

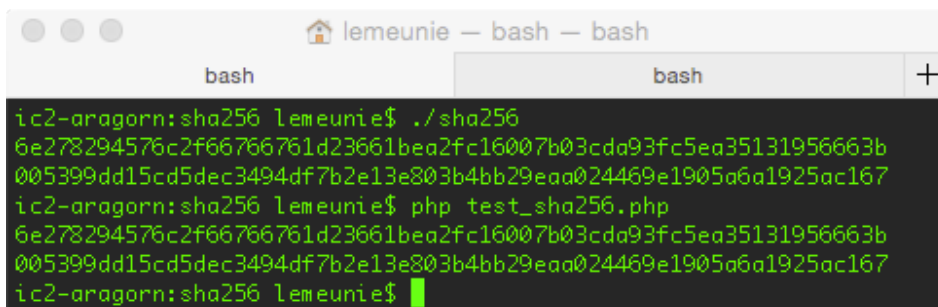
## TP2

### Calcul d'empreinte avec SHA512

Ce TP porte sur l'étude de l'algorithme de hachage SHA512 et son implémentation dans le langage C. Vous partirez de la documentation officielle ainsi que d'une implémentation de SHA256.

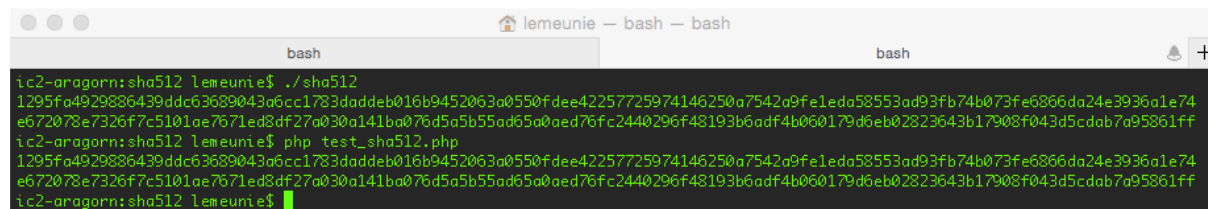
(a) Commencez par étudier le document intitulé *2012-FIPS-180-4.pdf* (sur UMTICE). Il s'agit du document officiel de la NIST (*National Institute of Standards and Technology*). Ce type de document décrit un standard cryptologique recommandé aux agences gouvernementales américaines. On peut aussi le trouver à l'adresse <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

(b) Récupérez sur UMTICE le fichier *sha256.zip*. Il contient une implémentation en langage C de l'algorithme SHA256. Etudiez le code source et compilez-le (par exemple avec le compilateur *gcc* dans un terminal : `gcc sha256.c sha256_example.c -o sha256`). Testez son bon fonctionnement comme cela est montré sur l'image suivante :



```
ic2-aragorn:sha256 lemeunie$ ./sha256
6e278294576c2f66766761d23661bea2fc16007b03cda93fc5ea35131956663b
005399dd15cd5dec3494df7b2e13e803b4bb29eaa024469e1905a6a1925ac167
ic2-aragorn:sha256 lemeunie$ php test_sha256.php
6e278294576c2f66766761d23661bea2fc16007b03cda93fc5ea35131956663b
005399dd15cd5dec3494df7b2e13e803b4bb29eaa024469e1905a6a1925ac167
ic2-aragorn:sha256 lemeunie$
```

(c) À partir de l'implémentation de SHA256 ainsi que de la documentation FIPS-180-4 implémentez l'algorithme de calcul d'empreinte SHA512. Vous créerez les fichiers suivants : *sha512.h* *sha512.c* *sha512\_example.c* et *test\_sha512.php*. Le résultat devrait être celui montré dans l'image suivante :



```
ic2-aragorn:sha512 lemeunie$ ./sha512
1295fa4929886439ddc63689043a6cc1783daddeb016b9452063a0550fdee42257725974146250a7542a9fe1eda58553ad93fb74b073fe6866da24e3936a1e74
e672078e7326f7c5101ae7671ed8df27a030a141ba076d5a5b55ad65a0aed76fc2440296f48193b6adf4b060179d6eb02823643b17908f043d5cdab7a95861ff
ic2-aragorn:sha512 lemeunie$ php test_sha512.php
1295fa4929886439ddc63689043a6cc1783daddeb016b9452063a0550fdee42257725974146250a7542a9fe1eda58553ad93fb74b073fe6866da24e3936a1e74
e672078e7326f7c5101ae7671ed8df27a030a141ba076d5a5b55ad65a0aed76fc2440296f48193b6adf4b060179d6eb02823643b17908f043d5cdab7a95861ff
ic2-aragorn:sha512 lemeunie$
```

#### Aides :

- Commencez par modifier les types de données dans le fichier *sha512.h* afin de prendre en compte les changements par rapport à SHA256 : travail sur des blocs de 1024 bits (128 byte) ; signature sur 512 bits ; longueur du message codé sur 128 bits.

- Vérifiez et modifiez les opérations binaires dans le fichier *sha512.c* comme cela est indiqué dans FIPS-140-2.
- Modifiez le tableau de constantes *k* dans le fichier *sha512.c* comme cela est indiqué dans FIPS-140-2.
- Vérifiez et modifiez la fonction *sha512\_compress*. Attention aux nombres de boucles effectués et des affectations de *m[i]* comme indiqué dans FIPS-140-2.
- Modifiez la fonction *sha512\_init* comme indiqué dans FIPS-140-2.
- Modifiez la fonction *sha512\_compute* en tenant compte du codage de la longueur du message (codé sur 128 bits) et de la taille du tableau *ctx->data*.
- Modifiez la fonction *sha512\_convert* en tenant compte de la taille du tableau *hash* et du tableau *ctx->state* et des types de valeurs stockées par ces deux tableaux.

## Travail à rendre

Dans un fichier nommé *Prénom\_Nom\_sha512.zip* (remplacer *Prénom\_Nom* par vos propres prénom et nom), compressez au format Zip les 4 fichiers de la question (c) et déposez-le sur UMTICE.