mathsf{Enc}^1, \mathsf{Dec}^1 \rangle$ and $\Pi^2 = \langle \mathsf{Gen}^2, \mathsf{Enc}^2, \mathsf{Dec}^2 \rangle$ be an encryption scheme with $\mathsf{Enc}^1 \colon \mathcal{K} \times \mathcal{M}^1 \mapsto \mathcal{C}^1$ and $\mathsf{Enc}^2 \colon \mathcal{K} \times \mathcal{M}^2 \mapsto \mathcal{C}^2$

a. If $\mathcal{C}^1 = \mathcal{M}^2$, let $\Pi = \langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ with

- $\mathsf{Gen} := (\mathsf{Gen}_1, \mathsf{Gen}_2)$ (that is, we obtain two different keys (k_1, k_2)
- $\mathsf{Enc}_{(k_1,k_2)}(m) := \mathsf{Enc}^2_{k_2}(\mathsf{Enc}^1_{k_1}(m))$
- $\mathsf{Dec}_{(k_1,k_2)}(c) := \mathsf{Dec}^1_{k_1}(\mathsf{Dec}^2_{k_2}(c))$

1. If Π^1 is CPA secure, is it Π CPA secure?
2. If Π^2 is CPA secure, is it Π CPA secure?
3. If Π is CPA secure, is it Π^1 CPA secure?
4. If Π is CPA secure, is it Π^2 CPA secure?

b. If $\mathcal{M}^1 = \mathcal{M}^2$ and $\mathcal{C}^1 = \mathcal{C}^2$. let $\Pi' = \langle \mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}' \rangle$ with

- $\mathsf{Gen}' := (\mathsf{Gen}^1, \mathsf{Gen}^2)$ (that is, we obtain two different keys (k_1, k_2)
- $\mathsf{Enc}'_{(k_1,k_2)}(m) := (c_1, c_2)$ with $c_1 = \mathsf{Enc}^1_{k_1}(m),\ c_2 = \mathsf{Enc}^2_{k_2}(m))$
- $\mathsf{Dec}'_{(k_1,k_2)}(c) := \mathsf{Dec}_{k_1}(c_1)$ with $c = c_1 \| c_2$ (c_1 is the first half of c)

5. If Π^1 is CPA secure, is it Π' CPA secure?
6. If Π^2 is CPA secure, is it Π' CPA secure?
7. If Π' is CPA secure, is it Π^1 CPA secure?
8. If Π' is CPA secure, is it Π^2 CPA secure?

APE 3

3.1: Exercise 0 (Simple attacks)

Let $\mathsf{MAC} = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ be existentially unforgeable under an adaptive chosen-message attack and let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a CCA-secure scheme. Consider the following schemes $\mathsf{MAC}' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ (resp. $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$) based on Mac with $\mathsf{Gen}' = \mathsf{Gen}$ and Mac' (resp. $\mathsf{Gen} = \mathsf{Gen}'$ and Enc') defined as follow:

1. $\mathsf{Mac}'_k(m) := (\mathsf{Mac}_k(m), \mathsf{Mac}_k(m \oplus 0\ldots01))$
2. $\mathsf{Mac}'_k(m) := \mathsf{Mac}_k\left(\bigoplus_{i=1}^{l} m_i\right)$
3. $\mathsf{Mac}'_k(m) := (\mathsf{Mac}_k(m_1), \ldots, \mathsf{Mac}_k(m_l))$
4. $\mathsf{Mac}'_k(m) := (\mathsf{Mac}_k(m_1), \mathsf{Mac}_k(m_1||m_2), \ldots, \mathsf{Mac}_k(m_1|| \ldots ||m_l))$
5. $\mathsf{Enc}'_k(m) := \left(\mathsf{Enc}_k(m), \mathsf{Enc}_k\left(\bigoplus_{i=1}^{l} m_i\right)\right)$
6. $\mathsf{Enc}'_k(m) := \left(\mathsf{Enc}_k(m), \bigoplus_{i=1}^{l} m_i\right)$
7. $\mathsf{Enc}'_k(m) := (\mathsf{Enc}_k(m), \mathsf{Enc}_k(m \oplus 110\ldots0))$
8. $\mathsf{Enc}'_k(m) := (\mathsf{Enc}_k(m||0), \mathsf{Enc}_k(m))$

Break all MAC' and Π'.

(In some cases m is parsed in $m_1, \ldots, m_l$ with $|m_1| = \cdots = |m_{l-1}| = n$, $|m_l| \leq n$ and $m_1|| \ldots ||m_l = m$ ($||$ is the concatenation) where n is the security parameter.)

3.2: Exercise 1 (Fixed-length MAC)

Consider the fixed-length MAC $\Pi := \langle \mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy} \rangle$ defined as follows:

- Gen: choose random $k \leftarrow \{0,1\}^n$
- Mac: on input $m, k \in \{0,1\}^n$, output $t := F_k(m)$
- Vrfy: on input $k, m, t \in \{0,1\}^n$ output 1 iff $t = F_k(m)$

Prove that, if F is a PRF, Π is existentially unforgeable under an adaptive chosen-message attack. Hint:

1. Consider the scheme Π' defined as Π except that a truly random function is used instead of a pseudo-random one. Show that Π' is existentially unforgeable under an adaptive chosen-message attack.
2. Consider a PPT adversary who can produce an adaptive forgery on Π with non negligible probability $\epsilon(n)$. Using this adversary, show that F cannot be a PRF.

