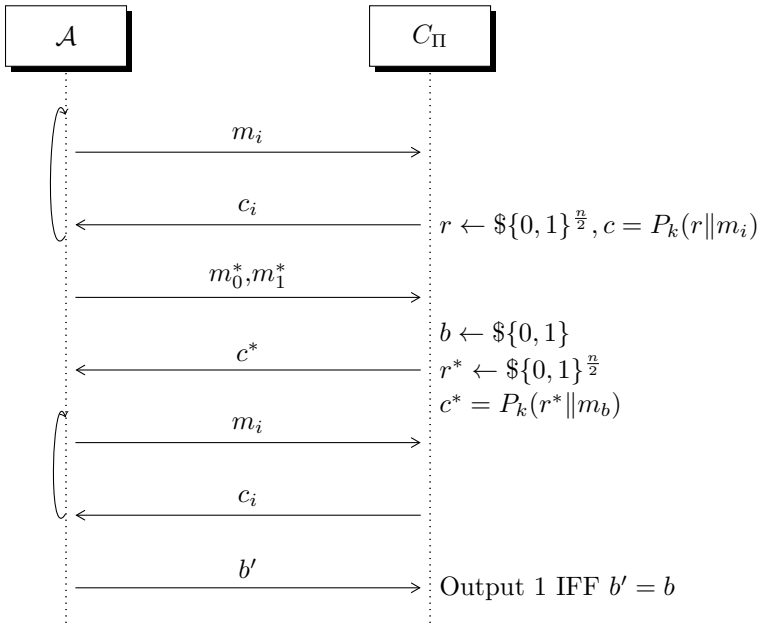
Contents

Exercise 1

4.1 point a

Suppose we have the given scheme Π and the CPA game



If P is a PRP family, Π is always CPA secure unless a *bad event* happens.

Suppose m_0^* and m_1^* have already been sent multiple times to $\mathcal{C}_\Pi$ and that $\mathcal{A}$ collected , at most, *poly* couples containing (m_0^* or m_1^*, c'). It could happen that, sent m_0^* and m_1^* as challenge messages, $\mathcal{A}$ receives one of the previuosly received c'. In that case, $\mathcal{A}$ knows which message has been encrypted, then he can easily win the game.

What is the probability $\mathcal{A}$ can win in this way?
When $c^* = P_k(r^*\|m_b)$ where r^* was already chosen by $\mathcal{C}$ in a previous request of m_b (where b can be 0 or 1), $\mathcal{A}$ can win. Let's call r_b a random number chosen when m_b was sent to $\mathcal{C}$:

$$Pr[\mathcal{A} \text{ wins }] = Pr[r^* = r_b \wedge m_b \text{ is chosen for encryption }] =$$
$$= \mathcal{P}[m_b]\mathcal{P}[r^* = r_b] =$$
$$= \frac{1}{2}\frac{1}{2^{\frac{n}{2}}} m_b \text{ asked just once}$$
$$= \frac{1}{2}\frac{q}{2^{\frac{n}{2}}} m_b \text{ asked q times}$$

Since $b \leftarrow \$\{0,1\}$ and $\mathcal{P}[r^* = r_0 \wedge m_0$ is chosen for encryption $]$ and $\mathcal{P}[r^* = r_1 \wedge m_1$ is chosen for encryption $]$ are disjoint probabilities, the probabilities of the two events is the sum of single probabilities.

So, $\mathcal{A}$ wins with probability at least $\frac{1}{2^{\frac{n}{2}}}$, at most $\frac{q}{2^{\frac{n}{2}}}$, which is still negligible since q can be , at most, *poly*.

