N.	Tema	Descrizione
34	CBC MAC	modification of CBC MAC are secure ? (i), (ii), (iii)
49	CBC MAC	show CBC Modif. with IV chosen from U rather than 0^n is not secure?
50	CBC MAC	show variant k=l(n)*n and |m|=j*n is still secure ?
51	CBC MAC	SSH v2
54	CBC MAC	variant C0=0^n, ci=EK1(ci-1⊕Mi) CBC-MACk1,k2=cm⊕K2 show the tag
36	FEISTEL PRP	Feistel permutation
17	Hash	Universal H → pairwise indipendent H
18	Hash	Universal H from F finite field h(s,x)=∑si*xi
31	Hash	h(s,m)=A*m+b with A matrix and case B) as Toeplitz Matrix
37	Hash	CHR is seeded OWF ? Does C Resistant -> OWF in this case?
38	Hash	H(s1,s2,x)=H'(s2, H(s1,x))
81	Hash	a) twise indipendent, 3-wise indipendent
82	Hash	min entropy, entropy loss, statistical error
41	ID SCHEME	Show Πt is canonical ID. Challenge Space? Prove SS and HVZK
44	ID SCHEME	Actively Secure ID schemes
88	ID SCHEME	Prove passively is weaker from active, prove uF-CMA, HVZK
13	MAC	Define 2-time Ɛ-secure MAC
16	MAC	MAC and OTP not 1-time statistically secure
35	MAC	UF-CMA and UF-CMVA
45	MAC	MAC with t=Fk(0 ||m0)||Fk(1 ||m1) is UF-CMA?
46	MAC	MAC with t=(Fk(m1, Fk(Fk(m2))) is UF-CMA secure ?
47	MAC	i) t=Fk(m1)⊕Fk(m2)….⊕Fk(ml) ii) t=t=F(r)⊕Fk(m1)⊕Fk(m2)….⊕Fk(ml) …
48	MAC	show MAC having unique tag is not CPA
52	MAC	a) MAC by Merkle not secure b) MAC secure if H CR c) Soundness Oracle
53	MAC	tag=Fk(m)||Fk(m) secure ? Strong secure ? Strong Mac ->Z PRF?
78	MAC	=Exercise 52
8	MAC & PKE	Selective Unforgeability (SF-CMA, UF-CMA)
33	MAC & PRF	Tag(k || m1||m2)=Fk(m1) || Fk(Fk(m2)) is secure ?
15	Negligible	Neglibile nd noticeable functions
14	NT	Verifica firma numericamente (N=221 ed e=13)
39	NT	CDH, RSA modulus N=18830129, Alice and Bob (bezout identity)
55	NT	Z*p, h is QR mod p iff h^q=1 mod p, Dlog lsb(x) is not HCP for exp, Z*23
64	OFB	Output feedback mode
19	OWF	inefficient A1 and efficient A2 with Pr=2^(-n)
20	OWF	f(x+y)=x+y
21	OWF	g(x1,x2)=(f(x1),x2)
22	OWF	g(x)=(f(x),f(f(x)))
23	OWF	g(x)=f(x ||0)
56	OWF	g(x1,x2)=(x1, f(x2)) gi OWF and if f has HCP --> g has
83	OWF	PRG with 1 bit stretch prove it is also an OWF
83	OWF	PRG with 1 bit stretch prove it is also an OWF
84	OWF	g with n+logn to n+logn+1 show is OWF if f is
9	PKE	Π and Π' s.t. m0=Dec(sk,c1) and m1=Desc(sk,c2)
39a	PKE	CCA1 and CCA2, CCA1 -> CPA
65	PKE	Key exchange protocol
66	PKE	ElGamal decryption and CPA secure Decisional DH
67	PKE	DH efficient algorithms
68	PKE	Elgmal not IND-CCA secure
69	PKE	= Exercise 9

70	PKE	Text Book Elgamal (not CCA secure)
71	PKE	Threshold ElGamal
77	PKE	Consider FDH-RSA and Weak UF-CMA and RSA assumption
79	PKE	Dlog and Incremental Hashing
27	PRF	Fk(x)=G'(k)⊕x
28	PRF	Fk(x)=Fx(k)
29	PRF	F'k(x)=Fk(x||0) || Fk(x||1)
43	PRF	F(k1,k2,x)=k1⊕x⊕k2 show F it is not a PRF
61	PRF	= Exerc. 43 F(k1,k2,x)=k1⊕x⊕k2 show F it is not a PRF
63	PRF	weak pseudo random
7	PRG	Prg with Weak Seed
24	PRG	Show no PRG can be secure against comput. Unbounded adv. Lenght doubl.
25	PRG	G''(s)=G(s1)||... ||G(sn)
26	PRG	G''(s)=G(s1) || 0^|s|)
42	PRG	G'(s)=f(G(s)) show G' is PRG if G is PRG
57	PRG	Let G be a lenght-doubling PRG. Prove G is OWF
58	PRG	Let G a PRG and G'(s)=G truncated to n bits (n=|s|) Fk(x)=G'(k)⊕x is not PRG
59	PRG	Let G'(k1,k2)G(k1)VG(k2) V=bit-wise logical OR, efficient PRG
85	PRG	Construct a PRG from 2 PRG (only 1 is real PRG). Optimal seed lenght
87	PRP & SKE	= ex 32-60-62 c=Pk(r||m)
1	PS	PS with K<M and t>1
2	PS	PS with K<M and t>1
3	PS	M=(0,4) K=(0,5)
3	PS	last bit of is 0, m ⊕ (k||0)
4	PS	(m+k) mod 4, prove or disprove
5	PS	K=M, K Ɛ Unif → PS ?
6	PS	OTP without elemento 0 on K is still PS ?
6a	PS	A) c= m ⊕ f(0^n) B) m ⊕ Fk(0^n), A) & B) are PS ? C) is Π CPA Secure?
10	PS	perfect secrecy for 2 messages with all M and for all C
11	PS	OTP without elemento 0 on K is still PS ?
12	PS	Refute for all M Distribut, for all m,m', for all c (hint Pr[m0]=3/4
80	PS	PS equivalent to Pr[Gme=1]=1/2
40	SIG SCHEME	Prove Π is UF-CMA and H CR then Π' is UF-CMA
75	SIG SCHEME	Non adaptive and Weak Unforgeability (naUFCMA)
76	SIG SCHEME	=Exercise 40
30	SKE	equivalence of computational 1-time secure to indistinguashability
72	SKE	Counter mode encryption
73	SKE	PRP and Ik(C0)⊕C1
74	SKE	Block Ciprhers in ECB and CBC mode
86	SKE	Recall CBC mode operation
60	SKE & PRF	= Ex. 32 and ex 62, c=Fk(r||m) prove CPA and CCA if strong PRP
62	SKE & PRF	= Ex. 32 and ex 60, c=Fk(r||m) prove CPA and CCA if strong PRP
32	SKE & PRP	strong PRP and CCA SKE with Enc(k,m)=Pk(m||r)

Affidabilità	Fonte
alta	Hw1 2018
alta	Burmeste
bassa	Maryland
bassa	Zurigo
bassa	Saarland
alta	Hw2 2017
alta	Hw1 2016
alta	
bassa	no solut.
alta	Hw2 2018
alta	Hw1 2018
alta	Hw1 2018
alta	Hw1 2018
bassa	Hw2 2017
bassa	ex 2017
alta	Hw2 2018
bassa	HW1 2017
alta	
alta	Hw1 2018
alta	Katz 4.6
alta	Katz 4.6
alta	Katz 4.7
alta	Burmeste
alta	Saarland
alta	Saarland
alta	Saarland
media	
alta	
alta	
alta	
alta	Hw2 2018
bassa	Saarland
alta	Hw1 2018
alta	Hw1 2016
alta	Hw1 2016
alta	Hw1 2016
alta	Hw1 2016
alta	Hw1 2016
bassa	Cornelious
alta	Hw1 2018
alta	Hw1 2018
alta	Hw1 2018
alta	Saarland
alta	Hw2 2018
bassa	Katz 10.4
bassa	Katz 11.6
bassa	
bassa	
alta	Saarland

media	
bassa	
alta	Hw2 2018
bassa	
alta	Hw1 2018
alta	Hw1 2018
alta	Hw1 2018
alta	Saarland
alta	Saarland
bassa	
media	Exam 01
alta	Hw1 2016
alta	Hw1 2016
alta	Hw1 2016
alta	Saarland
bassa	Cornelious
alta	Cornelious
bassa	Cornelious
alta	Hw1 2018
alta	Hw2 2018
alta	Katz 2.11
alta	Katz 2.11
alta	Katz 2.6
alta	Katz 2.6
alta	
alta	
alta	Hw1 2016
alta	Saarland
alta	Katz 2.13
alta	
alta	Katz 2.3
alta	Hw1 2018
media	Hw2 2017
bassa	exam
bassa	Hw 2017
alta	Hw1 2016
bassa	Maryland
bassa	Trevisan
bassa	Zurigo
alta	Hw1 2018
media	Katz 3.18
media	Katz 3.18
media	Hw1 2017

1_(PERFECT SECRECY) (FROM KATZ)

2.11 Assume we require only that an encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$ satisfy the following: For all $m \in \mathcal{M}$, we have $\Pr[\mathsf{Dec}_K(\mathsf{Enc}_K(m)) = m] \geq 2^{-t}$. (This probability is taken over choice of the key as well as any randomness used during encryption.) Show that perfect secrecy can be achieved with $|\mathcal{K}| < |\mathcal{M}|$ when $t \geq 1$. Prove a lower bound on the size of $\mathcal{K}$ in terms of t.

Explanation

When the correctness requirement is weakened the encryption scheme can omit part of the message m (of length $|m|$) to be encrypted and just "loose" it in a way that the cipher (the output of the Encrypt method) is totally independent of that part. Thus the message that effectively is encrypted is not m but rather just part of it of length $|\ell| < |m|$ and thus the key doesn't have to be of length at least $|m|$ but rather be of length at least $|\ell|$.

Example

The most basic example is an encryption using the One-Time-Pad Scheme, in a model where the correctness requirement from the scheme is weakened to:
$Pr[Dec_k(Enc_k(m)) = m] = \frac{1}{2}$. Take a message m of length $|m|$ and a random key $k \in \{0,1\}^{|m|-1}$; the encryption method take the first $|m| - 1$ bits of m, call it m' and output the cipher $c = m' \oplus k$. Then, given c and k, we can decipher (or decrypt) the cipher correctly with probability of exactly $\frac{1}{2}$ by computing $m' = c \oplus k$ and then randomly guessing the $|m|^{th}$ bit of m; since we get the correct bit with probability $\frac{1}{2}$ we satisfy the correctness requirement.

About the bound

We can say the when the correctness requirement is $Pr[Dec_k(Enc_k(m)) = m] \geq \frac{1}{2^t}$ the key space must be of length no less then $|m| - t$. If the key in the example above is of length $|m| - t$ then the decryption algorithm can guess the omitted t bits of the message with probability $\frac{1}{2^t}$ as required.

2_(PERFECT SECRECY)

Assume that we require only that an encryption scheme (Gen, Enc, Dec) over a message space $\mathcal{M}$ satisfies the following: for all $m \in \mathcal{M}$, the probability that $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$ is at least 2^{-t}. (This probability is taken over choice of k as well as any randomness that may be used during encryption or decryption.) Show that perfect secrecy (as in Definition 2.1) can be achieved with $|\mathcal{K}| < |\mathcal{M}|$ when $t \geq 1$. Can you guess a lower bound on the required size of $\mathcal{K}$?

Solution: Let $\mathcal{K} = \{0;1\}^\ell$ and $\mathcal{M} = \{0;1\}^{\ell+t}$. The key-generation algorithm chooses a uniform string from $\mathcal{K}$. To encrypt a message $m \in \mathcal{M}$ using key k, let m' denote the first ℓ-bits of m and output $c := m' \oplus k$ (both m' and k have length ℓ). To decrypt a ciphertext c using key k, choose a random string $r \leftarrow \{0;1\}^\ell$ and output $m := (c \oplus k)||r$. Note that $\Pr[\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m] = 2^{-t}$ because decryption is correct if and only if the random string r chosen during decryption happens to equal the last t bits of m (and this occurs with probability 2^{-t}). Perfect secrecy of this scheme follows from the proof of the one-time pad (indeed, this is exactly a one-time pad on the first ℓ-bits of the message).

Lower bound: $|\mathcal{K}| \geq |\mathcal{M}| \cdot 2^{-t}$.

3_(PERFECT SECRECY)

2.6 For each of the following encryption schemes, state whether the scheme is perfectly secret. Justify your answer in each case.

(a) The message space is $\mathcal{M} = \{0, \ldots, 4\}$. Algorithm Gen chooses a uniform key from the key space $\{0, \ldots, 5\}$. $\mathsf{Enc}_k(m)$ returns $[k + m \bmod 5]$, and $\mathsf{Dec}_k(c)$ returns $[c - k \bmod 5]$.

(b) The message space is $\mathcal{M} = \{m \in \{0,1\}^\ell \mid \text{the last bit of } m \text{ is } 0\}$. Gen chooses a uniform key from $\{0,1\}^{\ell-1}$. $\mathsf{Enc}_k(m)$ returns ciphertext $m \oplus (k\|0)$, and $\mathsf{Dec}_k(c)$ returns $c \oplus (k\|0)$.

(b)

The definition of **perfectly secret** which states: An encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$ is **perfectly secret** if for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $\Pr[C = c] > 0$:

$$\Pr[M = m \mid C = c] = \Pr[M = m].$$

We first compute $\Pr[C = c \mid M = m']$ for arbitrary $c \in \mathcal{C}$ and $m' \in \mathcal{M}$.

$$\begin{aligned} \Pr[C = c \mid M = m'] &= \Pr[Enc_K(m') = c] = \Pr[m' \oplus (K \parallel 0) = c] \\ &= \Pr[(K \parallel 0) = c \oplus m'] = 2^{1-\ell} \quad (1) \end{aligned}$$

where the final equality holds because the key K is a uniform $\ell - 1$-bit string. Fix any distribution over $\mathcal{M}$. For any $c \in \mathcal{C}$, we have

$$\begin{aligned} \Pr[C = c] &= \sum_{m' \in \mathcal{M}} \Pr[C = c \mid M = m'] \cdot \Pr[M = m'] \\ &= 2^{1-\ell} \cdot \sum_{m' \in \mathcal{M}} \Pr[M = m'] = 2^{1-\ell} \cdot 1 = 2^{1-\ell} \quad (2) \end{aligned}$$

where the sum is over $m' \in \mathcal{M}$ with $\Pr[M = m'] \neq 0$. Bayes' Theorem gives:

$$\begin{aligned} \Pr[M = m \mid C = c] &= \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{\Pr[C = c]} \\ &= \frac{2^{1-\ell} \cdot \Pr[M = m]}{2^{1-\ell}} = \Pr[M = m] \end{aligned}$$

a)

c	k									
m	**0**	**1**	**2**	**3**	**4**	**5**	c=	**0**	6	**Pr[c=x]=6/20** ∀ x ε (0,4) implica perfect secrecy
0	0	1	2	3	4	0	c=	1	6	
1	1	2	3	4	0	1	c=	2	6	
2	2	3	4	0	1	2	c=	3	6	
3	3	4	0	1	2	3	c=	4	6	
4	4	0	1	2	3	4				

Questi qui sotto sono sbagliati (secondo me):

a) PROVE IF IS PERFECT SECRET

$\mathcal{M} = \{0, \dots, 4\}$

$\mathcal{K} = \{0, \dots, 5\}$

$ENC_k(m) = k + \underset{}{m}^{\nearrow c} \text{ MOD } 5$

$DEC_k(c) = c - k \text{ MOD } 5$

TO BE PERFCT SECRET: $PR\left[ENC_k(m) = c\right] = 2^{\ell}$ KEY SPACE = 5 $\Rightarrow 2^9 \neq 2^5$ NO!

2^9

---

b) PROVE IF IS PERFECT SECRET

$\mathcal{M} = \{0,1\}^{\ell-1} \| 0$

$Gen \leftarrow \$ \ |k|^{\ell-1}$

$ENC : m \oplus k \| 0$

$DEC : c \oplus k \| 0$

WE ALWAYS HAVE THE LAST BIT = 0

IT IS NOT PERFECT BECAUSE WE PARTITION THE SET OF C IN TWO PARTS ((c THAT END IN 0, c THAT END IN 1) SO THE CIPHERTEXT ISN'T UNIFORMLY DISTRIBUTED OVER C SPACE

4_(PERFECT SECRECY)

Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ be an encryption scheme with key space $\mathcal{K}$, message space $\mathcal{M}$, and ciphertext space $\mathcal{C}$ where $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0, 1, 2, 3\}$. Algorithm Gen returns a uniformly random key k in $\mathcal{K}$. For any key k in $\mathcal{K}$ and any message m in $\mathcal{M}$, $\text{Enc}(m)$ using key k is defined as $(m + k) \bmod 4$. For any key k in $\mathcal{K}$ and ciphertext c, $\text{Dec}(c)$ using key k is defined as $(c - k) \bmod 4$.

a) Prove that $\text{Dec}(\text{Enc}(m)) = m$ using key k holds for any key k in $\mathcal{K}$ and any message m in $\mathcal{M}$.

b) Prove or disprove: Π is perfectly secret.

1. For every key $k \in \mathcal{K}$

$$Dec(Enc(m)) \stackrel{\text{def}}{=} (Enc(m) - k) \bmod 4,$$

where

$$Enc(m) \stackrel{\text{def}}{=} (m + k) \bmod 4,$$

so, combining it, we have

$$Dec(Enc(m)) = (((m + k) \bmod 4) - k) \bmod 4,$$

which, by properties of modular arithmetic (or, more precisely, congruences), is equivalent to

$$(m \bmod 4) + (k \bmod 4) - (k \bmod 4) = m + k - k = m$$

2. This is actually a great example, as it demonstrates the ultimate elegance of the mathematical definition of security. Encryption scheme (E, D) over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ is perfectly secret, if for all $m_i, m_j \in \mathcal{M}, c \in \mathcal{C}$

$$Pr[E(k, m_i) = c] = Pr[E(k, m_j) = c],$$

where k is uniform in $\mathcal{K}$. Here is a table of encryptions of m with key k for all $m \in \{0, 1, 2, 3\}$ and $k \in \{0, 1, 2, 3\}$:

m/k	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

As you can see, for every m, $Pr[Enc(m) = c] = 1/4$, thus given encryption scheme is perfectly secure.

5_(PERFECT SECRECY)

Prove or refute: Every encryption scheme for which the size of the key space equals the size of the message space, and for which the key is chosen uniformly from the key space, is perfectly secret.

Solution: This is false. Let the key space and message space be the set of ℓ-bit strings, and consider the encryption scheme defined by choosing a random key and setting $\mathsf{Enc}_k(m) = k \| m$ (where $\|$ denotes concatenation). This scheme fulfills the requirements of the exercise but is not secret at all; the plaintext is always the last ℓ-bits of the ciphertext.

6_(PERFECT SECRECY)_(OTP)

2.3 When using the one-time pad (Vernam's cipher) with the key $k = 0^\ell$, it follows that $\mathsf{Enc}_k(m) = k \oplus m = m$ and the message is effectively sent in the clear! It has therefore been suggested to improve the one-time pad by only encrypting with a key $k \neq 0^\ell$ (i.e., to have Gen choose k uniformly at random from the set of non-zero keys of length ℓ). Is this an improvement? In particular, is it still perfectly secret? Prove your answer. If your answer is positive, explain why the one-time pad is not described in this way. If your answer is negative, reconcile this fact with the fact that encrypting with 0^ℓ doesn't change the plaintext.

Solution: The modified scheme is not perfectly secret. To see this formally, consider the uniform distribution over $\mathcal{M} = \{0,1\}^\ell$. For any fixed message $\alpha \in \{0,1\}^\ell$, we have

$$\mathsf{Pr}[M = \alpha | C = \alpha] = 0 \neq \mathsf{Pr}[M = \alpha].$$

This contradicts perfect secrecy. We conclude that in order to obtain perfect secrecy, it must be possible to encrypt using the key 0^ℓ. This may seem counter-intuitive, since this key does not change the plaintext. However, note that an eavesdropper has no way of knowing if the key is 0^ℓ, so the fact that the ciphertext is the same as the plaintext in this case is really of no help to the adversary.

6a_(PERFECT SECRECY)_(SAARLAND)

Exercise No.4 (Perfect secrecy and indistinguishable encryptions)

(a) *(4 points)* Let func_n be the set of all functions $f : \{0,1\}^n \to \{0,1\}^n$ and consider the following encryption scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$:

- On input 1^n, Gen outputs an $f \in \mathsf{func}_n$ uniformly at random.
- Given a key $f \in \mathsf{func}_n$ and a message $m \in \{0,1\}^n$, Enc outputs the ciphertext $c = m \oplus f(0^n)$.
- Given a key $f \in \mathsf{func}_n$ and a ciphertext $c \in \{0,1\}^n$, Dec outputs the plaintext $m = c \oplus f(0^n)$.

Prove that Π is perfectly secret. (You can use any characterization of perfect secrecy that was introduced in the lecture.)

(b) *(10 points)* Now let F_k be a length-preserving PRF and consider the following encryption scheme $\Pi' = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$:

- On input 1^n, Gen outputs an $k \in \{0,1\}^n$ uniformly at random.
- Given a key k and a message $m \in \{0,1\}^n$, Enc outputs the ciphertext $c = m \oplus F_k(0^n)$.
- Given a key k and a ciphertext $c \in \{0,1\}^n$, Dec outputs the plaintext $m = c \oplus F_k(0^n)$.

Prove that Π' has indistinguishable encryptions in the presence of an eavesdropper. You may use the result from part (a).
Hint: It may be advisable to prove (by reduction) that the existence of an adversary that wins the adversarial indistinguishability experiment with probability non-negligibly greater than $\frac{1}{2}$ can be used to show that F_k is not a PRF.

(c) *(1 point)* Prove or disprove: Π' is CPA-secure.

Solution No.4 (Perfect secrecy and indistinguishable encryptions) (a) We know from the lecture that the one-time pad is perfectly secret , and one can see as follows that the encryption scheme is simply a different way of defining the one-time pad: Guessing a function $f : \{0,1\}^n \to \{0,1\}^n$ is the same as guessing 2^n elements from $\{0,1\}^n$ (i.e., guessing the image of every element in the domain). We can interpret the outcome of this such that the first string is the image of 0^n under f, which is then nothing else as a random element from $\{0,1\}^n$.

This means that picking $f(0^n)$ for uniformly random f is the same as picking uniformly at random some $k \in \{0,1\}^n$. The encryption scheme is then identical to the one-time pad, as claimed.

2) $PR[ENC(m)=c] = PR[m \oplus f(0)=c] = PR[f(0)=c \oplus m] = 2^{-n}$
$= PR[ENC(m')=c]$ (by rule 3)

(b) Let $\mathcal{A}$ be an adversary for the adversarial indistinguishability experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi'}$, and assume $\mathcal{A}$ wins with probability non-negligibly greater than $\frac{1}{2}$, say

$$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi'} = 1] \geq \frac{1}{2} + \frac{1}{p(n)}$$

for some positive polynomial p. We want to show how to use this to distinguish F_k from a truly random function, and we can do this as follows: Construct a distinguisher D that on input 1^n performs the experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\cdot}$ step-by-step, simulating the adversary $\mathcal{A}$ during the process and using the oracle $\mathcal{O}$ for encryption:

(1) Simulate the adversary $\mathcal{A}$ on input 1^n until it outputs the messages m_0, m_1

(2) Choose uniformly at random a bit $b \in \{0,1\}$ and generate the challenge ciphertext $c \leftarrow \mathcal{O}(0^n) \oplus m_b$

(3) Continue simulating $\mathcal{A}$ on c, until it outputs a bit b'

(4) Output 1 if $b' = b$ and 0 otherwise.

Clearly, since $\mathcal{A}$ is PPT, so is D. Additionally, by the very definition of $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\cdot}$, D behaves exactly like $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}$ or $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi'}$, depending on what $\mathcal{O}$ is, such that

$$\Pr_{k \in \{0,1\}^n}[D^{F_k(\cdot)} = 1] = \Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi'}] = 1]$$

and

$$\Pr_{f \in \mathsf{func}_n}[D^{f(\cdot)} = 1] = \Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} = 1]\,.$$

Then, since perfect secrecy implies that

$$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} = 1] = \frac{1}{2}\,,$$

we have

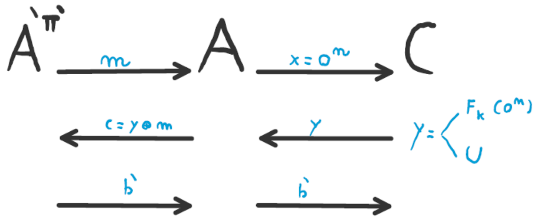
$$\Pr_{f \in \mathsf{func}_n}[D^{f(\cdot)} = 1] = \frac{1}{2}\,,$$

and thus

$$\begin{aligned}
&\left|\Pr_{k \in \{0,1\}^n}[D^{F_k(\cdot)}(1^n) = 1] - \Pr_{f \in \mathsf{func}_n}[D^{f(\cdot)}(1^n) = 1]\right| = \\
&\left|\Pr_{k \in \{0,1\}^n}[D^{F_k(\cdot)}(1^n) = 1] - \frac{1}{2}\right| \geq \\
&\Pr_{k \in \{0,1\}^n}[D^{F_k(\cdot)}(1^n) = 1] - \frac{1}{2} \geq \\
&\frac{1}{2} + \frac{1}{p(n)} - \frac{1}{2} = \frac{1}{p(n)}
\end{aligned}$$

which is non-negligible.

b)



(c) It cannot be, since encryption happens deterministically.

7_PRG (WEAK SEEDS FORM ARTICLE BY VENTURY)

1 PRGs with Weak Seeds 10 Points

Let $G : \{0,1\}^m \to \{0,1\}^{2m}$ be a $(t_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}})$-secure PRG. Explain how to safely use G in a setting where the seed S, instead of being uniform, is such that $\mathbb{H}_2(S) \geq m - d$. Assuming $m = 128$ and $d = 8$, show how to choose the parameters $t_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}}$ in such a way that your construction achieves security 2^{-80} against all adversaries running in time at most 2^{20}.

From the Alternative Dense Model Theorem, in the real-model, with weak seed S, small entropy deficiency d and collision entropy $H_2\,(S)$, the new PRG:

$G(h_{G(X)}(S)) \quad is \quad (t_{prg}/2, \varepsilon_{prg} + \sqrt{2^d \varepsilon_{prg}}\,)$ -secure, pseudorandom conditioned on S with d=8

Moreover taking into account that T/ε=2^n is the n bit security (*), for a uniform distribution and so for weak seed distribution, you have: $(2^{20})/(2^{-80}2^{d=8}) = 2^{20}/2^{-72} = 2^n$ -> n=92

(*) While there is no universally accepted, general, formal definition of n bit security, I used the most common for any attack with cost T and success probability ε, it must be T/ε> 2^n.

8_(MAC and Public Key)

2 Selective Unforgeability 10 Points

Define a variant of universal unforgeability against chosen-message attacks (UF-CMA) for digital signatures, in which the adversary has to commit to the message $m^* \in \mathcal{M}$ on which he will forge a signature before seeing the public key (where $\mathcal{M}$ is the message space); note that in the definition the adversary should still be allowed to sign arbitrary messages. Name your notion SF-CMA (i.e., selective unforgeability against chosen-message attacks). Prove or disprove:

(a) UF-CMA $\Rightarrow$ SF-CMA.

(b) SF-CMA $\Rightarrow$ UF-CMA.

We now introduce the notions of the existential unforgeability against the chosen message attack (euf-cma) and the selective unforgeability against the chosen message attack (suf-cma). Let $\mathsf{Sig} := (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme. The *ef-cma game* is defined in the following way: on a security parameter λ,

EF Init $\mathcal{F}$ is given a public key pk such that a challenger $\mathcal{C}$ generates $(sk, pk) \leftarrow \mathsf{KGen}(1^\lambda)$.

Signing Oracle When $\mathcal{F}$ hands an i-th message $\overline{m}_i$ to $\mathcal{C}$, $\mathcal{C}$ replies its signature $\overline{\sigma}_i \leftarrow \mathsf{Sign}(sk, pk, \overline{m}_i)$.

EF Challenge When $\mathcal{F}$ finally returns a pair (m^*, σ^*), $\mathcal{C}$ outputs 1 if $m^* \notin \{\overline{m}_i\}_i$ and $\mathsf{Ver}(pk, m^*, \sigma^*) = 1$.

Then $\mathcal{F}$ is said to *win the ef-cma game* if $\mathcal{C}$ outputs 1 in this game. In a similar manner, the *N-sf-cma game* is defined in the following way, where N is a polynomial in a security parameter λ: on a security parameter λ,

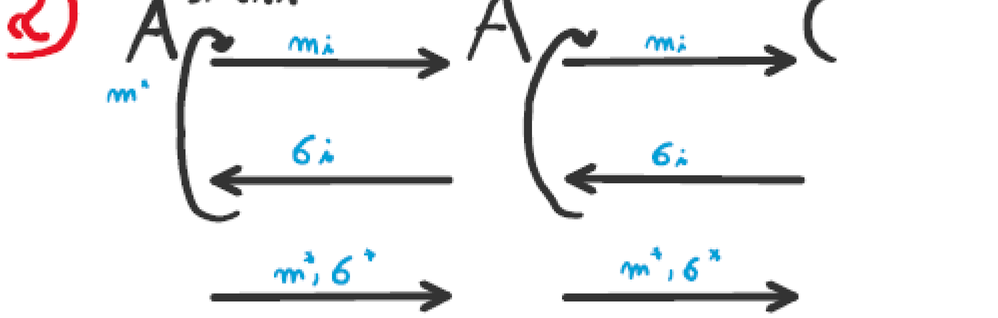
SF Init $\mathcal{F}$ is given a public key pk and a sequence $(m_1, m_2, \ldots, m_N)$ of N distinct messages such that a challenger $\mathcal{C}$ generates $(sk, pk) \leftarrow \mathsf{KGen}(1^\lambda)$ and samples $m_1, m_2, \ldots, m_N$ at random.