3.3: Exercise 2 (MAC length extension)

Let $\Pi' := \langle \mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}' \rangle$ be a secure fixed-length MAC. We define a variable-length MAC $\Pi := \langle \mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy} \rangle$ as follows:

- Gen: choose random $k \leftarrow \{0,1\}^n$
- Mac: on input $k \in \{0,1\}^n$ and $m \in \{0,1\}^*$ of length $l < 2^{\frac{n}{4}}$

  - Parse m into blocks $m_1, \ldots, m_d$ of length $\frac{n}{4}$ each (pad with 0's if necessary)
  - Choose random $r \leftarrow \{0,1\}^{\frac{n}{4}}$
  - Compute $t_i := \mathsf{Mac}_k(r||l||i||m_i)$ for $1 \le i \le d$, with $|r| = |l| = |i| = \frac{n}{4}$
  - Output $t := \langle r, t_1, \ldots, t_d \rangle$

- Vrfy: on input $k, m, t = \langle r, t_1, \ldots, t_{d'} \rangle$,
  - Parse m into blocks $m_1, \ldots, m_d$ of length $\frac{n}{4}$ each
  - Output 1 iff $d = d'$ and, $\forall 1 \le i \le d$, $\mathsf{Vrfy}'_k(r||l||i||m_i, t_i) = 1$

The goal of this exercise is to prove by reduction that Π is existentially unforgeable. Let $\mathcal{A}$ (resp. $\mathcal{A}'$) be an adversary attacking the unforgeability of Π (resp. Π') and let $\epsilon = \text{MacForge}_{\mathcal{A},\Pi}(n)$ (resp. $\epsilon' = \text{MacForge}_{\mathcal{A}',\Pi'}(n)$) denotes its advantage.

1. Make a quick draw sketching the proof.
2. Describe formally how $\mathcal{A}'$ should react to a query $\text{Mac}_k(m)$.
3. Define what is a mac forgery for the scheme Π. Does it necessarily implie a forgery on the scheme Π' (justify your answer).
4. Express ϵ in function of ϵ' and conclude.

3.4: Exercise 3 (Authenticated encryption and sPRP)

42

3.5: Exercise 4 (Authenticated encryption)

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an authenticated encryption scheme where $0 \notin \mathcal{C}$ (that is, the string "0" is not a possible ciphertext for Π). Consider the following scheme $\Pi' := (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ with:

- $\mathsf{Gen}' = \mathsf{Gen}$
- $\mathsf{Enc}' = \mathsf{Enc}$
- $\forall k : \mathsf{Dec}'(c) = \begin{cases} \mathsf{Dec}(c) \text{ if } c \neq 0, \\ 0 \text{ if } c = 0. \end{cases}$

1. Is Π' unforgeable?
2. Is Π' CCA secure?

3.6: Exercise 5

Let F be a PRF. Below, we describe three *insecure variable-length* message authentication codes (*a.k.a.* MACs), Π_1, Π_2 and Π_3, which all use the same key generation algorithm $\mathcal{G}$. The message space is *any (non negative) number* of message blocks in $\{0,1\}^n$, where n is the security parameter.

$\mathcal{G}(1^n)$ outputs a random key $k \leftarrow \{0,1\}^n$.

The scheme Π_3 is built from Π_2 which is itself built from Π_1 as an (unsuccessful) attempt to "patch" the previous scheme:

$\Pi_1 = (\mathsf{Gen}, \mathsf{Mac}^1, \mathsf{Vrfy}^1)$: *"Deterministic MAC – Chaining PRFs"*

$\mathsf{Mac}^1_k(m_1, \ldots, m_\ell)$ computes $t_1 = F_k(m_1)$ as well as $t_i = F_k(m_i \oplus t_{i-1})$, for $i = 2$ to ℓ, and returns $t := t_\ell$ (note that only the last block is returned).

$\mathsf{Vrfy}^1_k((m_1, \ldots, m_\ell), t)$ outputs 1 if $\mathsf{Mac}^1_k(m_1, \ldots, m_\ell) = t$, and 0 otherwise.

$\Pi_2 = (\mathsf{Gen}, \mathsf{Mac}^2, \mathsf{Vrfy}^2)$: *"Padding a random message block in the end"*

$\mathsf{Mac}^2_k(m_1, \ldots, m_\ell)$ first picks a random $r \leftarrow \{0,1\}^n$ and then runs $t = \mathsf{Mac}^1_k(m_1, \ldots, m_\ell, r)$ and outputs (r, t).

$\mathsf{Vrfy}^2_k((m_1, \ldots, m_\ell), (r, t))$ outputs 1 if $\mathsf{Mac}^1_k(m_1, \ldots, m_\ell, r) = t$, and 0 otherwise.

$\Pi_3 = (\mathsf{Gen}, \mathsf{Mac}^3, \mathsf{Vrfy}^3)$: *"Padding a random message block in the beginning"*

$\mathsf{Mac}^3_k(m_1, \ldots, m_\ell)$ first picks a random $s \leftarrow \{0,1\}^n$ and then runs $(r, t) = \mathsf{Mac}^2_k(s, m_1, \ldots, m_\ell)$ and outputs (r, s, t).

