# Problem Set 3

name: Guy Levy
id: 206865362

# 1 . Public-key Encryption from QR

## 1 .1 QR given factorization

by CRT $\mathbb{Z}_N^* \cong \mathbb{Z}_P^* \times \mathbb{Z}_Q^*$
so $x$ in $\mathbb{Z}_N^*$ corresponds to $(x(\bmod P), x(\bmod Q))$ in $\mathbb{Z}_P^* \times \mathbb{Z}_Q^*$.
so $x$ has root $r$ ( i.e. $r^2 \equiv_N x$ ) if and only if $(x(\bmod P), x(\bmod Q))$ has root $(r(\bmod P), r(\bmod Q))$.
Thats true if and only if $x$ has root mod $P$ and $x$ has root mod $Q$.
Thats true if and only if $x$ is $QR(P)$ and $x$ is $QR(Q)$.
Thats true if and only if $(\frac{x}{P}) = 1 \wedge (\frac{x}{Q}) = 1$.
We saw in the tutorial fast way to compute Legendre Symbol: $(\frac{x}{P}) = x^{\frac{P-1}{2}}$.

So I suggest the following algorithm:
Given $P, Q, x$ : return the boolean ( $(\frac{x}{P}) = 1$ and $(\frac{x}{Q}) = 1$ )

It is in poly(n) because $P, Q$ are n bit primes and from previous homework we can compute exponent of n bit primes in poly time.

## 1 .2 Generating QR

Let $x \in QNR^*(N)$. Prove that if $y \in_R \mathbb{Z}_N^*$ then $y^2 x \in_R QNR^*(N)$.
Want to prove: $\forall t \in QNR^*(N) : Pr_{y \leftarrow \mathbb{Z}_N^*}[y^2 x = t] = \frac{1}{|QNR^*(N)|}$
By CRT $\mathbb{Z}_N^* \cong \mathbb{Z}_P^* \times \mathbb{Z}_Q^*$

- Let $t \in QNR^*(N)$, from CRT $t$ corresponds to $(t(\bmod P), t(\bmod Q))$ in $\mathbb{Z}_P^* \times \mathbb{Z}_Q^*$. And we saw in the previous question $t$ is $QNR(P)$ and $QNR(Q)$.
- $x$ corresponds to $(x(\bmod P), x(\bmod Q))$ in $\mathbb{Z}_P^* \times \mathbb{Z}_Q^*$.
- Also from CRT sampling $y$ from $\mathbb{Z}_N^*$ is equivalent to sampling $y_P$ from $\mathbb{Z}_P^*$ and $y_Q$ from $\mathbb{Z}_Q^*$ ( because each unique pair $y_P, y_Q$ in $\mathbb{Z}_P^* \times \mathbb{Z}_Q^*$ defines unique element $y$ from $\mathbb{Z}_N^*$.

With this continue analysis:

$$Pr_{y \leftarrow \mathbb{Z}_N^*}[y^2 x = t]$$

$$= Pr_{y_P \leftarrow \mathbb{Z}_P^*, y_Q \leftarrow \mathbb{Z}_Q^*}[(y(\bmod P), y(\bmod Q))^2 (x(\bmod P), x(\bmod Q)) = (t(\bmod P), t(\bmod Q))]$$

$$= Pr_{y_P \leftarrow \mathbb{Z}_P^*, y_Q \leftarrow \mathbb{Z}_Q^*}[y_P^2 x \equiv_P t \wedge y_Q^2 x \equiv_Q t]$$

$$= Pr_{y_P \leftarrow \mathbb{Z}_P^*}[y_P^2 x \equiv_P t] \cdot Pr_{y_Q \leftarrow \mathbb{Z}_Q^*}[y_Q^2 x \equiv_Q t]$$

$$=^{(*)} \frac{1}{|QNR(P)|} \cdot \frac{1}{|QNR(Q)|}$$

$$=^{(**)} \frac{1}{|QNR*(N)|}$$

Explanation $(*)$:

Will prove $f(x) = y_P^2 x$ is a permutation from $QNR(P)$ to itself, similarly, $g(x) = y_Q^2 x$ is a permutation on $QNR(Q)$.
Will show $f$ (for $g$ same proof exacly) is bijective and that $Imf \subseteq QNR(P)$ and thus a permutation.