Signing Oracle It coincides with the one of the ef-cma game.

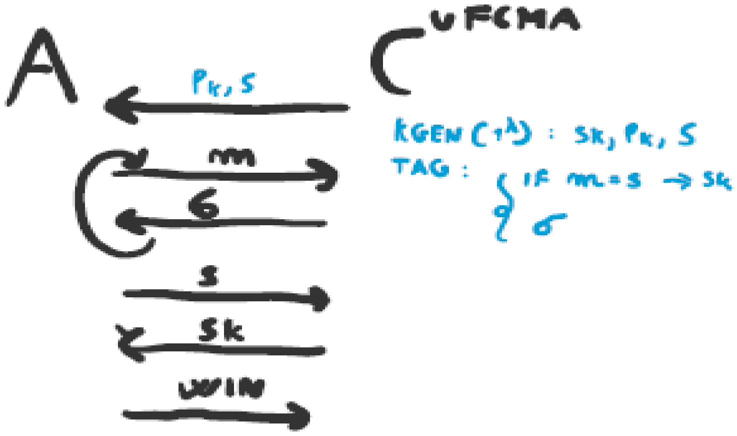
SF Challenge When $\mathcal{F}$ finally returns a pair (m_{i^*}, σ_{i^*}), $\mathcal{C}$ outputs 1 if $m_{i^*} \notin \{\overline{m}_i\}_i$ and $\mathsf{Ver}(pk, m_{i^*}, \sigma_{i^*}) = 1$.

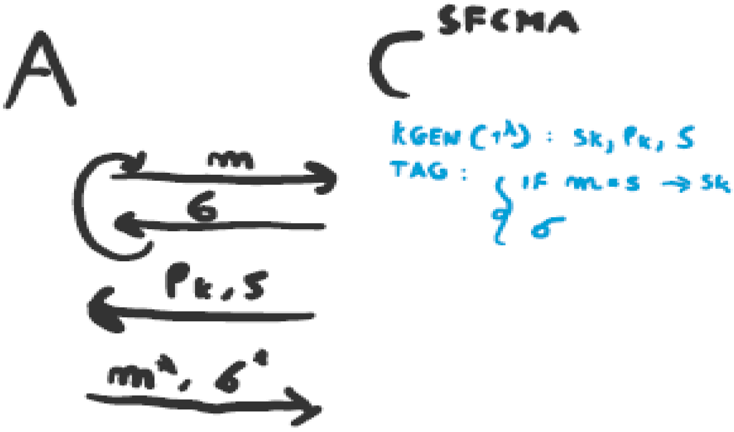
Then $\mathcal{F}$ is said to *win the N-sf-cma game* if $\mathcal{C}$ outputs 1 in this game. Let goal $\in \{\text{euf}, N\text{-suf}\}$. The signature scheme Sig is said to be *goal-cma* if for any PPT forger $\mathcal{F}$, $\mathcal{F}$ wins the corresponding game with negligible probability. The probability is taken over the internal coin flips of KGen and $\mathcal{F}$, and the choices of $m_1, m_2, \ldots, m_N$ only for the N-suf-cma. On the relationship between euf-cma and N-suf-cma, the following proposition holds.

Proposition 1 ([15])**.** Let Sig be a signature scheme, and let N be a polynomial in a security parameter λ. If Sig is euf-cma, then Sig is also N-suf-cma.



b)





9__(PUBLIC KEY)

Exercise No.5 (From public to private key) *(7 points)*
Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Construct a *private*-key encryption scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows:

- $\mathsf{Gen}'(1^n) := \mathsf{Gen}(1^n)$, that is, the single private key k of Π' is the pair (sk, pk) output by Gen.
- $\mathsf{Enc}'_{(pk,sk)}(m) := (\mathsf{Enc}_{pk}(m), \mathsf{Enc}_{pk}(m))$, that is, encryption of a message m produces a ciphertext (c_1, c_2), where for c_1 and c_2, encryption is performed independently as in Π, using only the part of k corresponding to pk.
- $\mathsf{Dec}'_{(pk,sk)}(c_1, c_2)$ is defined as follows: Let $m_0 := \mathsf{Dec}_{sk}(c_1), m_1 := \mathsf{Dec}_{sk}(c_2)$. Then, $\mathsf{Dec}'_{(pk,sk)}(c_1, c_2)$ is defined as $\perp$ if $m_0 \neq m_1$ (i.e., decryption failed), and as m_0 otherwise. That is, decryption is performed on both parts of the ciphertext as in Π, using only the second part sk of k. If both parts yield the same message, the algorithm outputs this message, and otherwise outputs an error.

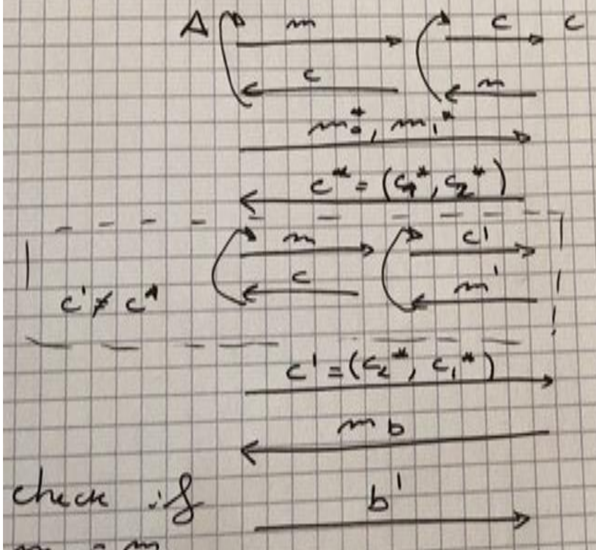
Show that Π' is *not* CCA-secure.

Solution No.5 (From public to private key) Construct an adversary $\mathcal{A}$ as follows: Output two messages $m_0 \neq m_1$ for the experimenter. Upon receiving the challenge ciphertext (c_1, c_2), query the decryption oracle to decrypt (c_2, c_1), receiving a message m'. Then, output b' with $m' = m_{b'}$. Only with negligible probability will it happen that $c_1 = c_2$ (otherwise Π wouldn't be CPA-secure), and hence, in all but negligibly many cases, $(c_1, c_2) \neq (c_2, c_1)$ and querying (c_2, c_1) is thus valid for the experiment. It is clear that, with all but negligible probability, m' is the same as m_b, where b is the bit chosen in the experiment.

$\Pi' \begin{cases} (sk, pk) \\ Enc_{pk}, Dec_{sk} \end{cases}$

$\Pi \begin{cases} k = (sk, pk) = sk \,\|\, pk \\ Enc_k = \overbrace{Enc_{pk}(m)}^{c_1}, \overbrace{Enc_{pk}(m)}^{c_2} \\ Dec_k = \underbrace{Dec_{sk}(c_1)}_{m_0}, \underbrace{Dec_{sk}(c_2)}_{m_1} \end{cases}$

Output m_0 iff $m_1 = m_0$



check if
$m_b = m_0$
then $b' = 0$
otw $b' = 1$

10

3. (2.13 in book) In this problem we consider definitions of perfect secrecy for the encryption of *two* messages (using the same key). Here we consider distributions over *pairs* of messages from the message space $\mathcal{M}$; we let M_1, M_2 be random variables denoting the first and second message, respectively. (We stress that these random variables are not assumed to be independent.) We generate a (single) key k, sample a pair of messages (m_1, m_2) according to the given distribution, and then compute ciphertexts $c_1 \leftarrow \mathsf{Enc}_k(m_1)$ and $c_2 \leftarrow \mathsf{Enc}_k(m_2)$; this induces a distribution over pairs of ciphertexts and we let C_1, C_2 be the corresponding random variables.

   (a) Say encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is *perfectly secret for two messages* if for all distributions over $\mathcal{M} \times \mathcal{M}$, all $m_1, m_2 \in \mathcal{M}$, and all ciphertexts $c_1, c_2 \in \mathcal{C}$ with $\Pr[C_1 = c_1 \wedge C_2 = c_2] > 0$:
   $$\begin{aligned} & \Pr[M_1 = m_1 \wedge M_2 = m_2 \mid C_1 = c_1 \wedge C_2 = c_2] \\ = \; & \Pr[M_1 = m_1 \wedge M_2 = m_2] \end{aligned}$$
   Prove that *no* encryption scheme can satisfy this definition.
   Hint: Take $c_1 = c_2$

   (b) Say encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is *perfectly secret for two messages* if for all distributions over $\mathcal{M} \times \mathcal{M}$ where the first and second messages are guaranteed to be different (i.e., distributions over pairs of *distinct* messages), all $m_1, m_2 \in \mathcal{M}$, and all $c_1, c_2 \in \mathcal{C}$ with $\Pr[C_1 = c_1 \wedge C_2 = c_2] > 0$:
   $$\begin{aligned} & \Pr[M_1 = m_1 \wedge M_2 = m_2 \mid C_1 = c_1 \wedge C_2 = c_2] \\ = \; & \Pr[M_1 = m_1 \wedge M_2 = m_2] \end{aligned}$$
   Show an encryption scheme that provably satisfies this definition.
   Hint: The encryption scheme you propose need not be efficient, although an efficient solution is possible.

Solution: The definition requires the equation to hold for any distribution over pairs of messages, any messages m_1, m_2 and any ciphertexts c_1, c_2. We consider the uniform distribution over $\mathcal{M} \times \mathcal{M}$, any m_1, m_2 such that $m_1 \neq m_2$ and some ciphertext $c \in \mathcal{C}$ with $\Pr[C_1 = c \wedge C_2 = c] > 0$. Note that such a c must exist since $\Pr[M_1 = M_2] > 0$ under the uniform message distribution, in which case $\Pr[C_1 = c \wedge C_2 = c] > 0$.
It is clear that $\Pr[M_1 = m_1 \wedge M_2 = m_2] = \frac{1}{|\mathcal{M}|^2}$ under the distribution that we chose. But due to the correctness of the scheme, we have that $\Pr[C_1 = c \wedge C_2 = c \mid M_1 = m_1 \wedge M_2 = m_2] = 0$ for $m_1 \neq m_2$, since otherwise decryption errors would necessarily have non zero probability. Applying Bayes' Theorem now shows that also $\Pr[M_1 = m_1 \wedge M_2 = m_2 \mid C_1 = c \wedge C_2 = c] = 0$, which contradicts the definition. Since this is true for any encryption scheme, no encryption scheme can achieve this definition. $\square$

b) Say encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is *perfectly secret for two messages* if for all distributions over $\mathcal{M} \times \mathcal{M}$ where the first and second messages are guaranteed to be different (i.e., distributions over pairs of *distinct* messages), all $m_1, m_2 \in \mathcal{M}$, and all $c_1, c_2 \in \mathcal{C}$ with $\Pr[C_1 = c_1 \wedge C_2 = c_2] > 0$:
$$\begin{aligned} & \Pr[M_1 = m_1 \wedge M_2 = m_2 \mid C_1 = c_1 \wedge C_2 = c_2] \\ = \; & \Pr[M_1 = m_1 \wedge M_2 = m_2] \end{aligned}$$

Show an encryption scheme that provably satisfies this definition.

Hint: The encryption scheme you propose need not be efficient, although an efficient solution is possible.

Solution: We show that a generlized version of the mono-alphabetic substitution cipher satisfies this definition. For any message space $\mathcal{M}$, we define the substitution cipher as:

- Gen: select a permutation π over $\mathcal{M}$ uniformly at random and return it as key.
- Enc: on input a message m and key π, return $\pi(m)$ as the ciphertext
- Dec: on input a ciphertext c and a key π, return $\pi^{-1}(c)$ as the message.

Clearly, this scheme is correct. For any distribution over distinct message pairs, we have

$$
\begin{aligned}
&\Pr[M_1 = m_1 \wedge M_2 = m_2 \mid C_1 = c_1 \wedge C_2 = c_2] \\
&= \frac{\Pr[C_1 = c_1 \wedge C_2 = c_2 \mid M_1 = m_1 \wedge M_2 = m_2]\Pr[M_1 = m_1 \wedge M_2 = m_2]}{\Pr[C_1 = c_1 \wedge C_2 = c_2]} \\
&= \frac{\Pr[C_1 = c_1 \wedge C_2 = c_2 \mid M_1 = m_1 \wedge M_2 = m_2]\Pr[M_1 = m_1 \wedge M_2 = m_2]}{\sum_{m_i \neq m_j}\Pr[C_1 = c_1 \wedge C_2 = c_2 \mid M_1 = m_i \wedge M_2 = m_j]\Pr[M_1 = m_i \wedge M_2 = m_j]}
\end{aligned}
$$

Now notice that for any distinct m, m', we have

$$\Pr[C_1 = c_1 \wedge C_2 = c_2 \mid M_1 = m \wedge M_2 = m'] = \frac{(|\mathcal{M}|-2)!}{|\mathcal{M}|!}$$

because exactly $(|\mathcal{M}|-2)!$ of the $|\mathcal{M}|!$ possible keys (which is picked uniformly at random) yield the two ciphertext. So let $\delta \stackrel{\text{def}}{=} \frac{(|\mathcal{M}|-2)!}{|\mathcal{M}|!}$. Plugging it into the above equation yields

$$\Pr[M_1 = m_1 \wedge M_2 = m_2 \mid C_1 = c_1 \wedge C_2 = c_2] = \frac{\delta \Pr[M_1 = m_1 \wedge M_2 = m_2]}{\sum_{i \neq j}\delta \Pr[M_1 = m_i \wedge M_2 = m_j]}$$

which is equal to $\Pr[M_1 = m_1 \wedge M_2 = m_2]$. □

11

(a) Note that whenever the all-zero key is chosen in the one-time pad, we obtain $\mathsf{Enc}(k, m) = 0^\ell \oplus m = m$. Is this a problem? In particular, suppose to modify the one-time pad by requiring that the key is sampled from the set $\mathcal{K}' := \{0,1\}^\ell \setminus \{0^\ell\}$. Is the resulting encryption scheme still perfectly secure? Prove your answer.

Solution: The modified scheme does not meet perfect secrecy, as now $|\mathcal{K}'| = 2^\ell - 1 < 2^\ell = |\mathcal{M}|$, and we know that $(\mathsf{Enc}, \mathsf{Dec})$ is perfectly secret if and only if $|\mathcal{K}| = |\mathcal{M}|$. The latter can be also seen directly by taking M to be uniform over $\{0,1\}^\ell$; for any possible $m \in \{0,1\}^\ell$ we get:

$$\Pr[M = m | C = m] = 0 \neq 2^{-\ell} = \Pr[M = m],$$

which contradicts perfect secrecy.

We conclude that we should not exclude the zero key 0^ℓ, and there is no problem at all if such a key would be chosen. The point is that the adversary has no way of knowing the key is 0^ℓ, so the fact that the ciphertext is equal to the plaintext in this case does not help the adversary.

or

(a) As for perfectly secure definition, we have that $|K| = |M|$. However, if we exclude 0^l from $|K|$ (eg. we define $K\prime := K\ 0^\lambda$), we'll have $|K\prime| < |M|$ (because $|K\prime| < |K|$ due to key deletion), so the schema with $K\prime$ and M will not be perfectly secure. Also, key 0^l is not an issue: the fact that the ciphertext is the same as the plaintext doesn't help the attacker (who doesn't know the plaintext).

12

(b) Let (Enc, Dec) be a perfectly secret encryption scheme. Refute the following statement: For all distributions M over the message space $\mathcal{M}$, for all $m, m' \in \mathcal{M}$, for all $c \in \mathcal{C}$, we have:

$$\Pr[M = m | C = c] = \Pr[M = m' | C = c].$$

(**Hint:** Let $\mathcal{M} = \{m_0, m_1\}$ and consider the distribution M such that $\Pr[M = m_0] = 3/4$ and $\Pr[M = m_1] = 1/4$.)

Solution: As hinted above, consider $\mathcal{M} = \{m_0, m_1\}$ with the distribution M such that $\Pr[M = m_0] = 3/4$ and $\Pr[M = m_1] = 1/4$. By definition of perfect secrecy, for all $c \in \mathcal{C}$, we get:

$$\Pr[M = m_0 | C = c] = \Pr[M = m_0] \neq \Pr[M = m_1] = \Pr[M = m_1 | C = c],$$

showing that the statement of the exercise does not hold.

2. (2.3 in book) Prove or refute: An encryption scheme with message space $\mathcal{M}$ is perfectly secret if and only if for every probability distribution over $\mathcal{M}$ and every $c_0, c_1 \in \mathcal{C}$ we have $\Pr[C = c_0] = \Pr[C = c_1]$.

Or

(b) Prove or refute: An encryption scheme with message space $\mathcal{M}$ is perfectly secret if and only if for all distributions M over $\mathcal{M}$, for all $m, m' \in \mathcal{M}$, and for every $c_0, c_1 \in \mathcal{C}$, we have:

$$\Pr[C = c_0] = \Pr[C = c_1].$$

(b) Refute: let's assume that it's perfect secrecy. From the hypotesis, $P[C = c_0] = P[C = c_1] = P[C = c]$ and M can be any distribution (even not uniformly distributed). By definition of perfect secrecy (from 3rd point of Shannon):

$$P[C = c | M = m_0] = P[C = c | M = m_1]$$

we can say:

$$P[C = c_0] = P[C = c] = P[C = c | M = m_0] = P[C = c | M = m_1] = P[C = c] = P[C = c_1]$$

so, by the fact that $P[C = c | M = m_0] = P[M = m_0 | C = c]$:

$$P[M = m_0] = P[M = m_0 | C = c] = P[M = m_1 | C = c] = P[M = m_1]$$

So, by saying that $P[C = c_0] = P[C = c_1]$ we need to have $P[M = m_0] = P[M = m_1]$ (in other words, we need that M is uniform if C is uniform)

or

Solution: Recall that one of the definitions of perfect secrecy states that an encryption scheme is perfectly secret if the ciphertext distribution does not depend on the message. In other words, every message induces the same ciphertext distribution. The condition stated in the exercise implies that every message induces the *uniform* ciphertext distribution. Clearly, if that is true, then the scheme is also perfectly secret, since every message induces the same ciphertext distribution (the uniform distribution). However, the condition is strictly stronger than perfect secrecy, since there are perfectly secret schemes that do not meet the condition. Since the question was if the condition is equivalent to perfect secrecy (i.e. "if and only if"), the statement is wrong.

We now give a counterexample of a perfectly secret scheme, where above condition does not hold. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be any perfectly secure encryption scheme (for example the One-Time Pad). We define $\Pi' = (\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}')$ as

$\mathsf{Enc}'_k(m)$:
  $c \leftarrow \Pi.\mathsf{Enc}_k(m)$
  pick b uniformly at random from $\{0,1\}^2$
  if $b = 00$
    return $c||0$
  else
    return $c||1$

Dec' simply truncates the last bit from the ciphertext and uses Dec to decrypt the ciphertext.

Clearly, Π' is correct if Π is. Furthermore, note that if the ciphertext distribution of Π does not depend on the message, then neither does the ciphertext distribution of Π', since it is the same distribution with a bit appended, and this bit is independent of the message. It follows that Π' is perfectly secret. However, the ciphertext distribution of Π' is clearly not uniform, since ciphertexts ending in 1 are three times as likely as ciphertexts ending in 0. This contradicts the condition given in the exercise. □

13

(c) Define an appropriate notion of a 2-time ε-secure MAC, and give a construction that meets your definition.

(c) A 2-secure ϵ-MAC is a message authentication code which is secure if an attacker cannot generate a new (fresh) (m, ϕ) with a probability $> \epsilon$ based on two pairs (m_0, ϕ_0), (m_1, ϕ_1). An example of a function that can be used to tag a message with a 2-time secure MAC is: $Tag(m) = am^2 + bm + c = \phi$. *Proof*: if we consider this equation system (with: m_0, m_1 two different messages, ϕ_0, ϕ_1 two different tags for messages m_0 and m_1 respectively):

$$\begin{cases} am_0^2 + bm_0 + c = \phi_0 \\ am_1^2 + bm_1 + c = \phi_1 \end{cases}$$

This system has $\infty^{3-2} = \infty$ solutions.

14____

Esercizio 8.1. Alice pubblica i seguenti dati: $N = pq = 221$ ed $e = 13$. Quindi utilizza lo schema naïve RSA per firmare il messaggio $m = 65$ ed invia la relativa firma $\sigma = 182$ al Bianconiglio. Verificare la firma.

C=m^e (mod221) → 65^13 mod 221 =39 N=p*q → 221=13*17

Verify= 182^17(mod 221)=182 then the signature it is not verified!

15____

3 Negligible and Noticeable Functions 15 Points

(a) Recall that a function $\nu : \mathbb{N} \to [0, 1]$ is negligible if $\nu(\lambda) \in O(1/p(\lambda))$ for all positive polynomials $p(\lambda) \in \text{poly}(\lambda)$. Show that the following alternative definition is equivalent: A function $\nu : \mathbb{N} \to [0, 1]$ is negligible if for all $c \in \mathbb{N}$, there exists $\lambda_0 \in \mathbb{N}$ such that for all $\lambda \geq \lambda_0$ we have $\nu(\lambda) < \lambda^{-c}$.

(b) Prove that $\nu(\lambda) = 2^{-\lambda}$ is negligible.

(c) A function $\mu : \mathbb{N} \to [0, 1]$ is noticeable if there exists $c, \lambda_0 \in \mathbb{N}$ sucht that, for all $\lambda \geq \lambda_0$, we have $\mu(\lambda) \geq \lambda^{-c}$.

Explain the difference between a noticeable function and a non-negligible function. Show that the following function is both non-negligible and non-noticeable:

$$f(\lambda) = \begin{cases} 2^{-\lambda} & \text{if } \lambda \text{ is even} \\ \lambda^{-3} & \text{if } \lambda \text{ is odd.} \end{cases}$$

1.2 Difference between Noticeable and Non-Negligible

Note that a non-negligible function is not necessarily a noticeable function. A non-negligible function $\mu(n)$ would satisfy the following:
$\exists c \in N$ such that $\forall n_0 \in N$, $\exists n \geq n_0$ such that $\mu(n) \geq n^{-c}$.

Note the key difference from a noticeable function - a non-negligible function only needs to have one $n \geq n_0$ for which $\mu(n) \geq n^{-c}$, whereas a noticeable function must satisfy this for *any* $n \geq n_0$.

For example, if we take any noticeable function and any negligible function and interleave them, then the resulting function will be non-negligible and non-noticeable. A concrete example is:

$$\mu(n) = \begin{cases} 2^{-n} & : x \texttt{ is even} \\ n^{-3} & : x \texttt{ is odd} \end{cases}$$

The function cannot be negligible, because for any odd integer, the function is only polynomially small, but it is not noticeable either, because for any even integer, it is exponentially small.

Or

(a) [$\nu(\lambda) \in O(\frac{1}{p(x)}) \Rightarrow \nu(\lambda) < \lambda^{-c}$] By definition of $O(\cdot)$:

$$\nu(\lambda) \in O(\frac{1}{p(x)}) \quad \rightarrow \quad \exists c_0 \text{ s.t. } \nu(\lambda) \leq \frac{1}{p(x)} c_0$$

Let's pick the poly $p(\lambda) = a_1\lambda^c + a_2\lambda^{c-1} + ... + a_n$ with $a_i > 0 \forall i \in \mathbb{N}$. We can write $\frac{1}{p(\lambda)}$ as:

$$\nu(\lambda) \leq (\lambda^-c + \lambda^{-c+1} + ... + \lambda^{-1})$$

For the purpuose of this demonstration, we can ignore all *poly* but λ^{-c}, so:

$$\nu(\lambda) \leq \lambda^{-c}$$

[$\nu(\lambda) < \lambda^{-c} \Rightarrow \nu(\lambda) \in O(\frac{1}{p(x)})$] We know that

$$\lim_{\lambda \to \inf} \frac{\nu(\lambda)}{\lambda^{-c}} = k$$

But this is the definition of $O(\cdot)$, so we can say that $\nu(\lambda) \in O(\lambda^{-c})$. As this statement is valid for all c, $\nu(\lambda) \in O(\frac{1}{p(\lambda)}), \forall p(\lambda) > 0$

(b) As for *negligible* definition, $\nu(\lambda) < \lambda^{-c}$, so it's valid to write:

$$\frac{1}{2^\lambda} < \frac{1}{\lambda^c}$$

However the left part of this expression is an inverse exponential in λ, and $\lim_{\lambda \to \inf} 2^{-\lambda}$ will go to zero faster than $\lim_{\lambda \to \inf} \lambda^{-c}, \forall c$, so $\nu(\lambda) < \lambda^{-c}, \forall c \in N$

(c) (i) The difference is that, in a *noticeable* function, we require only that *exists* a single c such that $\mu(\lambda) \geq \lambda^{-c}$, where in *non-negligible* you require that $\nu(\lambda) \geq \lambda^{-c}$ is valid for all $c \in \mathbb{N}$

(ii) The function is $f(\lambda)$ is noticeable when λ is even, and negligible when λ is odd.

16

(c) Assume we want to use the one-time pad as a deterministic MAC in the following natural way: Define $\mathsf{Mac}(k, m) = m \oplus k = \phi$, where $\mathcal{K} = \mathcal{M} = \Phi = \{0,1\}^n$. Show that this MAC is not one-time statistically secure.

Solution: Given a valid pair (m, ϕ) such that $\phi = m \oplus k$ consider the adversary that outputs (m^*, ϕ^*) where for any $\delta \in \{0,1\}^n \setminus \{0^n\}$ we define $m^* = m \oplus \delta$ and $\phi^* = \phi \oplus \delta$. Clearly, $m^* \neq m$ and moreover $\phi^* = (m \oplus \delta) \oplus k = m^* \oplus k$ is a valid pair, contradicting one-time statistical security of the MAC.

17

(a) A family of functions $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ is called *universal* if for all distinct inputs $x, x' \in \mathcal{X}$ we have:

$$\Pr\left[h_s(x) = h_s(x') : s \leftarrow\$ \mathcal{S}\right] \leq |\mathcal{Y}|^{-1}.$$

Show that any family $\mathcal{H}$ that is pairwise independent (as defined in class) is also universal.

Solution: This exercise had a small typo, in that the definition of universality and pairwise independence need to be considered with equalities. The correct definition is as follows: The family $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ is called *universal* if for all distinct inputs $x, x' \in \mathcal{X}$ we have:

$$\Pr_{s \leftarrow\$ \mathcal{S}}\left[h_s(x) = h_s(x')\right] = \frac{1}{|\mathcal{Y}|},$$

On the other hand, as defined in class, the family $\mathcal{H}$ is called *pairwise independent* if for all distinct $x, x' \in \mathcal{X}$ and all y, y' we have:

$$\Pr_{s \leftarrow\$ \mathcal{S}}\left[h_s(x) = y \wedge h_s(x') = y'\right] = \frac{1}{|\mathcal{Y}|^2}.$$

A straightforward calculation then shows that:

$$\Pr_{s \leftarrow\$ \mathcal{S}}\left[h_s(x) = h_s(x')\right] = \sum_{y \in \mathcal{Y}} \Pr_{s \leftarrow\$ \mathcal{S}}\left[h_s(x) = y \wedge h_s(x') = y\right] = |\mathcal{Y}| \cdot \frac{1}{|\mathcal{Y}|^2} = \frac{1}{|\mathcal{Y}|},$$

18

(b) Let $\mathbb{F}$ be a finite field, and consider the following family of hash functions $\mathcal{H}$ with $\mathcal{Y} = \mathbb{F}$ and $\mathcal{X} = \mathcal{S} = \mathbb{F}^t$ for some value $t \in \mathbb{N}$. For a secret key $s = (s_1, \ldots, s_t) \in \mathbb{F}^t$, and input $x = (x_1, \ldots, x_t) \in \mathbb{F}^t$, define $h_s(x) := \sum_{i=1}^t s_i \cdot x_i$ where all operations take place in $\mathbb{F}$. Show that $\mathcal{H}$ is universal.

Solution: Let $x = (x_1, \ldots, x_t)$ and $x' = (x'_1, \ldots, x'_t)$, and define $\delta_i = x'_i - x_i$ for all $i \in [t]$. If x and x' are distinct, at least one of the δ_i's is different from zero, say this happens for $\delta_1 = x_1 - x'_1$. We need to compute the probability that $h_s(x) = h_s(x')$ over the choice of the key $s \in \mathbb{F}^t$. But notice that,

$$h_s(x) = h_s(x') \Leftrightarrow \sum_{i=1}^t s_i \cdot x_i = \sum_{i=1}^t s_i \cdot x'_i \Leftrightarrow s_1 \cdot \delta_1 = -\sum_{i=1}^t s_i \cdot \delta_i \Leftrightarrow s_1 = \frac{-\sum_{i=1}^t s_i \cdot \delta_i}{\delta_1}.$$

The above quantity is well defined, since $\delta_1 \neq 0$; we conclude

$$\Pr_{s \leftarrow\$ \mathbb{F}^t}\left[h_s(x) = h_s(x')\right] = \frac{1}{|\mathbb{F}|}.$$

19

(a) Let $f : \{0,1\}^n \to \{0,1\}^n$ be a candidate OWF. Show that there always exists an inefficient attacker $\mathcal{A}_1$ inverting the function with probability one, and an efficient attacker $\mathcal{A}_2$ inverting the function with probability 2^{-n}.