$\mathsf{Vrfy}^3_k((m_1, \ldots, m_\ell), (r, s, t))$ outputs 1 if $\mathsf{Mac}^1_k(s, m_1, \ldots, m_\ell, r) = t$, and 0 otherwise.

1. Describe $\mathsf{Mac}^3_k(m_1, \ldots, m_\ell)$ explicitly in term of computations of F_k (and $\oplus$ of course).
2. Show the correctness of Π_3.
3. Mount a forgery attack on these MACs.

3.7: Exercise 6

Let F be a pseudorandom function, G be a pseudorandom permutation, T be a public n-bit constant, k be a n-bit secret key, m be a n-bit message, IV be a n-bit random value chosen by the party computing the encryption (resp. MAC) before each operation. Among the following constructions, determine the ones that would be acceptable and justify your answer. (Your justifications can rely on results that have been presented during the class.)

1. $\mathsf{Enc}_k(m) := F_k(m \oplus T)$ as an encryption scheme secure against eavesdropping.
2. $\mathsf{Enc}_k(m) := G_k(m \oplus T)$ as an encryption scheme secure against eavesdropping.
3. $\mathsf{Enc}_k(m) := G_k(m \oplus T)$ as an encryption scheme secure against a CPA-adversary.
4. $\mathsf{Enc}_k(m) := (IV, G_k(m \oplus T \oplus IV))$ as an encryption scheme secure against a CPA-adversary.
5. $\mathsf{Mac}_k(m) := F_k(m \oplus T)$ as a MAC scheme existentially unforgeable under an adaptive chosen-message attack.
6. $\mathsf{Mac}_k(m) := (IV, G_k(m \oplus IV \oplus T))$ as a MAC scheme existentially unforgeable under an adaptive chosen-message attack.

3.8: Exercise 7 (Blue-ray security)

The movie industry wants to protect digital content distributed on DVD's. We study one possible approach. Suppose there are at most a total of n DVD players in the world (e.g. $n = 2^{32}$). We view these n players as the leaves of a binary tree of height $\log_2 n$. Each node ν_j in this binary tree contains an AES key K_j such that $\mathrm{Enc}_{K_j} : \{0,1\}^l \to \{0,1\}^l$ is assumed to be a *secure* encryption. These keys are kept secret from consumers and are fixed for all time. At manufacturing time each DVD player is assigned a serial number $i \in [0, n-1]$. Consider the set S_i of $\log_2(n+1)$ nodes along the path from the root to leaf number i in the binary tree. The manufacturer of the DVD player embeds in player number i the $\log_2(n+1)$ keys associated with the nodes in S_i. In this way each DVD player ships with $\log_2(n+1)$ keys embedded in it (these keys are supposedly inaccessible to consumers).

1. Since all DVD players have the key *root* (noted K_{root}), find a way to protect the content $M \in \{0,1\}^l$ of a DVD such that all players can decrypt the movie (and then read it).
2. Now suppose that a hacker has been able to extract the key K_{root} embedded in his DVD player and has published it on the Internet. Show how the movie industry can encrypt the contents of a new DVD $M \in \{0,1\}^l$ such that only the owners of a DVD player can read it. Note that the movie indutry does not want to produce several encryptions of the same content M *i.e.* there will be a single manner to protect the DVD.

3. Suppose the $\log_2 n$ keys embedded in DVD player number r are exposed by hackers and published on the Internet. Show that when the movie industry is about to distribute a new DVD movie they can encrypt the contents of the DVD using a ciphertext of size $l \cdot (1 + \log_2 n)$ so that all DVD players can decrypt the movie except for player number r. In effect, the movie industry disables player number r.

   Hint: the DVD will contain $\log_2 n$ *ciphertexts where each ciphertext is the encryption of a unique* K *under certain* $\log_2 n$ *keys from the binary tree.*

3.9: Exercise 8 (Authenticated encryption, or not)

101

3.10: Exercise 9 (Authenticated encryption)

41

3.11: Exercise X (Mode of operation)

Show formally that ECB-mode encryption does not have indistinguishable encryptions in the presence of an eavesdropper.

APE 4

4.1: Exercise 1 (Authenticated encryption, August 2019 exam)

1. Let $F\colon \mathcal{K} \times \{0,1\}^n \mapsto \{0,1\}^n$ be a PRF. Consider the following Authenticated Encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follow:

   - Gen: pick a key uniformly at random in $\mathcal{K}$;
   - Enc: on input m and k:
     - Parse m in l blocks $m_1, \dots, m_l$ with $|m_1| = \cdots = |m_l| = n$
     - Pick r uniformly at random in $\{0,1\}^{\frac{n}{2}}$
     - For $i = 1, \dots, l$:
       $c_i = F_k(r||i) \oplus m_i$
     - $c_{l+1} = F_k(r||l+1) \oplus \left(\oplus_{i=1}^{l} m_i\right)$
     - Return (r, c) with $c = (c_1, \dots, c_l, c_{l+1})$.
   - Dec consequently.

   [For simplicity we suppose that for every message all blocks are *full*, that is, when parsed the last block has length n (i.e., $|m_l| = n$).

   When we write r||i we mean that the number i is written in binary notation putting as many zeros on the left as necessary.]

   (a) Is Π unforgeable? Prove or confute[1].

