CS6111: Foundations of Cryptography

Assignment 2

S Srinivas Saurab
CS16B039

Instructions

- Deadline is September 9.
- We encourage submissions by Latex. Paper is also accepted.

References

- Introduction to Cryptography - Delfs and Knebl
- A Graduate Course in Applied Cryptography - Boneh and Shoup (link)
- Introduction to Modern Cryptography - Katz and Lindell
- Handout 3

1 Number Theory

1. (2 points)

   Proposition 1. *Let $\mathbb{G}$ be a finite group, and $\mathbb{H} \subseteq \mathbb{G}$. Assume that $\mathbb{H}$ contains the identity element of $\mathbb{G}$, and that for all $a, b \in \mathbb{H}$ it holds that $ab \in \mathbb{H}$. Then $\mathbb{H}$ is a subgroup of $\mathbb{G}$.*

   Show that the above proposition does not necessarily hold when $\mathbb{G}$ is infinite. **Hint:** Consider the set $\{1\} \cup \{2, 4, 6, 8 \cdots\} \subset \mathbb{R}$.

   Solution: A set is said to be a group when it satisfies the all of these properties at once: closure, associativity, existence of identity, invertibility. We will not be proving Proposition-1 here, but provide a counter example, i.e., a set (subset of a infinite group) which satisfies the proposition but is not a subgroup.

   To arrive at such counter example, consider the set (given in the hint) as $\mathbb{H} = \{1\} \cup \{2, 4, 6, 8 \cdots\} \subset \mathbb{R}$. Clearly, the infinite parent group we are considering here is $\mathbb{R}$.

   Claim1: The given set H satisfies the condition that if $a, b \in \mathbb{H}$ then $ab \in \mathbb{H}$.
   Proof: This is because if a = 1 and b $\in \mathbb{H}$ then ab = b $\in \mathbb{H}$, similarly if b = 1 and a $\in \mathbb{H}$, ab = a $\in \mathbb{H}$. If

neither of them is 1, then we have a,b $\in \mathbb{H}$, a,b !=1. This means that they can be any two even numbers and thus their product must be even which comes under the set $\mathbb{H}$.

Claim2: The given set is not a group, therefore not a subgroup of $\mathbb{R}$, i.e., $\mathbb{H} \nleq \mathbb{R}$
Proof: To show that $\mathbb{H} \nleq \mathbb{R}$, it is enough to show that $\mathbb{H}$ is not a group. To this end, we will show that the invertibility is not satisfied by $\mathbb{H}$. Clearly, we can see that 2 does not have an inverse im $\mathbb{H}$, as the operator chosen is the regular real-multiplication.

Hence, $\mathbb{H} \nleq \mathbb{R}$ and thus the proposition does not hold for the subsets of a infinite group.

2. (2 points) Let $\mathbb{G}$ be a finite group and $g \in \mathbb{G}$. Show that $\langle g \rangle = \{g^i \mid i \geq 0\}$ is a subgroup of $\mathbb{G}$. Is the set $\{g^0, g^1, \cdots\}$ necessarily a subgroup of G when G is infinite?

Solution: For a set to be subgroup of another group, it is sufficient if it is a group by itself under the operator inherited from the subsuming group. Let us consider a group $\mathbb{G}$ and let us denote its operator by '*'.

If $\mathbb{G}$ is finite, it means that the series g, g^2, g^3, g^4, g^5..... cannot continue forever without repetitions. This is because, it would otherwise contradict the finiteness of group $\mathbb{G}$. Therefore, we have seen that there must be at least one i and j such that:

$$g^i = g^j$$

WLOG, assume that i > j, then we can see that

$$g^{i-j} = 1 \ (the\ identity)$$

Suppose k is the least power for the given g such that $g^k = 1$. Then it is easy to show that

$$G_{cycle} = g,\ g^2,\ g^3,\ g^4,\ g^5 ...., g^k\ is\ a\ group$$

This is because the set is closed, inherits associativity from the mother group $\mathbb{G}$, has an identity which is g^k and finally each element g^i has an inverse, which is g^{k-i}.

