Zero-Knowledge Proof

Rézoléo

13-03-2025

@benoitlx

Si vous voyez des typos, n'hésitez pas à faire une PR sur le repo de la forma, c'est encore plus drole en live.

https://github.com/benoitlx/forma-zkp

Il y a un bonus si je merge un truc qui éteind mon ordi à la fin de la forma (oui c'est possible 🙃)

Intuition

Dragibus®

Trouvez un protocol me permettant de prouver que j'ai choisis des bonbons de différentes couleurs, sans que vous puissiez savoir lesquels.

Sudoku

```
1 seed(73)
2
3 from sudoku import *
4
5 print_board(board)
```

[finished]

5	2	9	1		4	6		8
8			2		9		1	
4	1	6		8	7	9		3
1	4	5		7	8		6	2
2	6		3	1	5	7		9
7		3	4			8		1
	7	2	8	9	1		3	

Sudoku

```python
seed(73)

from sudoku import *

# P ask for verification of the line 3
# V send a shuffled version of line 3 of his solution
line = shuffle(solution[2])

# P verify that the constraints are respected
print(line)
print(all(line.count(n) == 1 for n in range(1, 9)))
```

[finished]

```
[1, 3, 7, 6, 8, 5, 9, 4, 2]
True
```

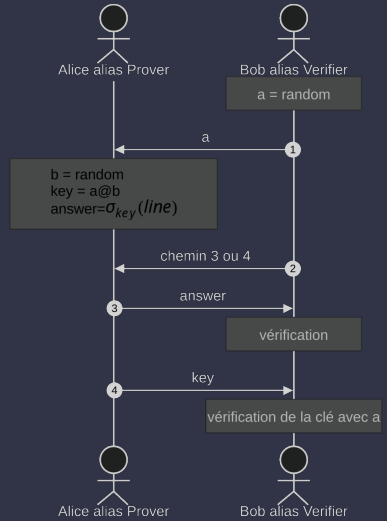
```
print(solution[2])
```

```
[finished]
```

```
[4, 1, 6, 5, 8, 7, 9, 2, 3]
```

xkcd: 74

Kururugi Sudoku



Formalisme

Algo quoi ??

Qu'est-ce qu'un Algorithme ?

> Algorithm does not have a generally accepted formal definition. Researchers[1] are actively working on this problem.

From (https://en.wikipedia.org/wiki/Algorithm_characterizations)

Pleins de modèle de calcul différents :

- automate fini
- automate à pile
- lambda-calcul
- machines à registres
- automates cellulaires
- **Machine de Turing**

Définition 1 - Machine de Turing

Une **machine de Turing** est un 7-uplet $(\Sigma, Q, \sigma, \delta, \Delta, q_0, F)$ où :

- Σ est un ensemble de symboles appelé alphabet, avec un symbole particulié noté $\#$.
- Q est un ensemble non vide fini d'états.
- $\sigma : Q \times \Sigma \longrightarrow \Sigma$ est une fonction d'**impression**.
- $\delta : Q \times \Sigma \longrightarrow Q$ est une fonction de **transition**.
- $\Delta : Q \times \Sigma \longrightarrow \{-1, 1\}$ est une fonction de **déplacement**.
- q_0 l'état initial de la machine.
- F l'ensemble des états finaux.

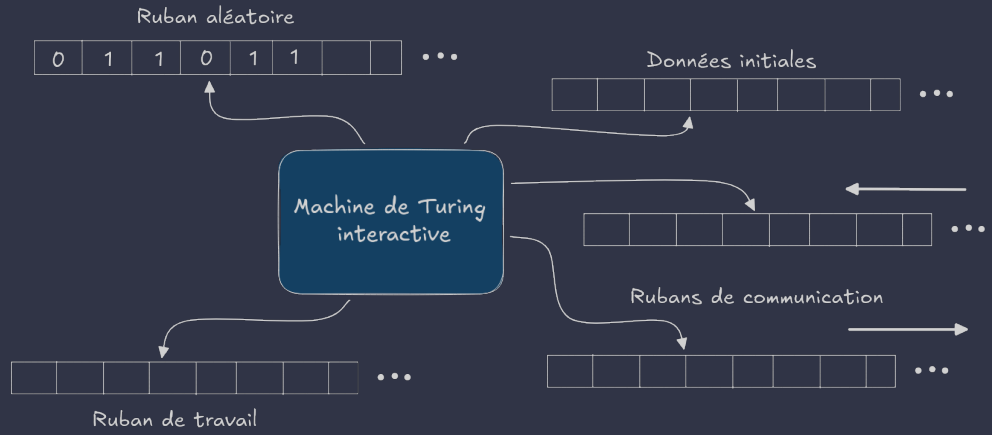
Fonctionnement (https://turingmachine.io/)