Bijective: assume for the sake of contradiction $\exists a, b \in QNR(P) : a \neq b \wedge y_P^2 a = y_P^2 b$.
From the fact that $\mathbb{Z}_P^*$ is a field we get the immediate contradiction $a = b$.

$Imf \subseteq QNR(P)$: let $x \in QNR(P)$ calculate Legendre Symbol of $y_P^2 x$:

$$(y_P^2 x)^{\frac{P-1}{2}} = y_P^{P-1} x^{\frac{P-1}{2}} = 1 \cdot (\frac{x}{P}) = -1$$
$$\Rightarrow (\frac{y_P^2}{P}) = -1$$
$$\Rightarrow y_P^2 \in QNR(P)$$

Explanation $(**)$:

$QNR*(N)$ contains exactly the elements from $\mathbb{Z}_N^*$ which correspond to the elements $(a, b)$ from $\mathbb{Z}_P^* \times \mathbb{Z}_Q^*$ such that $(\frac{a}{P}) = (\frac{b}{Q}) = -1$

# 1 .3 Public Key Encryption

Description of scheme:

$Gen(1^n)$: sample 2 random n-bit primes $P, Q$. assign $N = P \cdot Q$.
Sample $t \in_R QNR^*(N)$. ( by sampling $t_1$'s untill $t_1 \in QNR(P)$ and $t_2$'s untill $t_2 \in QNR(P)$ and calculating $t \in \mathbb{Z}_N^*$ s.t $t \equiv_P t_1, t \equiv_Q t_2$).
return: $pk = (N, t), sk = (P, Q)$

$Enc_{pk}(b)$: sample $r \in_R \mathbb{Z}_N^*$, return $r^2 \cdot t^b$.

$Dec_{sk}(c)$: if $c^{\frac{P-1}{2}} \equiv_P 1$ and $c^{\frac{Q-1}{2}} \equiv_Q 1$ return 0.
if $c^{\frac{P-1}{2}} \equiv_P -1$ and $c^{\frac{Q-1}{2}} \equiv_Q -1$ return 1.

Correctness:

if b = 1 $Dec_{sk}(Enc_{pk}(1)) = Dec_{sk}(r^2 \cdot t^b)$
$(r^2 \cdot t)^{\frac{P-1}{2}} \equiv_P r^{P-1} \cdot t^{\frac{P-1}{2}} = 1 \cdot (\frac{t}{P}) = -1$
$(r^2 \cdot t)^{\frac{Q-1}{2}} \equiv_Q r^{Q-1} \cdot t^{\frac{Q-1}{2}} = 1 \cdot (\frac{t}{Q}) = -1$
$\Rightarrow Dec_{sk}(r^2 \cdot t) = 1 = b$

if b = 0 $Dec_{sk}(Enc_{pk}(0)) = Dec_{sk}(r^2)$
$(r^2)^{\frac{P-1}{2}} \equiv_P r^{P-1} = 1$
$(r^2)^{\frac{Q-1}{2}} \equiv_P r^{Q-1} = 1$
$\Rightarrow Dec_{sk}(r^2 \cdot t^0) = 0 = b$

CPA security:

Game: challenger: generates $pk, sk$. sends $pk$ to Adversary. Flips coin $b \in_r \{0, 1\}$. Sends $Enc_{pk}(b)$.
Adversay runs some polynomial time and outputs $b'$.

Want to prove CPA security of scheme under QRP Assumption.
Will assume $\exists A$ that wins game with non-negligible advantage.
Will use $A$ to construct distinguisher $D$ that breaks QRP. i.e. distinguishes between uniform distributions over $QR(N), QNR*(N)$. denote those distributions $X_0, X_1$ respectivly.
Given $x$, $D$ samples $r \in_R \mathbb{Z}_N^*$.
$D$ simulates game with $A$ as adversary and instead of some encoding gives adversary $r^2 x$
If $x$ sampled from $X_1$ the setting is identical to a challenger encoding of $b = 1$.
If $x$ sampled from $X_0$ the setting is identical to a challenger encoding of $b = 0$.
$D$ decides based on $A$'s answer.
Because $A$ has a non-negligible advantage, so do $D$ and so $D$ breaks QRP.

# 1 .4 Malleability

Let $\sigma_1, \sigma_2$ be some boolean values.
$pk = (N, t), sk = (P, Q)$.