From here we can argue that the infinite set $\langle g \rangle$ is equal to this smaller set. This is because $G_{cycle} \subseteq \langle g \rangle$ and also for each element in $\langle g \rangle$, we can consider its power modulo k and the element generated will be same as this element. Therefore $\langle g \rangle \subseteq G_{cycle}$.

$$\text{As } G_{cycle} \subseteq \langle g \rangle \text{ and } \langle g \rangle \subseteq G_{cycle}, \text{ we have, } G_{cycle} = \langle g \rangle$$

Hence, $\langle g \rangle$ is a subgroup of g as G_{cycle} is a group inside G (i.e a subgroup of G).

This need not hold true when the G is infinite. As a counter example take the group $\mathbb{Z}$ over '+'. This is a group, but $\langle 1 \rangle = \{1^i \mid i \geq 0\}$ neither forms a cycle like above nor is the set a subgroup (no identity inside).

3. (2 points) If $N = pq$ and $ed = 1 \mod \phi(N)$ then for any $x \in \mathbb{Z}_N^*$ we have $(x^e)^d = x \mod N$. Show that this holds for all $x \in \mathbb{Z}_N$. **Hint:** Use the Chinese remainder theorem.

Solution: Taking p,q to be prime numbers, we have N = pq. Therefore, $\phi(N)$ = (p-1)*(q-1). From here, consider $(x^e)^d = x^{ed}$.

From the information in the question, we can write this as:

$$x^{ed} = x * x^{k(p-1)(q-1)}$$

Consider that both p,q are co-prime with x. Then from Fermat's little theorem, we have: $x^{p-1} = 1 mod p$ and $x^{q-1} = 1 mod q$. Thus $x^{(p-1)(q-1)} = 1 mod p$ and $x^{(p-1)(q-1)} = 1 mod q$. As p,q are primes, they are co-primes. Thus pq must divide $x^{(p-1)(q-1)} - 1$, we have $x^{(p-1)(q-1)} = 1 mod N$.
Taking power of k and then multiplying x on both sides gives: $x^{ed} = x mod N$.

If x is not co-prime with p but is co-prime with q, then we have p / x and q / $x^{k(p-1)(q-1)} - 1$ from Fermat's little theorem. Multiplying these two, we get:

$$\text{N=pq divides } x^{k(p-1)(q-1)+1} - x$$
$$\Rightarrow x^{k(p-1)(q-1)+1} = x mod N$$
$$\Rightarrow x^{ed} = x mod N$$

Symmetrically, we can prove the case where gcd(x,p) = 1 but x and q are not co-prime.
This leaves us with the case when x is not co-prime with both p,q. Here x = Apq = AN, and in that case LHS and RHS of $x^{ed} = x mod N$ go to zero trivially and it becomes true.

4. (2 points) Let $N = pq$ be a product of two distinct primes. Show that if N and $\phi(N)$ are known, it is possible to compute p and q in polynomial time.

Solution: We know that if m and n are co-prime then $\phi(mn) = \phi(m) * \phi(n)$. Similarly, in this case, we have:

$$\phi(pq) = \phi(p) * \phi(q)$$
$$\phi(N) = \phi(p) * \phi(q)$$
$$\phi(N) = (p-1) * (q-1)$$

Therefore, if we know the values of p-1*q-1 and p*q, then we can easily find the value (p,q) using this poly-time algorithm:

Say x = p-1 * q-1 and y = p * q.
Compute p+q = y-x+1
Compute p-q = $\pm((y - x + 1)^2 - 4y)^{\frac{1}{2}}$

Therefore, p = y-x+1 $\pm$ $((y-x+1)^2-4y)^{\frac{1}{2}}$ /2 and q = y-x+1 $\pm$ $((y-x+1)^2-4y)^{\frac{1}{2}}$ /2
All these computations take poly-time and this the algorithm is a poly-time algorithm. (Resolve $\pm$'s as needed).

