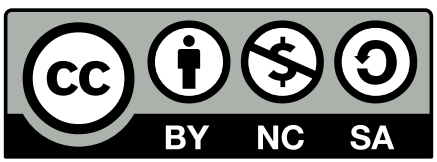
université de BORDEAUX

Implémentation de fonctions en éponge

Amélie Guémon
Ida Tucker
<amelie.guemon@etu.u-bordeaux.fr>

Master CSI, Université de Bordeaux, France

22 avril 2016



Plan

Définition : fonction de hashage

Une fonction de hashage est une application qui associe à un ensemble de départ infini $\{0,1\}^*$ un ensemble d'arrivée fini $\{0,1\}^n$ constitué de chaînes de bits de taille n.

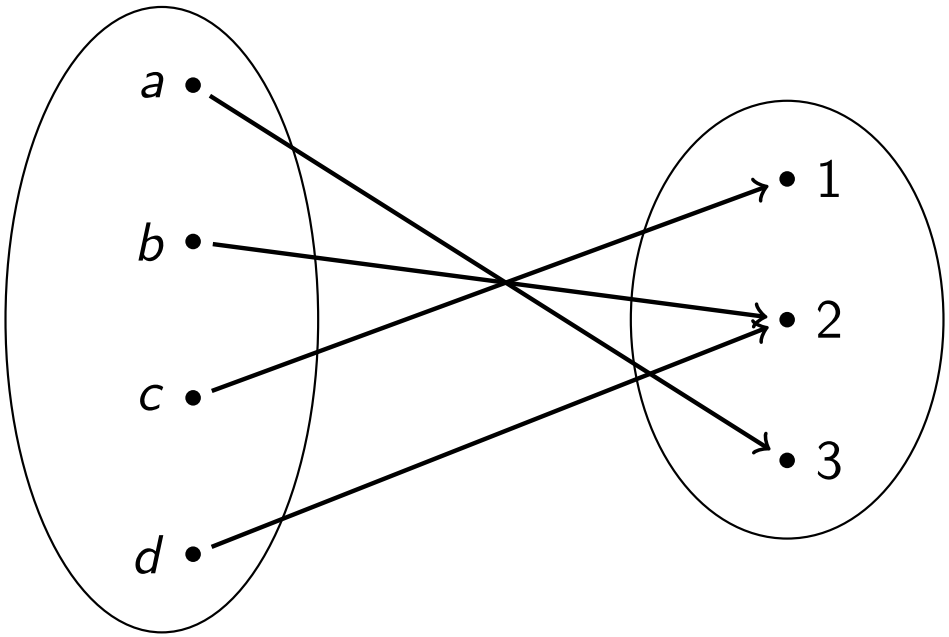


Figure: Collision dans une fonction de Hachage

Propriétés requises

université de BORDEAUX

- **Résistance à la Pré-image** : Pour un hash y donné, il est dur de trouver une pré-image $x \in f^{-1}(H)$ tel que $y = H(x)$.
- **Résistance à la Seconde Pré-image** : Pour un clair x, il est dur de trouver un autre clair x', $x' \neq x$ tel que $H(x) = H(x')$.
- **Résistance aux Collisions** : Il est dur de trouver 2 messages clairs x et x' avec $x \neq x'$ tel que $H(x) = H(x')$.

Résistance aux collisions $\Rightarrow$ Résistance à la seconde pré-image $\Rightarrow$ Résistance à la première pré-image

Padding

- **Padding Simple** : Représenté par $10*$, il faut rajouter un 1, puis un nombre fini de 0, de telle sorte que la longueur du resultat soit un multiple de le taille des blocks que l'on doit utiliser.

Figure: Simple padding.

Padding

- **Merkle-Damgård Padding** : Représenté par $10*1|M|$, il faut rajouter un 1, puis un nombre fini de 0, de telle sorte que la longueur du resultat soit congru à 448 mod 512. Ensuite, on y ajoute la longueur du message, sur 64 bits.