Solution: Consider the attacker $\mathcal{A}_1$ that given $y = f(x)$ for random $x \leftarrow\$ \{0,1\}^n$ tries all possible $x' \in \{0,1\}^n$, computes $f(x') = y'$, and outputs x' such that $y' = y$. Clearly, x' must exist as $y = f(x)$ for some $x \in \{0,1\}^n$. Hence, $\mathcal{A}_1$ succeeds with probability 1 but runs in exponential time (in n).

Similarly, consider the attacker $\mathcal{A}_2$ that simply outputs a random $x' \in \{0,1\}^n$; the probability that $x' = x$ is 2^{-n} that is also a bound on the adversary's advantage.

20

Analyze the following candidate OWFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $f(x, y) = x + y$, where $|x| = |y|$ and x, y are interpreted as natural numbers.
Solution: This candidate *does not yield* a OWF. Given $f(x, y) = z$, for random x, y, simply return $(z, 0)$. Clearly, $f(z, 0) = z + 0 = x + y = f(x, y)$.

21

Analyze the following candidate OWFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(ii) $g(x_1, x_2) = (f(x_1), x_2)$, where f is a OWF and $|x_1| = |x_2|$.

Solution: This candidate *does yield* a OWF. Let $|x_1| = |x_2| = n(\lambda)$. By contradiction, assume that there exists an adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many values of $\lambda \in \mathbb{N}$

$$\Pr\left[g(x_1', x_2') = \hat{y} : \begin{array}{c} x_1, x_2 \leftarrow\!\!\$\ \{0,1\}^{n(\lambda)}; \hat{y} = g(x_1, x_2) \\ (x_1', x_2') \leftarrow \mathcal{A}(1^\lambda, y) \end{array}\right] \geq 1/p(\lambda).$$

Consider the following attacker $\mathcal{A}'$ for inverting f: Upon input $y = f(x)$ for random $x \in \{0,1\}^n$, pick random $x_2 \in \{0,1\}^n$, run $(x_1', x_2') \leftarrow \mathcal{A}(1^\lambda, (y, x_2))$, and return x_1'. Notice that the input (y, x_2) that $\mathcal{A}'$ passes to $\mathcal{A}$ has the distribution $\mathcal{A}$ expects, since $(y, x_2) = (f(x), x_2)$ for random $x, x_2 \in \{0,1\}^n$. We conclude that $\mathcal{A}'$ inverts f with the same probability that $\mathcal{A}$ inverts g (i.e., with non-negligible probability). This concludes the proof.

or

$g(x_1, x_2) = (f(x_1), x_2)$ is a OWF: let's consider a game where A_f is an attacker that tries to break OWF $f(\cdot)$. We know (by OWF definition) that A_f has negligible probability to break $f(\cdot)$. So, we suppose that exists a PPT attacker A_g that breaks $g(\cdot)$ with greater probability than negligible. If such attacker exists, A_f can use that attacker to reverse $f(\cdot)$ (by making an $y\prime$ based on $f(\cdot)$) and acquire its probability:

C | A_f | A_g

$x \leftarrow \$\{0,1\}^\lambda$
$y = f(x_1)$
-----------y----------->
$x_2 \leftarrow \$\{0,1\}^\lambda$
------(y, x_2)------->
<-----(x_1, x_2)-------
<----- x_1 -----------

So, if A_g is able to break the OWF $g(\cdot)$ with probability non-negligible, A_f is also able to break the OWF $f(\cdot)$ with non-negligible probability due the fact that we said that A_f cannot break OWF, A_g cannot exists.

22

Analyze the following candidate OWFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(iii) $g(x) = (f(x), f(f(x)))$, where $f : \{0,1\}^n \to \{0,1\}^n$ is a OWF.
Solution: This candidate *does yield* a OWF. By contradiction, assume that there exists an adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many values of $\lambda \in \mathbb{N}$

$$\Pr\left[g(x') = \hat{y} : x \leftarrow\!\!\$\ \{0,1\}^n; \hat{y} = g(x); x' \leftarrow \mathcal{A}(1^\lambda, y)\right] \geq 1/p(\lambda).$$

Consider the following attacker $\mathcal{A}'$ for inverting f: Upon input $y = f(x)$ for random $x \in \{0,1\}^n$, compute $z = f(y)$, run $x' \leftarrow \mathcal{A}(1^\lambda, (y, z))$, and return x'. Notice that the input (y, z) that $\mathcal{A}'$ passes to $\mathcal{A}$ has the distribution $\mathcal{A}$ expects, since $(y, z) = (f(x), f(f(x)))$ for random $x \in \{0,1\}^n$. We conclude that $\mathcal{A}'$ inverts f with the same probability that $\mathcal{A}$ inverts g (i.e., with non-negligible probability). This concludes the proof.

Or

(iii) $g(x) = (f(x), f(f(x))$ is a OWF: let's consider a game where A_f is an adversary that tries to break OWF $f(\cdot)$ (as for *(ii)*). A_g is an hypotetic PPT adversary that can break $g(\cdot)$ with probability greater than negligible. We show that A_g cannot exists.

C	A_f	A_g
$x \leftarrow\$ \{0,1\}^\lambda$		
$y = f(x)$		
$\xrightarrow{\quad y \quad}$		
	$y\prime = f(y)$	
	$\xrightarrow{\quad (y, y\prime) \quad}$	
	$\xleftarrow{\quad x \quad}$	
$\xleftarrow{\quad x \quad}$		

So, if A_g is able to break the OWF $g(\cdot)$ with probability greater than negligible, A_f is also able to break the OWF $f(\cdot)$ with non-negligible probability due the fact that A_f builds $y\prime$ from $y = f(x)$ and asks A_g to break it. Because we stated that $f(\cdot)$ is a OWF and A_f is PPT adversary, A_g cannot exists.

23

Analyze the following candidate OWFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(iv) $g(x) = f(x||0)$, where $f : \{0,1\}^n \to \{0,1\}^n$ is a OWF.
Solution: This candidate *does yield* a OWF. By contradiction, assume that there exists an adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many values of $\lambda \in \mathbb{N}$

$$\Pr\left[g(x') = \hat{y} : \ x \leftarrow\$ \{0,1\}^{n-1}; \hat{y} = g(x); x' \leftarrow \mathcal{A}(1^\lambda, y)\right] \geq 1/p(\lambda).$$

Consider the following attacker $\mathcal{A}'$ for inverting f: Upon input $y = f(x)$ for random $x \in \{0,1\}^n$, run $x' \leftarrow \mathcal{A}(1^\lambda, y)$, and return $x'||0$. Notice that as long as the first bit of x is zero, which happens with probability $1/2$, the input y that $\mathcal{A}'$ passes to $\mathcal{A}$ has the distribution $\mathcal{A}$ expects, since $y = f(x''||0))$ for random $x'' \in \{0,1\}^{n-1}$. We conclude that $\mathcal{A}'$ inverts f with probability at least $p(\lambda)/2$ which is non-negligible. This concludes the proof.

or

$g(x) = f(x \parallel 0)$ is a OWF. By contradiction, let's consider a game such that C is a challenger for a OWF $f(x)$, A_f is a PPT adversary which tries to break a OWF, and A_g is an hypotetic adversary which can breaks $g(\cdot)$ with probability non-negligible $p(\lambda)$. We show that such A_g doesn't exists.

C	A_f	A_g
$x \leftarrow\$ \{0,1\}^\lambda$		
$y = f(x \parallel 0)$		
$\xrightarrow{\quad y \quad}$		
	$\xrightarrow{\quad y \quad}$	
	$\xleftarrow{\quad x \quad}$	
	$x\prime = x \parallel 0$	
$\xleftarrow{\quad x' \quad}$		

If A_g can breaks $g(\cdot)$ and recover x, also A_f can recover the argument of A_f and breaks the OWF with probability $\frac{1}{2}p(\lambda)$. But this is a contradiction, so A_g doesn't exists.

(a) Show that no PRG can be secure against computationally unbounded adversaries. In particular, let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a length-doubling PRG. Show that there is an exponential-time distinguisher breaking the PRG with probability almost one.

Solution: Consider the following exponential time distinguisher $\mathcal{D}$. Upon input $y \in \{0,1\}^{2\lambda}$ the goal of $\mathcal{D}$ is to distinguish whether $y = G(s)$ for a random $s \leftarrow\!\!\$ \{0,1\}^\lambda$, or $y \leftarrow\!\!\$ \{0,1\}^{2\lambda}$. The distinguisher simply outputs 1 if and only if there exists some $s \in \{0,1\}^\lambda$ such that $y = G(s)$; note that this computation is performed by computing $G(s)$ for all possible seeds $s \in \{0,1\}^\lambda$.

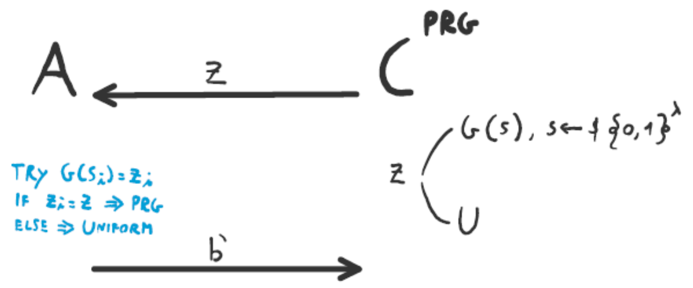
Now, if y really comes from the range of the PRG, $\mathcal{D}$ outputs 1 with probability 1. On the other hand, assume that $y \leftarrow\!\!\$ \{0,1\}^{2\lambda}$. Then with probability $2^{-\lambda}$ the value y is actually outside the range of the PRG; this is because G receives an input of length λ and thus its range consists of at most 2^λ values, whereas y is sampled from a set of $2^{2\lambda}$ possible values, and $2^\lambda/2^{2\lambda} = 2^{-\lambda}$. We conclude that,

$$\left|\Pr\left[\mathcal{D}(y) = 1 :\ s \leftarrow\!\!\$ \{0,1\}^\lambda; y = G(s)\right] - \Pr\left[\mathcal{D}(y) = 1 :\ y \leftarrow\!\!\$ \{0,1\}^{2\lambda}\right]\right| \geq 1 - 2^{-\lambda},$$

and thus $\mathcal{D}$ distinguishes with overwhelming probability.

2) $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$

$\exists$ unbonded A that can break G



There is a BAD EVENT, when A finds a valid s_i s.t. $G(S_i) = z$ where z is chosen from U, $\Pr[\text{BAD}] = \frac{2^\lambda}{2^{2\lambda}} = \frac{1}{2^\lambda} < \text{NEGL}(\lambda)$

25

Analyze the following candidate PRGs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $G'(s) = G(s_1)||\ldots||G(s_n)$, where $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and $s = s_1||\ldots||s_n \in \{0,1\}^{\lambda n}$.

Solution: This candidate *does yield* a PRG. Fix some polynomial $n := n(\lambda)$. Define the following hybrid distribution:

$$\mathbf{H}_i(\lambda) = (\underbrace{U_{\lambda+\ell}, \ldots, U_{\lambda+\ell}}_{i \text{ times}}, \underbrace{G(U_\lambda), \ldots, G(U_\lambda)}_{n-i \text{ times}}).$$

Notice that $\mathbf{H}_0(\lambda) \equiv G'(U_{\lambda n})$, whereas $\mathbf{H}_n(\lambda) \equiv U_{(\lambda+\ell)n}$. We prove security by a standard hybrid argument. Assume that there exists a distinguisher $\mathcal{D}$, an index $i \in [n]$, and a polynomial $p(\cdot)$ such that, for infinitely many values of $\lambda \in \mathbb{N}$:

$$|\Pr[\mathcal{D}(\mathbf{H}_i(\lambda)) = 1] - \Pr[\mathcal{D}(\mathbf{H}_{i+1}(\lambda)) = 1]| \geq 1/p(\lambda).$$

We construct a distinguisher $\mathcal{D}'$ breaking security of the underlying PRG G. The distinguisher $\mathcal{D}'$, upon input a target value y that is either sampled from $G(U_\lambda)$ or from $U_{\lambda+\ell}$, proceeds as follows:

1. Sample $y_1, \ldots, y_i \leftarrow\$ \{0,1\}^{\lambda+\ell}$.
2. Sample $s_{i+2}, \ldots, s_n \leftarrow\$ \{0,1\}^\lambda$ and let $y_j = G(s_j)$ for all $i+2 \leq j \leq n$.
3. Return the same as $\mathcal{D}(y_1, \ldots, y_i, y, y_{i+2}, \ldots, y_n)$.

For the analysis, it suffices to note that in the above reduction the input to $\mathcal{D}$ either comes from the distribution $\mathbf{H}_i(\lambda)) = 1$ (in case $y = G(s)$ comes from the PRG), or from the distribution $\mathbf{H}_{i+1}(\lambda)) = 1$ (in case y is random). We conclude that $G'(U_{\lambda n}) \equiv \mathbf{H}_0(\lambda) \approx_c \ldots \approx_c \mathbf{H}_n(\lambda) \equiv U_{(\lambda+\ell)n}$, as desired.

26

Analyze the following candidate PRGs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(ii) $G'(s) = G(s||0^{|s|})$, where $G : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda+\ell}$ is a PRG.

(**Hint:** Consider the contrived PRG $G(s)$ that ignores the first half of its input, and returns $\hat{G}(s_{\lambda+1}, \ldots, s_{2\lambda})$ where $\hat{G}$ is itself a PRG stretching λ bits into $2\lambda + \ell$ bits. Argue that G is a PRG, but G' as defined in the exercise is not.)

Solution: As hinted above, let $\hat{G}$ be a PRG stretching λ bits into $2\lambda + \ell$ bits, and consider the PRG $G(s)$ that ignores the first half of its input, and returns $\hat{G}(s_{\lambda+1}, \ldots, s_{2\lambda})$. We argue that G is still a PRG, but G' is not.

The second part of the statement is easy, indeed: $G'(s) = G(s||0^{|s|}) = \hat{G}(0^\lambda)$ and it is trivial to distinguish the outout of G' from random. It remains to prove that G is a PRG, but this is also easy to show. In fact, for any efficient $\mathcal{D}$, we have

$$\left|\Pr\left[\mathcal{D}(y) = 1 : s \leftarrow\$ \{0,1\}^{2\lambda}; y = G(s)\right] - \Pr\left[\mathcal{D}(y) = 1 : y \leftarrow\$ \{0,1\}^{2\lambda+\ell}\right]\right|$$
$$= \left|\Pr\left[\mathcal{D}(y) = 1 : s \leftarrow\$ \{0,1\}^{\lambda}; y = \hat{G}(s)\right] - \Pr\left[\mathcal{D}(y) = 1 : y \leftarrow\$ \{0,1\}^{2\lambda+\ell}\right]\right|$$

and the latter quantity is negligible by the fact that $\hat{G}$ is a PRG.

27

Analyze the following candidate PRFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $F_k(x) = G'(k) \oplus x$, where $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and G' denotes the output of G truncated to λ bits.

Solution: This candidate *does not yield* a PRF. Let $\mathcal{D}$ be a distinguisher that is given oracle access to either $F_k(\cdot)$ for a random $k \in \{0,1\}^\lambda$, or to a truly random function $R : \{0,1\}^\lambda \to \{0,1\}^\lambda$. The distinguisher queries its oracle upon input any two distinct values $x, x' \in \{0,1\}^\lambda$, receiving back outputs y, y'. Hence, $\mathcal{D}$ returns 1 if and only if $x \oplus x' = y \oplus y'$.

Clearly, if $\mathcal{D}$'s oracle is $F_k(\cdot)$ (for a uniform $k \leftarrow\$ \{0,1\}^\lambda$):

$$y \oplus y' = (G'(k) \oplus x) \oplus (G'(k) \oplus x') = x \oplus x',$$

and thus $\mathcal{D}$ always outputs 1 in such a case. On the other hand, if $\mathcal{D}$'s oracle is R, the probability (over the choice of the function) that $x \oplus x' = R(x) \oplus R(x')$ is equal to $2^{-\lambda}$. Hence, $\mathcal{D}$ has distinguishing advantage nearly equal to 1.

28

Analyze the following candidate PRFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(ii) $F_k(x) := F_x(k)$, where $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\ell$ is a PRF.

Solution: This candidate *does not yield* a PRF. Consider the contrived PRF F that upon input the all-zero key 0^λ always returns 0^ℓ independently of the input, i.e., $F_{0^\lambda}(x) = 0^\ell$ for all $x \in \{0,1\}^\lambda$. On the one hand, it is easy to prove that the above function still defines a PRF (as the "bad key" 0^λ is chosen only with negligible probability). On the other hand, it is easy to distinguish $F_x(k)$ (for a randomly chosen "key" $k \leftarrow\$ \{0,1\}^\lambda$) from a truly random function by simply querying input $x = 0^\lambda$ to the target oracle: If the oracle implements the PRF, we will obtain 0^ℓ with probability 1, whereas if the oracle implements a truly random function $R : \{0,1\}^\lambda \to \{0,1\}^\ell$ the value 0^ℓ will be obtained only with probability $2^{-\ell}$.

29

Analyze the following candidate PRFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(iii) $F'_k(x) = F_k(x||0)||F_k(x||1)$, where $x \in \{0,1\}^{n-1}$.

Solution: This candidate *does yield* a PRF. By contradiction, assume there exists an efficient distinguisher $\mathcal{D}'$ that can distinguish (oracle access to) $F'_k(\cdot)$ with a random key from a truly random function $R' : \{0,1\}^{n-1} \to \{0,1\}^n$ with non-negligible probability.

Consider the following distinguisher $\mathcal{D}$, whose goal is to distinguish (oracle access to) $F_k(\cdot)$ with a random key from a truly random function $R : \{0,1\}^n \to \{0,1\}^n$: (1) Upon input an oracle query $x' \in \{0,1\}^{n-1}$ from $\mathcal{D}$, define $x^0 := x'||0$ and $x^1 := x'||1$, query x^0, x^1 to the target oracle receiving back values y^0, y^1, and return $y^0||y^1$ to $\mathcal{D}'$; (2) Output the same as $\mathcal{D}'$. Clearly, $\mathcal{D}$ is roughly as efficient as $\mathcal{D}'$ (in fact, if $\mathcal{D}'$ makes q' oracle queries $\mathcal{D}$ needs to make $q = 2q'$ oracle queries), moreover it perfectly simulates the view of $\mathcal{D}'$ and thus it retains the same (non-negligible) advantage. This finishes the proof.

30

Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be a SKE scheme with key space $\mathcal{K}$. Consider the following variant of computational one-time security: For all PPT adversaries $\mathcal{A}$, there exists a negligible function $\varepsilon : \mathbb{N} \to \mathbb{R}$ such that

$$\Pr\left[b = b' : \begin{array}{c} k \leftarrow\$ \mathcal{K}; b \leftarrow\$ \{0,1\}; (m_0, m_1) \leftarrow \mathcal{A}(1^\lambda) \\ c = \mathsf{Enc}(k, m_b); b' \leftarrow \mathcal{A}(1^\lambda, c) \end{array}\right] \leq \frac{1}{2} + \varepsilon(\lambda), \tag{1}$$

with $|m_0| = |m_1|$, and where the probability is taken over the random choice of k, b, and over the randomness of the algorithm $\mathcal{A}$. Prove that this variant is equivalent to the notion we defined in class. (This means that the above formulation implies the one we gave in class, and viceversa.)

Solution: For $b \in \{0,1\}$, denote with $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b)$ the game we considered in class in the definition of one-time computational security for SKE:

Game $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b)$:

1. $k \leftarrow\$ \mathcal{K}$
2. $(m_0, m_1) \leftarrow \mathcal{A}(1^\lambda)$
3. $c = \mathsf{Enc}(k, m_b)$
4. $b' \leftarrow \mathcal{A}(1^\lambda, c)$

Our original formulation was that Π is one-time computationally secure if for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\varepsilon : \mathbb{N} \to \mathbb{R}$ such that

$$\left|\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right| \le \varepsilon(\lambda). \tag{2}$$

Let $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda)$ be identical to $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b)$ except that the bit b is not fixed anymore, but it is instead chosen uniformely at random at the beginning of the game, and furthermore the output of $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda)$ is defined to be 1 if and only if $b' = b$. Clearly, Eq. (1) is equivalent to

$$\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \le \frac{1}{2} + \varepsilon(\lambda). \tag{3}$$

We first prove that Eq. (2) implies Eq. (3). A simple calculation shows:

$$\begin{aligned}
\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] &= \Pr_{b \leftarrow\$ \{0,1\}}\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b) = b\right] \\
&= \frac{1}{2}\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 0\right] + \frac{1}{2}\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] \\
&= \frac{1}{2}\left(1 - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) + \frac{1}{2}\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] \\
&= \frac{1}{2} + \frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) \qquad (4) \\
&\le \frac{1}{2} + \frac{1}{2}\left|\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right| \\
&\le \frac{1}{2} + \frac{1}{2}\varepsilon(\lambda),
\end{aligned}$$

where the last inequality follows by Eq. (2). Since $\varepsilon(\lambda)$ is negligible, so is $\varepsilon(\lambda)/2$, which proves Eq. (3).

Next, we prove that Eq. (3) implies Eq. (2). Combining Eq. (3) with Eq. (4), we can write:

$$\frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) = \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \le \varepsilon(\lambda). \tag{5}$$

For any adversary $\mathcal{A}$ consider the adversary $\tilde{\mathcal{A}}$ that outputs the complement of $\mathcal{A}$ (i.e., if $\mathcal{A}$ outputs b', we get that $\tilde{\mathcal{A}}$ outputs $\tilde{b} = 1 - b'$). A calculation similar to the one above shows:

$$\begin{aligned}
\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\tilde{\mathcal{A}}}(\lambda) = 1\right] &= 1 - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \\
&= 1 - \frac{1}{2} - \frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) \\
&= \frac{1}{2} + \frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right). \qquad (6)
\end{aligned}$$

Combining Eq. (3) with Eq. (6), we can write:

$$\frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right) = \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\tilde{\mathcal{A}}}(\lambda) = 1\right] - \frac{1}{2} \le \tilde{\varepsilon}(\lambda), \tag{7}$$

for some negligible function $\tilde{\varepsilon} : \mathbb{N} \to [0,1]$.

Putting together Eq. (5) and Eq. (7) we obtain:

$$\left|\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right| \le 2\varepsilon^*(\lambda),$$

where $\varepsilon^* := \max\{\varepsilon, \tilde{\varepsilon}\}$. Since both $\varepsilon(\lambda)$ and $\tilde{\varepsilon}(\lambda)$ are negligible, so is $\varepsilon^*(\lambda)$, which proves Eq. (2).

31

(b) Let $\ell, n > 0$. Consider the family of hash functions

$$\mathcal{H} = \{h_{\mathbf{A},\mathbf{b}} : \{0,1\}^n \to \{0,1\}^\ell\}_{\mathbf{A}\in\{0,1\}^{\ell\times n},\mathbf{b}\in\{0,1\}^\ell},$$

defined by $h_s(\mathbf{m}) = \mathbf{A}\cdot\mathbf{m}\oplus\mathbf{b}$, where $s = (\mathbf{A},\mathbf{b}) \in \{0,1\}^{\ell\times n}\times\{0,1\}^\ell$ and $\mathbf{m} \in \{0,1\}^n$, and where $\mathbf{A},\mathbf{b},\mathbf{m}$ are interpreted as matrices/vectors and all operations are performed modulo 2. Prove that $\mathcal{H}$ is pairwise independent.

(c) Let $\ell, n > 0$. Consider the family of hash functions $\mathcal{H}$ that is identical to the one in the above exercise, except that the matrix $\mathbf{A}$ is now sampled from the set $\mathbb{T}^{\ell\times n}$ of $\ell\times n$ Toeplitz matrices, i.e. all matrices $\mathbf{A} = (a_{i,j})$ such that $a_{i,j} = a_{i-1,j-1}$ when $i, j > 1$ (this means that the values along any diagonal are all equal).

Prove that $\mathcal{H}$ is still pairwise independent. What is the advantage w.r.t. the previous construction?

32

Let $\mathcal{P} = \{P_k : \{0,1\}^{2n} \to \{0,1\}^{2n}\}_{k\in\{0,1\}^\lambda}$ be a family of *strong* PRPs. Show that the following construction of an SKE scheme (Enc, Dec) satisfies CCA security but it is not a secure authenticated encryption scheme.

Encryption: Upon input $m \in \{0,1\}^n$, sample $r \leftarrow\$ \{0,1\}^n$ and return $c = \mathsf{Enc}(k,m) = P_k(m||r)$.

Decryption: Upon input $c \in \{0,1\}^{2n}$, let $m||r = P^{-1}(c)$ and output m.

(CCA) The construction is CCA-secure. Let's suppose, by contradiction, that this scheme is not CCA-secure. Consider this game:

- A_{prp} is the attacker that simulates an oracle for A
- A is an hypotetic attacker that can breaks this scheme
- $O(Enc, Dec)$ is an Oracle that provides permutations or random outputs: we define, for this game, Enc_g either $\Pi^{-1}(\cdot)$ or $P(\cdot)$ (PRP) and Dec_g either $\Pi^{-1}(\cdot)$ or $P(\cdot)$ (with Π as the random permutation)

O		A_{prp}		A
			$\xleftarrow{\quad m_i \quad}$	
		$r_i \leftarrow \$\{0,1\}^n$		
	$\xleftarrow{\quad m_i \Vert r_i \quad}$			
	$\xrightarrow{\quad c_i \quad}$		$\xrightarrow{\quad c_i \quad}$	
	$\xleftarrow{\quad c_i \quad}$		$\xleftarrow{\quad c_i \quad}$	
	$\xrightarrow{\quad m_i \Vert r_i \quad}$			
			$\xrightarrow{\quad m_i \quad}$	
		$m_0, m_1 \in M$		
			$\xleftarrow{\quad m_0, m_1 \quad}$	
		$b \leftarrow \$\{0,1\}$		
		$r_b \leftarrow \$\{0,1\}^n$		
	$\xleftarrow{\quad m_b \Vert r_b \quad}$			
	$\xrightarrow{\quad c_b \quad}$			
			$\xrightarrow{\quad c_b \quad}$	
...				
			$\xleftarrow{\quad m_i \quad}$	
		$r_i \leftarrow \$\{0,1\}^n$		
	$\xleftarrow{\quad m_i \Vert r_i \quad}$			
	$\xrightarrow{\quad c_i \quad}$		$\xrightarrow{\quad c_i \quad}$	
	$\xleftarrow{\quad c_i \quad}$		$\xleftarrow{\quad c_i \quad}$	
	$\xrightarrow{\quad m_i \Vert r_i \quad}$			
			$\xrightarrow{\quad m_i \quad}$	
...				
			$\xleftarrow{\quad b\prime \quad}$	
		$b = b\prime$		
			$\xrightarrow{\quad \texttt{win/lose} \quad}$	

The game is played as follow:

- A_{prp} has access to the oracle for Enc_g or Dec_g: it can ask for encryption or decryption many times
- After that, A decides two fresh messages m_0 and m_1 and send both to A_{prp}
- A_{prp} choose (random) one of two messages (we call it m_b), and send it to the oracle to get c_b, which it relays back c_b to A
- At this time, A can still ask queries to A_{prp}, but it can't ask for decryption of c_b or ecryption of m_0, m_1
- Eventually A will reply $b\prime$ to A_{prp}

At the end, $P[b = b\prime] \leq \frac{1}{2} + \epsilon$ (in other words, A can only guess); otherwise A_{prp} can use A to break the PRP (distinguish PRP from random), and that is a contradiction.

(Enc is based on strong PRP: as, by definition, strong-PRPs are indistiguishable from a deterministic random function $R(\cdot)$, A cannot distinguish the ciphertext of two messages m_0 and m_1 with a probability greater than $\frac{1}{2} + \epsilon$)

(SAE) (*Secure-Authenticated-Encryption*) The construction is not *secure authenticated encryption*: due the fact that PRP has always an $m \Vert r$ (PRP input) for each c (PRP output), we can forge a complete new c which will be decrypted in a new $m \Vert r$. This behavior is allowed because there is no way to check if the message m is a valid m (in other words, the scheme is not strong unforgeable).

33

(b) Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^\lambda}$ be a PRF family. Analyze the following construction of a MAC with key space $\mathcal{K} = \{0,1\}^\lambda$ and message space $\mathcal{M} = \{0,1\}^{2n}$: $\mathsf{Tag}(k, m_1||m2) = F_k(m_1)||F_k(F_k(m_2))$, with $m_1, m_2 \in \{0,1\}^n$.

(b) The schema is not secure. In fact, if an attacker has access to $Tag_k(m_1 \parallel m_2) = F_k(m_1) \parallel F_k(F_k(m_2))$ it can compute:

$$\phi = F_k(m_1) \parallel F_k(F_k(m_2)) \qquad \text{for message } m_1 \parallel m_2$$

$$\phi\prime = F_k(m_3) \parallel F_k(F_k(m_4)) \qquad \text{for message } m_3 \parallel m_4$$

Now the attacker can build the message $m_1 \parallel m_4$ and calculate its tag by taking $F_k(m_1)$ from ϕ (first half), and $F_k(F_k(m_4))$ from $\phi\prime$ (second half) and send the valid pair $(m_1 \parallel m_4, F_k(m_1) \parallel F_k(F_k(m_4))$ (forgery).

