FONCTIONS DE HACHAGE

Une fonction de hachage est une fonction mathématique, très souvent utilisée dans le domaine de la cryptographie, qui permet de calculer ce que l’on appelle une empreinte numérique (un hash) de taille fixe qui est une chaine de bits permettant de caractériser un fichier de taille quelconque, au même titre qu'une empreinte permet d’identifier une personne.

Il existe différents types de fonction de hachage, les plus connues étant les algorithmes MD5 et SHA-2 ([SHA-256](https://fr.wikipedia.org/wiki/SHA-256), [SHA-384](https://fr.wikipedia.org/wiki/SHA-384) ou [SHA-512](https://fr.wikipedia.org/wiki/SHA-512) selon le nombre de bits).

Le MD5 (Message Digest 5) est développé par Ronald Rivest en 1991.

Le SHA-2 Quant à lui a été développé par la NSA et a été rendu publique en 2002. C’est une famille de fonction de hachage, ils sont tous assez similaire et inspiré de leurs anciennes versions, le SHA-1.

Par exemple, la fonction [SHA-256](https://fr.wikipedia.org/wiki/SHA-256) est celle utilisée pour la blockchain du bitcoin.

Un hash est une suite d’un certain nombre de bits. Pour faciliter la lecture d’un hash, on traduit la chaine de bits en hexadécimal.

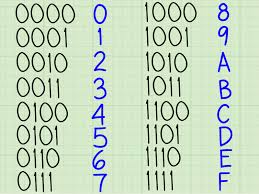


Tableau de conversion binaire-hexadécimal

De ce fait, suite à cette conversion, pour un hash mesurant 256 bits, il ne restera plus qu’une chaine de 64 caractères.

On utilise des empreintes suffisamment longue (aux moins 128 bits) pour justement éviter qu’il y ait ce que l’on appelle des collisions.

Un hash est une solution au problème de la signature d’un fichier. En effet, généralement une signature électronique a la même taille de stockage que son fichier. Donc si l’on souhaite réaliser la signature électronique pour un fichier de 10Go, alors la signature fera également 10Go.

Or, si l’on ne choisit de prendre que le hash de la signature électronique on gagne énormément de place de stockage car un hash ne prend que très peu d’espace de stockage. En ce qui concerne la sécurisation des mots de passe, il faut savoir qu’un disque dur ne stocke pas les mots de passe eux même, mais plutôt leur hash. Donc lors d’une authentification on compare le hash du mot de passe avec le hash de la chaine de caractères saisie.

Cela sert également à vérifier l’intégrité d’un fichier (savoir si le fichier n’a pas été modifié) car si un fichier est modifié alors son hash change automatiquement. Cela permet donc de gagner un temps considérable.

Enfin, cela permet également de sécuriser les BlockChain.

Afin d’obtenir une chaine de bits pour un fichier, la fonction de hachage va procéder à de nombreuses applications que l’on appelle « portes logiques ».

Les portes logiques sont des circuits combinatoires simples. Elles sont la base de la logique mathématique qui effectue les opérations à l'intérieur du processeur, c’est la base de tous les calculs internes du processeur. Les portes logiques sont à l'origine de tous les calculs effectués dans les transistors. Leur fonctionnement étant basé sur le passage éventuel du courant, elles ne peuvent que traiter des informations en langage binaire. Enfin l'association de portes logiques permet de traiter une instruction du microprocesseur.

C’est donc grâce à ces applications que l’on obtient des chaines de bits.

De façon générale, une fonction de hachage va « découper » un fichier en différents morceaux d’un certain nombre de bits puis elle va lui appliquer différentes applications selon l’algorithme utilisé.

La fonction de hachage est effectuée dans le but de pouvoir résister aux collisions, c’est-à-dire deux fichiers qui donnent le même hash.

Il y a tout de même un problème, nous n’avons actuellement aucune preuve mathématique qui permet de prouver qu’il existe une fonction de hachage pour laquelle il serait difficile, voire impossible de calculer des collisions. Il est même dit que mathématiquement, une fonction de hachage admet une infinité de collision.

Il faut donc faire attention avec la notion de collision car intuitivement nous pourrions penser que ces dernières sont impossibles grâce au nombre de combinaisons de hash possibles. Or, mathématiquement c’est tout le contraire.

Ce paradoxe se rapproche du paradoxe des anniversaires dans lequel on penserait intuitivement que dans une classe de trente élèves il n’y a presque aucune chance que deux élèves aient la même date de naissance. Or, mathématiquement il y a 70% de chance que deux élèves aient la même date de naissance. Cela prouve donc qu’il ne faut pas toujours se fier uniquement à notre intuition et que ces fonctions peuvent effectivement avoir des limites.

Par exemple, le MD5 n’est plus considérée comme sur car en 1996, une faille a été découvert où on pouvait réaliser des colissions à la demande au bout de 2^44 opérations. De plus, en 2004, une équipe de recherche chinois a trouvé un moyen afin de calculer des collisions pour l’algorithme MD5 en une heure. Il est donc à présent déconseillé d’utiliser cet algorithme.

Cependant, il existe plusieurs algorithmes de hachage, donc si ce problème survient sur un algorithme, il suffit de le remplacer cet algorithme par un autre.

Pour assurer le bon fonctionnement d’une fonction de hachage, celle-ci doit provoquer un « effet avalanche » lorsqu’une modification est fait sur le fichier. L’effet avalanche est une propriété provoquant des modifications importantes au sein de l’algorithme de hachage si l’on modifie ne serait ce qu’un seul bit en entrée. De ce fait, le hash obtenu en sortie de fonction est totalement différent du hash précédent. Cette propriété accentue donc l’aspect chaotique des fonctions de hachage.

Une fonction de hachage est dite à sens unique (ou asymétrique), c’est-à-dire que le calcul de la fonction inverse est impossible. On parle alors de résistance à la préimage (ce sont les éléments dans l'ensemble de départ qui sont liés à un certain élément de l'ensemble d'arrivée) : on ne peut pas passer par l’empreinte pour arriver au fichier original.

La résistance à la seconde préimage consiste, elle, à ne pas trouver un même hash pour deux fichiers différents.

Beaucoup de fonction de hachage sont associés à des clés de cryptage, permettant aux utilisateurs de crypter ou décrypter les fichiers. Ces clés peuvent-être de deux types différents : publique ou privé.

* Une clé privée est obtenue en convertissant une portion de texte générée automatiquement en un fichier clé à l'aide d'une fonction de hachage, ce qui lui donne une valeur unique. Cette clé permet de décrypter les données cryptées et de crypter les messages signés par un contrat.
* Une clé publique est utilisée afin de crypter des informations que seul le propriétaire de la clé privée est autorisé à recevoir. La combinaison unique de la clé publique et de la clé privée peut alors décrypter ces données. Une clé publique peut également être utilisée pour vérifier qu'un message a été envoyé par le propriétaire de la clé privée.

Il existe de catégorie de fonction de hachage :

* Sans clé : Elles sont appelées MDC (Modification Detection Code), elles permettent de vérifier grâce à l’empreinte si un fichier à était modifier ou non (si l’empreinte est la même pour les deux fichiers alors il n’a pas été modifier).
* Avec clé : Elles sont appelées MAC (Message Authentification Code), elles ont une clé (privée ou publique) en paramètre.