AlberPi

Escape Game – CÉPI – DICJ
Cégep de Jonquière

Résumé : AlberPi est un appareil intelligent qui accélère le déchiffrement des cryptogrammes générés par le chiffre d'Alberti.

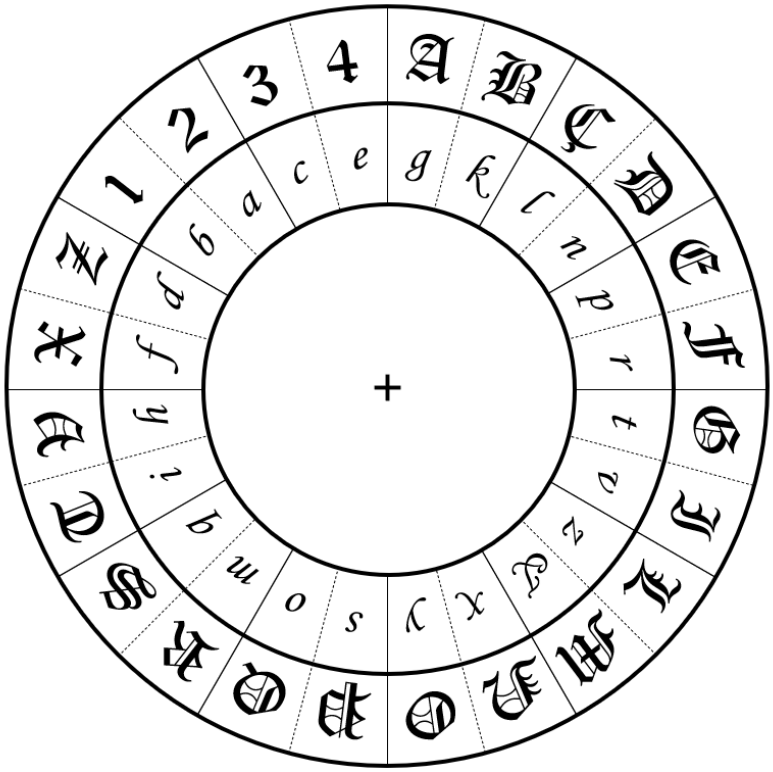


Table des matières

À quoi sert AlberPi?

AlberPi sert à accélérer le chiffrement et déchiffrement selon le chiffre d’Alberti.

Pour un cryptogramme donné, il le déchiffre en effectuant les substitutions selon la position du disque mobile.

Pour un message et une clé donnée, il bâtit un cryptogramme selon le chiffre d’Alberti.

Le cryptogramme peut être changé à l’aide d’un message MQTT.

Comment utiliser

Démarrage

Branchez l'appareil. Au démarrage, l'écran LCD affiche l'adresse IP de l'appareil et l'addresse IP, le port et le topic du serveur MQTT. Prenez-les en note.

Vérifiez que le disque est bien calibré : quand la LED est à son plus fort, centrer le g sur le A.

Une fois démarré, l'écran LCD affiche le cryptogramme sur la première ligne et le déchiffrement correspondant selon la position du disque. Losque le disque est tourné, la deuxième ligne de l'écran LCD s'ajuste en fonction de la position.

Si le cryptogramme fait plus de 16 caractères, il est trop long pour s'afficher complètement sur l'écran. Utilisez alors les deux boutons pour déplacer le cryptogramme et le déchiffrement correspondant vers la gauche ou la droite.

Changer le cryptogramme

Par défaut, le cryptogramme est « QbinxmFbxudssigyyutscooNcudty » qui est un chiffrement du message « LESCAROTTESSONTCVVITES » avec la clé k. Le cryptogramme peut être changé. Un cryptogramme peut aussi être généré à partir d'un message.

Chiffrer un message

Pour changer le message, envoyez un message au serveur MQTT sur le topic `alberti/msg`. Le premier caractère du message sera interprété comme la clé de chiffrement.

Sur un système linux, la commande `mosquitto_pub` (`sudo apt-get install mosquitto-clients`) permet d'envoyer un message sur un topic mqtt:

```
pi@raspberrypi:~$ mosquitto_pub -h 192.168.1.100 -t alberti/msg -m "kLes carottes sont cuites"
```

Où l'adresse IP doit être remplacée par celle du serveur MQTT. Le message sera encodé puis chiffré avec la clé k. Le nouveau cryptogramme apparaîtra sur l'écran LCD.

