MSecret

File Encryption program

שם מגישה : מעין מרדכי

תעודת זהות : 318607645

בית ספר : מרכז חינוך ליאו-באק

מקצוע : הגנת סייבר

שם פרויקט : Msecret

תוכן עניינים

מבוא

MSecretהיא תוכנית להצפנת קבצים באמצעות הצפנת בלוקים. בתוכנית ניתן להצפין קבצים בשתי הצפנות עליהן אכתוב בהמשך. בתוכנית ישנם שני מצבים:

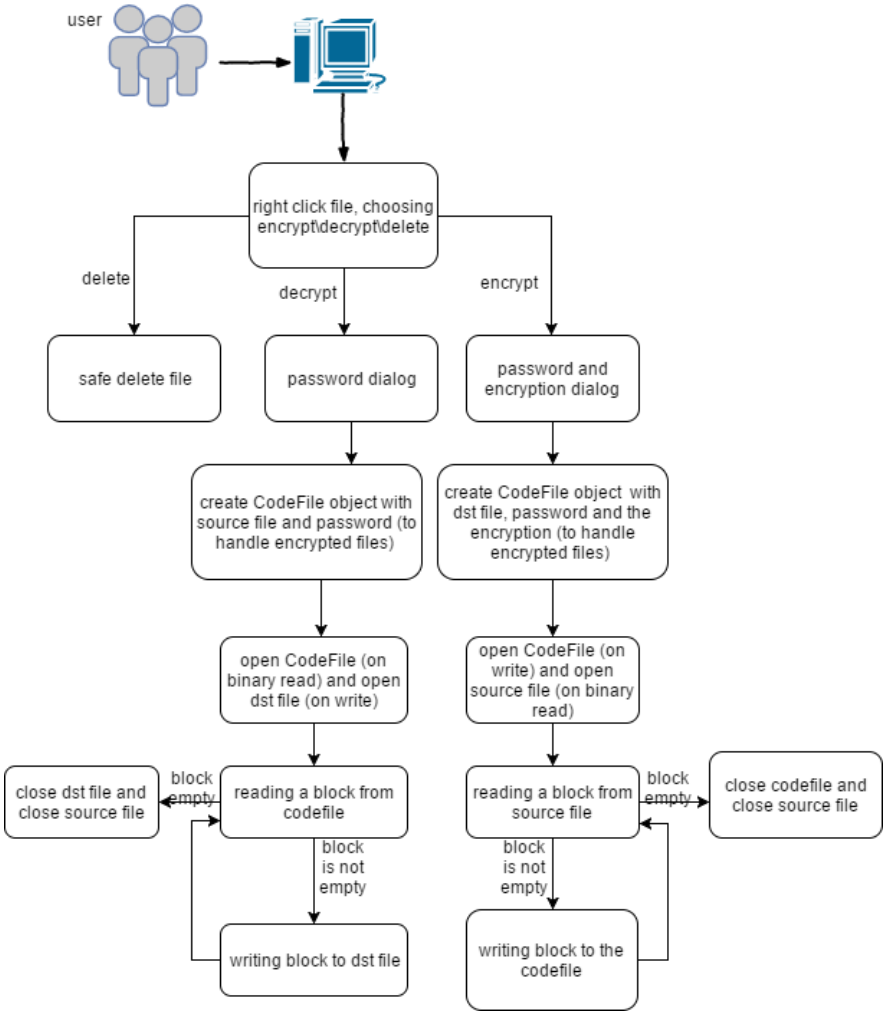
- כפתור ימיני על קובץ כלשהו, שם (לאחר התקנת התוכנית) יופיע encrypt, וdelete. בלחיצה על encrypt יופיע דיאלוג שבו המשתמש צריך לשים סיסמא ולבחור הצפנה ואם הקובץ הוא תיקייה לבחור אם הוא רוצה להצפין באופן רקורסיבי כלומר אם יש בתוך התיקייה עוד תיקיות להצפין גם אותן או לא באופן רקורסיבי ואז להצפין רק את הקבצים שבתוך התיקייה הראשונה גם אם יש עוד תיקיות בפנים. ולאחר מכן התוכנית תצפין את הקובץ לקובץ אחר ששמו הוא קובץ המקור רק עם סיומת של שם התוכנית, MSecret (אותו דבר לתיקייה). בלחיצה על המחיקה של התוכנית בכפתור הימני הקובץ ימחק, במחיקה מיוחדת - בטוחה עליה אפרט בהמשך העבודה. ישנה אפשרות גם לפענוח, בלחיצה על הכפתור הימני בקבצים שסופם הוא MSecret תופיע האפשרות decrypt שבלחיצה עליה יופיע דיאלוג בו המשתמש צריך לשים סיסמא (ואם זו תיקייה לבחור אם באופן רקורסיבי או לא) ולאחר מכן הקובץ יפעונח לתוך קובץ ששמו הוא אותו שם של הקובץ רק ללא הסיומת MSecret.
- מצב של תיקיה מוצפנת שאליה הגישה היחידה היא דרך ממשק המשתמש. השמות של הקבצים בתיקיה מוצפנים (ובאורך קבוע) וכך גם הקבצים. כאשר מריצים מצב זה (דרך כפתור ימני על תיקייה) עולה דיאלוג שבו המשתמש צריך לרשום סיסמא ולבחור הצפנה. לאחר מכן למשתמש נפתח ממשק בו מופיעה לו רשימה של כל השמות המפוענחים של הקבצים. בממשק זה המשתמש יכול ליצור קבצים חדשים, לראות את המידע המפוענח בקבצים, לערוך קבצים, למחוק קבצים ולשנות את שמם. במצב זה אפשר ליצור רק קבצי טקסט.

פרויקט זה נועד לאבטח מידע בעזרת הצפנה. פרויקט זה מנסה לוודא שהמידע לא יגיע למישהו ללא הסיסמא ולמנוע חשיפת מידע.

ארכיטקטורה

right click

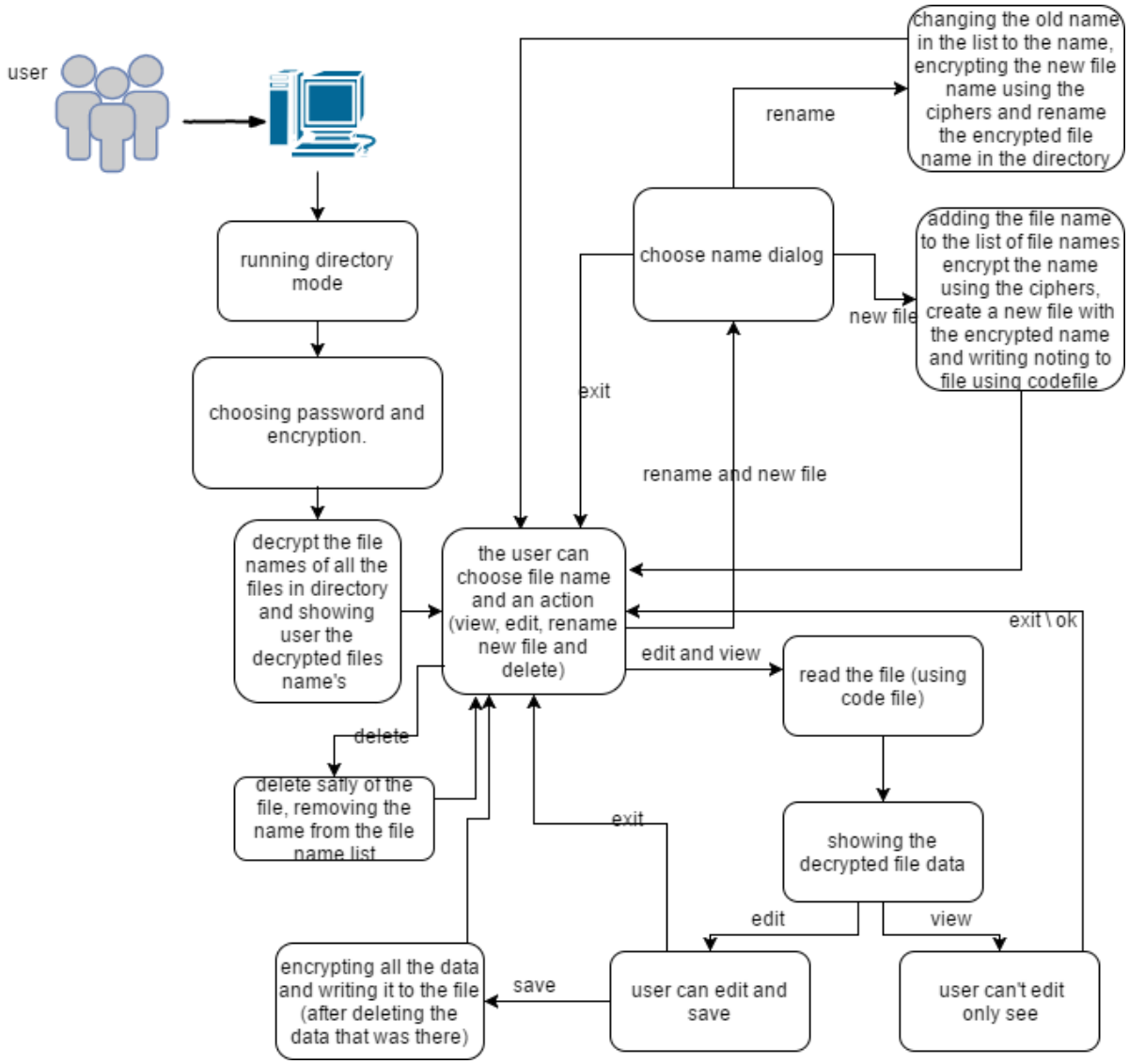
כאשר לוחצים על אחת מהאפשרויות של התוכנית שלי מהright click על קובץ שהוא לא תיקייה הדיאגרמה הבאה מתארת מה קורה :



