Security Attacks on RSA

A Computational Number Theoretic Approach

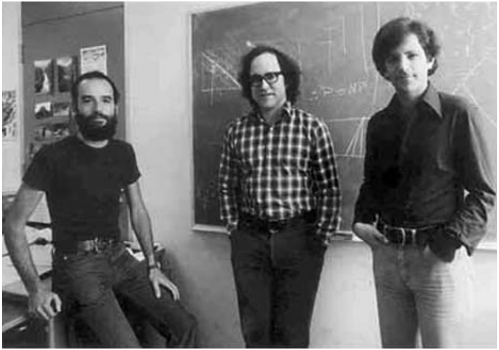
Pratik Poddar

Department of Computer Science and Engineering
IIT Bombay

April 10, 2009

Contents

RSA



Adi Shamir, Ron Rivest, Len Adleman in 1977

The Math behind RSA

- p and q are two distinct large prime numbers

$$N = pq \text{ and } \phi(N) = (p-1)(q-1)$$

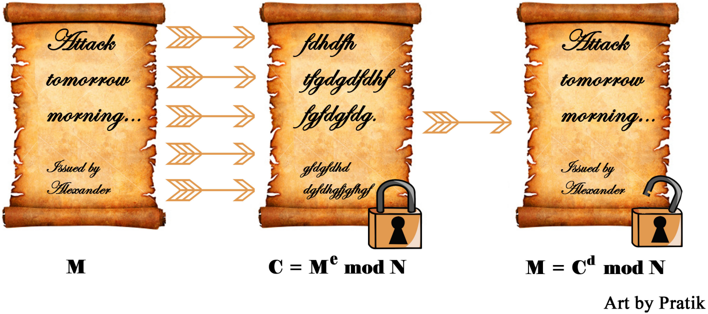
- Choose a large random number $d > 1$ such that $\gcd(d, \phi(N)) = 1$ and compute the number e, $1 < e < \phi(N)$ satisfying the congruence

$$ed \equiv 1 \bmod \phi(N)$$

- The numbers N, e, d are referred to as the *modulus*, *encryption exponent* and *decryption exponent* respectively.
- The public key is the pair (N, e) and the secret trapdoor is d.

Introduction to RSA Cryptosystem

Simplified Model of RSA Cryptosystem

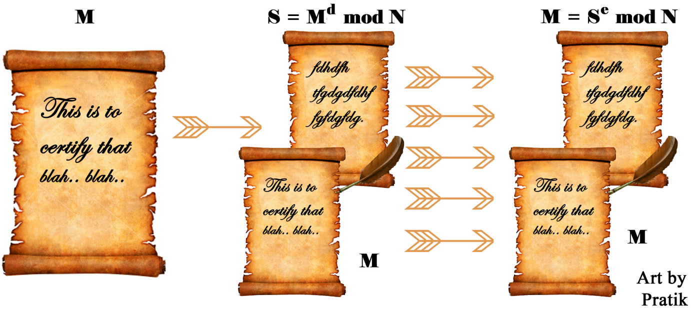


Why it it simplified?

Difference between RSA cryptosystem and RSA function

Introduction to RSA Signatures

Simplified Model of RSA Signatures



Basic Ideas of RSA

Based on the idea that . . . Factorization is difficult

But . . .

There is no formal proof that

1. Factorization is difficult
2. Factorization is needed for cryptanalysis of RSA

Contents

Factoring Algorithm 1: Fermat's Factorization Method

- First improvement over the $\sqrt{n}$ trial-division method
- Every odd composite number can be represented as a difference of two squares

$$n = cd$$

$$n = \left[\frac{c+d}{2}\right]^2 - \left[\frac{c-d}{2}\right]^2$$

- Start from $k = \sqrt{n}$, go on incrementing k till $k^2 - n$ is a perfect square
- Might be worse than trial division ...
- Effect on RSA ... Special case $p - q \leq 4n^{\frac{1}{4}}$

