Comprendre Bitcoin
Rapport de projet INFRES357

Pierre Galland, Benoît de Laitre

18 avril 2015

Table des matières

1 Briques de base

Le protocole de Bitcoin utilise plusieurs briques de bases de la cryptographie : le chiffrement asymétrique, la signature et le hachage. C'est d'ailleurs là que réside l'intérêt de ce protocole, c'est un assemblage astucieux de briques qui existent depuis de nombreuses années et qui sont très bien connues, pourtant cet assemblage crée une technologie totalement nouvelle.

1.1 Chiffrement asymétrique

Le principe du chiffrement asymétrique est que les clefs vont par paire, une clef publique et une clef privé (K_{Pub}, K_{Pri}). On peut voir les clefs comme des suites finies de caractères ou de chiffres, et les message aussi. Il est possible de chiffrer un message avec la clef privé et alors le message chiffré ne pourra être déchiffré qu'avec la clef publique correspondante. De même on peut chiffrer un message avec la clef publique et alors il ne pourra être déchiffré qu'avec la clef privé correspondante. Considérons l'exemple classique de Bob et Alice, ils possède chacun une paire clef publique/clef privée. La paire de clefs de Bob est (K^B_{Pub}, K^B_{Pri}) et celle d'Alice est (K^A_{Pub}, K^A_{Pri}).

Si Bob veut envoyer un message $mess$ à Alice qu'elle seule pourra lire, alors il chiffre son message avec le clef publique d'Alice, et il obtient le message chiffré $*mess*$:

$$chiffrer(mess,\ K^A_{Pub}) \rightarrow *mess*$$

Il envoie alors ce message $*mess*$ à Alice, qui quand elle le reçoit le déchiffre avec sa clef privée K^A_{Pri} :

$$déchiffrer(*mess*,\ K^A_{Pri}) \rightarrow mess$$

Si Alice essayait de déchiffrer le message $*mess*$ avec une autre clef que sa clef privée K^A_{Pri} alors cela ne marcherait pas, elle obtiendrait n'importe quoi (une suite de symbole qui n'a rien à voir avec le message $mess$). Donc comme on considère qu'Alice est la seule à connaître sa clef privée, elle est la seule à pouvoir déchiffrer le message.

$$déchiffrer(*mess*,\ K^{autre}_{Pri}) \rightarrow n'importe\ quoi$$

1.2 Hachage cryptographique

On considère toujours nos messages comme des suites finies de caractères ou de chiffres (d'ailleurs in fine sur un ordinateur toute donnée est une suite finie de chiffres). Le but d'une fonction de hachage cryptographique h est de transformer chaque message de taille inférieure à T (où T est très très grand) en un message de taille beaucoup plus petite que T (par exemple dans Bitcoin en message de longueur 256 chiffres). Si j'ai un message $mess$ assez long alors $h(mess)$ sera beaucoup plus petit que $mess$. On appellera $h(mess)$ le hash de $mess$.

Prenons un exemple avec la fonction de hachage SHA-256, qui est utilisée dans Bitcoin. Je considère le message $mess$ suivant et je calcule son $h(mess)$:

$$mess = Un\ message\ pas\ très\ long,\ juste\ pour\ faire\ un\ exemple$$

$$h(mess) \rightarrow 6423e8584411fee28e5064799d8a230c64f999c448c769ab7d309baba9a33f42$$

Le but de la fonction de hachage est également que le hash d'un message puisse l'identifier de manière presque unique. Ainsi à priori si je considère deux messages différents alors leurs hashs sont également différents.

$$mess1 \neq mess2 \Rightarrow presque\ toujours\ h(mess1) \neq h(mess2)$$

Un autre point important est que si je connais seulement le hash du message $h(mess)$ ainsi que la fonction de hachage h mais que je ne connais pas le message $mess$ alors je ne puisse pas retrouver quel est le message $mess$. Il est impossible (cela demande trop de puissance de calcul) de retrouver le message à partir de son hash.

$$h\ et\ h(mess) \nrightarrow mess$$

Enfin si je considère un message $mess1$ et ma fonction de hachage h, il m'est impossible (une fois encore cela demande trop de puissance de calcul) de trouver un message $mess2$ tel que $h(mess1) = h(mess2)$.

Les fonctions de hachage jouent un rôle très important dans le protocol Bitcoin !

1.3 Signature

2 Un paiement Bitcoin

3 Diffuser les paiements

4 Empêcher le double-spend

5 Le minage