34

(c) Recall that in CBC-MAC the tag of a message $m = (m_1, \ldots, m_t) \in (\{0,1\}^n)^t$ is the value $\phi_t \in \{0,1\}^n$ computed using the following recursive equations: $\forall i \in [t], \phi_i = F_k(m_i \oplus \phi_{i-1})$ with $\phi_0 = 0^n$ and where F_k is sampled from a PRF family. Establish whether the following modifications of CBC-MAC are secure or not.

(i) Using CBC-MAC directly for authenticating variable-length messages.

(ii) A variant of CBC-MAC where, each time a tag is computed, a different value ϕ_0 is sampled uniformly at random from $\{0,1\}^n$ and output together with ϕ_t.

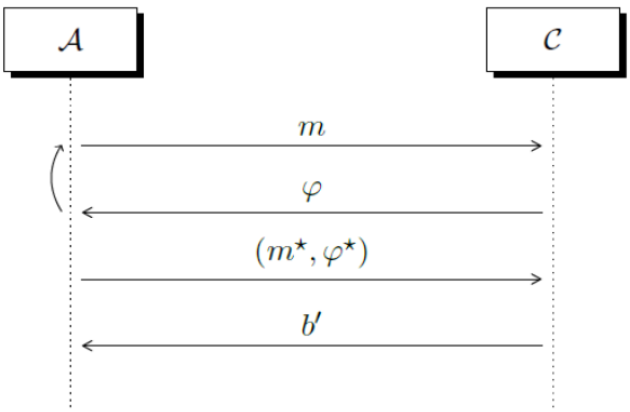
(iii) A variant of CBC-MAC where the output consists of all values $\phi_0, \phi_1, \ldots, \phi_t$.

(i)

This is not secure.

Let consider the following construction to show it: Obtain a message m0 ←\$ $\{0,1\}^n$ of 1 block and his Tag φ0 = Fk(m0).

We can construct a message m* = (m0, m0 ⊕ φ0) and a valid Tag φ * = φ0.



In this way when the challenger will verify the message he will compute φ *’ = Fk(m0 ⊕ φ0 ⊕ φ0) = Fk(m0) = φ0. So φ* = φ*’

(ii)
This is not secure.
Let consider the following construction to demonstrate it:
Obtain 2 messages m0, m1 with m0 different from m1 and their tags $\varphi 0 = (r0; Fk(m0 \oplus r0))$ and $\varphi 1 = (r1; Fk(m1 \oplus r1))$.
We can now forge a new message $m^* = (m0 \oplus m1)$ and a valid Tag $\varphi^* = (r1 \oplus m0, \varphi 1)$.
When the challenger will verify φ^* the will compute $\varphi^{*\prime} = Fk(m1 \oplus m0 \oplus r1 \oplus m0) = Fk(m0 \oplus r1) = \varphi 1$ then $\varphi^{*\prime} = \varphi^*$.

(iii)
This is not secure.
Let consider the following construction:
Obtain 2 messages $m0 = (m0,1, m0,2)$ and $m1 = (m1,1, m1,2)$, of at least 2 blocks, and their tags $\varphi 0 = (\varphi 0,1 , \varphi 0,2)$ and $\varphi 1 = (\varphi 1,1, \varphi 1,2)$.
Now we can construct $m^* = (m1,1, m2,1 \oplus \varphi 1,1 \oplus \varphi 2,1)$ and a valid Tag $\varphi^* = (\varphi 1,1; \varphi 2,2)$.
When the challenger will verify φ^* he will compute $\varphi^{*\prime} = (Fk(m1,1), Fk(m2,1 \oplus \varphi 1,1 \oplus \varphi 2,1 \oplus \varphi 1,1)) = \varphi 2,2$.
So $\varphi^* = \varphi^{*\prime}$

35____

(a) Assume a generalization of MACs where a MAC Π consists of a pair of algorithms $(\mathsf{Tag}, \mathsf{Vrfy})$, such that Tag is as defined in class (except that it could be randomized), whereas Vrfy is a deterministic algorithm that takes as input a candidate pair (m, ϕ) and returns a decision bit $d \in \{0, 1\}$ (indicating whether ϕ is a valid tag of m). Consider a variant of the game defining UF-CMA security of a MAC $\Pi = (\mathsf{Tag}, \mathsf{Vrfy})$, with key space $\mathcal{K} = \{0, 1\}^\lambda$, where the adversary is additionally granted access to a verification oracle $\mathsf{Vrfy}(k, \cdot, \cdot)$.

(i) Make the above definition precise, using the formalism we used in class. Call the new notion "unforgeability under chosen-message and verification attacks" (UF-CMVA).

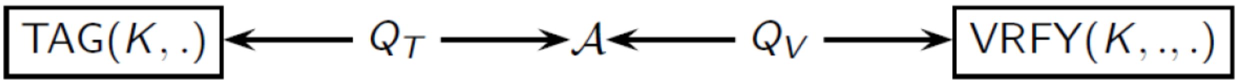
(ii) Show that whenever a MAC has unique tags (i.e., for every key k there is only one valid tag ϕ for each message m) then UF-CMA implies UF-CMVA.

(iii) Show that if tags are not unique there exists a MAC that satisfies UF-CMA but not UF-CMVA.

(**Hint:** Given an arbitrary MAC $\Pi = (\mathsf{Tag}, \mathsf{Vrfy})$ satisfying UF-CMA construct a contrived MAC $\Pi' = (\mathsf{Tag}', \mathsf{Vrfy}')$ with non-unique tags such that Π' is still UF-CMA but an attacker with access to a verification oracle can leak the entire secret key.)

(i)

Let assume uf-cmva security is associate with the following game scheme, with a similar notation as indicated in class:



uf-cmva : unforgeability under chosen message/verification attack

MAC = {KG, TAG, VRFY} is (t, Q_T, Q_V, ϵ)-uf-cmva secure if for all adversaries $\mathcal{A}$ of size t making Q_T/Q_V TAG/VRFY queries: The probability $\mathcal{A}^{\mathsf{TAG}(K,.),\mathsf{VRFY}(K,.,.)}$ makes accepting VRFY query (M, ϕ) and TAG was not queried on M before is $\leq \epsilon$.

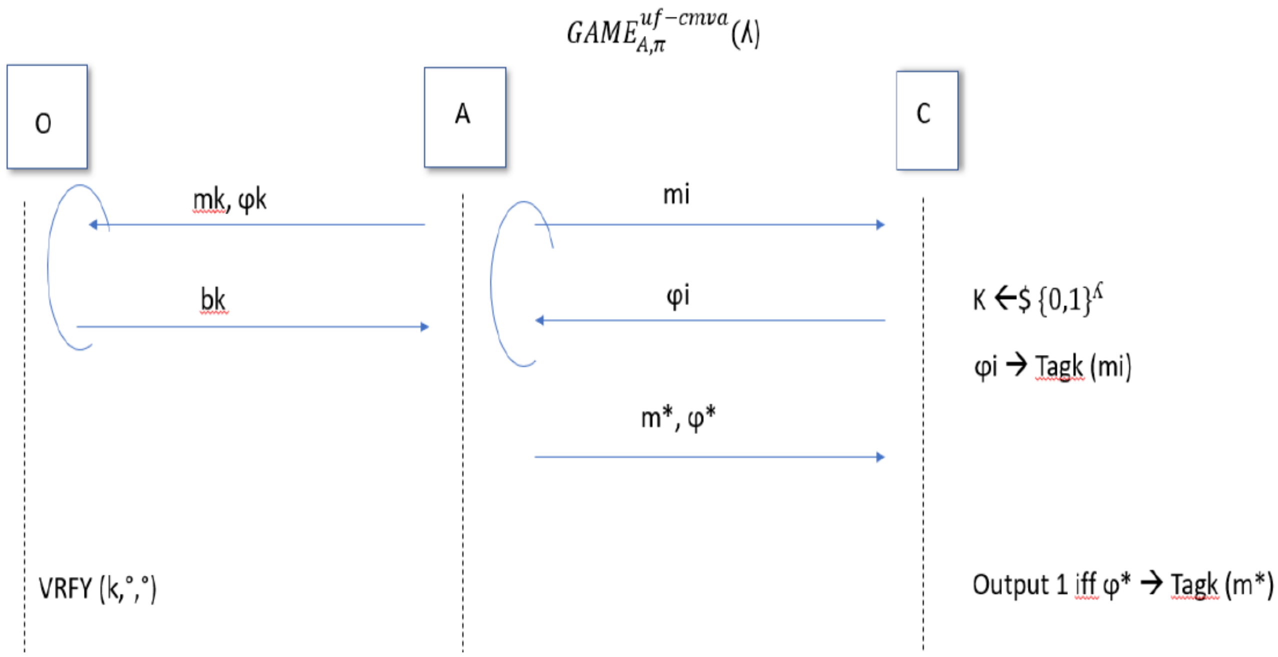
Or in equivalent and formally manner:

$G^{UF-CMVA}_{\pi,\mathcal{A}}(\lambda)$:

1) $k \leftarrow \$\{0,1\}^{\lambda}$

2) $(m^*, \phi^*) \leftarrow \mathcal{A}^{Tag_k(), Vrfy_k()}(1^{\lambda})$

3) Output 1 if and only if $Vrfy_k(m^*, \phi^*) = Accepted$ and m^* is fresh (never queried from $\mathcal{A}$). We are assuming that $\mathcal{A}$ can make Q_T queries to $Tag_k()$ and Q_v queries to $Vrfy_k()$.



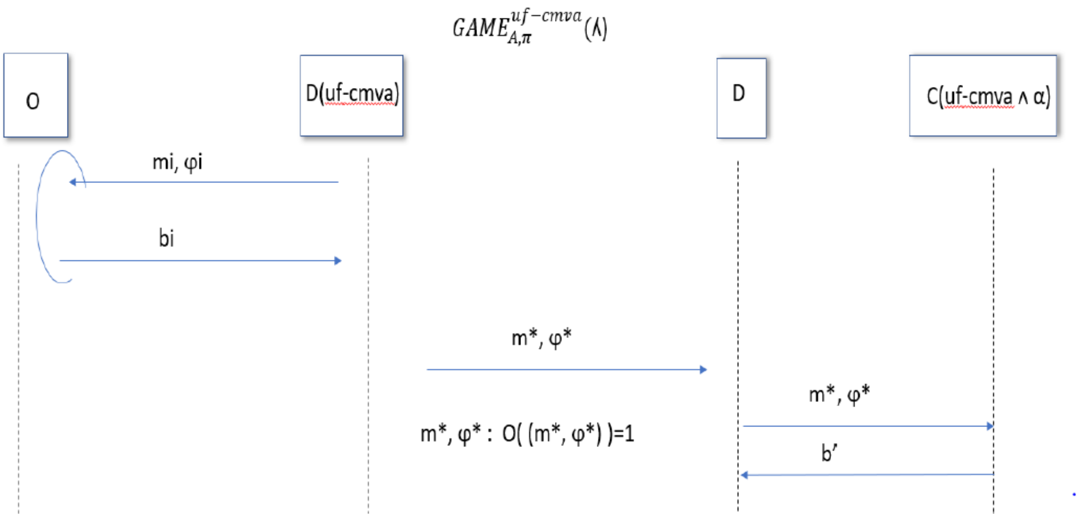
(ii)

Let assume uf-cmva security is associate with the following game scheme and with the following property of “unique” which can call α

α $\forall k\ ,\forall (m, m') \in M\ with\ m \neq m'\ : Tagk(m) = Tagk(m')\ or\ Tagk(m) \neq Tagk(m')$

$\forall k, \forall m \in M, \forall (k, m, \varphi), \forall (k, m, \varphi')\ \rightarrow\ \varphi = \varphi'$

We have to show that : (uf-cmva ∧ α) → uf-cmva



The Oracle, for construction makes operation in poly (Λ), so D(uf-cmva) it works also in poly (Λ) and so forth D
will be in poly (Λ) so implies UF-CMVA.
Why? Because if Tag is probabilistic it can assume different values with high probability, and so the Adversary
can not verify in poly-time if φ is a possible Tag for the message.

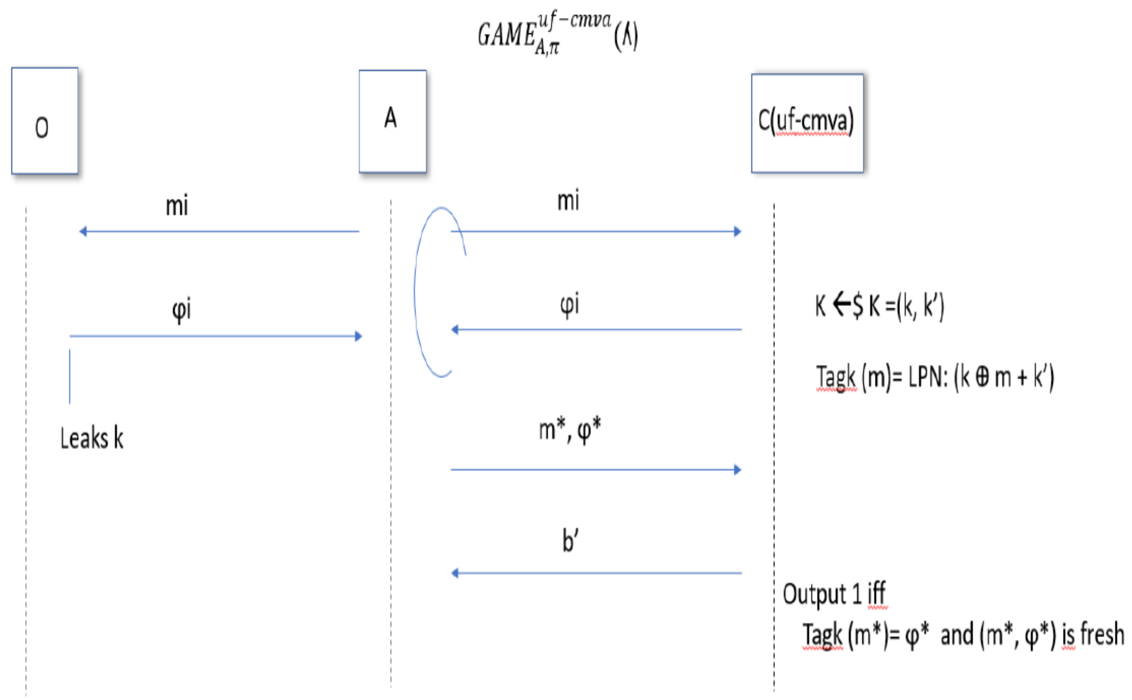
(iii)

Let assume uf-cmva security is associate with the following game scheme and with the following property of “unique” which can call α

α $\forall k\ ,\forall (m, m') \in M\ with\ m \neq m'\ : Tagk(m) = Tagk(m')\ or\ Tagk(m) \neq Tagk(m')$

$\forall k, \forall m \in M, \forall (k, m, \varphi), \forall (k, m, \varphi')\ \rightarrow\ \varphi = \varphi'$

We have to show that: (uf-cmva ∧ not α) → not uf-cmva (but only uf-cma)



The general idea is that if the tag for each m is not unique, the probability of adversary to guess a message from a given φ is
higher and so the verified attacks can not be guaranteed.

36

Recall that the Feistel permutation $\Psi_f : \{0,1\}^{2n} \to \{0,1\}^{2n}$ is defined by $\Psi_f(X) := (R, L \oplus f(R))$, where $X = L||R$, with both $L, R \in \{0,1\}^n$, and $f : \{0,1\}^n \to \{0,1\}^n$ is a function. Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^\lambda}$ be a PRF family. An r-round Feistel network $\Psi_{\mathcal{F}}[r]$ is the concatenation of r Feistel permutations, each using an independent PRF $F_{k_i} : \{0,1\}^n \to \{0,1\}^n$ from the family $\mathcal{F}$. More precisely, for every choice of the keys $k_1, \ldots, k_r \leftarrow\!\!\$ \{0,1\}^n$, a permutation $\Psi_{F_{k_1},\ldots,F_{k_r}}$ in $\Psi_{\mathcal{F}}[r]$ is defined as follows:

$$\Psi_{F_{k_1},\ldots,F_{k_r}}(X) := \Psi_{F_{k_r}}(\cdots \Psi_{F_{k_2}}(\Psi_{F_{k_1}}(X))\cdots).$$

Answer the following questions:

(a) Show that a $\Psi_{\mathcal{F}}[1]$ is not a PRP family.

(b) Show that a $\Psi_{\mathcal{F}}[2]$ is not a PRP family.

(c) Show that a $\Psi_{\mathcal{F}}[3]$ is not a *strong* PRP family.[1]

(a) As we have $\Psi[1] = (R, L \oplus f(R))$, we can distinguish two X by simply looking at the first half (which will not change if we choose two inputs with the second half identical).

(b) If we choose two inputs (L_1, R) and (L_2, R), by computing $\Psi[2]$ we have $X_1 = (L_1 \oplus f(R), R \oplus f(L_1 \oplus f(R)))$ and $X_2 = (L_2 \oplus f(R), R \oplus f(L_2 \oplus f(R)))$. If we do the XOR between the first half of X_1 and X_2, we have

$$(L_1 \oplus f(R)) \oplus (L_2 \oplus f(R)) = L_1 \oplus f(R) \oplus f(R) \oplus L_2 = L_1 \oplus L_2$$

which, in a truly random permutation, occurs with a negligible probability (so we can distinguish between two-round Feistel network and truly random permutation).

(c) To show that $\Psi[3]$ is not a S-PRP, let's consider a distinguisher D with access to $Enc(\cdot)$-oracle and $Dec(\cdot)$-oracle. We can write the output of the decryption oracle $Dec(L \parallel R)$ as:

$$X_{left}^{Dec(\cdot)} = R \oplus F_{k_2}(L \oplus F_{k_3}(R))$$

$$X_{right}^{Dec(\cdot)} = L \oplus F_{k_3}(R) \oplus F_{k_1}(R \oplus F_{k_2}(L \oplus F_{k_3}(R)))$$

And the encryption oracle $Enc(L \parallel R)$ as:

$$X_{left}^{Enc(\cdot)} = R \oplus F_{k_2}(L \oplus F_{k_1}(R))$$

$$X_{right}^{Enc(\cdot)} = L \oplus F_{k_1}(R) \oplus F_{k_3}(R \oplus F_{k_2}(L \oplus F_{k_1}(R)))$$

So we can make these queries:

- $Dec(0 \parallel 0)$ which outputs $(a = F_{k_2}(F_{k_3}(0)), b = F_{k_3}(0) \oplus F_{k_1}(a))$
- $Enc(0 \parallel a)$ which outputs $(c = a \oplus F_{k_2}(F_{k_1}(a)), d = F_{k_1}(a) \oplus F_{k_3}(c))$
- Due the fact that both b and d contains $F_{k_1}(a)$, we can compute $b \oplus d = F_{k_3}(0) \oplus F_{k_3}(c)$ and ask to the decryption oracle: $(b \oplus d \parallel c)$ and get $(e \parallel f)$

In a truly random permutation, $e = c \oplus a$ appears with negligible probability, but this is not true in 3-round Feistel network PRP, so this is not a S-PRP.

37

(a) Let $\mathcal{H} = \{H_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s \in \{0,1\}^\lambda}$ be a family of collision-resistant hash functions compressing $2n$ bits into n bits. Answer the following questions.

(i) Show that $\mathcal{H}$ is a seeded one-way function in the following sense: For all PPT adversaries A there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that

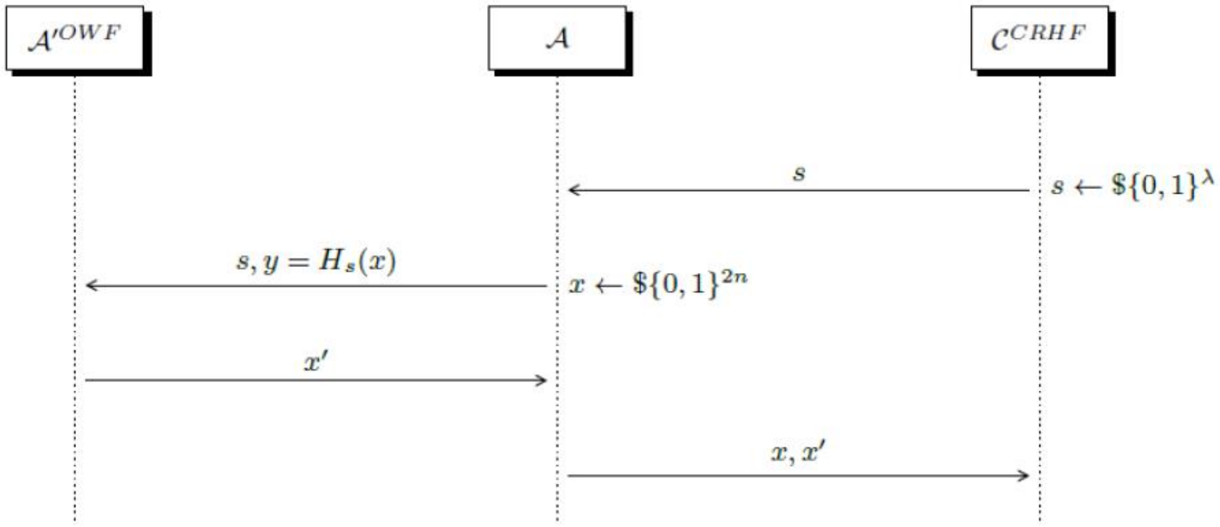
$$\Pr\left[H_s(x') = y : \; s \leftarrow \{0,1\}^\lambda; x \leftarrow \{0,1\}^{2n}; y = H_s(x); x' \leftarrow \mathsf{A}(s,y)\right] \leq \nu(n).$$

(ii) What happens in case the set of functions $\mathcal{H}$ is not compressing (i.e., the domain of each function H_s is also $\{0,1\}^n$)? Does collision resistance imply one-wayness in this case?

(a)(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

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

(a) (ii)

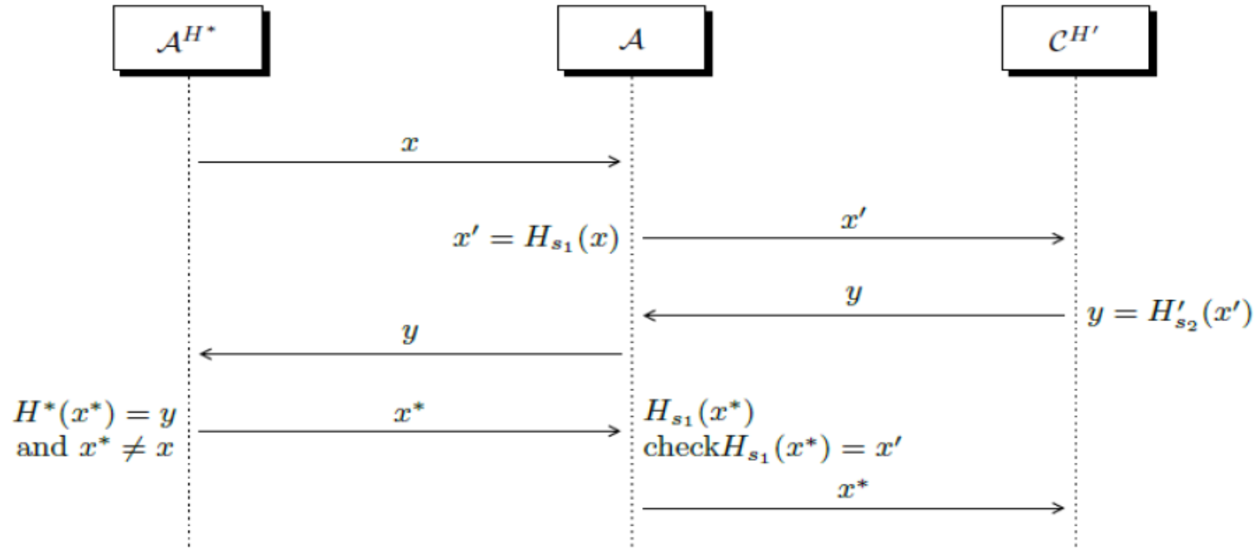
Intuitively if a function is not compressing (h : n → n) and is collision resistant we can assume it will coincident to the best hash function possible (a bijective one). In this case the function will obviously be CRHF since it is impossible to find x'≠ x such that h(x') = h(x), however this will not be one way since there will be a unique correspondence between an element in the domain and an element in the codomain.

38

(b) Let $\mathcal{H} = \{H_s : \{0,1\}^{4n} \to \{0,1\}^{2n}\}_{s\in\{0,1\}^\lambda}$ and $\mathcal{H}' = \{H'_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s\in\{0,1\}^\lambda}$ be families of collision-resistant hash functions. Analyse the following candidate hash function family compressing $4n$ bits into n bits: $\mathcal{H}^* := \{H^*_{s_1,s_2}\{0,1\}^{4n} \to \{0,1\}^n\}_{s_1,s_2\in\{0,1\}^\lambda}$ such that $H^*_{s_1,s_2}(x) = H'_{s_2}(H_{s_1}(x))$ for $s_1, s_2 \leftarrow_\$ \{0,1\}^\lambda$.

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.

39

(a) Recall that the CDH problem asks to compute g^{ab} given (g, g^a, g^b) for $(\mathbb{G}, g, q) \leftarrow\$ \mathsf{GroupGen}(1^\lambda)$ and $a, b \leftarrow\$ \mathbb{Z}_q$. Prove that the CDH problem is equivalent to the following problem: Given (g, g^a) compute g^{a^2}, where $(\mathbb{G}, g, q) \leftarrow\$ \mathsf{GroupGen}(1^\lambda)$ and $a \leftarrow\$ \mathbb{Z}_q$.

(b) Let $N = p \cdot q$ be an RSA modulus. Show how one can find p and q given $(N, \varphi(N))$ without factoring N. (Recall that $\varphi(N) := (p-1)(q-1)$.)

Apply the above fact to the following setting: $N = 18830129$ and $\varphi(N) = 18819060$.

(c) Alice and Bob belong to the same organization and are given public keys e_A and e_B, respectively, corresponding to a common public RSA modulus N. Assume that e_A and e_B are relatively prime. Let $c_A = m^{e_A} \bmod N$ and $c_B = m^{e_B} \bmod N$ be two RSA encryptions of the same message $m \in \mathbb{Z}_N^*$, under public keys e_A and e_B (respectively). Prove that an eavesdropper given e_A, e_B, c_A and c_B can recover m.

(**Hint:** Use Bézout's identity.)

(a) If we can compute g^{ab} given (g, g^a, g^b), then to compute g^{a^2} it's sufficies to have (g, g^a, g^a).
On the other hand, if I can compute g^{a^2} then I can do the following:

$$g^{(a+b)^2} = g^{a^2+b^2+2ab} = g^{a^2} g^{b^2} g^{2ab}$$

then dividing by $g^{a^2} g^{b^2}$, get g^{2ab} so $g^{ab} = \sqrt{g^{2ab}}$

(b) Knowning that $\varphi(N) = (p-1)(q-1)$ we can write:

$$\varphi(N) = pq - p - q + 1 = N - p - q + 1$$

$$p + q = N + 1 - \varphi(N) = p + \frac{N}{p} = N + 1 - \varphi(N)$$

$$\frac{p^2 + N}{p} = p(N + 1 - \varphi(N))$$

Then we need to solve:

$$p^2 - p(N + 1 - \varphi(N)) + N = 0$$

By given $\varphi(N) = 18819060$ and $N = 18830129$:

$$p^2 - p(18830129 + 1 - 18819060) + 18830129 = 0$$

$$p_{1,2} = 5535 \pm \sqrt{5535^2 - 18830129} = \begin{cases} p_1 = 8971 \\ p_2 = 2099 \end{cases} \tag{2}$$

So we found $p = p_1$ and $q = p_2$ such that $p \cdot q = N$ $(8971 \cdot 2099 = 18830129)$

(c) By Bézout's identity, exists x, y such that $ax + by = gcd(a, b)$. By definition of this problem, $gcd(e_A, e_B) = 1$, so we can compute the Bézout's identity for $1 = x \cdot e_A + y \cdot e_B$. Since:

$$c_A = m^{e_A} \bmod N$$

$$c_B = m^{e_B} \bmod N$$

So, if we multiply c_A and c_B we obtain m:

$$c_A^x \cdot c_B^y = (m^{e_A})^x \cdot (m^{e_B})^y \bmod N = m^{x \cdot e_A + y \cdot e_B} \bmod N = m \bmod N$$

39

(a) Consider the following relaxation of CCA security for PKE schemes, so-called CCA1, where the adversary can only access the decryption oracle prior to receiving the challenge ciphertext (whereas in CCA security, also known as CCA2, the attacker is allowed to make decryption queries even after being given the challenge ciphertext). Give a formal definition of CCA1 security for PKE schemes.

(b) Prove formally that CCA1 security implies CPA security for any PKE scheme. On the other hand, show that there exists a PKE scheme that is CPA secure but not CCA1 secure.

