3. Factorization. Primality testing.

A. Ushakov

MA503, September 21, 2022

Contents

Here we discuss some important details of RSA implementation.

- Simple primality test.
- Fermat primality test.
- Fermat pseudoprimes.
- Carmichael numbers.
- Miller–Rabin primality test. Efficiency.
- Random prime generation.
- Shared primes.
- Pollard's $p - 1$ method to factor $N = pq$.
- Factorization via difference of squares.
- Pollard's rho algorithm.

Simple primality test

Primality problem *is an algorithmic question to decide if a given n is prime or not.*

A **primality test** *is an algorithm to solve primality problem.*

- There is a polynomial-time primality test, called **AKS-primality test**.
- There are even more efficient simple randomized tests, e.g., **Fermat test**, **Miller-Rabin test**, etc.

Lemma

An integer $n > 1$ is composite $\Leftrightarrow$ *n is divisible by some prime $p \leq \sqrt{n}$.*

$$\begin{aligned} n \text{ is composite} &\Leftrightarrow n = ab && \text{for some } 1 < a, b < n \\ &\Leftrightarrow \min(a, b) \mid n && \text{where } \min(a, b) \leq \sqrt{n} \end{aligned}$$

(Simple primality test)

Enumerate numbers $2, \ldots, \lfloor \sqrt{n} \rfloor$. If one number divides n, then output No. Otherwise, output Yes.

- This algorithm terminates very fast when n has a small divisor.
- It is very inefficient if n is prime.

Fermat primality test

Suppose that $\gcd(a, n) = 1$. By Fermat's little theorem

$$n \text{ is prime} \quad \Rightarrow \quad a^{n-1} \equiv_n 1.$$

(Contrapositive) $a^{n-1} \not\equiv_n 1 \quad \Rightarrow \quad n$ **is not prime**.

(Fermat primality test)

(1) *Generate a random* a *such that* $1 < a < n$.

(2) $a^{n-1} \not\equiv_n 1 \quad \Rightarrow \quad$ *output No.*

(3) *Otherwise output ProbablyYes.*

Fermat primality test can recognize composite n only. It does not solves the primality problem completely. In fact, it can fail to recognize a composite number too.

Show that $n = 6$ *is not prime using Fermat primality test.*

Generate a random unit mod 6, e.g., $a = 5$, and observe that $a^{n-1} = 5^5 \equiv_6 5 \not\equiv_6 1$.

Show that $n = 35$ *is not prime using Fermat primality test.*

Generate a random unit mod 35, e.g., $a = 2$, and observe that $2^{34} \equiv_{35} 9 \not\equiv_{35} 1$;

Fermat pseudoprime

Fermat primality test used with a is called **base-a test**.

Definition

A number $n > 2$ is called a **Fermat pseudoprime to base a** if

- n is composite;
- $a^{n-1} \equiv_n 1$ and, hence, base-a test fails to recognize that n is composite.

Fermat pseudoprime is a Fermat pseudoprime to base 2.

E.g., $341 = 31 \cdot 11$ is a Fermat pseudoprime because it fails the base-2 test:

$$2^{340} = (2^{10})^{34} = 1024^{34} \equiv_{341} 1^{34} = 1.$$

Theorem (no proof)

There are infinitely many base-2 pseudoprimes.

Idea: if n is pseudoprime, then $x = 2^n - 1$ is a pseudoprime too.

Fermat primality test can be performed in nearly-quadratic time $\widetilde{O}(\log^2(n))$.

Carmichael numbers

Q. Is it true that for any composite n there exists an appropriate base a such that the base-a test shows that n is indeed composite.
A. No. There are composite n that fail base-a test for every a. Such numbers are called **Carmichael numbers**.

Definition

A number $n > 2$ is called a **Carmichael number** if

- n is composite;
- $a^{n-1} \equiv_n 1$ for every $a \in \mathbb{Z}$.

