TOPIC 1. Factoring & TFNP subclasses

presenter : Yuriko Nishijima
supporter : Akash Kumar

Agenda :

- General Overview

paper 1

- defs of PPA (LONELY), PPP (PIGEON)
- Theorem 5 (reduce factoring specific kind of ints to LONELY) + proof
- Theorem 6 (reduce factoring another specific kind of ints to PIGEON + randomness) + proof

paper 2

- defs of Legendre & Jacobi symbol
- defs of diff kinds of factoring problems
- Theorem 3.5 (FACROOTMUL $\in$ PWPP) + proof
- Lemmas to make Thrm 3.5 more interesting
- Theorem 3.7 (Factoring $\subseteq$ PPA)

- Overall Conclusion

Paper 1 : On the TFNP Complexity of Factoring
by Buresh-Oppenheim

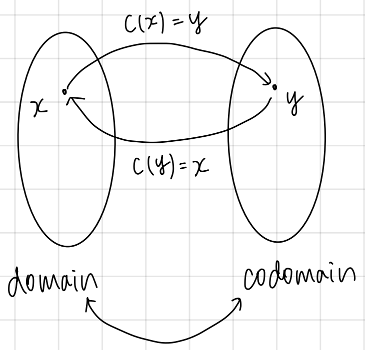
- total (solution always exists for any input problem)
- solution verifiable in polynomial time

$\boxed{\text{Main Idea}}$: We can reduce factoring problem of specific kinda integers to PPA (LONELY) + more general integers (in comparison with) to PPP (PIGEON) + using randomness

subclass of TFNP

Def. PPA is the class of search problems in TFNP that are polytime reducible to LONELY problem

two strings $x \neq y$ are matched if $C(x) = y$ and $C(y) = x$ "x/y is matched"



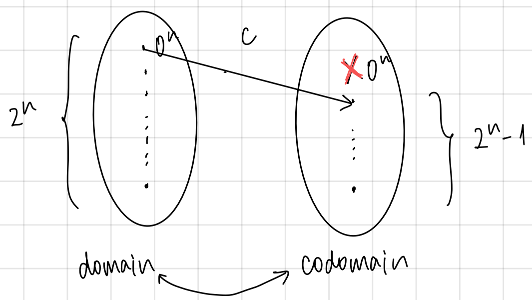
do pairing / matching / mapping on odd # elements

Def. (total NP search problem) LONELY : Given a circuit $C : \{0,1\}^n \to \{0,1\}^n$, find either

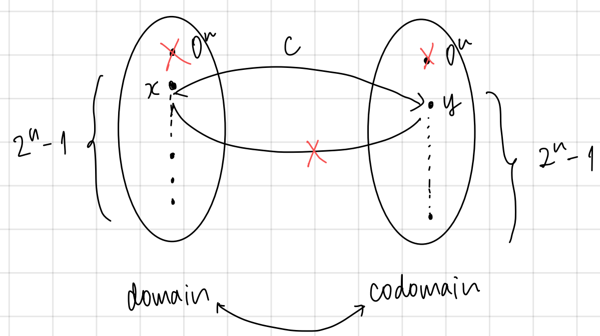
(i) $x \in \{0,1\}^n \setminus \{0^n\}$ s.t. $C(0^n) = x$

"0^n is not necessarily unmatched, easy case" !! ☹
(could be $C(0^n) = x$ and $C(x) = 0^n$, then it's a match)
(or in other words, 0^n has to be taken away from domain, to make the # elements to be paired odd)

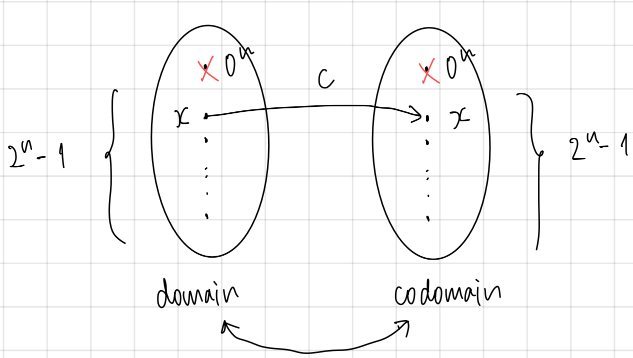
⇒ condition of the circuit is not satisfied
⇒ we don't need to find an unmatched element