(c) Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\{0,1\}$ (i.e., for encrypting a single bit). Consider the following natural construction of a multi-bit PKE scheme $\Pi' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ with message space $\{0,1\}^t$, for some polynomial $t = t(\lambda)$: (i) The key generation stays the same, i.e. $\mathsf{KGen}'(1^\lambda) = \mathsf{KGen}(1^\lambda)$; (ii) Upon input $m = (m[1], \ldots, m[t]) \in \{0,1\}^t$ the encryption algorithm $\mathsf{Enc}'(pk, m)$ outputs a ciphertext $c = (c_1, \ldots, c_t)$ where $c_i \leftarrow\$ \mathsf{Enc}(pk, m[i])$ for all $i \in [t]$; (iii) Upon input a ciphertext $c = (c_1, \ldots, c_t)$ the decryption algorithm $\mathsf{Dec}'(sk, c)$ outputs the same as $(\mathsf{Dec}(sk, c_1), \ldots, \mathsf{Dec}(sk, c_t))$.

(i) Show that if Π is CCA1 secure, so is Π'.

(ii) Show that, even if Π is CCA2 secure, Π' is not CCA2 secure.

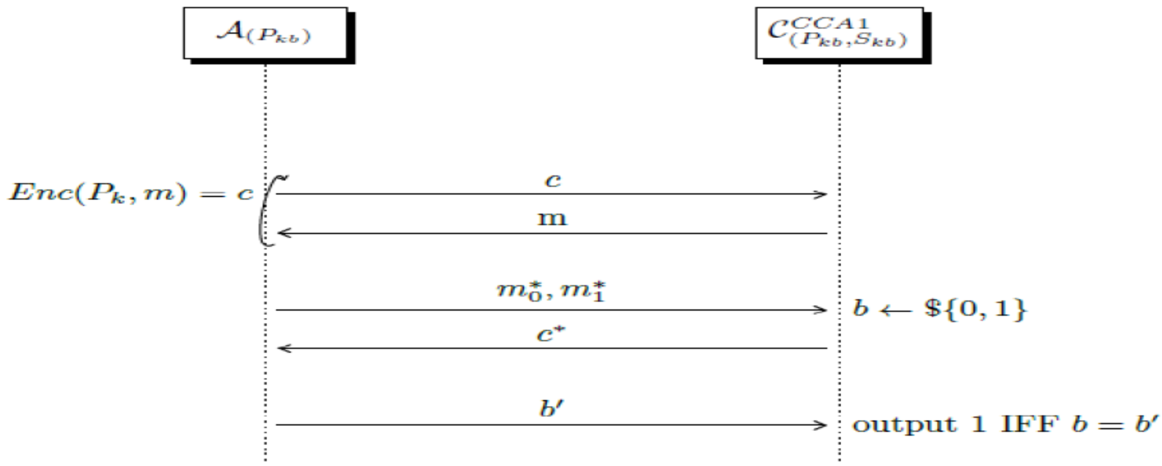
(d) Recall the Padded RSA PKE scheme. Let $N = p \cdot q$, and e, d be such that $d \cdot e \equiv 1 \bmod (p-1)(q-1)$. The public key is $pk := (N, e)$ and the secret key is $sk := (N, d)$. Upon input a message $m \in \{0,1\}^{\ell(\lambda)}$, pick a random $r \leftarrow \{0,1\}^{|N|-\ell(\lambda)-1}$ and output $c := (r||m)^e \bmod N$. To decrypt a ciphertext $c \in \mathbb{Z}_N^*$ compute $\hat{m} := c^d \bmod N$, parse $\hat{m} := r||m$, and return m.

Show that Padded RSA is not CCA2 secure, by exhibiting a concrete chosen-ciphertext attack and analyzing its success probability.

(**Hint:** For simplicity, you may assume that the answer to a decryption query consists of the entire padded message $r||m$ and not just of the last ℓ significant bits of it.)

(a)

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



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

If Malice has no strategy to win Game 1 better than random guessing, then (*Gen*,*Enc*,*Dec*) is IND-CCA1 secure for any single message.
In other way and with other equivalent symbols:

A cryptosystem is (t, ε)-IND-CCA1 secure if for all t-time adversaries $\mathcal{A}$:

$$\mathsf{Adv}^{\text{ind-cca1}}(\mathcal{A}) = \left|\Pr\left[\mathcal{G}_0^{\mathcal{A}} = 1 | \mathcal{G}_0\right] - \Pr\left[\mathcal{G}_1^{\mathcal{A}} = 1 | \mathcal{G}_1\right]\right| \leq \varepsilon \ ,$$

where the security games are defined as follows

$\mathcal{G}_0^{\mathcal{A}}$

$$\left[\begin{array}{l} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\mathsf{pk}) \\ \text{return } \mathcal{A}(\mathsf{Enc}_{\mathsf{pk}}(m_0)) \end{array}\right.$$

$\mathcal{G}_1^{\mathcal{A}}$

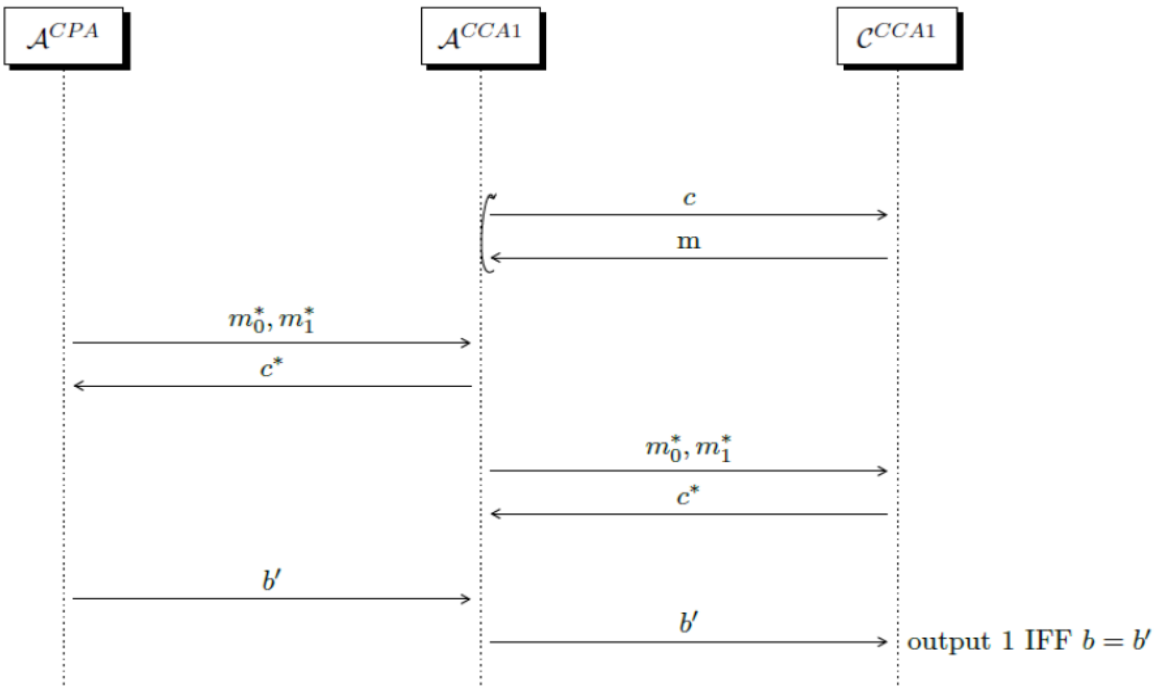
$$\left[\begin{array}{l} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\mathsf{pk}) \\ \text{return } \mathcal{A}(\mathsf{Enc}_{\mathsf{pk}}(m_1)) \end{array}\right.$$

and the oracle $\mathcal{O}_1$ serves decryption queries, i.e., $\mathcal{O}_1(c) = \mathsf{Dec}_{\mathsf{sk}}(c)$.

(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).

$PKE^{CPA} \implies CCA1$

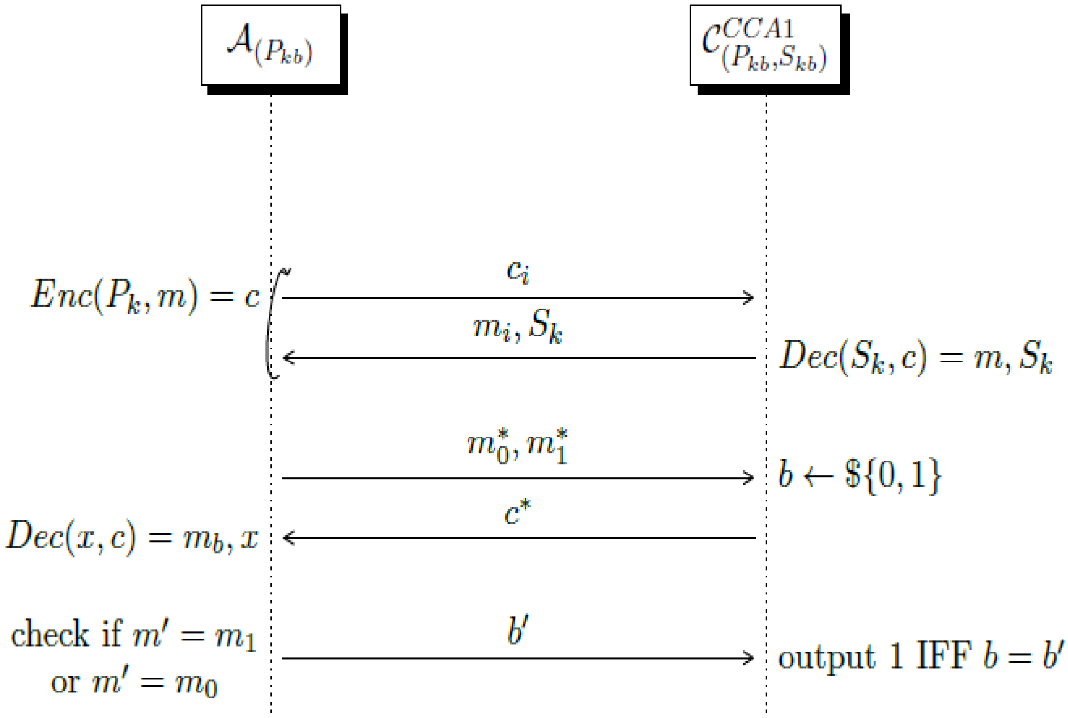
Now consider the following Game which is still CPA secure but on the other hand it leaks the key whenever C receives a decryption query.

The scheme is defined as follows

Correctness:

$Dec(S_k, \underbrace{Enc(P_k, m)}_{c}) = m, S_k$

The scheme is still correct since in the decryption query I will still have the message as output + a second member which is the leaked key.



(c)(i)

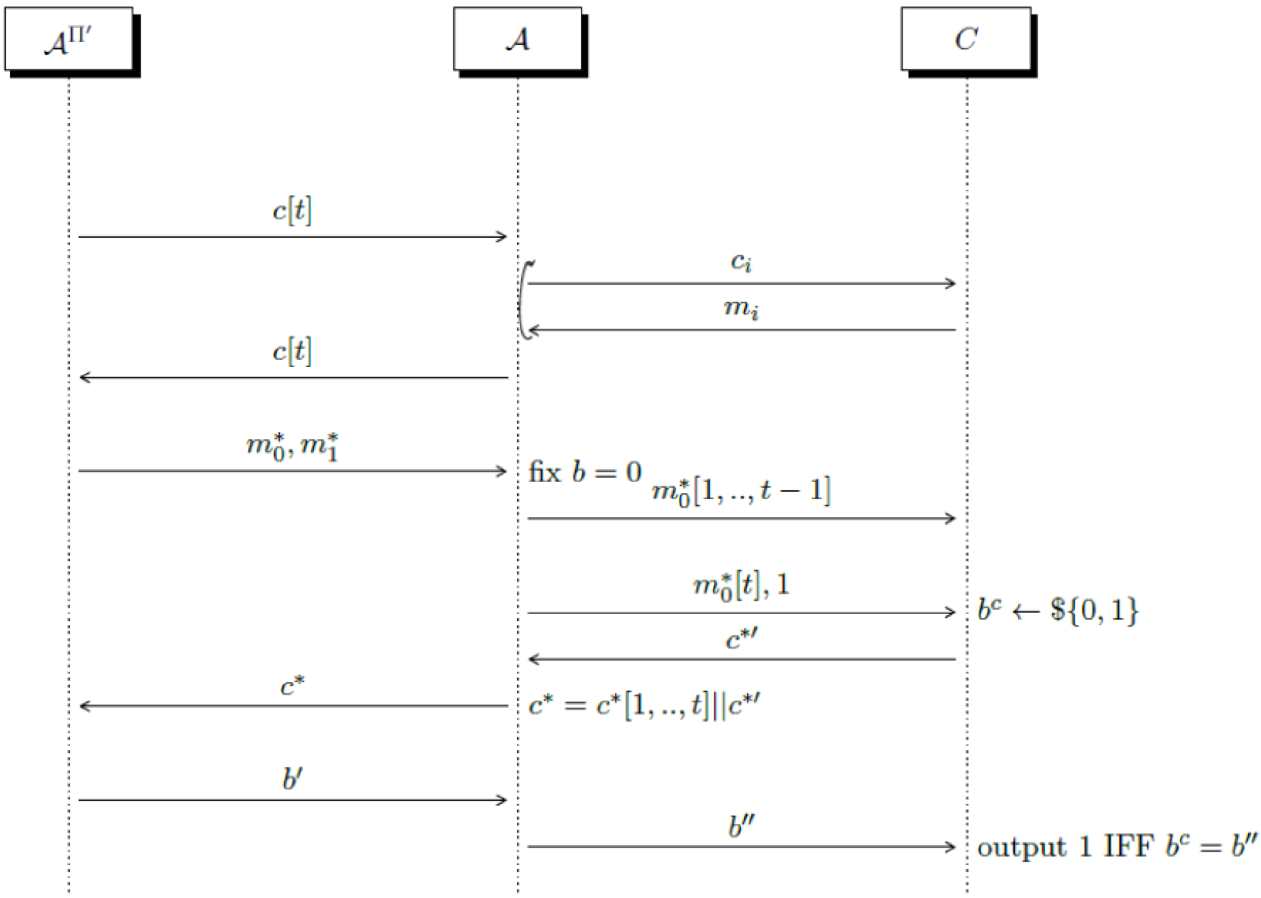
goal is to demonstrate if Π is $CCA1 \Rightarrow \Pi'$ is also CCA1, in order to do this observe the following reduction scheme:

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



(c)(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.

(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.

40 (SIGNATURE SCHEME) (HW2 2017/2018) mark=11/20

Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with message space $\mathcal{M} = \{0,1\}^\ell$, for some fixed $\ell \in \mathbb{N}$. Consider a family of hash functions $\mathcal{H} = \{h_s : \{0,1\}^* \to \{0,1\}^\ell\}_{s\in\{0,1\}^\lambda}$.

Define the following derived signature scheme $\Pi' = (\mathsf{KGen}', \mathsf{Sign}', \mathsf{Vrfy}')$: (i) Algorithm $\mathsf{KGen}'(1^\lambda)$ returns (pk', sk') such that $pk' = (pk, s)$ and $sk' = (sk, s)$, where $(pk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$ and $s \leftarrow\!\!\$\ \{0,1\}^\lambda$; (ii) Algorithm $\mathsf{Sign}'(sk', m)$ takes a message $m \in \{0,1\}^*$ of arbitrary length, and outputs $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(sk, h_s(m))$; (iii) Algorithm $\mathsf{Vrfy}'(pk', m, \sigma)$ returns the same as $\mathsf{Vrfy}(pk, h_s(m), \sigma)$. Prove that if Π is UF-CMA and $\mathcal{H}$ is collision-resistant, then Π' is UF-CMA.

To show that Π' is *UF-CMA* let's consider an experiment (we can name it $Forge_{\mathcal{A}',\Pi'}$) with $\mathcal{A}'$ as a PPT adversary that tries to break Π'. By the definition, we can write as follow:

$$Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1] = Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1 \wedge collide] + Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1 \wedge \overline{collide}]$$

Where *collide* is the event of collision for $h_s(\cdot)$ and $Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1]$ is the probability that $\mathcal{A}'$ can break Π' (ie. forge a new message).

By looking at the previous formula, we can write this:

$$Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1] \le Pr[collide] + Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1 \wedge \overline{collide}]$$

So it is sufficient to show that both $Pr[collide]$ and $Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1 \wedge \overline{collide}]$ are negligible.

About the former: $Pr[collide]$ is negligible by definition of $\mathcal{H}$: if we have a collision with probability non negligible, $\mathcal{H}$ is not a CRH function family (this is a contradiction as we stated, in the problem, that $\mathcal{H}$ is CRH function family).

About the latter: we can easly see that (by definition of Π' schema):

$$Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1 \wedge \overline{collide}] = Pr[Forge_{\mathcal{A},\Pi}(n) = 1]$$

Where $\mathcal{A}$ is the attacker that tries to break Π schema. Then, by definition of Π (which is UF-CMA), this probability is negligible. So, $Pr[collide] + Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1 \wedge \overline{collide}]$ is negligible. The same, of course, for $Pr[Forge_{\mathcal{A}',\Pi'}(n) = 1]$.

41 (IDENTIFICATION SCHEMES) (HW2 2017/2018) mark=XX/20

6 Identification Schemes 20 Points

Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a canonical ID scheme with challenge space $\mathcal{B}_\lambda$. For any polynomial $t = t(\lambda)$, consider the following t-fold parallel repetition of Π, denoted $\Pi^t := (\mathsf{Setup}, \mathsf{P}^t, \mathsf{V}^t)$: First $(pk, sk) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$ is run and then $\mathsf{P}^t(pk, sk)$ and $\mathsf{V}^t(pk)$ interact by simply running t independent executions of the original ID scheme between $\mathsf{P}(pk, sk)$ and $\mathsf{V}(pk)$ in parallel.

A little more formally, let $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ and $\mathsf{V} = (\mathsf{V}_1, \mathsf{V}_2)$ be the algorithms of the underlying canonical ID scheme, where recall that $\mathsf{V}_1(1^\lambda)$ simply returns a random challenge $\beta \leftarrow\!\!\$\ \mathcal{B}_\lambda$. An interaction between P^t and V^t results in a transcript $\vec{\tau} \leftarrow\!\!\$\ (\mathsf{P}^t(pk, sk) \leftrightarrows \mathsf{V}^t(pk))$, where $\vec{\tau} := (\vec{\alpha}, \vec{\beta}, \vec{\gamma})$ is computed as follows:

1. P^t runs $\alpha_i \leftarrow\!\!\$\ \mathsf{P}_1(pk, sk)$ for all $i \in [t]$ and forwards $\vec{\alpha} = (\alpha_1, \ldots, \alpha_t)$ to V^t.
2. V^t samples $\beta_i \leftarrow\!\!\$\ \mathcal{B}_\lambda$ for all $i \in [t]$ and forwards $\vec{\beta} = (\beta_1, \ldots, \beta_t)$ to P^t.
3. P^t runs $\gamma_i \leftarrow\!\!\$\ \mathsf{P}_2(pk, sk, \alpha_i, \beta_i)$ for all $i \in [t]$, and forwards $\vec{\gamma} = (\gamma_1, \ldots, \gamma_t)$ to V^t.
4. V^t returns 1 if and only if $\mathsf{V}_2(pk, (\alpha_i, \beta_i, \gamma_i)) = 1$ for all $i \in [t]$.

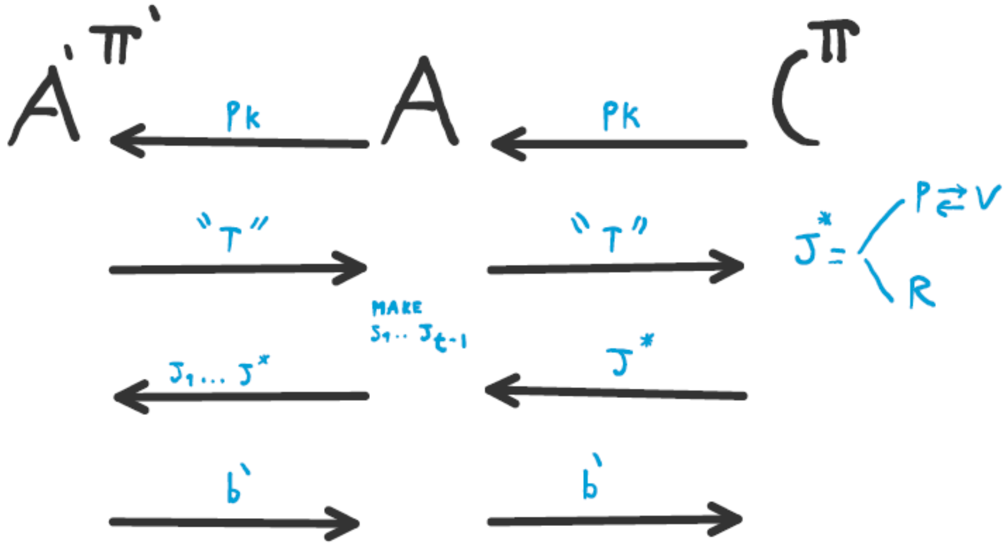
Answer the following questions.

(a) Show that Π^t is a canonical ID scheme. What is the challenge space?

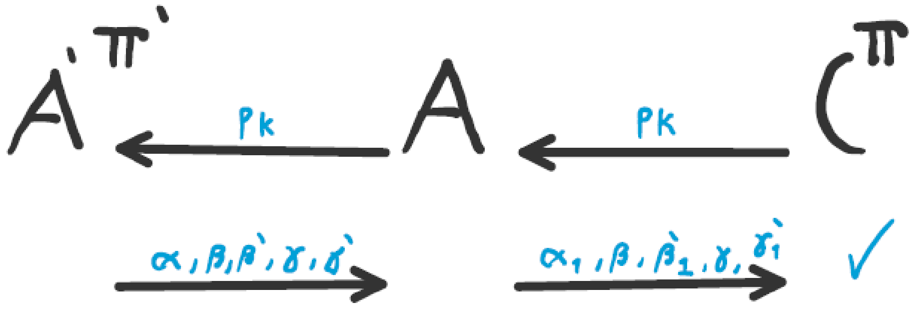
(b) Prove that as long as Π satisfies completeness, special soundness, and honest-verifier zero knowledge, so does Π^t (for any polynomial $t(\lambda)$).

(a) Due to the specification of the exercise, $\vec{a}$ is non-degenerate because it's composed by a_t which are not-degenerates (due to the fact that they're in a canonical ID scheme). Also correctness is the same. The message space is b^t

Π HVZK $\rightarrow$ Π' HVZK



Π ss $\rightarrow$ Π' ss



42 PRG (SAARLAND)

Exercise No.2 (PRGs) *(7 points)*
Let G be a PRG with expansion factor $\ell(n) > n$ and let $f : \{0,1\}^* \to \{0,1\}^*$ be a length-preserving bijection (i.e., a permutation) such that f is computable in deterministic polynomial time and define G' as follows:

$$G'(s) := f(G(s))$$

Show that G' is also a PRG.

Solution No.2 (PRGs) From the definition it follows that the expansion factor of G' is also ℓ. We prove the claim by reduction: Assuming G' is not a PRG, there is a ppt distinguisher D' for G' such that there is a polynomial q such that for all n

$$\left|\Pr_{s\leftarrow\{0,1\}^n}[D'(G'(s)) = 1] - \Pr_{r\leftarrow\{0,1\}^{\ell(n)}}[D'(r) = 1]\right| > \frac{1}{q(n)}\,. \tag{1}$$

We construct a distinguisher D for G from D' as follows: On input t, D just simulates $D'(f(t))$, that is, D outputs 1 if and only if D' does on input $f(t)$. As D' is ppt and f is polynomial time computable it follows that D is also ppt. Now we have that for all n:

$$\begin{aligned}
&\left|\Pr_{s\leftarrow\{0,1\}^n}[D(G(s)) = 1] - \Pr_{r\leftarrow\{0,1\}^{\ell(n)}}[D(r) = 1]\right| \\
=&\left|\Pr_{s\leftarrow\{0,1\}^n}[D'(f(G(s))) = 1] - \Pr_{r\leftarrow\{0,1\}^{\ell(n)}}[D'(f(r)) = 1]\right| \\
=&\left|\Pr_{s\leftarrow\{0,1\}^n}[D'(G'(s)) = 1] - \Pr_{r\leftarrow\{0,1\}^{\ell(n)}}[D'(f(r)) = 1]\right| \\
=&\left|\Pr_{s\leftarrow\{0,1\}^n}[D'(G'(s)) = 1] - \Pr_{r'\leftarrow\{0,1\}^{\ell(n)}}[D'(r') = 1]\right| \\
>&\frac{1}{q(n)}
\end{aligned}$$

where the third equality follows from the fact that f is a length-preserving bijection and the inequality follows from (1).
This contradicts the fact that G is a PRG which completes the proof.

43 PRF (SAARLAND)

Exercise No.3 (PRFs) *(8 points)*
Consider the following keyed function F: For security parameter n, the key is a pair (k_1, k_2) where $k_1, k_2 \in \{0,1\}^n$. Define $F_{(k_1,k_2)} : \{0,1\}^n \to \{0,1\}^n$ by

$$F_{(k_1,k_2)}(x) := k_1 \oplus x \oplus k_2$$

Show that F is not a PRF.

Solution No.3 (PRFs) We construct a distinguisher D as follows: On input 1^n and having access to oracle $\mathcal{O}$, D queries $m_0 := \mathcal{O}(0^n)$ and $m_1 := \mathcal{O}(1^n)$. After that, D checks whether $m_0 \oplus m_1 = 1^n$ and outputs 1 accordingly.
If $\mathcal{O} = F_{(k_1,k_2)}$ for some (randomly chosen) k_1, k_2, we have that

$$m_0 \oplus m_1 = \mathcal{O}(0^n) \oplus \mathcal{O}(1^n) = k_1 \oplus 0^n \oplus k_2 \oplus k_1 \oplus 1^n \oplus k_2 = 1^n$$

and therefore

$$\Pr_{\substack{k_1\leftarrow\{0,1\}^n\\ k_2\leftarrow\{0,1\}^n}}[D^{F_{(k_1,k_2)}(\cdot)}(1^n) = 1] = 1\,.$$

Now if $\mathcal{O}$ is a truly random function f, we have that $f(0^n)$ and $f(1^n)$ are random strings and hence $f(0^n) \oplus f(1^n)$ is also. This implies that

$$\Pr_{f\leftarrow\mathrm{func}_n}[D^{f(\cdot)}(1^n) = 1] = 2^{-n}\,.$$

We conclude that

$$\left|\Pr_{\substack{k_1\leftarrow\{0,1\}^n\\ k_2\leftarrow\{0,1\}^n}}[D^{F_{(k_1,k_2)}(\cdot)}(1^n) = 1] - \Pr_{f\leftarrow\mathrm{func}_n}[D^{f(\cdot)}(1^n) = 1]\right| = 1 - 2^{-n}$$

which is clearly not negligible. It follows that F is not a PRF.

44 (IDENTIFICATION SCHEMES) (ESAME) mark=XX/10

3 Actively Secure ID Schemes 10 Points

Let $\Pi = (\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ be an ID scheme. Informally, an ID scheme is actively secure if no efficient adversary $\mathcal{A}$ (given just the public key pk) can make $\mathcal{V}$ accept, even after $\mathcal{A}$ participates maliciously in polynomially many interactions with $\mathcal{P}$ (given both the public

key pk and the secret key sk). More formally, we say that Π satisfies active security if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there is a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that for any polynomial $n := n(\lambda)$ the following holds:

$$\Pr\left[\mathsf{out}_{\mathcal{V}}(\mathcal{A}_2(pk, s_n) \rightleftarrows \mathcal{V}(pk)) = 1 : \begin{array}{c}(pk, sk) \leftarrow\!\!\$\; \mathsf{Gen}(1^\lambda); s_0 := \varepsilon \\ (\forall i \in [n]) s_i \leftarrow (\mathcal{P}(pk, sk) \rightleftarrows \mathcal{A}_1(pk, s_{i-1}))\end{array}\right] \leq \nu(\lambda),$$

where s_0 is the empty string, $s_i \in \{0, 1\}^*$ is some arbitrary state information, and where the probability is taken over the random coin tosses of algorithms Gen, $\mathcal{A}$, and $\mathcal{P}$. Answer the following questions.