   (b) Is Π CCA-secure? Prove or confute with an attack. [Hint: the previous answer may be useful…]

2. Let $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ be an authenticated encryption scheme with binary messages of length n. For two binary vectors of length n we denote $\oplus$ the coordinate-wise XOR. 1_n denotes the all-1 vector of length n. Consider the following schemes:

   - $\Pi^1 := (\mathsf{Gen}^1, \mathsf{Enc}^1, \mathsf{Dec}^1)$:
     - $\mathsf{Gen}^1 := \mathsf{Gen}'$,
     - $\mathsf{Enc}^1_k(m) := (c_1, c_2) = (\mathsf{Enc}'_k(m), \mathsf{Enc}'_k(m \oplus 1_n))$,
     - $\mathsf{Dec}^1_k((c_1, c_2)) := \mathsf{Dec}'_k(c_1)$ if $\mathsf{Dec}'_k(c_1) \oplus \mathsf{Dec}'_k(c_2) = 1_n$, $\perp$ otherwise.
   - $\Pi^2 := (\mathsf{Gen}^2, \mathsf{Enc}^2, \mathsf{Dec}^2)$:
     - $\mathsf{Gen}^2 := \mathsf{Gen}'$,
     - $\mathsf{Enc}^2_k(m) := \mathsf{Enc}'_k(m \oplus 1_n)$,
     - $\mathsf{Dec}^2_k(c) := 1_n$ if $\mathsf{Dec}'_k(c) = \perp$, $\mathsf{Dec}'_k(c) \oplus 1_n$ otherwise.

   (a) Is Π^1 an authenticated encryption scheme? If not, explain which property you can break and how.

   (b) Is Π^2 an authenticated encryption scheme? If not, explain which property you can break and how.

[1] *refute

4.2: Exercise 2 (Authenticated Encryption and sPRP)

Consider the following scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ based on the strong pseudorandom permutation $\mathsf{F}\colon \mathcal{K} \times \{0,1\}^n$, defined as follow:

- $\mathcal{M} = \{0,1\}^{\frac{n}{2}}$ (the message space)
- Gen picks a random key $k \in \mathcal{K}$
- $\mathsf{Enc}_k(m)$ picks a random value $r \in \{0,1\}^{\frac{n}{2}}$, and computes $c := \mathsf{F}_k(m\|r)$
- $\mathsf{Dec}_k(c)$ computes $(m\|r) = \mathsf{F}_k^{-1}(c)$ and outputs m (the first half).

Answers the following questions:

- is Π unforgeable?
- is Π CCA-secure? (*To do at home*)
- is Π an authenticated encryption scheme? (*To do at home*)

Definition 1 (*Strong PseudoRandom Permutation*)
A function $\mathsf{F}\colon \mathcal{K} \times \mathcal{M} \mapsto \mathcal{M}$ is a (q, t, ϵ)- *strong pseudorandom permutation* (sprp) if for any (q, t)-bounded adversary, the advantage:

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathsf{adv}} := \left|\Pr\left[\mathsf{adv}^{\mathsf{F}_k(\cdot),\mathsf{F}_k^{-1}(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathsf{adv}^{\mathsf{f}(\cdot,\cdot),\mathsf{f}^{-1}(\cdot,\cdot)} \Rightarrow 1\right]\right| \leq \epsilon$$

with k and f picked uniformly at random from their domains, respectively $\mathcal{K}$ and the set of permutations $\mathcal{M} \mapsto \mathcal{M}$.

4.3: Exercise 3 (Hash functions from...hash functions)

Let $H_2\colon \{0,1\}^{2l} \mapsto \{0,1\}^l$ and $H_3\colon \{0,1\}^{3l} \mapsto \{0,1\}^l$ be collision resistant hash functions. For $2l$-bit strings x_i's, consider the following two constructions.

- $H_4\colon \{0,1\}^{4l} \mapsto \{0,1\}^l$; $x = x_1||x_2 \to H_2\left(H_2(x_1)||H_2(x_1 \oplus x_2)\right)$
- $H_6\colon \{0,1\}^{6l} \mapsto \{0,1\}^l$; $x = x_1||x_2||x_3 \to H_3\left(H_2(x_1 \oplus x_2)||H_2(x_2 \oplus x_3)||H_2(x_3 \oplus x_1)\right)$

Determine whether these hash functions are still collision resistant or not.

4.4: Exercise 4 (Block-cipher based hash function)

Considering a block cipher $E\colon \mathcal{K} \times \mathcal{M} \mapsto \mathcal{C}$; $(k, m) \to E(k, m) = \mathsf{Enc}_k(m)$ with $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^l$, one may try to construct a collision resistant compression function from $\{0,1\}^{2l}$ to $\{0,1\}^l$. Show that the following methods do not work :

$$f_1(x, y) = E(y, x) \oplus y \quad \text{ and } \quad f_2(x, y) = E(x, x) \oplus y$$

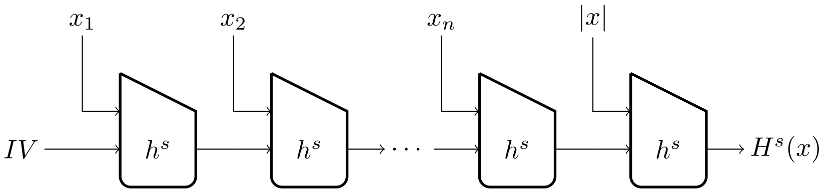
That is, show an efficient algorithm for constructing collisions for f_1 and f_2. Recall that the block cipher E and the corresponding decryption algorithm D are both known to you (and they are bijective functions).

4.5: Exercise 5 (Authenticated encryption, or not)

101

4.6: Exercise 6 (Variable-length MAC)

Considering a known hash function $h^s\colon \{0,1\}^{2l} \mapsto \{0,1\}^l$, let's note by H^s the corresponding Merkle-Damgård transform hash function, *i.e.*



when $x = x_1 || \cdots || x_n$ for some integer n and when the x_i's are l-bit strings.

Show why, with a private key k of length l, the MAC scheme

$$t := H^s(k||m),$$

is *not* existentially unforgeable under an adaptive chosen-message attack.

4.7: Exercise 7 (Hash-MAC)

Suppose H_0 and H_1 are compression functions but only one is believed to be collision resistant. Besides, suppose Mac_0 and Mac_1 are message authentication codes but only one of the both schemes is known to be unforgeable. Is it possible to build a secure "hash-MAC" from these inputs? Justify your answer.

4.8: Exercise 8 (Blue-ray security)

37

APE 5

The TP5 of 2019-2020 reviewed the basics of number theory and group theory for this course. As they are not really useful for the exam and are fairly trivial (once the basics are know, that is), they are skipped in this document.

Below are the exercises of last year pertaining to groups and number theory.

5.1: Exercise 0 (Group order)

35

5.2: Exercise 3 (Euclidean algorithm for gcd)

Let $a, b \in \mathbb{Z}$, $b \neq 0$, consider the following algorithm, presented in Algorithm 2. ($r = a\%b$ means that $a = qb + r$ where q is the quotient and r is the remainder).

Prove that x, the value returned by Algorithm 2, is $\mathsf{gcd}(a, b)$.

Hint:

- Prove that x divides $\mathsf{gcd}(a, b)$
- Prove that $\mathsf{gcd}(a, b)$ divides x

```
Input: $a$, $b$
Output: $\mathsf{gcd}(a, b)$
while $b \neq 0$ do
    $r \leftarrow a\%b$;
    $a \leftarrow b$;
    $b \leftarrow r$;
end
return ($a$)
```

Algorithm 2: The Euclidean gcd algorithm.

5.3: Exercise 4

Consider the group $\mathbb{Z}_{17}^*$.

1. Compute 5^{-1}.
2. Compute 3^2, 3^3 and 3^4.
3. Does 3 generate the group?
4. Find $\log_7(11)$.

5.4: Exercise 5 (Group order)

In this exercise we consider the group $\mathbb{Z}_{59}^*$.

1. What is the order of 58?
2. What are the possible orders of an element of this group?
3. Find an element of order more than 20.

APE 6

6.1: Exercise 1 (ElGamal Public Key Encryption and CCA Security)

1. Write the security definition of CCA security for a public key encryption scheme.
2. Let (c_1, c_2) and (c'_1, c'_2) be two ElGamal ciphertexts, of plaintext m and m' respectively. Can $(c_1c'_1, c_2c'_2)$ be a ciphertext relatively to this scheme?
3. From (c_1, c_2) a ciphertext of m, can you build another ciphertext valid for m (remember that the public key is $(\mathbb{G}, g, q, h = g^x)$)? If yes, what is its decryption?
4. Show that ElGamal Public Key Encryption is not CCA secure.

6.2: Exercise 2 (A Variation of ElGamal in PKE)

Let consider ElGamal public encryption scheme with Encryption algorithm modified in the following way, where $\mathcal{M} = \{0, 1\}$:

- If $b = 0$ then choose a uniform $y \in \mathbb{Z}_q$ set $c_1 = g^y$, $c_2 = h^y$, and the ciphertext is (c_1, c_2).
- If $b = 1$ then choose independent uniform $y, z \in \mathbb{Z}_q$ set $c_1 = g^y$, $c_2 = g^z$, and the ciphertext is (c_1, c_2).

1. How is it possible to decrypt correctly such ciphertexts with the private key?
2. Show that this scheme is CPA secure if DDH holds in $\mathbb{G}$.

6.3: Exercise 3 (Decisional Diffie-Hellman, $\mathbb{Z}_p^*$, and QR_p)

74

6.4: Exercise 4 (A variation of ElGamal in QR_p)

Let $p = 2q + 1$ with q prime, let $\mathbb{G} = QR_p$ the group of squares modulo p, and g be a generator of $\mathbb{G}$. We define ElGamal encryption scheme in this group: The private key is $(\mathbb{G}, g, q, x)$, the public key is $(\mathbb{G}, g, q, h = g^x)$ where $x \in \mathbb{Z}_q^*$ is chosen uniformly. To encrypt a message $m \in \mathbb{Z}_q$, choose a uniform $r \in \mathbb{Z}_q$, compute $c_1 = g^r \mod p$ and $c_2 = h^r + m \mod p$ and let the ciphertext be (c_1, c_2).

