بسم الله الرحمن الرحيم

Shor’s Algorithm for Cryptanalysis

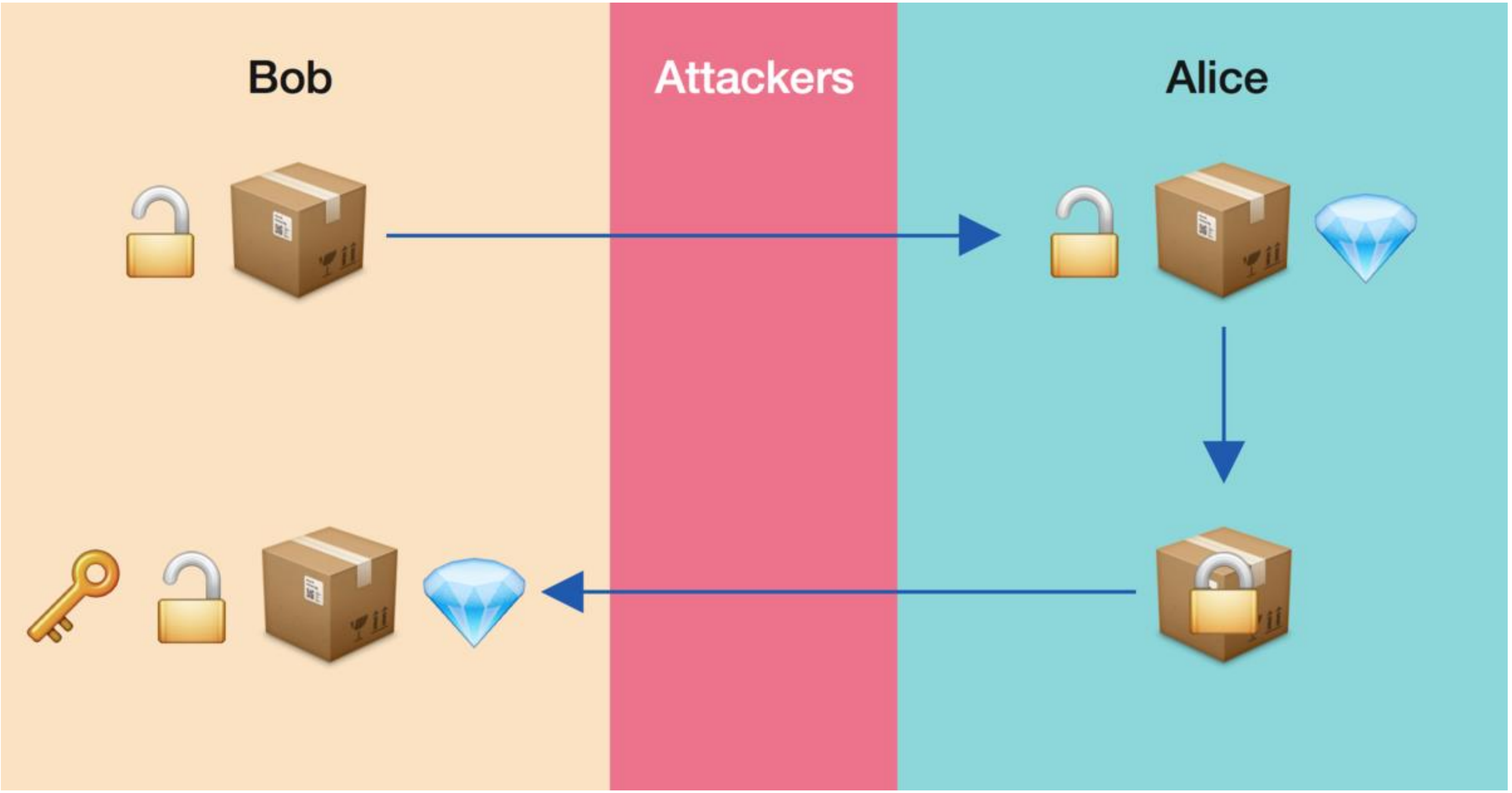
Mohammad Sabokdast

October 2024

مرکز تحقیقات
فناوری‌های
کوانتومی ایران

RSA Cryptography

RSA Cryptography



RSA Applications

Secure Web Browsing (SSL/TLS): RSA is commonly used in securing HTTPS connections. When you visit a website with "https://", RSA may be part of the process that encrypts the communication between your browser and the website.

Email Encryption: RSA can be used to encrypt emails, ensuring that only the intended recipient can read the contents. Technologies like PGP (Pretty Good Privacy) use RSA for this purpose.

Digital Signatures: RSA is used in creating digital signatures that verify the authenticity and integrity of a message, software, or document. Digital signatures help confirm that a message has not been altered and was sent by the claimed sender.

Secure Software Distribution: RSA can be used to verify that software being installed comes from a legitimate source, protecting against malicious software.

VPNs and Secure Communication Protocols: Virtual Private Networks (VPNs) and other secure communication channels often use RSA as part of their encryption process to ensure secure data transmission.

Cryptographic Tokens and Smart Cards: RSA is used in various hardware security tokens and smart cards, ensuring authentication and secure access to systems.

