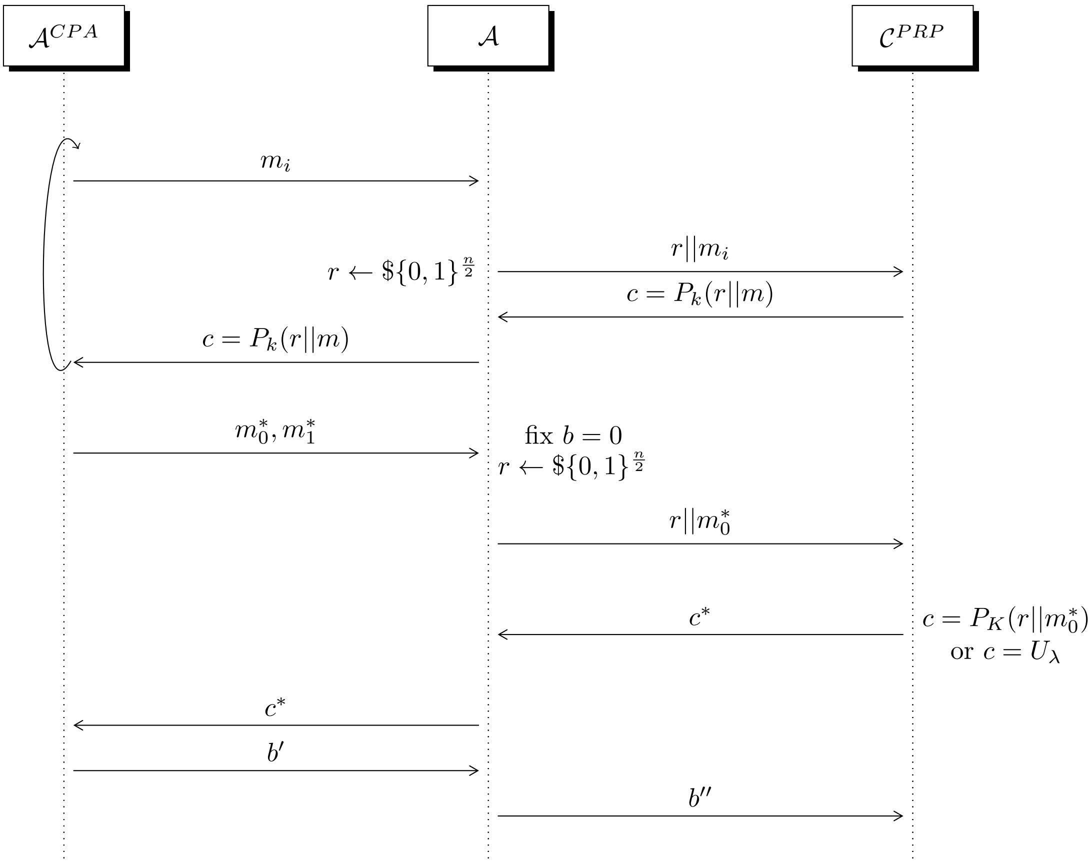
Contents

Exercise 1

1.1 point a

Since PRP in by definition computationally close to U breaking CPA security implies to distinguish between the PRP and a uniform distribution.



When A receives b' he will check if $b' = b$ then C^{PRP} will have chosen a PRP otherwise c^* will come from U. We must consider the case in which A^{CPA} will output $b' = b$ even when c^* comes from Uniform.

So the final probability will be:

$$\underbrace{P[A^{CPA} = 0|PRP]}_{\text{right guess}} - \overbrace{P[A^{CPA} = 0|U]}^{\text{wrong guess}} = \frac{1}{2} + \varepsilon - \frac{1}{2}$$

1.2 point b

It is not CCA secure because we can construct a contrived PRP family in the following way:

$\forall k, P_k^{-1}(0) = 0^{n/2}||k$ this is still CPA since A cannot do decryption queries however an adversary playing a CCA game can query the decryption of 0 and thus obtain the key (assuming $\lambda \leq$).

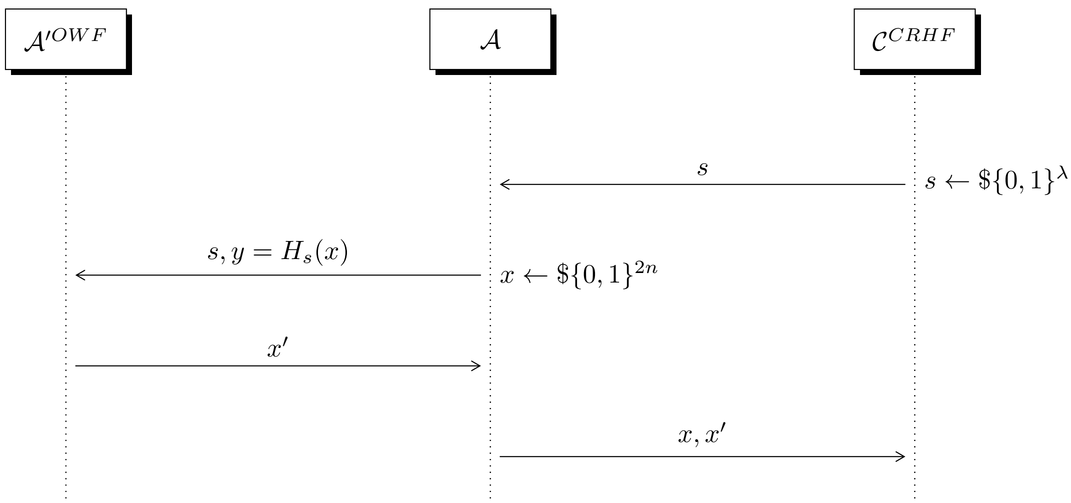
Exercise 2

2.1 point a

2.1.1 point 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

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

.

