Optimisation des modes opératoires en arbre pour le hachage parallèle

Une étude de cas

Kevin Atighehchi [1] Robert Rolland [2]

[1] Aix-Marseille Université, CNRS, LIF UMR 7279

[2] Aix-Marseille Université, CNRS, I2M UMR 7373

30 mars 2017

Introduction

Utilisations de primitives cryptographiques pour assurer certaines propriétés :

- confidentialité
- authenticité
- d'autres propriétés...

Besoins de performances :

- systèmes critiques
- transparence de l'emploi des mécanismes cryptographiques

Sommaire

Sommaire

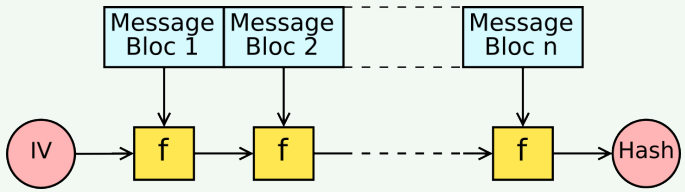
Fonctions de hachage

- Rôle : transformer un message de taille quelconque en un condensé de taille fixe
- Usage : signature ou MAC
- Propriétés principales souhaitées : résistance à la pré-image, à la seconde pré-image, aux collisions.
- Objectifs possibles, optimiser :
  - son temps d'exécution parallèle (nombre de processeurs fixé ou non)
  - le nombre de processeurs pour un temps d'exécution optimal
  - la consommation mémoire

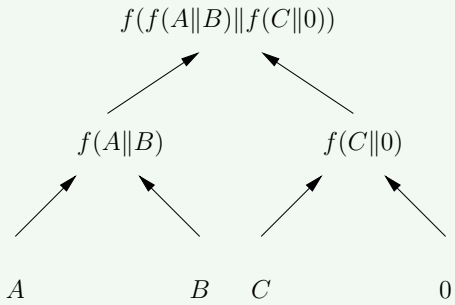
Fonctions de hachage séquentielles/parallèles

Extension du domaine d'une fonction de compression $f : \{0,1\}^{2N} \to \{0,1\}^{N}$.
Algorithmes historiques :

- construction séquentielle de Merkle-Damgård,



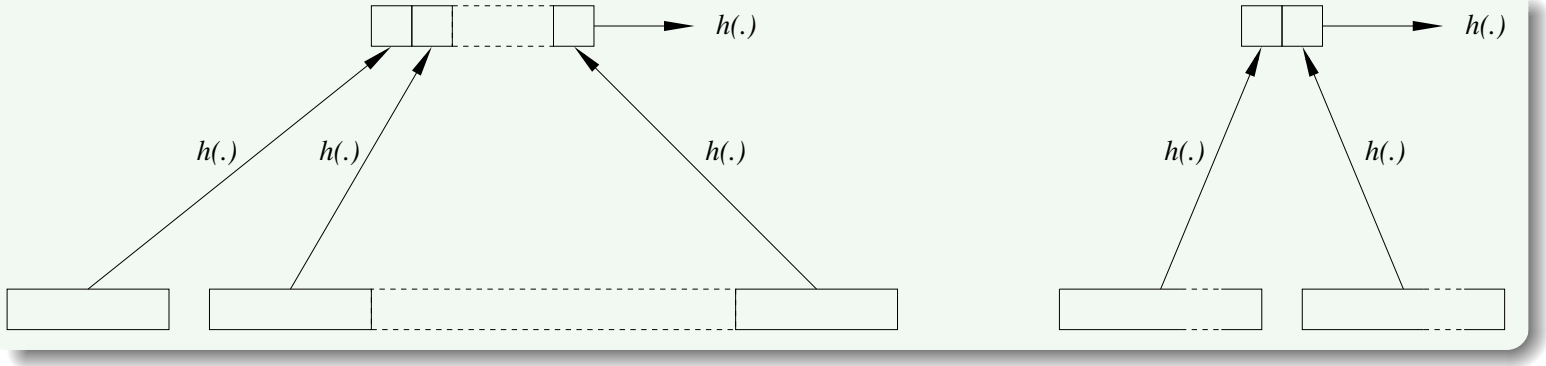
- construction parallèle de Merkle (et Damgård).