(ii) $x, y \in \{0,1\}^n \setminus \{0^n\}$ s.t. $C(x) = y$ but $C(y) \neq x$ "unmatched string" !! ☺



(iii) $x \in \{0,1\}^n \setminus \{0^n\}$ s.t. $C(x) = x$ "another unmatched string" !! ☺



you can flip the domain and codomain

Def. good integer N': odd, positive integer s.t. $\nexists x$ s.t $x^2 = -1 \pmod{N'}$

$\leftrightarrow$ -1 is not a quadratic residue mod N'

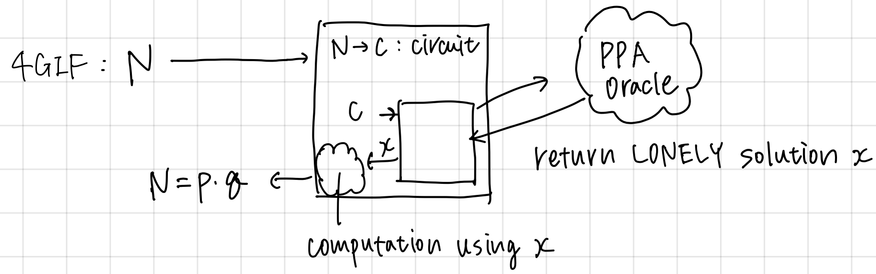
Def. 4good integer N: $\uparrow$ requirement + is $N \equiv 1 \pmod 4$

$\hookrightarrow$ NP search problem (basically factoring 4good integers except for some cases)

4Good-Integer-Factoring (4GIF): Given a positive integer N, return
(i) N, if $N \neq 1 \bmod 4$ or if N is prime or if $N = 1$; $\leftarrow$ cases when you don't have to factor N (excluding these cases cuz it's too hard ?)
(ii) a non-trivial factor of N or a square root of -1 modulo N, otherwise.

Theorem 5 4GIF $\in$ PPA (4GIF is reduciable to instance of LONELY problem)

Proof (reduction) sketch:



proof. let N be a 4good integer.

We know

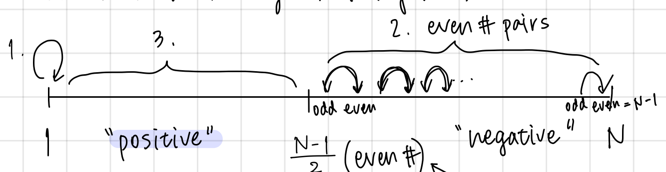
- N is an integer that's not prime
- $N \equiv 1 \pmod 4$ $\rightarrow$ odd

let $n = \log N$ (i.e. N can be represented as n-bit binary string)

We create the following correspondence:

group element x (mod N)	n-bits binary strings
1	0^n
any #	binary representation of it -1

We create a matching of integers $x \in [1, N]$ (= create circuit C that finds a matching)



because $N \equiv 1 \pmod 4$

$\leftrightarrow N = 4k+1$

$\rightarrow \frac{N-1}{2} = \frac{\cancel{4k+1-1}}{2} = 2k \rightarrow$ even #

1. if $x = 1$, matches to itself.

2. if $x > \frac{N-1}{2}$: $x+1 \,/\, x-1$

matches with an integer next to it $\Rightarrow$ there's gonna be even # of pairs.