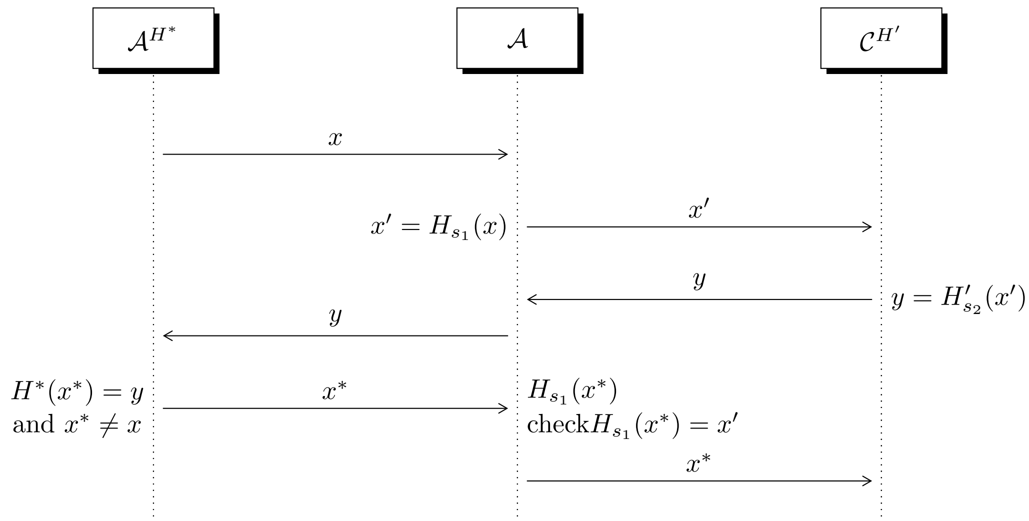
2.1.2 ii

Now if the domain and codomain are both n we want to demonstrate that $CRHF \implies OWF$. Suppose there exists an adversary which is able to break OWF with non negligible prob. Because in the best case (bijective) you will find only one x for any y you will never be able to find a collision. So $CRHF \not\Rightarrow OWF$.

2.2 point b

Given $H^*_{s_1,s_2}(x) = H'_{s_2}(H_{s_1}(x))$ with $H^* : 4n \to n$. Suppose $\exists A^{H^*}$ which is able to find a collision in H^*.

Consider the following two Games:



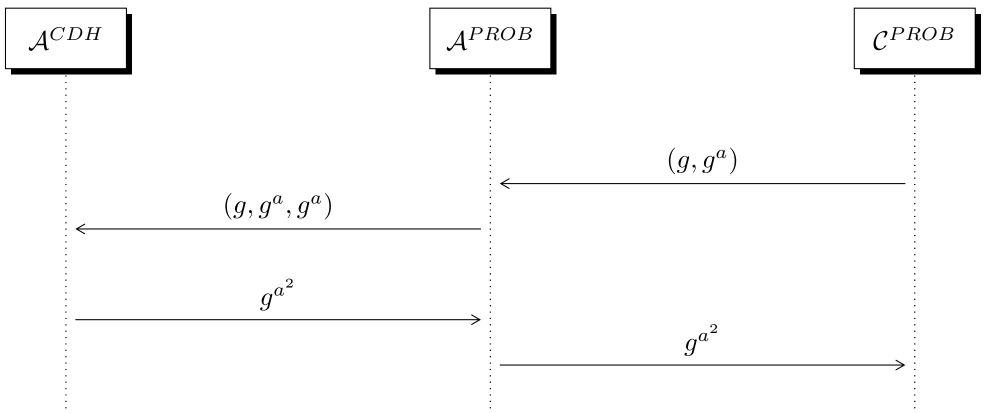
There is a "BAD event" in which A^{H^*} outputs a collision for H_{s_1}, meaning that $H_{s_1}(x) = H_{s_1}(x')$ in this case the second part of the reduction doesn't work. But $Pr[BAD]$ is negligible since H_{s_1} is collision resistant by definition. But now $H^*(x') = H^*(x)$ since x' was a collision for H^* but this must be a collision also for H'_{s_2} which was a CRHF for hypothesis.