Factoring Algorithm 2: Euler's Factorization Method

- Based upon representing a positive integer n as the sum of two squares in two different ways
$$n = a^2 + b^2 = c^2 + d^2$$
- If n can be represented as a sum of two squares in two different ways, it can be factorized!!
$$(a-c)(a+c) = (d-b)(d+b)$$
Let k be the gcd of $a-c$ and $d-b$, so that $a-c = kl$ and $d-b = km$.
$$l(a+c) = m(d+b) \text{ and } a+c = mr \text{ and } d+b = lr \text{ (Why?)}$$
$$n = \left[\left(\tfrac{k}{2}\right)^2 + \left(\tfrac{r}{2}\right)^2\right] \cdot (m^2 + l^2) \text{ (Are } k \text{ and } r \text{ always even?)}$$
- Start from $k = \sqrt{\frac{n}{2}}$, go on decrementing k till $n - k^2$ is a perfect square
- Historically important!! Flaws!!

Factoring Algorithm 3: Pollard's $p-1$ Method

- Basic idea
- Effect on RSA

Factoring Algorithm 3: Pollard's $p-1$ Method

Pollard's $p-1$ algorithm

Require: A composite integer N
Ensure: A non-trivial factor of N or failure

$B \Leftarrow$ *Chosen Smoothness Bound*
$a \Leftarrow$ *Random number coprime to* N
for $q = 1$ to B **do**
  if q *is prime* **then**
    $e \leftarrow \left\lfloor \frac{\log N}{\log q} \right\rfloor$
    $a \leftarrow a^{q^e} \mod N$
  end if
end for

Factoring Algorithm 3: Pollard's $p-1$ Method

Pollard's $p-1$ algorithm

$g \leftarrow \gcd(a-1, N)$
if $1 < g < N$ **then**
  return g
else if $g = 1$ **then**
  Select a higher B and go to step 2 or return failure
else
  Go to step 2 or return failure
end if

Factoring Algorithm 3: Pollard's $p-1$ Method

- Example... Factorize 5917
- Say $B = 5$, we make $\alpha = 2^{13}3^{8}5^{6}$
- So, $\beta = 2^{\alpha} - 1$
- $\gcd(\beta, 5917) = 61$!!

Factoring Algorithm 4: Pollard's ρ Method

Ideas:

- Birthday Paradox
- Floyd's cycle finding algorithm

Factoring Algorithm 4: Pollard's ρ Method

Pollard's ρ algorithm

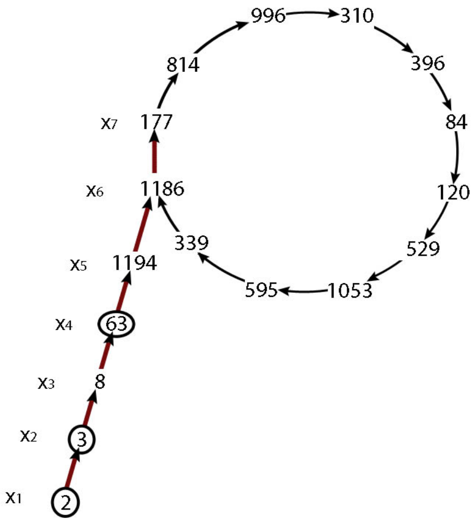
Require: n, the integer to be factored; x_1, such that $0 \le x_1 \le n$; and $f(x)$, a pseudo-random function modulo n.
Ensure: A non-trivial factor of n or failure

$i \Leftarrow 1; y \Leftarrow x_1; k \Leftarrow 2$
loop
  $i \Leftarrow i + 1$
  $x_i \Leftarrow (x_{i-1}^2 - 1) \bmod n$
  $d \Leftarrow \gcd(y - x_i, n)$
  if $d \neq 1$ *and* $d \neq n$ **then**
    return d
  end if
  if $i = k$ **then**
    $y \Leftarrow x_i$
    $k \Leftarrow 2k$
  end if
end loop

Example... Factorize 1387

Factoring Algorithm 4: Pollard's ρ Method .. Example

Example... Factorize 1387



Factoring Algorithm 4: Pollard's ρ Method .. Example

Example... Factorize 1387 Take $x_1 = 2$ and $f(x) = x^2 - 1 \bmod 1387$

i	x_i	$\gcd(x_i - y, 1387)$	y
1	2	-	2
2	3	$\gcd(3-2, 1387) = 1$	3
3	8	$\gcd(8-3, 1387) = 1$	
4	63	$\gcd(63-3, 1387) = 1$	63
5	1194	$\gcd(1194-63, 1387) = 1$	
6	1186	$\gcd(1186-83, 1387) = 1$	
7	177	$\gcd(177-63, 1387) = \mathbf{19}$	
8	814	$\gcd(814-63, 1387)$	814
9	996	$\gcd(996-814, 1387)$	

Complexity??