$Enc_{pk}(\sigma_1) = r^2 t^{\sigma_1} = c_1$
$Enc_{pk}(\sigma_2) = r^2 t^{\sigma_2} = c_2$

Claim: $Dec_{sk}(c_1 \cdot c_2) = \sigma_1 \oplus \sigma_2$.

From the claim we get that $c_1 \cdot c_2$ is an encryption for $\sigma_1 \oplus \sigma_2$ and can be obtained efficiently using just the encryptions for $\sigma_1, \sigma_2$. So prove claim and done.
Proof of claim: $Dec_{sk}(c_1 \cdot c_2) =?$

If $\sigma_1 \oplus \sigma_2 = 0$ i.e. $\sigma_1 = \sigma_2$, denote them just $\sigma$ for now.
Then $c_1 \cdot c_2 = r_1^2 t^{\sigma_1} r_2^2 t^{\sigma_2} = (r_1 r_2 t^{\sigma})^2 = c$
$\Rightarrow c^{\frac{P-1}{2}} \equiv_P (r_1 r_2 t^{\sigma})^{P-1} \equiv_P 1, c^{\frac{Q-1}{2}} = 1$
$\Rightarrow Dec_{sk}(c_1 \cdot c_2) = 0 = \sigma_1 \oplus \sigma_2$

Else $\sigma_1 \neq \sigma_2$, and so $\sigma_1 + \sigma_2 = 1$
$c_1 \cdot c_2 = r_1^2 t^{\sigma_1} r_2^2 t^{\sigma_2} = r_1^2 r_2^2 t^1 = c \Rightarrow c^{\frac{P-1}{2}} \equiv_P (r_1 r_2)^{P-1} t^1 \equiv_P (\frac{t}{P}) = -1, c^{\frac{Q-1}{2}} = -1$
$\Rightarrow Dec_{sk}(c_1 \cdot c_2) = 1 = \sigma_1 \oplus \sigma_2$

## 1 .5 Refresh

Description of $Refresh(pk, c)$: assign $c_0 = Enc_{pk}(0)$, return $c_0 \cdot c$

(sanity check) Refresh goves valid encryption for $m$:
from 1.4: $Dec_{sk}(Refresh(pk, c)) = Dec_{sk}(c_0 \cdot c) = m \oplus 0 = m$

identical distribution:

$Enc_{pk}(m) = r^2 \cdot t^m$
$Refresh_{pk}(c) = c \cdot Enc_{pk}(0) = r_1^2 t^m r_2^2 t^0 = (r_1, r_2)^2 t^m$

When $r, r_1, r_2$ all sampled from uniform distribution over $\mathbb{Z}_N^*$. Left to prove:

$$R^2 t^m \sim (R_1 \cdot R_2)^2 t^m$$

Enough to show:

$$R^2 \sim (R_1 \cdot R_2)^2$$

That holds because:

$$R \sim R_1 \cdot R_2$$

# 2 . Statistically Hiding Commitments

## 2 .1 Inner Product with Random String

Let $b \in \{0,1\}^n, b \neq \vec{0}$.
Let $A_0 = \{a \in \{0,1\}^n | < a, b >= 0\}$, $A_1 = \{a \in \{0,1\}^n | < a, b >= 1\}$

notice $A_0 \cup A_0 = \{0,1\}^n$.

Also

$$Pr_{a \leftarrow \{0,1\}^n}[< a, b >= 0] = Pr_{a \leftarrow \{0,1\}^n}[a \in A_0] = \frac{|A_0|}{|\{0,1\}^n|} = \frac{|A_0|}{2^n}$$

Similarly,

$$Pr_{a \leftarrow \{0,1\}^n}[< a, b >= 1] = \frac{|A_1|}{2^n}$$

Enough to show $|A_0| = |A_1|$, then it will follow that both probabilities are equal.
Further more, because they sum to 1, each of them gets the value of $\frac{1}{2}$ as I was asked to prove.

$|A_0| = |A_1|$: Will show by existance of permutation $f : A_0 \rightarrow A_1$.
Let $j \in [n]$ be an index in which $b_j \neq 0$ (exists from premise $b \neq \vec{0}$ ).

Existance of permutation $f$:

Defining $f$: $f(a) = a'$ such that $a'_j = 1 - a_j$ and $\forall i \neq j$, $a'_i = a_i$.