Blockchain and Cryptocurrencies: RSA is occasionally used for secure communication or signatures in some blockchain-related technologies.

RSA Cryptography

$$n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$$

$$\varphi(n) = \prod_{j=1}^{k} p_j^{\alpha_j - 1}(p_j - 1).$$

Theorem : Suppose a is co-prime to n. Then $a^{\varphi(n)} = 1(\text{mod } n)$.

RSA Cryptography

(1) Select two large prime numbers, p and q.

(2) Compute the product $n \equiv pq$.

(3) Select at random a small odd integer, e, that is relatively prime to $\varphi(n) = (p-1)(q-1)$.

(4) Compute d, the multiplicative inverse of e, modulo $\varphi(n)$.

(5) The *RSA public key* is the pair $P = (e, n)$. The *RSA secret key* is the pair $S = (d, n)$.

RSA Cryptography

$$E(M) = M^e (\text{mod } n).$$

$$E(M) \rightarrow D(E(M)) = E(M)^d (\text{mod } n).$$

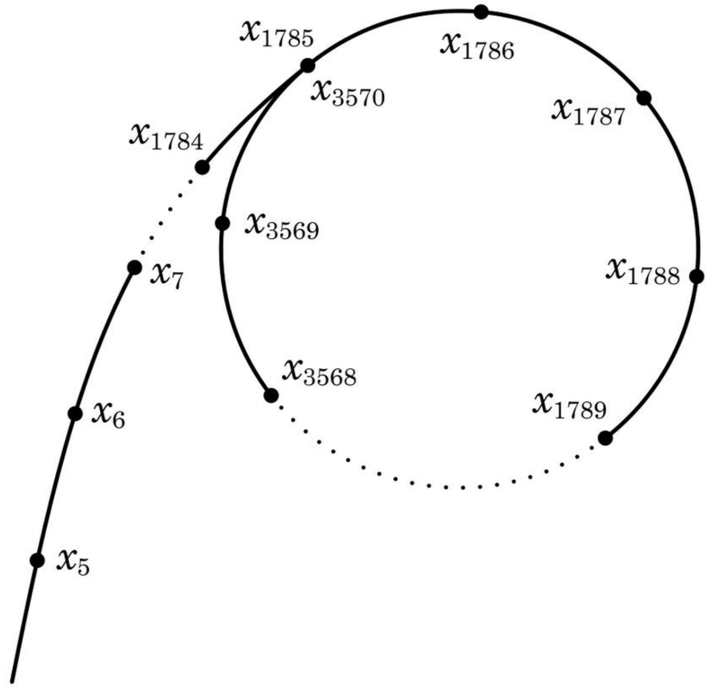
$$\begin{aligned} D(E(M)) &= E(M)^d (\text{mod } n) \\ &= M^{ed} (\text{mod } n) \\ &= M^{1+k\varphi(n)} (\text{mod } n) \\ &= M \cdot M^{k\varphi(n)} (\text{mod } n) \\ &= M (\text{mod } n), \end{aligned}$$

RSA Example

Alice wants to send message $M = 104$ to Bob.
Bob chooses two prime numbers, p and q
For example $p = 17$ and $q = 41$.
Bob calculates $n = pq = 697$.
Bob computes $\phi(n) = (p-1)(q-1)$
in our example: $\phi(697) = (17-1)(41-1) = 640$
Bob chooses two number e and d such that ed = 1 (mod 640)
For example e = 3 and d = 427 work. (3 * 427 = 1281).
Bob *publishes* n and e.
Alice calculate $C = M^e \ (mod\ n) = 104^3 \ (mod\ 697) = 603$
Alice sends C to Bob.
He computes $C^d \ (mod\ n) = 603^{427} \ (mod\ 697) = (104^3)^{427} \ (mod\ 697)$
$= 104^{1281} \ (mod\ 697) = 104^1$
With $104^{640} = 1\ (mod\ 697)$ because $M^{\phi(n)} = 1\ (mod\ n)$

RSA Cryptography

```
POLLARD-RHO(n)
i = 1
x_1 = RANDOM(0, n − 1)
y = x_1
k = 2
while TRUE
    i = i + 1
    x_i = (x_{i−1}^2 − 1) mod n
    d = gcd(y − x_i, n)
    if d ≠ 1 and d ≠ n
        print d
    if i == k
        y = x_i
        k = 2k
```



Order Finding And Factoring

Order Finding And Factoring

Theorem : Suppose N is a composite number L bits long, and x is a non-trivial solution to the equation $x^2 = 1(\text{mod } N)$ in the range $1 \leq x \leq N$, that is, neither $x = 1(\text{mod } N)$ nor $x = N - 1 = -1(\text{mod } N)$. Then at least one of $\gcd(x-1, N)$ and $\gcd(x+1, N)$ is a non-trivial factor of N that can be computed using $O(L^3)$ operations.