Breaking RSA v/s Factoring

- Breaking RSA $\leq$ Factoring (Obvious!)
- Open Problem: Factoring $\leq$ Breaking RSA??
- Expectation: No!!! Factoring is expected to be strictly $>$ Breaking RSA

Exposing d v/s Factoring

Theorem

Exposing the private key d and factoring N are equivalent

Proof

- Determining $d \leq$ Factoring N (Why??)
- Determining $d \geq$ Factoring N (Non-obvious algorithm)

We will now present a randomized algorithm by which knowing d, factors of N can be *easily* determined.

Exposing d v/s Factoring . . . Miller Rabin Test

Miller Rabin Test

- Randomized primality testing algorithm
- Miller version $\cdots$ Rabin version

Exposing d v/s Factoring . . . Miller Rabin Test

Miller Rabin Test

Require: $n > 2$, an odd integer to be tested for primality
Ensure: Composite if n is composite, otherwise probably Prime

Write $n - 1$ as $2^s d$ with d odd
$a \Leftarrow$ *Random number between* 1 *and* $n - 1$
$x_0 \Leftarrow a^d \bmod n$
if $x = 1$ *OR* $x = n - 1$ **then**
  return *Probably Prime*
end if

Exposing d v/s Factoring . . . Miller Rabin Test

Miller Rabin Test

for $i = 1$ to $s - 1$ **do**
    $x_i \Leftarrow x_{i-1}^2 \bmod n$
    if $x_i = 1$ *and* $x_{i-1} \neq 1$ *and* $x_{i-1} \neq n - 1$ **then**
        return *Composite*
    end if
end for
if $x_t \neq 1$ **then**
    return *Composite*
else
    return *Probably Prime*
end if

Exposing d v/s Factoring ... Miller Rabin Test

Miller Rabin Error Rate Analysis

If n is a composite number, then the number of witnesses to the compositeness of n is at least $\frac{n-1}{2}$.

Proof

- Prove that number of non-witnesses is at most $\frac{n-1}{2}$
- Creating a subgroup B, which is a subgroup of $\mathbb{Z}_n^*$, which contains all the non-witnesses
- Show the existence of an element in $\mathbb{Z}_n^* - B$,
- Order of $B \leq \frac{n-1}{2}$. Number of non-witnesses $\leq \frac{n-1}{2}$

We break the proof into two cases.

Exposing d v/s Factoring . . . Miller Rabin Test

Case 1: There exists an $x \in \mathbb{Z}_n^*$ such that $x^{n-1} \not\equiv 1 \mod n$

- Let $B = \{b \in \mathbb{Z}_n^* : b^{n-1} \equiv 1 \pmod{n}\}$
- Since there exists an element x for which $x^{n-1} \not\equiv 1 \mod n$, $\mathbb{Z}_n^* - B$ is non-empty
- Number of non-witnesses $\leq \frac{n-1}{2}$

Exposing d v/s Factoring ... Miller Rabin Test

Case 2: For all $x \in \mathbb{Z}_n^*$, $x^{n-1} \equiv 1 \mod n$

Represent n as $n_1 n_2$ where n_1 and n_2 are relatively prime
Note that $n-1 = 2^s u$ and for *input* a, we can compute the following sequence modulo n: $a^u, a^{2u}, a^{2^2 u}, a^{2^3 u}, a^{2^4 u} \cdots a^{2^s u}$
Let us call a pair of integers (v, j) *acceptable* if $v \in \mathbb{Z}_n^*$, $j = 0, 1, 2, \cdots, s$ and

$$v^{2^j u} \equiv -1 \pmod n$$

Set of acceptable pairs contains $(n-1, 0)$. So, the set is non-empty.
Pick the largest possible j for which there exists an v such that (v, j) is an acceptable pair. We will use this j in the proof.

$$B = \{x \in \mathbb{Z}_n^* : x^{2^j u} \equiv \pm 1 \pmod n)\}$$

Clearly, B is a subgroup of $\mathbb{Z}_n^*$. Also note that the sequence produced by a non-witness must be either all 1's or contain -1 no later than the jth position (due to maximality of j). So, every non-witness belongs to B.