First show, for input $a$ in $A_0$: $f(a) \in A_1$.
$a \in A_0$
$\Rightarrow < a, b >= 0$
$\Rightarrow \sum_i^n a_i b_i (mod 2) = 0$
$\Rightarrow < a', b >= \sum_i^n (a_i b_i) - a_j b_j + (1 - a_j) b_j \equiv_2 -2a_j b_j + 1 \equiv_2 1$
$\Rightarrow a' \in A_1$

$f$ is bijective: if $a_1 \neq a_2$:
if $a_{1_j} \neq a_{2_j}$ then $a'_{1_j} = 1 - a_{1_j} \neq 1 - a_{2_j} = a'_{2_j}$.
else $\exists i \neq j$ s.t. $a_{1_i} \neq a_{2_i} \Rightarrow a'_{1_i} = a_{1_i} \neq a_{2_i} = a'_{2_i}$
anyhow we get $a'_1 \neq a'_2$ i.e. $f(a_1) \neq f(a_2)$.

$f$ onto: let $a \in A_1 \Rightarrow f(a) \in A_0$ (similarly to what we already showed).
claim $f(f(a)) = a$ : indeed flipping the $j$th bit twice makes no difference.

# 2 .2 Inner Product is Pairwise Independent

For every $a \in \{0,1\}^n$ define $h_a : \{0,1\}^n \rightarrow \{0,1\}$ as the function $h_a(b) =< a, b >$.
Will show $\{h_a\}_{a \in \{0,1\}^n}$ is UHF.

Let $b_1, b_2 \in \{0,1\}^n$, $b_1 \neq b_2$

$$Pr_{a \leftarrow \{0,1\}^n}[h_a(b_1) = h_a(b_2)] = Pr_{a \leftarrow \{0,1\}^n}[< a, b_1 >=< a, b_2 >]$$

$$= Pr_{a \leftarrow \{0,1\}^n}[a = 0 \wedge < a, b_1 >=< a, b_2 >] + Pr_{a \leftarrow \{0,1\}^n}[a \neq 0 \wedge < a, b_1 >=< a, b_2 >]$$

$$= Pr_{a \leftarrow \{0,1\}^n}[a = 0] + Pr_{a \leftarrow \{0,1\}^n}[< a, b_1 >=< a, b_2 > | a \neq 0] \cdot Pr_{a \leftarrow \{0,1\}^n}[a \neq 0]$$

$$= \frac{1}{2^n} + (1 - \frac{1}{2^n}) Pr_{a \leftarrow \{0,1\}^n}[< a, b_1 - b_2 >= 0 | a \neq 0]$$

$$= \frac{1}{2^n} + (1 - \frac{1}{2^n}) Pr_{a \leftarrow \{0,1\}^n}[a \in A_{0[b = b_1 - b_2]} | a \neq 0]$$

$$= \frac{1}{2^n} + (1 - \frac{1}{2^n})(\frac{|A_0| - 1}{2^n - 1})$$

$$= \frac{1}{2^n} + (1 - \frac{1}{2^n})(\frac{2^{n-1} - 1}{2^n - 1})$$

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

$$= \frac{2^{2n-1} - 2^{n-1}}{2^n(2^n - 1)}$$

$$= \frac{\frac{1}{2}(2^{2n} - 2^n)}{2^{2n} - 2^n)}$$

$$= \frac{1}{2} = \frac{1}{|\{0,1\}|}$$

## 2 .3 Purifying Randomness

$(r, < r, s >), (r, b)$ denote distributions as $(R, < R, Us >), (R, U1)$ when $R \sim U_n$ and $Us \sim uniform(S)$.

Let $S \subseteq \{0,1\}^n$ show $\Delta((R, < R, Us >), (R, U1)) = O(\sqrt{1/|S|})$.

From theorem mentioned in class $\forall f : \Delta(A, B) \geq \Delta(f(A), f(B))$.

It follows immediatly that for a permutation $f$: $\Delta(A, B) = \Delta(f(A), f(B))$

We saw $\{h_r\}_{r \in \{0,1\}^n}$ is a UHF.

