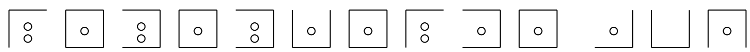
Security Models

2017/2018

P. Lafourcade and M. Giraud

Session 1 – History, Public-key cryptography and Signature

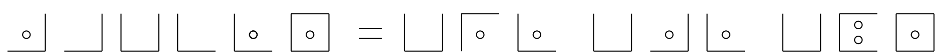
Exercise 1



History: This cipher is written on the tomb in the cimetery of Trinity Churchyard (New York) since 1794. It has been decrypted in 1896.

1. Using the hint (ASTUCE in French) to find the original message ?
2. Can you deduce how this encryption works ?

ASTUCE = TIC TAC TOE



Solution :

1. *Substitution.*
2. *Not enought cipher text to perform a frequency analysis.*
3. *REMEMBER DEATH*
4. *The substitution is designed by this scheme:*

Exercise 2

We recall RSA encryption.

- Public key: (n, e), where $n = pq$, $\Phi(n) = (p-1)(q-1)$ and $\gcd(e, \Phi(n)) = 1$.
- Private key: d, such that $d.e = 1 \mod \Phi(n)$
- Encryption of M: $c = M^e \mod n$
- Decryption of c: $M = c^d \mod n$

1. Let $p = 3$, $q = 7$, compute n and $\Phi(n)$.
2. Let $e = 5$, encrypt the message $M = 2$.
3. Let $d = 5$, decrypt the cipher $c = 3$.
4. Recall what is the integer factorisation problem.

5. Show that if we know how to solve the integer factorisation problem, we know how to decryption any RSA cipher.

Solution :

1. *$n = 21$, $\Phi(n) = 12$*
2. *$c = M^e \mod n = 2^5 \mod 21 = 32 \mod 21 = 11$*
3. *$M = c^d \mod n = 3^5 \mod 21 = 3.3.3.3.3 \mod 21 = 27.9 \mod 21 = 6.9 \mod 21 = 54 \mod 21 = 12$*
4. *$n = pq \rightarrow p$ and q*
5. *we can compute d knowing p and q*

Exercise 3

We recall ElGamal encryption.

- Private key: a and public key: (p, g, h), where $h = g^a \mod p$.
- Encryption of M: Select a random number r and compute $(u, v) = (g^r \mod p, Mh^r \mod p)$
- Decryption: $M \equiv_p \frac{v}{u^a}$

1. Let $a = 2$ and $(p, g) = (5, 3)$, compute h and decrypt the cipher $c = (4, 2)$.
2. The random number $r = 2$ was used to compute the cipher c. Check that the message found in the previous question give c if $r = 2$ is used.
3. Recall the dicrete logarithm problem.
4. Show that the security of ElGamal relies on this problem.

Solution :

1. *Soit $a = 2$ et $(p, g) = (5, 3)$ les paramètres privé et publique d'un chiffrement d'Elgamal.*
   $4 = h = g^a \mod p = 3^2 \mod 5 = 9 \mod 5$.
   $r = 2$, $M = 2$ $(u, v) = (g^r, Mh^r) = (3^2 \mod 5, 2 \times 4^2 \mod 5) = (4, 2)$
   $\frac{v}{u^a} = \frac{2}{4^2} = \frac{2}{1} = 2$
2. *Rapeler ce qu'est le problème du logarithme discret.*
3. *Montrer que si l'on sait résoudre le logarithem discrét alors on sait déchiffremt le chiffrement d'Elgamal.*

Exercise 4

We recall RSA signature.

- Public key: (n, e), where $n = pq$, $\Phi(n) = (p - 1)(q - 1)$ and $\gcd(e, \Phi(n)) = 1$.
- Private key: d, such that $d.e = 1 \mod \Phi(n)$
- Signature of M: $s = M^d \mod n$

- Verification: 1 if $M = s^e \mod n$, 0 otherwise.