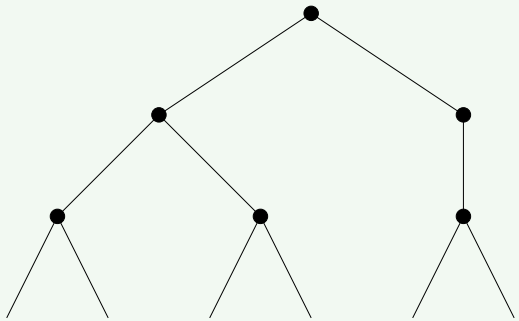
$$haché = f(f(A\|B)\|f(C\|0))\|3N.$$

Deux approches

- Arbre à deux niveaux, choix entre scalabilité (standard SP 800-185) et partie séquentielle réduite :



- **Nous abordons une approche différente.**



Quelle(s) construction(s) pour un temps d'exécution optimal en distance finie ? Quel nombre de processeurs ?

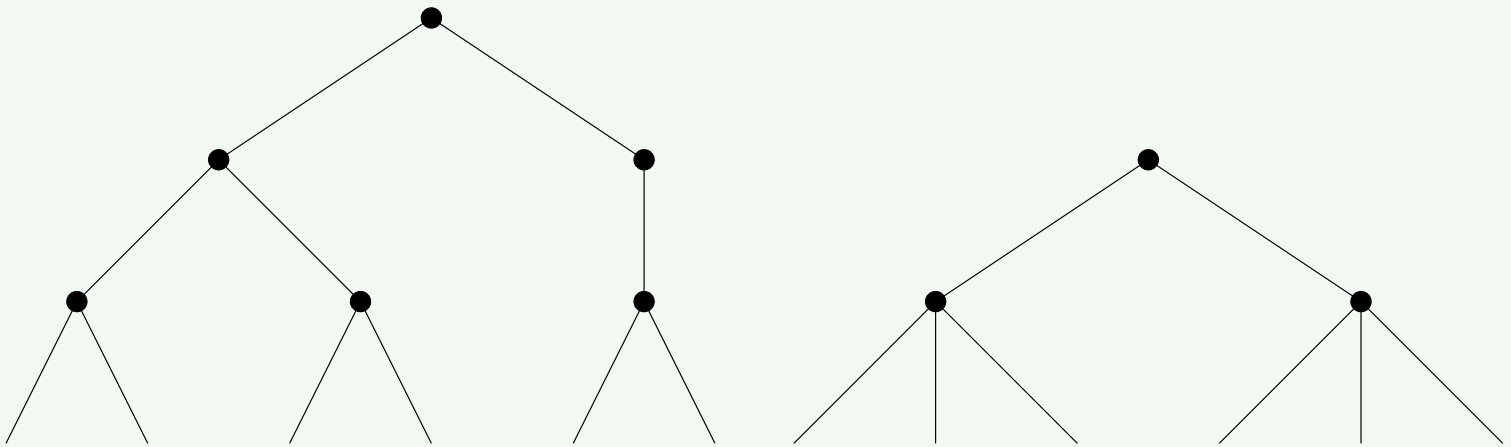
Sommaire

Optimisation des structures arborescentes

Hypothèses :

- Modèle PRAM avec stratégie EREW
- Plusieurs fonctions de hachage internes
- Le traitement d'**un bloc** du message par une fonction de hachage coûte **une unité de temps**
- Les nœuds d'un même niveau sont calculés par des processeurs distincts

Exemple d'optimisation pour un message de 6 blocs :



Optimisation de structures arborescentes

Problème

Soit l la taille du message. Trouver une hauteur d'arbre h et des arités entières x_1, x_2 , ..., x_h tels que $\sum_{i=1}^{h} x_i$ soit minimisé.

Cas des arbres ayant leurs feuilles à la même profondeur : une solution au problème doit nécessairement satisfaire les contraintes

$$\prod_{i=1}^{h} x_i \geq l \text{ et } \left(\prod_{i=1}^{h} x_i\right)/x_j < l \quad \forall j.$$

Remarque : pour x, y entiers strictement positifs

$$\left\lceil \frac{\lceil \frac{l}{x} \rceil}{y} \right\rceil = \left\lceil \frac{l}{xy} \right\rceil .$$

Optimisation en temps d'exécution

Que se passe t-il lorsque h et les x_i (pour $i = 1 \dots h$) sont des réels ?