※ reason why half range
$(N-1)^2 \equiv 1 \bmod N$
$N^2 - 2N + 1 \equiv 1 \bmod N$
can just return $N-1$ if it's full-range
but it'll break computation in (3)

3. if $1 \leq x \leq \frac{N-1}{2}$

if x is unit (i.e. invertible): matches with the inverse

" not unit (i.e. non-invertible): matches with itself $\leftarrow$ assume that you get a solution to this circuit

Let x is a solution to LONELY problem, which can be categorized into type (i) ~ (iii)

We know that x is in "positive" because x that lands in "negative" portion are all perfectly matched.
nor it is 1 because 1 maps to itself by definition.

if $y = |x^{-1}|$ then $x = |y^{-1}|$ (i.e. x and y are the inverse to each other mod N)

so there are no solutions in LONELY type (ii)

$\Downarrow$

All the solutions are in LONELY type (iii)

$\downarrow$

Any # (other than 0) matched with itself $x \rightarrow x = |x^{-1}|$

$\rightarrow$ not invertible

$\rightarrow x^2 \equiv \pm 1 \pmod N$

$x = x^{-1}$ $\quad$ $x = -x^{-1}$

$\downarrow \times x$ $\downarrow \times x$ $\quad$ $\downarrow \times x$ $\downarrow \times x$

$\boxed{x^2 = 1 \qquad x^2 = -1}$

Given x, we can do the following computation to factor 4good integers :

1) if $x^2 = -1 \pmod N$
   return x (this is a N we don't have to consider)

$\downarrow$

2) if $\gcd(x, N) \neq 1$ (i.e. x and N are not co-prime to each other / x is not invertible mod N)
   return $\gcd(x, N)$ (using extended Euclid Algorithm, this is computable in polytime)
   ↳ non-trivial factor of N ✓

$\downarrow$

3) if $x^2 = 1 \pmod N$,
   $\Rightarrow x^2 - 1 = 0$ divisible by N ($\equiv 0 \pmod N$)
   $\Rightarrow (x+1)(x-1) = 0$ and we know $|x| \neq 1$
   return $\gcd(x+1, N)$
   ↳ non-trivial factor of N ✓

$\therefore$ 4GIF $\in$ PPA (4GIF is reduciable to instance of LONELY problem) $\square$

Def. PPP is the class of search problems in TFNP that are polytime reducible to PIGEON

Def. PIGEON : Given $C : \{0,1\}^n \to \{0,1\}^n$ (circuit)

return either (i) $x \in \{0,1\}^n$ s.t. $C(x) = 0^n$ (solution)

or (ii) $x_1, x_2 \in \{0,1\}^n$ s.t. $x_1 \neq x_2$ and $C(x_1) = C(x_2)$

(or both)

e.g. $f(x) = 0^n$ / $f(x) = x0$ (appending 0 at the end)

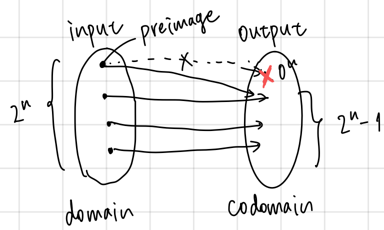
proof (of why PPP $\subseteq$ TFNP)

if (i) holds, this problem is verifiable in polytime and total, hence TFNP ✓

if (i) doesn't hold, (meaning there is no preimage that maps to 0^n (any arbitrary output you choose works)

then domain size $= 2^n$

codomain size $\leq 2^n - 1$



$\rightarrow$ By P.h.P,

$\exists\, x_1 \neq x_2 \quad C(x_1) = C(x_2)$

Def. FP : class of search problems in FNP that are solvable by deterministic Turing Machine

↳ covered in last class but slightly different definitions

↳ def. TFP = FP $\cap$ TFNP : total version of FP

Def. FZPP : class of search problems in FNP that are solvable by randomized Turing Machine

↳ you can toss coins

↳ def. TFZPP = TZPP $\cap$ TFNP : total version of FZPP

Def. good integer N : odd, positive integer s.t. $\nexists x$ s.t $x^2 = -1 \pmod N$