1. Let $p = 3$ and $q = 5$. Compute (e, d) such that $\gcd(e, \Phi(n)) = 1$ and $d.e = 1 \mod \Phi(n)$.
2. Give the signature of the message $M = 2$.

Solution :

1. $(e, d) = (3, 3)$
2. $s = 2^3 \mod 15 = 8$ *and* $8^3 \mod 15 = 2$

Exercise 5
Let consider the two following hash functions: $G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0}$ and $H : \{0,1\}^{k-k_0} \to \{0,1\}^{k_0}$. To compute the OAEP-RSA cipher $c = E_{pk}(m, r)$ of the message $m \in \{0,1\}^n$, and $r \leftarrow \{0,1\}^{k_0}$ is computed as follows:

- $s = (m||0^{k_1}) \oplus G(r)$
- $t = r \oplus H(s)$

And the cipher is $c = f(s, t)$, where the fonction f is RSA encryption. Find the decryption function, knowing an inverse function f^{-1} of RSA ?

Solution : $D_{sk}(c)$

- $g(c) = (s, t)$
- $r = t \oplus H(s)$
- $M = s \oplus G(r)$

If $[M]_{k_1} = 0^{k_1}$, *the algorithm returns* $[M]^n$, *otherwise it returns "Reject"*

- $[M]_{k_1}$ *denotes the* k_1 *least significant bits of* M
- $[M]^n$ *denotes the* n *most significant bits of* M

Exercise 6
Zheng and Seberry in 1993 proposed the following encryption scheme:

$$f(r)||(G(r) \oplus (x||H(x)))$$

where x is the plain text, f is a one way trap-door function (like RSA), G and H are two public hash functions, $||$ denotes the concatenation of bitstrings and $\oplus$ is the exclusive-or operator.

- Give the associated decryption algorithm.

Solution :

- *Give the associated decryption algorithm. First uncrypt* $f(r)$ *to get* r, *then compute* $G(r)$ *and xor the result with* $G(r) \oplus (x||H(x))$ *to get* $x||H(x)$ *then check if the application of* H *on the first element of the concatenation is equal to the second to be sure that you get the right plaintext.*

Session 2 – Security Notions

Exercise 7
Prove that $DDH \leq CDH \leq DL$

Solution : *First we recall:*

$$\mathbf{Adv}^{DL}(\mathcal{A}) = Pr\Big[\mathcal{A}(g^x) \rightarrow x \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big]$$

$$\mathbf{Adv}^{CDH}(\mathcal{A}) = Pr\Big[\mathcal{A}(g^x, g^y) \rightarrow g^{xy} \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big]$$

$$\mathbf{Adv}^{DDH}(\mathcal{A}) = Pr\Big[\mathcal{A}(g^x, g^y, g^{xy}) \rightarrow 1 \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big]$$
$$-Pr\Big[\mathcal{A}(g^x, g^y, g^r) \rightarrow 1 \Big| x, y, r \stackrel{R}{\leftarrow} [1,q]\Big]$$

1. $CDH \leq DL$, *Let $\mathcal{A}$ be an adversary against the DL assumption. Then adversary $\mathcal{B}$ against the CDH is designed as follows:*

   Adversary $\mathcal{B}(X, Y)$:
   Run $x = \mathcal{A}(X)$ **then return** Y^x

   The advantage of adversary $\mathcal{B}$ is given by:

$$\begin{aligned}\mathbf{Adv}^{CDH}(\mathcal{B}) &= Pr\Big[\mathcal{B}(g^x, g^y) \rightarrow g^{xy} \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big] \\ &= Pr\Big[v \leftarrow \mathcal{A}(g^x) : (g^y)^v = g^{xy} \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big] \\ &= Pr\Big[v \leftarrow \mathcal{A}(g^x) : v = x \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big] \\ &= Pr\Big[x \leftarrow \mathcal{A}(g^x) \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big] \\ &= \mathbf{Adv}^{DL}(\mathcal{A})\end{aligned}$$

2. $DDH \leq CDH$ *Let $\mathcal{A}$ be an adversary against the CDH assumption. Then adversary $\mathcal{B}$ against DDH is designed as follows:*