- Minimum atteint lorsque les $x_i = x$;
- $x^h = l$, c-à-d. $x = l^{\frac{1}{h}}$;
- On doit donc déterminer h de sorte que $hl^{\frac{1}{h}}$ soit minimisé.

Le minimum est atteint pour $h = \ln(l)$, à savoir $x = e$.

Question du choix entre une arbre binaire parfait et un arbre ternaire parfait ?

Optimisation en temps d'exécution (suite)

Lemme

Un arbre dont le temps d'exécution est optimal ne peut être constitué que de niveaux d'arités ≤ 5.

Démonstration.

- Tout message de longueur i, $2 \leq i \leq 5$ peut être compressé en un temps optimal par un arbre de hauteur 1 et d'arité i.
- Tout nœud d'arité $a > 6$ peut être remplacé par un arbre d'arité 2, de sorte que $2\lceil \log_2 a \rceil < a$.
- Un nœud d'arité 6 peut être remplacé par un arbre d'arités $\{2, 3\}$.

Optimisation en temps d'exécution (suite)

Théorème

Un arbre dont le temps d'exécution est optimal peut être construit avec des niveaux d'arités 2 et 3.

Démonstration.

Remarque : un arbre optimal en temps d'exécution peut être construit avec au plus deux niveaux d'arités 2.

Optimisation en temps d'exécution (suite)

Théorème (réécrit)

Notons $x \in [\![0, 2]\!]$ la valeur qui minimise le produit $3^{i-x}2^x$ sous la contrainte $3^{i-x}2^x \geq l$. On note 3 plages d'optimalité :

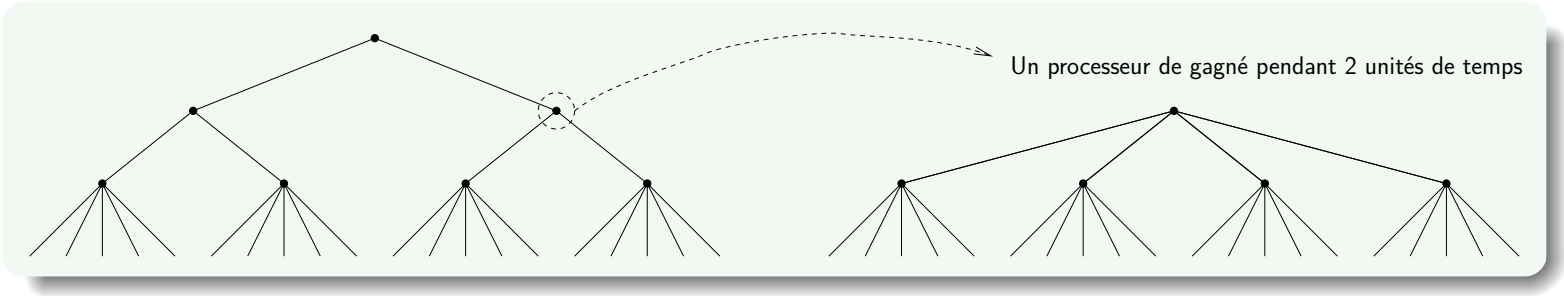
- *Si $l \leq 3^i < \frac{3l}{2}$ alors i niveaux d'arité 3.*
- *Si $\frac{3l}{2} \leq 3^i < \frac{9l}{4}$ alors $i-1$ niveaux d'arité 3 et un niveau d'arité 2.*
- *Si $\frac{9l}{4} \leq 3^i < 3l$, $i-2$ niveaux d'arité 3 et 2 niveaux d'arité 2.*

Le nombre de niveaux d'arité 3 est maximisé.

Gain de temps d'exécution par rapport à un arbre binaire ?

Optimisation du nombre de processeurs

- Le nombre de processeurs requis est égal à $\lceil l/3 \rceil$ dans le meilleur des cas, et égal à $\lceil l/2 \rceil$ lorsqu'il n'y a que des niveaux d'arité 2.
- Après optimisation, ce nombre pourrait être réduit à $\lceil l/5 \rceil$ (gain d'un facteur au mieux $\sim 5/2$).
- Il y a aussi un intérêt à augmenter l'arité de chaque niveau autant que possible.