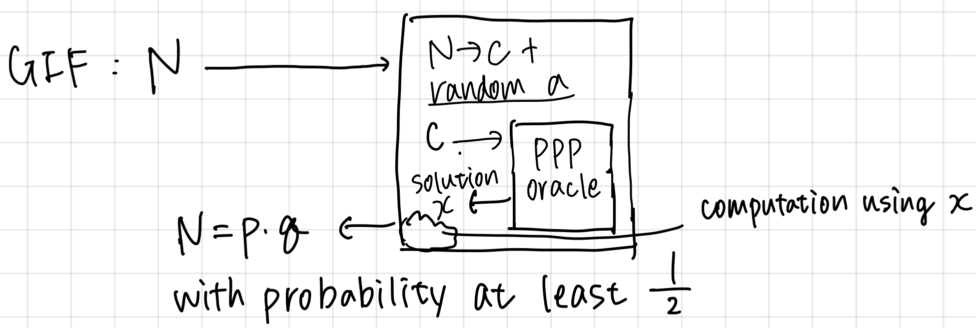
$\leftrightarrow$ '-1 is not a quadratic residue mod N'

↳ NP search problem with restriction

Good-Integer-Factoring (GIF): Given a positive integer N, return
(i) N, if N is prime or if $N = 1$;
(ii) a non-trivial factor of N or a square root of -1 modulo N, otherwise.

Theorem 6 GIF $\in$ TFZPP^{PPP} (GIF can be reduce to an instance of TFZPP with PPP oracle)

Proof (reduction) sketch :



proof. Let N be a good integer that we want to factor.

We know that
- N is composite
- N is odd

(return N
" 2 if given input doesn't satisfy this condition)

Choose a number $a \in \{1, \dots, \cancel{N}\}$ randomly
$\frac{N-1}{2}$ "positive" range

let $n = \log N$ (i.e. N can be represented as n-bit binary string)

We create the following correspondence:

#s	n-bits binary strings (# of them = $x-1$)
-1	0^n
any # $(1, 2, \dots, N)$	binary representation of it

Now, we create a mapping of integers $x \in [1, N]$ (construct a circuit)
- if $x = -1$, then x to a $\leftrightarrow$ $C(-1) = a$ ... case 1
- if $x > \frac{N-1}{2}$, then x to x $\leftrightarrow$ $C(x) = x$ ... " 2
- if $x \leq \frac{N-1}{2}$, then x to $|x^2|$ $\leftrightarrow$ $C(x) = |x^2|$ ... " 3

Recall we are allowed to have access PPP oracle, which gives you a solution to a PIGEON problem
$x, y \in \{0,1\}^n$ s.t. $C(x) = C(y)$
bc it's not possible for C to output -1, all oracle solutions you'd get is of type (ii)

Let us analyze possible solutions x, y that PPP oracle would return and the probability distribution of them
- Case 1 : one of the solution (let's say x) is -1
  then x maps to a : $C(x) = a$ and $C(y) = a = |y^2|$ (case 3)
  $\rightarrow$ $\exists y$ s.t. $y^2 = a$ or $y^2 = -a \pmod N$
  (i.e. a or $-a$ is a q.r)

  ※ This is a bad case, which happens w/p less than $\frac{1}{2}$
  → if it happens, we will have to repeat the whole process with new random a

  WHY?

  def. # of q.r. of N
  $= \frac{\ell(N)}{2^k}$ ← # of distinct primes (which is at least 2)

  since there are a and $-a$

  $$\frac{\ell(N)}{2^k} = \frac{(p-1)(q-1)}{4} \Rightarrow \text{prob}(n.q.r) = \frac{\frac{(p-1)(q-1)}{4} \times 2}{N} = \frac{\frac{(p-1)(q-1)}{2}}{p \cdot q} \leq \frac{1}{2}$$

- Case 2 : Neither of x, y is $=-1$ and $x \neq y$
  then $C(x) = C(y) = z$

  ※ This is a good case, which happens w/p more than $\frac{1}{2}$

Given x and y, we can do the following computation to factor good integers :