כאשר הקובץ הוא כן תיקייה לדיאלוג הנוצר לאחר הבחירה גם להצפן וגם לפענח ניתנת האפשרות לבחור האם ההצפנה/פענוח יהיו רקורסיביים או לא. כלומר אם יש בתוך התיקייה עוד תיקיות האם המשתמש רוצה שנצפין/נפענח את התיקיות שבתוך אותה תיקייה? אם לא אז התוכנית שלי עוברת על כל הקבצים בתיקייה שאינם תיקיות ומצפינה/מפענחת אותם לתוך תיקייה חדשה ששמה הוא שם המקורית רק עם סיום של MSecret. אם כן התוכנית שלי עוברת על כל הקבצים בתוך התיקייה ואם קובץ הוא תיקייה הפונקציה קוראת לעצמה על הסיפרייה שבפנים. אם הקובץ אינו תיקייה אז התוכנית שלי מצפינה/מפענחת לתוך התיקייה הנכונה.

directory mode

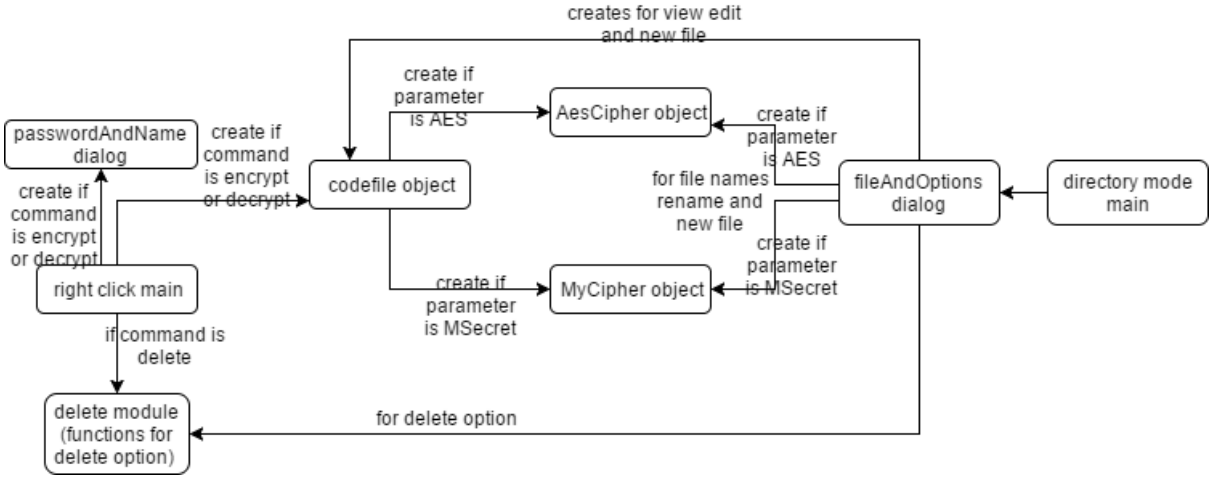
כאשר מפעילים את מצב תיקייה ניתן לתאר את מה שקורה על פי הדיאגרמה הבאה:



delete

כאשר מוחקים קובץ מהתוכנית כדי לגרום לכך שלא יצליחו לזהות את המידע שהיה בקובץ, שעדיין נמצא בדיסק הקשיח, משתמשים במחיקה מיוחדת. במחיקה זו לפני שמוחקים את הקובץ כותבים אליו (אל כל חלק שהיה כתוב בו) 0 (בינארית), אל כולו ואז אל כולו 1 (בינארית) ואז כותבים אל כולו בית אקראי. חוזרים על כתיבת בית אקראי שונה אל כל הקובץ כ6 פעמים. ולאחר מכן מוחקים את הקובץ וכך מאוד קשה לשחזר את המידע שעל הדיסק.

העבודה בין חלקי התוכנית



רקע תיאורטי

הצפנה -הסתרת משמעותו של מסר קריא באמצעות פונקציה שמקבלת כפרמטר מפתח הצפנה והופכת את המסר לרצף של סימנים המכונה צופן שאינו מובן לאיש. שחזור הטקסט המוצפן למצבו הקריא באמצעות פונקציה הופכית מתאימה עם מפתח הפענוח, קרוי פענוח. המונח צופן (Cipher) מתייחס לאלגוריתם הצפנה בדרך כלל במחשב, כאשר קלט האלגוריתם נקרא Plaintext ואילו פלט האלגוריתם נקרא Ciphertext. פעולת אלגוריתם ההצפנה נשלטת על ידי מפתח ההצפנה הסודי הידוע רק לשולח והמקבל.

צופן בלוקים - הוא צופן הפועל על מחרוזת סיביות באורך קבוע הנקראת בלוק באמצעות טרנספורמציה קבועה. צופן בלוקים מקבל בלוק של סיביות (plaintext) ומפתח הצפנה סודי ומפיק בלוק טקסט מוצפן (ciphertext). אופן פעולת הטרנספורמציה נשלט על ידי מפתח ההצפנה. הפענוח מתבצע באופן רק דומה, האלגוריתם מקבל בלוק סיביות טקסט מוצפן והמפתח שאיתו הוצפן ומחזיר את בלוק הסיביות המקורי. אפשר להצפין מסר באורך העולה על גודל הבלוק שהצופן מסוגל לקבל, פשוט על ידי חלוקתו לבלוקים בגודל הרצוי והצפנתם בזה אחר זה. כל הבלוקים מוצפנים עם אותו מפתח, עובדה שמשפיעה על ביטחון ההצפנה. בגלל שאלגוריתם ההצפנה דטרמיניסטי, במקרה שמוצפנים בלוקים זהים של plaintext עם אותו מפתח, התוצאות תהיה בלוקים זהים של ciphertext. עובדה זו חושפת מידע מסוים ועלולה להוות חולשה שיש להימנע ממנה. כדי לפתור בעיה זו אפשר להפעיל את הצופן באחד מסגנונות ההפעלה של צופן בלוקים שמבטיח מידה של אקראיות (כמו מצב **CBC**), כך ששני בלוקים זהים יוצפנו בצורה אחרת והתוצאה תהיה תמיד שונה. כאשר גודל הבלוקים קטן מאוד הם חושפים את האלגוריתם להתקפות וככל שהבלוק גדול יותר כך סיבוכיות ההתקפה על הצופן תגדל, אך יעילותו תרד בהתאם. האיזון בין יעילות לביטחון הוא הגורם המכריע בקביעת גודל הבלוק.

מצבי הפעלה של צופן בלוקים - מצב הפעלה מתייחס לאלגוריתם שמגדיר כיצד להפעיל צופן בלוקים להצפנת plaintaxt שאורכו גדול מאורך הבלוק שהצופן מסוגל להצפין בבת אחת. צופן בלוקים לבדו מסוגל רק להבטיח סודיות של בלוק plaintext. מעצם ההגדרה בהינתן צופן בלוקים כמו **AES** ומפתח הצפנה זהה, הצופן ימיר תמיד בלוק קלט זהה לבלוק ciphertext זהה. לכן ניתן לזהות הצפנה חוזרת של בלוקים זהים עם אותו מפתח. כדי למנוע את הזיהוי משתמשים בסגנונות הפעלה.
מצב **CBC** - שרשור בלוקים מוצפנים, מצב הפעלה המשלב וקטור אתחול IV. במצב זה כל בלוק plaintext מחובר בפעולה בינארית עם תוצאת הצפנת הבלוק הקודם לפני הצפנתו. הצפנה זו גורמת לכך שלא יהיו בלוקים זהים עבור אותו טקסט (בתנאי שמשנים את הIV או את המפתח בהצפנות שונות). בנוסף, לשיטת CBC יש יכולת התאוששות עצמית כלומר אם בלוק שלם מתוך הצופן שגוי או נעדר רק הבלוק הבא לא יפוענח כראוי, כל היתר לא יפגעו (אך שגיאה בבלוק אחד במהלך ההצפנה תגרור שגיאה בכל הבלוקים הבאים).

IV - וקטור אתחול, מחרוזת כלשהי באורך מוגדר המשמשת להתחלת תהליך ההצפנה בצופן זרם או בצופן בלוקים במצב הפעלה שרשור (CBC).
בצופן בלוקים מיישמים מצב הפעלה המדמה צופן זרם כלומר גורם לתלות בין הבלוקים על ידי הפעלה פעולה חישובית בין כל בלוק עם בלוק מוצפן קודם. פעולה זו נקראת "שירשור" והמימוש הנפוץ בו הוא

מצב CBC בו כל בלוק טקסט גלוי מחובר בXOR עם הבלוק המוצפן הקודם לפני שהוא מוצפן. היות שהשרשור מתבצע עם בלוק קודם, לא ניתן לבצע שרשור בבלוק הראשון כי לא קיים בלוק קודם. מסיבה זו משתמשים בוקטור אתחול כבלוק דמה.
וקטור האתחול אינו חייב להיות סודי אך צריך להיות ידוע גם למקבל הצופן, אפשר לשדרו גם בצורה גלויה. למעשה אפשר להשתמש במונה או מספר מוסכם אחר בין השולח והמקבל, כל עוד מובטח שלא נעשה בו שימוש חוזר.

