Accréditation anonyme et coloriage de graphe

Contexte

> Un utilisateur souhaite convaincre un vérificateur qu'il connaît une manière de colorier entièrement un graphe avec 3 couleurs de telle façon qu'aucun noeud du graphe n'ait la même couleur que l'un des noeuds voisins mais sans révéler aucune autre information à propos de ce coloriage.

Questions

```
1. Pourquoi est ce que si l'utilisateur et le vérificateur sont honnêtes et suivent les directives du protocole, un utilisateur possédant la preuve d'un 3-coloriage pourra toujours convaincre le vérificateur (propriété de completeness) ?
```

Si l'utilisateur et le vérificateur sont honnêtes, le calcul du hashé de (ri, ci) est bien égal à yi. En effet, `ri` correspond à du sel généré pour le noeud i et `ci` la couleur pour le noeud i. Et `yi` le résultat du hash de (ri, ci) créé et envoyé par l'utilisateur au vérificateur.
De plus, une solution valide vérifie toujours que la couleur du noeud i est différente de la couleur du noeud j.

```
2. Pourquoi est ce qu'un utilisateur qui ne possède pas de preuve d'un 3-coloriage ne pourra pas réussir à convaincre un vérificateur, sauf avec une probabilité négligeable (propriété de soundness) ?
```

L'étape d'engagement impose que les couleurs des noeuds doivent être fixées avant d'effectuer la mise en gage. L'utilisateur qui ne connaît pas de coloriage doit donc en tirer un aléatoirement.

Puis, le vérificateur demande les couleurs de deux noeuds adjacents choisies aléatoirement. La probabilité que ces deux noeuds soient de la même couleur est donc de 1/3. Et donc la probabilité que les noeuds soient de couleur différente (succès) est de 2/3 (puisqu'il n'y a que 3 couleurs).

En répétant la procédure 400 fois, la probabilité de succès pour l'utilisateur ne connaissant pas de solution est donc très faible.

```
3. Expliquer en quoi ce protocole est à divulgation nulle (zero-knowledge en anglais), c'est à dire qu'il n'apporte aucune autre information au vérificateur que la véracité de l'énoncé.
```

Tout d'abord grâce à l'**engagement**: Le vérificateur ne reçoit que la **mise en gage** dans laquelle les informations des couleurs des noeuds sont hashées avec du sel.

Ensuite grâce à la procédure: Le vérificateur sélectionne deux noeuds adjacents, et l'utilisateur lui renvoie les clés (sel, couleur) des deux noeuds sélectionnés au hasard. Puisque les couleurs ont été permuntés, les

véritables couleurs n'ont pas été divulguée.

Enfin grâce à la permutation: Un espion ne pourra pas établir de corrélations entre les couleurs du graphe se basant sur les itérations.

Aucune information concernant la solution de l'utilisateur n'est alors transmise au vérificateur.