   Adversary $\mathcal{B}(X, Y, Z)$:
   if $Z = \mathcal{A}(X, Y)$ **then return** 1
   else return 0

   The advantage of adversary $\mathcal{B}$ is given by:

$$\begin{aligned}\mathbf{Adv}^{DDH}(\mathcal{B}) &= Pr\Big[\mathcal{B}(g^x, g^y, g^{xy}) \rightarrow 1 \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big] - Pr\Big[\mathcal{B}(g^x, g^y, g^r) \rightarrow 1 \Big| x, y, r \stackrel{R}{\leftarrow} [1,q]\Big] \\ &= Pr\Big[\mathcal{A}(g^x, g^y) \rightarrow g^{xy} \Big| x, y \stackrel{R}{\leftarrow} [1,q]\Big] - Pr\Big[\mathcal{A}(g^x, g^y) \rightarrow g^r \Big| x, y, r \stackrel{R}{\leftarrow} [1,q]\Big] \\ &= \mathbf{Adv}^{CDH}(\mathcal{A}) - \frac{1}{q}\end{aligned}$$

   The number of elements in G is supposed large hence $1/q$ is negligible. As the DDH assumption holds, the advantage of $\mathcal{B}$ is negligible. Hence the advantage of $\mathcal{A}$ against CDH is also negligible and the CDH assumption holds.

Exercise 8
Prove that under CDH assumption El-Gamal is OW-CPA.

Solution : *Consider an adversary A that can invert random Elgamal encryptions with probability. We will show that this quantity is negligible.*

We first use A to build an adversary D for computing the Diffie-Hellman function:

D : Adversary to compute the Diffie-Hellman function:

On input (g^a, g^b), we must output g^{ab}.

1. *Give g^a to A as the public key.*
2. *Pick a random $d \in G$ and give (g^b, d) to A as the ciphertext.*
3. *When A outputs $m = \frac{d}{g^{ab}}$, we output $\frac{d}{m}$.*

Note that the distribution $(g^a, (g^b, d))$ does indeed correspond to a random Elgamal public key and encryption of a random message under that key. Thus, with probability ϵ, A outputs the "correct" plaintext m (that is, m such that $d = mg^{ab}$). When this is the case, the output of D matches the Diffie-Hellman function. By our assumption that the CDH assumption holds for the underlying group, ϵ must be negligible, as desired.

OTHER PRETTY Solution using $a = x$, $u = \frac{g}{g^y}$, and $v = g^x$

Exercise 9

Suppose that E_1 and E_2 are symmetric encryption schemes on strings of arbitrary length. Show that the encryption scheme defined by $E'((k_1, k_2), m) = E_2(k_2, E_1(k_1, m))$ (for randomly sampled keys k_1 and k_2) is IND-CPA secure if *either* E_1 or E_2 is IND-CPA secure.

Solution : *Given an adversary $\mathcal{A}$ that breaks the encryption scheme E', we construct adversaries $\mathcal{B}$ and $\mathcal{C}$ that break encryption schemes E_1 and E_2 respectively as follows:*

Adversary $\mathcal{B}$:

- *randomly sample a key k_2 for encryption scheme E_2*
- *run algorithm $\mathcal{A}$. When $\mathcal{A}$ makes a lr-query $(m_{i,0}, m_{i,1})$, $\mathcal{B}$ queries its own lr-oracle with $(m_{i,0}, m_{i,1})$ to obtain a ciphertext c_i, computes $c'_i = E_2(k_2, c_i)$ and returns c'_i to $\mathcal{A}$*
- *when $\mathcal{A}$ outputs a bit b', $\mathcal{B}$ outputs b' as well*

Advesary $\mathcal{C}$:

- *randomly sample a key k_1 for encryption scheme E_1*
- *run algorithm $\mathcal{A}$. When makes a lr-query $(m_{i,0}, m_{i,1})$, $\mathcal{C}$ first computes $(c_{i,0}, c_{i,1}) = (E_1(k_1, m_{i,0}), E_1(k_1, m_{i,1}))$, queries its lr-oracle with $(c_{i,0}, c_{i,1})$ to obtain ciphertext c'_i and returns c'_i to $\mathcal{A}$*
- *when $\mathcal{A}$ outputs a bit b', $\mathcal{C}$ outputs b' as well*

Analysis:

It should be clear that both $\mathcal{B}$ and $\mathcal{C}$ perfectly simulates $\mathcal{A}$'s attack environment, and will correctly guess the bit b exactly when $\mathcal{A}$ does. Therefore, we have $Adv^{IND-CPA}_{E',\mathcal{A}} \leq Adv^{IND-CPA}_{E_1,\mathcal{B}}$ and $Adv^{IND-CPA}_{E',\mathcal{A}} \leq Adv^{IND-CPA}_{E_2,\mathcal{C}}$. Thus, $Adv^{IND-CPA}_{E',\mathcal{A}} \leq min(Adv^{IND-CPA}_{E_1,\mathcal{B}} Adv^{IND-CPA}_{E_2,\mathcal{C}})$, and E' is IND-CPA secure whenever either E_1 or E_2 is secure.

Exercise 10

Prove that if there is an adversary which can break DDH then there is an adversary which can break the IND-CPA security of El-Gamal.

Solution : *Two solutions*

1. *Assume there is an adversary A against DDH, it means it can decide with probability one the following game: given the triple (g^a, g^b, g^c) then $ab = c$ or c is random.*

   We build an adversary B using A for solving the IND-CPA game with El-Gamal. Consider two possible messages (m_0, m_1) the challenge is $y = Elgamal_k(m_b) = (g^r, m_b h^r)$ where r is a random bumber, $h = g^x$ and x the privatrte key.

   B asks the following triple to A: $(g^r, h, y/m_0)$ if the answer of A is yes it means that $y/m_0 = h^r$ so the message m_b was indeed m_0 otherwise it is random it means that m_b was m_1

2. *Recall the definition of IND-CPA security for public-key encryption. We will use the variant where the adversary only gives one challenge. Consider an adversary V that has advantage ϵ in the Elgamal IND-CPA experiment. We will show that this quantity is negligible.*

   We first use V to build an adversary V^* for distinguishing the two distributions from the definition of the DDH assumption:

   V^* : Adversary to distinguish DDH tuples from random tuples:

   On input $(A = g^a, B = g^b, C = g^c)$, we must decide whether $c = ab$ or c is randomly distributed.

   (a) *Give A to V as the public key.*
   (b) *When V outputs a challenge (m_0, m_1), we pick a random β and give $(B, m_\beta C)$ as the response.*
   (c) *If V guesses β correctly, output is $m = \frac{m_\beta g^c}{g^{ab}}$ output 1 if $m = m_\beta$, otherwise output 0.*

   When $c = ab$, the response in step 2 is a correctly distributed ciphertext of m_β, so we perfectly simulate the IND-CPA experiment with V. The probability we output 1 is $1/2 + \epsilon$.

   When c is distributed randomly, the value $(B, m_\beta C)$ is distributed independently of β. Thus the probability we output 1 is exactly 1/2.

   This shows that V^* can distinguish between the two distributions in the definition of the DDH assumption with probability ϵ. By our assumption that the DDH assumption holds for the underlying group, ϵ must be negligible, as desired.

Session 3 – Symmetric cryptography, Modes and Hash functions

Exercise 11

Give the decryption formula for the following encryption modes.