Optimisation du nombre de processeurs (suite)

Lemme

Dans un arbre ayant un temps d'exécution optimal il y au plus 1 *niveau d'arité* 5 *et* 6 *niveaux d'arité* 4.

Démonstration.

- On peut remplacer avantageusement 2 niveaux d'arité 5 par 3 niveaux d'arité 3 :

$$3^3 > 5^2 \text{ et } 3 \cdot 3 \ < 2 \cdot 5.$$

- On cherche la première paire d'entiers (i, j) satisfaisant à la fois $3^i > 4^j$ et $3 \cdot i < 4 \cdot j$.
  On trouve $i = 9$ et $j = 7$.

Optimisation du nombre de processeurs (suite)

Théorème (première partie)

Quelque soit $l \geq 2$ il existe un unique multi-ensemble ordonné A ayant h_5 arités 5, h_4 arités 4, h_3 arités 3 et h_2 arités 2 tel que l'arbre correspondant couvre un message de taille l, ait un temps d'exécution minimisé et maximise :

1. *en premier lieu h_5 ;*
2. *puis h_4 ;*
3. *et enfin h_3.*

Optimisation du nombre de processeurs (suite)

Théorème (deuxième partie)

Si i est le plus petit entier tel que $l \leq 3^i < 3l$, ce multi-ensemble ordonné est tel que :

$$\begin{cases}
|A| = i, h_5 = 0, h_4 = 0, h_3 = i, h_2 = 0 & si \quad l \leq 3^i < \frac{9l}{8}, \\
|A| = i, h_5 = 0, h_4 = 1, h_3 = i-2, h_2 = 1 & si \quad \frac{9l}{8} \leq 3^i < \frac{81l}{64}, \\
|A| = i-1, h_5 = 0, h_4 = 3, h_3 = i-4, h_2 = 0 & si \quad \frac{81l}{64} \leq 3^i < \frac{27l}{20}, \\
|A| = i-1, h_5 = 1, h_4 = 1, h_3 = i-3, h_2 = 0 & si \quad \frac{27l}{20} \leq 3^i < \frac{3l}{2}, \\
|A| = i, h_5 = 0, h_4 = 0, h_3 = i-1, h_2 = 1 & si \quad \frac{3l}{2} \leq 3^i < \frac{27l}{16}, \\
|A| = i-1, h_5 = 0, h_4 = 2, h_3 = i-3, h_2 = 0 & si \quad \frac{27l}{16} \leq 3^i < \frac{9l}{5}, \\
|A| = i-1, h_5 = 1, h_4 = 0, h_3 = i-2, h_2 = 0 & si \quad \frac{9l}{5} \leq 3^i < \frac{81l}{40}, \\
|A| = i-1, h_5 = 1, h_4 = 1, h_3 = i-4, h_2 = 1 & si \quad \frac{81l}{40} \leq 3^i < \frac{9l}{4}, \\
|A| = i-1, h_5 = 0, h_4 = 1, h_3 = i-2, h_2 = 0 & si \quad \frac{9l}{4} \leq 3^i < \frac{81l}{32}, \\
|A| = i-1, h_5 = 0, h_4 = 2, h_3 = i-4, h_2 = 1 & si \quad \frac{81l}{32} \leq 3^i < \frac{27l}{10}, \\
|A| = i-1, h_5 = 1, h_4 = 0, h_3 = i-3, h_2 = 1 & si \quad \frac{27l}{10} \leq 3^i < 3l,
\end{cases}$$

où le nombre h_3 est au moins 1 dans le premier cas et peut être 0 dans les autres.

Optimisation du nombre de processeurs (suite)

Démonstration.

On note :

- a le nombre initial de niveaux d'arité 2,
- $i - a$ le nombre initial (maximisé) de niveaux d'arité 3.

Nous avons $h_5 \in [\![0, 1]\!]$, $h_4 \in [\![0, 6]\!]$ et $h_2 \in [\![0, 1]\!]$, soit au plus 28 cas ?

Transformons le produit initial $2^a 3^{i-a}$ en $2^w 3^{i-a-b} 4^v 5^u$ où :

- b est le nombre de niveaux d'arité 3 transformés,
- u, v, w les nombres de niveaux d'arité 5, 4 et 2 respectivement.

Optimisation du nombre de processeurs (suite)

Démonstration.