Changer le cryptogramme

Pour changer le cryptogramme, envoyez un message MQTT au serveur sur le topic `alberti/crypto`. Sur un système linux :

```
pi@raspberrypi:~$ mosquitto_pub -h 192.168.1.100 -t alberti/crypto -m
"QbinxmFbxudssigyyutscooNcudty"
```

Éteindre l’appareil

Pour éteindre l’appareil, appuyez sur le bouton seul soudé à même le circuit.
Attendez trente secondes puis débranchez l’appareil.

Utilisation sans boutons ni LCD

Brancher un écran et un clavier sur le RapsberryPi, ou se connecter par SSH.

Pour démarrer le programme

```
pi@raspberrypi:~$ python3 disqueAlberti.py
```

Ceci démarre le programme avec un cryptogramme par défaut.

Pour démarrer le programme avec un cryptogramme donné

```
pi@raspberrypi:~$ python3 disqueAlberti.py QbinxmFbxudssigyyutscooNc
```

Les caractères qui n’apparaissent pas sur le disque mobile sont ignorés

Pour chiffrer un message et obtenir un cryptogramme

```
pi@raspberrypi:~$ python3 chiffreAlberti.py "La carotte est cuite." x
```

Si la clé n’est pas donnée ou est invalide, la clé par défaut est k.

Le chiffre d'Alberti

Autour de 1420, l'architecte italien Léon Battista Alberti développe un chiffre par substitution polyalphabétique faisant usage d'un disque physique. Le célèbre chiffre de Vigenère est une variante du chiffre d'Alberti.

Dans sa version originale, le dispositif, appelé la **formula** est constitué de deux disques concentriques. Le disque externe est stationnaire (appelé **stabilis**) et contient des lettres majuscules et des chiffres. Le plus petit (appelé **mobilis**) est mobile et peut tourner autour de l'axe central. Il contient des lettres minuscules.

Chaque disque est divisé en 24 segments égaux.

Sur le disque externe, les lettres suivantes sont inscrites :
ABCDEFGILMNOPQRSTVXZ1234
Sur le disque interne, les lettres suivantes sont inscrites :
gklnprtuz&xysomqihfdbace

Avec ce chiffre, les messages secrets ne peuvent contenir que les lettres A, B, C, D, E, F, G, I, L, M, N, O, P, Q, R, S, T, V ou X. Les lettres H, J, K, U, W et Y sont absentes. On peut cependant les substituer par les règles suivantes :

H = FF J = II K = QQ U = VV W = XX Y = ZZ

Si un message doit contenir des nombres, on peut les épeler ou les écrire en chiffres romains. Les espaces et les caractères spéciaux sont effacés. La personne qui déchiffrera devra utiliser son gros bon sens pour comprendre le message.

Soit Alice et Bob deux personnes qui souhaitent partager un message secret. Alice et Bob ont chacun une formula d'Alberti. Pour chiffre un message, Alice utilise les lettres du *stabilis*. Elle place le *mobilis* dans une configuration aléatoire, puis substitue chaque lettre du message par son vis-à-vis sur le *mobilis*. Pour déchiffrer le cryptogramme, Bob doit placer son *mobilis* dans la même configuration et défaire les substitutions. Afin de limiter les attaques par fréquence, la position du mobilis est changée quelques fois en cours de chiffrement et de déchiffrement. Afin d'obtenir la même position de *mobilis*, Alice et Bob doivent partager une clé secrète.

La *formula* et le cryptogramme peuvent être publics. La clé secrète doit être conservée secrètement par Alice et Bob.

Voyons en détail, en suivant un exemple, les étapes du chiffre d'Alberti.

Précondition

Alice et Bob ont chacun un disque d'Alberti. Alice et Bob partagent une clé secrète. Dans notre exemple, la clé est la lettre k.

Chiffrement d'un message

Alice souhaite envoyer à Bob le message secret suivant : "les carottes sont cuites".

Étape A. Assainissement du message secret

Elle transforme son message de manière à pouvoir le chiffrer. Elle retire tous les symboles qui n'apparaissent pas sur le *stabilis*. Elle substitue la lettre U par VV.

Le message après encodage et avant chiffrement est : LESCAROTTESSONTCVVITES

Étape B. Ajout de bruit dans le message secret

Alice ajoute quelques chiffres entre 1 et 4 au hasard dans le message. Ceci servira à limiter la force des attaques par analyse de fréquence.