Example. Show that 561 *is Carmichael.*

$561 = 3 \cdot 11 \cdot 17$ is composite and satisfies $a^{560} \equiv_{561} 1$ for every $a \in \mathbb{Z}$ because

$$a^{560} \equiv_{561} 1 \Leftrightarrow \begin{cases} a^{560} \equiv_3 1 & \text{(holds because } a^2 \equiv_3 1 \text{ by FLT)} \\ a^{560} \equiv_{11} 1 & \text{(holds because } a^{10} \equiv_{11} 1 \text{ by FLT)} \\ a^{560} \equiv_{17} 1 & \text{(holds because } a^{16} \equiv_{17} 1 \text{ by FLT)} \end{cases}$$

Theorem (no proof)

There are infinitely many Carmichael numbers.

Miller–Rabin primality test

Later we will prove that 1 has exactly two square roots ± 1 modulo a prime number p. Modulo a composite number 1 can have more than two square roots.

Proposition

Let p be an odd prime. Express $p-1$ as $p-1=2^k q$, where q is odd. Then for every a coprime with p one of the following conditions holds:

(1) $a^q \equiv_p 1$,

(2) $a^{2^i q} \equiv_p -1$ *for some* $0 \leq i \leq k-1$.

- Consider the list of powers $a^q, a^{2q}, a^{4q}, \ldots, a^{2^k q}$.
- The last number satisfies $a^{2^k q} = a^{p-1} \equiv_p 1$.
- Each number is a square root of the following number because $(a^{2^i q})^2 = a^{2^{i+1} q}$.
- Hence, for a prime modulus, the list of numbers must end with 1's

$$a^q, a^{2q}, a^{4q}, \ldots, \underbrace{a^{2^i q}, \ldots, a^{2^k q}}_{\text{ones}}.$$

- Case-I: If the whole list is the list of ones, then (1) holds.
- Case-II: Otherwise, the rightmost non-one must be -1 and so (2) holds.

(Miller–Rabin primality test for an odd n)

(1) *Generate a random a such that $1 < a < n$ satisfying $\gcd(a, n) = 1$.*

(2) *Compute q and k such that $n - 1 = 2^k q$.*

(3) *If $a^q \equiv_n 1$, then output ProbablyYes.*

(4) *If $a^{2^i q} \equiv_n -1$ for some $i = 0, \ldots, k-1$, then output ProbablyYes.*

(5) *Otherwise output No.*

Fermat base-a test outputs No $\Rightarrow$ $a^{n-1} \not\equiv_n 1$

$\Rightarrow$ the sequence of roots has no ± 1's

$\Rightarrow$ Miller-Rabin test mod-a outputs No.

Miller–Rabin test is not weaker than Fermat test.

For a Carmichael number $n = 561$, choose $a = 2$, compute $n - 1 = 2^4 \cdot 35$ and a sequence of power

$$\begin{aligned} a^q = 2^{35} &\equiv_{561} 263 \\ 2^{2\cdot 35} &\equiv_{561} 166 \\ 2^{4\cdot 35} &\equiv_{561} 67 \\ 2^{8\cdot 35} &\equiv_{561} 1. \end{aligned}$$

Hence, the algorithm outputs No.

Miller–Rabin primality test: efficiency

Definition

A unit a modulo n is called a **Miller–Rabin witness** for the compositeness of n if Miller–Rabin algorithm base-a recognizes n as composite.

Proposition

Let n be an odd composite number. Then at least 75% of the numbers a between 1 and $n-1$ are Miller–Rabin witnesses for n.

Therefore, running the Miller–Rabin base-a algorithm on 10 randomly generated a's recognizes a composite number with probability at least

$$1 - \left(\tfrac{1}{4}\right)^{10} = 0.99999904632568359375.$$

This test has a tiny chance to give a wrong answer. Yet, it is preferred over the AKS test (that makes no mistakes) due to its efficiency.

Miller–Rabin primality test can be performed in nearly-quadratic time $\widetilde{O}(\log^2(n))$.

Generating a random prime

Definition

For $N \in \mathbb{N}$ define $\pi(N)$ to be the number of primes p between 1 and N.