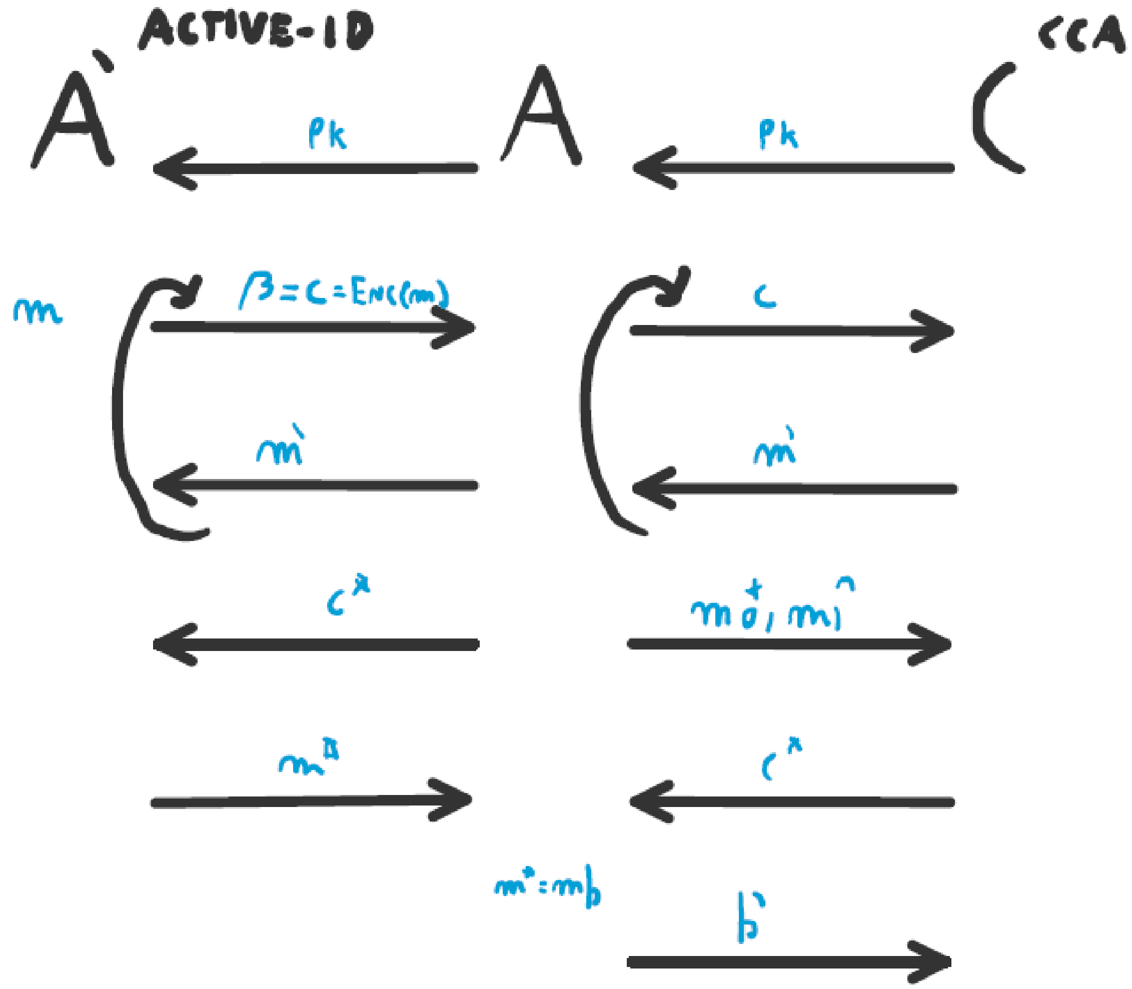
(a) Let $\Pi' = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, with message space $\mathcal{M}$. Prove that if Π' is CCA secure, the following ID scheme Π (based on Π') achives active security:

$\mathsf{Gen}(1^\lambda)$: Run $(pk, sk) \leftarrow\!\!\$\; \mathsf{KGen}(1^\lambda)$ and output (pk, sk).

$\mathcal{P}(pk, sk) \rightleftarrows \mathcal{V}(pk)$: The verifier picks random $m \leftarrow\!\!\$\; \mathcal{M}$, and forwards $c \leftarrow\!\!\$\; \mathsf{Enc}(pk, m)$ to the prover. The prover replies with $m' = \mathsf{Dec}(sk, c)$, and finally the verifier accepts if and only if $m' = m$.

(b) Is the above protocol honest-verifier zero-knowledge? Prove your answer.

2) Π' IS CCA $\rightarrow$ Π ACTIVE



b) Non so rispondere perché la funzione non è conosciuta (NON SO COSTRUIRE SIMULATORE)

45 (MAC) (KATZ 4.6)

1. Consider the following fixed-length MAC for messages of length $\ell(n) = 2n - 2$ using a pseudorandom function F: On input a message $m_0||m_1$ (with $|m_0| = |m_1| = n - 1$) and key $k \in \{0,1\}^n$, algorithm Mac_k outputs $t = F_k(0||m_0)||F_k(1||m_1)$. Algorithm Vrfy is defined in the natural way. Is $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ existentially unforgeable under a chosen-message attack? Prove your answer.

   Answer. This scheme is not secure. Let A be an adversary that queries its oracle with two messages $m = m_0||m_1$ and $m' = m'_0||m'_1$, where $m_0 \neq m'_0$ and $m_1 \neq m'_1$. Let $t = t_0||t_1$ and $t' = t'_0||t'_1$ be the respective responses from its oracle. $\mathcal{A}$ then outputs the message $\bar{m} = m_0||m'_1$ and tag $\bar{t} = t_0||t'_1$. By the definition of Mac, it follows that $\bar{t}$ is a correct tag for $\bar{m}$ and thus $\mathsf{Vrfy}_k(\bar{m}, \bar{t}) = 1$ always. Furthermore, since $m_0 \neq m'_0$ and $m_1 \neq m'_1$ we have that $\bar{m} \notin Q$. Thus $\mathcal{A}$ succeeds with probability 1 and the scheme is not secure.

46 (MAC) (KATZ 4.6 like)

2. Let F be a pseudorandom function. Show that the following MAC for messages of length $2n$ is insecure: The shared key is a random $k \in \{0,1\}^n$. To authenticate a message $m_1||m_2$ with $|m_1| = |m_2| = n$, compute the tag $\langle F_k(m_1), F_k(F_k(m_2))\rangle$.

   Answer. The idea for this solution is exactly the same as for the previous exercise. Other attacks are also possible.

47 (MAC) (KATZ 4.7)

3. Let F be a pseudorandom function. Show that each of the following message authentication codes is insecure. (In each case the shared key is a random $k \in \{0,1\}^n$.)

   (a) To authenticate a message $m = m_1||\cdots||m_\ell$, where $m_i \in \{0,1\}^n$, compute $t := F_k(m_1) \oplus \cdots \oplus F_k(m_\ell)$.

   (b) To authenticate a message $m = m_1||\cdots||m_\ell$, where $m_i \in \{0,1\}^n$, choose $r \leftarrow \{0,1\}^n$ at random, compute $t := F_k(r) \oplus F_k(m_1) \oplus \cdots \oplus F_k(m_\ell)$, and send $\langle r, t\rangle$.

   (c) To authenticate a message $m = m_1||\cdots||m_\ell$, where $m_i \in \{0,1\}^{n/2}$, choose $r \leftarrow \{0,1\}^n$ at random, compute

   $$t := F_k(r) \oplus F_k(\langle 1\rangle||m_1) \oplus \cdots \oplus F_k(\langle \ell\rangle||m_\ell),$$

   (where $\langle i\rangle$ is the $n/2$-bit encoding of the integer i), and send $\langle r, t\rangle$.

Answer.

(a) Let $m_1||m_2$ be any message with $m_1, m_2 \in \{0,1\}^n$. Then, the tag on $m_1||m_2$ is identical to the tag on $m_2||m_1$. Thus, an adversary $\mathcal{A}$ can ask for a tag on $m_1||m_2$ and output the message $m_2||m_1$ together with the tag it received.

(b) As with the previous item, the tag $\langle r, t\rangle$ on $m_1||m_2$ is acceptable also for $m_2||m_1$.

(c) There is an attack on this scheme that does not request any tags. Let $m_1 \in \{0,1\}^{n/2}$ be arbitrary, and set $r := \langle 1\rangle||m_1$. Then $\langle r, 0^n\rangle$ is a valid tag on m_1.

a) $m = m_0, \dots, m_\ell$ where $m_i \in \{0,1\}^n$

compute $t = F_k(m_1) \oplus \dots \oplus F_k(m_\ell)$

SHOW THAT IS INSICURE

$$A \xrightarrow{\quad m = m_0 \| m_1 \quad} C$$

$$A \xleftarrow{\quad \Phi \quad} C$$

$$A \xrightarrow{\quad m^* = m_1 \| m_0 ,\ \Phi \quad} C$$

C

$m = m_0, \dots, m_\ell$ where $m_i \in \{0,1\}^{n/2}$

compute $t = F_k(\langle 1 \rangle \| m_1) \oplus \dots \oplus F_k(\langle \ell \rangle \| m_\ell)$

SHOW THAT IT IS INSECURE

$$A \xrightarrow{\quad m_1 \| m_2 \quad} C$$

$$A \xleftarrow{\quad \Phi = F_k(1 \| m_1) \oplus F_k(2 \| m_2) \quad} C$$

$$A \xrightarrow{\quad m_2 \| m_1 \quad} C$$

$$A \xleftarrow{\quad \Phi' = F_k(1 \| m_2) \oplus F_k(2 \| m_1) \quad} C$$

$$A \xrightarrow{\quad m_2 \| m_2 \quad} C$$

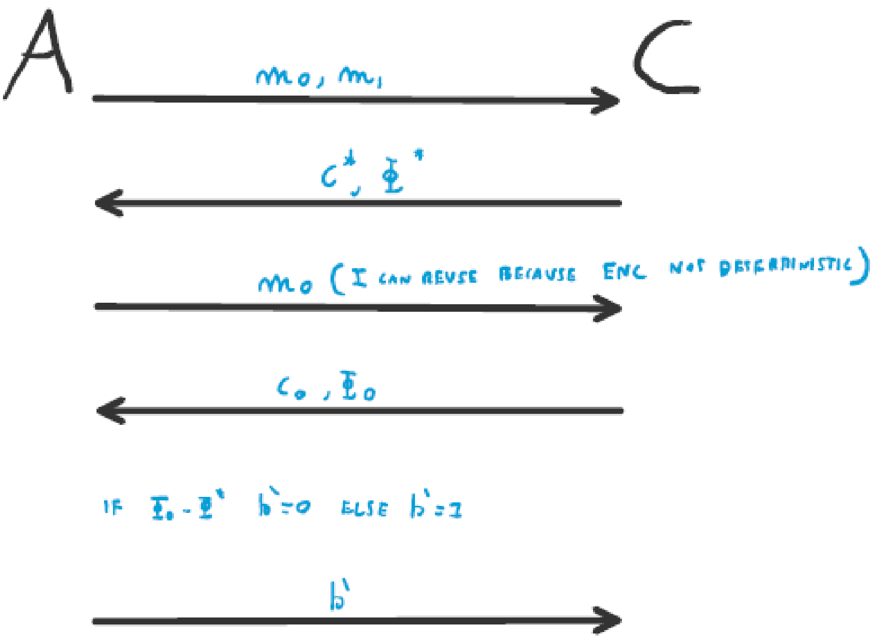
$$A \xleftarrow{\quad \Phi'' = F_k(1 \| m_2) \oplus F_k(2 \| m_2) \quad} C$$

$$A \xrightarrow{\quad m^* = m_1 \| m_1 ,\ \Phi^* = \Phi \oplus \Phi' \oplus \Phi'' \quad} C$$

48 (MAC) (BURMESTE)

Show that if any message authentication code having unique tags is used in the encrypt-and-authenticate approach, the resulting combination is not CPA-secure.

Answer. If the MAC has unique tags then two encryptions of the same message will have the same tag. Thus, a CPA adversary $\mathcal{A}$ can output any pair of messages m_0, m_1 and then ask for an encryption of m_0. If the tag portion of the ciphertext of the encryption of m_0 is the same as in the challenge ciphertext, then $\mathcal{A}$ outputs $b_0 = 0$; otherwise it outputs $b' = 1$. $\mathcal{A}$ succeeds in this game with probability 1.

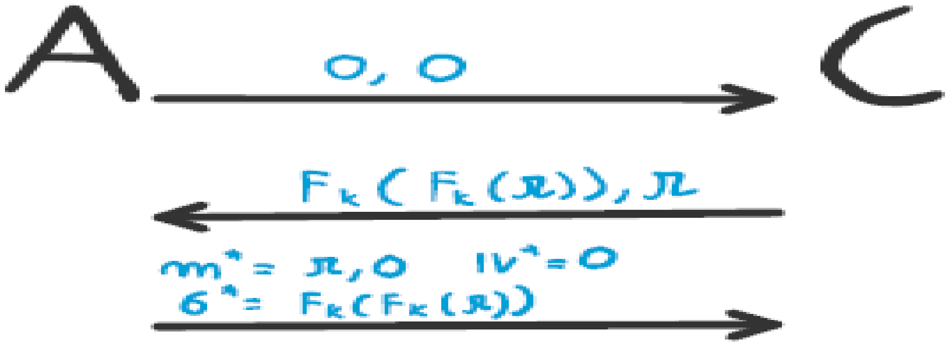


49 (MAC CBC) (BURMESTE)

Prove that the following modification of CBC-MAC does not yield a secure fixed-length MAC:

Modify CBC-MAC so that a random IV is used each time a tag is computed (and the IV is output along with t_ℓ). I.e., $t_0 \leftarrow \{0,1\}^n$ is chosen uniformly at random rather than being fixed to 0^n, and the tag is t_0, t_ℓ.

Answer. The scheme is not secure. In particular, let m be a one-block message and let (IV, t) be its MAC-tag received from the oracle. Then, the tag (m, t) is a valid MAC for the message IV . (More generally, for any m' the tag $(IV \oplus m', t)$ is a valid MAC for the message $m \oplus m'$. This means that it is possible to generate a forgery for any desired single-block message.)

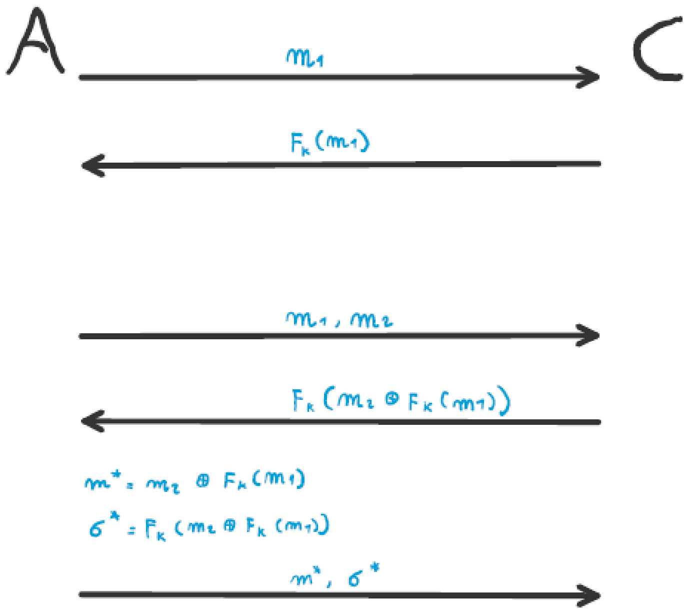


50 (MAC CBC) (MARYLAND)

Consider the following variant of CBC-MAC: The sender and receiver share a secret key k of length $\ell(n) \cdot n$, viewed as a vector of keys $k = \langle k_1, \ldots, k_\ell \rangle$ with $|k_i| = n$ for all i. Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a pseudorandom function. To authenticate a message m, the parties do the following:

- Let $|m| = j \cdot n$, with $1 \leq j \leq \ell$. (If the message is too long, or its length is not a multiple of n, no authentication tag is computed.)
- Compute the CBC-MAC on m using key k_j.

(a) Is this scheme secure (when the adversary can obtains authentication tags on messages of different lengths) or not? If not, show an attack. If yes, give a proof (in this case you may assume security of CBC-MAC for *fixed-length* messages only).



51 (MAC CBC) (ZURIGO)

c. SSHv2 uses the following mechanism for combining symmetric encryption and MAC:

$$C = E_{k_1}(M) \| MAC_{k_2}(M) \tag{1}$$

Show that there exist a semantically secure encryption system (against a passive adversary) and an existentially unforgeable MAC for which the construction above is not semantically secure. Use a semantically secure encryption system $\overline{E}$ and a MAC that is existentially unforgeable, $\overline{MAC}$, as building blocks.
Hint: try changing $\overline{MAC}$ in a way that retains the security of the MAC, but would break semantic security of the construction in (1).

In this case MAC is deterministic $\Rightarrow$ C NOT CPA secure

52 (MAC) (SAARLAND)

Exercise 8.3 (Random-Oracle model, 2+2+2 Points) Let (Gen, H) be a collision-resistant hash function with inputs of arbitrary size. We define a MAC for arbitrary-length messages by

$$\text{Mac}_{s,k}(m) = H^s(k\|m) .$$

(a) Show that this is not a secure MAC if H is constructed by the Merkle-Damgard transform from an arbitrary collision-resistant hash function h. (We assume that s is known to the attacker.)

(b) Show that this MAC is secure if H is modeled as a random oracle.
Hint: You do *not* need to prove this by hand. Instead, use a property of random oracles that was introduced in the lecture and the canonical construction of a MAC (Theorem 4.6).

(c) Explain *briefly* the consequences of (a) and (b) for the soundness of the Random-Oracle model.

(a) Let h be the collision-resisitant hash function from which H is constructed by applying the Merkle-Damgard transform. We show that the MAC is not secure by constructing an adversary: We first query an arbitrary message m of length n and obtain

$$t = \mathrm{Mac}_k(m) = H(k||m) = h(h(0^n||k)||m)\,.$$

The adversary outputs $m' = m||t$ and $t' = h(t||t)$. Now it holds that

$$\begin{aligned}\mathrm{Mac}_k(m') &= H(k||m') \\ &= H(k||m||t) \\ &= h(h(h(0^n||k)||m)||t) \\ &= h(t||t) \\ &= t'\end{aligned}$$

It follows that the adversary wins with probability 1, which is certainly not negligible.

(b) It is known from the lecture that the function $F_k(m) = H(k||m)$ is a PRF if H is modeled as a random oracle. Therefore, applying Theorem 4.6, the MAC is secure.

(c) The results above illustrate that a proof of security in the Random-Oracle model does not guarantee security for every practical instance. It is mere evidence. However, a proof of security in the Random-Oracle model is better than no proof at all.

53 (MAC CBC) (SAARLAND)

Exercise No.4 (MACs) *(6+1+3 points)*

(a) Let F_k be a PRF and let $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ be a MAC defined as follows:

- Gen outputs a uniform $k \in \{0,1\}^n$.
- $\mathsf{Mac}_k(m) := (F_k(m)||F_k(m))$
- $\mathsf{Vrfy}_k(m, t)$ outputs 1 if and only if $t = (F_k(m)||F_k(m))$

Prove that Π is a secure MAC.
Remark: You may use *without proof* that the following MAC $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ is secure:

- Gen' outputs a uniform $f \in \mathsf{func}_n$.
- $\mathsf{Mac}'_f(m) := (f(m)||f(m))$
- $\mathsf{Vrfy}'_f(m, t)$ outputs 1 if and only if $t = (f(m)||f(m))$

(b) Is Π from (a) also strongly secure? Explain your answer.

(c) Prove or disprove: If $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ is a strongly secure MAC then Mac is a pseudorandom function.

Solution No.4 (MACs)

a) We do a proof by reduction. Assuming Π is not secure yields the existence of an adversary $\mathcal{A}$ such that

$$\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1] > \frac{1}{p(n)}$$

for a polynomial p. We construct a distinguisher D as follows: On input 1^n and with access to oracle $\mathcal{O}$, D simulates $\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n)$. Whenever $\mathcal{A}$ queries $\mathsf{Mac}_k(m)$, D returns $\mathcal{O}(m)||\mathcal{O}(m)$. Finally D outputs 1 if $\mathcal{A}$ wins and 0 otherwise. Then it holds that $\Pr[D^{F_k(\cdot)}(1^n) = 1] = \Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1] > \frac{1}{p(n)}$ and $\Pr[D^{f(\cdot)}(1^n) = 1] = \Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi'}(n) = 1] \leq \mathsf{negl}(n)$. Therefore

$$\left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \geq \frac{1}{p(n)} - \mathsf{negl}(n)$$

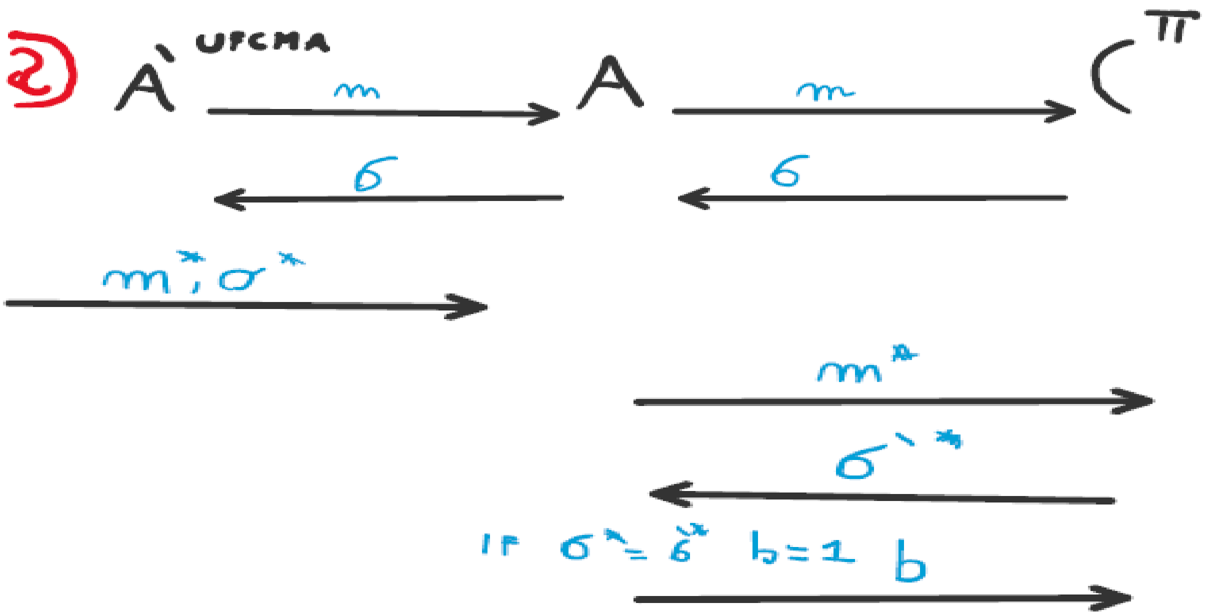
which is not negligible.

b) Yes, because it uses canonical verification.

c) This does not hold. Π from (a) is strongly secure but $m \to F_k(m)||F_k(m)$ is not a PRF. To prove this, let D be a distinguisher that, given 1^n as input and oracle access to $\mathcal{O}$, computes $t = \mathcal{O}(0^n)$ and checks whether $t_1 \dots t_{\frac{|t|}{2}} = t_{\frac{|t|}{2}+1} \dots t_{|t|}$. Then it holds that

$$\left|\Pr[D^{\mathsf{Mac}(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| = 1 - 2^{-n}$$

which is not negligible.



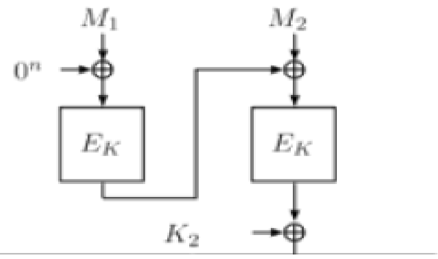
b) Yes same GAME of point A

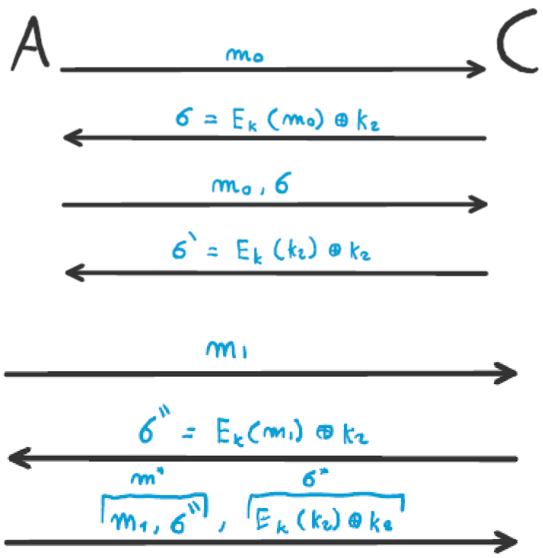
54 (MAC CBC) (SAARLAND)

Question 10 – CBC-MAC' (4 Points)
Assume that $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is a secure block cipher and $K_1 \in \{0,1\}^k$ a secret key. Let $K_2 \in \{0,1\}^n$ be a second independent key. We consider the following variant of CBC-MAC, called CBC-MAC'. Given an m-block message $M = (M_1, \ldots, M_m)$, CBC-MAC' computes an authentication tag as follows:

$$\begin{aligned} C_0 &= 0^n, \\ C_i &= E_{K_1}(C_{i-1} \oplus M_i), \qquad \text{for } 1 \leq i \leq m, \\ \text{CBC-MAC}'_{K_1,K_2}(M) &:= C_m \oplus K_2, \end{aligned}$$

Show, how to efficiently predict the tag for a message with CBC-MAC'. You may ask for the tags of at most three (other) messages before. Note that you can vary the lengths of your chosen messages.





55 (NUMBER THEORY) (SAARLAND)

This question concerns the group $\mathbb{Z}_p^*$, where $p = 2q + 1$ with p, q prime. Let $g \in \mathbb{Z}_p^*$ be a generator.

(a) Let $h \in \mathbb{Z}_p^*$. Show that h is a quadratic residue modulo p if and only if $h^q = 1 \bmod p$. (Hint: it is relatively easy to show that h is a quadratic residue implies $h^q = 1 \bmod p$. For the other direction, use the fact that $\mathbb{Z}_p^*$ is cyclic.)

(b) The discrete logarithm problem is assumed to be hard in $\mathbb{Z}_p^*$, meaning that the function $\mathsf{exp} : \mathbb{Z}_{p-1} \longrightarrow \mathbb{Z}_p^*$ defined by $\mathsf{exp}(x) = g^x \bmod p$ is assumed to be one-way. Let $\mathsf{lsb}(x)$ denote the least-significant bit of x. Prove that lsb is not a hard-core predicate for exp.

(c) Prove that the decisional Diffie-Hellman assumption does not hold in $\mathbb{Z}_p^*$.

(d) **(Extra credit:)** The decisional Diffie-Hellman assumption is believed to hold in the subgroup $\mathbb{G} < \mathbb{Z}_p^*$ of quadratic residues modulo p. Show that this implies that the *computational* Diffie-Hellman assumption holds in $\mathbb{Z}_p^*$. (Note: this question requires a small bit of group theory not covered in class. Specifically, use the fact that $\mathbb{Z}_p^* \cong \mathbb{Z}_q \times \mathbb{Z}_2$.)

2) $h \in QR \Rightarrow h^q = 1 \bmod p$

By definition $h \in QR_p$ IFF $h^{\frac{p-1}{2}} = 1 \bmod p$

In our case $q = \frac{p-1}{2}$ ✓

$h^q = 1 \bmod p \not\Rightarrow h \in QR$

If $h^{2 \cdot z'}$ for some z' then:

$\underbrace{h^{\frac{p-1}{2}}}_{q} = \gamma^{z'(p-1)} = 1 \bmod p$ due the

fact it is a cyclic group.

2. Consider group $\mathcal{Z}_{23}^*$.

(a) What are the orders of 5 and 11?

Answer.

$$5^1 = 5 \neq 1 \bmod 23,\ 5^2 = 2 \neq 1 \bmod 23, \ldots, 5^{22} = 1 \bmod 23, ord(5) = 22. \quad (4)$$

$$11^1 = 11 \neq 1 \bmod 23,\ 11^2 = 6 \neq 1 \bmod 23, \ldots, 11^{22} = 1 \bmod 23, ord(11) = 22. \quad (5)$$

□

(b) Find all generators for $\mathcal{Z}_{23}^*$.

Answer. $\mathcal{Z}_{23}^* = \{1, 2, \ldots, 22\}$. The generators in $\mathcal{Z}_{23}^*$ are $\{5, 7, 10, 11, 14, 15, 17, 19, 20, 21\}$

□

(c) Find subgroups of orders 2 and 11 if they exist.

Answer.

Subgroups of order 11=$< 2 >, < 3 >, < 4 >, < 6 >, < 8 >, < 9 >, < 12 >, < 13 >, < 16 >, < 18 >$

Subgroups of order 2=$< 22 >$ □

(d) Find $QR23$.

Answer.

$QR23 = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$ □

56 (OFW) (CORNELIOUS)

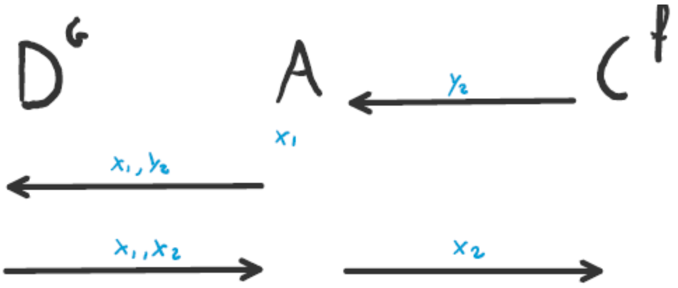
CORNELIOUS

Exercise 8.4 (One-way functions and hard-core predicates, 2+2 Points) Let f be a length-preserving one-way function. Define the function $g(x_1, x_2) = (x_1, f(x_2))$, where $|x_1| = |x_2|$.

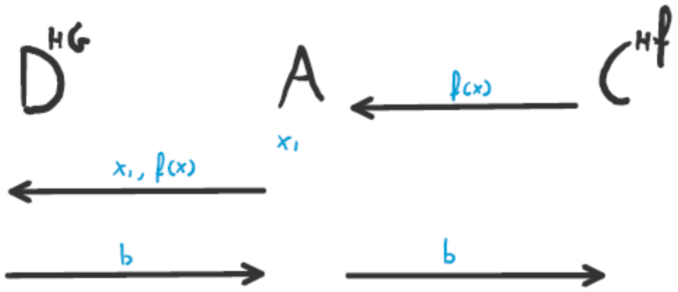
(a) Show that g is one-way as well.