Exposing d v/s Factoring . . . Miller Rabin Test

Case 2: For all $x \in \mathbb{Z}_n^*$, $x^{n-1} \equiv 1 \mod n$

To complete the proof, we have to prove that $\mathbb{Z}_n^* - B$ is non-empty. Note that there exists a w such that

$$w \equiv v \pmod{n_1} \textit{ and } w \equiv 1 \pmod{n_2}$$

where v is an element in B such that $v^{2^j u} \equiv -1 \pmod{n}$. So,

$$w^{2^j u} \equiv -1 \pmod{n_1} \textit{ and } w^{2^j u} \equiv 1 \pmod{n_2}$$

This implies $w^{2^j u} \not\equiv -1 \pmod{n}$ and $w^{2^j u} \not\equiv 1 \pmod{n}$. So, $w \notin B$.
All we need to prove is that $w \in \mathbb{Z}_n^*$. Since $v \in \mathbb{Z}_n^*$, $\gcd(v, n) = 1$, which implies $\gcd(v, n_1) = 1$. Since $\gcd(w, n_1) = \gcd(v, n_1)$, $\gcd(w, n_1) = 1$. By construction of w, $\gcd(w, n_2) = 1$. So, $\gcd(w, n_1 n_2) = \gcd(w, n) = 1$.

Exposing d v/s Factoring ... The Randomized Algorithm

The Algorithm... knowing d, factors of N can be easily determined

- Let $k = ed - 1$
- If g is chosen at random from $\mathbb{Z}_n^*$, the with probability at least $\frac{1}{2}$, one of the elements in the sequence $g^{k/2}, g^{k/4}, g^{k/8}, ..., g^{k/2^t}$ mod N is a witness for the compositeness of N
- A witness of compositeness of Miller Rabin test reveals a factor of N as square roots of 1 mod N (other than 1 and -1) would be x and $-x$ where $x \equiv 1$ mod p and $x \equiv -1$ mod q
- $\gcd(x - 1, N)$ would get the factor of N

Guessing $\phi(N)$ v/s Factoring

Guessing $\phi(N)$ and factoring N are equivalent

- Guessing $\phi(N) \geq$ Factoring (Obvious!)
- Factoring $\geq$ Guessing $\phi(N)$ (Why??)

Strong Primes

What are *strong* primes?

A prime p is considered to be a "strong prime" if it satifies the following conditions:

- p is a large prime (say $|p| \geq 256$)
- The largest prime factor of $p-1$, say p^{-}, is large (say $|p^{-}| \geq 100$)
- The largest prime factor of $p^{-}-1$, say p^{--}, is large (say $|p^{--}| \geq 100$)
- The largest prime factor of $p+1$, say p^{+}, is large (say $|p^{+}| \geq 100$)

A prime is

- p^{-}-strong if p^{-} is large
- p^{--}-strong if p^{--} is large
- p^{+}-strong if p^{+} is large
- (p^{-}, p^{+})-strong if both p^{-} and p^{+} are large
- strong if all p^{-}, p^{--} and p^{+} are large

Are Strong Primes Needed?

- Believed that p and q in RSA have to be strong
- Original RSA paper, Cycling Attack, X.509 Standard
- Rivest and Silverman proved that use of strong primes is unnecessary
- PKCS#1 v2.1 does not recommend strong primes

Conclusion

- We discussed about the factoring algorithms present at the time of RSA publication
- How those factorization algorithms affected RSA paper
- We discussed various other ways to attempt RSA breaking and compared them to factoring
- The myth of RSA needing strong primes

Contents

Elementary Attack 1: Dictionary Attack

- One-to-one mapping between ciphertext and plaintext : vulnerable to Dictionary Attack
- Security measure: Random Padding (PKCS#1 v1.5)

Elementary Attack 2: Common Modulus Attack

- RSA modulus should not be used by more than one entity
- *Alice* recieves the ciphertext $C = M^{e_a} \bmod N$
- *Mallory* does not have d_a, but using e_b and d_b, *Mallory* can factor N
- d_a can be calculated and *Mallory* can decrypt the message intended for *Alice*