Pour chaque triplet $(h_5 = u, h_4 = v, h_2 = w)$ avec $u \in [\![0, 1]\!]$, $v \in [\![0, 6]\!]$ et $w \in [\![0, 1]\!]$, on cherche (a, b) avec $a \in [\![0, 2]\!]$ et b un entier positif tels que

$$3b + 2a = 5u + 4v + 2w.$$

Remarques sur une solution (a, b) :

- $3b + 2a$ se réécrit :
  - $3b$ si $a = 0$,
  - $3(b + 1) + 1$ si $a = 2$,
  - et $3b + 2$ si $a = 1$.
- Elle est unique.

Optimisation du nombre de processeurs (suite)

Démonstration.

Une telle solution doit satisfaire $3^{i-a-b}2^w4^v5^u \geq l$, à savoir $3^i \geq 3^{a+b}l/(2^w4^v5^u)$.

Nous avons

$$(3/2)^a l \leq 3^i < \min(3l, (3/2)^{a+1}l).$$

Donc, si $\frac{3^{a+b}l}{2^w4^v5^u} \geq \min\left(3l, \left(\frac{3}{2}\right)^{a+1} l\right)$, cette solution n'existe pas.

Sur les 28 cas, seuls 15 sont valides, qu'on peut noter comme des 5-uplets (u, v, w, a, b), et pour lesquels on calcule et trie les valeurs $3^{a+b}l/(2^w4^v5^u)$.

Optimisation du nombre de processeurs (suite)

Démonstration.

Nos 15 cas :

$$
\text{(I)}\begin{cases}
|A| = i, h_5 = 0, h_4 = 0, h_3 = i, h_2 = 0 & \text{si} \quad l \leq 3^i < \frac{3l}{2}, \\
|A| = i, h_5 = 0, h_4 = 1, h_3 = i-2, h_2 = 1 & \text{si} \quad \frac{9l}{8} \leq 3^i < \frac{3l}{2}, \\
|A| = i-1, h_5 = 0, h_4 = 3, h_3 = i-4, h_2 = 0 & \text{si} \quad \frac{81l}{64} \leq 3^i < \frac{3l}{2}, \\
|A| = i-1, h_5 = 1, h_4 = 1, h_3 = i-3, h_2 = 0 & \text{si} \quad \frac{27l}{20} \leq 3^i < \frac{3l}{2}, \\
|A| = i-1, h_5 = 0, h_4 = 4, h_3 = i-6, h_2 = 1 & \text{si} \quad \frac{729l}{512} \leq 3^i < \frac{3l}{2},
\end{cases}
$$

$$
\text{(II)}\begin{cases}
|A| = i, h_5 = 0, h_4 = 0, h_3 = i-1, h_2 = 1 & \text{si} \quad \frac{3l}{2} \leq 3^i < \frac{9l}{4}, \\
|A| = i-1, h_5 = 0, h_4 = 2, h_3 = i-3, h_2 = 0 & \text{si} \quad \frac{27l}{16} \leq 3^i < \frac{9l}{4}, \\
|A| = i-1, h_5 = 1, h_4 = 0, h_3 = i-2, h_2 = 0 & \text{si} \quad \frac{9l}{5} \leq 3^i < \frac{9l}{4}, \\
|A| = i-1, h_5 = 0, h_4 = 3, h_3 = i-5, h_2 = 1 & \text{si} \quad \frac{243l}{128} \leq 3^i < \frac{9l}{4}, \\
|A| = i-1, h_5 = 1, h_4 = 1, h_3 = i-4, h_2 = 1 & \text{si} \quad \frac{81l}{40} \leq 3^i < \frac{9l}{4}, \\
|A| = i-2, h_5 = 0, h_4 = 5, h_3 = i-7, h_2 = 0 & \text{si} \quad \frac{2187l}{1024} \leq 3^i < \frac{9l}{4},
\end{cases}
$$

$$
\text{(III)}\begin{cases}
|A| = i-1, h_5 = 0, h_4 = 1, h_3 = i-2, h_2 = 0 & \text{si} \quad \frac{9l}{4} \leq 3^i < 3l, \\
|A| = i-1, h_5 = 0, h_4 = 2, h_3 = i-4, h_2 = 1 & \text{si} \quad \frac{81l}{32} \leq 3^i < 3l, \\
|A| = i-1, h_5 = 1, h_4 = 0, h_3 = i-3, h_2 = 1 & \text{si} \quad \frac{27l}{10} \leq 3^i < 3l, \\
|A| = i-2, h_5 = 0, h_4 = 4, h_3 = i-6, h_2 = 0 & \text{si} \quad \frac{729l}{256} \leq 3^i < 3l,
\end{cases}
$$