4.2 point b

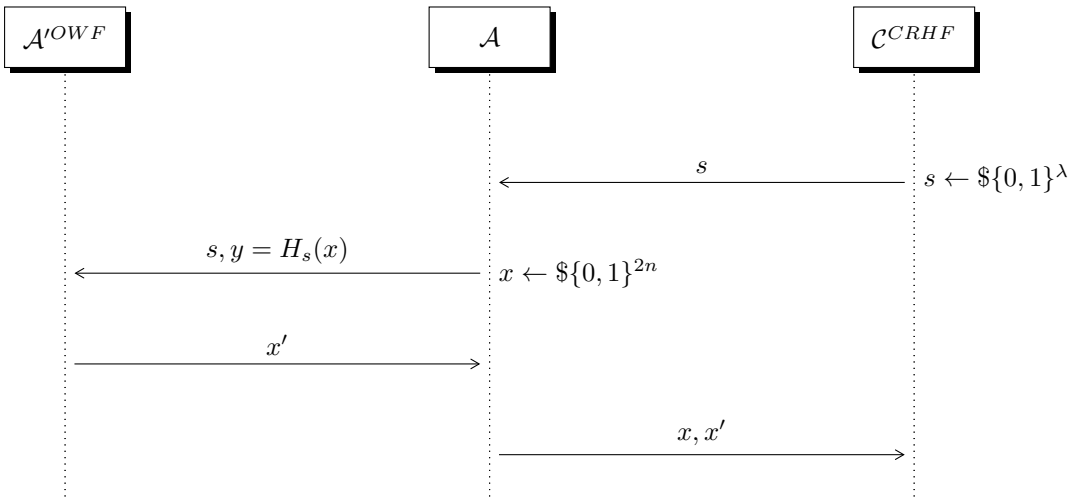
Exercise 2

4.3 point a

4.3.1 i

$$\mathcal{H} \text{ is CRHF } \Rightarrow \mathcal{H} \text{ is OWF}$$

To show this property, let's make a reduction:



When does not $\mathcal{A}$ win?
Since CRHF game wants the final couple (x, x') with $x \neq x'$, if $\mathcal{A}'^{OWF}$ returns $x' = x$ the CRHF game doesn't work.

This **BAD** event happens with

$$\mathcal{P}[x = x'] = Col(X, X') = \sum_x \mathcal{P}[X = x \wedge X' = x] = \sum_x \mathcal{P}[X = x]\mathcal{P}[X' = x] = \frac{1}{2^{2n}}$$

.

4.3.2 ii

If functions from $\mathcal{H}$ family aren't compressing, the probability of **BAD** event changes:

$$\mathcal{P}[x = x'] = Col(X, X') = \sum_x \mathcal{P}[X = x \wedge X' = x] = \sum_x \mathcal{P}[X = x]\mathcal{P}[X' = x] = \frac{1}{2^n}$$

.

Now, if our functions from $\mathcal{H}$ were compressing (from 2n bits to n bits), the best CRHF function (the function with the minimum number of collisions) had $2^n + 1$ inputs generating a collision (in the same codomain's element).

In this case, the best possible CRHF function is bijective (since it could be a permutation over 2^n elements).

In general, for non-compressing functions we can show that

$$\mathcal{H} \text{ is CRHF mapping n bits to n bits} \Rightarrow \mathcal{H} \text{ is OWF}$$

with the same reduction of the above **point i** .

4.4 point b

Given $H^*_{s_1,s_2}(x) = H'_{s_2}(H_{s_1}(x))$ from $H^* : 4n \to n$.

We can calculate the probability of collision ($H'(H(x)) = H'(H(x'))$) in the above defined function:

$$Pr[H(x) = H(x')|x \neq x'] + Pr[H^*(x) = H^*(H(x'))|H(x) \neq H(x')] =$$

Since the the events are independent, i can write

$$Pr[x \neq x']Pr[H(x) \neq H(x')]Pr[H'(H(x)) = H'(H(x'))]+$$
$$+Pr[x \neq x']Pr[H(x) = H(x')]Pr[H'(H(x)) = H'(H(x'))]$$

Now:

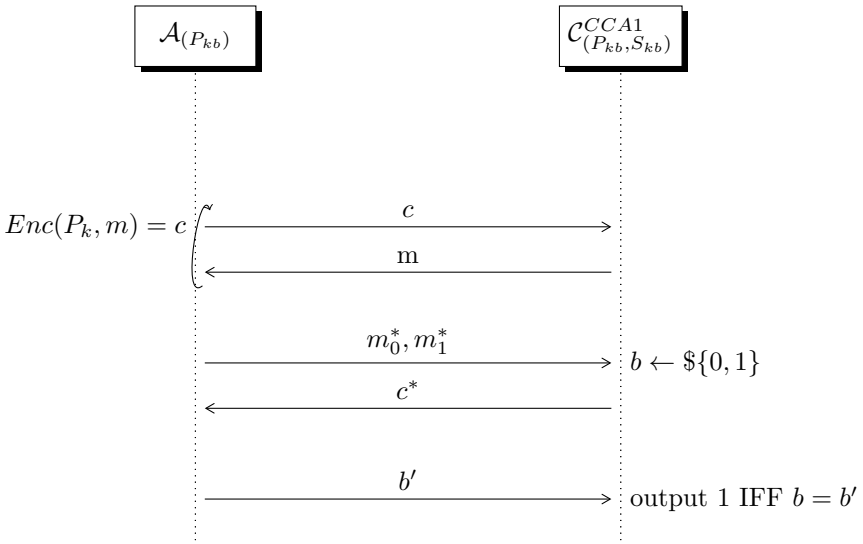
$$Pr[x \neq x'] = \frac{1}{2^{4n}} + \frac{1}{2^{4n-1}}$$

