Falcon post-quantum cryptography

Tom Ballantyne

December 2024

1 Introduction

Falcon (short for Fast Fourier latice-based compact signatures over NTRU) is a cryptographic signature algorithm submitted to NIST on November 30th, 2017. It has been designed by: Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte and Zhenfei Zhang.

2 Mathematical concepts behind Falcon

Falcon is based on the NTRU lattice, which is created by large order polynomials based on two numbers (p,q) that are coprime, where q is much larger than p.

2.1 Private key generation

The private key consists of two high-order polynomials, f(x) and g(x), with coefficients much smaller than q. Additionally, f(x) must be invertible for both modulo p and modulo q, which is critical for the decryption process (these inverse polynomials can be found efficiently using Euclid's algorithm). A function a(x) is invertible mod z if:
$\mathrm{a}(\mathrm{a}^{-1}(x)) \equiv x(modz) \quad and \quad a^{-1}(a(x)) \equiv x(modz)$

Let $f_p(x)$ denote the inverse of f(x) mod p.

Let $f_q(x)$ denote the inverse of f(x) mod q.

2.2 Public key generation

The public key is derived from the private key by calculating h(x) defined as:
$\mathrm{h}(\mathrm{x}) = \mathrm{g}(\mathrm{x}) \cdot \mathrm{f}_q(\mathrm{x}) \bmod \mathrm{q}$

2.3 Encryption

To encrypt a message m(x), which has coefficients smaller than p, a random polynomial r(x) is generated with coefficients much smaller than q. The ciphertext c(x) is then computed as:
c(x) = p $\cdot h(x) \cdot r(x) + m(x) mod q$

The multiplication of these high order polynomials can be done quickly and efficiently using Fast Fourier Transform.

2.4 Decryption

To decrypt the cyphertext c(x) the following steps are taken:
c(x)$\cdot f(x) = p \cdot g(x) \cdot r(x) + m(x) \cdot f(x) \quad mod q$
$Therefore:$
$c(x) \cdot f(x) = m(x) \cdot f(x) mod p$
$So:$
$m(x) = c(x) \cdot f(x) \cdot f_p(x) mod p$

2.5 Advantages

The advantages of Falcon are:

- It uses compact public key and signature sizes compared to other post-quantum ciphers.
- The verification of Falcon signatures is fast and efficient.
- Signing is relatively fast in Falcon, altough not as fast as Dilithium.

2.6 Disadvantages

However, the disadvantages of Falcon are:

- Key generation and signing of Falcon require floating-point arithmetic which can be problematic for some computers in some scenarios.
- Unlike other schemes, Falcon does not support certain types of masking for counteracting side-channel attacks, making it more vulnerable to these types of attacks.
- Key generation and signing are complex to implement which can be challenging for developers.
- Although signing and verification are fast, the process of generating keys in Falcon can be slower compared to other cryptographic schemes.

2.7 code

https://github.com/tprest/falcon.py/blob/master/falcon.py