AES - אלגוריתם הצפנה, צופן בלוקים סימטרי. הצופן אומץ על ידי ממשלת ארצות הברית באופן רשמי להצפנת נתונים מסווגים עבור הממשל. זהו הצופן הסימטרי הפומבי הראשון שקיבל את אישור הסוכנות לביטחון לאומי האמריקאי כראוי להצפנת נתונים המוגדרים ברמת סיווג SECRET ו TOP SECRET עבור ממשלת ארצות הברית, אם נעשה בו שימוש כחלק ממודול הצפנה מאושר. אלגוריתם AES נמצא בשימוש מעשי נרחב בכל העולם הן בתוכנה והן בחומרה וידוע כאלגוריתם בטוח.

פונקצית גיבוב (**Hash**) - פונקציה שממירה קלט חופשי באורך משתנה לפלט באורך קבוע, בדרך כלל קצר בהרבה. פונקציות Hash עלולות לתת את אותו פלט ע כרגע ז'בוטינסקיבור מספר קלטים שונים. פונקצית Hash שנחשבת טובה היא פונקציה שבה ההסתברות שיצא לכמה קלטים שונים את אותו הפלט היא קטנה מאוד (כמו 1sha). בהצפנה לפעמים משתמשים בפונקצית גיבוב על הסיסמא כדי לקבוע את מפתח באורך קבוע מסיסמא בכל אורך.

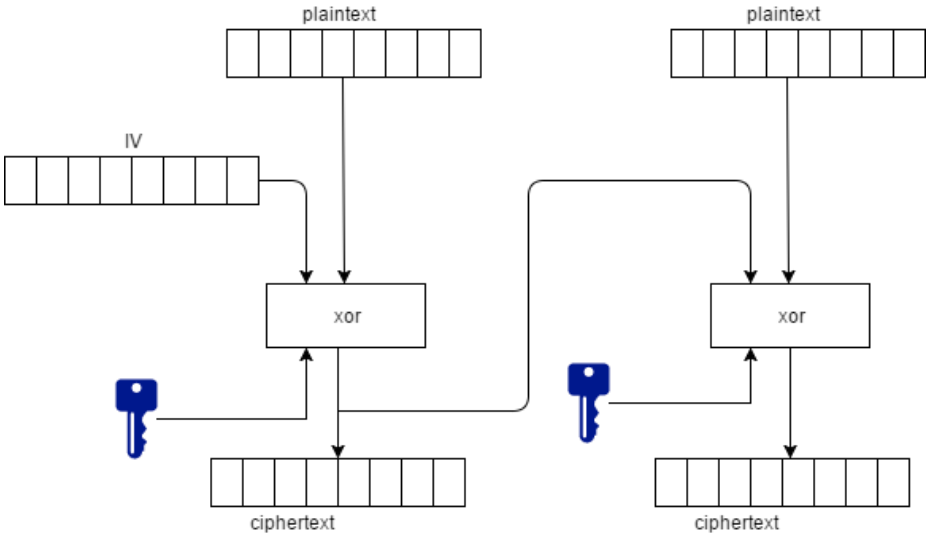
Padding - בהצפנות של בלוקים אורך כל בלוק יכול להיות רק מספר מסוים וכך נוצר מצב שאורך המידע צריך להתחלק באורך הבלוק אך יש פעמים שאורך המידע שצריך להצפין לא מתחלק באורך של בלוק כך יוצא מצב שיש מידע שנותר, שלא ניתן להצפין אותו בגלל אורכו. כדי לפתור בעיה זו משלימים למידע זה את מספר הבתים החסרים בבתים שבדרך כלל בכל אחד מהם כתוב את מספר הבתים שהיו חסרים (כל הבתים שמשלימים הם אותו הדבר). לאחר הפענוח ניתן להוריד את הבתים שהוסיפו בהצפנה.(PKCS#5)

Ciphers

בפרויקט שלי יש שתי הצפנות:

MyCipher

הצפנת MyCipher - MSecret היא הצפנת בלוקים. אורך כל בלוק הוא 64 סיביות. הצפנה זו משלבת בתוכה את וקטור האתחול IV שחייב להיות גם באורך של 64 סיביות. ההצפנה היא בעצם xor ל64 סיביות האחרונים של הhash של הסיסמא (שימוש בsha1), לplaintext ולהצפנה של הבלוק הקודם. בבלוק הראשון אין בלוק קודם ולכן הIV משמש כהצפנת הבלוק הקודם לבלוק הראשון.



במקרים בהם נשאר padding מוסיפים בתים אקראיים להשלמה לאורך של 7 בתים (אורך בלוק פחות הבית האחרון), הבית האחרון הוא מספר הבתים האקראיים שהוספנו בהקסדצימלי. אם לא נשארו בתים עדיין מוסיפים בתים אקראיים כשהבית האחרון הוא האורך שלהם, שזה הבלוק האחרון.

AesCipher

הצפנת AesCipher - CBC היא הצפנת בלוקים. אורך כל בלוק הוא 128 סיביות. הצפנה זו משלבת בתוכה את וקטור האתחול IV שחייב להיות גם באורך של 128 סיביות. בכדי שהתוכנית שלי תתמוך בהצפנה זו השתמשתי בסיפרייה pyaes בה יש יישום של ההצפנה בפייטון.

בסוף ההצפנה (padding) מוסיפים בתים אקראיים להשלמה לאורך של 15 בתים (אורך בלוק פחות הבית האחרון), הבית האחרון הוא מספר הבתים האקראיים שהוספנו בהקסדצימלי. אם לא נשארו בתים עדיין מוסיפים בתים אקראיים כשהבית האחרון הוא האורך שלהם שזה הבלוק האחרון.

טכנולוגיה

דיסק קשיח - רכיב במחשב המשמש לשמירת נתונים. הדיסק הקשיח עובד בעזרת שדה מגנטי, הנתונים נשמרים בבינארית משום שכל שדה מגנטי זעיר מסמן 0 או 1 בהתאם לכיוונו. בדיסק הקשיח אין אפשרות למחוק מידע אלא רק לכתוב עליו. כאשר מוחקים קבצים במחשב המחיקה לא מעלימה את המידע אלא רק גורמת לכך שהמערכת תראה את המיקום של המידע שנמחק כפנוי. עד שלא כותבים על אותו מקום מידע חדש אפשר לשחזר את המידע וגם לאחר שכותבים לאותו מקום מידע חדש יש אפשרות לשחזר בעזרת כוח מגנטי מיקרוסקופי (מכיוון שהמידע הוא לפי הכיוון ויש אינסוף כיוונים כך שעל פי הכיוון של השדה המגנטי הזעיר יכולה להיות האפשרות לזהות את כיוונו הקודם ובכך את ערכו), ככל שכותבים יותר על המקום בו היה המידע כך קשה יותר לשחזר את המידע.

רקורסיה - בתכנות רקורסיה היא התופעה שבה פונקציה מוגדרת באמצעות עצמה (כלומר קוראת לעצמה).

Environment variables - משתנה המחזיק בערך אחד/יותר שהם בעלי שמות דינאמיים, ומשפיעים על הדרך בה יתנהגו תהליכים בסביבה הנתונה, כמו מעטפת מערכת ההפעלה של המחשב. אם כי מקומם הוא בשכבת ליבת מערכת ההפעלה, ולא במעטפת.

PYTHONPATH- משתנה סביבתי שבו נמצא הנתיב (Path) המוגדר לחיפוש של קבצי מודולים של פייטון.

GUI - ממשק משתמש גרפי לתוכנה או לאתר אינטרנט המבוסס על עיצוב גרפי של המסך המוצג למשתמש.

wxpython - אחת מהערכות של GUI עבור שפת התכנות פייטון המאפשרת למתכנתי פייטון ליצור תוכניות עם ממשק משתמש איכותי בקלות.

pyaes - יישום של אלגוריתם ההצפנה AES בפייטון (ורק פייטון) במצבי ההפעלה CBC, CFB, CTR, ECB ו.OFB

Registry - הרישום של Windows הוא מסד נתונים שמאחסן מגוון רחב של הגדרות תצורה. כמעט כל הגדרות התצורה הכלולות ב- Windows מנוהלות על ידי הregistry editor (כמו להוסיף תוכניות לכפתור ימני (רגיל, קבצים ואפילו קבצים עם סיומת מסוימת), לשנות משתנים סביבתיים וכדומה).