$$Pr[]$$

Exercise 4

4.4.1 point a

Formal definition of CCA1. Consider the following $GAME^{CCA}$



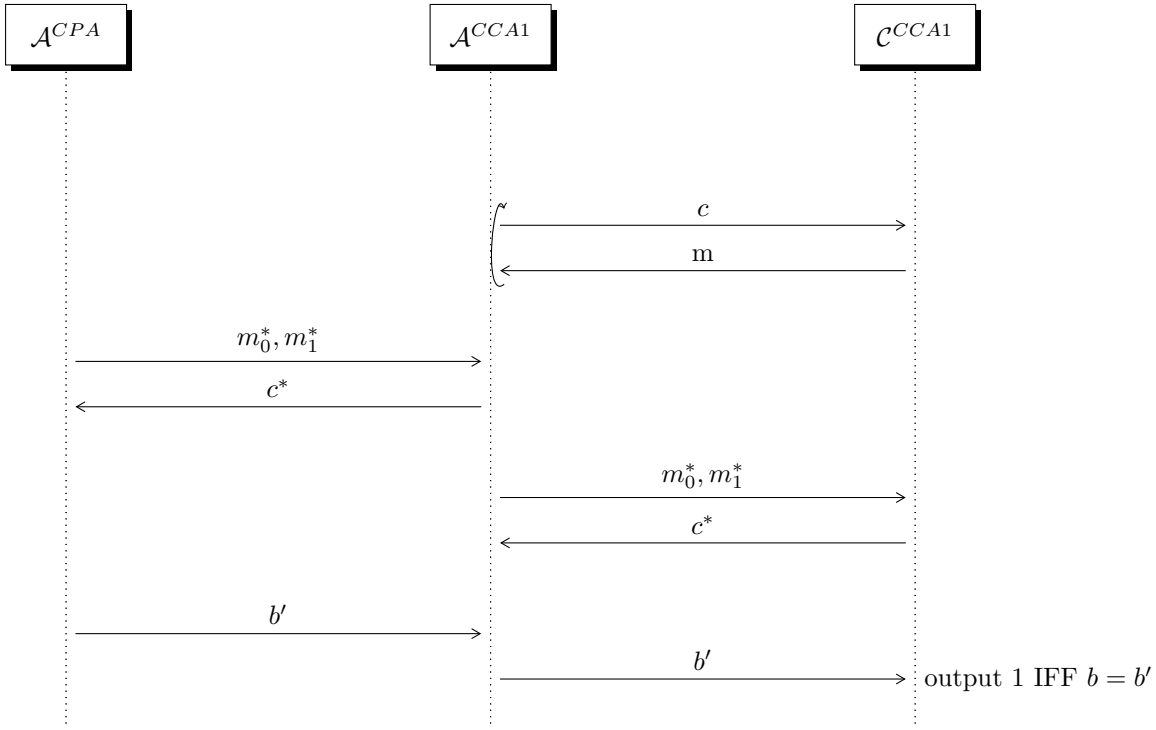
$$Pr[A(\lambda, 0) = 1] - Pr[A(\lambda, 1) = 1]$$

T0D0 1: Put an explanation and finalize probability

4.4.2 point b

CCA1 $\implies$ CPA

Assume $\exists \mathcal{A}^{CPA}$ which is able to break CPA. $\mathcal{A}^{CCA1}$ will use this $\mathcal{A}^{CPA}$ to break $CCA1$



Intuitively this works because we used CPA to define CCA security. Therefore if an attacker is able to break CPA he is also "automatically" able to break CCA (the challenge part is the same for both games).

$PKE^{CPA} \implies CCA1$

Now consider the following Game which is still CPA secure but on the other hand it leaks the key whenever C receives a decryption query.