For example:
$N = 35, x = 6$
$x^2 = 36 = 1 \ (mod \ 35)$
$x - 1 = 5, x + 1 = 7$

Order Finding And Factoring

Suppose N is a positive integer, and x is co-prime to N, $1 \leq x < N$. The *order* of x modulo N is defined to be the least positive integer r such that $x^r = 1(\mathrm{mod}\ N)$. The *order-finding problem* is to determine r, given x and N.

Theorem Suppose $N = p_1^{\alpha_1} \cdots p_m^{\alpha_m}$ is the prime factorization of an odd composite positive integer. Let x be chosen uniformly at random from $\mathbf{Z}_N^*$, and let r be the order of x, modulo N. Then

$$p(r \text{ is even and } x^{r/2} \neq -1(\mathrm{mod}\ N)) \geq 1 - \frac{1}{2^m}.$$

Order Finding And Factoring

(1) If N is even, return the factor 2.
(2) determine whether $N = a^b$ for integers $a \geq 1$ and $b \geq 2$, and if so return the factor a.
(3) Randomly choose x in the range 1 to $N - 1$. If $\gcd(x, N) > 1$ then return the factor $\gcd(x, N)$.
(4) Use the order-finding subroutine to find the order r of x, modulo N.
(5) If r is even and $x^{r/2} \neq -1 (\text{mod } N)$ then compute $\gcd(x^{r/2} - 1, N)$ and $\gcd(x^{r/2} + 1, N)$, and test to see which is a non-trivial factor, returning that factor. Otherwise, the algorithm fails.

Shor’s Quantum Algorithm For Order Finding

Shor's Quantum Algorithm For Order Finding

The quantum algorithm for order-finding is just the phase estimation algorithm applied to the unitary operator

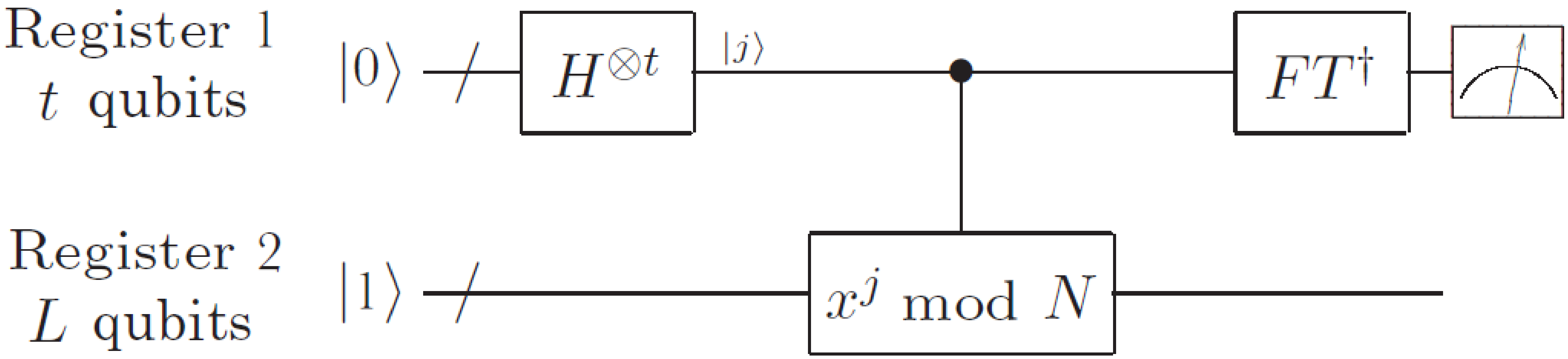
$$U|y\rangle \equiv |xy(\text{mod } N)\rangle \, ,$$

$$|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] |x^k \text{ mod } N\rangle \, ,$$

$$U|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] |x^{k+1} \text{ mod } N\rangle$$

$$= \exp\left[\frac{2\pi i s}{r}\right] |u_s\rangle \, .$$

Shor's Quantum Algorithm For Order Finding



Shor’s Quantum Algorithm For Order Finding

Algorithm: Quantum order-finding

Inputs: (1) A black box $U_{x,N}$ which performs the transformation $|j\rangle|k\rangle \to |j\rangle|x^j k \bmod N\rangle$, for x co-prime to the L-bit number N, (2) $t = 2L + 1 + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$ qubits initialized to $|0\rangle$, and (3) L qubits initialized to the state $|1\rangle$.

Outputs: The least integer $r > 0$ such that $x^r = 1 \pmod N$.

Runtime: $O(L^3)$ operations. Succeeds with probability $O(1)$.

Shor's Quantum Algorithm For Order Finding

1.	$\|0\rangle\|1\rangle$	initial state
2.	$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle\|1\rangle$	create superposition
3.	$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle\|x^j \bmod N\rangle$	apply $U_{x,N}$
	$\approx \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi i s j/r} \|j\rangle\|u_s\rangle$	
4.	$\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \|\widetilde{s/r}\rangle\|u_s\rangle$	apply inverse Fourier transform to first register
5.	$\rightarrow \widetilde{s/r}$	measure first register
6.	$\rightarrow r$	apply continued fractions algorithm

مرکز تحقیق
فناوری‌های
کوانتومی ایران

Thank You

For Your Attention

مرکز تحقیقات
فناوری‌های
کوانتومی ایران