Structures de données (LU2IN006)

Cours 10 : Structures de données sécurisées

Nawal Benabbou

Licence Informatique - Sorbonne Université

2022-2023

Fonction de hachage cryptographique

Présentation

Une fonction de hachage cryptographique est une fonction de hachage déterministe qui retourne une valeur de hachage de taille fixe, qu'importe la taille du message (données) en entrée, et qui doit être "facile" à calculer mais impossible à inverser (en pratique). Autres propriétés désirables :

- Modifier légèrement le message modifie grandement la valeur de hachage.
- Il est impossible de trouver un message possédant la même valeur hachée qu'un message donné.
- Il est impossible de trouver deux messages avec la même valeur hachée.

Selon les applications, on peut vouloir imposer certaines de ces propriétés, conduisant à des niveaux de sécurité différents.

Exemples connus : les fonctions de hachage *Message Digest* (MD4, MD5...) ou les fonctions *Secure Hash Algorithm* (SHA-1, SHA-2...).

Remarque : lien avec les tables de hachage

On pourrait utiliser une fonction de hachage cryptographique dans le cadre d'une table de hachage (cf. cours 3), mais comme elles sont assez coûteuses en temps de calcul, on ne les utilise en pratique que dans des contextes où il est nécessaire de se protéger contre des falsifications.

Quelques applications

Vérification de mots de passe

Stocker des mots de passe en clair constitue une faille de sécurité. À la place, on pourrait stocker la valeur hachée des mots de passe. Quand un utilisateur souhaite se connecter, on applique la fonction de hachage cryptographique sur le mot de passe renseigné par l'utilisateur, et on compare le résultat avec la valeur de hachage stockée.

Catalogue de données complexes

Quand les données à stocker sont complexes, on peut vouloir les résumer par leur valeur de hachage (taille fixe). De cette manière, quand on souhaite savoir si une donnée est présente dans la base, il suffit de lui appliquer la fonction de hachage et de comparer le résultat avec toutes les valeurs de hachage stockées.

Signature électronique

Pour créer une signature électronique, on utilise souvent une méthode de chiffrement que l'on applique sur la valeur de hachage du message (pour des questions d'efficacité). Cette signature est ensuite envoyée en même temps que le message. De cette manière, le récepteur peut vérifier l'identité de l'émetteur en déchiffrant la signature, le résultat devant correspondre à la valeur de hachage du message qu'il a reçu.

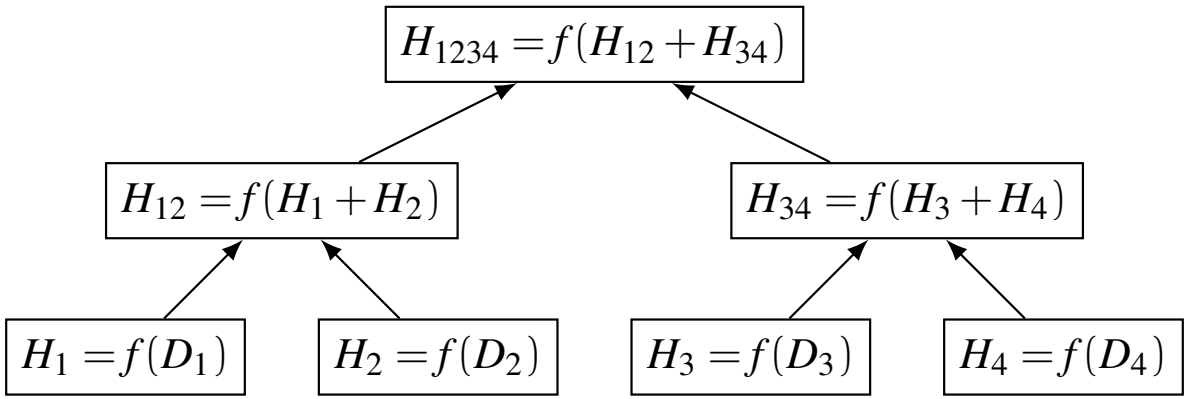
Les arbres de hachage

Autre application : vérification de l'intégrité des données.