1) if $x^2 = -y^2 \pmod N$ then N is not good so
   return xy^{-1} (= square root of -1)

$\downarrow$

2) if $\gcd(y, N) \neq 1$ (i.e. y and N are not co-prime to each other / y is not invertible mod N)
   return $\gcd(y, N)$ (using extended Euclid Algorithm , this is computable in polytime)
   ↳ non-trivial factor of N ✓

$\downarrow$

3) if $x^2 = y^2 \pmod N$
   $\Rightarrow x^2 - y^2 = 0$
   $\Rightarrow (x+y)(x-y) = 0$ but since we know x, y are positive and $x \neq y$
   return $\underline{\gcd(x+y, N)}$
   ↳ non-trivial factor of N ✓

$\therefore$ GIF $\in$ TFZPP^{PPP} (GIF can be reduce to an instance of TFZPP with PPP oracle) $\square$

Open Problem

if ERH (Extended Riemann Hypothesis) is true,
it's guaranteed that you can find a, which is n.q.r $\in [1, \overset{n^2 \,\|}{\log^2(N)}]$

$\Downarrow$

small range

$\Downarrow$

can test them all in a brute-force way

Paper 2: Integer Factoring and modular square roots by Jerabek

Main Idea:

special case of factoring problem

Theorem 3.5: FACROOTMUL problems reduces to an instance of WEAK-PIGEON problem.

improvement from Theorem 5 in Paper 1

Theorem 3.7: Factoring problem reduces to an instance of LONELY problem with randomness.

significant improvement from Theorem 6 in Paper 1

Def. Legendre and Jacobi symbols

Given $a \in \mathbb{Z}$, and p odd prime

we define the Legendre symbols $\left(\frac{a}{p}\right)$ as

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } p \mid a \\ 1 & \text{if } p \nmid a \text{ and } a \text{ is a quadratic res. mod } p \\ -1 & \text{if } p \nmid a \text{ and } a \text{ is a quadratic non res. mod } p \end{cases}$$

prime

applys universally

$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \pmod{p}$ by Euler's criterion

you can also use this p^{α} (power of primes, special composite)

Given $a \in \mathbb{Z}$, and $n \geq 1$ odd number, $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$, p_i odd prime

we define Jacobi symbols $\left(\frac{a}{n}\right)$ as

composite

$$\begin{cases} \left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \cdot \left(\frac{a}{p_2}\right)^{\alpha_2} \cdots \left(\frac{a}{p_k}\right)^{\alpha_k} \\ \left(\frac{a}{1}\right) = 1 \end{cases}$$

$[a] \in \mathbb{Z}_n^*$

a is a q.r mod n

$\leftrightarrow$ a is a quadratic res modulo p_i $\forall i$

Legendre symbols
denominator: prime

$$\left.\begin{cases} = 1 & : a \text{ q.r.} \\ = -1 & : a \text{ q. non r.} \end{cases}\right\}$$ can apply Euler's criterion

Jacobi symbols
denominator: composite

$$\begin{cases} = 1 & : \text{no information} \\ = -1 & : a \text{ q. non r.} \end{cases}$$

e.g. $n = pq$ $\quad \left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$

$35 = 5 \cdot 7$

if $\left(\frac{a}{p}\right) = 1$, $\left(\frac{a}{q}\right) = 1 \Rightarrow a$ is a q.r. mod n and $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right) = 1$

if $\left(\frac{a}{p}\right) = -1$, $\left(\frac{a}{q}\right) = -1 \nRightarrow a$ is a q. non. r mod n and $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right) = 1$

Different kinds of factoring problems

Jacobi symbol

Def. FACROOT problem: given an odd integer $n > 0$ and integer a s.t. $\left(\frac{a}{n}\right) = 1$,
find either a nontrivial divisor of n, or a square root of a mod n.

Def. FACROOTMUL problem: special cases of FACROOT. Given odd $n > 0$ and integers a and b,
find a nontrivial divisor of n, or a square root of one of a, b or ab mod n.