À chaque étape, la machine se trouve dans un état q et lit un symbole a, puis suit les instructions suivantes :

- écrit le symbole $\sigma(q, a)$ sur le ruban.
- déplace la tête de lecture en fonction de $\Delta(q, a)$.
- passe de l'état q à l'état $\delta(q, a)$.

$$\mathrm{BB}(5) = 47\ 176\ 870$$ 💔

Machine de Turing interactive



Qu'est-ce qu'une Preuve ?

- en Maths => preuve comme séquence statique de symboles
- en Science => accumulation statistique de preuves
- En droit pénal => L'accusation doit prouver son cas "au-delà de tout doute raisonnable"
- En info, c'est ...

Définition 2 - Système de Preuve interactif

Soit $\mathcal{L}$ un langage sur $\{0,1\}$ ($\mathcal{L} \subseteq \{0,1\}^*$).
On appel **système de preuve interactif pour $\mathcal{L}$**
toute paire de **machine de turing interactive** (P, V),
avec V qui termine en $\mathcal{O}(n^k)$ étapes, $k \in \mathbb{N}$,
(n étant la taille du nombre sur le ruban de données initiales), vérifiant :

- **Complétude**: $\forall (x,w) \in \mathcal{L},\ \ \mathcal{P}(P(x \bullet w) \rightleftarrows V(x) = 1) \geqslant 0.9$
- **Robustesse**: $\forall P^* \in \mathcal{M}_{\text{int}},\ \ \forall (x,w) \notin \mathcal{L},\ \ \mathcal{P}(P^*(x \bullet w) \rightleftarrows V(x) = 1) \leqslant 0.1$

xkcd: 1153

Définition 3 - Preuve à divulgation nulle de connaissance

Un **système de preuve interactif** (P, V) **sur** $\mathcal{L}$ est dit à **divulgation nulle de connaissance** si pour toute stratégie de vérification efficace V^*, il existe un algorithme probabiliste efficace S^*, tel que pour tout $(x, w) \in \mathcal{L}$ les variables aléatoires suivantes sont calculatoirement indiscernable :

- Le transcripte des interactions de P et V^* sur l'entrée x (P disposant de w)
- La sortie de S^* sur l'entrée x.

Résumé

- complétude
- robustesse
- divulgation nulle de connaissance

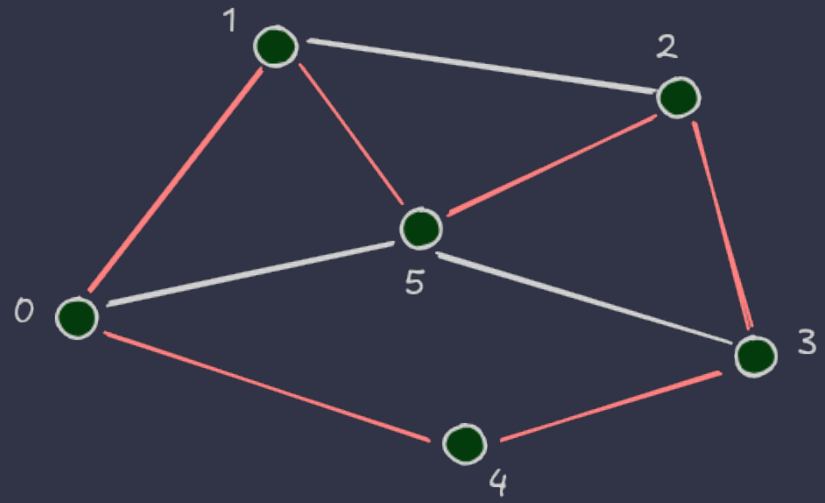
Exemples

Paramètre : n un entier
Donnée secrète : $S \in \mathbb{Z}_n{}^*$
Donnée initiale : $I = S^2 \bmod n$

Répéter ℓ fois :