Arbres de hachage

Un arbre de hachage, aussi appelé arbre de Merkle, est une structure de données qui permet de résumer un ensemble de données de manière à pouvoir vérifier efficacement leur intégrité. Il se présente sous la forme d'un arbre (binaire ou non), où chaque feuille contient la valeur de hachage d'une partie des données (obtenue par application d'une fonction de hachage cryptographique sur ces données). Puis, la valeur de hachage de chaque noeud interne est obtenu en applicant la fonction de hachage cryptographique sur la concaténation des valeurs de hachage de ses fils.

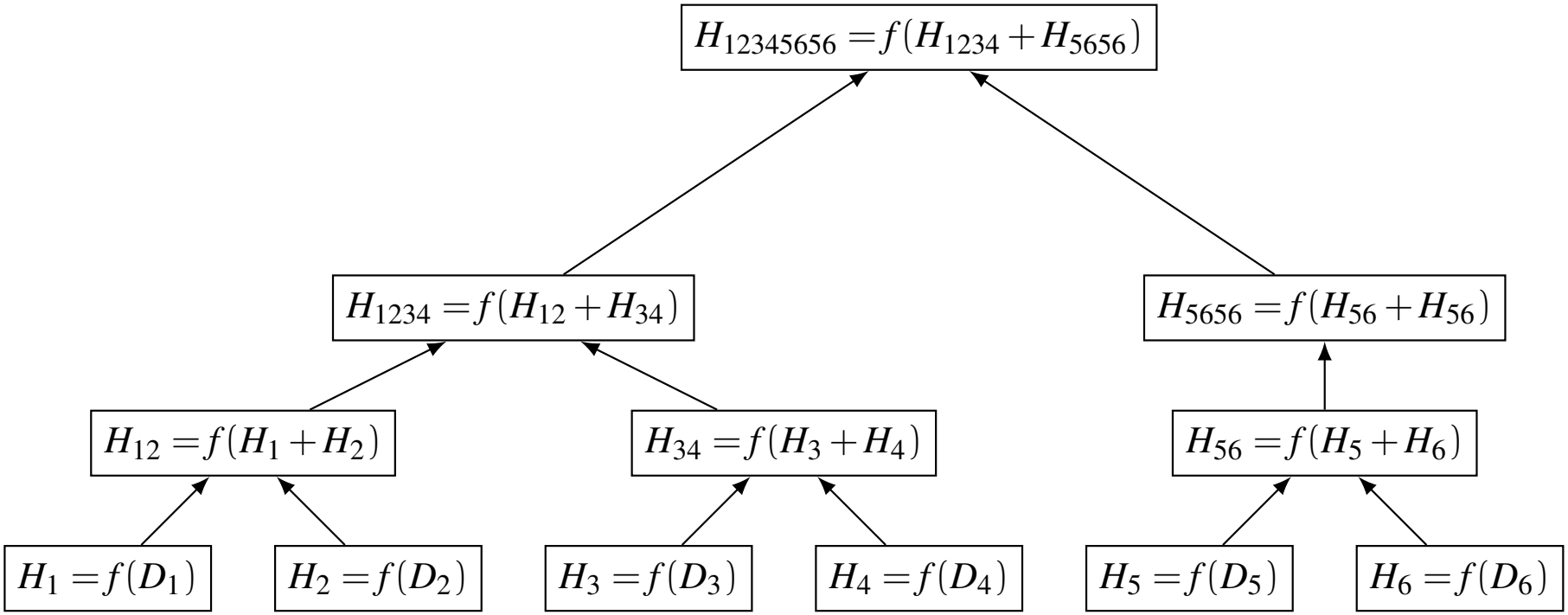
Illustration : un arbre de Merkle permettant de résumer quatre blocs de données (D_1, D_2, D_3 et D_4) par le biais d'une fonction de hachage f.