Exercise 3

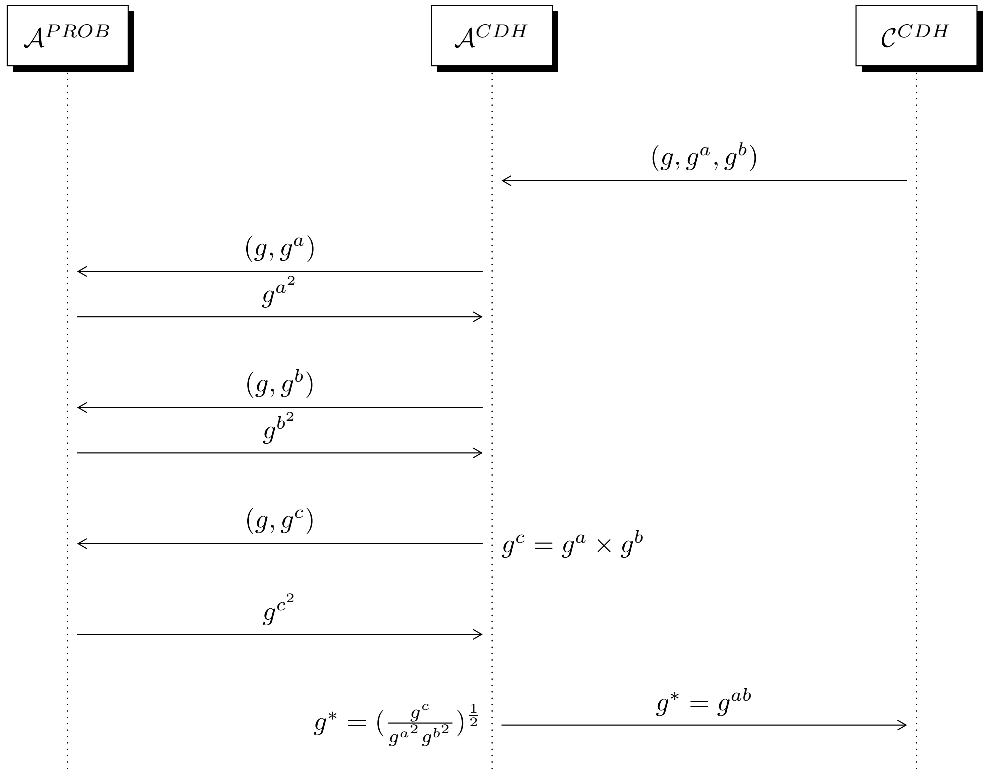
3.1 point a

Call (g, g^a) compute g^{a^2} PROBLEM.
Now we want to prove that PROBLEM $\Leftrightarrow$ CDH

PROBLEM $\implies$ CDH:
Suppose $\exists A^{CDH}$ which is able to break CDH with non negligible probability.



CDH $\implies$ PROBLEM



3.2 point b

$$\begin{cases} N = pq \\ \varphi(N) = (p-1)(q-1) = pq - p - q + 1 \end{cases} \Rightarrow \begin{cases} N = pq \\ p + q = N - \varphi(N) + 1 = \beta \end{cases}$$

$$\Rightarrow \begin{cases} N = \beta q - q^2 \Rightarrow -q^2 + \beta q - N = 0 \\ \beta = N - \varphi(N) + 1 \end{cases}$$

$$\Rightarrow q^2 - \beta q + N = 0 \Rightarrow p, q = \frac{\beta \pm \sqrt{\beta^2 - 4N}}{2}$$

Now with $N = 18830129$ and $\varphi(N) = 18819060$

$$\beta = 18830129 - 18819060 + 1 = 11070$$
$$\Delta = 11070^2 - 4 * 18830129 = 122544900 - 75320516 = 47224384$$
$$q_1 = \frac{11070 + 6872}{2} = 8971$$
$$q_2 = \frac{11070 - 6872}{2} = 2099$$

3.3 point c

The Bézout's identity states that it is always possible to find x,y s.t. $ax + by = d$ if GCD(a,b)=d.

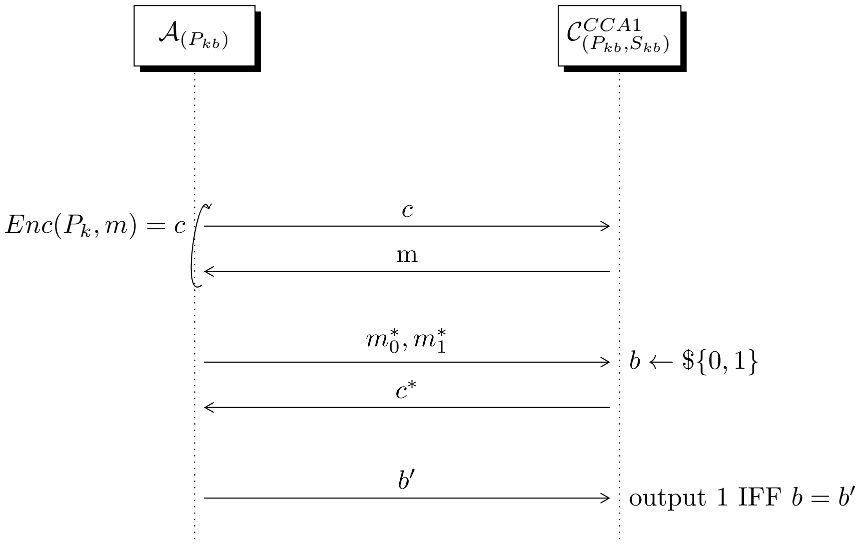
Now since e_A, e_b are coprime, GCD(e_A, e_b)=1 and then we can (using Euclidean algorithm) polinomially retrieve x,y s.t. $e_A x + e_b y = 1$.

