QloQ Public Key Algorithm

Karl Zander
karl.c.zander@gmail.com

KryptoMagick R&D

Abstract. QloQ is a proposed public key algorithm capable of encryption and signing. It is a RSA variant and aims to double the security of RSA at the cost of double the encryption and signing time.

1 Introduction

While RSA is slowly being phased out in favor of elliptic curve solutions, RSA is still core to SSL/TLS certificates, DNSSEC, YubiKeys and much more. RSA remains a solid public key algorithm. QloQ, as proposed below, hopes to double the security level of RSA while still coming in twice as slow as the RSA algorithm for all operations. It's design doubles the effort in factoring attacks as the attacker must create a second infrastructure for the second modulus.

1.1 Contribution

I propose a new public key algorithm, QloQ, which consists of 2 moduli, 1 public key, 1 private key. The crux of QloQ is that each of the 2 moduli (N and M), shared the same totient or phi of N. Each prime minus 1 is multiplied together (4 in total) to create a singular totient from which to derive the public key from.

2 Specification

QloQ is composed of two moduli derived from at least 3072 bit primes. Modulus N composed of primes P and Q and M composed of primes A and B. The public key is derived exactly in the same manner as the RSA public key as well as the private key.

2.1 Notations

N – a modulus defined as the product of two primes (P and Q)
M – a modulus defined as the product of two primes (A and B)
T – a totient or phi of N defined as the product of (P – 1, Q – 1, A – 1, B – 1)
PK – randomly generated public key integer defined as having a GCD of 1 with T
SK – defined as the modular inverse of PK in T
P – a plaintext integer
C – a ciphertext integer
S – a signature integer
V – a signature verification integer

2.2 Key Generation

Randomly generate 2 primes of size at least 3072 bits and let them not be equal, call them P and Q. Randomly generated 2 primes of size at least 3072 bits and let them not be equal, call them A and B. Multiply P and Q and the result call N. Multiply A and B and call the result M.

Next, create the totient by multiplying (P – 1 * Q – 1 * A – 1 * B – 1) and call the result T.

Create the public key by generating a random number between 1 and T until the GCD of a random number equals 1.

Create the private key by calculating the modular inverse of PK mod T.

If N is larger than M, swap the N and M values (i.e. N would become M and M would become N). N must be defined as the smaller of the two moduli.

2.3 Encryption and Decryption

Encryption is achieved by modular exponentiation of the plaintext by PK first mod N (called phase1) and then mod M called (phase2 or the ciphertext).

$\text{phase1} = (P^{PK}) \bmod N$
$\text{phase2} = (\text{phase1}^{PK}) \bmod M$

Phase2 is the ciphertext C.

Decryption is achieved by modular exponentiation of the ciphertext by SK first mod M (called phase1) and then mod N called (phase2 or the plaintext).

$\text{phase1} = (C^{SK}) \bmod M$
$\text{phase2} = (\text{phase1}^{SK}) \bmod N$

Phase2 is the plaintext P.

2.4 Signing and Verification

Signing of a message is achieved by using the decryption function on the message. P, a plaintext message (or hash of a message) is modular exponentiated by SK mod M and then the result (phase1) by SK mod N.

$\text{phase1} = (P^{SK}) \bmod M$
$\text{phase2} = (\text{phase1}^{SK}) \bmod N$

Phase2 is the signature S.

Verification of a signature is achieved by using the encryption function on the signature. S, a signature is modular exponentiated by PK mod N and then the result (phase1) by PK mod M resulting in V. V is compared against the plaintext P or hash of the message, if they are equal then verification has succeed, if they are not equal, the signature is not valid.

$\text{phase1} = (S^{PK}) \bmod N$
$\text{phase2} = (\text{phase1}^{PK}) \bmod M$

Phase2 is the signature verification V.

3 Design Principles

Double strength. QloQ was designed with the primary goal of having double the strength of RSA. Having multiple moduli spurs the attacker to have double the infrastructure to factor moduli.

Built on tried and true design of RSA. RSA has been solid in security even with attacks getting ever closer to a practical break. RSA with large key size has even been a contender for post quantum algorithm standarizations.

General compatibility with existing API's. QloQ is minimally different than RSA and allows ease for a near drop-in replacement for RSA.

4 Security Analysis

TBD

5 Conclusion

Proposed in this document is the QloQ public key algorithm. It allows for two or more parties to communicate securely without having met to exchange keys. QloQ is good for key encapsulation and digital signatures. It's security relies on that of RSA.