

Problem 1: Cryptosystems secure against side-channel attacks

Solution: Consider the PRF $F' : \{0, 1\}^{n+1} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$

$$F'(k || b_k, x) = \begin{cases} F(k, 0^n)[1 \dots n-1] || b_k & \text{if } x = 0^n \\ F(k, x) & \text{Otherwise} \end{cases}$$

In other words, the last bit of $F(k, 0^n)$ has been replaced with the last bit of the key.

- (a) Let \mathcal{A} be an adversary which breaks the PRF security of F' with non-negligible advantage ϵ . We will build a reduction \mathcal{B} which breaks the PRF security of F with the same advantage.

Problem 1(a)

- Challenger picks a uniformly random bit $b \leftarrow \{0, 1\}$ and a key $k \leftarrow \mathcal{K}$.
- \mathcal{B} samples a random $b_k \leftarrow \{0, 1\}$.
- The adversary \mathcal{A} makes polynomially many queries $\{x_i\}$ to \mathcal{B} who passes them to the challenger. Challenger replies as in the PRF Game.
- Upon receiving the response y_i of each query, \mathcal{B} checks if $x_i = 0$. If so, it modifies y_i by exchanging its last bit with b_k . Otherwise, it just passes y_i to \mathcal{A} .
- After polynomially many queries, \mathcal{B} forwards the response send by \mathcal{A} (b') and wins if $b = b'$.

Figure 1: Reduction for Problem 1(a)

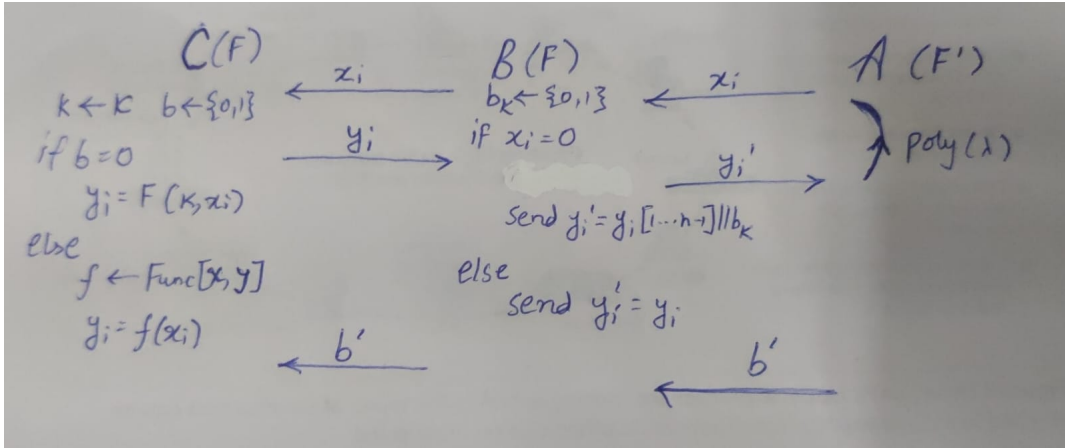


Figure 2: Image for Problem 1(a) Image

When the challenger chooses $b = 0$, the game is equivalent to the challenger choosing 0 in PRF game of F' .

$$\Pr[b' = 0 | b = 0] = \Pr[\mathcal{A} \text{ outputs zero when the challenger chooses 0 in PRF game of } F']$$

When the challenger chooses $b = 1$, \mathcal{A} receives the output of a random function for all $x_i \neq 0^n$. For $x_i = 0^n$, the output received is $r || b_k$. Since b_k is chosen randomly, this too is random.

$$\Pr[b' = 0 | b = 1] = \Pr[\mathcal{A} \text{ outputs zero when the challenger chooses 1 in PRF game of } F']$$

Hence we can conclude,

$$\text{PRFAdv}[\mathcal{B}, F] = \text{PRFAdv}[\mathcal{A}, F']$$

(b) We will show that F' does not satisfy 1-leakage resilience by constructing an adversary \mathcal{A}' who makes a leakage query for the last bit of the key and breaks F' .

- **Leakage Query:** \mathcal{A}' makes a query for the last bit of the key and receives b_k from the challenger.
- **PRF Query:** \mathcal{A}' queries for the $x = 0^n$ and receives y_i . He checks if the last bit of y_i is b_k . If yes it outputs $b' = 0$ (PRF), otherwise it outputs $b' = 1$ (Random Function).

From the game and definition of F' , it is evident that:

$$\Pr[b' = 0 | b = 0] = 1$$

When the challenger chooses $b = 0$, the evaluation of a random function at 0^n can have its last bit as 0 or 1 with $1/2$ probability. So,

$$\Pr[b' = 0 | b = 1] = \frac{1}{2}$$

And the advantage of \mathcal{A}' is

$$\text{PRFAdv}[\mathcal{A}, F'] = \Pr[b' = 0 | b = 0] - \Pr[b' = 0 | b = 1] = 1 - \frac{1}{2} = \frac{1}{2}$$

Which is non-negligible.

Problem 2 : MACs: unique queries vs non-unique queries

Solution:

Problem 3 : A mistake in the lecture notes

Solution: According to the given flawed argument, for any (even unbounded) adversary \mathcal{A} who wins the MAC game with verification queries (MAC^{vq}) with advantage ϵ , we can construct an adversary \mathcal{B} who wins the MAC game without verification queries (MAC) with probability ϵ . However, we will show an adversary \mathcal{A}' who wins macvq with advantage 1 but the reduction \mathcal{B} cannot use it to win MAC.

The key observation here is that since every message has a unique signature, \mathcal{B} cannot send a forgery of a message which it has already queried.

- \mathcal{A}' sends verification queries $(\text{Verify}, m, \sigma) \forall \sigma \in \mathcal{T}$ where \mathcal{T} is the signature space.
- For the first verification query, \mathcal{B} queries the challenger to obtain the signature σ^* , and checks all the verification queries against this.

One of the queries by \mathcal{A}' must be $(\text{Verify}, m, \sigma^*)$ and thus he wins the MAC^{vq} game. However, \mathcal{B} cannot use this forgery to win the MAC game since he has already queried it from the challenger.

Problem 4 : Even-Mansour instantiated with a bad permutation

Solution: The key observation here is that for any query x_i which results in an output y_i :

$$(y_i - k_2)(x_i + k_1) = 1 \pmod{p}$$

So, we query the oracle at 3 points 0,1,2 and form three equations:

$$(y_0 - k_2)(k_1) = 1 \pmod{p}$$

$$(y_1 - k_2)(1 + k_1) = 1 \pmod{p}$$

$$(y_2 - k_2)(2 + k_1) = 1 \pmod{p}$$

On solving we can calculate (k_1, k_2)

$$k_1 = 2(y_1 - y_2)(y_0 + y_2 - 2y_1)^{-1}$$

$$k_2 = y_1 - k_1(y_0 - y_1)$$

Now, we can just query π and check if these (k_1, k_2) satisfy $P(x_i) = \pi(x_i + k_1) + k_2$

Problem 5 : 3-round Luby-Rackoff with inversion queries

Solution:

Problem 6 : CBC mode with bad initialization

Solution:

Problem Part B : Theoretical Problem