Define permutation $f : f(r, b) = (h_r, b)$
($f$ is a permutation: $r$ implies $h_r$ trivially and $h_r$ implies $r$ by applying $h_r(e_i)$ to get $r_i$ for all $i \in [n]$)

From mentioned theorem:

$$\Delta((R, < R, Us >), (R, U1)) = \Delta((h_R, < R, Us >), (h_R, U1))$$

From {h_r} definition:

$$= \Delta((h_R, h_R(Us)), (h_R, U1))$$

$U_s$ is a K-source for $k = [log_2(|S|)]$, from LHL we get:

$$\leq \frac{1}{2}\sqrt{2^{1-log_2(|S|)}} = \frac{\sqrt{2}}{2}\sqrt{1/|S|} = O(\sqrt{1/|S|})$$

## 2 .4 Commitments

Assume $\exists$ CRHF $h : \{0,1\}^n \to \{0,1\}^{n/2}$.

Will show $C(b) = (h(s), r, < r, s > \oplus b)$ ; $r, s \in_R \{0,1\}^n$
is both statistically hiding and computationally binding.

C is statistically hiding:

estimate:

$$\Delta(((h(s), r, < r, s > \oplus 0), (h(s), r, < r, s > \oplus 1))$$

claim: $h'(r) = (h(s_0), < r, s_0 >)$ is UHF.

proof of claim:

$$Pr_{s_0 \leftarrow \{0,1\}^n}[((h(s_0), < r_1, s_0 >)) = ((h(s_0), < r_2, s_0 >))]$$

$$= Pr_{s_0 \leftarrow \{0,1\}^n}[< r_1, s_0 >=< r_2, s_0 >] = \frac{1}{2^n}$$

So with : $R \sim U_n$ a n-source and $h'$ an UHF, will apply LHL:

$$\Delta(((h(s), r, < r, s > \oplus 0), (U_{n/2}, r, U_1))$$

$$= \Delta(((h(s), r, < r, s >), (U_{n/2}, r, U_1)) \leq \frac{1}{2}\sqrt{2^{\frac{n}{2}+1-n}} = \frac{\sqrt{2}}{2}2^{\frac{-n}{4}} = negl(n)$$

i.e. $C$ is statistically hiding.

C is computationally binding:

Assume $C$ not computationally binding.
i.e. there exists $A$ that can produce $(s, r), (s', r')$ s.t. :
$C_{r,s}(0) = C_{r',s'}(1)$ with non-negligible probability.
In particular $A$ finds $s, s'$ s.t. $h(s) = h(s')$ with non-negligible probability.
In contradiction to the assumption that $h$ is a CRHF.
So from the contradiction we get: $C$ is computationally binding.

# 3 . Is Factoring NP Complete

## 3 .1 Equivalence to Factoring

Denote $L = \{(N, M) \in \mathbb{N} \times \mathbb{N} \mid N \text{ has a prime factor larger than } M\}$.
( I assumed throughout that by 'larger than' we mean > ).

Show: can factor in poly $\iff$ can decide L in poly.

proof left to right $\Rightarrow$

Assume $\exists$ poly $A$ s.t. given $N$ outputs $(p_1, p_2, \ldots, p_m)$ s.t. p_1,p_2,...,p_m are the prime factors of $N$ with repitition, i.e. $N = \Pi_{i=1}^{m} p_i$.

Construct $M_L$ to decide $L$:

Given $(N, M)$, $M_L$ runs $A(N)$ to get $(p_1, p_2, \ldots, p_m)$.
(Should mention output is of length $O(2ml)$ when $l$ is the max length entry in output sequence.
$l = O(log(N)), m = O(log(N)) \Rightarrow$ length of output is $O(log^2(N))$ which is $O(n^2)$ for $n$ length of input)

Now $M_L$ checks if $max_i(p_i) > M$ and returns YES/NO accordingly.

correctness: is immediate from $A$'s correctness.
time: running $A(N)$ takes poly time, running over list and checking > M for each entry is $O(length(list))$ which is also in poly.

proof right to left $\Leftarrow$
Assume $\exists$ poly $M_L$ s.t. $L(M_L) = L$.
Construct $A$ to factor in poly time.
$A$ in pseudo code:

In [2]:

```
def A(N):
    factors = [] # with repititions
    while N > 1:

        # binary search for M in 2,...,N-1 the first number M for which M_L((N,M)) = YES
        M = search(range(2,N))

        factors.append(M)
        N = N/M
```

correctness:
Let $N = \Pi_{i=1}^m p_i$ be the prime factorization of $N$.
Let $(p_1, p_2, \ldots, p_m)$ be their sequence with repititions s.t. $\forall i \in [m-1] : p_{i+1} \leq p_i$.
Denote $N_k = \Pi_{i=1}^k p_i$ $(N = N_m)$.