$\pi(2) = 1$	$\pi(5) = 3$	$\pi(8) = 4$	$\pi(11) = 5$	$\pi(14) = 6$	$\pi(17) = 7$
$\pi(3) = 2$	$\pi(6) = 3$	$\pi(9) = 4$	$\pi(12) = 5$	$\pi(15) = 6$	$\pi(18) = 7$
$\pi(4) = 2$	$\pi(7) = 4$	$\pi(10) = 4$	$\pi(13) = 6$	$\pi(16) = 6$	$\pi(19) = 8$

$\frac{\pi(N)}{N}$ is the density of the set of primes in $[1, N]$.

Theorem (Prime number theorem)

$\frac{\pi(N)}{N/\ln(N)} \to 1$ *as* $N \to \infty$.

Therefore, the chance that a randomly chosen number $n \in [1, N]$ is prime is $\frac{\pi(N)}{N} \approx \frac{1}{\ln(N)}$.

Corollary

The chance to that a randomly chosen number $n \in [1, 2^m]$ *is prime is approximately* $\frac{1}{\ln(N)} \approx \frac{1}{\ln(2^m)} = \frac{1}{m \ln(2)}$.

Hence, a sequence of $m \ln(2)$ random numbers has a prime with high probability. So, to generate a prime in $[1, 2^m]$ we can generate $m \ln(2)$ random integers and test if one of them is prime.

Randomness is important!

Poorly generated RSA primes p and q lead to "easy attacks", like

N. Heninger, Z. Durumeric, E. Wustrow, J. Halderman, *Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices*. 2012.

Abstract. *RSA and DSA can fail catastrophically when used with malfunctioning random number generators, but the extent to which these problems arise in practice has never been comprehensively studied at Internet scale. We perform the largest ever network survey of TLS and SSH servers and present evidence that vulnerable keys are surprisingly widespread. We find that 0.75% of TLS certificates share keys due to insufficient entropy during key generation, and we suspect that another 1.70% come from the same faulty implementations and may be susceptible to compromise. Even more alarmingly, we are able to obtain RSA private keys for 0.50% of TLS hosts and 0.03% of SSH hosts, because their public keys shared nontrivial common factors due to entropy problems, and DSA private keys for 1.03% of SSH hosts, because of insufficient signature randomness.*

The authors analyzed a large collection of RSA public keys $\{(n_i, e_i)\}_i$ and realized that for significantly many pairs $i \neq j$

$$1 < \gcd(n_i, n_j) \neq n_i, n_j$$

and, hence, $\gcd(n_i, n_j)$ must be a prime shared by n_i and n_j.

Pollard's $p-1$ method to factor $N = pq$

For any $L \in \mathbb{N}$ and a randomly generated $a \in \mathbb{N}$ coprime with N

$$\begin{array}{l}(p-1) \mid L \\ (q-1) \nmid L\end{array} \quad \Rightarrow \quad \begin{array}{l}L = i(p-1) \\ L = j(q-1)+k\end{array} \quad \Rightarrow \quad \begin{array}{l}a^L = a^{i(p-1)} = \left(a^{p-1}\right)^i \equiv 1 \mod p \\ a^L = a^{j(q-1)+k} = \left(a^{q-1}\right)^j a^k \equiv a^k \mod q.\end{array}$$

Since $k \neq 0$, it is "likely" that $a^k \not\equiv_q 1$, i.e., for most choices of a we have

$$p \mid a^L - 1 \text{ and } q \nmid a^L - 1.$$

which implies that $\gcd(N, a^L - 1) = p$.

Q. *How do we find L satisfying $(p-1) \mid L$ and $(q-1) \nmid L$?*

A. If $p-1$ is a product of small primes, then it divides $L = n!$ for some not-too-large n.

Q. *How do we compute* $\gcd(N, a^{n!} - 1)$*?*

- Compute the sequence $a^{n!} \% N$ using $a^{(n+1)!} = (a^{n!})^{n+1}$.
- Directly compute $\gcd(N, a^{n!} - 1)$.