5. (2 points) Let $N = pq$ be a product of two distinct primes. Show that if N and an integer d such that $3d \equiv 1 \mod \phi(n)$ are known, then it is possible to compute p and q in polynomial time.
(**Hint:** First obtain a small list of possible values of $\phi(n)$.)

Solution: The solution to this question is not very different from the solution to the above question: except that we first need to get $\phi(n)$ from given d. For this divide d by N fist to get the quotient a, then we have:

$$\text{N.a} < \text{d} < \text{N.(a+1)}$$
$$\Rightarrow \frac{\phi(N).a}{(1-\frac{1}{p})(1-\frac{1}{q})} < d < \frac{\phi(N).(a+1)}{(1-\frac{1}{p})(1-\frac{1}{q})}$$

From here we see that, to get the quotient we have narrowed down the range of $\phi(N)$ as:

$$\frac{d*(1-\frac{1}{p})*(1-\frac{1}{q})}{a} > \phi(N) > \frac{d*(1-\frac{1}{p})*(1-\frac{1}{q})}{a+1}$$

The range that $\phi(N)$ can be now has the length of d * $\frac{\phi(N)}{N} * \frac{1}{a^2} = \frac{\phi(N)*N}{d}$ as a is approximately d/N. Overall, the length is of order O($\frac{N^2}{d}$), we can search each element and check if it divides 3d-1 (in polynomial time). Once we get the **candidates** for $\phi(N)$, then we can use the algorithm of the question above to get p,q. Whenever p,q as returned by the above algorithm are integers we have arrived at the correct answer. All this can be done in **polynomial time** because we need to run a poly-time algorithm on poly-number of candidates for $\phi(N)$.

2 One Way Functions and Negligible Functions

1. (2 points) If $\mu(.)$ and $\nu(.)$ are negligible functions then show that $\mu(.) \cdot \nu(.)$ is a negligible function.

Solution: Suppose f,g are two negligible functions. By definition of negligible functions, we have:

$$\forall c\ f(x) < \frac{1}{x^c}\ when\ x > N_f^c$$
$$\forall c\ g(x) < \frac{1}{x^c}\ when\ x > N_g^c$$

Now consider f.g such i.e f(.).g(.), without loss of generality assume that $N_f^c > N_g^c$. From here, we can see that:

$$\forall c\ f(x) < \frac{1}{x^c}\ when\ x > N_f^c$$

$$\forall c\ g(x) < \frac{1}{x^c}\ when\ x > N_f^c$$

As c, N_f^c are positive integers this means that

$$\forall c\ g(x) < 1\ when\ x > N_f^c$$

Because $\frac{1}{x^n} < 1$ for integer x's. Otherwise, we can ensure that x > 1 by setting appropriate $N_f^{c\prime}s$
Therefore

$$\forall c\ f(x).g(x) < f(x) < \frac{1}{x^c}\ when\ x > N_f^c$$

Thus, by definition, f.g is a negligible function as we found a positive integer N for each 'c' such that f.g $< \frac{1}{x^c}$.

2. (2 points) If $\mu()$ is a negligible function and $f()$ is a function polynomial in its input then show that $\mu(f())$ are negligible functions.

Solution: Depending on the coefficient of its leading term, a polynomial shoots up to either positive or negative infinity at large values of x (> its largest root). In this problem, let us only consider the polynomials with positive leading coefficient.

Let p(x) be a polynomial in x with degree n. As we assumed that the leading coefficient is positive, we have:

$$p(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n\ where\ a_n > 0$$

Now if consider p(x)-x, we will prove that if this will be positive after a certain constant say N_p. For this, consider

$$(p(x) - x)' = a_1 + 2 * a_2x + ... + n * a_nx^{n-1} - 1$$

Clearly, the difference between p(x) and x reduces if p(x) < x for large values of x. Now consider:

$$(p(x) - x)'' = 2 * a_2 + ... + n * n - 1 * a_nx^{n-2}$$

From this, we find that p(x) closes the gap to x faster and faster as x increases. i.e **p(x)> x for all x > say N_p.**

Now consider a negligible function f. From here using the above derivation that p(x)>x for x> N_p, we can choose the N_fp^cs for each positive integer c such that f(p(x)) > $\frac{1}{x^c}$ $when\ x > N_c$.

To this end, consider the $N_c^{f\prime}s$ of f such that f(x) $> \frac{1}{x^c}$ *when* $x > N_c^f$. Now we arrive at two cases for any chosen integer, c:

Case 1) $\forall x > N_c^f$; p(x) > x, in this case, for any x $> N_c^f$ *we have* $f(p(x)) < \frac{1}{p(x)^c} < \frac{1}{x^c}$ from the negligibility condition of f. Thus $N_f p^c = N_f^c$.

Case 2) p(N_c^f) $< N_c^f$. Clearly, $N_p > N_c^f$ This calls for a new constant for $N_f p^c$. We claim that using N_p would be sufficient here for the respective c. This is because, we have $f(p(x)) < \frac{1}{p(x)^c} < \frac{1}{x^c}$ as $\frac{1}{x^c}$ is **decreasing** function. Therefore, we have $f(p(x)) < \frac{1}{x^c}$ for all x $> N_p$. Therefore, if we set $N_f p^c = N_p$, we have the negligibility condition satisfied $\forall x > N_f p^c$.

As we have shown a strategy to set new $N_f p^c$'s for an arbitrary f,p,c we can extend this to all negligible functions f, polynomials p and integers c such that negligibility is met for x $> N_f p^c$. **Therefore, if f is negligible, p is a polynomial then f(g(.)) is a negligible function.**

3. (2 points) Prove that the existence of one-way functions implies $\mathsf{P} \neq \mathsf{NP}$.

Solution: We need to show that existence of one-way functions implies $\mathsf{P} \neq \mathsf{NP}$. We will prove the contrapositive of this by showing that if $\mathsf{P} = \mathsf{NP}$, then there can be no OWFs.

Assume P=NP (unlike many researchers who don't), this means that whatever language membership test that can be done by a non-deterministic Turing machine in polynomial time can also be done by a deterministic Turing machine in polynomial time.

Now consider the language L in this way; L = {x$\$^n$y | x is a prefix of some x' such that f(x')=y}. Note that the $ here is just a buffer tape symbol that acts as our separator. This language can be recognized by a NDTM in polynomial time in this way:

1) Move tape to end of x without any overwrite i.e until the first $ is read, keep the length of x as the name of the state currently in. (This way the TM knows the length of the prefix)

2) Suppose $|x| = i$, then once $ is read at $i + 1^{th}$ position, do the non-deterministic overwrites until n bits are read from the start as:

In each move through in this non-deterministic stage, write either a '1' or '0' instead of $.

3) Now that n bits have been written and there are virtually 2^{n-i} non-deterministic threads in hand, let each of this thread move back by n bits. i.e when a $ is read at the $n + 1^{th}$, go to a series of n states that just moves the tape back to the initial position.

4) We know that a OWF must be computable by a Monte-Carlo Algorithm, that is a PPT must be able to compute it with $\frac{2}{3}$ (say) probability. This is very easy to be emulated by a DTM, and thus also easy by a NDTM in polynomial time. Therefore, after the series of n states that take the tape to the initial

position, go to a new state (new start) and where this emulation can be applied.

5) Now we have the tape reading: y'$^{x}y, where x depends on the length of output of OWF. Thus our tape now looks like this:

$$$$$$1101101$$$$$1101101

6) From here, it is easy to verify if the left and the right part of the tape, separated by $^{x} are the same (very similar to accepting {ww | w ϵ{0,1}*}).