- CBC encryption mode is $C_i = E_K(P_i \oplus C_{i-1})$ and $C_0 = IV$
- CFB encryption mode is $C_i = E_K(C_{i-1}) \oplus P_i$ and $C_0 =$ IV
- OFB encryption mode is

  $C_i = P_i \oplus O_i$; $O_i =$ $E_K(O_{i-1})$ and $O_0 =$ IV.
- CTR encryption mode is $C_0 = IV$ and $C_i = P_i \oplus \mathcal{E}_k(IV + i - 1)$

Solution :

- *CBC*

$$P_i = D_K(C_i) \oplus C_{i-1}, C_0 = IV$$

- *CFB*

$$P_i = E_K(C_{i-1}) \oplus C_i$$

- *OFB*

$$P_i = C_i \oplus O_i$$

- *CTR*

$$P_i = C_i \oplus \mathcal{E}_k(IV + i - 1)$$

Exercise 12
Find an attack on CBC encryption with counter IV, (proving that this encryption mode is not IND-CPA secure). In this scheme the frist IV used is 0 and for generating the next IV we just increase by one the value of the previous IV.

Solution : *Recall CBC*

$$C_i = E_K(P_i \oplus C_{i-1}), C_0 = IV$$

while the mathematical formula for CBC decryption is

$$P_i = D_K(C_i) \oplus C_{i-1}, C_0 = IV$$

Let us fix a block cipher $\mathcal{E} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$. Let $S\mathcal{E} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the corresponding counter-based version of the CBC encryption mode. We show that this scheme is insecure. The reason is that the adversary can predict the counter value.

Notice that:

$$\mathcal{E}_K(LR(m_l, m_r, b)) = \begin{cases} \mathcal{E}_K(m_l) & if\ b = 1 \\ \mathcal{E}_K(m_r) & if\ b = 0 \end{cases}$$

To justify our claim of insecurity, we present an adversary A. As usual it is given an lr-encryption oracle $\mathcal{E}_K(LR(.,.,b))$ and wants to determine b. Our adversary works like this:

Adversary $A\mathcal{E}_K(LR(.,.,b))$
$M_{0,1} \leftarrow 0^n$;
$M_{1,1} \leftarrow 0^n$;
$M_{0,2} \leftarrow 0^n$;
$M_{1,2} \leftarrow 0^{n-1}1$;
$< IV_1, C_1 > \leftarrow^r \mathcal{E}_K(LR(M_{0,1}, M_{1,1}, b))$
$< IV_2, C_2 > \leftarrow^r \mathcal{E}_K(LR(M_{0,2}, M_{1,2}, b))$
If $C_1 = C_2$ *then return* 1 *else return* 0

We claim that:

$$Pr[Exp^{IND-CPA\ 1}_{A\mathcal{E}_K}(A) = 1] = 1$$

and

$$Pr[Exp^{IND-CPA\ 0}_{A\mathcal{E}_K}(A) = 1] = 0$$

First consider the case $b = 0$: $IV_1 = 0$ and $IV_2 = 1$ and $C_1 = \mathcal{E}_K(0)$ and $C_2 = \mathcal{E}_K(1 \oplus 1) = \mathcal{E}_K(O)$ and so $C_1 = C_2$ and the defined experiment returns 1.

On the other hand, if $b = 1$, then $IV_1 = 0$ and $IV_2 = 1$ and $C_1 = \mathcal{E}_K(0)$ and $C_2 = \mathcal{E}_K(1)$, so $C_1 \neq C_2$ the defined experiment returns 0.

Subtracting, we get $Adv^{IND-CPA}_{S\mathcal{E}}(A) = 1 - 0 = 1$, showing that A has a very high advantage. Moreover, A is practical, using very few resources. So the scheme is insecure.

Exercise 13
Prove that CTR is not IND-CCA2 secure.

Solution : *Recall CTR*

$$C_0 = IV$$
$$C_i = P_i \oplus \mathcal{E}_k(IV + i - 1)$$
$$P_i = C_i \oplus \mathcal{E}_k(IV + i - 1)$$

Adversary: A_1:
$m_0 \leftarrow 0^n 0^n$
$m_1 \leftarrow 0^n 1^n$

$b \leftarrow \{0, 1\}$

A_2:

$< IV, C[1], C[2] >= CTR_k(m_b)$
$< IV, D[1], D[2] >= D_k(< IV + 1, C[2], C[2] >)$
if $D[1] = 0^n$ *then* 0 *else* 1

Proof :
First notice that $C[1] = \mathcal{E}_k(IV) \oplus m_b[1]$ *and* $C[2] = \mathcal{E}_k(IV + 1) \oplus m_b[2]$, *we consider the two possible cases for* b, *knowing that* $C_0 = IV$.

