Q’loQ

KryptoMagik

Q’loQ

A Public Key Encryption Algorithm

Based on RSA

- RSA encryption algorithm invented in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman
- RSA is the most popular public key encryption algorithm today

Q'loQ

- Invented in 2019 by Karl Zander
- Reference implementation in Python
- Based on the idea of the Klingon cloaking device

Governing Principles

- Base
- Cloak
- Key generation

Base

- Generate 4 primes of equal prime size, let them be P, Q, A, B and let them not be equal
- Establish a sub-totient S as the product of P – 1, Q – 1 and P
- Establish the totient as the product of A – 1, B – 1, S, A and Q

Cloak

- Establish the cloaking parameter C by taking P modulo Q
- Establish the cloaking parameter K by taking Q modulo P
- Establish the cloaking parameter G by taking (P modulo Q) + Q

Cloak

- Generate the cloaking mask M with the following formula:

$$((K * G) * (C+K)/K) + (((p/q) + (q/p))/(K+C)) = M$$

- Establish the cloaked modulus N with the following formula:

$$((C * K) * (C+K)/C) + (((a/b) + (b/a))/(K+C)) = N$$

Public Key Generation

- Next find a number between 1 and the totient T where the number and T are co-prime and call it PK. This is the public key.

Private Key Generation

- Find the multiplicative inverse of the public key PK and the totient T and call it SK, the secret key.

Encryption/Decryption

- Encryption is achieved by taking the plain text and raising it to the power of the public key modulo N. This is called phase1. Then phase1 is raised to the power of the public modulo M
- Decryption is achieved by taking the cipher text and raising it to the power of the private key modulo M producing phase1 and then taking phase1 to the power of the N

Cryptanalysis

- Factoring the base primes from N or M is not always possible using Fermat’s theorem. Q’loQ was designed to be resistant to the theorem.
- P and Q can be factored using Fermat’s theorem in the modulus. A and B cannot be identified in the modulus or mask, they are cloaked.