Les arbres de hachage

Remarque :

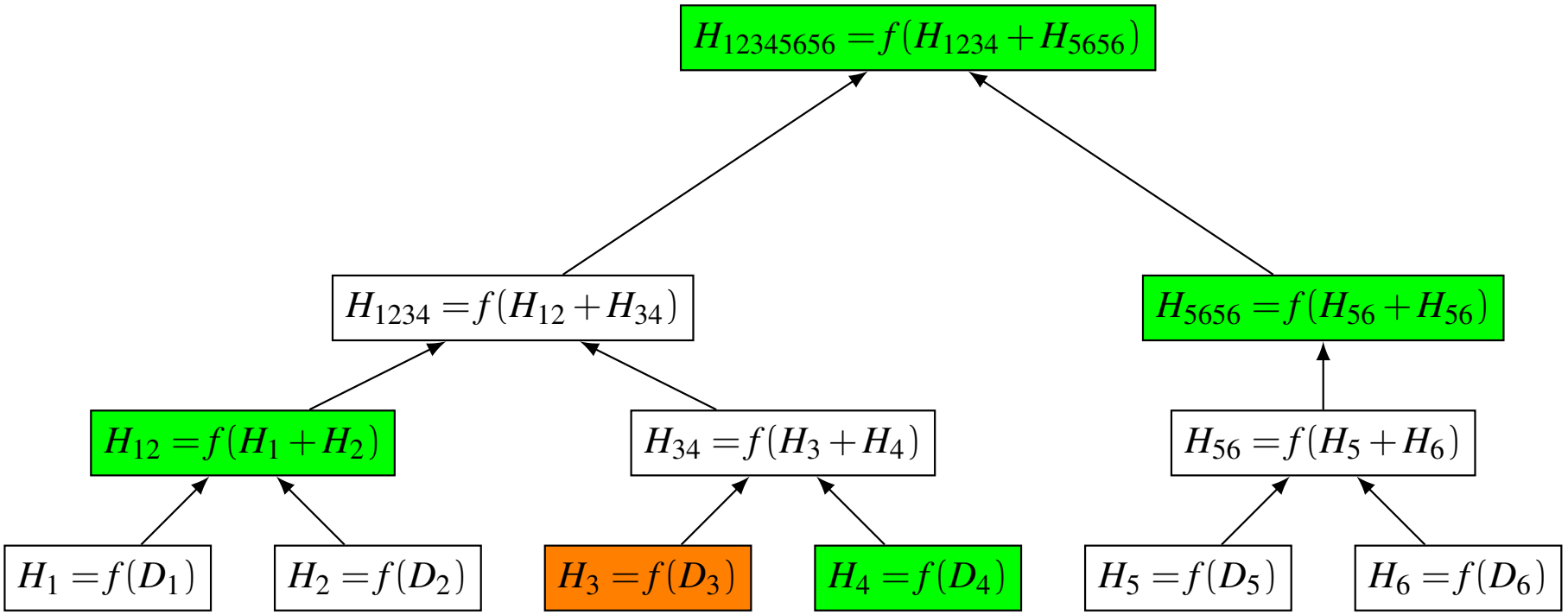
Les arbres de hachage sont construits itérativement, niveau par niveau, des feuilles vers la racine. Si à un niveau, le nombre de noeuds est impair, on duplique le hash du dernier noeud pour créer son père.



Les arbres de hachage

Vérification de l'intégrité des données avec un arbre de hachage

Pour vérifier que les données n'ont pas été corrompues, il suffit de connaître la valeur de hachage de la racine de l'arbre et la fonction de hachage qui a été utilisée. En effet, si en construisant l'arbre de hachage à partir des données, on n'obtient pas la même valeur de hachage à la racine, alors les données ont été corrompues. En cas de doute sur une partie des données, il suffit de connaître les valeurs de hachage des noeuds intervenant dans la construction du chemin de ces données vers la racine.