Thus we have shown that a NDTM can accept the language L = {x$^{n}y | x is a prefix of some x' such that f(x')=y}. As P=NP, there exists a DTM that can accept the same language in polynomial time. Let us call this DTM as 'D'. Now we can construct an adversary A that can make use of the polytime membership testing that 'D' does and invert any OWF in polytime. The scheme of A, for a given y, is as follows:

1) Start with the empty string a = ϵ

2) For i = 1 to n:

Check if (a||0$^{n}y)$\epsilon$ L.

If Yes:

a = a||0

Else:

a = a||1

There is no chance that neither of a||0 or a||1 is a solution to $f^{-1}(y)$ because the given OWF must have had an pre-image in {0,1}^{n}. Thus we have a polynomial time 100% inverter of f.

Taking the contrapositive now, we get that existence of OWF implies that P$\neq$ NP.

4. (2 points) Prove that there is no one-way function $f : \{0,1\}^n \to \{0,1\}^{\lfloor \log_2 n \rfloor}$.

Solution: For a function f to be a OWF, we have: for all PPT A, Pr(A(f(x))= $f^{-1}f$(x): x$\leftarrow^u$D) $\leq \frac{1}{Q(k)}$ for all polynomials Q.

Here, we have a domain of size 2^n and a range of size $2^{\lfloor \log_2 n \rfloor}$. For a 'x' taken from these 2^n with uniform distribution, we have:

Pr(A(f(x))= $f^{-1}f$(x): x$\leftarrow^u${0,1}^{n})$\leq \frac{1}{Q(k)}$ for all polynomials Q.

From pigeon hole principle, we have a 'y' in range such that it has at least $\frac{2^n}{n}$ elements. From here, it is easy to construct a guesser A which always guesses this element 'y' in O(1) time. On a randomly chosen input, A will be correct with the probability of:

$$\frac{\frac{2^n}{n}}{2^n} = \frac{1}{n}$$

Clearly, $\frac{1}{n}$ is not a negligible function. Therefore, there can be no OWFs from $\{0,1\}^n \to \{0,1\}^{\lfloor \log_2 n \rfloor}$.

5. (2 points) Let $f : \{0,1\}^n \to \{0,1\}^n$ be any one-way function then is $f'(x) \stackrel{def}{=} f(x) \oplus x$ necessarily one-way?

Solution:

Doubt: There are cases when f'(x) is not OWF. But, are there cases when f'(x) is actually a OWF?

Actual Solution:

By looking at the $\oplus$ operator, we get an idea to XOR the given input 'x' with a few zeroes so as to extract x directly by looking at the output in O(n) time. After some thought, we can come up with the following construction for the case of $g : \{0,1\}^{2n} \to \{0,1\}^{2n}$ (as n is arbitrary, proving for 2n will suffice):

Define g in this way:

$$g(x\|y) \;=\; 0^n \| f(x)$$

for any y $\epsilon \{0,1\}^n$ For this g, g' as defined in the question, i.e,

$$g'(x\|y) \;=\; g(x\|y) \oplus x\|y$$

where x,y $\epsilon \{0,1\}^n$ is clearly not one-way because: just returning the first n-bits concatenated with y[n+1:2n] $\oplus$ f(y[1:n])(assuming adversary has access to OWF f) would be a valid pre-image for the given y, under g', i.e., $g(x\|y) \oplus x\|y$. Thus we have inverted the proposed construction of g' with probability of 1 in O(n)+OWF computation time. As 1 is not negligible, clearly it cannot be the case that g' is a OWF. As we have a **counterexample**, we can say that **for at least one OWF f, f' will not be a OWF.** However, there might be some OWFs such that f' would be a OWF.

6. (2 points) Prove or disprove: If $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function, then $g : \{0,1\}^n \to \{0,1\}^{n-\log n}$ is a one-way function, where $g(x)$ outputs the $n - \log n$ higher order bits of $f(x)$.

Solution: If k x's in $\{0,1\}^n$ map to a certain y in $\{0,1\}^n$, then the expected number of x's in $\{0,1\}^n$ mapping to a y' in $\{0,1\}^{n-logn}$ is nk.