$b = 0$ $\quad m_0 = 0^n 0^n$ *then* $C[1] = \mathcal{E}(IV) \oplus 0^n = \mathcal{E}_k(IV)$ *and* $C[2] = \mathcal{E}(IV + 1) \oplus 0^n = \mathcal{E}(IV + 1)$.
$D[2] = C[2] \oplus \mathcal{E}(IV + 2) = \mathcal{E}(IV + 1) \oplus \mathcal{E}(IV + 2)$
$D[1] = C[2] \oplus \mathcal{E}(IV + 1) = \mathcal{E}(IV + 1) \oplus \mathcal{E}(IV + 1) = 0^n$
So $D[1] = 0^n$ *is true.*

$b = 1$ $\quad m_1 = 0^n 1^n$ *then* $C[1] = \mathcal{E}(IV) \oplus 0^n = \mathcal{E}_k(IV)$ *and* $C[2] = \mathcal{E}(IV + 1) \oplus 1^n$.
$D[2] = C[2] \oplus \mathcal{E}(IV + 2) = \mathcal{E}(IV + 1) \oplus 1^n \oplus \mathcal{E}(IV + 2)$
$D[1] = C[2] \oplus \mathcal{E}(IV + 1) = \mathcal{E}(IV + 1) \oplus 1^n \oplus \mathcal{E}(IV + 1) = 1^n$
So $D[1] = 0^n$ *is false.*

Alternative attack A_1: *output* $(0^n, 1^n)$
A_2: *upon receiving* $c = E_{ctr}(m_b)$,
- parse $c = IV \| B$;
- compute $m = B \oplus D_{ctr}(IV \| 0^n)$; *- if* $m = 0^n$, *output 0; else output 1;*

Exercise 14
Prove that CFB is not IND-CCA2 secure.

Solution : *Recall CFB*

$$C_0 = IV$$
$$C_i = P_i \oplus \mathcal{E}_k(C_{i-1})$$
$$P_i = C_i \oplus \mathcal{E}_k(c_{i-1})$$

Adversary: A_1:
$m_0 \leftarrow 0^n$
$m_1 \leftarrow 1^n$

$b \leftarrow \{0,1\}$

A_2:

$< IV, C[1] >= OFB_k(m_b)$
$< IV, D'[1] >= D_k(< IV, C[1] \oplus 1^n >)$
if $D'[1] = 0^n$ *then* 1 *else* 0

Proof :
First notice that $C[1] = \mathcal{E}_k(IV) \oplus m_b$, *we consider the two possible cases for* b, *knowing that* $C_0 = IV$.

$b = 0 \quad m_0 = 0^n$ *then* $C[1] = \mathcal{E}(IV) \oplus 0^n = \mathcal{E}_k(IV)$.
So $D'[1] = C[1] \oplus 1^n \oplus \mathcal{E}_k(c_0) = \mathcal{E}_k(IV) \oplus 1^n \oplus \mathcal{E}_k(IV) = 1^n$ *then* $D'[1] = 0^n$ *is false.*

$b = 1 \quad m_1 = 1^n$ *then* $C[1] = \mathcal{E}_k(IV) \oplus 1^n$.
So $D'[1] = C[1] \oplus 1^n \oplus C_0 = \mathcal{E}_k(IV) \oplus 1^n \oplus 1^n \oplus \mathcal{E}_k(IV) = 0^n$ *then* $D'[1] = m_1$ *is true.*

Exercise 15

Prove that OFB is not IND-CCA2 secure.