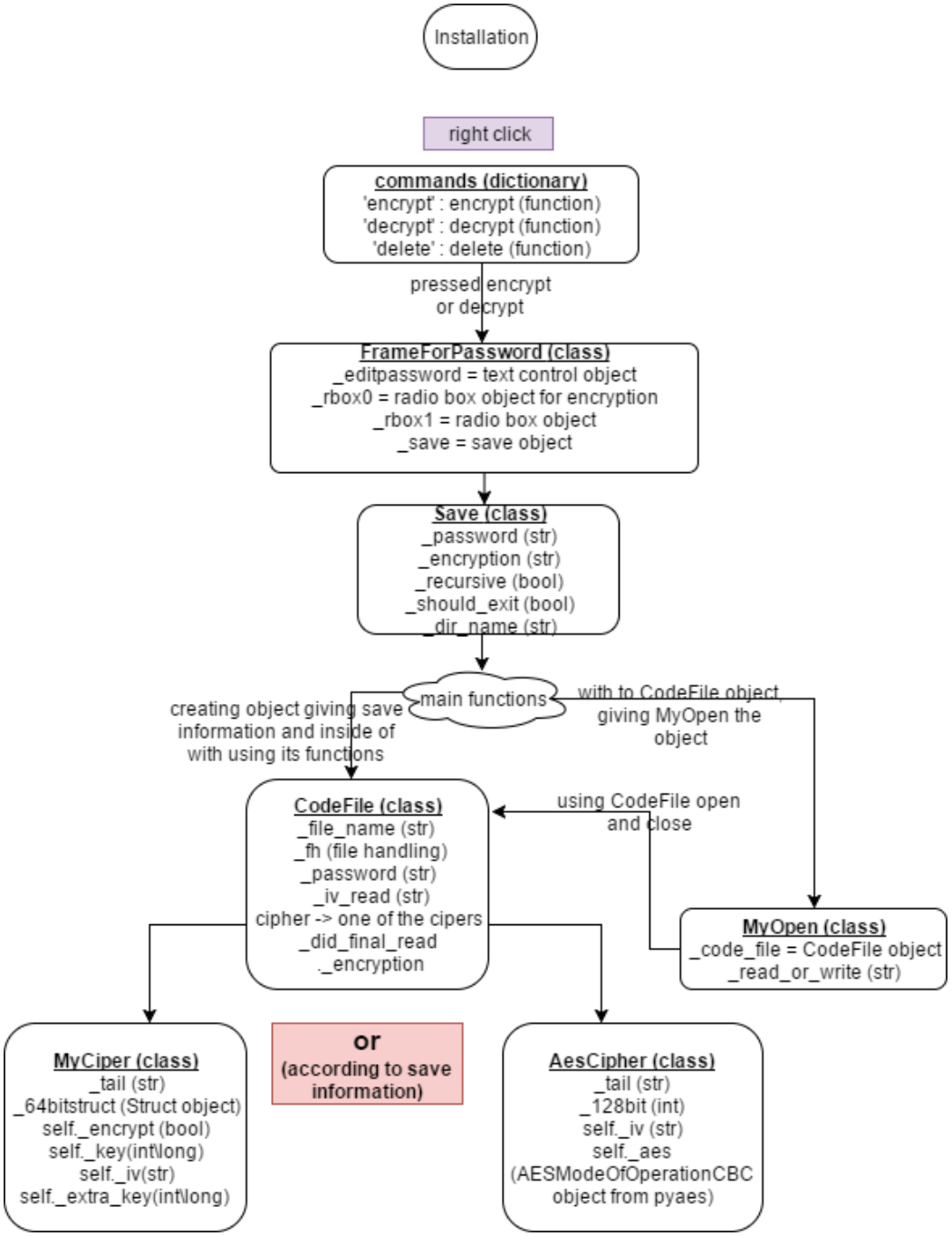
מימוש

ישויות

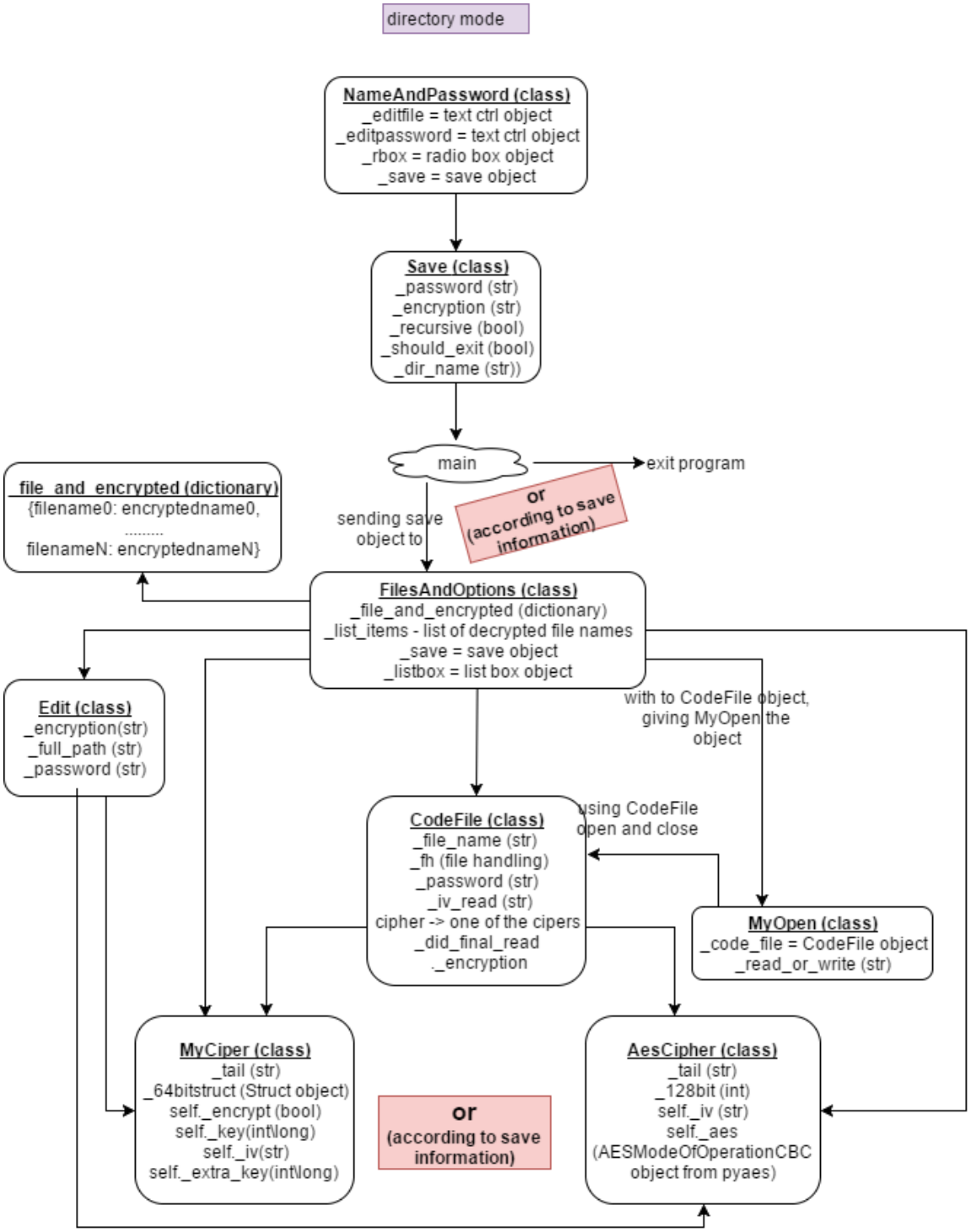
בפרויקט שלי קיימים האובייקטים הבאים:

- Installation - מחלקה היורשת מהמחלקה Frame של wxpython. המחלקה היא חלון בו מוסבר למשתמש על התוכנית שהוא עומד להתקין וברגע שהמשתמש לוחץ על התקנה המחלקה מתקינה את התוכנית (כלומר העבודה עם הregistry editor).
- FrameForPassword - מחלקה היורשת מהמחלקה Frame של wxpython. המחלקה היא דיאלוג עבור המשתמש על פי הפרמטרים שהיא מקבלת (הצפנה או פענוח, תיקייה או של קובץ רגיל) בו המשתמש ממלא את הפרטים.
- Save - מחלקה הנועדה לשמור מידע מאובייקטים של ממשק המשתמש. ברגע שאובייקטים של ממשק המשתמש מסיימים את פעולתם הם נמחקים וכך גם המידע שבהם לכן מחלקה זו מאפשר את שמירת המידע.
- MyOpen - מחלקה הנועדה לתת תמיכה לCodeFile בפונקציה with.
- CodeFile - מחלקה הנועדה לטיפול בקבצים מוצפנים.
- MyCipher - מחלקת צופן הנועד להצפנת בלוקים של מידע. (הצפנה שיצרתי)
- AesCipher - מחלקת צופן הנועד להצפנת בלוקים של מידע. (שימוש בהצפנה המוכרת AES).
- FilesAndOptions -מחלקה היורשת מהמחלקה Frame של wxpython. המחלקה היא ממשק לעבודה עם תיקייה מוצפנת עבור המשתמש. בממשק זה המשתמש רואה רשימה של שמות קבצים מפוענחים מתוך התיקייה, המשתמש יכול להוסיף קבצים, לראות את התוכן של הקבצים, לשנות שמות של קבצים, למחוק קבצים ולערוך קבצים דרך ממשק זה. מחלקה זו יוצרת בתוכה מופע למחלקות CodeFile, MyOpen לצורך הצפנת הקבצים החדשים (כלומר הצפנה של קובץ ריק) כאשר יוצרים קובץ חדש, לפענוח כאשר המשתמש מבקש לראות קובץ. היא היוצרת מופע של אחת המחלקות MyCipher\AesCipher כדי להצפין ולפענח שמות קבצים. המחלקה משתמשת באובייקט Save כדי לקבל את המידע שהמשתמש נתן. בנוסף מחלקה זו קוראת לאובייקט Edit המטפל בעריכת קבצים מוצפנים.
- Edit - מחלקה היורשת מהמחלקה Frame של wxpython. מחלקה זו היא דיאלוג של תיבת טקסט. מחלקה זו מפענחת את הכתוב בקובץ מסוים וכותבת את המידע המפוענח אל תיבת הטקסט. המשתמש יכול לערוך את המידע וכשמסיים ללחוץ על הכפתור save ואז המידע הערוך יוצפן ויכתב בקובץ.
מחלקה זו יוצרת בתוכה וקוראת לאובייקטים CodeFile, MyOpen לצורך הצפנה ופענוח של הקובץ.