Pollard's $p-1$ algorithm

Pick a random a s.t. $\gcd(a,N)=1$. For $n=2,3\ldots$

- *compute $d=\gcd(N,a^{n!}-1)$;*
- *if $1<d<N$, then output d;*
- *if $d=N$, then output FAILURE.*

Find a nontrivial factor in $N=13927189$ using Pollard's $p-1$ algorithm:

$$\begin{aligned}
2^{9!} &\equiv_N 13867883 & \gcd(2^{9!}-1,13927189)&=1,\\
2^{10!} &\equiv_N 5129508 & \gcd(2^{10!}-1,13927189)&=1,\\
2^{11!} &\equiv_N 4405233 & \gcd(2^{11!}-1,13927189)&=1,\\
2^{12!} &\equiv_N 6680550 & \gcd(2^{12!}-1,13927189)&=1,\\
2^{13!} &\equiv_N 6161077 & \gcd(2^{13!}-1,13927189)&=1,\\
2^{14!} &\equiv_N 879290 & \gcd(2^{14!}-1,13927189)&=3823.
\end{aligned}$$

Hence, $p=3823$ is a factor in N. The method worked for $n=14$ because $3823-1=2\cdot 3\cdot 7^2\cdot 13$ which divides 14! and does not divide 13!. For the other factor $q=3643$ we have $q-1=2\cdot 3\cdot 607$.

$p-1$ and $q-1$ should not factor into small primes for RSA primes p and q.

Pollard's $p-1$ algorithm: examples

1. Find a nontrivial factor in $N = 35$ using Pollard's $p-1$ algorithm.

$$2^{1!} \equiv 2 \qquad \gcd(2-1, 35) = 1$$
$$2^{2!} \equiv 4 \qquad \gcd(4-1, 35) = 1$$
$$2^{3!} \equiv 29 \qquad \gcd(29-1, 35) = 7.$$

2. Find a nontrivial factor in $N = 3869$ using Pollard's $p-1$ algorithm.

$$2^{1!} \equiv 2 \qquad \gcd(2-1, 3869) = 1$$
$$2^{2!} \equiv 4 \qquad \gcd(4-1, 3869) = 1$$
$$2^{3!} \equiv 64 \qquad \gcd(64-1, 3869) = 1$$
$$2^{4!} \equiv 1232 \qquad \gcd(1232-1, 3869) = 1$$
$$2^{5!} \equiv 81 \qquad \gcd(81-1, 3869) = 1$$
$$2^{6!} \equiv 3651 \qquad \gcd(3651-1, 3869) = 73.$$

3. Find a nontrivial factor in $N = 24341 = 241 \cdot 101$ using Pollard's $p-1$ algorithm.

$$3^{1!} \equiv 3 \qquad \gcd(3-1, 24341) = 1$$
$$3^{2!} \equiv 9 \qquad \gcd(9-1, 24341) = 1$$
$$3^{3!} \equiv 729 \qquad \gcd(729-1, 24341) = 1$$
$$3^{4!} \equiv 12864 \qquad \gcd(12864-1, 24341) = 1$$
$$3^{5!} \equiv 20486 \qquad \gcd(20486-1, 24341) = 241.$$

B-smooth numbers

Definition

m is called *B-smooth* if all of its prime factors are less than or equal to B.

For instance,

- $6600 = 2^3 \cdot 3 \cdot 5^2 \cdot 11$ is 11-smooth;
- $72000 = 2^6 \cdot 3^2 \cdot 5^3$ is 13-smooth (you can also say it is 5-smooth, or 55-smooth);

It is easy to check B-smoothness of n.

- Write down all primes $2, 3, 5, 7, \ldots, p_k$ less than or equal to B.
- Check directly, if n can be expressed as a product of those primes (by dividing).

Factorization of N via difference of squares

$$a^2 \equiv_N b^2 \quad \Rightarrow \quad N \mid a^2 - b^2 \quad \Rightarrow \quad (a-b)(a+b) = a^2 - b^2 = Nq \text{ for some } q \in \mathbb{Z}$$
$$\Rightarrow \quad \text{it is possible that } d = \gcd(a \pm b, N) \neq 1, N$$
$$\Rightarrow \quad d \text{ is a factor in } N.$$

To find a, b we randomly generate pairs (a_i, c_i) such that $a_i^2 \equiv_N c_i$ and c_i is B-smooth.