Suppose there is an inverter for g, then for a given y', we can get it's pre-image with a non-negligible probability. For a given y in $\{0,1\}^n$, if we need to get its pre-image w.r.t f, then we can simply truncate the last logn bits and ask the inverter of g to invert it.

The answer that the inverter of g will be correct with a non-negligible probability because:
Pr(A(f(x)) is in f^{-1}(f(x))) = Pr(A(g(x)) is in $g^{-1}(g(x))$) * $\frac{1}{n}$
This can be seen from the first claim about expected x's that map to a specific g(x) being n times the number of x's that map to a specific f(x) where the g(x) is prefix of f(x).

As the RHS is a product of a non-negligible function and the inverse of a polynomial, it has to be non-negligible. Thus if g had a non-negligible inverter, that would invert f also non-negligibly in poly-time. Hence g has no inverter. Therefore, the given construction of g maintains one-wayness.

7. (2 points) If f is a one-way function then is $f^2(x) = f(f(x))$ always a one-way function?

Solution: Let us begin by assuming the existence of one-way functions.
Let there be a OWF g: $\{0,1\}^n \to \{0,1\}^n$. From here, let us construct another OWF f such that f($x_1 \circ x_2$) in this way, where $|x_1| = |x_2| = n$:

$$f(x_1 \circ x_2) = 0^n \circ g(x_1)$$

Clearly f: $\{0,1\}^{2n} \to \{0,1\}^{2n}$. f must be a OWF because if it has a PPT inverter, we can form a contradiction as follows: if A inverts f, then A($0^n \circ g(x_1)$) $= x_1 \circ x_2$ with a non-negligible probability say $\mu(n)$.

Then using this inverter, we can construct another inverter: B which takes g(x) pads n-0 bits in the front and calls A. This way B can retrieve x with the same non-negligible probability $\mu(n)$.

Therefore, f is an OWF
Now consider $f^2(x1 \circ x2) = f(f(x1 \circ x2)) = 0^n \circ g(0^n)$. Then the claim that f^2 is not a OWF can easily seen to be true because, the range of f is a singleton set. Therefore any trivial inversion is a correct inversion which means we can invert f^2 with a probability of 1. 1 is not a negligible function and this we have:
If f is a OWF, f^2 need not be a OWF.

3 Fun With One Way Functions

Suppose that $f(x)$ is a one-way function. Let $|x|$ denote the length of the binary string x. We let $\circ$ denote the concatenation operator. Similarly, $(\circ)$ is the parse operator which we can use to represent a string x as $x = x_1(\circ)x_2$ where $|x_1| = |x_2|$. (Assume for simplicity that all strings to which this operator is applied are of even length; for example, this can be accomplished by appending a 0 to the end of an odd-length string prior to applying this operator.) Function f here is *length-preserving*, which means that $|f(x)| = |x|$, and also that we need not give the adversary 1^k as input.

1. (3 points) Prove that the following is not a one-way function:
   $f_a(x) = f(x_1) \oplus x_2$, where $x = x_1(\circ)x_2$.

Solution: We will give a very efficient adversary that can return a valid pre-image w.r.t the function f_a when y, an element in its range, is given. That is, we will develop an adversary A for f_a that can take in y

and can give $f^{-1}(y)$ successfully.

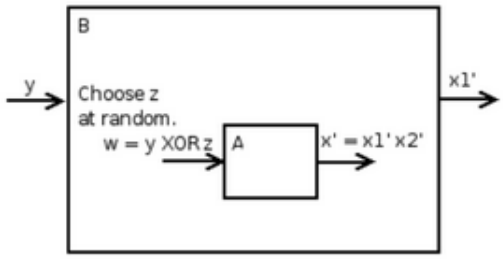
Let us assume that the adversary A has function f as a black-box. This should not change the difficulty of OWF invertibility. The algorithm for A:

Generate a random n-digit binary say a
Compute f(a) using OWF black box
Generate b such that b = f(a)⊕y
Return a‖b

Clearly $f_a(a, b) = y$ from our construction. Therefore, we have found a valid pre-image without in just O(n)+OWF-computation time. Thus, from the given construction of f_a, it cannot be a OWF.