ניתן לתאר את העבודה בין מבני הנתונים השונים בחלק הright click בעזרת הדיאגרמה הבאה:



ניתן לתאר את העבודה בין מבני הנתונים השונים בחלק של directory mode בעזרת הדיאגרמה הבאה :


directory mode
NameAndPassword (class)
_editfile = text ctrl object
_editpassword = text ctrl object
_rbox = radio box object
_save = save object
Save (class)
_password (str)
_encryption (str)
_recursive (bool)
_should_exit (bool)
_dir_name (str))
main
exit program
or
(according to save information)
sending save object to
file_and_encrypted (dictionary)
{filename0: encryptedname0,
.....
filenameN: encryptednameN}
FilesAndOptions (class)
_file_and_encrypted (dictionary)
_list_items - list of decrypted file names
_save = save object
_listbox = list box object
Edit (class)
_encryption(str)
_full_path (str)
_password (str)
with to CodeFile object, giving MyOpen the object
using CodeFile open and close
CodeFile (class)
_file_name (str)
_fh (file handling)
_password (str)
_iv_read (str)
cipher -> one of the cipers
_did_final_read
._encryption
MyOpen (class)
_code_file = CodeFile object
_read_or_write (str)
MyCiper (class)
_tail (str)
_64bitstruct (Struct object)
self._encrypt (bool)
self._key(int\long)
self._iv(str)
self._extra_key(int\long)
or
(according to save information)
AesCipher (class)
_tail (str)
_128bit (int)
self._iv (str)
self._aes
(AESModeOfOperationCBC
object from pyaes)

Install

כאשר משתמש מריץ את ההתקנה, עולה ממשק המשתמש, שם הסבר על התוכנית וכפתור להתקנה. ברגע שהמשתמש לוחץ על הכפתור תוכנית ההתקנה עושה את הדברים הבאים:

- מוסיפה לPYTHONPATH דרך הregistry editor את המיקום שבו היא נמצאת על מנת שיהיה אפשר להריץ את התוכנית בעזרת המודולים של פייטון (מכיוון שהחיפוש של המודולים יהיה בתיקייה הזו). אם PYTHONPATH לא שם היא יוצרת אותו. המיקום של PYTHONPATH מצריך הפעלה מחדש של המחשב כדי שהשינויים ישמרו.
- מוסיפה לקליק ימני של כל קובץ את האפשרויות encrypt, delete גם זה דרך הregistry editor ולהן להוסיף שבלחיצה תרוץ התוכנית המתאימה עם הפרמטרים המתאימים.
- מוסיפה לקליק ימני של תיקיות את האפשרויות encrypt, decrypt דרך הregistry editor ולהן להוסיף שבלחיצה תרוץ התוכנית המתאימה עם הפרמטרים המתאימים.
- מוסיפה לקליק ימני רק עבור קבצים הנגמרים בMSecret את האפשרות decrypt דרך ה registry editor ולה להוסיף שבלחיצה תרוץ התוכנית המתאימה עם הפרמטרים המתאימים.
- מוסיפה לקליק ימני של תיקיות את האפשרות open directory mode.

Uninstall

כאשר משתמש מריץ את תוכנית הסרת ההתקנה, עולה ממשק המשתמש, השואל האם המשתמש בטוח שהוא רוצה להסיר את ההתקנה. ברגע שהמשתמש לוחץ על כן תוכנית הסרת ההתקנה עושה את הדברים הבאים:

- מוחקת מPYTHONPATH דרך הregistry editor את המיקום שבו היא נמצאת.
- מוחקת מקליק ימני של כל קובץ את האפשרויות encrypt, delete גם זה דרך הregistry editor.
- מוחקת מקליק ימני של תיקיות את האפשרויות encrypt, decrypt, open directory mode דרך הregistry editor.
- מוחקת מקליק ימני רק עבור קבצים הנגמרים בMSecret את האפשרות decrypt דרך ה registry editor.

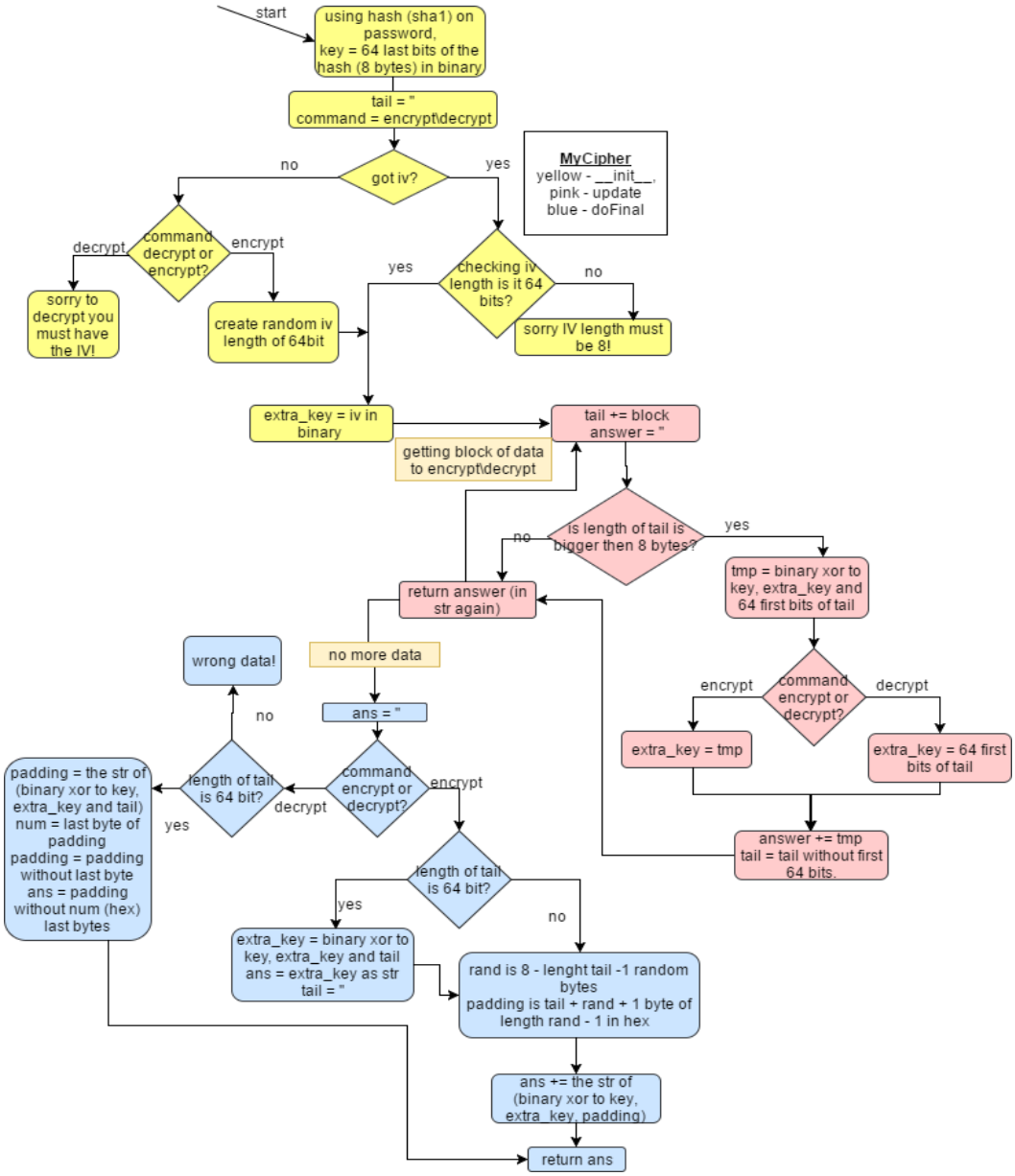
מימוש ההצפנה

שתי ההצפנות נמצאות במחלקות שונות אך באותו מבנה

- פונקצית __init__לשמירת המפתח והIV.
- פונקצית update - להצפנת בלוקים.
- ופונקציה doFinal - לטיפול בpadding.

