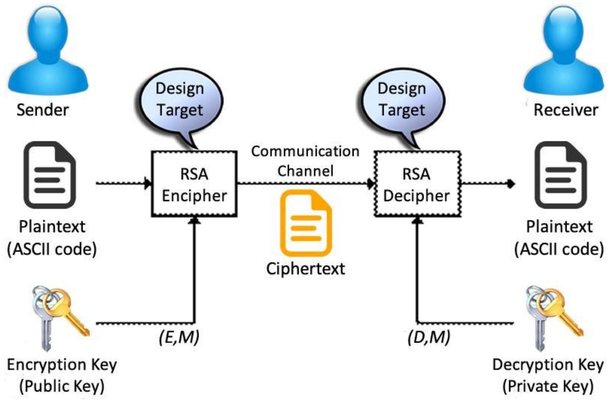
Breaking RSA Encryption

Grant Haataja

University of North Dakota

Abstract

Developed in 1977 by three MIT researchers, RSA encryption is a public-key cryptographic scheme that is heavily relied on for many of our modern security needs. Although it is an extremely secure method of encrypting data, nothing is perfect and there are some flaws in the method. It is necessary to understand how RSA encryption can be broken, and how viable the methods of breaking it would be in both the present world and the future.



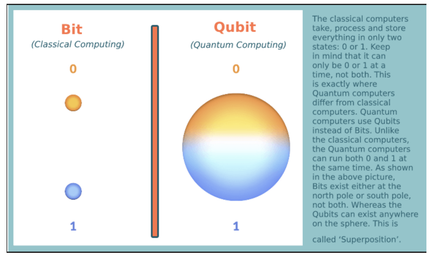
Shor's Algorithm

1. Find the GCD of N and m, where N, is the number you are trying to factor, and m is a random positive integer less than N. Most likely, $\gcd(N, m) = 1$, and the algorithm continues. If the gcd does not equal one, then you have found a factor of N and the work is complete.
2. Find the period of $m \pmod N$, $m^2 \pmod N$, and $m^3 \pmod N$. This is the only step that requires a quantum computer superior to traditional computers.
3. If the period P is even, continue with the algorithm. If it is odd, go back to step 1 and choose another random m.
4. Confirm that $m^{P/2} + 1 \neq 0 \pmod N$. If this is true, continue on to step 5. Otherwise, go back to step 1.
5. Compute $\gcd(m^{P/2} - 1, N)$. The result will be a non-trivial prime factor of N, and will give you the key to break anything encrypted using RSA with the key N.

If quantum technology strong enough to break RSA is developed, it will also allow for a new system of **quantum key generation**, which will offer **resistance** to quantum attacks.

Take a picture to download the full paper

Quantum Superposition



RSA Algorithm Steps:

1. Pick two prime numbers p and q with primality test
2. Compute modulus $n = p \times q$
3. Carmichael's totient function: $\lambda(n) = \text{lcm}(p-1, q-1)$
4. Generate prime number e to use with modulus n for the public key
5. To encrypt plaintext message m, use: $c = m^e \bmod n$
6. Compute the number d to use with modulus n for the private key
7. $d = 1/e \bmod \lambda(n)$
8. d can be used to decrypt messages encrypted with public key given above by using the following formula: $m = c^d \bmod n$

RSA Algorithm Example: