In the Lecture we learned how the Tallinn and Tartu Bus cards were used to authenticate users on public transport. For this homework, you will make an access system using the Bus cards.

System Parameters

- The system should accept MIFARE Classic cards
  - Including: Tallinn Bus cards and ISIC cards
- The system should accept Ultralight C cards
  - Including: Tartu Bus cards

How a user is added

- When a user wishes to use your service they go to your website and enter the card number written on the back of their card.
  - For Tallinn Bus cards the format should be: 9XXXXXXXXXX (e.g. 90007599719)
    - the numbers after the 30864900 prefix in sector 2
  - For Tartu Bus cards the format should be: 995XXXXXXXX (e.g. 99500382283)
    - the numbers after the 30864900 prefix in pages 14 to 19
  - For ISIC cards (we will assume that the user has scanned their card with an NFC reader) the format should be: 8XXXXXXXXXXX (e.g. 80140500507)
    - the numbers after the 92337331, 92337316 or 92337365 prefix in sector 2
  - Any other MIFARE Classic card e.g. Coop card or Ultralight C card e.g. RIMI card cannot be used in your system
- When a user is accepted the card number (and only the card number) is stored in a text file which represents our database (*registered_users.txt*)
- **TLDR**; add a user to your system by writing their card number to the *registered_users.txt* file (i.e. an actual registration website or real database is not required)

Providing service to your customers:

When a card is placed on your reader, your script should:

- Check if it is a MIFARE Classic or Ultralight C card **[0.5 pts]**
  - Any other card, the user should see the message: “[!] Card type not supported”
- Determine if the card is a bus card: **[1 pt]**
  - This can be done by checking for “pilet.ee:ekaart” in the card memory
  - If this is not found, the user should see the message: “[!] Invalid card format”
- Compare the UID of the card with the UID stored in the external NDEF record **[0.5 pts]**
  - The UID of the card should be obtained using the Get UID APDU
  - If the UID comparison fails, the user should see the message: “[!] Invalid card”
- Extract the data from the external NDEF record **[1 pt]**
- Extract the signature from the signature NDEF record **[1 pt]**
- Verify that the data has not been modified **[1 pt]**
  - For MIFARE Classic (i.e. Tallinn bus card and ISIC cards)
    - Create a hash of the data
    - Verify hash using openssl
  - For Ultralight C (i.e. Tartu bus card)
    - Verify the data using openssl
  - If verification fails
    - The user should see the message: “[!] Invalid card”
- If verification is successful **[1 pt]**
  - Extract the card number
  - Check if the card number is in your database
    - Open the text file of registered users and check if the card number is there
  - If the card number is there
    - The user should see the message: “[+] Welcome, Player 1” (or something like this)

        - Otherwise: “[!] Card not registered”

Bonus (2 pts):

The user should be able to add their ISIC card number as seen on the front of the card.

- Check if card is an ISIC card
    - e.g. The card number should start with 8
- Verify the ISIC card, the same as above but instead of using the card number 8XXX… the card number would be the value stored in the first block of Sector 7. This value should be in your *registered_users.txt* file.
- You can assume the format for the card number in Sector 7 will follow the scheme used by the School/Swedbank ISIC i.e. 14 characters then 2 blank spaces/2 null characters

Files on Moodle:

- A text file (r*egistered_users.txt*) that contains a list of ISIC, Tallinn and Tartu Bus card numbers from the lecture. This is a sample file and your own card number can be added.
- *copy_sample_Tartu.py* which can be used to write the contents of the Tartu Bus card seen in the lecture to the blue fob (please note the UID of the blue fob will not change therefore only page 4 onwards is written)
- *copy_sample_Tallinn.py* which can be used to write the contents of the Tallinn Bus card seen in the lecture to the the smiley face fob or white NFC card (only sectors 0 to 6 will be written) but block 0 will not change
- *copy_sample_SEB_ISIC.py* which can be used to write the contents of the SEB ISIC card seen in the lecture to the smiley face fob or white NFC card (sectors 0 - 6)
- *copy_sample_ISIC7.py* which can be used to write the contents of the School ISIC card used in the lecture to the smiley face fob or white NFC card (sectors 0 - 6)
- *copy_sample_Swedbank_ISIC.py* which can be used to write the contents of the Swedbank card seen in the lecture to the smiley face fob or white NFC card (sectors 0 - 6)
- *access_system_temp.py* a template file to aid you but it can be completely ignored

Testing cards:

- Your own Tallinn Bus card. Can be purchased from: https://tallinn.pilet.ee/pages/retailers
- Your own Tartu Bus card. Can be purchased from: https://tartu.pilet.ee/pages/retailers
- If you do not have a Tartu Bus card: use the *copy_sample_Tartu.py* file to copy the card used in Lecture 4 to your Ultralight C fob (blue fob)
- If you do not have a Tallinn Bus card: use the *copy_sample_Tallinn.py* file to copy the Tallinn bus card from Lecture 4 to your Classic fob (smiley face fob/white NFC card)
- If you do not have an ISIC card: use the *copy_sample_SEB_ISIC.py, copy_sample_ISIC7.py* or *copy_sample_Swedbank_ISIC.py* file to copy the ISIC cards from the Lecture to your Classic fob/card (sectors 0 – 6, you will have to add the ISIC specific sectors yourself :))

Assumptions

- The Database text file (r*egistered_users.txt*) will be in the same location as your .py file
- The Tartu public key certificate (*ecdsa-live-pub.pem*) will be in the same location as your .py file

Due Date: Check Moodle for date
Late Submission Deadline: Check Moodle for date (usually 1 week after due date)
Submission Files: .py file
Location: Moodle Homework 4 link

Additional Notes

- Most of what you need is in the Python lab files on Moodle