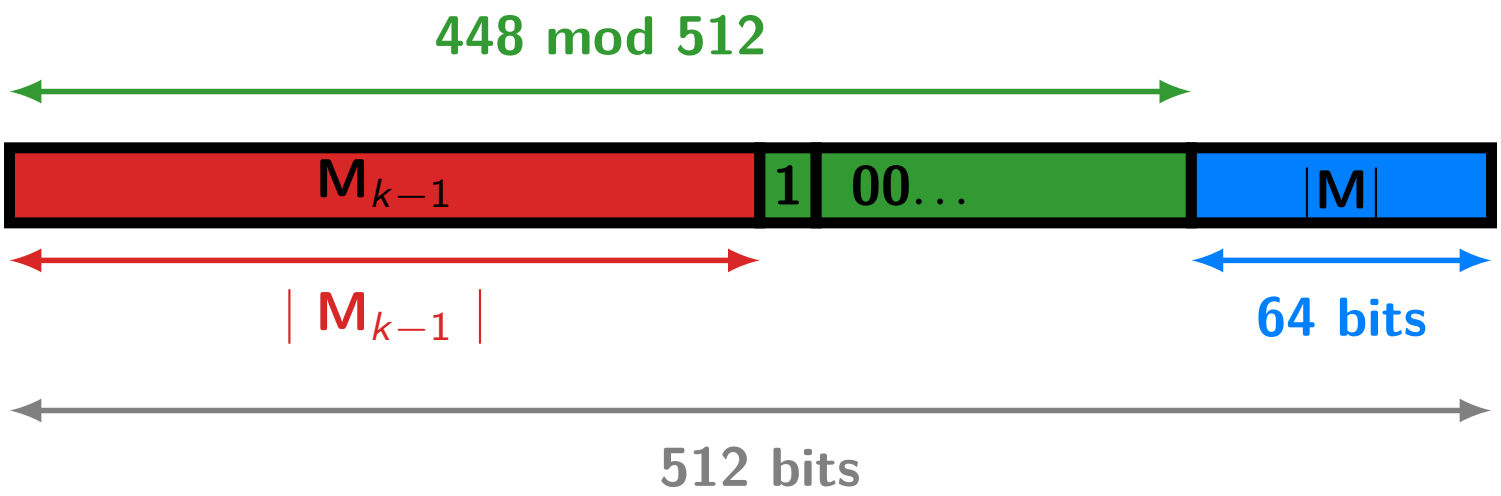


Figure: Merkle-Damgård padding.

Plan

Construction de Merkle-Damgård

La construction de Merkle-Damgård permet de définir des fonctions de hachage en itérant des fonctions de compression.

- Une fonction de compression part d'un ensemble fini vers un ensemble fini.
- Une fonction de hachage part d'un ensemble infini vers un ensemble fini.

Construction de Merkle-Damgård

université de BORDEAUX

- Théorème : Si la fonction de compression h utilisée par la fonction de hachage H l'est aussi.

Applications

- MD5
- SHA1

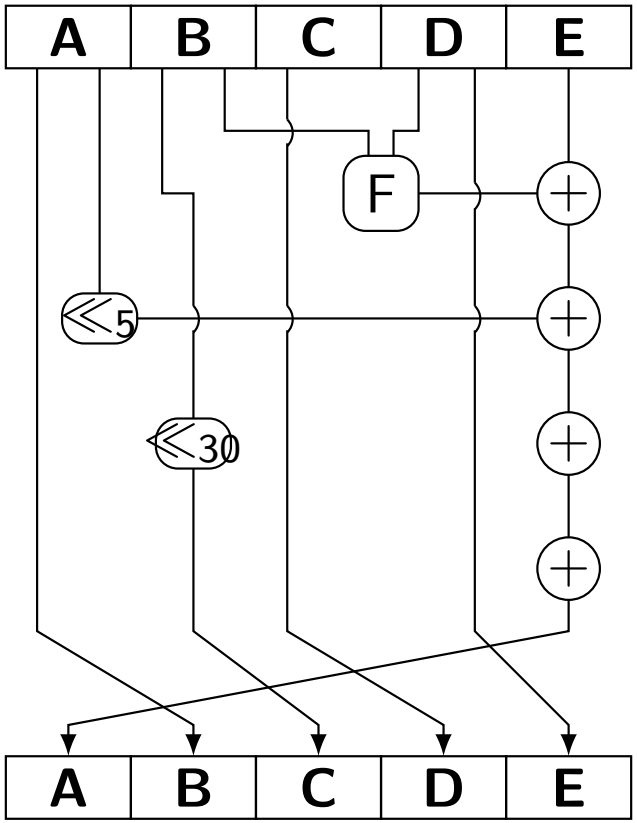


Figure: The i^{th} round in SHA-1 ($0 \leq i \leq 79$).

Plan

On continue ! ! !

Ce slide est quasiment vide !

Plan

Livres et références I


Chris Anley, John Heasman, Felix Linder, and Gerardo Richarte.
The Shellcoder's Handbook : Discovering and Exploiting Security Holes.
John Wiley & Sons, 2nd edition, 2007.


Jon Erickson.
Hacking : The Art of Exploitation.
No Starch Press, 2nd edition, 2007.


Greg Hoglund and Gary McGraw.
Exploiting Software : How to Break Code.
Software Security Serie. Addison Wesley, 2004.


Jamie Hoglund, Greg et Butler.
Rootkits : Subverting the Windows Kernel.
Software Security Serie. Addison Wesley, 2005.


Randall Hyde.
The Art of Assembly Language.
No Starch, 2003.

Livres et références II


Joseph Kong.
Designing BSD Rootkits : An Introduction to Kernel Hacking.
No Starch, 2007.


Robert Love.
Linux Kernel Development.
Sams, 2nd edition, 2005.


Robert Love.
Linux System Programming : Talking Directly to the Kernel and C Library.
O'Reilly Media, 2007.


Robert C. Seacord.
Secure Coding in C and C++.
SEI Series. Addison Wesley, 2005.


Peter Szor.
The Art of Computer Virus Research and Defense.
Addison Wesley, 2005.

Questions ?