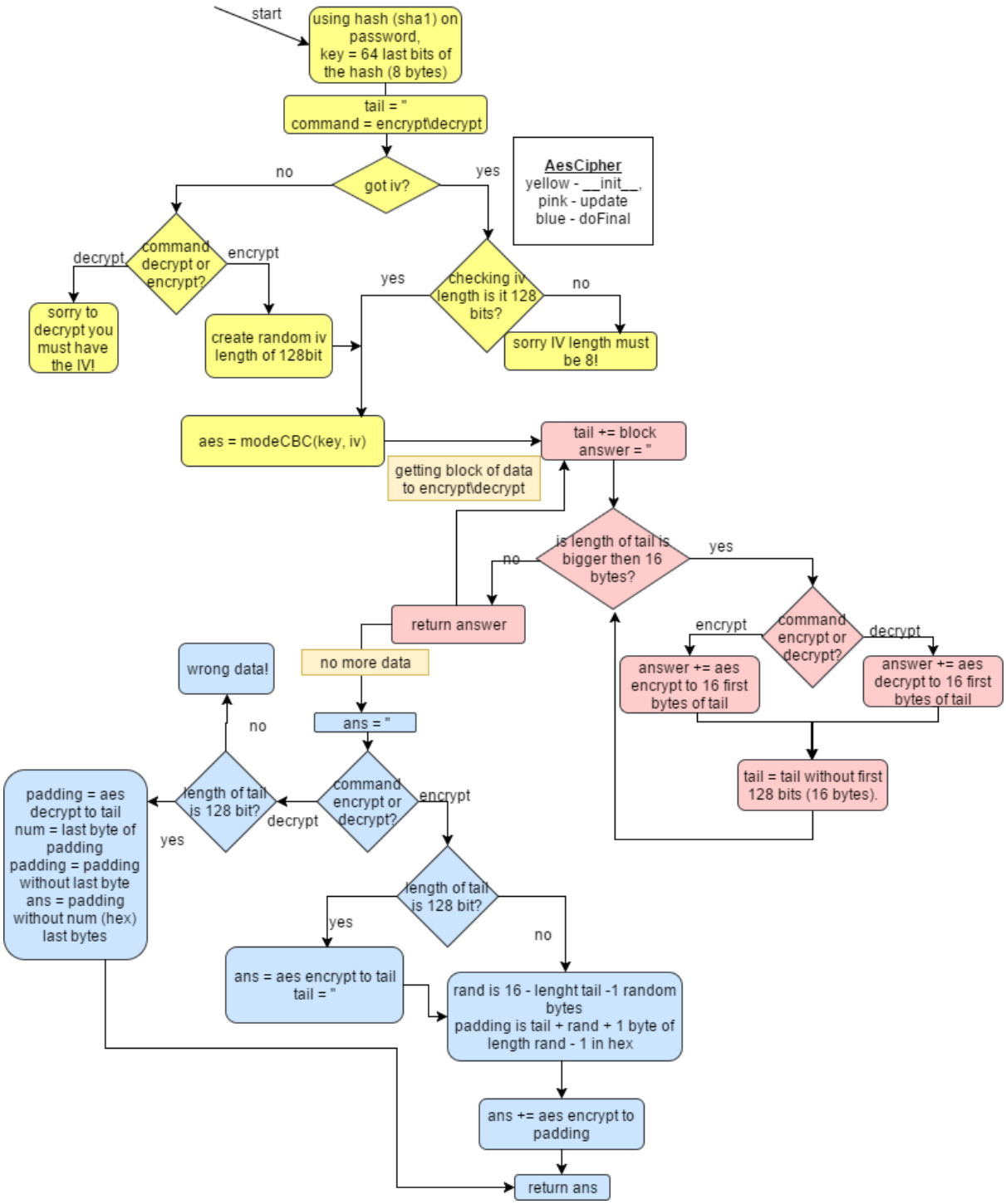
בנוסף לפונקציות אלו קיימות שתי פונקציות נוספות אחת לקבלת הIV ואורכו.

MyCipher:



המעבר בין מחרוזת למספר ולהפך בשביל xor בינארי נעשה בעזרת הסיפרייה struct בפייטון.

AesCiper:


start
using hash (sha1) on password, key = 64 last bits of the hash (8 bytes)
tail = ''
command = encrypt\decrypt
got iv?
no
yes
AesCipher
yellow - __init__,
pink - update
blue - doFinal
command decrypt or encrypt?
decrypt
encrypt
sorry to decrypt you must have the IV!
create random iv length of 128bit
checking iv length is it 128 bits?
yes
no
sorry IV length must be 8!
aes = modeCBC(key, iv)
tail += block
answer = ''
getting block of data to encrypt\decrypt
is length of tail is bigger then 16 bytes?
no
yes
return answer
no more data
wrong data!
ans = ''
command encrypt or decrypt?
encrypt
decrypt
length of tail is 128 bit?
no
yes
padding = aes decrypt to tail
num = last byte of padding
padding = padding without last byte
ans = padding without num (hex) last bytes
length of tail is 128 bit?
yes
no
ans = aes encrypt to tail
tail = ''
rand is 16 - lenght tail -1 random bytes
padding is tail + rand + 1 byte of length rand - 1 in hex
ans += aes encrypt to padding
return ans
command encrypt or decrypt?
encrypt
decrypt
answer += aes encrypt to 16 first bytes of tail
answer += aes decrypt to 16 first bytes of tail
tail = tail without first 128 bits (16 bytes).

CodeFile

אובייקט של קובץ מוצפן שדרכו ניתן לקרוא את המידע המפוענח מהקובץ ולכתוב לקובץ מידע מוצפן בקלות. התכונות של אובייקט זה הם: סיסמא, וקטור התחלתי של קריאה, צופן כתיבה, צופן קריאה, שם הקובץ, משתנה בוליאנה האומר האם עשו את הפונקציה dofinal בread והfile handling.

הפונקציה open:

הפונקציה open של אובייקט CodeFile פותחת את הקובץ על פי הפרמטר שהיא מקבלת (לקריאה או לכתיבה). אם לכתיבה הפונקציה יוצרת cipher בהתאם לסוג ההצפנה (אם כתיבה אז קיבלנו כשיצרנו אובייקט את סוג ההצפנה) וכותבת בקובץ את האורך של השם של ההצפנה (בשני בתים), את השם של ההצפנה (MSecret\AES), את האורך של הIV אותו היא מקבלת מהcipher (גם בשני בתים) וכותבת לתוך הקובץ גם את הIV. אם הפרמטר הוא לקריאה אז הפונקציה קוראת את ה2 בתים הראשונים ועל פיהם קוראת את סוג ההצפנה (ושומרת לתכונה של סוג ההצפנה) לאחר מכן היא קוראת עוד 2 בתים ועל פיהם היא קוראת את הIV וכך יוצרת cipher מתאים.

הפונקציה read:

קוראת בלוק, אם אורכו גדול מאורך של בלוק על פי ההצפנה היא מחזירה את אותו הבלוק מפוענח. אם הוא קטן מאורך בלוק הפונקציה ולא קרא בעבר לפונקצית הdoFinal קוראת לפונקציה doFinal בcipher שבה הcipher מטפל בpadding ומחזיר את הבלוק האחרון מוצפן. הפונקציה מחזירה את אותו בלוק. אם קרא בעבר אז הפונקציה מחזירה את מחרוזת ריקה.

הפונקציה write:

מקבלת בלוקים וכותבת לתוך הקובץ את הבלוקים מפוענחים

הפונקציה close:

הפונקציה בודקת האם כתבו לתוך הקובץ או קראו מתוך הקובץ ואם כתבו לתוך הקובץ הפונקציה קוראת לפונקציה הdoFinal בciper המטפלת בבלוק האחרון ומחזירה אותו ללא הבתים האקראיים שנכתבו בהצפנה. הפונקציה close כותבת את את הבתים האחרונים בקובץ וסוגרת אותו.

בזכות אובייקט זה ניתן לכתוב ולקרוא אל\מ קובץ מוצפן בקלות.

בנוסף לאובייקט זה יצרתי class נוסף בו אני משתמשת בפונקציות ה__enter__ וה__exit__ כדי לאפשר שימוש בwith על האובייקט codefile.

מבני נתונים

מבנה קובץ מוצפן:

- אורך שם האלגוריתם - שתי הבתים הראשונים של הקובץ, מספר integer (בהקסדצימלי), אך מעבירים אותו למחרוזת לצורך השרשור.
- שם האלגוריתם - טיפוס string, ישנן שתי אפשרויות - MSecret\AES.
- אורך וקטור האתחול - שתי הבתים הבאים של השרשור, מספר integer (בהקסדצימלי), אך מעבירים אותו למחרוזת לצורך השרשור.
- וקטור ההאתחול - טיפוס str.
- בלוקים מוצפנים - טיפוס str, אורך כל בלוק ואורך הIV שווים
- Padding - טיפוס str, הבלוק האחרון מוצפן אורכו כאורך שאר הבלוקים.