Since Eve has e_a, e_b, c_a, c_b and also x, y he can calculate $(m^{e_a})^x (m^{e_b})^y = m^{e_A x + e_b y} = m$

Exercise 4

4.1 point a

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

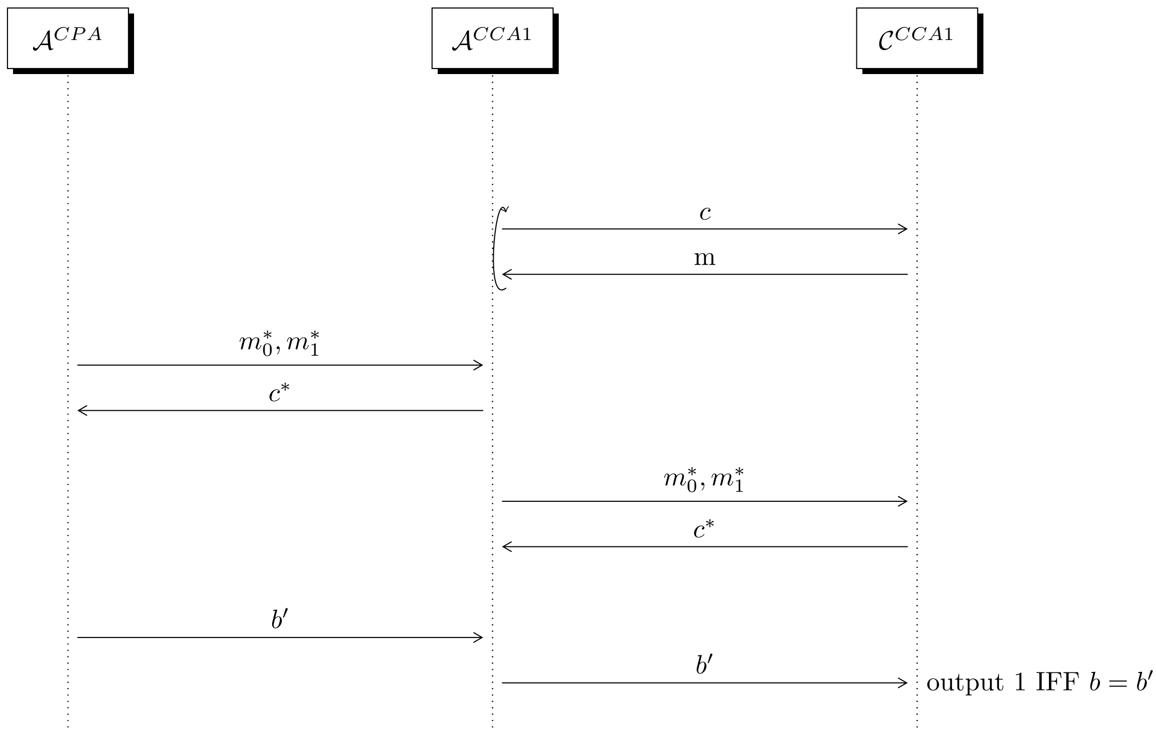


$$|Pr[A(\lambda, 0) = 1] - Pr[A(\lambda, 1) = 1]| \leq negl(\lambda)$$

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 $CCA1$

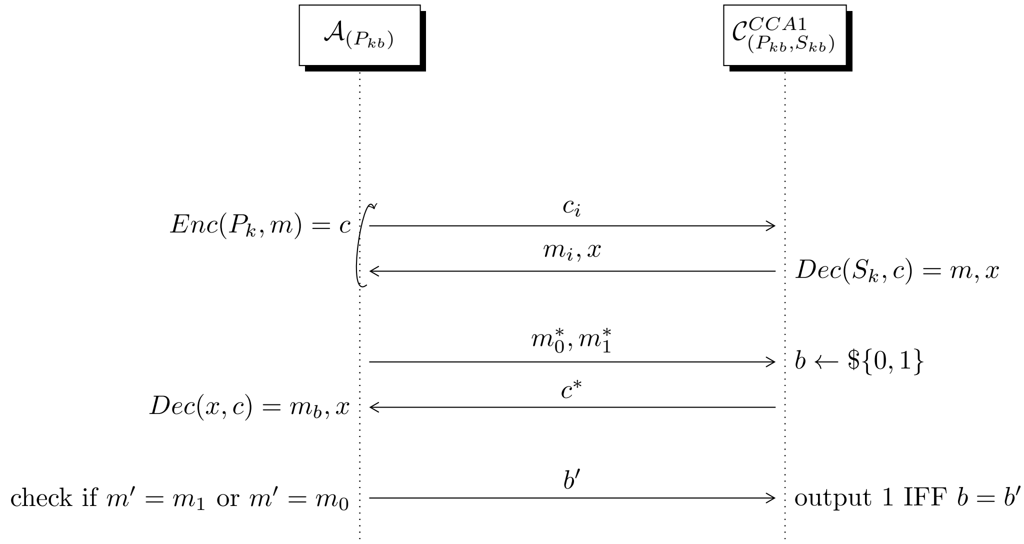


Intuitively this works because we used CPA to define CCA security. Therefore if an attacker is able to break CPA he is also "automatically" able to break CCA (the challenge part for CPA is the same for CCA).