The scheme is defined as follows

4.4.3 point c

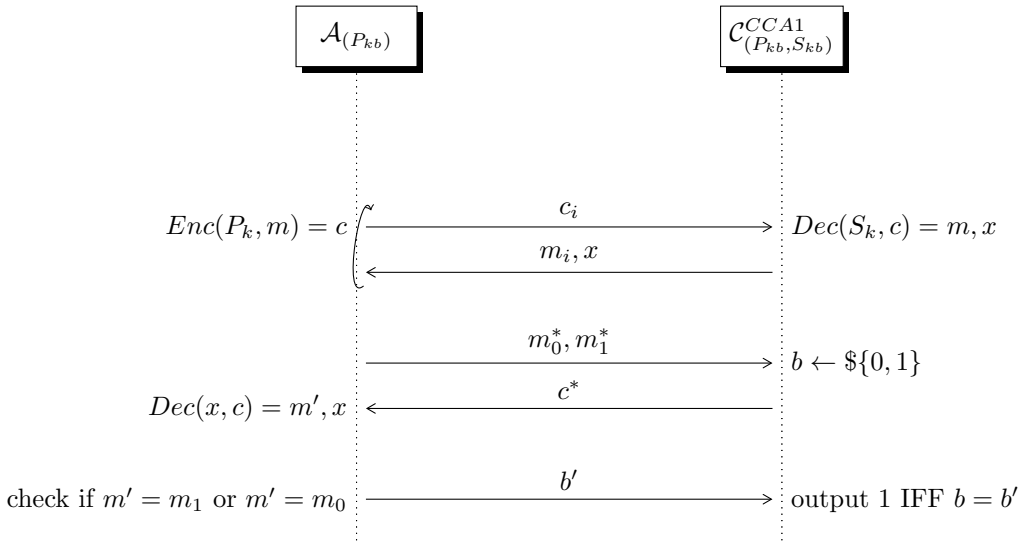
point i

My goal is to demostrate if PI is CCA1 -¿ PI' is also CCA1, in order to this observe the following reduction scheme:

A'(PI') A C

A'sends a message m composed by 't' elements. A takes every single elements and sends to C to get the cyphertexts of each. Then recombines the cyphertext and sends back the single cyphertext to A'.

At the start of the challenge, A' sends to messages: m0, m1 of 't' bytes to A. A sends to C the t-1 bytes of m0 and receives thr corrispondent t-1 cyphertext



then A sends to C two bytes: m0t and 1, and receive the cyphertext c*' of one of these. At this point A recombines all of the t-1 cyphertext + the last received, c*' and sends back to A'. Now if A' distinguishes that c* is the cyphertext of m0 sends b' in the other case C has enrypted the fixed byte and send BOT.

At the end if A receives b', sends b', if receives BOT sends 0.

point ii

Π CCA2 $\implies$ Π′¬CCA2

Consider the following PKE Scheme:

- $Enc(P_k, m[t]) = Enc(P_k, m_1)||...||Enc(P_k, m_t)$
- $Dec(S_k, c[t]) = Dec(S_k, c_1)||...||Dec(S_k, c_t)$

Since in CCA2 I can make decryption queries after the challenge, I can create a $c' \neq c^*$ just by inverting the first two bits of c^* ($c* = c_1^*||c_2^*||...||c_t^*$ now $c' = c_2^*||c_1^*||...||c_t^*$). Now when I receive the decrypted message I can simply switch the first two bits again and discover which of the two challenge messages was encrypted.

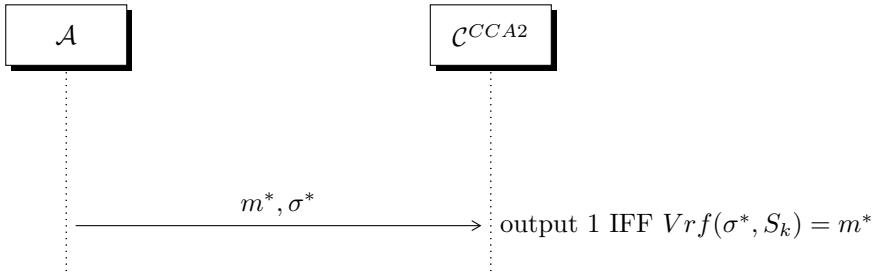
4.4.4 point d

From the definition of padded-RSA I can construct the following attack:

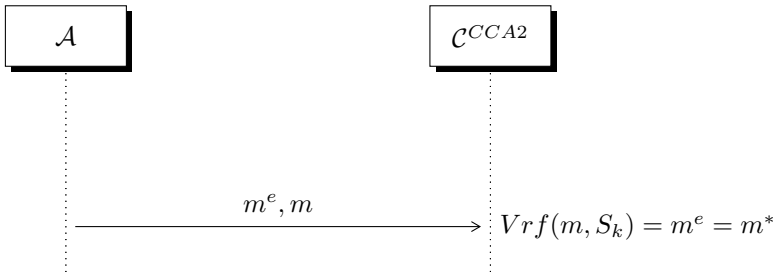
Suppose we do the challenge query, when I receive C^* I will have something in this form: $c^* = (r||m_b)^e mod N$. Now, since RSA is malleable, I can change C^* in order to be able to ask a valid decryption query, therefore $C' = C^* \times (r')^e mod N = ((r||m_b) \times r')^e \neq C^*$. Now when I ask for the decryption of c' I will get $m' = ((r||m)r')^{ed} = (r||m)r'$ since I know r' I can simply divide $\frac{m'}{r'}$ and take the last l bits. This was the encrypted message in the challenge.
chapter*Exercise 5