Blockchain

Autre application des fonctions de hachage cryptographiques : les blockchains.

Présentation

Une blockchain, ou chaîne de blocs, est une structure de données distribuée, sécurisée et qui fonctionne sans organe de contrôle. Une blockchain organise les données en bloc. Chaque bloc contient une partie des données à stocker et des méta-données qui comprennent notamment une référence vers le bloc précédent. Cette référence est obtenue en applicant une fonction de hachage cryptographique sur les méta-données du bloc précédent.

Comparaison avec une liste chaînée

Une blockchain peut être vue comme liste chaînée, où chaque bloc correspond à un élément de la liste, et où les pointeurs précédents sont sous forme de valeurs de hachage. Néanmoins, avec une blockchain, on peut difficilement :

- modifier un bloc quelconque, car cette modification change sa valeur de hachage, ce qui nécessite de changer la référence contenu dans le bloc suivant, qui aura donc lui aussi une autre valeur de hachage, etc.
- supprimer/ajouter un bloc à une position quelconque, pour les mêmes raisons que dans le cas précédent.

Une blockchain ne permet donc en pratique que des ajouts en fin de liste.

Fonctionnement d'une blockchain

Principe général

Chaque utilisateur possède une copie intégrale de la blockchain en local, ainsi qu'une clé privée lui permettant de signer ses transactions (et une clé publique permettant aux autres de l'identifier). Certains utilisateurs, souvent appelés mineurs, ont pour rôle de créer des blocs regroupant des transactions valides en attente. Quand un bloc est créé par un mineur, il est horodaté puis le transmis à tous les utilisateurs du réseau, qui doivent alors mettre à jour leur blockchain locale en ajoutant le nouveau bloc à la fin de leur blockchain.

Algorithme de consensus

Pour sécuriser une blockchain et la rendre fiable, il est nécessaire de mettre en place un algorithme de consensus, c'est-à-dire un mécanisme permettant de mettre d'accord les utilisateurs sur la création et l'enchaînement des blocs en cas d'apparition de bifurcations/branches. Par exemple, dans un mécanisme de consensus type "proof of work", il est demandé aux mineurs de réaliser un travail nécessitant de fournir une certaine puissance de calcul pour créer un bloc, et le consensus consiste à choisir la chaîne de blocs correspondant à la plus grande quantité de travail accumulée (ou la branche la plus "longue"). De cette manière, un agent malveillant souhaitant faire accepter sa branche ne pourra fournir une puissance de calcul suffisante face aux autres mineurs qui mettent leur puissance en commun en créant des blocs sur la "vraie" branche.

Structure d'un bloc

Un bloc est divisé en deux parties : une en-tête et les données.

En-tête (méta-données)

L'en-tête d'un bloc contient plusieurs informations, notamment :