$PKE^{CPA} \implies CCA1$

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



4.3 point c

4.3.1 point i

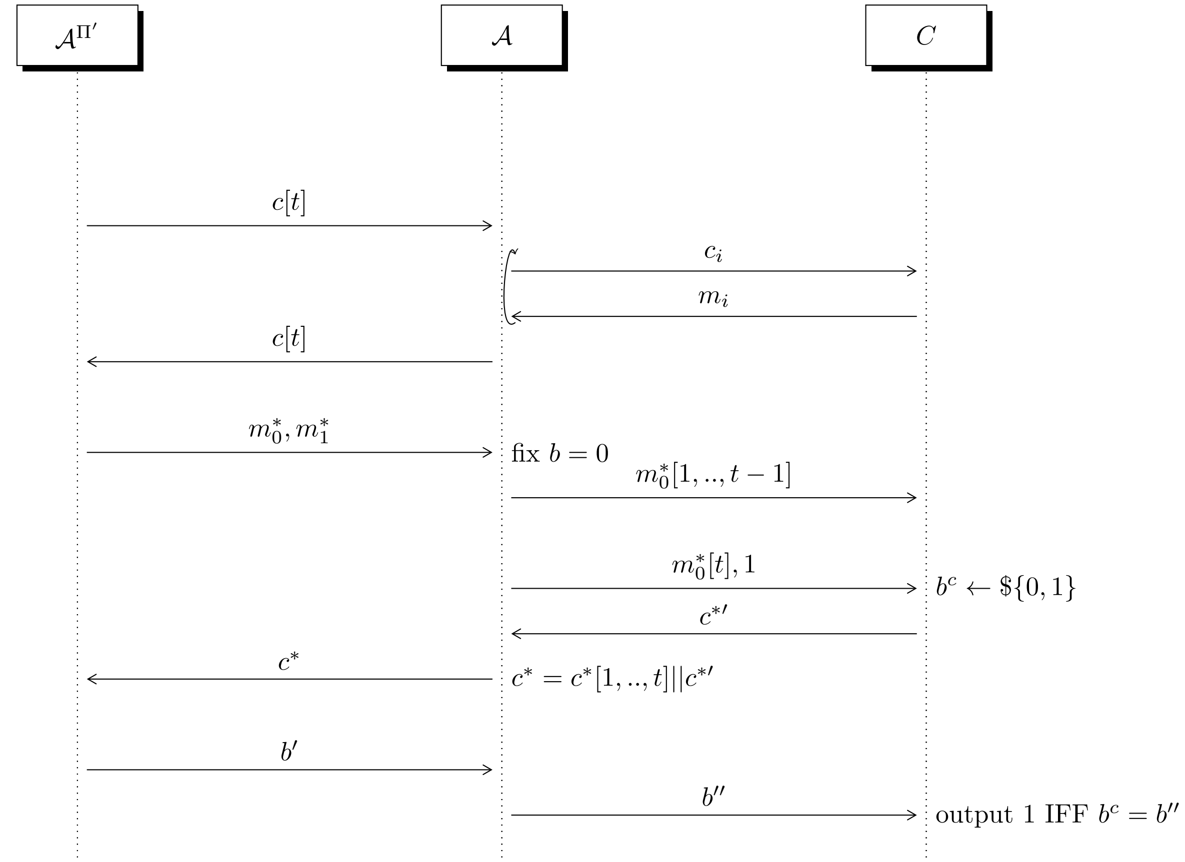
My goal is to demonstrate if Π is $CCA1 \Rightarrow \Pi'$ is also CCA1, in order to do this observe the following reduction scheme:

Suppose $\exists A^{\Pi'}$ which is able to break the CCA1 security of Π'

$A^{\Pi'}$ sends ciphertext composed by t elements. A takes every single elements and sends to C to get the plaintext of each one. Then recombines the plaintext and sends back the single plaintext to $A^{\Pi'}$.

At the start of the challenge, $A^{\Pi'}$ sends two messages: m_0, m_1 of t bits to A. A sends to C the first t-1 bytes of m_0 and receives the corresponding $t-1$ ciphertexts then A sends to C the challenge as: $m_0[t]$ and 1, receiving the ciphertext $c{*}'$ of one of the two. At this point A recombines all of the t-1 ciphertext + the last received, $c^{*\prime}$ and sends back to A^{Π}. Now A will just forward the response.

The probability will be $|P[A^{\Pi} = 0|b^c = 0] - P[A^{\Pi} = 0|b^c = 1]| = \frac{1}{2} + negl(\lambda) - \frac{1}{2} > negl(\lambda)$



4.3.2 point ii

Π CCA2 $\implies$ $\Pi'\neg$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 point d

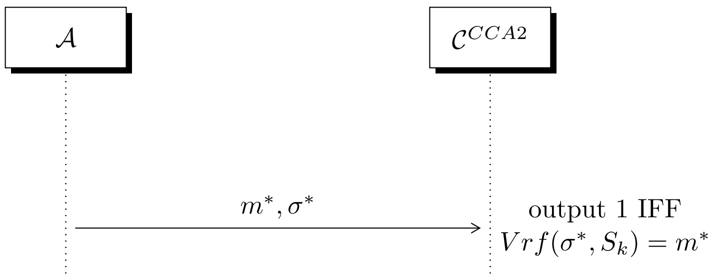
From the definition of padded-RSA I can construct the following attack:

Suppose we do the challenge query, when I receive C^* I will have something in this form: $c^* = (r||m_b)^e modN$. Now, since RSA is malleable, I can change C^* in order to be able to ask a valid decryption query, therefore $C' = C^* \times (r')^e modN = ((r||m_b) \times r')^e \neq C^*$. 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.

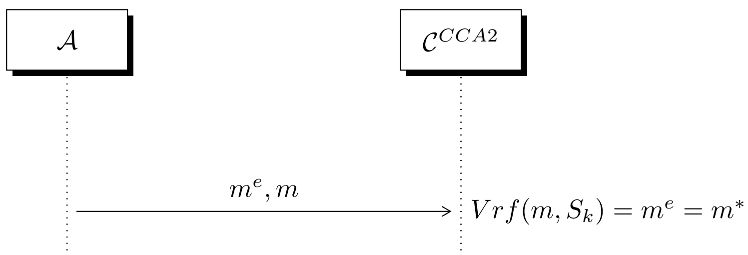
Exercise 5

5.0.1 point a

We want to break the following game:



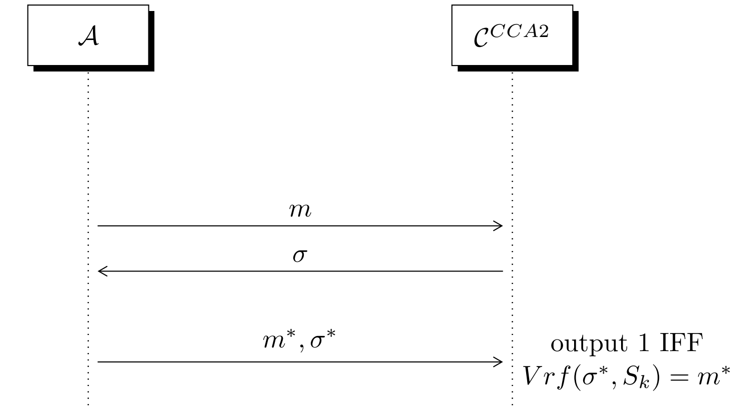
We win in this way:

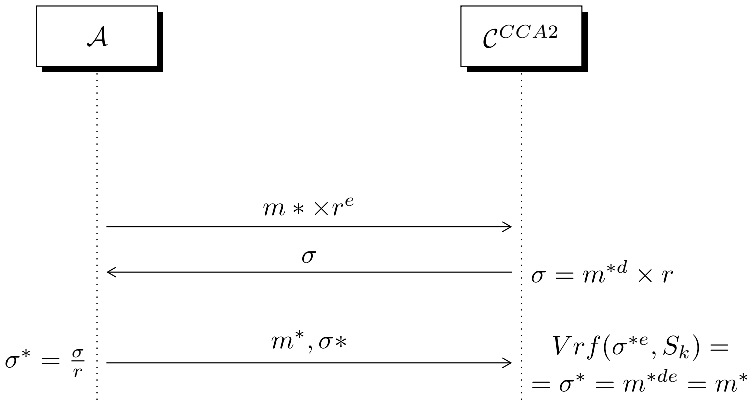


5.0.2 point b

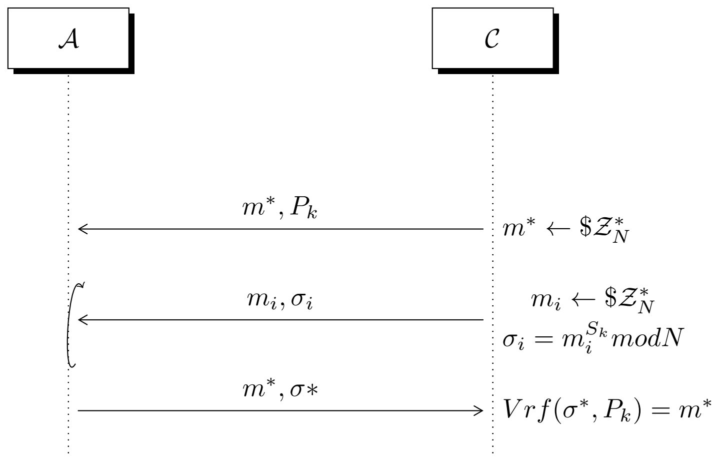
We want to break the following game when m^* is fixed:
We can win in the following way:

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

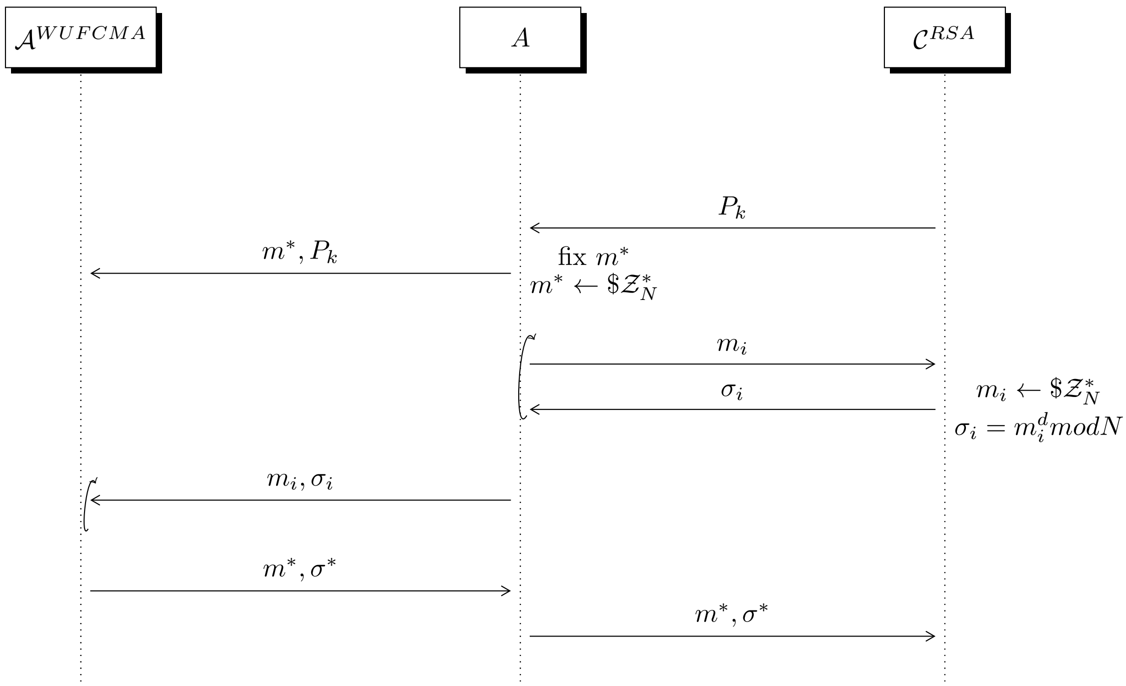

$\mathcal{A}$
$\mathcal{C}^{CCA2}$
m
σ
m^*, σ^*
output 1 IFF
$Vrf(\sigma^*, S_k) = m^*$


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

5.0.3 point c


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

Consider the following reduction:

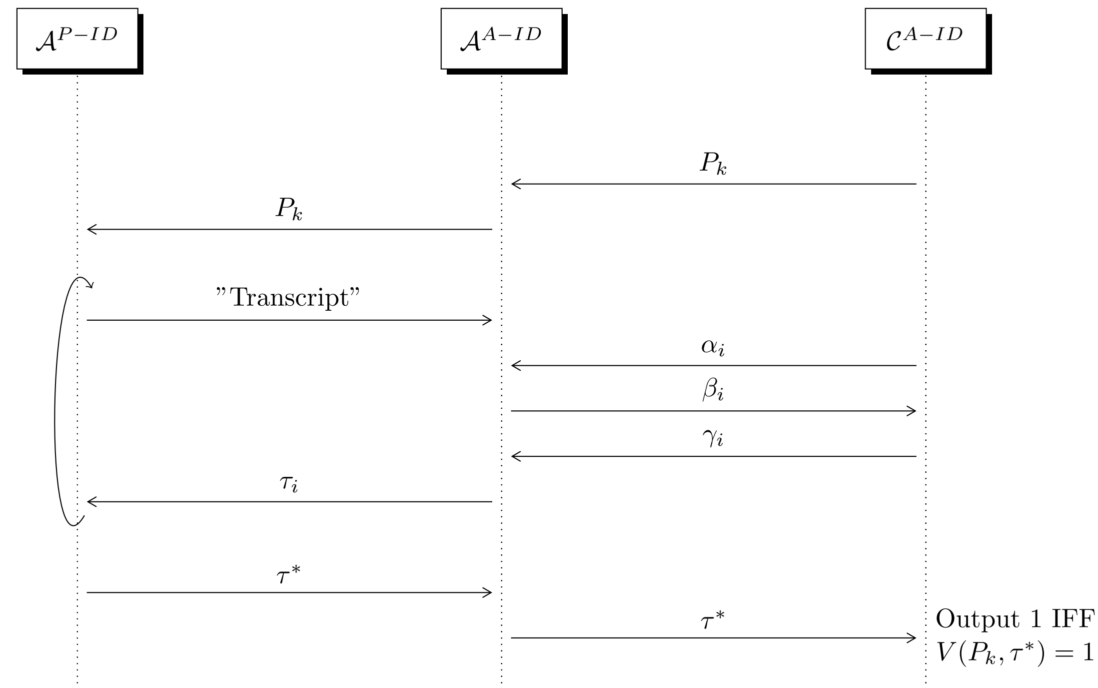


So forging a σ^* is equivalent to forging on RSA.

