

|  |  |  |
| --- | --- | --- |
| **Mati Halfa** | **Shoham Gamliel** | **Arkady Koretsky** |
| **204190516** | **316311547** | **311008361** |

**Cryptography Final Project**

|  |  |
| --- | --- |
| **Subject:** | Storing of the encrypted database. Providing a secure access to the data on two levels: read and edit.  Encryption-Decryption provided by CAST-128. |

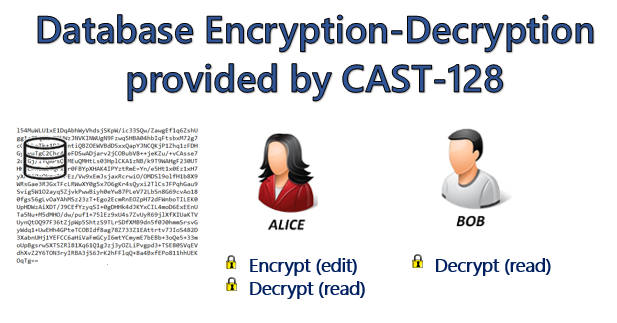
**Our Process:**

**1.** We implemented our project in C#.

**2.** We created separated classes for the encryption-decryption and key generator. Also created GUI classes, each class for each form.

**3.** We have 2 data bases, one for users and one for student’s data.  
users.csv and students.csv.

**4.** The users.csv contains 2 users: Alice and Bob.  
Each user has username and hashed password and its own capabilities.   
Alice is an administrator, she can edit, add, or delete students from the data base. So, her user can Encrypt and Decrypt.  
Bob is a regular user and he is read-only user. He can only decrypt.



**5.** The students.csv file contains the database itself. The file is encrypted and decrypted only at our application.

**6.** Each time the user opens the application, the students.csv file goes through decryption process. The decrypted data presented at the GUI of our application

**7.** If someone will try to reach the file outside our application, he will not be able to read any values without the key.

**8.** If you want to run the application on your own computer, make sure you put the encrypted files (no need to change what we give you) in these folders your\_path\Cast\_128\Cast128\_CS\bin\Debug.

**9.** We implemented by the official article:  
<https://tools.ietf.org/html/rfc2144#section-2.2>

Provided in 1997 by Carlisle Adams – one of the inventors of this algorithm.

**10.** Our implementation is ECB - each block is encrypted separately. Therefore, the first step in the encryption process was to divide the data to 64-bit blocks.   
In according to ECB mode one mistake in the encryption of a block will not affect the other blocks.

**11.** We created an instruct to organize the blocks to 2 part of 32-bits. So first part will be the L0 and second part will be R0.

**12.** You do not need Visual Studio to run the application. You can click on Cast\_128.exe file located at your\_path\Cast\_128\Cast128\_CS\bin\Debug.

**13.** You can clone the project from this repository:  
<https://github.com/ArkadyKoretsky/Cast-128-Cipher>