2 bytes - length of algorithm name (hex)	algorithm name	2 byte - length of IV (hex)	IV	Encrypted Blocks	Last Block - padding

מבנה הצפנת שם קובץ:

- אורך הצפנת שם הקובץ - 2 בתים ראשונים, מספר integer (בהקסדצימלי), אך מעבירים אותו למחרוזת לצורך השרשור. אורך זה הוא האורך של ההצפנה עד שלב הPadding, כולל.
- אורך שם האלגוריתם - 2 בתים, טיפוס integer (בהקסדצימלי), אך מעבירים אותו למחרוזת לצורך השרשור.
- שם האלגוריתם - טיפוס string, ישנן שתי אפשרויות - MSecret\AES.
- אורך וקטור האתחול - שתי הבתים הבאים של השרשור, טיפוס integer (בהקסדצימלי), אך מעבירים אותו למחרוזת לצורך השרשור.
- וקטור ההאתחול - טיפוס str.
- בלוקים מוצפנים - טיפוס str, אורך כל בלוק ואורך הIV שווים
- Padding - טיפוס str, הבלוק האחרון מוצפן אורכו כאורך שאר הבלוקים.
- השלמת השרשור לאורך קבוע כדי שכל שמות הקבצים יהיו באורך קבוע וכך לא יהיה ניתן לזהות מידע על פי שמות הקבצים

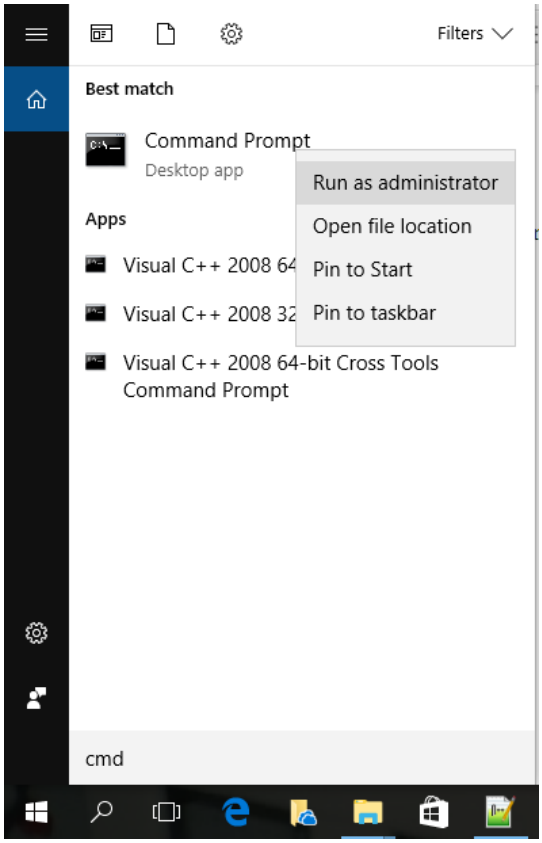
לבסוף משתמשים בסיפריית base 64 על השרשור שקיבלנו .כל שמות הקבצים המוצפנים הם בעלי אורך קבוע ומשתמש אינו יכול לראות בהם באיזה אלגוריתם השתמשנו ובאיזה IV.

מגבלות ידועות

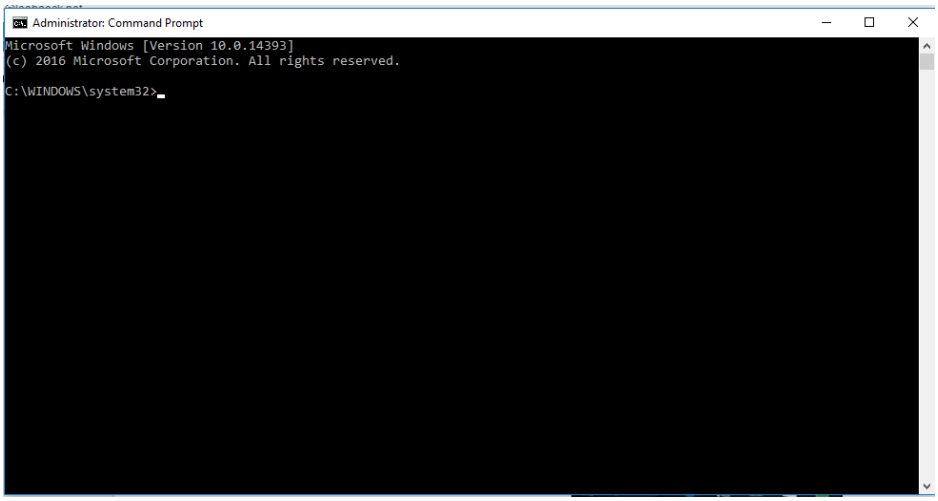
- על המחשב עליה תוכנית רצה חובה שיהיה מותקן wxpython.
- על המחשב עליה תוכנית רצה חובה שיהיה מותקן python 2.7.
- קבצי ההתקנה והסר התקנה לא יעבדו אם לא יריצו אותם כadmin (כלומר חייב להריץ אותם דרך Administor: Command Prompt).
- בפרויקט חלו שינויים על מנת לשפר אותו, חלק משינויים אלה עדיין לא מעודכנים בחלק מתיק הפרויקט. (מעבר של directory mode לכפתור ימני עבור תיקיות ובכך מחיקת המחלקה 'NameAndPassword', מחיקת המain של הdirectory mode, הוספת directory לmain של right click, שינוי ההתקנה והסר התקנה, שינוי הוראות ההפעלה).

התקנה ותפעול

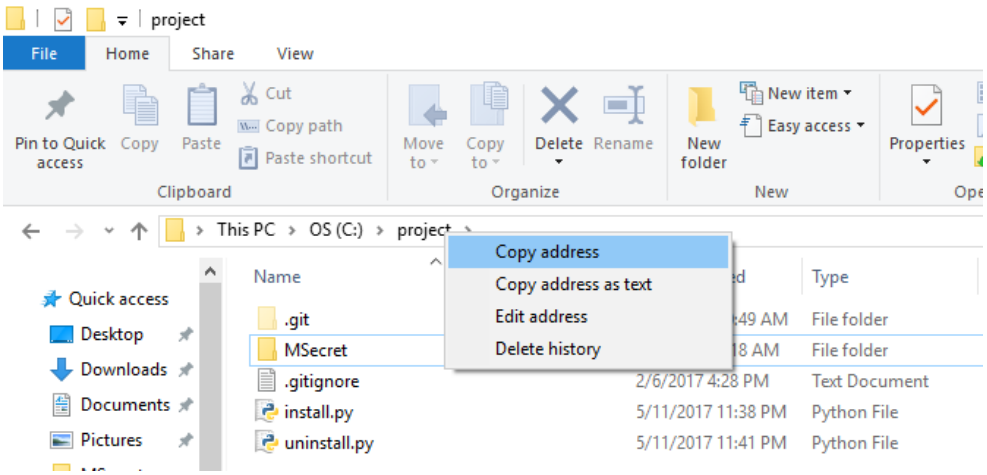
כדי להתקין את התוכנית צריך קודם כל לפתוח את Command Prompt (cmd) במצב של Administor.
יש לכתוב בחיפוש 'cmd' או 'Command Prompt'. יש ללחוץ על הכפתור הימני ואז ללחוץ על האפשרות 'Run as administor'.



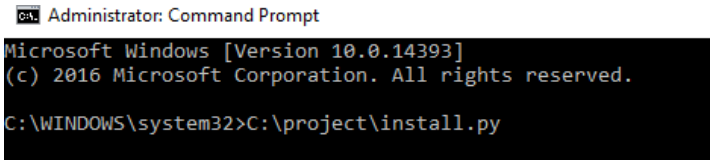
לאחר אישור אמור להיפתח החלון הזה:



לאחר שהורדתם את התיקייה project ומה שבתוכה היכנסו אליה, והעתיקו את המיקום בו אתם נמצאים על ידי כפתור ימני על הproject ולחיצה על 'Copy address'



לאחר מכן חזרו לAdministor: Command Prompt ולשם הדביקו את הכתובת שהעתקתם והוסיפו לה 'install.py\' כך שהשורה הזאת תהיה כתובה:



כאשר 'c:\project' זה הכתובת בה נמצא קובץ ההתקנה במחשב שלי, אצלכם זוהי הכתובת שהעתקתם.

לאחר מכן לוחצים על enter. חלון זה אמור להיפתח:

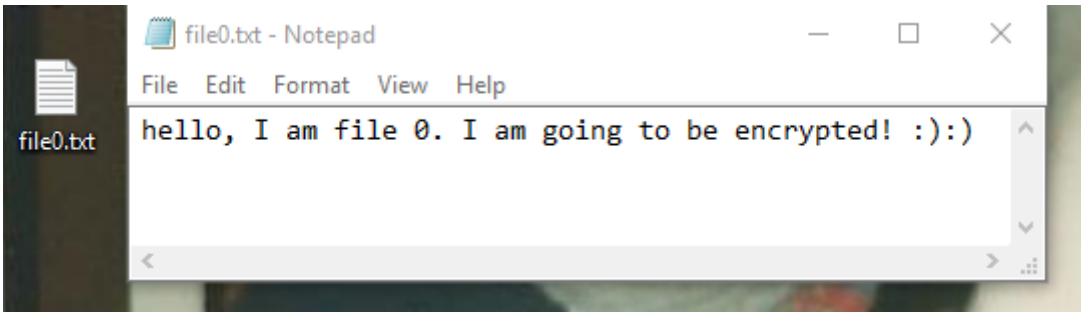
לחצו על install.

כעת בשביל שההתקנה תושלם עליך לכבות ולהדליק את המחשב.