Solution : *Recall OFB*

$$O_1 = IV$$
$$O_i = \mathcal{E}_k(O_{i-1})$$
$$C_{i+1} = P_i \oplus O_i$$
$$P_i = C_{i+1} \oplus O_i$$

Adversary: A_1:
$m_0 \leftarrow 0^n$
$m_1 \leftarrow 1^n$

$b \leftarrow \{0,1\}$

A_2:

$< IV, C[1] >= OFB_k(m_b)$
$< IV, D'[1] >= D_k(< IV, C[1] \oplus 1^n >)$
if $D'[1] = m_1$ *then* 0 *else* 1

Proof :
First notice that $C[1] = IV \oplus m_b$, *we consider the two possible cases for* b, *knowing that* $O_1 = IV$ *and* $O_2 = \mathcal{E}(IV)$

$b = 0 \quad m_0 = 0^n$ *then* $C[1] = IV \oplus 0^n = IV$, *hence* $C[1] \oplus 1^n = IV \oplus 1^n$.
So $D'[1] = C[1] \oplus 1^n \oplus O_1 = IV \oplus 1^n \oplus IV = 1^n$ *then* $D'[1] = m_1$ *is true.*

$b = 1 \quad m_1 = 1^n$ *then* $C[1] = IV \oplus 1^n$, *hence* $C[1] \oplus 1^n = IV \oplus 1^n \oplus 1^n = IV$.

So $D'[1] = C[1] \oplus 1^n \oplus O_1 = IV \oplus IV = 0^n$ *then* $D'[1] = m_1$ *is false.*

Exercise 16

Prove that CBC with random IV is not IND-CCA2 secure. This time IV is a random number. But notice that this mode is IND-CPA secure.

Solution : *Recall CBC*

$$C_i = E_K(P_i \oplus C_{i-1}), C_0 = IV$$

while the mathematical formula for CBC decryption is

$$P_i = D_K(C_i) \oplus C_{i-1}, C_0 = IV$$

Adversary $A\mathcal{E}_K(LR(.,.,b)), \mathcal{D}_K(.)$
$M_0 \leftarrow 0^n$;
$M_1 \leftarrow 1^n$;
$< IV, C[1] > \leftarrow \mathcal{E}_K(LR(M_0, M_1, b))$;
$IV' \leftarrow IV \oplus 1^n$
$M \leftarrow \mathcal{D}_K(IV', C[1])$
If $M = M_0$ *then return* 1 *else return* 0

The adversary's single lr-encryption oracle query is the pair of distinct messages M_0, M_1, *each one block long. It is returned a ciphertext* $< IV, C[1] >$. *It flips the bits of the* IV *to get a new* IV, IV', *and then feeds the ciphertext* $< IV', C[1] >$ *to the decryption oracle. It bets on world* 1 *if it gets back* M_0, *and otherwise on world* 0. *It is important that* $< IV', C[1] > \neq < IV, C[1] >$ *so the decryption oracle query is legitimate. Now, we claim that:*

$$[PrExp_{\mathcal{SE}}^{IND-CCA\ 1}(A) = 1] = O$$
$$[PrExp_{\mathcal{SE}}^{IND-CCA\ 0}(A) = 1] = 1$$

Hence $Adv_{\mathcal{SE}}^{IND-CCA}(A) = 1 - 0 = 1$. *And* A *achieved this advantage by making just one lr-encryption oracle query, whose length, which as per our conventions is just the length of* M_0, *is* n *bits, and just one decryption oracle query, whose length is* 2^n *bits.*

- **In world** 1**, meaning** $b = 1$, *the lr-encryption oracle returns* $< IV, C[1] >$ *with*

$$C[1] = \mathcal{E}_K(IV \oplus M_1) = \mathcal{E}_K(IV \oplus 1^n)$$

Now notice that

$$\begin{aligned} M &= \mathcal{D}_K(IV', C[1]) \\ &= \mathcal{E}_K^{-1}(C[1]) \oplus IV' \\ &= \mathcal{E}_K^{-1}(\mathcal{E}_K(IV \oplus 1^n))IV' \\ &= (IV \oplus 1^n) \oplus IV' \\ &= (IV \oplus 1^n) \oplus (IV \oplus 1^n) \\ &= 0^n \\ &= M_0 \end{aligned}$$

Thus, the decryption oracle will return M_0, *and* A *will return* 1.