(b) Show that if f has a hard-core predicate, then so has g.

a)



b)

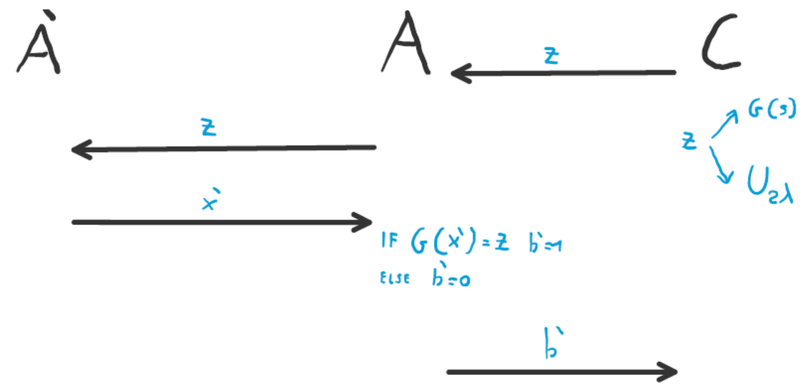


57 (PRG) (CORNELIOUS)

KATZ

Let G be a length-doubling pseudorandom generator. Prove that G is a one-way function.

$$G : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda} \Rightarrow \text{OWF}$$



There is a BAD case when z is U and A` chooses a x` s.t. G(x`) = z.

The $P[\text{BAD}] = \frac{1}{2^{2\lambda}} \cdot 2^{\lambda} = \frac{1}{2^{\lambda}} \in negl(\lambda)$

58 (PRG) (CORNELIOUS)

Let G be a pseudorandom generator and define $G'(s)$ to be the output of G truncated to n bits (where $n = |s|$). Prove that the function $F_k(x) = G'(k) \oplus x$ is not pseudorandom.

Answer. Consider a distinguisher D that is given oracle access either to a function f that is truly random, or F_k as defined above for a uniform choice of k. The distinguisher D queries the oracle with (any) two distinct inputs x and x' of length n, and gets the responses y and y'. D then outputs 1 if and only if $x \oplus x' = y \oplus y'$.

When D is given oracle access to F_k, we have that

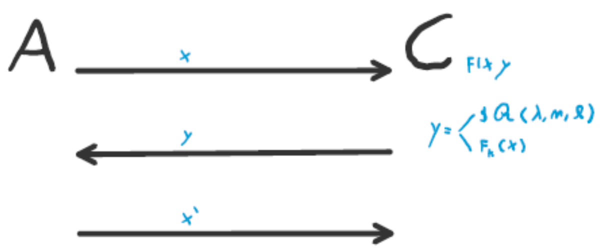
$$y \oplus y' = F_k(x) \oplus F_k(x') = G'(k) \oplus x \oplus G'(k) \oplus x' = x \oplus x',$$

regardless of the exact choice of k. In contrast, for a random function f, the probability that $y \oplus y' = x \oplus x'$ is exactly 2^{-n}. Therefore, $\Pr[D^{f(\cdot)}(1^n) = 1] = 2^{-n}$. So $\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1] \geq 1 - \frac{1}{2^n}$.

G = PRG

G`(S) = TRUNCATE G TO |S| BITS

$F_K(x)$ = G`(x) ⊕ x IS NOT PRF.



$$\xrightarrow{\quad x' \quad}$$

$$\xleftarrow{\quad y' \quad} \qquad y' = \begin{cases} \$\,\mathcal{Q}(\lambda, m, \ell) \\ F_k(x) \end{cases}$$

$$y \oplus y' = F_k(x) \oplus F_k(x') = \cancel{G'(x)} \oplus x \oplus \cancel{G'(x)} \oplus x' = x \oplus x'$$

$$\xrightarrow{\quad b' \quad}$$

59 (PRG) (CORNELIOUS)

PRG OR V

Question 2 – PRG (4 Points)
Let $G : \{0,1\}^k \rightarrow \{0,1\}^n$ be a secure PRG and $K_1, K_2 \in \{0,1\}^k$ independent secret seeds. Define $G'(K_1, K_2) := G(K_1) \vee G(K_2)$ where $\vee$ denotes the bit-wise logical OR. For each $n \in \{1,2,3\}$, describe an efficient PRG adversary $\mathcal{A}$ and describe the PRG advantage of $\mathcal{A}$ on G', $\mathbf{Adv}^{\mathrm{PRG}}_{G'}(\mathcal{A})$. You may assume that each output $X \in \{0,1\}^n$ from $G(K)$ has probability exactly $1/2^n$ over all seeds $K \in \{0,1\}^k$.

- FOR $n=1$ The advantage is $\frac{3}{4}$

$0 \vee 0$	0
$0 \vee 1$	1
$1 \vee 0$	1
$1 \vee 1$	1

- FOR $n=2$ " " is $\frac{5}{12}$

60 (PRIVATE KEY) (KATZ Exercise 3.18)

3.18 Let F be a pseudorandom permutation, and define a fixed-length encryption scheme (Enc, Dec) as follows: On input $m \in \{0,1\}^{n/2}$ and key $k \in \{0,1\}^n$, algorithm Enc chooses a uniform string $r \in \{0,1\}^{n/2}$ of length $n/2$ and computes $c := F_k(r\|m)$.

Show how to decrypt, and prove that this scheme is CPA-secure for messages of length $n/2$. (If you are looking for a real challenge, prove that this scheme is CCA-secure if F is a *strong* pseudorandom permutation.)

$\mathcal{M} = \{0,1\}^{n/2}$

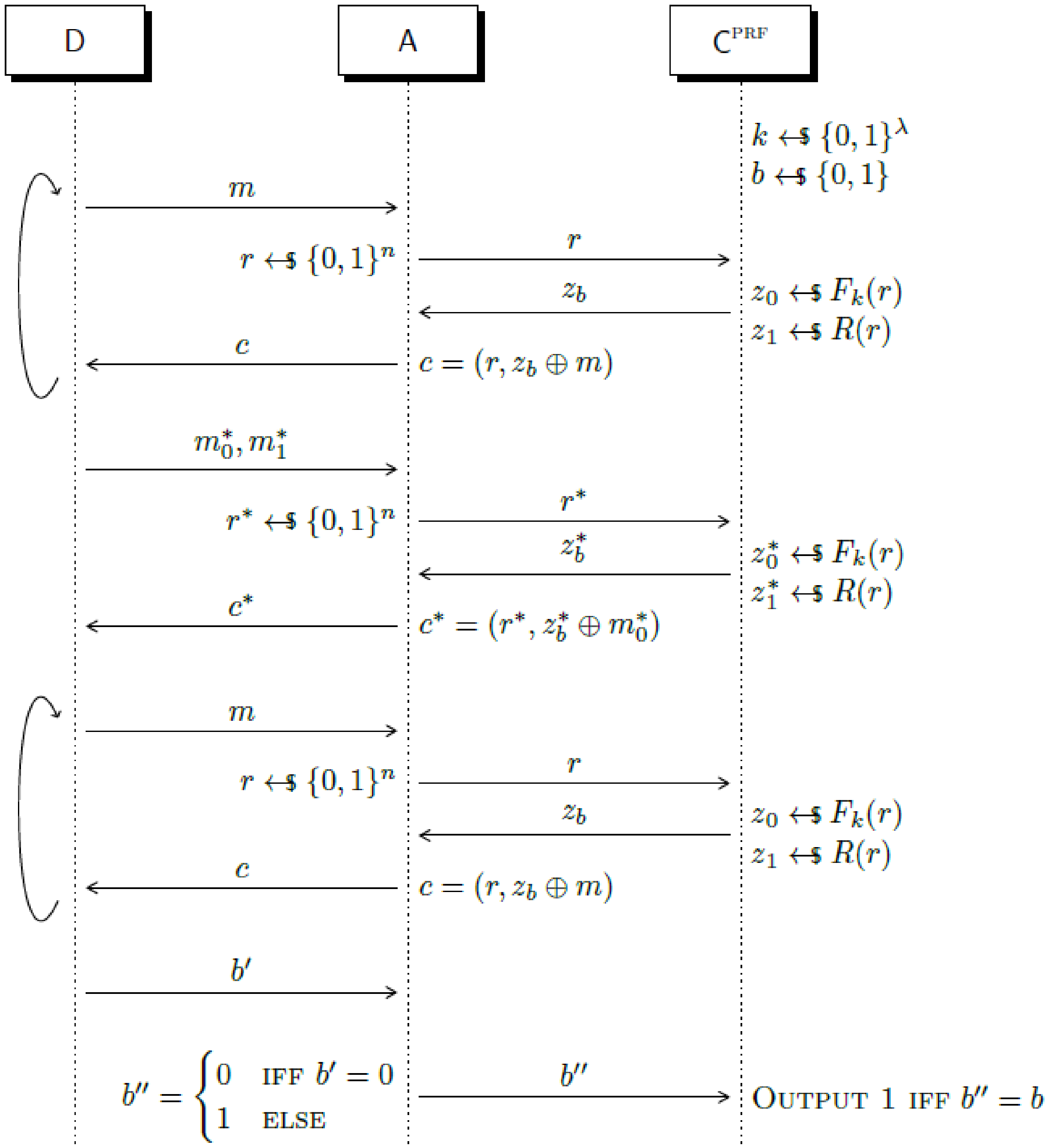
$\mathcal{K} = \{0,1\}^{n}$

RANDOM STRING $\{0,1\}^{n/2}$

$\text{ENC}_k(m) = F_k(r \| m)$

HOW TO DECRYPT?

$\text{DEC}(c) = F_k^{-1}(c)$ and takes the last $n/2$ BITS



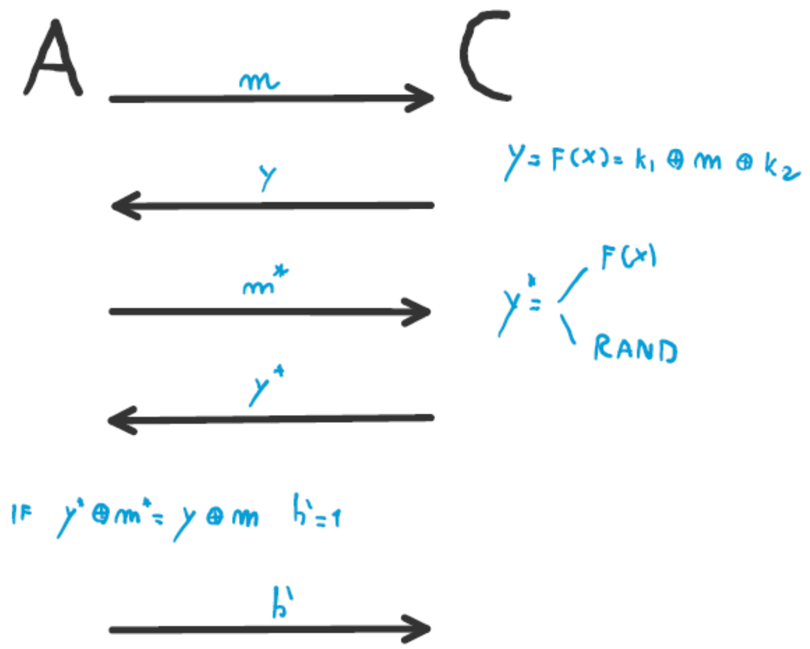
Where replace xor (⊕) with concatenate (||) and Fk(r) with Fk(r ||m)

61 (PRF) ()

Exercise No.3 (PRFs) *(8 points)*
Consider the following keyed function F: For security parameter n, the key is a pair (k_1, k_2) where $k_1, k_2 \in \{0,1\}^n$. Define $F_{(k_1,k_2)} : \{0,1\}^n \to \{0,1\}^n$ by

$$F_{(k_1,k_2)}(x) := k_1 \oplus x \oplus k_2$$

Show that F is not a PRF.

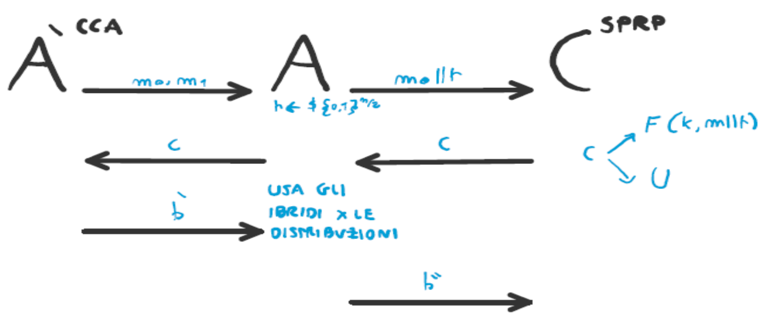


62 (PRF) ()

Let F be a strong pseudo random permutation and define a fixed-length encryption scheme $(\mathrm{Enc}, \mathrm{Dec})$ as follows:

On input $m \in \{0,1\}^{\frac{n}{2}}$ and key $k \in \{0,1\}^n$, algorithm Enc chooses a uniform string $r \in \{0,1\}^{\frac{n}{2}}$ of length $r/2$ and computes $c := \mathrm{F}(k, r \| m)$. Prove that this scheme is CCA-secure.

How can I show that strong Pseudo random permutation implies CCA security? I don't know how to do the reduction.



63 (PRF) ()

WEAK PSEUDORANDOM

For any function $g : \{0,1\}^n \to \{0,1\}^n$, define $g^{\$}(.)$ to be a probabilistic oracle that, on input 1^n, choose uniform $r \in \{0,1\}^n$ and return $(r, g(r))$. A keyed function F is a *weak pseudorandom function* if for all PPT algorithm D, there exists a negligible function **negl** such that:

$$\left|\Pr[D^{F_k^{\$}(.)}(1^n) = 1] - \Pr[D^{f^{\$}(.)}(1^n) = 1]\right| \leq negl(n) \tag{1}$$

where $k \in \{0,1\}^n$ and $f \in Func_n$ and chosen uniformly.

1. Let F' be a pseudorandom function, and define

$$\mathrm{F_k(x)} \overset{\mathrm{def}}{=} \begin{cases} \mathrm{F'_k(x)} & \text{if x is even} \\ \mathrm{F'_k(x+1)} & \text{if x is odd} \end{cases} \tag{2}$$

   Prove that F is weakly pseudorandom.

2. Is CTR-mode encryption using a weak pseudorandom function necessary CPA-secure? Does it necessarily have indistinguishable encryptions in the presence of an eavesdropper? Prove your answers.

3. Prove that the following construction is CPA-secure if F is a weak pseudorandom function.

Construction: Let F be a pseudorandom function. Define a private-key encryption scheme for messages of length n as follows:

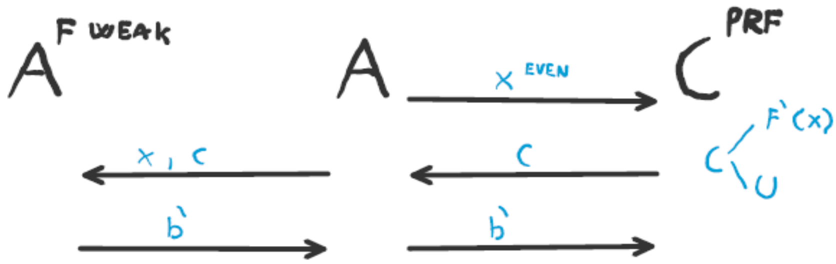
- Gen: on input 1^n, choose uniform $k \in \{0,1\}^n$ and output it.
- Enc: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, choose uniform $r \in \{0,1\}^n$ and output the ciphertext:

$$c := \langle r, F_k(r) \oplus m \rangle \tag{3}$$

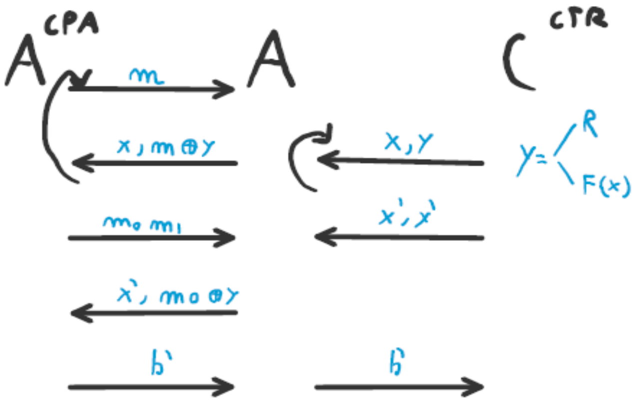
- Dec: on input a $k \in \{0,1\}^n$ and a ciphertext $c = \langle r, s \rangle$, output the plaintext message

$$m := F_k(r) \oplus s \tag{4}$$

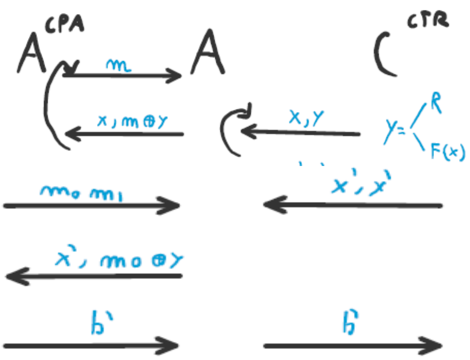
1) F' PRF → F WEAK-PRF



2) WEAK CTR → CPA

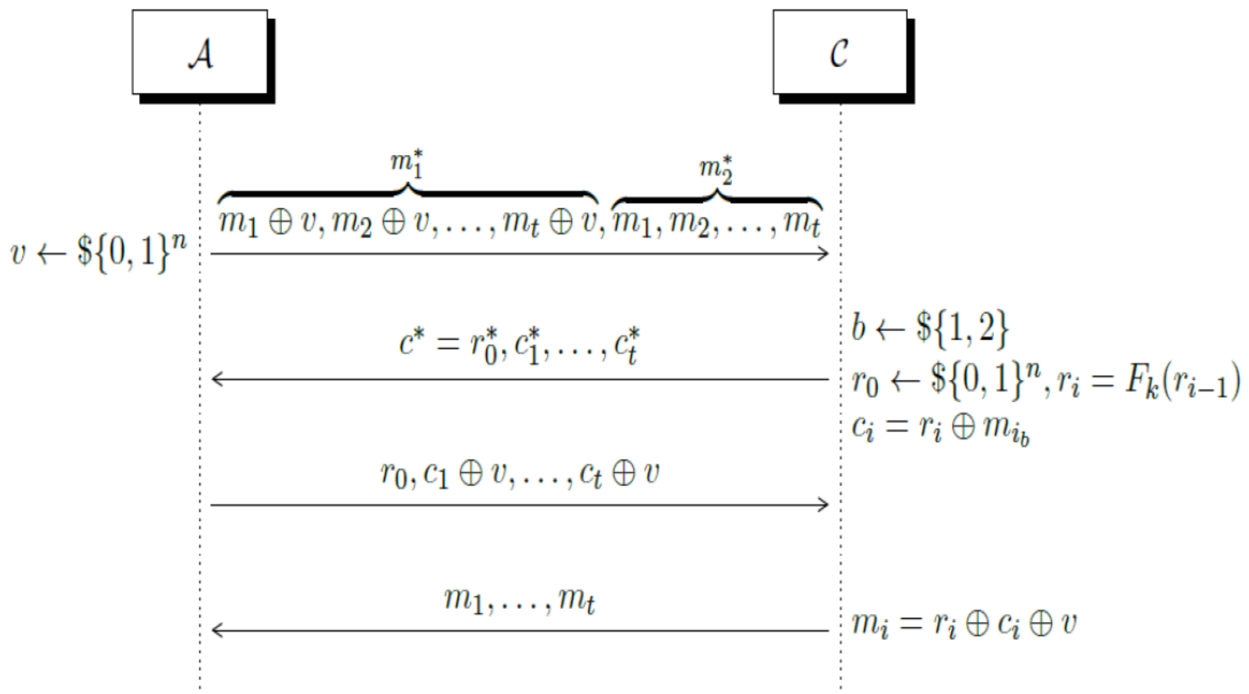


3) WEAK F → CPA



(b) Consider the following mode of operation (a.k.a. the Output Feedback mod
Given a message $m = (m_1, \dots, m_t)$ consisting of t blocks $m_i \in \{0,1\}^r$
random key $k \in \{0,1\}^\lambda$, the ciphertext is $c = (r_0, c_1, \dots, c_t)$ where $r_0 \leftarrow$
$r_i = F_k(r_{i-1})$, and $c_i = r_i \oplus m_i$ for all $i \in [t]$, and $F : \{0,1\}^\lambda \times \{0,1\}^n \to \{0$
secure pseudorandom function.

We can imagine the following game to show a counterexample for the OFB mode as not CCA secure.

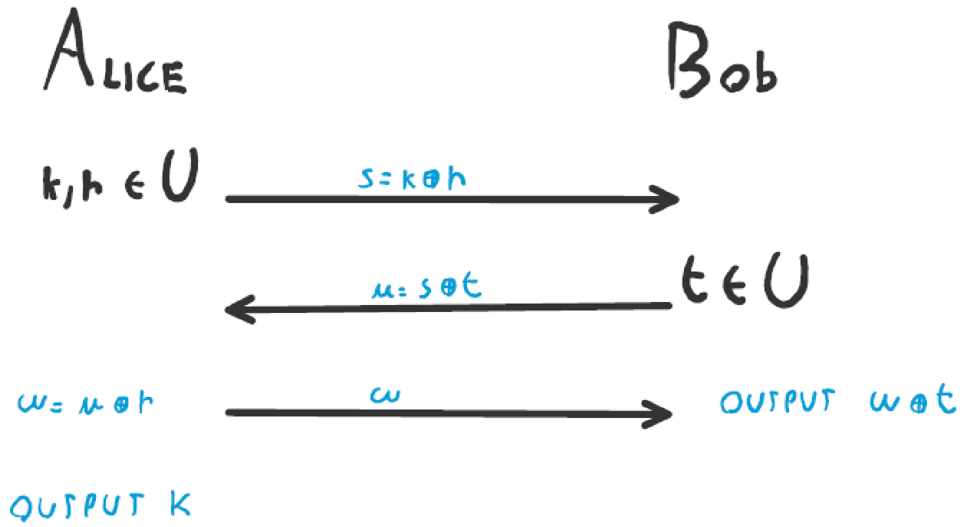


The possibilities are:
if b = 1, the received m_i contains $r_i \oplus c_i \oplus$ v = $r_i \oplus (r_i \oplus m_i \oplus$ v) $\oplus$ v = m_i
if b = 0, the received m_i contains $r_i \oplus c_i \oplus$ v = $r_i \oplus (r_i \oplus m_i) \oplus$ v = $m_i \oplus$ v
and of course this allow A to understand which message was used for encryption

10.4 Consider the following key-exchange protocol:

(a) Alice chooses uniform $k, r \in \{0,1\}^n$, and sends $s := k \oplus r$ to Bob.
(b) Bob chooses uniform $t \in \{0,1\}^n$, and sends $u := s \oplus t$ to Alice.
(c) Alice computes $w := u \oplus r$ and sends w to Bob.
(d) Alice outputs k and Bob outputs $w \oplus t$.

Show that Alice and Bob output the same key. Analyze the security of the scheme (i.e., either prove its security or show a concrete attack).



IT IS NOT SECURE because if a A intercepts s, u and w can discover K

66 (PUBLIC KEY ENCRYPTION) (KATZ 11.6)

11.6 Consider the following public-key encryption scheme. The public key is $(\mathbb{G}, q, g, h)$ and the private key is x, generated exactly as in the El Gamal encryption scheme. In order to encrypt a bit b, the sender does the following:

(a) If $b = 0$ then choose a uniform $y \in \mathbb{Z}_q$ and compute $c_1 := g^y$ and $c_2 := h^y$. The ciphertext is $\langle c_1, c_2 \rangle$.

(b) If $b = 1$ then choose independent uniform $y, z \in \mathbb{Z}_q$, compute $c_1 := g^y$ and $c_2 := g^z$, and set the ciphertext equal to $\langle c_1, c_2 \rangle$.

Show that it is possible to decrypt efficiently given knowledge of x. Prove that this encryption scheme is CPA-secure if the decisional Diffie–Hellman problem is hard relative to $\mathcal{G}$.

PUBLICK KEY : $(\mathbb{G}, q, g, h)$

PRIVATE KEY : EL GAMAL

IF :

- $b=0$, chooses $y \in \mathbb{Z}_q$ and compute
  $c_1 = g^y$ $c_2 = h^y$

- $b=1$, chooses indipendly $y, z \in \mathbb{Z}_q$
  $c_1 = g^y$, $c_2 = g^z$

SHOW IT IS POSSIBLE TO DECRYPT EFFICENTLY

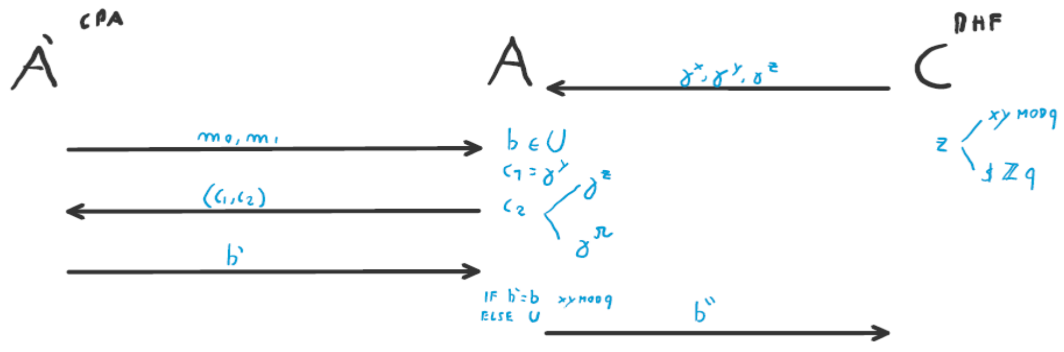
HAVING THE KNOWLEDGE x .

IF the result of DEC is 1 → $b=0$; ELSE $b=1$

DEC: $\frac{c^2}{c_1^x}$

IS THIS SCHEME CPA ?

DHF $\Rightarrow$ CPA



THERE IS A BAD EVENT ; WHEN $z \leftarrow \$ \mathbb{Z}$ and A' responses a true b' .

$$PR[BAD] = \frac{|\mathbb{Z}_q|}{|q|} \in NEGL$$

67 (PUBLIC KEY ENCRYPTION)

Problem 2 Let g be an element of prime order q in $\mathbb{Z}_p^*$. Suppose there exists an efficient algorithm that computes the Diffie-Hellman function base g. More precisely, there is an efficient algorithm $\mathcal{A}$ such that $\mathcal{A}(g^x, g^y) = g^{xy}$ for all $x, y \in \{1, \ldots, q\}$. Let $h = g^\alpha$ for some $\alpha \in \{1, \ldots, q-1\}$. Our goal is to show that there is an efficient algorithm $\mathcal{D}$ that is able to compute the Diffie-Hellman function base h, i.e. $\mathcal{D}(h, h^x, h^y) = h^{xy}$.

a. Show that there is an efficient algorithm $\mathcal{B}$ that given α is able to compute the Diffie-Hellman function base h, i.e. $\mathcal{B}(h, \alpha, h^x, h^y) = h^{xy}$.
Algorithm $\mathcal{B}$ may use algorithm $\mathcal{A}$ as a subroutine.

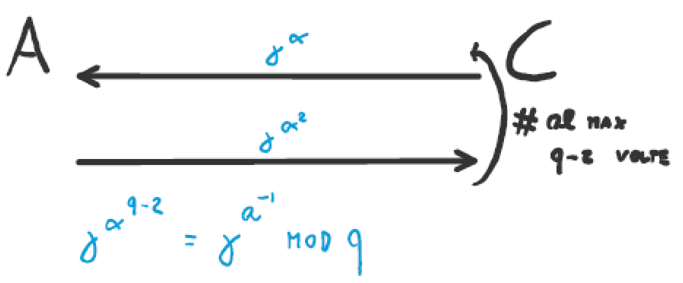
b. Show that there is an efficient algorithm $\mathcal{C}$ such that $\mathcal{C}(g, g^\alpha) = g^{(\alpha^{-1})}$ for all $\alpha \in \{1, \ldots, q-1\}$.
Hint: Recall that $\alpha^{q-2} = \alpha^{-1} \bmod q$. Algorithm $\mathcal{C}$ uses algorithm $\mathcal{A}$ as a subroutine.

c. Show that there is an efficient algorithm $\mathcal{D}$ that given h is able to compute the Diffie-Hellman function base h, i.e. $\mathcal{D}(h, h^x, h^y) = h^{xy}$. Use part (b).

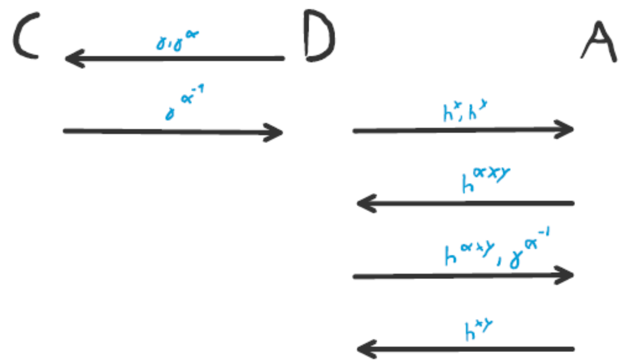
a) $B(h, \alpha, \underset{g^{\alpha x}}{h^x}, \underset{g^{\alpha y}}{h^y})$ goal h^{xy}

B uses A to calculate $g^{\alpha x \alpha y}$
at this point can do $\sqrt[\alpha]{g^{\alpha x \alpha y}} = g^{\alpha x y} = h^{xy}$ ✓

b)



c)



68 (PUBLIC KEY ENCRYPTION)

4.1 The (In)security of the ElGamal Public-Key Cryptosystem

Goal: *The ElGamal public-key cryptosystem uses the Diffie-Hellmann protocol to build a PKE scheme. We prove that this scheme is IND-CPA secure, but not IND-CCA secure.*

The Diffie-Hellman protocol can be used as a PKE scheme, as discussed in the lecture. In this task we consider the security of one such scheme, the well known ElGamal public-key cryptosystem. Let the group G and the generator g be fixed and the order q be publicly known. The ElGamal scheme then works as follows:

Key generation: Choose x_B uniformly at random from $\mathbb{Z}_q$. The secret key is x_B, the public key is $y_B := g^{x_B}$.