Example. Find a nontrivial factor of $N = 914387$.

$$\begin{array}{lllllll} 1869^2 \equiv_N & 750000 = & 2^4 & 3 & 5^6 & \\ 1909^2 \equiv_N & 901120 = & 2^{14} & & 5 & 11 \\ 3387^2 \equiv_N & 499125 = & & 3 & 5^3 & 11^3. \end{array}$$

The numbers on the right are not squares. But their product is

$$9835^2 \equiv_N (1869 \cdot 1909 \cdot 3387)^2 \equiv_N 2^{18}3^2 5^{10} 11^4 = (2^9 \cdot 3 \cdot 5^5 \cdot 11^2)^2$$
$$= 580800000^2 \equiv_N 164255^2.$$

Hence, $a = 9835$ and $b = 164255$. Finally, $\gcd(914387, 9835 - 164255) = 1103$ which is a factor in N.

Factorization via difference of squares for N

Example. Find a nontrivial factor in 78391.

$$\begin{aligned} 12515^2 &\equiv_{78391} 7 &&= 2^0\ 3^0\ 5^0\ 7^1 \\ 22869^2 &\equiv_{78391} 44800 &&= 2^8\ 3^0\ 5^2\ 7^1. \end{aligned}$$

$\gcd(78391, 12515 \cdot 22869 - 2^4 \cdot 5 \cdot 7) = 283.$

Example. Find a nontrivial factor in 40301.

$$\begin{aligned} 27756^2 &\equiv_{40301} 1620 = 2^2\ 3^4\ 5^1 \\ 18651^2 &\equiv_{40301} 21870 = 2^1\ 3^7\ 5^1 \\ 11759^2 &\equiv_{40301} 1350 = 2^1\ 3^3\ 5^2 \end{aligned}$$

$\gcd(40301, 27756 \cdot 18651 \cdot 11759 - 2^2 \cdot 3^7 \cdot 5^2) = 40301$ (failure)

$$\begin{aligned} 15629^2 &\equiv_{40301} 1280 = 2^8\ 3^0\ 5^1 \\ 18651^2 &\equiv_{40301} 21870 = 2^1\ 3^7\ 5^1 \\ 11759^2 &\equiv_{40301} 1350 = 2^1\ 3^3\ 5^2 \end{aligned}$$

$\gcd(40301, 15629 \cdot 18651 \cdot 11759 - 2^5 \cdot 3^5 \cdot 5^2) = 1$ (failure)

$$\begin{aligned} 26394^2 &\equiv_{40301} 150 = 2^1\ 3^1\ 5^2 \\ 5388^2 &\equiv_{40301} 13824 = 2^9\ 3^3\ 5^0 \end{aligned}$$

$\gcd(40301, 26394 \cdot 5388 - 2^5 \cdot 3^2 \cdot 5^1) = 191.$

Birthday problem

$1 - x < e^{-x}$ *for every* $x > 0$.

Consider $f(x) = e^{-x} - (1 - x)$ and notice the following:

- $f(0) = 0$;
- $f'(x) = 1 - e^{-x} > 0$ for every $x > 0$ and, hence, $f(x)$ is continuous and increasing.

Therefore, $f(x) > 0$ for every $x > 0$.

Let $x_1, \ldots, x_k$ *be random elements of a set of* n *elements. Then* $\Pr(x_i \neq x_j) < e^{-\frac{k(k-1)}{2n}}$.

$$\Pr(x_i \neq x_j, \quad \forall i \neq j) = \Pr(x_2 \neq x_1) \cdot \Pr(x_3 \neq x_1 \ \& \ x_3 \neq x_2) \cdot \ldots \cdot \Pr(x_k \neq x_1 \ \& \ \ldots \ x_k \neq x_{k-1})$$

$$= \tfrac{n-1}{n} \cdot \tfrac{n-2}{n} \cdot \ldots \cdot \tfrac{n-(k-1)}{n} = \left(1 - \tfrac{1}{n}\right) \cdot \left(1 - \tfrac{2}{n}\right) \cdot \ldots \cdot \left(1 - \tfrac{k-1}{n}\right)$$

$$< e^{-\frac{1}{n}} e^{-\frac{2}{n}} \ldots e^{-\frac{k-1}{n}} = e^{-\frac{k(k-1)}{2n}}.$$