Def. WEAKFACROOT problem: given an odd $n > 0$ and a, b s.t. $\left(\frac{a}{n}\right) = 1$ and $\left(\frac{b}{n}\right) = -1$,
find a nontrivial divisor of n, or a square root of a mod n.

cop-out case

Theorem 3.5: FACROOTMUL $\in$ PWPP i.e.
FACROOTMUL problems reduces to an instance of WEAK-PIGEON problem.

improvement from Theorem 5 in Paper 1

proof. Assume n = odd int. a, b = int.

if $\gcd(a, n) \neq 1$ or $\gcd(b, n) \neq 1$, return (n, a) or (n, b) respectively.
※ we can assume that both a, b are coprime to n.

let $f : \underbrace{\{0, 1, 2\}}_{=3} \times [\underbrace{1, \tfrac{n-1}{2}}_{\frac{n}{2}}] \to [\underbrace{1, n-1}_{n}]$:

domain size — range size

$$f(\underbrace{i, x}_{3 \times 2}) = \begin{cases} a_i x^2 \bmod n & \text{if } (n, x) = 1 \\ x & \text{otherwise} \end{cases} \quad \cdots ①$$

combination of 2 inputs

where $a_0 = 1$, $a_1 = a$, $a_2 = b$.

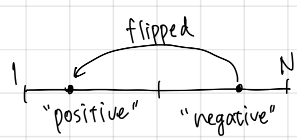
Since the size(domain) = size(range) $\times \frac{3}{2}$,
we can use WEAKPIGEON to find a collision: $f(i, x) = f(i, y)$ and $\langle i, x \rangle \neq \langle i, y \rangle$

Now, assume you get a solution x, y to WAAKPIGEON problem:

(you can assume that $\gcd(n, x) = \gcd(n, y) = 1$ as otherwise we can factor n.)
→ if not just return $\gcd(n, x)$ or $\gcd(n, y)$

Given $\langle i, x \rangle \neq \langle j, y \rangle$, there are 2 cases:

Case 1: $i = j$ (this is a good case, bc you can actually factor N)



then you must have $x \not\equiv y \pmod n$
also $x \not\equiv -y \pmod n$ (bc solution resides in "positive" range)
(i.e. first half of [1, N-1]

$$\begin{aligned} f(i, x) &= f(j, y) \quad \text{(collision)} \\ \Leftrightarrow \cancel{a_i} x^2 &= \cancel{a_j} y^2 \quad \text{by ①} \\ x^2 &= y^2 \\ x^2 - y^2 &= 0 \\ (x+y)(x-y) &= 0 \end{aligned}$$

since $x \neq y$, $\neq 0$

↳ return $\gcd(x-y, N)$, which is a nontrivial factor of N.

Case 2 : $i \neq j$ (returning cop-out values)

For simplicity, assume $i < j$

$$f(i, x) = f(j, y) \quad \text{(collision)}$$

$$\Leftrightarrow a_i x^2 = a_j y^2 \quad \text{by } ①$$

$$\downarrow \times (y^{-1})^2 \qquad \downarrow \times (y^{-1})^2$$

$$a_i (xy^{-1})^2 = a_j$$

$$\underline{(xy^{-1})^2 = a_j a_i^{-1}}$$

Recall $a_0 = 1$, $a_1 = a$, $a_2 = b$

1) $i = 0$, $j = 1$

$$a_1 a_0^{-1} = (xy^{-1})^2$$
$$a \cdot \cancel{1^{-1}} = (xy^{-1})^2$$
$$a = (xy^{-1})^2$$
$$\Rightarrow \text{return } xy^{-1}$$

2) $i = 1$, $j = 2$

$$a_2 a_1^{-1} = (xy^{-1})^2$$
$$ba^{-1} = (xy^{-1})^2$$
$$\downarrow \times a^2 \qquad \downarrow \times a^2$$
$$ab = (axy^{-1})^2$$
$$\Rightarrow \text{return } axy^{-1}$$

3) $i = 0$, $j = 2$

$$a_2 a_0^{-1} = (xy^{-1})^2$$
$$b \cancel{1^{-1}} = (xy^{-1})^2$$
$$b = (xy^{-1})^2$$
$$\Rightarrow \text{return } xy^{-1}$$

we are allowed to return <u>square root of one of a, b or ab mod n.</u> as FACTROOTMUL defines.