Le message avec bruit est : L3ESCARO3TTESSONTCVVITES4

Étape C. Chiffrement du message secret

Alice sépare le message secret en plusieurs portions de longueur aléatoire : L3ESCARO3TTE, SSON, T, CVV, ITES4. Alice chiffre chaque portion :

Alice sélectionne la première portion L3ESCARO3TTE. Elle pige une lettre du *stabilis* au hasard. Disons T. Nous appelons cette lettre *l'indice*. Elle tourne le *mobilis* afin que la clé k soit vis-à-vis l'indice T. Elle note la lettre du *mobilis* qui est vis-à-vis les lettres de la portion du message sur le *stabilis* :

ABCDEFGILMNOPQRS**T**VXZ1234
&xysomqihfdbaceg**k**lnprtuz
L3ESCARO3TTE ␣ huogy&ebukko

Alice effectue les mêmes étapes avec les portions SSON, T, CVV, ITES4, pour lesquelles elle pige les indices N, M, D et E, respectivement :

ABCDEFGILM**N**OPQRSTVXZ1234
qihfdbaceg**k**lnprtuz&xysom
SSON ␣ ttlk

ABCDEFGIL**M**NOPQRSTVXZ1234
ihfdbaceg**k**lnprtuz&xysomq
T ␣ z

ABC**D**EFGILMNOPQRSTVXZ1234
ceg**k**lnprtuz&xysomqihfdba

CVV ␣ gqq

ABCD**E**FGILMNOPQRSTVXZ1234
aceg**k**lnprtuz&xysomqihfdb
TES4 ␣ oksb

Finalement, elle concatène toutes les portions chiffrées, chacune précédée de son indice : **T**huogy&ebukko**N**ttlk**M**z**D**gqq**E**poksb

Étape D. Publication

Alice envoie le crypogramme Thuogy&ebukkoNttlkMzDgqqEpoksb à Bob.

Déchiffrement d’un message

Étape E. Déchiffrement

Bob reçoit le cryptogramme Thuogy&ebukkoNttlkMzDgqqEpoksb.

Il sait que la clé secrète est k. Quand il voit une lettre majuscule, il tourne son dispositif afin que celle-ci soit vis-à-vis la clé k et substitue chaque lettre du cryptogramme par la lettre correspondante du stabilis.

Étape F. Interprétation

Bob retire les chiffres du message. Il devine le message clair grâce aux règles de substitution et à sa connaissance du français.

Cryptanalyse

Alberti a complexifié son chiffre en donnant aux nombres de 11 à 4444 une signification dans un livre de code. Par exemple, le nombre 45 pouvait signifier “le pape”. Le nombre 3223 pouvait signifier : “À la lecture d’un chiffre *n* entre 1 et 4, tourner le disque de n positions dans le sens anti-horaire”. Dans cette version avancée, Alice et Bob doivent avoir chacun une copie identique du livre de code et le conserver secrètement.

Le chiffre d’Alberti était très sûr en son temps. Comme la position du disque change en cours de message selon des règles complexes, il est robuste contre les attaques par analyse de fréquences. En plus, le chiffrement et le déchiffrement sont plutôt rapides à exécuter manuellement.

Le disque d’Alberti n’est évidemment pas sûr en pratique. Une attaque par force brute est faisable. Pour deviner le message, un adversaire peut essayer chacune des 24 clés et reconstruire le message.