Encryption: On input a message $m \in G$, choose $x \in \mathbb{Z}_q$ uniformly at random. The ciphertext for a message $m \in G$ is the pair $(g^x, m \cdot y_B^x)$.

a) Describe the decryption of the ElGamal scheme, i.e., show how to obtain the message m given $(g^x, m \cdot y_B^x)$ and the secret key x_B.

b) Show that the ElGamal cryptosystem is IND-CPA secure under the DDH-assumption. More precisely, show that if there is an efficient distinguisher D that has advantage α in the IND-CPA game for public-key encryption, then there is an efficient distinguisher D' that given $(A, B, C) \in G^3$ has advantage $\alpha/2$ in distinguishing the case where A, B, C are independent and uniform in G from the case $A = g^a$, $B = g^b$, and $C = g^{ab}$ for uniform and independent $a, b \in \mathbb{Z}_q$.

c) Prove that the ElGamal scheme is not IND-CCA secure, that is, present an efficient attacker with non-negligible advantage in the IND-CCA game for public-key encryption.

a) $$DEC(g^x, m \cdot y_B^x) = \frac{c_2}{c_1^x} = m \cdot y_B^x \cdot (g^x)^{-x_B} = m \cdot g^{x_B \cdot x} \cdot g^{-x x_B} = m$$

b) FATTO IN CLASSE

c) Given pk, $(c_1, c_2) \to m$, $(c_1', c_2') \to m'$ $\Rightarrow (c_1 \cdot c_1', c_2 \cdot c_2') = (g^{x+x'}, h^{x+x'}(m, m'))$

69 (PUBLIC KEY ENCRYPTION)

Exercise No.5 (From public to private key) *(7 points)*
Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Construct a *private*-key encryption scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows:

- $\mathsf{Gen}'(1^n) := \mathsf{Gen}(1^n)$, that is, the single private key k of Π' is the pair (sk, pk) output by Gen.
- $\mathsf{Enc}'_{(pk,sk)}(m) := (\mathsf{Enc}_{pk}(m), \mathsf{Enc}_{pk}(m))$, that is, encryption of a message m produces a ciphertext (c_1, c_2), where for c_1 and c_2, encryption is performed independently as in Π, using only the part of k corresponding to pk.
- $\mathsf{Dec}'_{(pk,sk)}(c_1, c_2)$ is defined as follows: Let $m_0 := \mathsf{Dec}_{sk}(c_1), m_1 := \mathsf{Dec}_{sk}(c_2)$. Then, $\mathsf{Dec}'_{(pk,sk)}(c_1, c_2)$ is defined as $\perp$ if $m_0 \neq m_1$ (i.e., decryption failed), and as m_0 otherwise. That is, decryption is performed on both parts of the ciphertext as in Π, using only the second part sk of k. If both parts yield the same message, the algorithm outputs this message, and otherwise outputs an error.

Show that Π' is *not* CCA-secure.

A C

$m_0^* \neq m_1^*$

$A \xrightarrow{m_0^*,\ m_1^*} C$

$A \xleftarrow{c^* = (c_0, c_1)} C$

$A \xrightarrow{(c_1, c_0)} C$

$A \xleftarrow{m} C$

IF $m = m_0^*$ $b' = 0$ ELSE $b' = 1$

$A \xrightarrow{b'} C$

70 (PUBLIC KEY ENCRYPTION)

TEXTBOOK ELGAMAL

Question 6 – Textbook ElGamal Encryption (3 Points)

Let $\mathbb{Z}_p^*$ be a cyclic group for some prime p. Let g be a generator in some subgroup of order q in $\mathbb{Z}_p^*$. Alice' auction house chooses a secret key $a \xleftarrow{\$} \mathbb{Z}_p$ and publishes as public key (g, p, A) with $A \equiv g^a \bmod p$. Bob chooses $b \xleftarrow{\$} \mathbb{Z}_p$ and encrypts his bid m as

$$B \equiv g^b \bmod p, \qquad C \equiv m \cdot A^b \bmod p,$$

and sends the ciphertext (B, C) to Alice. Show that an adversary Eve can intercept and replace Bob's ciphertext with a valid ciphertext (B', C') of a higher bid m'. This shows that textbook ElGamal does not provide chosen-ciphertext security.

$$A \xleftarrow{\; C = (\overbrace{\gamma^b}^{C_1},\ \overbrace{m \cdot (\gamma^a)^b}^{C_2}) \;}$$

CON L'ALGORITMO VISTO
L'ALTRO IERI MI CALCOLO

$\gamma^{b^{-1}} \quad \gamma^{a^{-1}}$

$C_2 \cdot \gamma \; \gamma = m$

ORA POSSO FORGIARE m'

71 (PUBLIC KEY ENCRYPTION)

Problem 5. Threshold ElGamal. Let p be a prime and g a generator of $\mathbb{Z}_p^*$. Let x be an ElGamal private decryption key. To protect x one may wish to split x into three pieces and store each piece on a different server. An attacker who breaks into one of the servers should learn no information about x. Consider the following scheme: pick three random numbers x_1, x_2, x_3 in $[0, p-1]$ so that $x_1 + x_2 + x_3 = x \bmod p-1$. Store x_i on server i.

a. Very briefly explain how the ElGamal encryption algorithm works.

b. Suppose Alice wants to decrypt an ElGamal ciphertext C. Show that Alice can do the following: (1) she sends C to the three servers, (2) each server i performs a local computation (using x_i) and responds with M_i to Alice, and (3) given M_1, M_2, M_3 Alice decrypts C. Explain how server i computes M_i and how Alice combines M_1, M_2, M_3 to obtain the plaintext M.

c. To provide fault tolerance, show how the key x can be shared among the three servers so that any two of the three can be used to decrypt C as in part (b). You may store multiple x_i's on each server. An attacker who breaks into one of the servers should learn no information about x. As in part (b), your solution should not reconstruct the key x and there should be no interaction between the servers.

b) $C_{TOT} = \underbrace{x - x_1}_{M_1} \cdot \underbrace{x - x_2}_{M_2} \,/\, \underbrace{x + x_3}_{M_3} \Rightarrow \dfrac{M_1 \cdot M_2}{M_3}$

$$\frac{C_2}{C_1^{x_1}} \qquad \frac{C_2}{C_1^{x_2}} \qquad \frac{C_2}{C_1^{-x_3}}$$

c) all the servers must have a recovery value X_R s.t. $X_1 \oplus X_2 \oplus X_3 = X_R$

72 (SECURE KEY)

MARYLAND

Recall that in counter mode encryption, a message $m = m_1 \| \cdots \| m_\ell$ is encrypted under key k by choosing a random nonce r and outputting the ciphertext

$$r,\ m_1 \oplus F_k(r),\ \ldots,\ m_\ell \oplus F_k(r + \ell - 1).$$

Prove that counter mode encryption is secure in the sense of multi-message indistinguishability. For the purposes of this question, you may assume that the adversary always outputs two vectors containing $q(n)$ messages, and each message contains $\ell(n)$ message blocks.

$$A \xrightarrow{\ m = m_1 \| m_2\ } C$$

$$A \xleftarrow{\ c = r,\ F_k(r) \oplus m,\ F_k(r+1) \oplus m_2\ } C$$

$m^* = m_2 \| m_1$

$c^* = r,\ F_k(r) \oplus \cancel{m_1} \oplus \cancel{m_1} \oplus m_2,$

$F_k(r+1) \oplus \cancel{m_2} \oplus \cancel{m_2} \oplus m_1$

$$A \longrightarrow C$$

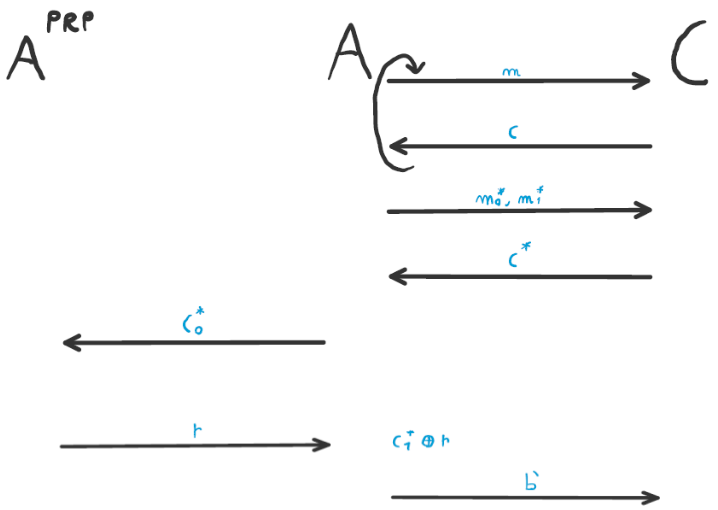
73 (SECURE KEY)

TREVISAN

Suppose that F is a pseudorandom permutation. Consider the following encryption scheme:

- $Enc(K, M)$: pick a random string r, output $(F_K(r), r \oplus M)$
- $Dec(K, C_0, C_1) := I_K(C_0) \oplus C_1$

Is it CPA secure?



74 (SECURE KEY)

ZURIGO

2.1 Block Ciphers in ECB and CBC Mode

Goal: *When should a symmetric encryption scheme be considered secure? We discuss how (not) to use block ciphers and introduce common modes of operation.*

Let $F\colon \{0,1\}^n \times \{0,1\}^\kappa \to \{0,1\}^n$ be a block cipher and $k \in \{0,1\}^\kappa$ a uniformly distributed key.

a) A straightforward technique to encrypt bit strings of length $\ell \cdot n$ for $\ell \geq 1$ is called *electronic codebook (ECB)* mode: Split $m \in \{0,1\}^{\ell n}$ into $m = m_1|\ldots|m_\ell$ with $m_1, \ldots, m_\ell \in \{0,1\}^n$ and compute $c := F(m_1, k)|\ldots|F(m_\ell, k)$.
Should this encryption scheme be considered secure if we assume that an attacker does not know anything about the encrypted messages?

b) Assume only messages of length n need to be encrypted. Describe an attack scenario in which it is insecure to encrypt a message $m \in \{0,1\}^n$ as $c := F(m, k)$.

c) A widely used alternative to ECB mode is the so-called *cipher-block chaining (CBC)* mode: To encrypt a message $m = m_1|\ldots|m_\ell$ with $m_1, \ldots, m_\ell \in \{0,1\}^n$, choose $c_0 \in \{0,1\}^n$ uniformly at random, compute $c_i := F(m_i \oplus c_{i-1}, k)$ for $i = 1, \ldots, \ell$, and let the ciphertext be $c := c_0|\ldots|c_\ell$. The value c_0 is called *initialization vector (IV)*.
How can a ciphertext be decrypted?

a) IT is NOT secure because 2 equal plain text → 2 difentes equal

b) IT IS DETERMINISTIC → NO CPA SECURE

75 (SIGNATURE SCHEMES)

3 Non-Adaptive and Weak Unforgeability 10 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. Answer the following questions. (For each question asking to prove/disprove an implication between two notions, if you think the implication holds you must show a reduction from one definition to the other; on the other hand, if you think the implication does not hold, you must exhibit a scheme which satisfies one definition but not the other.)

(a) Formally define a variant of universal unforgeability against chosen-message attacks (UF-CMA), where the adversary is given (together with the public key) $q \in \mathsf{poly}(\lambda)$ message/signature pairs $(m_i, \sigma_i)_{i \in [q]}$, where the messages $(m_i)_{i \in [q]}$ are chosen by the

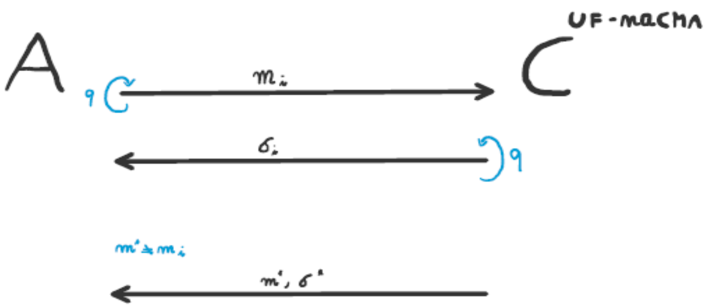
adversary non-adaptively (i.e., all the same time) before obtaining the public key. As in UF-CMA, in order to win the game, the adversary then needs to forge on a message m^* which is fresh (i.e., not equal to any of the messages $m_1, \dots, m_q$). Call the latter notion UF-naCMA.

(b) Formally define a variant of universal unforgeability against chosen-message attacks (UF-CMA), where the adversary is given (together with the public key) $q \in \mathsf{poly}(\lambda)$ message/signature pairs $(m_i, \sigma_i)_{i \in [q]}$, where each of the messages $(m_i)_{i \in [q]}$ is drawn uniformly at random from the message space. As in UF-CMA, in order to win the game, the adversary then needs to forge on a message m^* which is fresh (i.e., not equal to any of the messages $m_1, \dots, m_q$). Call the latter notion UF-RMA.

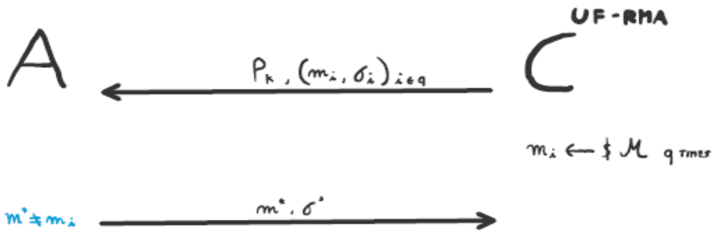
(c) Prove or disprove: UF-naCMA implies UF-RMA.

(d) Prove or disprove: UF-naCMA implies UF-CMA.

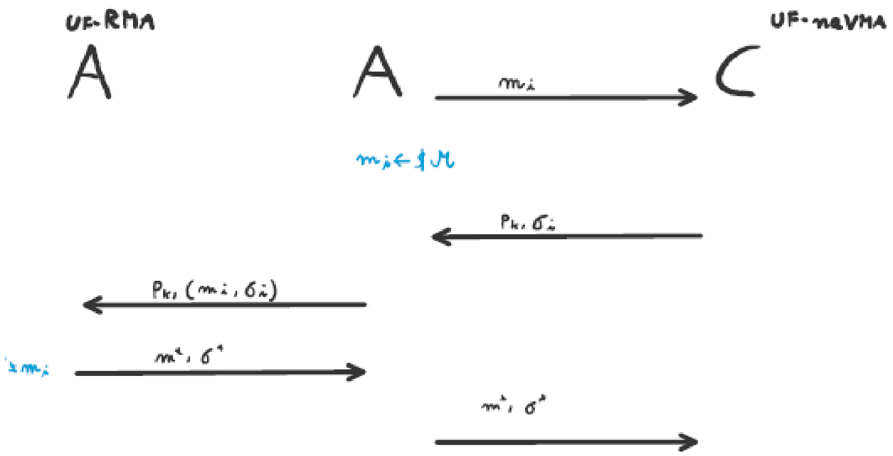
a) UF - naCMA



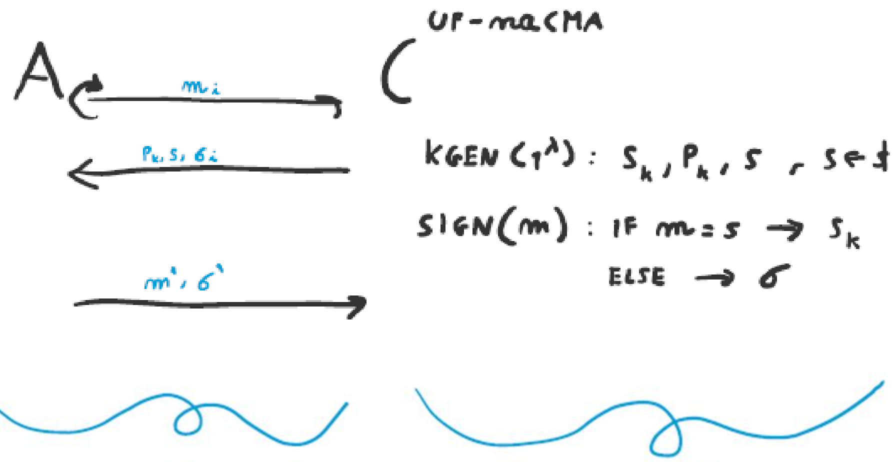
b) UF-RMA



C) UF-naCMA ⇒ UF-RMA



D) UF-CMA ⇏ UF-naCMA ?



THIS IS UF-naCMA
BUT NOT UF-CMA

THIS ISN'T UFCMA

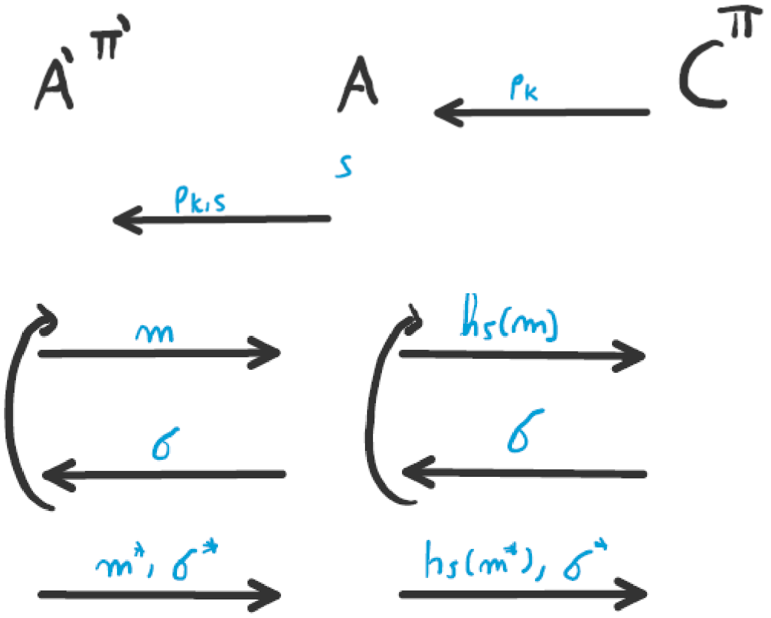
76 (SIGNATURE SCHEMES)

5 Signature Schemes **20 Points**

Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with message space $\mathcal{M} = \{0,1\}^\ell$, for some fixed $\ell \in \mathbb{N}$. Consider a family of hash functions $\mathcal{H} = \{h_s : \{0,1\}^* \to \{0,1\}^\ell\}_{s \in \{0,1\}^\lambda}$.

Define the following derived signature scheme $\Pi' = (\mathsf{KGen}', \mathsf{Sign}', \mathsf{Vrfy}')$: (i) Algorithm $\mathsf{KGen}'(1^\lambda)$ returns (pk', sk') such that $pk' = (pk, s)$ and $sk' = (sk, s)$, where $(pk, sk) \leftarrow\$ \mathsf{KGen}(1^\lambda)$ and $s \leftarrow\$ \{0,1\}^\lambda$; (ii) Algorithm $\mathsf{Sign}'(sk', m)$ takes a message $m \in \{0,1\}^*$ of arbitrary length, and outputs $\sigma \leftarrow\$ \mathsf{Sign}(sk, h_s(m))$; (iii) Algorithm $\mathsf{Vrfy}'(pk', m, \sigma)$ returns the same as $\mathsf{Vrfy}(pk, h_s(m), \sigma)$. Prove that if Π is UF-CMA and $\mathcal{H}$ is collision-resistant, then Π' is UF-CMA.

Π UFCMA $\wedge$ CHR $\rightarrow$ Π' UFCMA



76 (SIGNATURE SCHEMES)

WEAK UFCMA

Consider a variant of FDH-RSA signatures where the message is not hashed during signature generation. Namely, let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be the following signature scheme.

$\mathsf{KGen}(1^\lambda)$: Run $(N, e, d) \leftarrow\!\!\$\; \mathsf{GenModulus}(1^\lambda)$, and let $pk = (e, N)$ and $sk = (N, d)$.

$\mathsf{Sign}(sk, m)$: Output $\sigma = m^d \bmod N$.

$\mathsf{Vrfy}(pk, m, \sigma)$: Output 1 if and only if $\sigma^e \equiv m \bmod N$.

Answer the following questions:

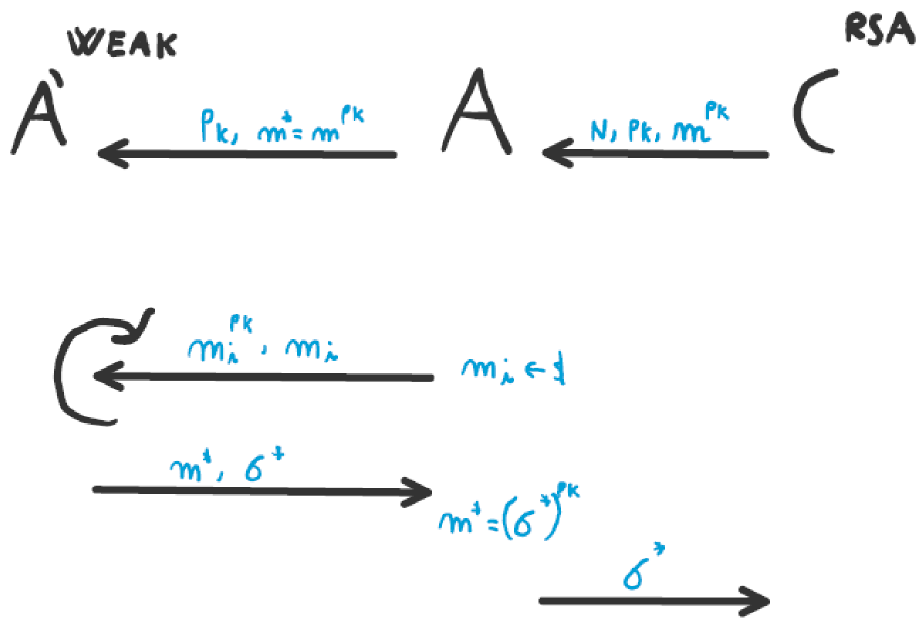
(a) Show that Π is not UF-CMA by exhibiting an attacker that produces a forgery on some message $m^* \in \mathbb{Z}_N^*$ without making any signature query.

(b) Show that Π is not UF-CMA by exhibiting an attacker that produces a forgery on any fixed message $m^* \in \mathbb{Z}_N^*$ by making a single signature query.

(c) Consider a weaker variant of UF-CMA in which the attacker receives (pk, m^*) at the beginning of the experiment, where the message m^* is uniformly random over $\mathbb{Z}_N^*$, and thus it has to forge on m^* after possibly seeing polynomially-many signatures σ_i on uniformly random messages $m_i \leftarrow\!\!\$\; \mathbb{Z}_N^*$ chosen by the challenger.

Formalize the above security notion, and prove that the signature scheme Π satisfies it under the RSA assumption.

c)



76 (UNIVERSAL HASHING)

5.10 Before HMAC, it was common to define a MAC for arbitrary-length messages by $\mathsf{Mac}_{s,k}(m) = H^s(k\|m)$ where H is a collision-resistant hash function.

(a) Show that this is never a secure MAC when H is constructed via the Merkle–Damgård transform. (Assume the hash key s is known to the attacker, and only k is kept secret.)

(b) Prove that this is a secure MAC if H is modeled as a random oracle.

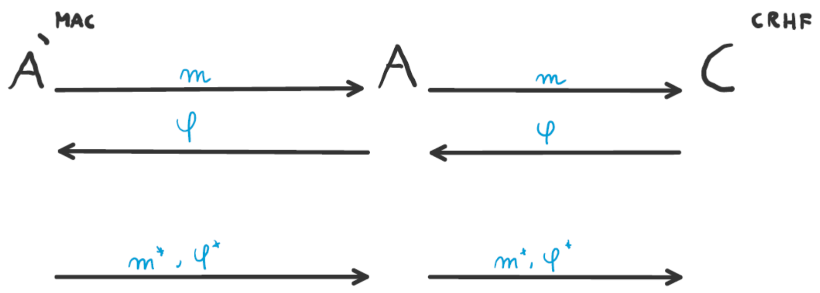
2) SHOW THAT IS NEVER SECURE MAC WHEN H IS COSTRUCTED BY MERKLE - DANGARD TRANSFORM

Assume $H_s(0^{n+1}) = 0^n$

$H'_s(x) = H'_s(0^m \| x)$

VARIABLE INPUT LENGHT

b) IS SECURE MAC IF H ORACLE



77 (UNIVERSAL HASHING)

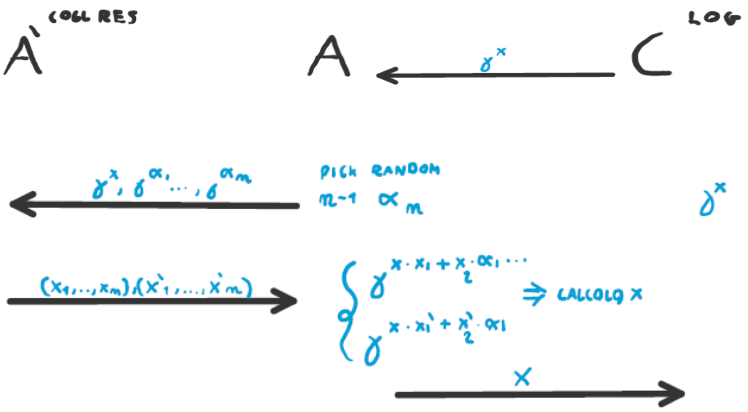
Problem 5. (incremental hashing) Let p be a prime and let $g \in \mathbb{Z}_p^*$ be an element of prime order q. We let G denote the group generated by g and we let I denote the set of integers $\{1, \ldots, q\}$. Fix n values $g_1, \ldots, g_n \in G$ and define the hash function $H : I^n \to G$ by

$$H(x_1, \ldots, x_n) = g_1^{x_1} g_2^{x_2} \cdots g_n^{x_n}$$

a. Show that H is collision resistant assuming discrete-log in G is intractable. That is, show that an attacker capable of finding a collision for H for a *random* $g_1, \ldots, g_n \in G$ can be used to compute discrete-log in G.
Hint: given a pair $g, h \in G$ your goal is to find an $\alpha \in \mathbb{Z}$ such that $g^\alpha = h$. Choose $g_1, \ldots, g_n \in G$ so that a collision on the resulting H will reveal α.

b. Let M be a message in I^n. Suppose user Alice already computed the hash of M, namely $H(m)$. Now, Alice changes only one coordinate of M to obtain a new message M'. Show that Alice can quickly compute $H(M')$ from $H(M)$ in time that is independent of the length of M.
You have just shown that after making a small change to a message there is no need to rehash the entire message. Collision resistant hash functions of this type are said to support *incremental hashing*.

2) DISCRETE LOG → COLLISION RES



b) A x_1, x_2, x_3

x_1, x_2', x_3

$H(m) = \gamma^{x_1}, \gamma^{x_2}, \gamma^{x_3}$

$H(m') = \gamma^{x_1}, \gamma^{x_2'}, \gamma^{x_3}$

A can do $H(m)\, \gamma^{x_2^{-1}}\, \gamma^{x_2'} = H(m')$

3 Strong Unforgeability 10 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. Answer the following questions.[1]

(a) Formally define a variant of universal unforgeability against chosen-message attacks (UF-CMA), where the adversary is allowed to forge even on messages m asked to the signing oracle, as long as the forged signature σ^* is fresh, i.e. $\sigma^* \neq \sigma$ where σ is the signature returned by the oracle. Call the latter notion, *strong* UF-CMA.

(b) Prove or disprove: Strong UF-CMA implies UF-CMA.

(c) Prove or disprove: UF-CMA implies strong UF-CMA.

Any MA scheme $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ that is $\mathrm{SUF\text{-}CMA}$ scheme is also $\mathrm{UF\text{-}CMA}$ scheme.

Why? Suppose A's **Tag** queries are $M_1, \ldots, M_q$, resulting in tags

$$T_1 \xleftarrow{\$} \mathcal{T}_K(M_1), \ldots, T_q \xleftarrow{\$} \mathcal{T}_K(M_q)$$

Now suppose A queries **Verify**(M, T). Then

$$M \notin \{M_1, \ldots, M_q\} \Rightarrow (M, T) \notin \{(M_1, T_1), \ldots, (M_q, T_q)\}$$

So if A wins in game $\mathrm{UFCMA}_{\mathcal{MA}}$ it also wins in game $\mathrm{SUFCMA}_{\mathcal{MA}}$.

Theorem: For any A,

$$\mathbf{Adv}^{\mathrm{uf\text{-}cma}}_{\mathcal{MA}}(A) \leq \mathbf{Adv}^{\mathrm{suf\text{-}cma}}_{\mathcal{MA}}(A)$$