2. (3 points) Find the fault in the following proof that f_a is one-way.

   $f_a(x)$ is a one-way function. Assume for the sake of contradiction that we have a PPT inverter $\mathcal{A}$ for $f_a(x)$ that, when given w, outputs some x' such that $f_a(x') = w$ with nonnegligible probability. We want to use this $\mathcal{A}$ to construct an inverter for the one-way function $f(x)$. Let $\mathcal{B}$ be a PPT that on input y picks a random string $z \leftarrow \{0,1\}^{|y|}$, runs $\mathcal{A}$ on $w = y \oplus z$ to get back some value $x' = x'_1(\circ)x'_2$, and then returns x'_1.



   What happens when $\mathcal{A}$ succeeds? This means that the x' that $\mathcal{A}$ returns is such that $f(x'_1) \oplus x'_2 = f_1(x') = w = y \oplus z$. Because f is length preserving and y and z have the same length, we know that $f(x'_1) = y$ and $x'_2 = z$. Therefore, the x'_1 that $\mathcal{B}$ returns is a preimage of y.

   This means that when $\mathcal{A}$ succeeds, so does $\mathcal{B}$, which further implies that the probability of $\mathcal{B}$ succeeding is at least the probability of $\mathcal{A}$ succeeding inside $\mathcal{B}$. Since the input to $\mathcal{A}$ inside $\mathcal{B}$ is distributed identically to the input to $\mathcal{A}$ in the wild, the probability of $\mathcal{A}$ succeeding inside $\mathcal{B}$ is equal to the probability of $\mathcal{A}$ succeeding in the wild, which is non-negligible by assumption. So the probability of $\mathcal{B}$ succeeding is also non-negligible. But this means that $\mathcal{B}$ is an inverter for the one-way function $f(x)$ that works with non-negligible probability, which is a contradiction. So $f_a(x)$ must be a one-way function.

Solution: This is one of the first proofs that comes into mind when proving the above problem. However, it is fallacious. The main fallacy lies in the step:

This means that the x' that $\mathcal{A}$ returns is such that $f(x'_1) \oplus x'_2 = f_1(x') = w = y \oplus z$. Because f is length preserving and y and z have the same length, we know that $f(x'_1) = y$ and $x'_2 = z$. Therefore, the x'_1 that $\mathcal{B}$ returns is a pre-image of y.

The fallacy occurs when the author explicates that $f(x'_1) = y$ and $x'_2 = z$ from $f(x'_1) \oplus x'_2 = w = y \oplus z$ using the length preserving argument which simply does not hold true here (it might have worked if $\oplus$ meant concatenation). This is because it is easy to come up with other candidate solutions. For example, $f(x'_1) = y \oplus 00....01$ and $x'_2 = z \oplus 00...01$ would also work fine. In fact, for any s $\epsilon\{0,1\}^n$, $f(x'_1) = y \oplus s$ and $x'_2 = z \oplus s$ would be a valid solution.

From here, it is easy to see that the rest of the argument becomes fallacious too, thus making the proof incorrect in general. This is because of the assumption that

This means that when $\mathcal{A}$ succeeds, so does $\mathcal{B}$.