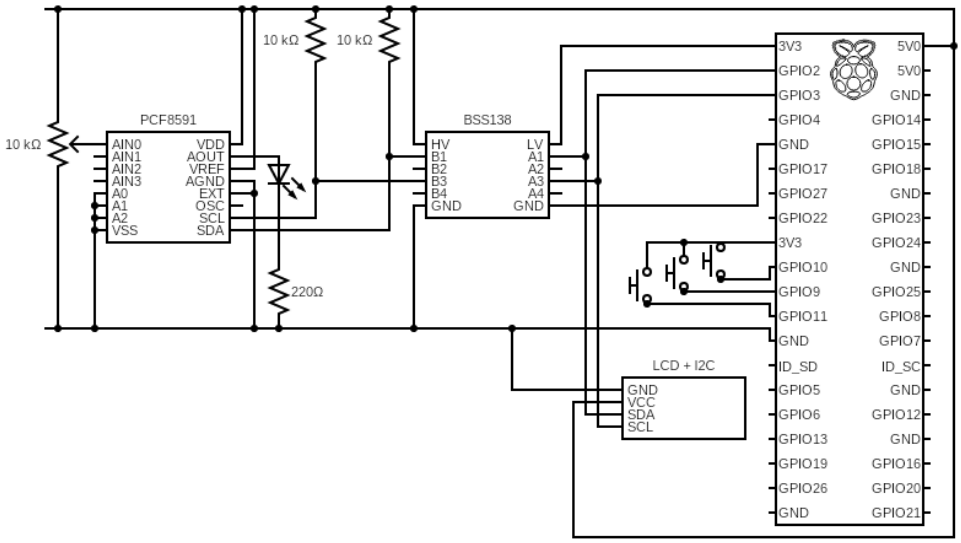
Comment bâtir AlberPi

Si AlberPi est perdu, ou brisé, il peut être reconstruit selon les indications suivantes.

Materiel

- Raspberry Pi (avec GPIO) et son alimentation.
- Potentiomètre 360° continu (5V)
  - https://www.digikey.ca/en/products/detail/tt-electronics-bi/6127V1A360L-5FS/2620662
- Convertisseur analogue à digital PCF8591T
  - https://www.nxp.com/docs/en/data-sheet/PCF8591.pdf
- Convertisseur de niveau logique bidirectionnel compatible avec i2C. (BSS138)
  - https://www.adafruit.com/product/757
- 1 résistor 220Ω
- 2 résistors 10kΩ
- Une DEL rouge
- Perfboard et matériel de soudure
- *Formula* d'Alberti cartonné
- Boîte en carton
- (Facultatif) 3 boutons
- (Facultatif) Afficheur LCD 16x2 avec module d'adaptation I2C
  - https://www.amazon.ca/SunFounder-Serial-Module-Display-Arduino/dp/B019K5X53O

Schéma du circuit



Installation et configuration du RaspberryPi

Sur un RaspberryPi avec GPIO et Wifi

- Mettre à jour le PI
  - `sudo apt-get update`
  - `sudo apt-get upgrade`
- Activer l'interface I2C
  - `sudo apt-get install -y python-smbus`
  - `sudo apt-get install -y i2c-tools`
  - `sudo raspi-config`
    - Dans Interfacing options, activer I2C (et SSH)
- Clôner le code source (https://github.com/reblapointe/Alberti) dans /home/pi/alberti
  - `cd /home/pi`
  - `git clone https://github.com/reblapointe/Alberti`
- Installer les paquets python nécessaires
  - `sudo pip3 install -r requirements.txt`
- Placer le script de démarage albertiLauncher.sh dans /etc/init.d et le rendre exécutable au démarrage
  - `sudo chmod a+x /home/pi/Alberti/disqueAlberti.py`
  - `sudo mv /home/pi/Alberti/albertiLauncher.sh /etc/init.d/`
  - `sudo chmod a+x /etc/init.d/albertiLauncher.sh`
  - `sudo update-rc.d albertiLauncher.sh defaults`
- Noter les adresses de l'afficheur LCD et du potentiomètre
  - `sudo i2cdetect -y 1`
- Définir les adresses I2C et GPIO pour correspondre aux connexions de l'appareil :
  - Dans deviceAddresses.py définir les valeurs pour
    - L'écran LCD (LCD_ADDRESS)
    - Le potentiomètre (POT_ADDRESS)
    - Le boutons gauche (LEFT_BUTTON_GPIO)
    - Le bouton droite (RIGHT_BUTTON_GPIO)
    - Le bouton de fermeture (SHUTDOWN_BUTTON_GPIO)
- Démarrer un service MQTT
  - Le service MQTT peut rouler sur le Pi en question, sur un autre ordinateur ou un service en ligne. Pour démarrer un serveur Mosquitto sur le Pi:
    - `sudo apt install -y mosquitto mosquitto-clients`
    - `sudo systemctl enable mosquitto.service`
  - Dans deviceAddresses.py définir les valeurs pour
    - L'adresse IP ou le domaine du serveur MQTT (MQTT_BROKER)
    - Le port (MQTT_PORT)
- Redémarrer
  - `sudo reboot`

Sources

Code source Python https://github.com/reblapointe/Alberti

Circuit basé sur https://embeddedcircuits.com/raspberry-pi/tutorial/raspberry-pi-potentiometer-tutorial

https://fr.wikipedia.org/wiki/Chiffre_d%27Alberti

Installation de Mosquitto (MQTT) https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/

Activation de I2C https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c