1. What is the order of g?
2. Is this scheme CPA-secure?

6.5: Exercise 5 (DDH PRG)

Let $\mathbb{G}$ be a cyclic group of prime order q generated by $g \in \mathbb{G}$. Consider the following PRG defined over $(\mathbb{Z}^2 - q, \mathbb{G}^3)$: $G(\alpha, \beta) := (g^\alpha, g^\beta, g^{\alpha\beta})$. Define what it means for a PRG over $(\mathbb{Z}^2 - q, \mathbb{G}^3)$ to be secure and show that G is a secure PRG assuming DDH holds in $\mathbb{G}$.

6.6: Exercise X (A variant of ElGamal Encryption.)

Let us consider the following variant of ElGamal encryption. Let

- Gen output a pair $\langle pk, sk \rangle := \langle (\mathsf{Gr}, q, g, h), (\mathsf{Gr}, q, g, x) \rangle$ as in traditional ElGamal encryption, except that x is selected in $\mathbb{Z}_q - \{0\}$;
- $\mathsf{Enc}_{pk}(m) := \langle m \cdot g^y, h^y \rangle$ with $y \leftarrow \mathbb{Z}_q$ and $m \in \mathsf{Gr}$.

1. Define the corresponding decryption operation.
2. Why did we exclude “0” from the set in which x is selected?
3. Prove that this variant of ElGamal is CPA-secure if the DDH problem is hard with respect to the group key generation algorithm Gr.

APE 7

7.1: Exercise 1 (Commitment scheme)

Define the bit-commitment scheme $\langle \mathcal{G}, \mathsf{Com}, \mathsf{Open}\rangle$ with the following PPT algorithms:

- $\mathsf{Gen}(1^n)$ sets pk as (PRG, R), where
  - G is a random generator $\{0,1\}^n \longmapsto \{0,1\}^{3n}$
  - R is a random $3n$-bit string
- $\mathsf{Com}_{pk}(b)$ with $b \in \{0,1\}$ provides (c,d) where:
  - Y is an n-bit string
  - if $b = 0$ $c = \mathsf{G}(Y)$
  - if $b = 1$, $c = \mathsf{G}(Y) \oplus R$
  - $d = (b, Y)$
- $\mathsf{Open}_{pk}(c,d)$ outputs b if it can recompute c from d and pk, or $\perp$ otherwise

1. Is this scheme perfectly hiding?
2. Is this scheme computationaly binding?
3. If the committer choose R is the scheme secure?

7.2: Exercise 2 (Commitment with DL)

Let $(\mathbb{G}, \cdot)$ be a group in which the discrete logarithm is difficult, with $|\mathbb{G}| = q$. Let g be a generator of the group and h be a random element of the group ((g,h) may be seen as the key of the hash function). Define the following hash funtion $\mathsf{H}\colon \mathbb{Z}_q^* \times \mathbb{Z}_q^* \mapsto \mathsf{G}$:

$$\mathsf{H}_{g,h}(\alpha, \beta) := g^\alpha h^\beta$$

Prove that if the DL is difficult, then, the hash function is collision resistant. For simplicity we assume that q is prime.

7.3: Exercise 3 (Commitment scheme and batching)

90

7.4: Exercise 4 (Decisional Diffie-Hellman and $\mathbb{Z}_p^*$)

The goals of this exercise are to define QR_p, prove some of its properties, and to show that in some groups DDH and CDH assumptions are conjectured not equivalent, as DDH is easy whereas CDH is conjectured to be hard.

1. For all element a of $\mathbb{Z}_{11}^*$, compute $a^2 \mod 11$.

   For a prime number p, we denote QR_p the set $\{x \in \mathbb{Z}_p^* \mid \exists a \in \mathbb{Z}_p^*, a^2 = x\}$, such x are called quadratic residues modulo p. Show that if p is odd then $|QR_p| = \frac{p-1}{2}$.
2. Show that, if p is odd, QR_p is a cyclic group (therefore, QR_p is a subgroup of $\mathbb{Z}_p^*$).
3. For all element a of $\mathbb{Z}_{11}^*$, compute $a^5 \mod 11$. Show that for any odd prime p, $x \in QR_p \Leftrightarrow x^{\frac{p-1}{2}} = 1 \mod p$, and that $x \notin QR_p \Leftrightarrow x^{\frac{p-1}{2}} = -1 \mod p$.

4. Show that 2 is a generator of $\mathbb{Z}_{11}^*$. For the following pairs (a, b), compute g^a, g^b and g^{ab} in $\mathbb{Z}_{11}^*$ where $g = 2$:

   - $(2, 8)$,
   - $(1, 4)$,
   - $(3, 5)$.

   Show that for p an odd prime, $g^{ab} \notin QR_p \Leftrightarrow g^a \notin QR_p$ and $g^b \notin QR_p$.

5. Show that DDH does not hold in $\mathbb{Z}_p^*$ with p an odd prime.

7.5: Exercise 5

81

APE 8

8.1: Exercise 1 (Zero knowledge Petersen)

We work in a group $\mathbb{G}$ of prime order q with generator g. The Schnorr protocol, used to prove the knowledge of discrete logarithm, is (honest-verifier) zero-knowledge. However, the value $y = g^x \pmod{p}$ (for a safe prime $p = 2q + 1$) leaks some information about the discrete logarithm x (since for a given generator g of order q there is exactly one such x in $\mathbb{Z}_q$). On the other hand, the Pedersen commitment is perfectly hiding and thus does not reveal information about the committed value. The following protocol attempts to merge the both properties i.e., to prove the knowledge of a commited value under the Pedersen commitment scheme in a zero-knowledge manner.

