RSA assumption:

Given $N=pq$ and $e \in [1, \ldots \phi(N)]$.
it is infeasible to find x from $x^e \bmod N$ in polynomial time.

Factoring Assumption:

given $N = p \cdot q$ for some large primes p, q.
It is infeasible to find out the factors of N in polynomial time.

- GenRSA.
- GenModulli.

1. If RSA is hard with respect to GenRSA then factoring is hard with respect to GenModulli.

RSA-Hard $\Rightarrow$ Factoring is hard.

Can we say the other direction is true?

2. Factoring is hard with respect to GenModulli $\Rightarrow$ RSA is hard with respect to GenRSA. (open).

Thm: There is a PPT alg. that given as i/p N, e, d, such that $ed \equiv 1 \bmod \phi(N)$, outputs a factor of N except with negligible probability in n, where $n = \|N\|$.

Pf: We assume that N is the product of two distinct odd primes.

Fact 1. There are exactly four square roots of 1 modulo N.

$$x^2 \equiv 1 \bmod N \Rightarrow \left.\begin{array}{l} x^2 \equiv 1 \bmod p \\ x^2 \equiv 1 \bmod q \end{array}\right\} \in RT. \left.\begin{array}{l} (1_p, -1_p) \\ (1_q, -1_q) \end{array}\right.$$

Two trivial sq. roots: $(1,1), (-1,-1)$
Two non-trivial " " $(-1, 1), (1, -1)$

2. Any non-trivial square root of 1 modulo N can be used to factor N.

Let y be one such non-trivial square root.

$$y^2 \equiv 1 \bmod N, \text{ where } y \in \mathbb{Z}_N.$$

$$\Rightarrow N \mid (y-1)(y+1).$$

But $N \nmid (y-1)$ and $N \nmid (y+1)$ because $y \neq \pm 1 \bmod N$

Therefore $\gcd(N, y-1)$ must be one of the factors of N.

Therefore, if we can find out one non-trivial square root of 1 mod N, then, one can find out the factors of N.

Consider $x \in \mathbb{Z}_N^*$

By Euler's theorem, $x^{\phi(N)} \equiv 1 \bmod N$

We know that $ed \equiv 1 \bmod \phi(N)$.

$$\Rightarrow \phi(N) \mid \underbrace{(ed-1)}_{K}$$

$$\Rightarrow k = \phi(N) \cdot t$$

Therefore, $x^k \equiv 1 \bmod N$

Let $k = 2^r \cdot u$, where u is odd and $r \geq 1$

We consider the seq. $x^u, x^{2u}, \ldots, x^{2^r u} \bmod N$.

We take the largest i for which

$$y = x^{2^i u} \bmod N \not\equiv 1 \bmod N.$$

Therefore, $y^2 \equiv 1 \bmod N$ (By the choice of i)

If $y \neq -1 \bmod N \Rightarrow y$ is a non-trivial square root of 1 modulo N and thus we can factorize N.

Analysis: Let $i \in \{0, \ldots, r-1\}$ be the largest value for which there exists an $x \in \mathbb{Z}_N^*$ such that

$$x^{2^i u} \not\equiv 1 \bmod N.$$

Fix some i, and x for which

$x^{2^i u} \not\equiv 1 \bmod N$. $\Rightarrow$ $x^{2^{i+1} u} \equiv 1 \bmod N$

Therefore, $x^{2^i u}$ is a square root of 1 modulo N.

$$Bad = \left\{ x \in \mathbb{Z}_N^* : x^{2^i u} \equiv \pm 1 \bmod N \right\}.$$

- Bad is a strict subgroup of $\mathbb{Z}_N^*$
- $|Bad| < |\mathbb{Z}_N^*| / 2$

If the algorithm chooses an $x \notin Bad \Rightarrow x$ is a non-trivial square root of 1 modulo N. (QED).

Conclusion:

Assuming that factoring is hard, the above result rules out the possibility of solving RSA by first computing d from N and e.

However, it does not rule out the possibility that there might be completely different way of solving RSA without factoring N.

RSA - algorithm:

- $KeyGEN(1^n)$:
  - Randomly samples two n bit distinct primes. p, q and then compute $N = pq$.
  - Compute $\phi(N) = (p-1)(q-1)$.
  - choose an e, $1 \leq e \leq \phi(N)$.
  - compute d such that $ed \equiv 1 \bmod (p-1)(q-1)$.

return $(pk = (N, e), sk = (N, d))$.