- **In world 0, meaning** $b = 0$, *the lr-encryption oracle returns* $< IV, C[1] >$ *with*

$$C[1] = \mathcal{E}_K(IV \oplus M_0) = \mathcal{E}_K(IV \oplus 0^n)$$

Now notice that

$$\begin{aligned} M &= \mathcal{D}_K(IV', C[1]) \\ &= \mathcal{E}_K^{-1}(C[1]) \oplus IV' \\ &= \mathcal{E}_K^{-1}(\mathcal{E}_K(IV \oplus 0^n)) \oplus IV' \\ &= (IV \oplus 0^n) \oplus IV \\ &= (IV \oplus 0^n) \oplus (IV \oplus 1^n) \\ &= 1^n \\ &= M_1 \end{aligned}$$

Thus, the decryption oracle will return M_1*, and* A *will return* 0*, meaning will return* 1 *with probability zero.*

SOLUTION 2
Adversary $A\mathcal{E}_K(LR(.,.,b)), \mathcal{D}_K(.)$
$M_0 \leftarrow 0^n 0^n$;
$M_1 \leftarrow 0^n 1^n$;
$< IV, C[1]C[2] > \leftarrow \mathcal{E}_K(LR(M_0, M_1, b))$;
$M'[1]M'[2] \leftarrow \mathcal{D}_K(IV, 0^n C[2])$

If $M'[2] \oplus C[1] = 0$ *then return* 0 *else return* 1

- **In world 0, meaning** $b = 0$, *the lr-encryption oracle returns* $< IV, C[1] >$ *with*

$$C[1] = \mathcal{E}_K(IV \oplus M_0) = \mathcal{E}_K(IV \oplus 0^n) = \mathcal{E}_K(IV)$$

$$C[2] = \mathcal{E}_K(\mathcal{E}_K(IV))$$

Now notice that

$$\begin{aligned} M'[2] \oplus C[1] &= \mathcal{D}_K(IV, C[2]) \oplus C[1] \\ &= \mathcal{E}_K(IV) \oplus \mathcal{E}_K(IV) \\ &= 0^n \end{aligned}$$

- **In world 1, meaning** $b = 1$, *the lr-encryption oracle returns* $< IV, C[1] >$ *with*

$$C[1] = \mathcal{E}_K(IV \oplus M_0) = \mathcal{E}_K(IV \oplus 0^n) = \mathcal{E}_K(IV)$$

$$C[2] = \mathcal{E}_K(1^n \oplus \mathcal{E}_K(IV)) = \mathcal{E}_K(\overline{\mathcal{E}_K(IV)})$$

Now notice that

$$\begin{aligned} M'[2] \oplus C[1] &= \mathcal{D}_K(IV, C[2]) \oplus C[1] \\ &= \overline{\mathcal{E}_K(IV)} \oplus \mathcal{E}_K(IV) \\ &= 1^n \end{aligned}$$

Exercise 17

Let $\mathcal{E}$ be a (secret key) block cipher, and CBC-MAC be the message authentication code defined as follows:

$$\begin{array}{l} \text{CBC-MAC}(k, m_1 | \ldots | m_n) \\ \quad c_1 = \mathcal{E}_k(m_1); \\ \quad \text{for } i = 2 \text{ to } n, \text{ do:} \\ \quad\quad c_i = \mathcal{E}_k(c_{i-1} \oplus m_i); \\ \quad \text{Output } c_n; \end{array}$$

Show that CBC-MAC is not a secure message authentication code by finding a collision in the MAC. The attacking adversary can query an oracle that will compute the MAC of any message, but cannot compute the block cipher $\mathcal{E}_k$ on his own.

Solution : *Note that CBC-MAC*$(k, 0^{128})$ *and CBC-MAC*$(k, 0^{128}|$*CBC-Mac*$(0^{128}))$ *will both have the same MAC, so an adversary needs only query its oracle on the first to obtain mac, and output* $(0^{128}|mac, mac)$ *as its forgery.*