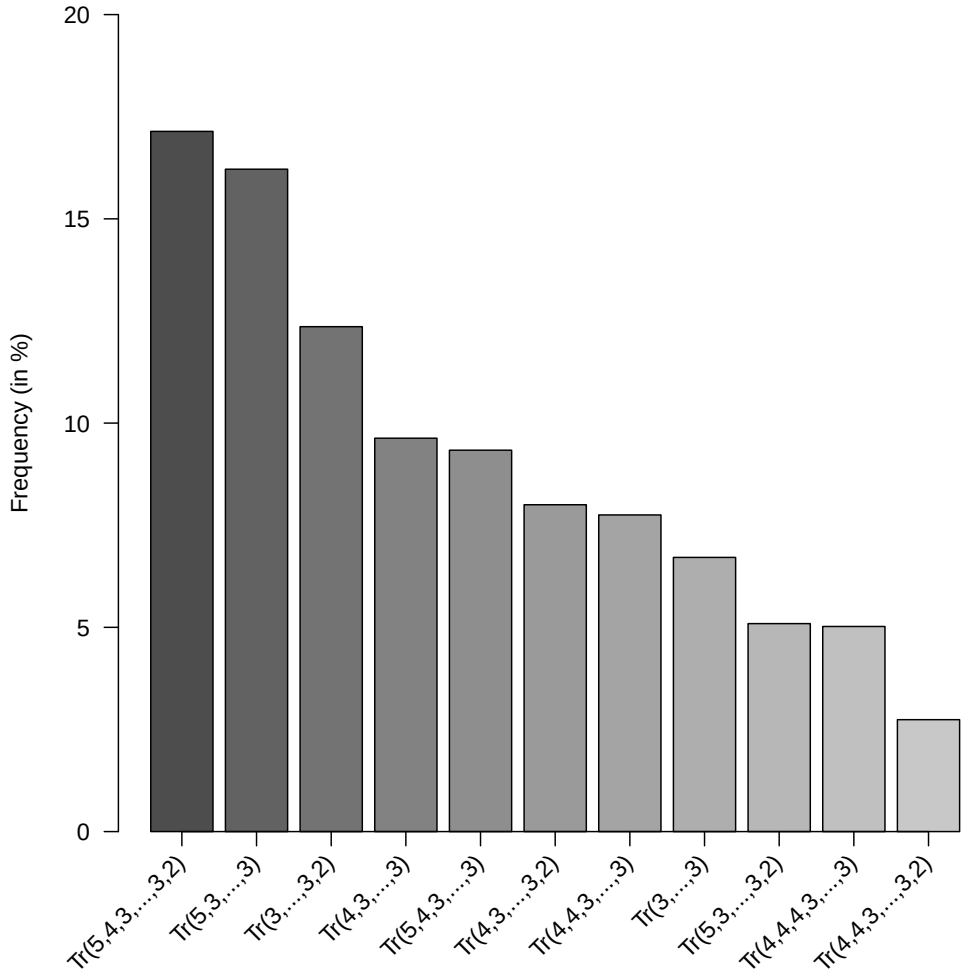
Distribution des 11 cas

Estimation exacte possible si les tailles sont uniformément distribuées.

Estimation empirique avec une simulation de Monte Carlo, pour une distribution asymétrique :

- Taille moyenne des messages transférés ≈ 10 KB
- Tailles distribuées selon la loi de Pareto
- Avec des paramètres de forme $\rho = 1,5$ et de position $\nu = 10^4 \cdot (\rho - 1)/\rho$
- Fréquences estimées avec R et la librairie VGAM

Distribution des 11 cas (suite)



Nombre de processeurs : $\lceil l/3 \rceil$ dans env. $19,1\%$, $\lceil l/4 \rceil$ dans env. $33,1\%$ et $\lceil l/5 \rceil$ dans env. $47,8\%$.

Merci pour votre attention...

Questions ?