The protocol. The public inputs of the proof are the prime p, the Pedersen public key (g, h), a security parameter k and a (hypothetic) commitment $c \in QR(p)$. The prover's private intputs are x and r in $\mathbb{Z}_q$ s.t. $c = g^x h^r \pmod{p}$. The protocol executes as follows.

- The prover randomly chooses $y, s \in_R \mathbb{Z}_q$ and sends $d = g^y h^s \pmod{p}$ to the verifier.
- The verifier randomly chooses $e \in_R \{0, 1\}^k$ and sends it to the prover.
- The prover computes $z = y - ex$ and $t = s - er$ modulo q and sends it to the verifier.
- The verifier accepts the proof iff $d = c^e g^z h^t \pmod{p}$.

If the verifier accepts the proof, we say that the conversation $\langle d, e, (z, t)\rangle$ is valid.

1. Prove the correctness property of this construction.
2. Assume that an adversary is able to produce two valid responses for two distinct challenges, under the same commit. How can you use this faculty to extract an opening of c? Discuss the soundness property of the protocol.
3. Assume you are able to "rewind" an adversarial prover who tries to build a valid conversation. How can you use this faculty to extract an opening of c. Which property did you break ? Briefly discuss the soundness property of the protocol.
4. Show how a valid conversation $\langle d, e, (z, t)\rangle$ can be simulated from c, without the use of any private inputs. (Assume that the valid conversation involves honest parties.)
5. Generalize the process to prove the knowledge of an opening to a multi-Pedersen commitment as in exercise 3.

8.2: Exercise 2 (Schnorr ZKP with faulty PRG)

Let us study what happens when the prover of a Schnorr ZKP uses a faulty random generator. this generator is used to choose the secret α used to generate the commitment $c = g^\alpha$, where g is a generator of the group. Assuming that the prover made two proofs, one with secret α_0, and the other one with secret $\alpha_1 = a\alpha_0 + b$, how can you reciver the secret witness, knowing the public values, the transcripts of the proofs, a and b?

8.3: Exercise 3 (Commitment scheme and batching)

90

APE 9

9.1: Exercise 0 (Commitment scheme and batching)

By design secure public-key encryption schemes are perfectly binding commitment schemes (which are also computationally hiding, why?). Then, if perfect hiding property is not a concern, do commitment schemes really consist of a new usefull cryptographic building block? This exercise aims to build a perfectly hiding commitment scheme which supports a *batching* property that encryption schemes cannot achieve.

Let p be a prime and let $g \in QR(p)$ be an element of prime order $q > 2^l$. We let G denote the group generated by g and we let I denote the set of integers $\{1, \ldots, q\}$. Fix n random values $g_1, \ldots, g_n \in G$ and define the commitment function $\mathsf{Com}\colon I^n \mapsto G$ by

$$\mathsf{Com}(x_1, \ldots, x_n\,; r) = g^r g_1^{x_1} g_2^{x_2} \cdots g_n^{x_n}$$

1. Describe formally the commitment scheme. Discuss its efficiency and its correctness.
2. Show that the scheme is computationally binding assuming that DLog is intractable in G. That is, show that an adversary computing two openings of a commitment c for random $g, g_1, \ldots, g_n \in G$ can be used to compute discrete-log in G.

   Hint: given a pair $g, h \in G$ your goal is to find an $\alpha \in \mathbb{Z}_q$ such that $g^\alpha = h \mod p$. Choose $g_1, \ldots, g_n \in G$ so that two valid openings will reveal α.
3. Show that the scheme results in a perfectly hiding commitment on several messages. Compare the size of the construction with respect to an encryption (viewed as a commitment) of all these messages.

9.2: Exercise 1 (Jan 2011 evaluation)

The Digital Signature Standard (DSS, also often called DSA) is one of the most commonly used signature algorithms. Its three algorithms Gen, Sign and Vrfy work as follows.

- Gen: on input 1^n, select prime integers p and q such that $|q| = n$, $q|(p-1)$ and $q^2 \not| (p-1)$, together with an integer g that generates the subgroup of $\mathbb{Z}_p^*$ of prime order q. Also choose a hash function $H\colon \{0,1\}^* \mapsto \mathbb{Z}_q$. Then, select $x \leftarrow \mathbb{Z}_q$ uniformly at random, and compute $y := g^x \mod p$. The public key is $\langle H, p, q, g, y \rangle$, and the private key is $\langle x \rangle$.
- Sign: in order to sign the message $m \in \{0,1\}^*$, choose $k \leftarrow \mathbb{Z}_q^*$ uniformly at random and set $r := [g^k \mod p] \mod q$. Then, compute $s := (H(m) + xr) \cdot k^{-1} \mod q$, and output the signature (r, s).
- Vrfy: compute $u_1 := H(m) \cdot s^{-1} \mod q$ and $u_2 := r \cdot s^{-1} \mod q$, and output 1 if and only if $r = [g^{u_1} y^{u_2} \mod p] \mod q$.

1. Show the correctness of the DSS algorithm.
2. As randomness is an expensive resource, it is proposed to select the random value k once and for all, and to sign all messages using that value of k. Is this variant of DSS secure?

