Gen Key

Bắt đầu: Thuật toán tạo một seed ngẫu nhiên gồm 256-bit bằng approved RBG (Random Bit Generator), seed này được mở rộng khi cần bằng cách sử dụng XOF (cụ thể là SHAKE-256) để tạo ra các giá trị ngẫu nhiên khác.

- Một seed ngẫu nhiên công khai ρ: Sử dụng seed này tạo ra một ma trận đa thức A.
  - Ma trận $A \in R_q^{k \times l}$ được tạo ra ngẫy từ ρ.
- Một seed ngẫu nhiên riêng tư ρ′: Sử dụng seed này để tạo ra vector đa thức s_1 và s_2.
  - Các vector $s_1 \in R_q^l$ và $s_2 \in R_q^k$ được lấy mẫu ngẫu nhiên từ ρ′ với các hệ số ngắn (trong khoảng $[-\eta, \eta]$).
- Một seed ngẫu nhiên riêng tư K: Dùng trong quá trình ký văn bằng

Tiến hành tính toán giá trị công khai t:

- $t = As_1 + s_2$

Nén vector t:

- Vector t cùng với ma trận A xem như là một dạng mở rộng của public key.
- Vector t được nén trong public key bằng cách bỏ đi các bit ít quan trọng nhất từ mỗi hệ số, từ đó tạo ra vector đa thức nén t_1

Tính toán mã băm của khoá công khai:

- Tạo mã băm 512-bit tr của khóa công khai để sử dụng trong khi thực hiện việc ký

Tạo khoá công khai public key:

- Public key được tạo ra là một mảng byte từ seed ngẫu nhiên p và vector đa thức nén t_1

Tạo khoá riêng tư private key:

- Private key sẽ bao gồm ρ, K, tr, s1, s2 và thông tin để tái tạo các bit ít quan trọng của t.

Signing

Đầu vào: Khoá riêng tư private key và thông điệp M (cả hai đều là chuỗi byte)

Đầu ra: Chữ ký dưới dạng chuỗi byte

Trích xuất từ khoá riêng tư:

- Seed ngẫu nhiên công khai ρ để tạo ma trận đa thức A
- Seed ngẫu nhiên riêng tư K để dùng trong quá trình ký
- Mã băm 512-bit của khoá công khai tr
- Các vector đa thức bí mật s_1 và s_2 dùng để tính toán giá trị công khai t
- Vector đa thức để mã hoá các bit ít quan trọng của t

Tạo đại diện thông điệp μ:

- Nối thông điệp M với mã băm của khoá công khai tr.
- Băm chuỗi kết quả thành một đại diện cho thông điệp 512-bit μ.

Tạo seed ngẫu nhiên để ký:

- Seed ngẫu nhiên 512-bit ρ′ được dùng trong mỗi lần ký.

Vòng lặp lấy mẫu chấp nhận:

- Tạo ra chữ ký hợp lệ hoặc tiếp tục vòng lặp nếu không hợp lệ

  Các bước trong vòng lặp:

- Tạo commitment w_1
  - Sử dụng hàm ExpandMask: Tạo một vector đa thức y có các hệ số trong khoảng $[-\gamma+1, \gamma]$.
  - Tính toán commitment $w_1 = Ay$ và làm tròn đến bội số gần nhất của $2^2\gamma$
- Tạo commitment hash
  - Nối w_1 và μ.
  - Băm chuỗi kết quả tạo ra commitment hash c.
- Lấy mẫu đa thức c:
  - Sử dụng w1Encode: Mã hoá commitment hash.
  - Lấy mẫu đa thức c: Từ tập hợp các đa thức có hệ số trong {-1, 0, 1}.
- Tạo phản hồi z:
  - Tính toán phản hồi $z = y + cs_1$
  - Thực hiện các kiểm tra xem có hợp lệ không: Nếu bất kỳ kiểm tra nào không đạt sẽ tiếp tục vòng lặp để lấy được mẫu chấp nhận chính xác
- Tạo đa thức hint h:
  - Nếu các kiểm tra đã hợp lệ: Tạo một đa thức hint h, nhằm xác thực được w_1 và sử dụng khoá công khai để nén
- Xuất chữ ký:
  - Chữ ký bao gồm: Commitment hash c, phản hồi z và hint h

Verify

Đầu vào:

- Public key (pk) được mã hoá thành chuỗi byte
- Thông điệp M
- Chữ ký σ được mã hoá thành chuỗi byte

Đầu ra:

- True (nếu chữ ký hợp lệ)
- False (nếu chữ ký không hợp lệ)

Các bước:

- Giải mã thông tin từ public key và chữ ký σ
    - Trích xuất seed ngẫu nhiên ρ và vật thể đa thức nén t_1 từ public key pk
    - Trích xuất commitment hash c, phản hồi z và hint h từ chữ ký.
- Kiểm tra tính hợp lệ của hint h:
    - Nếu hint h không được mã hoá byte chính xác, thuật toán sẽ trả về false ngay lập tức, cho biết chữ ký không hợp lệ.
- Xử lí thông điệp M:
    - Tạo đại diện thông điệp: Băm chuỗi nối của tr (băm public key) và thông điệp M.
- Xây dựng lại commitment của người ký
    - Tính toán giá trị xấp xỉ w'
        - Tạo mẫu đa thức c như Sign
        - Tính toán w'_{Approx}: $Az - ct_1 * 2^d$
        - Sử dụng hint h để lấy w' từ w'_{Approx}
- Xác minh chữ ký có hợp lệ hay không:
    - Kiểm tra xem phản hồi z và hint h của người ký có hợp lệ không:
        - Xác minh rằng tất cả hệ số của z đều đủ nhỏ (tức là nằm trong phạm vi $[-(\gamma - \beta), \gamma - \beta]$).
        - Xác minh rằng h không chứa nhiều hơn o hệ số khác 0
    - Kiểm tra xem w' được xây dựng lại có đồng nhất với commitment hash c hay không.

Sau đó trả về giá trị true false dựa trên việc kiểm tra