שער הפרויקט

En-Crypts

מבצע: עידו ויסברט

מנחות: ענת בן-משה ומורן טובול

בית ספר: מקיף ג' אשדוד ע"ש רוגוזין

שנת ההגשה: 2019

תוכן עניינים

הקדמה

הפרויקט "En-Crypts" כולל בעיקרו ספריית הצפנות, אך נכללת גם אפליקציה שמשתמשת בהצפנות האלה.

הפרויקט נכתב בשפת #C, ובמערכת Visual Studio 2017, וכולל ברובו שימוש בקובץ "dll." שהייתי צריך ליצור בעצמי.

השם נבחר כי "Encrypt" משמעותו הצפנה, ו-"Crypt" שאחת משמעויות המלה היא " An underground vault or chamber", מה שמתבטא בפרויקט מבחינה עיצובית, את החלונות השונים בפרויקט עיצבתי בסגנון ספרייה תת-קרקעית.

בתוך ספריית הצפנים, מימשתי הצפנה ופיענוח של הצפנים: צופן קיסר, צופן "Vigenere", צופן "RSA" (Rivest–Shamir–Adleman) וצופן "AES" (Advanced Encryption Standard).

הספרייה והאפליקציה מיועדות למי שרוצה ללמוד על איך שהצפנים עובדים, למי שרוצה להצפין מחרוזות בשביל הכיף ואף למורים שרוצים להדגים איך הצפנים ולהראות איך קוד המקור אמור להיראות במימוש הצפנים האלו.

מדריך למשתמש - מצפין

עבור המשתמש המצפין, בניתי ממשק גרפי שהמשתמש יכול להשתמש בו בקלות כדי להצפין טקסטים פשוטים מבלי להסתבך בשורות הקוד של הספרייה.

התקנה

כדי להתקין את התוכנה, צירפתי בר-קוד שמוביל את המשתמש לתיקיית drive (שהיא תיקיית קבצים שאליה נתתי גישה לכל אדם שמגיע אל התיקייה).

הכנס ברקוד

כאשר המשתמש מגיע לתיקיית הדרייב, תצטרך ללחוץ על כפתור ההורדה שנמצא כאן:

הכנס תמונה

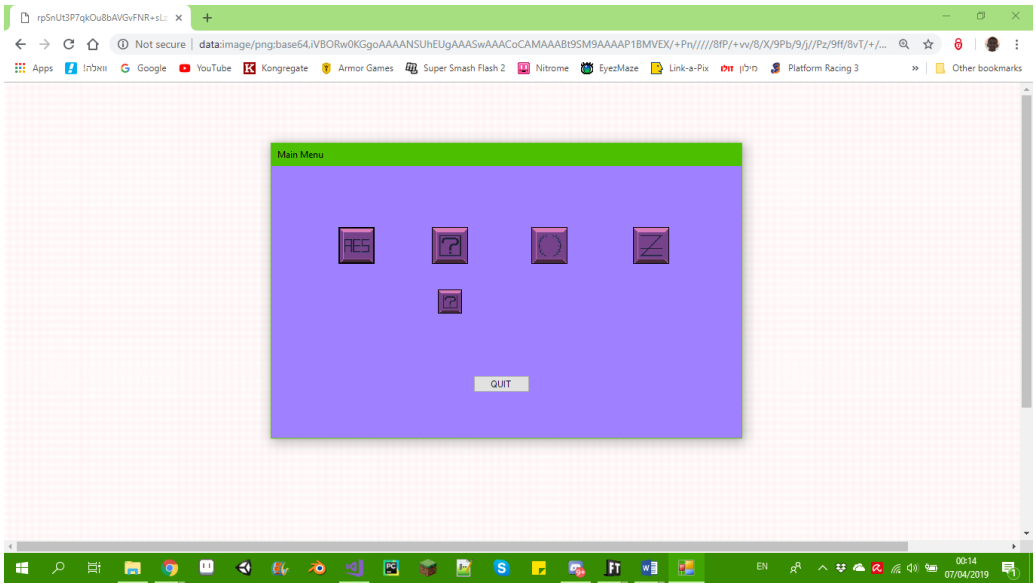
אז, המשתמש יכול להעביר את הקובץ למקום שהוא רוצה במחשב.

הרצה (הפעלת הקובץ)

כיוון שהקובץ הוא קובץ מסוג exe., המשתמש לא צריך להתקין שום תוכנות נוספות כדי להפעיל את התוכנה – רק לחיצה כפולה על הקובץ.

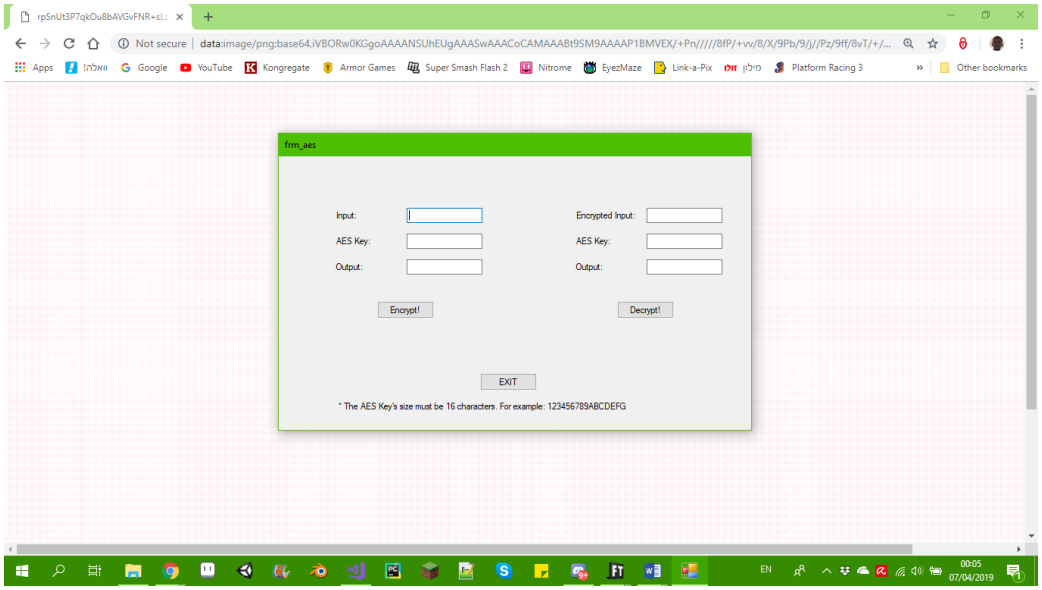
אינטראקציה עם התוכנה

כשפותחים את התוכנה, רואים את החלון הבא:



שם, יש כפתורים שונים שמובילים לחלונות אחרים, כל אחד שייך לסוג הצפנה אחר (הכפתור הקטן הוא יוצא דופן, כי הוא משמש חלק נוסף של הכפתור שמעליו.

הכפתור השמאלי ביותר הוא הכפתור להצפנת AES – שיוביל את המשתמש לחלון הבא:

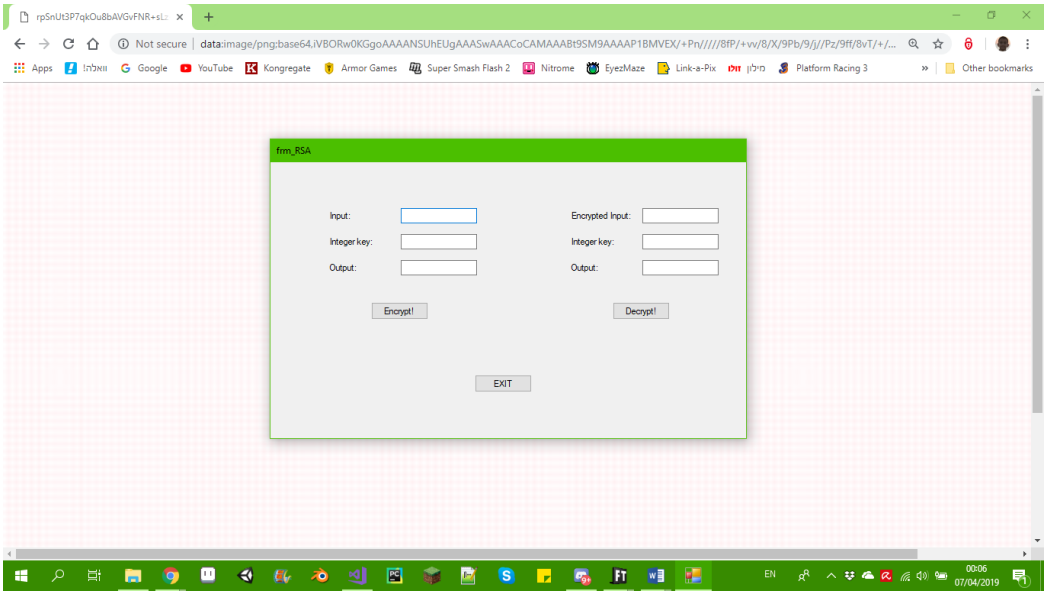


הממשק מקבל אליו מחרוזת בתיבת הטקסט שמסומנת במלה "Input" שמהווה את הטקסט שאותו מצפינים, ומחרוזת נוספת בתיבת הטקסט שמסומנת במלים "AES Key" שמהווה את המפתח שבעזרתו מצפינים. אחרי ששתי המחרוזות הוכנסו והכפתור נלחץ, תתקבל מחרוזת ארוכה וחסרת משמעות מתחת ל-2 תיבות הטקסט – כמו שצריך.

אפשר להעביר את מחרוזת הפלט ואת המפתח לחבר שיש לו את אותה התוכנה, והוא יוכל להכניס את הטקסט המוצפן לתיבת הטקסט שמסומנת במלים "Encrypted Input" שמהווה את הטקסט שאותו מפענחים, ואת המפתח לתיבת הטקסט שמסומנת במלה "AES Key". אחרי שלוחצים על הכפתור, הטקסט שהוכנס תחילה אצל המשתמש הראשון בתיבת הטקסט שמסומנת במלה "Input" יוצג למשתמש האחר.

*המפתחות של הצופן הזה צריכות להיות בגודל של 16 אותיות בהכרח.

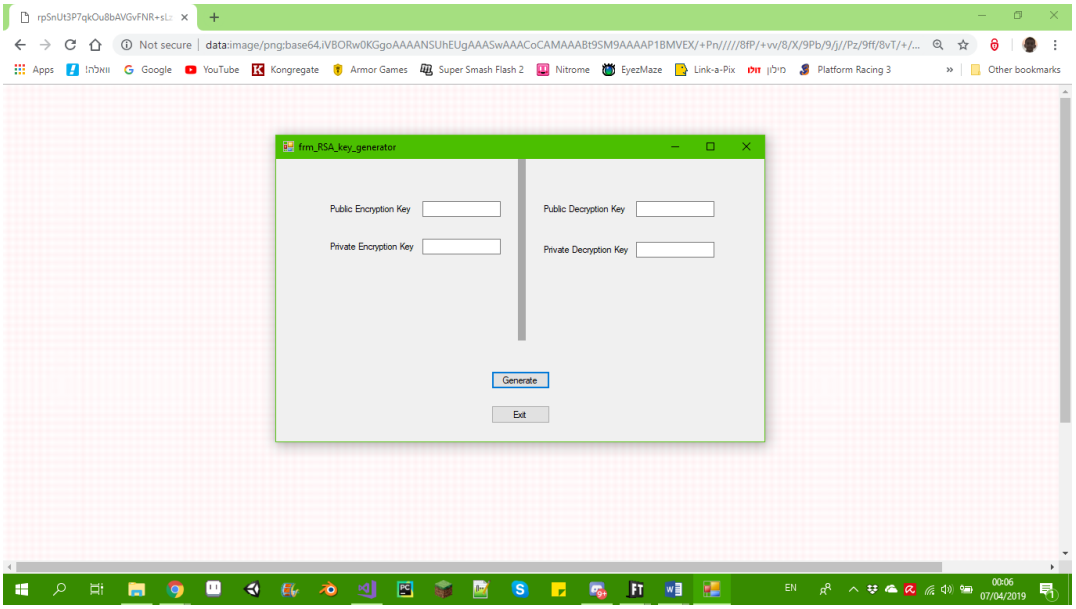
הכפתור השני משמאל הוא הכפתור להצפנת RSA – שיוביל את המשתמש לחלון הבא:



הממשק מקבל אליו מחרוזת בתיבת הטקסט שמסומנת במלה "Input" שמהווה את הטקסט שאותו מצפינים, ומחרוזת נוספת בתיבת הטקסט שמסומנת במלים "AES Key" שמהווה את המפתח שבעזרתו מצפינים. אחרי ששתי המחרוזות הוכנסו והכפתור נלחץ, תתקבל מחרוזת ארוכה וחסרת משמעות מתחת ל-2 תיבות הטקסט – כמו שצריך.

אפשר להעביר את מחרוזת הפלט ואת המפתח לחבר שיש לו את אותה התוכנה, והוא יוכל להכניס את הטקסט המוצפן לתיבת הטקסט שמסומנת במלים "Encrypted Input" שמהווה את הטקסט שאותו מפענחים, ואת המפתח לתיבת הטקסט שמסומנת במלה "AES Key". אחרי שלוחצים על הכפתור, הטקסט שהוכנס תחילה אצל המשתמש הראשון בתיבת הטקסט שמסומנת במלה "Input" יוצג למשתמש האחר.

הכפתור שנמצא מתחת לכפתור הזה הוא הכפתור לייצור מפתח RSA – שיוביל את המשתמש לחלון הבא:

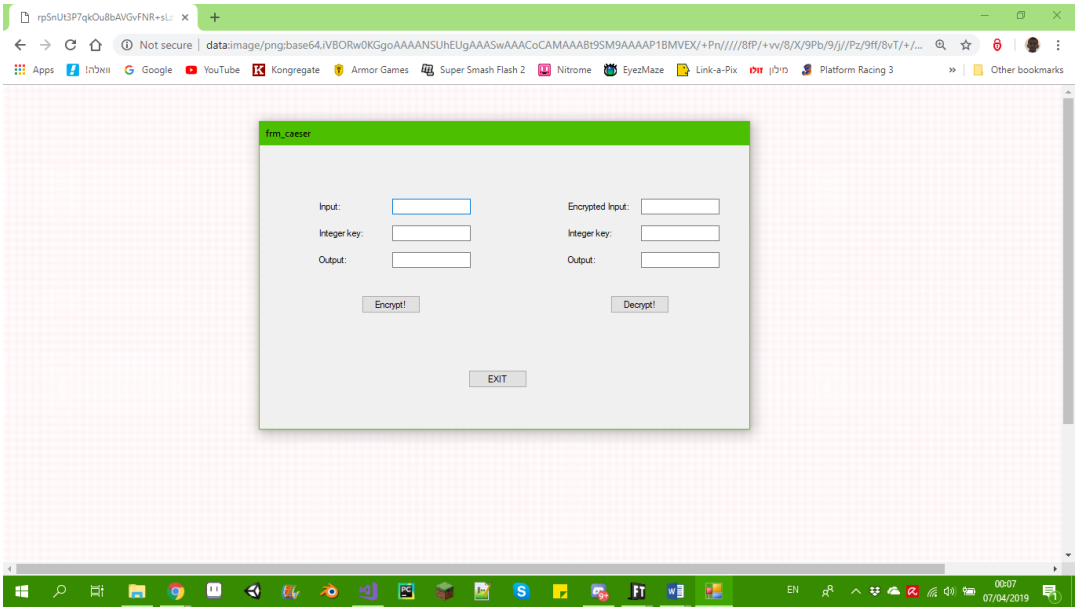


כיוון שהצפנת RSA היא מבוססת על אקראיות במפתחות, פיתחתי בממשק חלון ספציפית לייצור של המפתחות האלו. את האלה יוצרים בלחיצה על הכפתור השם "Generate". אחר כך, **חשוב** לרשום את המפתחות כולן בדף או משהו דומה. דרך העבודה עם המפתחות היא כזאת:

1. קבע 3 תפקידים לאנשים שונים: המצפין, המפענח ומחלק המפתחות.

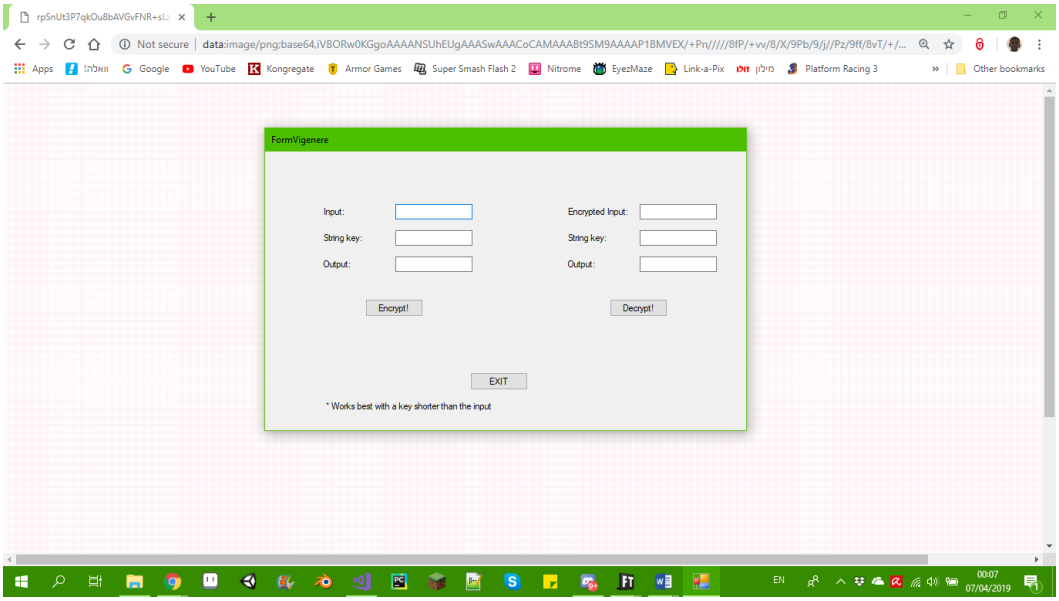
2. אמור למחלק המפתחות ליצור את המפתחות, ולתת בפרטיות למצפין את מפתחות ההצפנה (שבשם שלהם יש "Encryption"), ולמפענח את מפתחות הפיענוח (שבשם שלהם יש "Decryption").

הכפתור השלישי משמאל הוא הכפתור להצפנת קיסר – שיוביל את המשתמש לחלון הבא:



החלון הזה הוא זהה כמעט לחלוטין לחלון הצופן AES, רק שכאן המפתחות הם מספרים.

הכפתור הימני ביותר הוא הכפתור להצפנת Vigenere– שיוביל את המשתמש לחלון הבא:



החלון הזה הוא זהה כמעט לחלוטין לחלון הצופן AES, רק שכאן המפתחות הם מחרוזות, שכדאי שיהיו קצרות יותר מהטקסט שרוצים להצפין אותו.

מדריך למשתמש - מתכנת

עבור המשתמש המתכנת, ישנה ספריית הצפנים שכל הפרויקט שלי מבוסס עליו. כדי לגשת לקובץ הספרייה (שהסוג שלו הוא dll.), צירפתי בר-קוד שמוביל את המשתמש לתיקיית drive (שהיא תיקיית קבצים שאליה נתתי גישה לכל אדם שמגיע אל התיקייה).

הכנס ברקוד

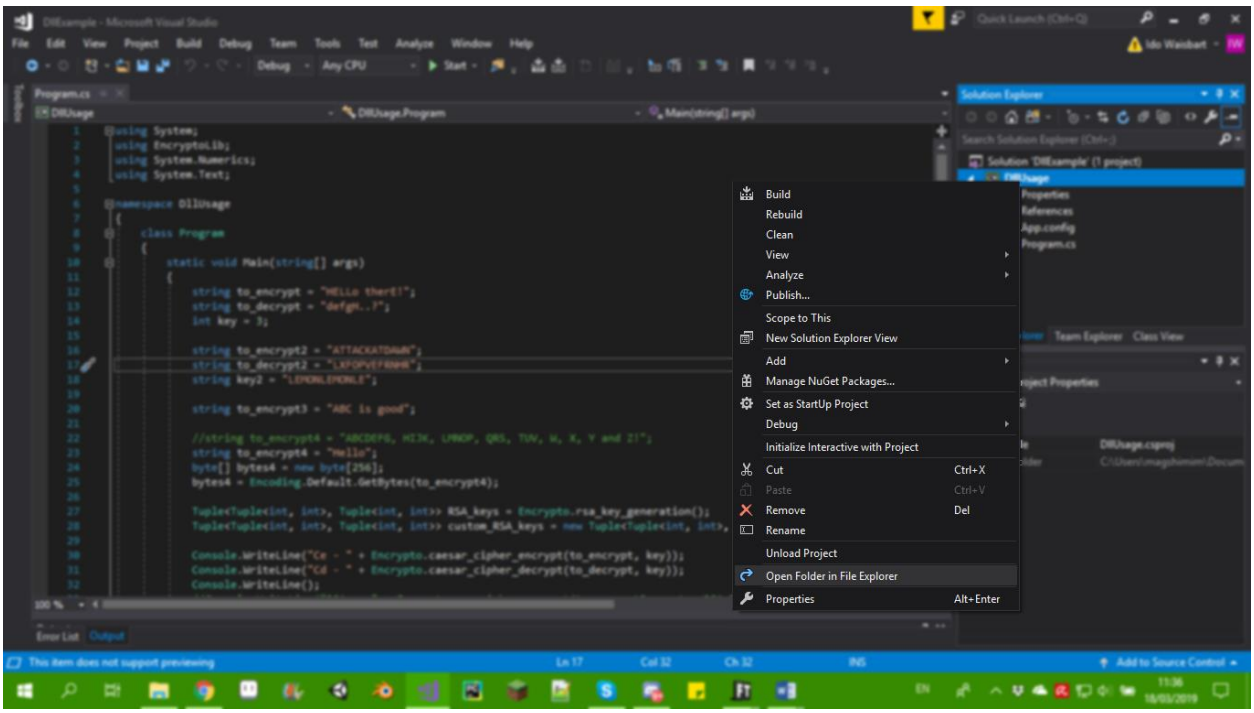
בלחיצה כפולה על הקובץ "Encrypto.dll", המפתח יגלה (ברוב המקרים) שאין לו שום תוכנה שיכולה לפתוח את הקובץ. המפתח לא צריך לפתוח את הקובץ.

מה שהמפתח כן צריך לעשות, זה לצרף את הקובץ לפרויקט תכנותי. כיוון שעבדתי ב-#C לאורך כל תהליך עשיית הפרויקט, אראה הדרכה לצירוף הספרייה רק לפרויקט "C# Forms" ולפרויקט "C# Console App" (ואולי עוד?).

צירוף הספרייה בפרויקטים "C# Forms" ו-"C# Console App"

כאן, יש לגשת לתיקייה: "...\SolutionName\ProjectName\bin\Debug", ולשים שם את הקובץ "Encrypto.dll".

כדי להגיע לתיקייה הזאת, צריך ללחוץ על מקש העכבר הימני על הפרויקט, ואז ללחוץ על הכפתור " Open Folder in File Explorer", כך:



אחרי הכנסת הספרייה לפרויקט, המשתמש מקבל גישה לפונקציות השונות שבה:

string caesar_cipher_encrypt(string to_encrypt, int key)

פונקציה ראשית של צופן קיסר, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

string caesar_cipher_decrypt(string to_decrypt, int key)

פונקציה ראשית של צופן קיסר, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

bool is_prime(int num)

פונקציית עזר של צופן RSA.

int gcd(int num1, int num2)

פונקציית עזר של צופן RSA.

bool is_coprime(int num1, int num2)

פונקציית עזר של צופן RSA.

Tuple<Tuple<int, int>, Tuple<int, int>> rsa_key_generation()

פונקציה ראשית של צופן RSA, שיוצרת את המפתחות הנדרשות להצפנה ולפיענוח של הצופן.

BigInteger[] rsa_cipher_encrypt(string to_encrypt, Tuple<int, int> e_key)

פונקציה ראשית של צופן RSA, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

string rsa_cipher_decrypt(BigInteger[] to_decrypt, Tuple<int, int> d_key)

פונקציה ראשית של צופן RSA, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

int letter_to_value(char ch)

פונקציית עזר של צופן Vigenere.

string vigenere_cipher_encrypt(string to_encrypt, string key)

פונקציה ראשית של צופן Vigenere, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

string vigenere_cipher_decrypt(string to_decrypt, string key)

פונקציה ראשית של צופן Vigenere, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

int hexchar_to_int(char c)

פונקציית עזר של צופן AES.

byte hexstring_to_byte(string s)

פונקציית עזר של צופן AES.

byte aes_byte_sub(byte b, bool encryption)

פונקציה משנית של צופן AES.

byte[,] aes_bytes_sub(byte[,] b, bool encryption)

פונקציה משנית של צופן AES.

byte[,] aes_shift_rows(byte[,] b, bool encryption)

פונקציה משנית של צופן AES.

byte[,] internet_aes_mix_cols(byte[,] b, bool encryption)

פונקציה משנית של צופן AES.

byte internet_GMul(byte b1, byte b2)

פונקציה משנית של צופן AES.

byte[,] aes_add_round_key(byte[,] b, byte[] key)

פונקציה משנית של צופן AES.

byte[] aes_cipher_encrypt(byte[] to_encrypt, byte[] key)

פונקציה ראשית של צופן AES, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

byte[] aes_cipher_decrypt(byte[] to_decrypt, byte[] key)

פונקציה ראשית של צופן AES, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

מדריך למפתח

עבור המפתח, כמו למשתמש המתכנת, ישנה ספריית הצפנים שכל הפרויקט שלי מבוסס עליו. כדי לגשת לקובץ הספרייה (שהסוג שלו הוא dll.), צירפתי בר-קוד שמוביל את המשתמש לתיקיית drive (שהיא תיקיית קבצים שאליה נתתי גישה לכל אדם שמגיע אל התיקייה).

מבחינה טכנית, הספרייה מורכבת ממחלקה אחת (EncryptoLib) שכוללת בתוכה הרבה מאוד פונקציות.

string caesar_cipher_encrypt(string to_encrypt, int key)

פונקציה ראשית של צופן קיסר, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

כל אות מוחלפת באות שנמצאת אחריה מספר אותיות באלף בית, ואחרי האות האחרונה (ת', בעברית) באה האות הראשונה (א', בעברית).

Ifmmp <-1– Hello

string caesar_cipher_decrypt(string to_decrypt, int key)

פונקציה ראשית של צופן קיסר, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

כל אות מוחלפת באות שנמצאת לפניה מספר אותיות באלף בית, ולפני האות הראשונה (א', בעברית) באה האות האחרונה (ת', בעברית).

Gdkkn <-1– Hello

Tuple<Tuple<int, int>, Tuple<int, int>> rsa_key_generation()

פונקציה ראשית של צופן RSA, שיוצרת את המפתחות הנדרשות להצפנה ולפיענוח של הצופן.

BigInteger[] rsa_cipher_encrypt(string to_encrypt, Tuple<int, int> e_key)

פונקציה ראשית של צופן RSA, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

שמשתמש במפתחות שנוצרות באופן אקראי ואז עוברות תהליך של הוספת שכבות של סודיות כך שהמפתחות קשות לפיענוח ותלויות אחת בשנייה, ואז מוצפנות דרך שורת הקוד:

.number = Pow(char, private_encryption_key) % public_encryption_key

string rsa_cipher_decrypt(BigInteger[] to_decrypt, Tuple<int, int> d_key)

פונקציה ראשית של צופן RSA, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

שמשתמש במפתחות שנוצרות באופן אקראי ואז עוברות תהליך של הוספת שכבות של סודיות כך שהמפתחות קשות לפיענוח ותלויות אחת בשנייה, ואז מוצפנות דרך שורת הקוד:

number = Pow(char, private_encryption_key) % public_encryption_key.

string vigenere_cipher_encrypt(string to_encrypt, string key)

פונקציה ראשית של צופן Vigenere, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

כל אות מוחלפת באות שנמצאת אחריה X אותיות באלף בית, ואחרי האות האחרונה (ת', בעברית) באה האות הראשונה (א', בעברית), כאשר X משמעותו הערך הגימטרי של האות של המפתח באותו המיקום של האות המוצפנת.

Hello –ABC-> Igomq

string vigenere_cipher_decrypt(string to_decrypt, string key)

פונקציה ראשית של צופן Vigenere, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

כל אות מוחלפת באות שנמצאת לפניה X אותיות באלף בית, ולפני האות הראשונה (א', בעברית) באה האות האחרונה (ת', בעברית), כאשר X משמעותו הערך הגימטרי של האות של המפתח באותו המיקום של האות המוצפנת.

Hello –ABC-> Gcikm

byte[] aes_cipher_encrypt(byte[] to_encrypt, byte[] key)

פונקציה ראשית של צופן AES, שמצפינה את to_encrypt דרך המפתח key ומחזירה את תוצאת ההצפנה.

כל 16 אותיות בטקסט שמצפינים אותו עוברות תהליך שכולל 4 שלבים, שהם: bytes substitution, shift rows, mix columns ו-add round key.

byte[] aes_cipher_decrypt(byte[] to_decrypt, byte[] key)

פונקציה ראשית של צופן AES, שמפענחת את to_decrypt דרך המפתח key ומחזירה את תוצאת הפיענוח.

כל 16 אותיות בטקסט שמצפינים אותו עוברות תהליך שכולל 4 שלבים, שהם: bytes substitution, shift rows, mix columns ו-add round key.

תרשים UML של המודול היחיד בפרויקט

רקע תיאורטי (מדעי)

בעמוד הזה אפשר לקרוא על האלגוריתמים של כל הצפנים שמימשתי.

צופן קיסר – כל אות מוחלפת באות שנמצאת אחריה מספר אותיות באלף בית, ואחרי האות האחרונה (ת', בעברית) באה האות הראשונה (א', בעברית).

Ifmmp <-1– Hello

צופן Vigenere – כל אות מוחלפת באות שנמצאת אחריה X אותיות באלף בית, ואחרי האות האחרונה (ת', בעברית) באה האות הראשונה (א', בעברית), כאשר X משמעותו הערך הגימטרי של האות של המפתח באותו המיקום של האות המוצפנת.

Igomq <-ABC– Hello

צופן RSA – שמשתמש במפתחות שנוצרות באופן אקראי ואז עוברות תהליך של הוספת שכבות של סודיות כך שהמפתחות קשות לפיענוח ותלויות אחת בשנייה, ואז מוצפנות דרך שורת הקוד:

.number = Pow(char, private_encryption_key) % public_encryption_key

צופן ASE – כל 16 אותיות בטקסט שמצפינים אותו עוברות תהליך שכולל 4 שלבים, שהם: bytes substitution, shift rows, mix columns ו-add round key.

סיכום אישי

בסך הכל, הפרויקט שימש לי הוכחה לכך שאני יודע לתכנת טוב. לא ציפיתי שאצליח לתכנת את ההצפנות והפיענוחים בכל כך קלות יחסית.
דבר שתרם לי להצליח לעבור את הפרויקט במהירות היה סדר הצפנים שמימשתי. לא ידעתי למה לצפות מהצפנים AES, RSA ו-Vigenere. למזלי, בחרתי בסדר מהקל לקשה (קודם Vigenere, אז RSA ואז AES).

עכשיו, כשסיימתי את הפרויקט, יש לי הרבה מימושים של פונקציות מאוד שימושיות (לא רק פונקציות ההצפנה והפיענוח), יש לי ידע בייצור של קבצי ספרייה ובייבוא שלהם וגם עבודה מעניינת להראות למראיינים שמעוניינים לראות איזה פרויקטים עשיתי בעבר.

נספחים

המימוש של – https://en.wikipedia.org/wiki/Rijndael_MixColumns#Implementation_example
internet_aes_mix_cols() ושל internet_Gmul().

המימוש של - https://www.quora.com/Whats-the-best-algorithm-to-check-if-a-number-is-prime
is_prime().

https://www.geeksforgeeks.org/check-two-numbers-co-prime-not/ - המימוש של gcd().