   (Hint: see what you can deduce from the signature of two different messages.)

9.3: Exercise 2 (RSA permutation with modulus 221)

Suppose we decide to use an RSA permutation with modulus 221, we consider RSA encryption scheme, and RSA signature.

1. What is the smallest non trivial public exponent e than can be chosen?

2. Can we choose $e = 11$? What is the corresponding private exponent d? Give the public and private key of the corresponding RSA encryption scheme.
3. Compute $c \coloneqq 219^e \pmod{221}$.
4. Verify that $c^d = 219 \pmod{221}$ as expected.
5. How Alice (owning the private key) could sign a message m? Sign the message $m = 3$ (hint: $22^7 = 61 \pmod{221}$).
6. Is 160 a valid signature for $m = 218$?

9.4: Exercise 3 (Derandomizing signatures)

102

9.5: Exercise 4

Let f be a one-way permutation on $\{0,1\}^\lambda$. Consider the following signature scheme for messages in the set $\{1, \dots, n\}$, where $n \in \mathsf{poly}(\lambda)$:

1. To generate keys, choose $x \leftarrow \{0,1\}^\lambda$ at random and set $y \coloneqq f^n(x)$. The public key is y and the private key is x.
2. To sign message $i \in \{1, \dots, n\}$, output $f^{n-i}(x)$ (where $f^0(x) \overset{\text{def}}{=} x$).
3. To verify signature σ on message i with respect to public key y, check whether $y \overset{?}{=} f^i(\sigma)$.

1. Show that the above is not a one-time signature scheme. Given a signature on a message i, for what messages j can an adversary output a forgery?
2. Prove that no PPT adversary given a signature on i can output a forgery on any message $j > i$ except with negligible probability.
3. Suggest how to modify the scheme so as to obtain a one-time signature scheme.

   Hint: include two values y, y' in the public key.

9.6: Exercise X (Jan 2011 evaluation)

Consider the following one-time signature scheme $\Pi \coloneqq \langle \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy} \rangle$, parameterized by a PPT function $f\colon \{0,1\}^* \mapsto \{0,1\}^*$.

- Gen: on input 1^n, select $(x_0, x_1) \leftarrow \{0,1\}^n \times \{0,1\}^n$ uniformly at random, compute $(y_0, y_1) \coloneqq (f(x_0), f(x_1))$ and output the pair $(pk, sk) \coloneqq ((y_0, y_1), (x_0, x_1))$.
- Sign: the signature σ of the bit m is x_m.
- Vrfy: on input (m, σ), output 1 iff $y_m = f(\sigma)$.

Show that if Π is existentially unforgeable under a single-message attack, then f is a one-way function.

APE 10

10.1: Exercise 1 (Authenticated encryption, or not; August Exam)

Let $\Pi := \langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be an authenticated encryption scheme such that Enc encrypts messages of n bits. Do the following systems provide authenticated encryption? For those that do, briefly explain why. For those that do not, present an attack that breaks one of the security properties of an authenticated encryption scheme.

1. $\Pi' := \langle \mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}' \rangle$ with $\mathsf{Enc}'_k(m) = (\mathsf{Enc}_k(m), \mathsf{Enc}_k(m \oplus (0^{n-1} \| 1)))$ and $\mathsf{Dec}'_k(c_1, c_2) = \mathsf{Dec}_k(c_1)$ if $\mathsf{Dec}_k(c_1) \oplus \mathsf{Dec}_k(c_2) = 0^{n-1} \| 1$ and $\perp$ otherwise.
2. $\Pi' := \langle \mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}' \rangle$ with $\mathsf{Enc}'_k(m) = (\mathsf{Enc}_k(m), \mathsf{Mac}_k(m))$ and $\mathsf{Dec}'_k(c_1, c_2) = \mathsf{Dec}_k(c_1)$

   if $\mathsf{Vrfy}_k(\mathsf{Dec}_k(c_1), c_2) = 1$ and $\perp$ otherwise. Here, Mac and Vrfy are deterministic algorithms that are part of a secure MAC scheme that is compatible with Gen.

10.2: Exercise 2 (Derandomizing signatures)

Let $S = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be an EUF-CMA signature scheme defined over (M, Σ), where the signing algorithm Sign is probabilistic. In particular, algorithm Sign uses randomness chosen from a space R. We let $\mathsf{S}(sk, m; r)$ denote the execution of algorithm S with randomness r. Let F be a secure PRF defined over (K, M, R). Show that the following signature scheme with deterministic signing $S' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vrfy})$ is EUF-CMA:

$$\mathsf{G}'(1^n) := \{(pk, sk) \leftarrow \mathcal{G}(1^n), \qquad k \leftarrow K, \qquad sk' := (sk, k), \qquad \text{output } (pk, sk')\}\,;$$

$$\mathsf{Sign}'((sk, k), m) := \{r \leftarrow F_k(m), \qquad \sigma \leftarrow \mathsf{S}(sk, m; r), \qquad \text{output } \sigma\}\,.$$

(Hint: Define S'' which is like S' byt uses a perfect random function. Make a reduction of the security of S'' to the security of S, then build a PRF distinguisher based on a adversary against the signature. Finally, compute the link of the advantages of three relevant games.)

10.3: Exercise 3 (Jan 11 evaluation)

91

References