$k > \sqrt{2n\ln(2)} \quad \Rightarrow \quad e^{-\frac{k(k-1)}{2n}} < \frac{1}{2}$.

$$e^{-\frac{k(k-1)}{2n}} < \tfrac{1}{2} \quad \Leftrightarrow \quad -\tfrac{k(k-1)}{2n} < \ln(\tfrac{1}{2}) = -\ln(2) \quad \Leftrightarrow \quad k(k-1) > 2n\ln(2)$$

$$\Leftarrow \quad k = 1 + \sqrt{2\ln(2)n} \approx 1 + 1.177\sqrt{n} = O(\sqrt{n}).$$

$k > \sqrt{2n\ln(2)} \quad \Rightarrow \quad \Pr(x_i = x_j \text{ for some } i \neq j) = 1 - \Pr(x_i \neq x_j) > 1 - e^{-\frac{k(k-1)}{2n}} > \frac{1}{2}$.

Conclusion. A randomly chosen sequence $x_1, \ldots, x_k$ of numbers $x_i \in \{1, \ldots, n\}$ has a collision with probability at least $\frac{1}{2}$ if $k > \sqrt{2n\ln(2)}$.

Pollard's rho algorithm (general purpose factoring)

Suppose that p is the least prime factor in N. Then for any $x, y \in \mathbb{N}$ we have

$$\begin{array}{l} x \equiv_p y \\ x \not\equiv_N y \end{array} \quad \Rightarrow \quad \begin{array}{l} p \mid x - y \\ N \nmid x - y \end{array} \quad \Rightarrow \quad \gcd(x - y, N) \text{ is a non-trivial factor of } N.$$

How can we find such x, y?

Birthday problem: *Generate random numbers $x_1, \ldots, x_k \in [0, N-1]$. If $k = O(\sqrt{p}) = O(\sqrt[4]{N})$, then some $x = x_i, y = x_j$ satisfy the assumption above, with probability at least $\frac{1}{2}$.*

Unfortunately, it would require to check $k^2 = O(\sqrt{N})$ pairs x_i, x_j to find one satisfying $\gcd(x_i - x_j, N) > 1$. So, we will do a trick. We will generate $x_1, x_2, \ldots, x_{2k}$ in a special way which will require testing only k pairs.

For any polynomial $f(x) = c_n x^n + \ldots + c_1 x + c_0$ and a modulus $N \in \mathbb{N}$

$$a \equiv_N b \quad \Rightarrow \quad \begin{array}{ccccccc} c_n a^n & + \ldots + & c_1 a & + & c_0 & = & f(a) \\ \| & & \| & & \| & & \| \\ c_n b^n & + \ldots + & c_1 b & + & c_0 & = & f(b). \end{array}$$

Fix a polynomial $f(x)$ and generate a sequence $x_1, x_2, \ldots$ defined by

- x_1 is randomly chosen in $[1, N-1]$, and
- $x_{i+1} = f(x_i) \,\%\, N$.

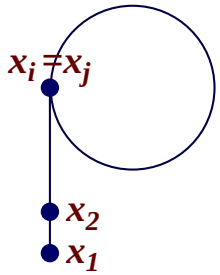
The sequence $x_1, x_2, \ldots$ is not random (only x_1 is random) and does not satisfy assumptions of the birthday problem. Yet, it works in practice and is called **pseudo-random**.

f plays a role of a pseudo-random number generator.

Pollard's rho algorithm

Observe that the sequence $x_1, x_2, x_3, \ldots$

- has a collision (because there are finitely many numbers modulo N)
- is periodic modulo N and modulo p.



$$
\begin{aligned}
x_i \equiv_p x_j \quad &\Rightarrow \quad x_{i+1} = f(x_i) \equiv_p f(x_j) = x_{j+1} \\
&\Rightarrow \quad x_{i+2} = f(x_{i+1}) \equiv_p f(x_{j+1}) = x_{j+2} \\
&\Rightarrow \quad x_{i+3} = f(x_{i+2}) \equiv_p f(x_{j+2}) = x_{j+3} \\
&\text{etc.}
\end{aligned}
$$

Proposition

If $x_i \equiv_p x_j$, then $x_m \equiv_p x_{2m}$ for some $1 \leq m < j$.