By induction will prove: in the i'th iteration the variable $N$ turn from $N_{m-i+1}$ to $N_{m-i}$ and $p_{m-i+1}$ is added to the list.

Base: $N_m = N$

Step: The first integer for which $M_L$ will return NO is the biggest prime to divide $N_{m-i+1}$ i.e. $p_{m-i+1}$.
Assume it returns NO before, then $p_{m-i+1} \nmid N_{m-i+1}$ in contradiction.
Assume it does not return YES for $M = p_{m-i+1}$, then exists a prime factor for $N_{m-i+1}$ bigger than p_{m-i+1} in contradiction.

From the induction we got $N_{m-m} = \frac{N}{N} = 1$ so we stop in the $m$'th iteration.
By the fact that in each iteration $i : p_{m-i+1}$ was added to the list, then in the exit from the while loop the list contains the prime factorization of $N$.

time:
Denote by $n$ : length of $N$'s encoding i.e. $log(N)$.
We saw that the while block terminates after $m$ loops.

Each containing a binary search which runs in $O(log(N)) = O(n)$ calls to $M_L$ with O(n) input.
In addition each iteration of the while loop contains some $O(1)$ operations.
Overall time complexity:

$$O(m) \cdot O(log(N)) \cdot [O(p(log(N))) + O(1)]$$

$$= O(log(N)) \cdot O(log(N)) \cdot [O(p(log(N))) + O(1)]$$

$$= O(n) \cdot O(n) \cdot [O(p(n)) + O(1)]$$

# 3 .2 coNP

$L \in NP$ : Will show existance of $R : \Sigma^* \times \Sigma^*$ such that $R$ is polynomialy bounded and $L = \{x | \exists y : (x, y) \in R\}$.

Choose
$R = \{((N, M), (p_1, \dots, p_m)) \mid (N, M) \in L \text{ and } (p_1, \dots, p_m) \text{ is the prime factorization of N } \}$

Explained in 3.1 why $(p_1, \dots, p_m)$ is polynomialy bounded.

Show $R \in P$. Construct $M_R$ to decide $R$ in poly time:
First $M_R$ checks that all $p_i$'s are prime (known to be in poly).
Take product $\Pi_{i=1}^{m} p_i$ and verify equals to N. (if not return NO).
Compare $M$ to largest element in $(p_1, \dots, p_m)$ return YES if $M$ is smaller, else return NO.

$L(M_R) = L$ :

$$((N, M), (p_1, \dots, p_m)) \in R$$

$$\iff (N, M) \in L \wedge N = \Pi_{i=1}^{m} p_i \wedge \forall i \in [m] : p_i \text{ is prime}$$

$$\iff M < max_i(p_i) \wedge N = \Pi_{i=1}^{m} p_i \wedge \forall i \in [m] : p_i \text{ is prime}$$

$$\iff ((N, M), (p_1, \dots, p_m)) \in L(M_R)$$

$M_R$ stops in poly time as is checks factorization which takes O(n) steps of prime checking in O(p(n)) and O(n) product operations. plus an O(1) equality check and O(1) comparison of $p_{max}$ and $M$.

$L \in coNP$ : i.e. $\bar{L} \in NP$.
That follows from a similar proof with
$R = \{((N, M), (p_1, \dots, p_m)) \mid (N, M) \in \bar{L} \text{ and } (p_1, \dots, p_m) \text{ is the prime factorization of N } \}$
And the machine which decides it is similar to $M_R$ with the exception of returning according to the comparison $M \leq max_i(p_i)$ (insead of $>$).

Deducing: $L$ is $NP$ complete $\implies NP \subseteq coNP$:
Assume $L$ is $NP$ complete $\implies L$ is harder than all $L' \in NP$.
$L \in coNP$.

Let $L'$ be some language in $NP$.

$L' \leq_P L$
$\implies \bar{L}' \leq_P \bar{L}$
$\implies \bar{L}' \in NP$
$\implies L' \in coNP$

thus $NP \subseteq coNP$