Spécifications Mathématiques : Protocole Collatz-Trapdoor v2.0

Ce document détaille les fondements arithmétiques du système cryptographique basé sur la dynamique de Collatz perturbée par injection de bruit déterministe.

1. Définition de l'Opérateur Perturbé

Soit $\mathbb{I} = \{2n+1 \mid n \in \mathbb{N}\}$ l'ensemble des entiers impairs positifs. Pour renforcer la sécurité contre l'analyse 2-adique, nous définissons l'opérateur de Collatz avec erreur U^* :

$$U^*(n, e_i) = \frac{3n+1+2^{v_2(3n+1)} \cdot e_i}{2^{a_i}}$$

Où :

- $v_2(x)$ est la valuation 2-adique de x.
- e_i est un élément du **vecteur d'erreur** secret $\vec{e}$.
- a_i est l'exposant de normalisation tel que le résultat reste dans $\mathbb{I}$.

2. Construction des Clés (Basée sur les Réseaux)

Le système passe d'une simple itération à un problème de type **Learning With Errors (LWE)** sur les entiers.

A. Clé Privée

Le secret est un couple $(S, \vec{e})$:

- $S \in \mathbb{I}$: Point de départ de grande taille (ex: 2048 bits).
- $\vec{e} \in \{0,1\}^k$: Vecteur de bruit binaire secret de dimension k.

B. Clé Publique

La clé publique est le triplet (P, k, λ) :

- $P = U^*(...U^*(S, e_0)..., e_{k-1})$: Point d'arrivée après k itérations.
- k : Nombre d'itérations (profondeur du réseau).
- λ : Paramètre de sécurité (somme publique des valuations).

C. Relation de Trapdoor

La trajectoire suit la relation fondamentale perturbée :

$$2^{A_k} \cdot P = 3^k \cdot S + C_k + \sum_{j=0}^{k-1} 3^{k-1-j} \cdot 2^{A_j} \cdot e_j$$

L'attaquant ne peut pas résoudre S car il ignore les erreurs e_j injectées à chaque étape.

3. Mécanisme d'Encapsulation de Clé (KEM)

Le chiffrement direct est remplacé par une dérivation de clé pour assurer le secret de l'information (Forward Secrecy).

1. **Génération du Secret :** L'expéditeur choisit un nonce r.
2. **Calcul du Point Partagé :** Il calcule $K_{seed} = \text{Hash}(U^{(k)}(r) \oplus P)$.
3. **Encapsulation :** Le message M est chiffré par $C = \text{AES-GCM}(K_{seed}, M)$.

4. Signature Numérique (Fiat-Shamir)

Pour signer un message m, l'utilisateur prouve qu'il connaît S sans le révéler :

1. **Engagement :** Choisir $t \in \mathbb{I}$, calculer $T = U^{(k)}(t)$.
2. **Défi :** $c = \text{Hash}(T \parallel m)$.
3. **Réponse :** $\sigma = t + c \cdot S$.

4. **Vérification :** Le vérificateur s'assure que la trajectoire de σ est cohérente avec T et P via la linéarité de l'opérateur sur k pas.

5. Analyse de Sécurité Post-Quantique

A. Réduction au Problème du Plus Court Vecteur (SVP)

L'injection du vecteur $\vec{e}$ transforme l'inversion de la fonction en une recherche de cible dans un réseau euclidien de dimension k. Ce problème est prouvé difficile, même pour les algorithmes quantiques (Shor ne s'applique pas).

B. Confusion et Diffusion

- **Confusion :** L'aléa binaire e_i masque la séquence de parité réelle.
- **Diffusion :** Une modification d'un seul bit de S ou de $\vec{e}$ change radicalement P (effet avalanche).

6. Paramètres Recommandés

Paramètre	Valeur Standard	Valeur Haute Sécurité
Taille de S	2048 bits	4096 bits
Itérations k	512	1024
Taille du Bruit	128 bits	256 bits

Note : Le bannissement des nombres de Mersenne et des entiers n tels que $3n + 1 = 2^x$ reste impératif pour éviter les raccourcis arithmétiques.