RSA-algorithm: (Textbook RSA)

- $KeyGen(1^n)$:
  - Randomly samples two n bit distinct primes. p, q and then compute $N = pq$.
  - Compute $\phi(N) = (p-1)(q-1)$.
  - choose an e, $1 < e < \phi(N)$.
  - compute d such that $ed \equiv 1 \bmod (p-1)(q-1)$.

  return $(pk = (N, e), sk = (N, d))$.

- Enc(pk, m): for input $m \in \mathbb{Z}_N$ and $pk = (N, e)$ compute $c = m^e \bmod N$.
- Dec(sk, c): for input $c \in \mathbb{Z}_N$ and $sk = (N, d)$ compute $m = c^d \bmod N$.

Correctness: $\forall n \in \mathbb{N}$, $\forall (pk, sk) \leftarrow KeyGEN(1^n)$ and $\forall m \in \mathbb{Z}_N$.

$$Dec(sk, Enc(pk, m)) = m.$$

Verify the correctness by applying CRT.

- $\underline{Enc(pk, m)}$: for input $m \in \mathbb{Z}_N$ and $pk = (N, e)$ compute $c = m^e \bmod N$.
- $Dec(sk, c)$: for input $c \in \mathbb{Z}_N$ and $sk = (N, d)$. compute $m = c^d \bmod N$.

$\underline{\text{Correctness}}$: $\forall n \in \mathbb{N}$, $\forall (pk, sk) \leftarrow KeyGEN(1^n)$ and $\forall m \in \mathbb{Z}_N$.

$$Dec(sk, Enc(pk, m)) = m.$$

- Verify the correctness by applying CRT.
- Decryption can be made faster using CRT.

Textbook RSA is not secure!

(i) Natural choice of $e = 3$.

If you encrypt a small message $m < N^{1/3}$.

with small exponent e.

Then $c = m^3 < N$

Therefore. When an eavesdropper gets c, it just computes

$c^{1/3}$ to recover m.

(ii) General attack using small exponent. ($e=3$)

Let there are three parties:

- Alice: $(N_1, 3)$ $N_1 = p_1 q_1$
- Bob – $(N_2, 3)$ $N_2 = p_2 q_2$
- Carol – $(N_3, 3)$ $N_3 = p_3 q_3$.

We are assuming that they are encrypting same message m

$C_1 = m^3 \bmod N_1$
$C_2 = m^3 \bmod N_2$
$C_3 = m^3 \bmod N_3$.

N_1, N_2, N_3 are mutually co-prime to each other. Otherwise one can find out the prime factors.

By the virtue of Chinese Remainder Theorem, there exists a unique soln

$$C = m^3 \bmod (N_1 N_2 N_3)$$

But $m < \min\{N_1, N_2, N_3\}$.

$$m^3 < N_1 N_2 N_3.$$

Then. by observing c, one can recover m by. just computing $c^{1/3}$.

(iii) Common modulus attack

$pk_i = (N, e_i)$

$sk_i = (N, d_i)$.

$e_i d_i \equiv 1 \bmod \phi(N)$ for $i = 1, \ldots, n$

(iv) Common-modulus attack-II

$c_1 = m^{e_1} \bmod N$

$c_2 = m^{e_2} \bmod N$

Assume that $\gcd(e_1, e_2) = 1$

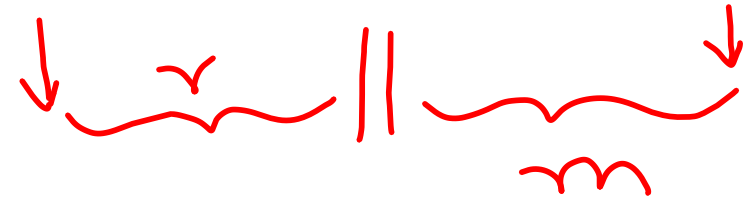
$e_1\alpha + e_2\beta = 1$

$c_1^{\alpha} \cdot c_2^{\beta} \bmod N = m^{e_1\alpha + e_2\beta} \bmod N$

$= m \bmod N$.

Conclusion:

Textbook RSA is not secure against eavesdropper.



Padded-RSA.

Let N be the modulli and $n = \|N\|$.

Let $m \in \mathbb{Z}_N$ such that $m \in \{0,1\}^{\ell(n)}$ and $\ell(n) < n$.

Enc: Randomly sample $r \leftarrow\$ \{0,1\}^{n-\ell(n)}$ and then:

$$m' = r \| m$$

Compute $c' = (r \| m)^e \bmod N$.

Dec: Compute $c^d \bmod N$ and output the least significant $\ell(n)$ bits.

- If has been proved that if $\ell(n) = 1$, then under the RSA assumption Padded RSA is secure.

- If $\ell(n) = c \cdot n$, then it is believed to be secure
- If $\ell(n)$ is large enough, then Padded RSA is not secure