If $x_i \equiv_p x_j$, then, $j - i$ is a period of the sequence. Let $q \in \mathbb{Z}_{\geq 0}$ be the least number satisfying $i \leq q(j-i)$. It is easy to see that

- $x_{q(j-i)} = x_{2q(j-i)}$ because $2q(j-i) - q(j-i) = q(j-i)$ is a multiple of a period.
- $q(j-i) < j$, because q is the least satisfying $i \leq q(j-i)$.

(The algorithm)

Use the polynomial function $f(x) = x^2 + 1$ to generate $x_1, x_2, \ldots, x_{2\sqrt[4]{N}}$ and if $\gcd(x_{2i} - x_i, N) = d > 1$, then output d.

Pollard's rho algorithm: examples

1. Find a nontrivial factor in $N = 8051$ using the Pollard's rho algorithm.

$x_1 = 5$ (randomly chosen)

$x_2 = x_1^2 + 1 = 26$ $\quad \gcd(x_2 - x_1, 8051) = \gcd(26 - 5, 8051) = 1$

$x_3 = x_2^2 + 1 = 677$

$x_4 = x_3^2 + 1 = 7474$ $\quad \gcd(x_4 - x_2, 8051) = \gcd(7474 - 26, 8051) = 1$

$x_5 = x_4^2 + 1 = 2839$

$x_6 = x_5^2 + 1 = 871$ $\quad \gcd(x_6 - x_3, 8051) = \gcd(871 - 677, 8051) = 97.$

2. Find a nontrivial factor in $N = 3763$ using the Pollard's rho algorithm.

$x_1 = 2$

$x_2 = 5$ $\quad \gcd(x_2 - x_1, 3763) = \gcd(5 - 2, 3763) = 1$

$x_3 = 26$

$x_4 = 677$ $\quad \gcd(x_4 - x_2, 3763) = \gcd(677 - 5, 3763) = 1$

$x_5 = 3007$

$x_6 = 3324$ $\quad \gcd(x_6 - x_3, 3763) = \gcd(3324 - 26, 3763) = 1$

$x_7 = 809$

$x_8 = 3483$ $\quad \gcd(x_8 - x_4, 3763) = \gcd(3483 - 677, 3763) = 1$

$x_9 = 3141$

$x_{10} = 3059$ $\quad \gcd(x_{10} - x_5, 3763) = \gcd(3059 - 3007, 3763) = 1$

$x_{11} = 2664$

$x_{12} = 3642$ $\quad \gcd(x_{12} - x_6, 3763) = \gcd(3642 - 3324, 3763) = 53.$

Pollard's rho algorithm: examples

Different choices of x_1 give different pseudo-random sequences and can give different results.

3. Find a nontrivial factor in $N = 143$ using the Pollard's rho algorithm.

$x_1 = 2$

$x_2 = 5$ $\quad \gcd(x_2 - x_1, 143) = \gcd(5 - 2, 143) = 1$

$x_3 = 26$

$x_4 = 105$ $\quad \gcd(x_4 - x_2, 143) = \gcd(105 - 5, 143) = 1$

$x_5 = 15$

$x_6 = 83$ $\quad \gcd(x_6 - x_3, 143) = \gcd(83 - 26, 143) = 1$

$x_7 = 26$

$x_8 = 105$ $\quad \gcd(x_8 - x_4, 143) = \gcd(105 - 105, 143) = 143$

$x_9 = 15$

$x_{10} = 83$ $\quad \gcd(x_{10} - x_5, 143) = \gcd(83 - 15, 143) = 1$

$x_{11} = 26$

$x_{12} = 105$ $\quad \gcd(x_{12} - x_6, 143) = \gcd(105 - 83, 143) = 11.$

4. Find a nontrivial factor in $N = 143$ using the Pollard's rho algorithm.

$x_1 = 3$

$x_2 = 10$ $\quad \gcd(x_2 - x_1, 143) = \gcd(10 - 3, 143) = 1$

$x_3 = 101$

$x_4 = 49$ $\quad \gcd(x_4 - x_2, 143) = \gcd(49 - 10, 143) = 13.$