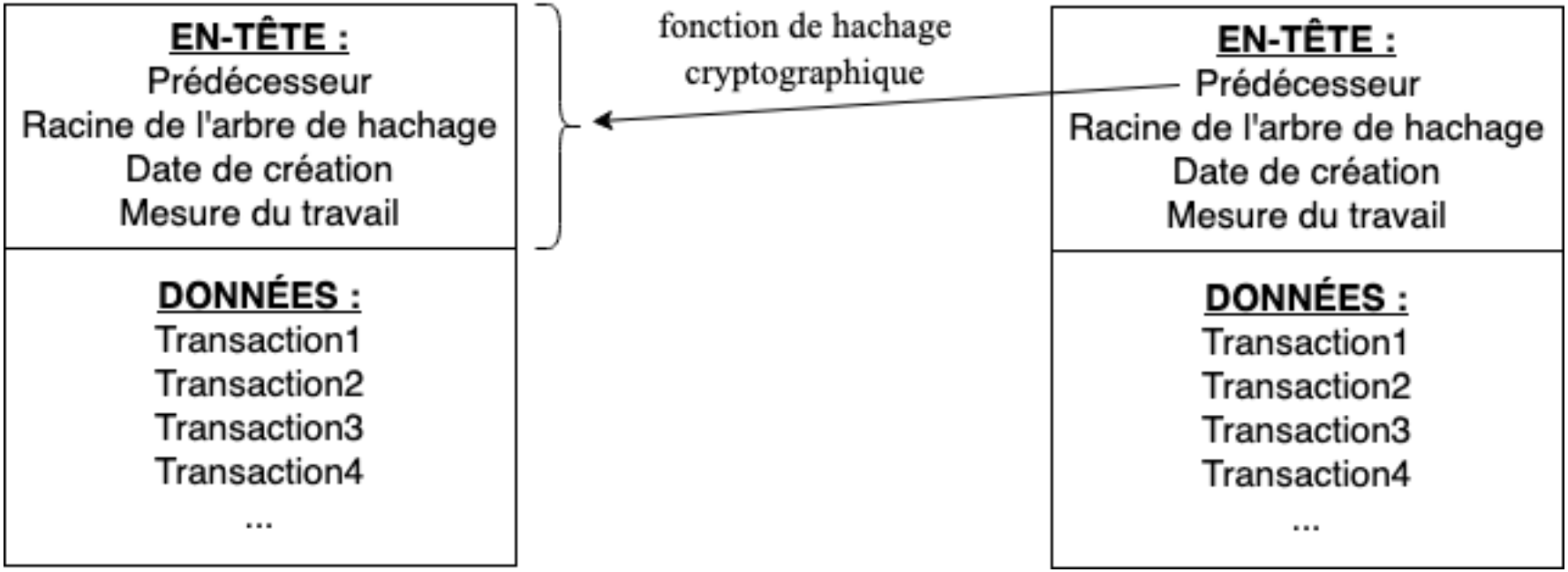
- Une référence sur le bloc précédent (valeur de hachage de son en-tête).
- Un résumé des données du bloc (ex : sous forme d'arbre de hachage).
- La date de création du bloc (en heures Unix).
- Une mesure de la quantité de travail nécessaire pour produire le bloc.

Les données

Dans une blockchain, les données sont souvent appelées transactions. Chaque transaction contient notamment :

- Les données en elle-même.
- Une signature électronique (réalisée avec la clé privée de l'utilisateur).
- Un moyen d'identifier l'auteur (clé publique de l'utilisateur).

Illustration graphique



Remarque : Dans l'en-tête, ajouter la racine de l'arbre de hachage suffit à garantir l'intégrité des données. En effet, en cas de modifications des données, la racine de l'arbre de hachage change, ce qui a pour conséquence de changer la valeur de hachage de l'en-tête du bloc et donc la référence sur ce bloc (ce qui se répercute sur le bloc suivant, etc.).

Défis techniques liés à la blockchain

- **Identité numérique :** Il est nécessaire de pouvoir certifier l'identité des utilisateurs de la blockchain, tout en garantissant l'anonymat et la traçabilité des transactions.

- **Le passage à l'échelle :** selon l'algorithme de consensus utilisé, la validation des données peut être longue. Par exemple, Bitcoin qui utilise un mécanisme par "proof of work" permet de ne valider qu'une poignée de transactions par seconde, contrairement à une banque qui peut en gérer plusieurs centaines. Réduire la quantité de travail nécessaire à la création de blocs est une solution, mais qui risque de dégrader la sécurité des données.

- **Enjeu environnemental :** les opérations de validation/création de blocs sont coûteuses en énergie électrique, ce qui a un impact négatif sur l'environnement. Il s'agit donc de trouver des protocoles de consensus moins coûteuse en énergie, offrant les mêmes garantie en terme de sécurité des données.