Here's another finding of this paper that makes Theorem 3.5 more interesting ...

Lemma 3.2

(*i*) $\text{WeakFacRoot} \underbrace{\le_m}_{\text{hardness}} \text{FacRootMul} \le_m \text{FacRoot};$

$\xrightarrow{\text{reduces}}$ harder problem

proof. WEAKFACROOT is a special case of FACROOTMUL since

※ note that Jacobi symbol is multiplicative.

$$\left(\frac{a}{n}\right) = 1 \text{ and } \left(\frac{b}{n}\right) = -1 \Rightarrow \left(\frac{a}{n}\right) \times \left(\frac{b}{n}\right) = 1 \times -1 = -1 \Rightarrow \left(\frac{ab}{n}\right) = -1$$

$\Rightarrow$ neither b nor ab is a q.r. mod n

For an instance of FACTROOTMUL problem

$\left(\frac{x}{n}\right) = 1$ for some $x \in \{a, b, ab\}$ $\rightarrow$ we can choose such an x as the Jacobi symbol is poly-time computable.

$=$ no information about the qudratic reciprocity.

Lemma 3.3

(*iii*) FACTORING $\leq_m^{RP,1/2}$ WEAKFACROOT.

proof (skip ?)

(iii): FACROOT $\leq_m^{RP}$ WEAKFACROOT by (i) and amplification of the success rate of $\leq_m^{ZPP}$, hence FACTORING $\leq_m^{RP,1/2+\varepsilon}$ WEAKFACROOT for any $\varepsilon > 0$ by (ii). We can get rid of the ε by observing that the proof of (ii) actually shows FACTORING $\leq_m^{RP,1/2-1/\sqrt{n}}$ FACROOT, taking into account residues that share a factor with n. We can reduce the error of the $\leq_m^{ZPP}$ reduction in (i) to $1/\sqrt{n}$, hence FACTORING $\leq_m^{RP,1/2}$ WEAKFACROOT. □

We remark that there is another well-known randomized reduction of factoring to square root computation modulo n due to Rabin [15], but it is suited for a different model. In the notation above, the basic idea of Rabin's reduction is that we choose a random $1 < a < n$, and if it is coprime to n, we pass n, a^2 to the FACROOT oracle. If the oracle were implemented as a (deterministic or randomized) algorithm working independently of the reduction without access to its random coin tosses, we would have a 1/2 chance that the root b of a^2 returned by the oracle satisfies $a \not\equiv \pm b$ (n), allowing us to factorize n. However, this does not work in our setup. According to the definition of a search problem reduction, the reduction function must be able to cope with *any* valid answer to the oracle query—there is no implied guarantee that oracle answers are computed independently of the environment. In particular, it may happen the oracle is devious enough to always return the root $b = a$ we already know.

What we need now is to show that FACROOT or some of its variants belongs to PPA and PWPP.

Summary so far :

weak-pigeon pigeon

FACTORING $\leq^{RP,\frac{1}{2}}$ WEAKFACROOT $\leq$ FACROOTMUL $\in$ PWPP $\in$ PPP ... ①

Lemma 3.3 (iii) Lemma 3.2 (i) Theorem 3.5

Another result of this paper ...

Theorem 3.7

(*i*) FACTORING, FULLFAC $\leq_m^{RP}$ PPA; .. ②

(proof omitted)

By ① and ②
we can conclude that

FACTORING problem reduces to both PWPP + randomness (w/p $\geq \frac{1}{2}$)
PPA + randomness

as the state of the art in the factoring & complexity research field.