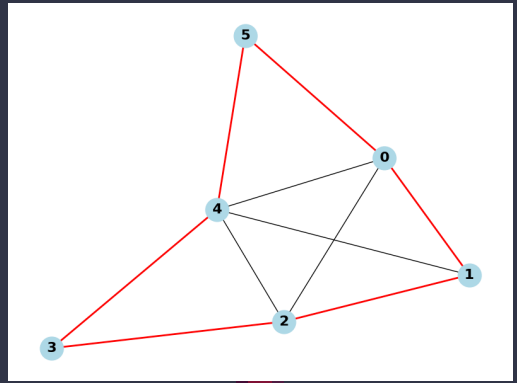
Prouveur		Vérifieur
$r \in_R \mathbb{Z}_n{}^*$		
$x = r^2 \bmod n$	$\xrightarrow{x}$	
	$\xleftarrow{e}$	$e \in_R \{0,1\}$
$y = r \times S^e \bmod n$	$\xrightarrow{y}$	vérifier $y^2 \stackrel{?}{=} x \times I^e \bmod n$

=> système d'authentification


1
2
0
5
3
4

```
/home/bleroux/Documents/forma-zkp/.venv/bin/python graph.py > /dev/null
```

[finished]



permutation

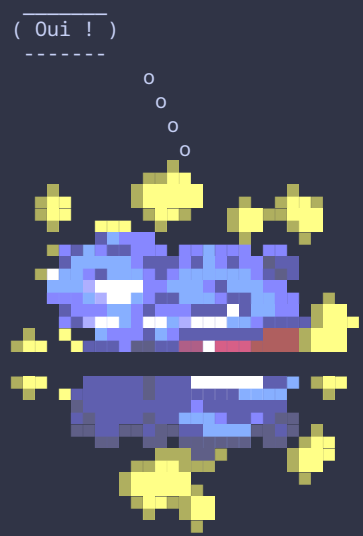
face cachée

Le vérifieur décide de vérifier l'isomorphisme de graphe

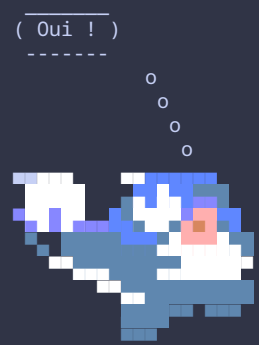
Le vérifieur décide de vérifier que P connait un cycle Hamiltonien pour H

Applications

Voltorb



Weezing



Poliwag

```
Already up to date.
#######    ,#####. .#####.    benoitlx ~ git version
2.48.1
  ###     ########.########   ---------------------------
  ###     ##################  Project: forma-zkp (1 branch)
  ###     `################'  HEAD: a47c929 (main, origin/main)
  ###      `##############'   Pending: 2+- 9+
  ###        `##########'     Created: 3 days ago
  ###          `######'       Languages:
#######          `#'                 • Markdown (68.3 %) • Python (31.7 %)
                               Authors: 91% benoitlx 10
                                         9% Benoit 1
####     ####     ###         Last change: a day ago
#####   #####     ###         URL: https://github.com/benoitlx/forma-zkp.git
######.######     ###         Commits: 11
### ##### ###     ###         Churn (1): zkp.md 1
###  ###  ###   #######       Lines of code: 483
###   #   ###    #####        Size: 1.02 MiB (12 files)
###       ###     ###         License: MIT
###       ###      #
```

Références

Vidéos

- **Wired** (https://www.youtube.com/watch?v=fOGdb1CTu5c&t=1145s)
- **Up and Atom** (https://www.youtube.com/watch?v=V5uVKZn3F_4)
- **Passe-Science** (https://www.youtube.com/watch?v=OSdcnoAmohs)

Papiers

Lien Randoms

- **Pages wikipédia (fr et en)** (https://en.wikipedia.org/wiki/Zero-knowledge_proof)
- **Cours ENS** (https://www.di.ens.fr/~granboul/enseignement/crypto/MPRI1-Crypto-ZeroKnowledge.pdf)
- **Cours ENS (bis)** (https://www.irif.fr/~carton/Enseignement/Complexite/ENS/Redaction/2009-2010/ludovic.patey.pdf)
- **TD ENS** (https://www.di.ens.fr/brice.minaud/cours/2018/TD4.pdf)
- **Cours du MIT** (https://courses.csail.mit.edu/6.857/2018/files/L22-ZK-Boaz.pdf)
- **StackExchange Crypto** (https://crypto.stackexchange.com)

Lien vers la présentation

- **Repo github** (https://github.com/benoitlx/forma-zkp)
- **Drive Rézo** (#todo)

Misc.

- presenterm (https://github.com/mfontanini/presenterm)
- typst (https://github.com/typst/typst)
- pokemonsay (https://github.com/possatti/pokemonsay)
- onefetch (https://github.com/o2sh/onefetch)
- mmdc (https://github.com/mermaid-js/mermaid-cli)