Cryptographie par la pratique avec Python et OpenSSL

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 décembre 2024

https://www.bortzmeyer.org/cryptographie-pratique.html

Auteur(s) : Rémi Boulle
ISBN n°978-2-10-087133-9
Éditeur : Dunod
Publié en 2024

———————-

Vous le savez tous et toutes, la cryptographie est une composante essentielle de la sécurité de l'Internet. Si tout le monde ne va pas devenir expert·e en cryptographie, en revanche, il est utile de connaitre les principes de base. Ce livre <https://www.dunod.com/sciences-techniques/cryptographie-par-pratique- destiné à des étudiant·es plutôt qu'au grand public, est une approche très concrète de la cryptographie, avec des exemples pratiques. J'ai beaucoup apprécié cette approche terre-à-terre.

Le livre <https://www.dunod.com/sciences-techniques/cryptographie-par-pratique-avec-python commence par rappeler l'enjeu de la cryptographie, indispensable pour protéger l'utilisateurice de l'Internet des nombreuses activités de surveillance qui s'exercent contre lui ou elle. Les révélations d'Edward Snowden, par exemple, ont montré que la surveillance massive par les États est une réalité. La cryptographie soulève de nombreuses questions politiques <https://www.bortzmeyer.org/concealing-for-free html>, ce livre les mentionne mais parle surtout de technique.

Vous aurez en effet ensuite une explication des principes de la cryptographie, puis des différents algorithmes. Il faudra accepter un peu de mathématique, mais c'est nécessaire dans ce domaine.

Pour les exemples de code, le livre utilise Python, et commence par une introduction (ou un rappel) de ce langage, si vous ne le connaissez pas bien. Il dépend ensuite de plusieurs bibliothèques qu'il présente, dont `cryptography` et, comme le titre l'indique, OpenSSL. La cryptographie nécessite des opérations de bas niveau, comme des décalages de bits mais ne vous inquiétez pas, tout cela est rappelé.

On commence par un exemple trivial et d'intérêt purement historique, l'algorithme de César :

```
import sys

KEY=3

def chiffre_Cesar_texte(texte_clair, k):
    texte_chiffre = ""
    for lettre in texte_clair:
        code_lettre = ord(lettre)-65
        chiffre = (code_lettre + k) % 26
        lettre_chiffre = chr(chiffre + 65)
        texte_chiffre = texte_chiffre + lettre_chiffre
    return texte_chiffre

for texte in sys.argv[1:]:
    print(chiffre_Cesar_texte(texte, KEY))
```

Mais vous aurez d'innombrables exemples de codes plus sérieux, utilisant des services réels, par exemple récupérer des données sur la chaine de blocs Bitcoin, l'analyser, la vérifier et même faire du minage de bitcoins. Même un·e connaisseur·se en cryptographie y découvrira certainement quelque chose, par exemple qu'il existe une courbe elliptique souveraine en France <https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000024668816>.

Conçu comme livre de classe, il contient de nombreux exercices. Si vous voulez récupérer l'intégralité des exemples et des solutions, allez sur le site officiel <https://dunod.com/EAN/9782100871339> (on vous demande votre adresse de courrier mais mettez n'importe quoi, ça marche).

Le livre ne se contente pas de généralités sur la cryptographie mais explique aussi les détails de certains protocoles de communication sécurisés, très utilisés sur l'Internet, comme HTTPS et SSH, ou bien dans le contexte de l'Internet des objets, comme LoRaWAN, avec décorticage de paquets. Bref, si vous êtes prêt à vous plonger dans la cryptographie, voici un excellent guide.

La photo du livre, ci-dessous, contient une autre image, cachée. Lisez le chapitre sur la stéganographie et vous pourrez récupérer cette image cachée (les quatre bits de poids faible contiennent les bits de poids fort de l'image cachée, cf. section 4.5.2, p. 43 et 44).