Does not hold true anymore. **Thus we have found a fault which also makes the further proof fallacious, in the argument given.**

3. (3 points) Prove that one-way functions cannot have polynomial-size ranges. More precisely, prove that if f is a one-way function, then for every polynomial $p()$ and all sufficiently large n's, $|\{f(x) : x \in \{0,1\}^n\}| > p(n)$

Solution: By the definiton of one-way functions, we have:

$$\Pr(\mathrm{A}(\mathrm{f}(\mathrm{x})) = f^{-1}(f(x)) : \mathrm{x} \leftarrow^{\mathrm{u}} \{0,1\}^n) = \mu(n)$$

Where μ is a negligible function. Let us also plug-in the mathematical definition of negligible functions into this:

$$\forall P \forall A (\forall n > n_{c,p}) Pr(A(f(x)) = f^{-1}(f(x)) \ : \ x \leftarrow^{\mathrm{u}} \{0,1\}^n) < \tfrac{1}{P(n)}$$

For this proposition to be false, it is enough to show that there is an adversary A_n for each integer n such that $Pr(A_n(f(x)) = f^{-1}(f(x)) \ : \ x \leftarrow^{\mathrm{u}} \{0,1\}^n) > \frac{1}{P(n)}$ for at least one polynomial P, i.e., at each integer n, there is a non-negligible inverter.

This is the kind of adversary that we will aim to construct.

Construction of Adversaries:

Consider an arbitrary integer n, if the OWF f: $\{0,1\}^n \to R$ where R is the range that has a polynomial size P(n). Then from pigeon hole, principle we have:

$\exists a \epsilon R, \exists S \subset \{0,1\}^n \ \|S\| \geq \frac{2^n}{P(n)}$ and $\forall x \in S, f(x) = a$
The adversary A_n that always guesses this '**a**' will be correct with a probability of:

$$\frac{\|S\|}{2^n} \geq \frac{1}{P(n)}$$

As our choice of n has been arbitrary, we can do this for any n. This means that for each n, we have an inverter A_n that can invert it with the above non-negligible probability.

$$\forall n, \exists A_n \text{ s.t } \Pr(A_n(\text{f(x)}) = f^{-1}(f(x)) : \text{x} \leftarrow^{\text{u}} \{0,1\}^n) \text{ is non negligible.}$$
$$\Rightarrow \sim [\forall n, \forall A \ \Pr(\text{A(f(x))} = f^{-1}(f(x)) : \text{x} \leftarrow^{\text{u}} \{0,1\}^n) \text{ is negligible.}]$$

Clearly, we have proven the negation of the condition over based on which we classify a function as a OWF. **Therefore f is not a one-way function.** Hence no function that has a polynomial range is a OWF. The same contradiction can be arrived at if we consider $\|R\| < P(n)$ for some polynomial. Therefore, we have at sufficiently large n's, $|\{f(x) : x \in \{0,1\}^n\}| > p(n)$.

4. (3 points) Let f be a one-way function. Prove that $g(x) = f(x_1)$, where $x = x_1(\circ)x_2$, is a one-way function.

Solution: The solution is straightforward if we use a possible efficient adversary of g to construct a efficient adversary for f.

Let us consider an arbitrary OWF f, mapping $\{0,1\}^n \rightarrow \{0,1\}^k$. g(x) = f(x1) where $\text{x} = x_1 \circ x_2$. For the sake of contradiction, let us assume that g is not a OWF. This means that there is an efficient adversary B such that it inverts g with non-negligible probability over a randomly chosen input.

$$\exists B \text{ s.t } Pr(B(g(x)) = g^{-1}(g(x)) \ : \ x \leftarrow^{\text{u}} \{0,1\}^n) \text{ is not negligible.}$$

Now consider the adversary A that just invokes B on it's own input and considers the first n of 2n bits that B outputs. i.e.,

$$A(y) = B(y)_{[1....n]}$$

Obviously, A is efficient. Consider the case when B successfully inverts the output of g, i.e., y. This means that when the first half of B(y) is fed into f, we get back y (from definition of g).

$$f(B(y)_{[1....n]}) = y$$

But this also means that first half of B(y) is a correct pre-image for y w.r.t the function f (as $\text{f}(B(x)_{[1:n]})$=y). Thus with the given construction, A succeeds whenever B does. Also A is efficient. Hence A is a PPT inverter that inverts f with a **non-negligible probability**. Hence a contradiction as f is OWF.

Therefore, g has no PPT adversaries. **Therefore, g is a one-way function.**