Elementary Attack 3: Blinding Attack

- Attack specific to RSA signatures
- Suppose attacker A wants to get a document M signed by B
- A needs M^d mod N $\cdots$ A sends $r^e M$ mod N for B to sign
- Signing the sent message gives

$$r^{ed} M^d \bmod N = rM^d \bmod N$$

- Security measure: Random Padding, Signing Hash

Contents

Low Private Exponent Attack

Wiener's Attack

Let $N = pq$ with p and q approximately of the same size, i.e. $q < p < 2q$. Let $d < \frac{1}{3}N^{1/4}$. Given (N, e) with $ed \equiv 1 \bmod \phi(N)$, the attacker can *easily* recover d.

Continued Fractions

Definition

The function of $n+1$ variables

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots \cfrac{1}{a_n}}}}$$

or

$$[a_0, a_1, a_2, a_3 \cdots, a_n]$$

is defined as a continued fraction. We consider only simple and positive continued fractions, i.e. all $a_i's$ are integral and positive. $[a_0, a_1, a_2, a_3 \cdots, a_i]$, $0 \leq i \leq n$ are said to be the *convergents* of $[a_0, a_1, a_2, a_3 \cdots, a_n]$.

Continued Fractions $\cdots$ Theorems

Theorem 1

The continued fraction

$$[a_0, a_1, a_2, \cdots, a_n] = [a_0, a_1, \cdots, a_{m-1}, [a_m, \cdots a_n]]$$

Theorem 2

The continued fraction

$$[a_0, a_1, a_2, \cdots, a_{m-1}, a_m] = \frac{p_m}{q_m} = \frac{a_m p_{m-1} + p_{m-2}}{a_m q_{m-1} + q_{m-2}}$$

Continued Fractions $\cdots$ Theorems

Theorem 3

Continued Fraction Algorithm: Any rational number x can be represented as a simple finite continued fraction.

Proof

$$x = \frac{h_0}{k_0}$$

Comparing x with $[a_0, a_1, a_2, \cdots, a_N, a_{N+1}]$,

$$x = a_0 + \xi_0 \textit{ where } \xi_0 < 1 \textit{ i.e. } h_0 = a_0 k_0 + \xi_0 k_0$$

If $\xi_0 \neq 0$

$$\frac{1}{\xi_0} = \frac{k_0}{h_0 - a_0 k_0}$$

Let $k_1 = h_0 - a_0 k_0$, (since, $k_1 = \xi_0 k_0$, we have $k_1 < k_0$)

$$\frac{k_0}{k_1} = a_1 + \xi_1 \textit{ and } k_0 = a_1 k_1 + \xi_1 k_1$$

Continued Fractions $\cdots$ Theorems

Proof Contd $\cdots$

Doing this repeatedly gives a system of equations

$$h_0 = a_0 k_0 + k_1 \ , \ \ k_0 = a_1 k_1 + k_2$$

$$h_1 = a_1 k_1 + k_2 \ , \ \ k_1 = a_2 k_2 + k_3$$

$$\cdots$$

as long as $k_{N+1} \neq 0$.

Equations same as when executing Euclid's extended algorithm to compute gcd of h_0 and k_0.

So, Continued fraction algorithm terminates and the number of convergents for x is $\Theta(\min(\log h_0, \log k_0))$.

Continued Fractions · · · Theorems

Theorem 4

If

$$\left|\frac{p}{q} - x\right| \leq \frac{1}{2q^2}$$

then $\dfrac{p}{q}$ is a convergent of continued fraction expansion of x.

Proof

Assuming that the statement is true, then

$$\frac{p}{q} - x = \frac{\epsilon\theta}{q^2}$$

where $\epsilon = \pm 1$ and $0 < \theta < \frac{1}{2}$. Let $\frac{p_n}{q_n}$ and $\frac{p_{n-1}}{q_{n-1}}$ be the last and second last convergents of continued fraction of $\frac{p}{q}$. Note that $\frac{p_n}{q_n} = \frac{p}{q}$. We can write, for some ω,

$$x = \frac{\omega p_n + p_{n-1}}{\omega q_n + q_{n-1}}$$

Continued Fractions $\cdots$ Theorems

Proof Contd $\cdots$

$$\theta = \frac{q_n}{\omega q_n + q_{n-1}} \text{ and } \omega = \frac{1}{\theta} - \frac{q_{n-1}}{q_n}$$

Note that since $\theta < \frac{1}{2}$, we have, $\omega > 1$

Let $\frac{p}{q} = [a_0, a_1, \cdots a_{m1}]$

Let $\omega = [b_0, b_1, b_2 \cdots b_{m2}]$

$$\begin{aligned}[a_0, a_1, a_2 \cdots a_{m1}, \omega] &= [a_0, a_1, a_2 \cdots a_{m1}, [b_0, b_1, b_2 \cdots b_{m2}]] \\ &= \frac{\omega p_m + p_{m-1}}{\omega q_m + q_{m-1}} = x\end{aligned}$$

So, $x = [a_0, a_1, a_2 \cdots a_{m1}, b_0, b_1, \cdots b_{m2}]$.

So, by construction, we have proved that $\frac{p}{q}$ is a convergent of continued fraction expansion of x.

Wiener's Attack

Attack by Wiener, 1990

Let $N = pq$ with p and q approximately of the same size, i.e. $q < p < 2q$. Let $d < \frac{1}{3}N^{1/4}$. Given (N, e) with $ed \equiv 1 \bmod \phi(N)$, the attacker can easily recover d.

Proof

There exists k such that $ed - k\phi(N) = 1$. We will first show that that $\frac{k}{d}$ is an approximation of $\frac{e}{N}$. Also note that $N - \phi(N) < 3\sqrt{N}$.

$$\left|\frac{e}{N} - \frac{k}{d}\right| = \left|\frac{1 - k(N - \phi(N))}{Nd}\right|$$

$$\left|\frac{e}{N} - \frac{k}{d}\right| \leq \frac{3k}{d\sqrt{N}}$$

Also, since $k\phi(N) = ed - 1 < ed$ and $e < \phi(N)$, we have $k < d$. In the case when $d < \frac{1}{3}N^{1/4}$, we obtain $k < d < \frac{1}{3}N^{1/4}$ and so

$$\left|\frac{e}{N} - \frac{k}{d}\right| \leq \frac{1}{d\sqrt[4]{N}} \leq \frac{1}{3d^2} < \frac{1}{2d^2}$$

Wiener's Attack

Proof Contd$\cdots$

- $\frac{k}{d}$ is a convergent of continued fraction expansion of $\frac{e}{N}$
- Number of fractions to be checked for $\frac{k}{d}$ is bounded by $\Theta(\log N)$
- One of the $\Theta(\log N)$ convergents of continued fraction for $\frac{e}{N}$ is $\frac{k}{d}$
- $\frac{k}{d}$ is a reduced fraction
- We have $\Theta(\log N)$ available options for d

Avoiding Wiener's Attack

Method 1: Large e

- Instead of using $e < N$, use $e' = e + t\phi(N)$ for a large t
- This would mean $e' > N$
- Large e' would mean a large k which would counter Wiener's attack
- If $e' > N^{1.5}$, Wiener's attack is not possible even for very small d

Avoiding Wiener's Attack

Method 2: Using CRT

- Use CRT to reduce the decryption time (and signing time) even while using large d
- Choose d such that $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$ are small
- For decryption, compute $M_p = C^{d_p} \bmod p$ and $M_q = C^{d_q} \bmod q$
- Then using CRT, compute M satisfying
  $M \in \mathbb{Z}_n$
  $M = M_p \bmod p$ and
  $M = M_q \bmod q$
  Note that here $M = C^d \bmod N$

References

- D. Boneh. *"Twenty Years of Attacks on the RSA Cryptosystem."* Notices of the American Mathematical Society, 46(2):203–213, 1999.
- M. Wiener. *"Cryptanalysis of short RSA secret exponents."* IEEE Transactions on Information Theory, 36:553 558, 1990
- Ron Rivest and Robert Silverman. *"Are 'Strong' Primes Needed for RSA?"*, Cryptology ePrint Archive: Report 2001/007
- Rivest, R.; A. Shamir; L. Adleman. *"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems"*. Communications of the ACM 21 (2): pp.120-126.