Exercise 6

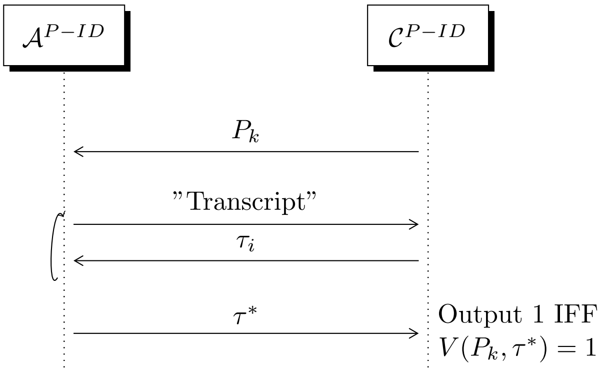
6.1 point a

Active $\implies$ Passive



We can construct a Π_{BAD} s.t. Passive $\not\Rightarrow$ Active, where if we play as a dishonest Verifier we can leak the entire S_k.

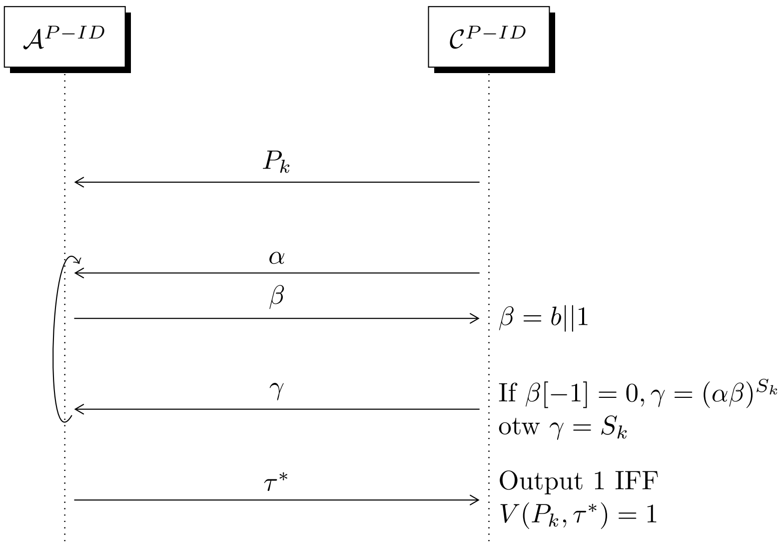
For the passive Game:



Now τ_i will be as follows, from the point of view of A:

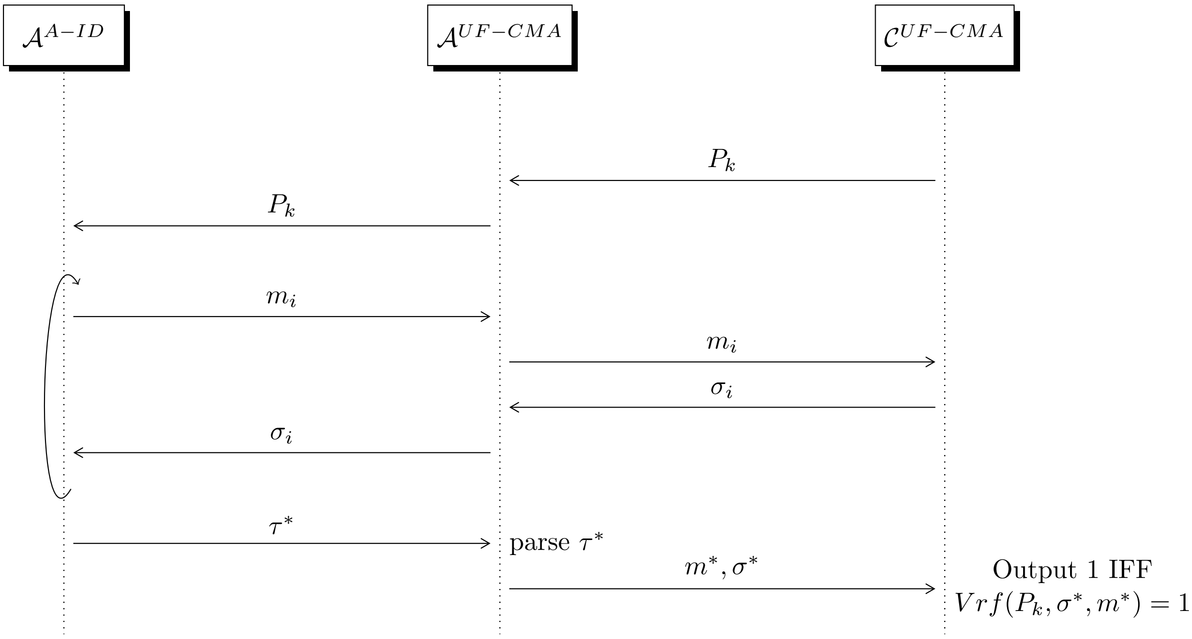
- $\alpha \leftarrow U$
- $\beta = b||0$ where $b \leftarrow \$U$
- $\gamma = \beta[-1](S_k) + (1 - \beta[-1])(\alpha\beta)^{S_k}$

So the above ID-scheme has still Passive Security. Instead for the Active Security we can play the following interaction:



Now that we have S_k we can always create a valid τ^*.

6.2 point b



6.3 point c

The HVZK property comes from the fact that the signature scheme used in the following way:

- The Verifier sends the message
- The Prover signs the message

doesn't leak any information about the secret used for signing the message (assuming A always as honest verifier).