4.4.5 point a

We want to break the following game:

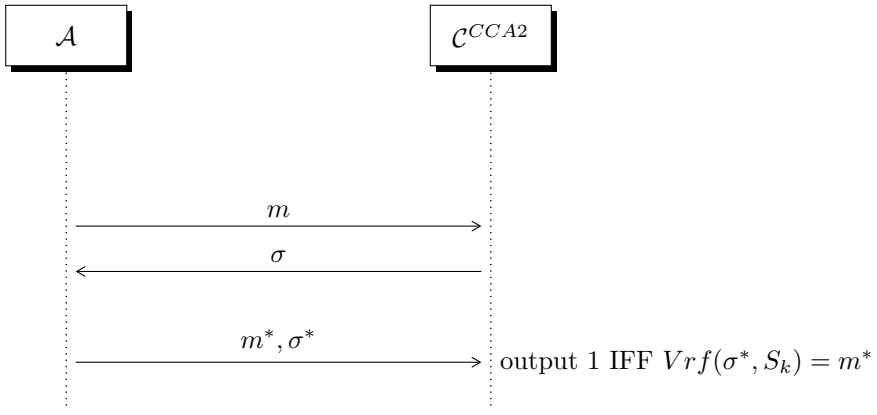


We win in this way:



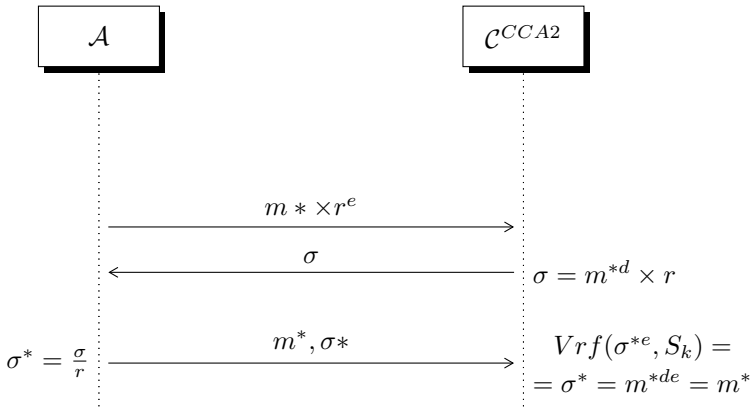
4.4.6 point b

We want to break the following game when m^* is fixed:

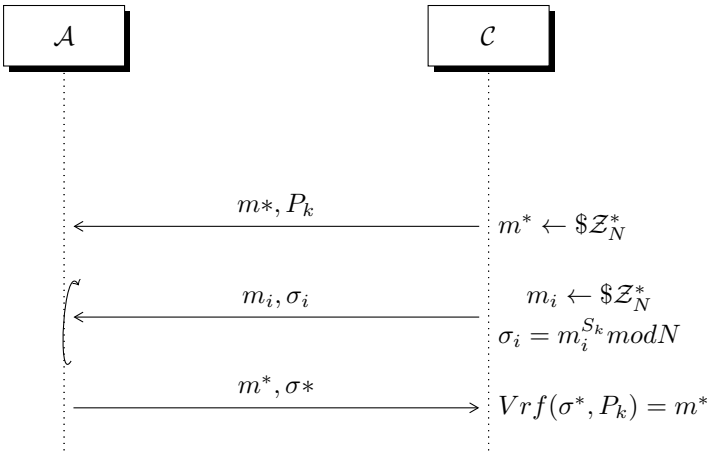


We can win in the following way:

- Select $r \in \mathcal{Z}_n^*$
- Compute r^{-1}


$\mathcal{A}$
$\mathcal{C}^{CCA2}$
$m* \times r^e$
σ
$\sigma = m^{*d} \times r$
$\sigma^* = \frac{\sigma}{r}$
$m^*, \sigma*$
$Vrf(\sigma^{*e}, S_k) =$
$= \sigma^* = m^{*de} = m^*$

4.4.7 point c


$\mathcal{A}$
$\mathcal{C}$
$m*, P_k$
$m^* \leftarrow \$\mathcal{Z}_N^*$
m_i, σ_i
$m_i \leftarrow \$\mathcal{Z}_N^*$
$\sigma_i = m_i^{S_k} mod N$
$m^*, \sigma*$
$Vrf(\sigma^*, P_k) = m^*$