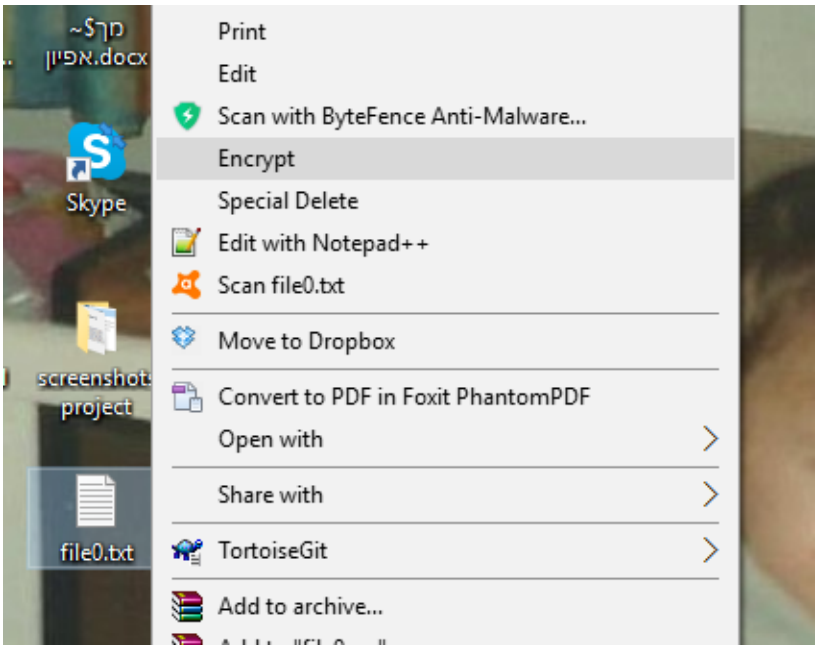
לאחר שעשיתם זאת התוכנית מותקנת אצלכם.

תפעול קליק ימני על קובץ

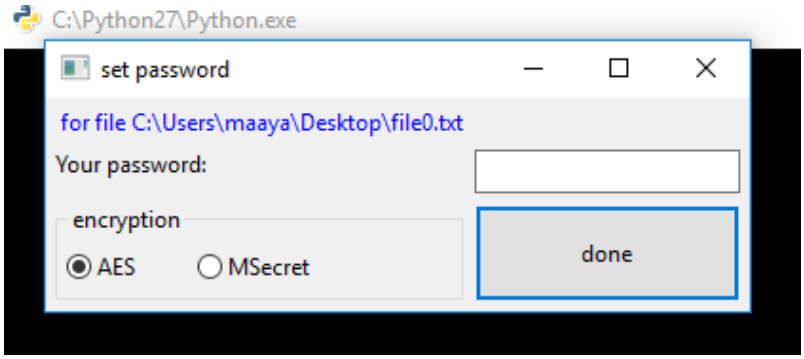
כעת בואו נצפין קובץ. תבחרו קובץ כלשהו



תלחצו על כפתור ימני ותבחרו באפשרות 'Encrypt'



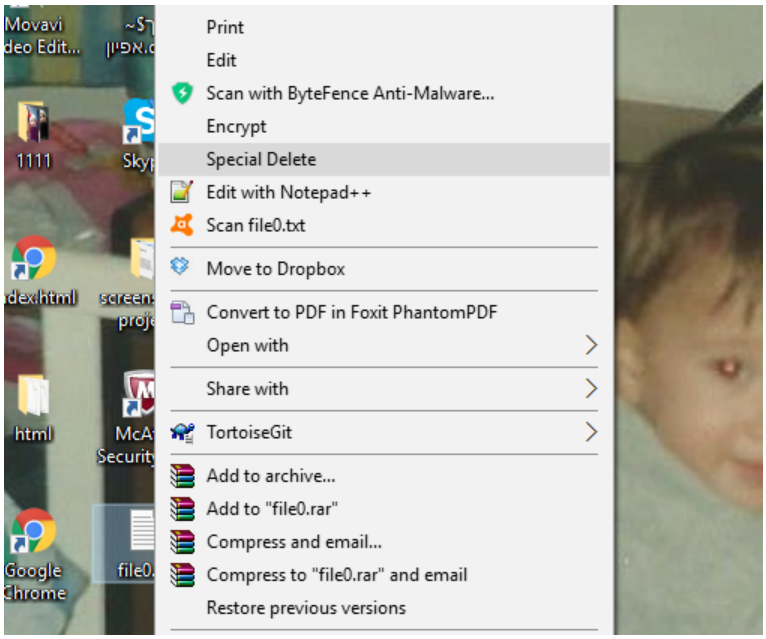
כעת יפתח החלון הבא :



בחלון זה עליכם לכתוב את הסיסמא שתרצו ולבחור את ההצפנה. כשתסיימו תלחצו על הכפתור done.

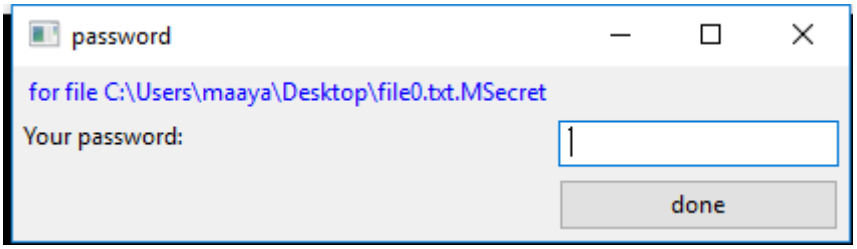
לאחר מכן תוכלו לראות קובץ חדש ששמו הוא שם הקובץ המקורי עם סיומת של MSecret.

אם ברצונכם למחוק את הקובץ אותו הצפנתם (במחיקה בטוחה) עליכם ללחוץ שוב על כפתור ימני כשאתם על הקובץ המקורי ולבחור באפשרות 'Special Delete'.



והקובץ ימחק.

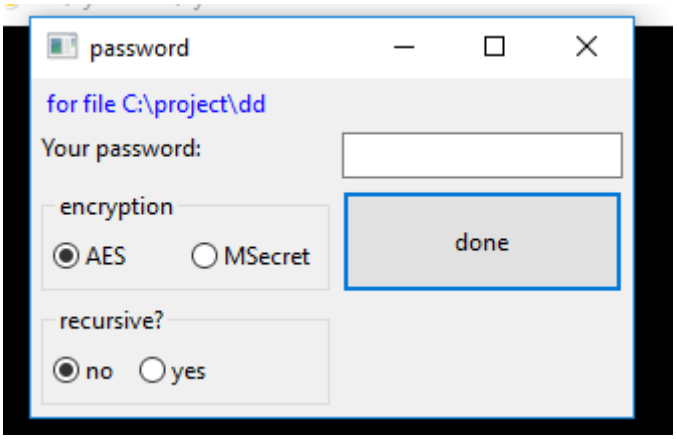
כעת אם תרצו לפענח את הקובץ אותו הצפנתם עליכם ללחוץ על הקובץ המוצפן (או ללחוץ על כפתור ימני על Decrypt) ויפתח החלון :



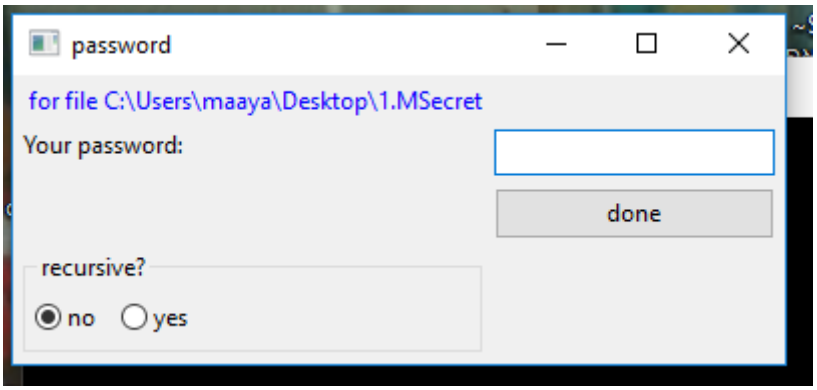
עליכם לכתוב בו את הסיסמא של הקובץ.

ואם הסיסמא היא נכונה אז נוצר לכם הקובץ המפענח.

אם תרצו להצפין תיקייה הדרך היא בכפתור ימני encrypt. בחלון שיפתח תופיע עוד אפשרות בחירה שהיא האם להצפין באופן רקורסיבי כלומר גם תיקיות שבתוך התיקיות, לעומת אופן לא רקורסיבי שבו ההצפנה היא רק על הקבצים שבתיקייה הראשונה, ללא תיקיות שבפנים.

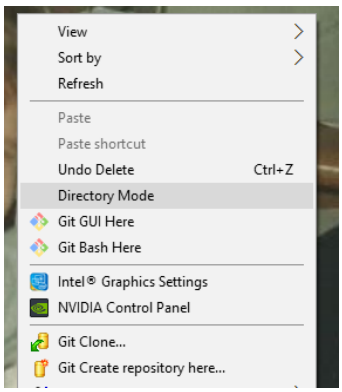


ישנה את האפשרות של פענוח גם לתיקייה, וגם כן בחירה אם רקורסיבי או לא. עבור פענוח של תיקייה יופיע החלון הבא :

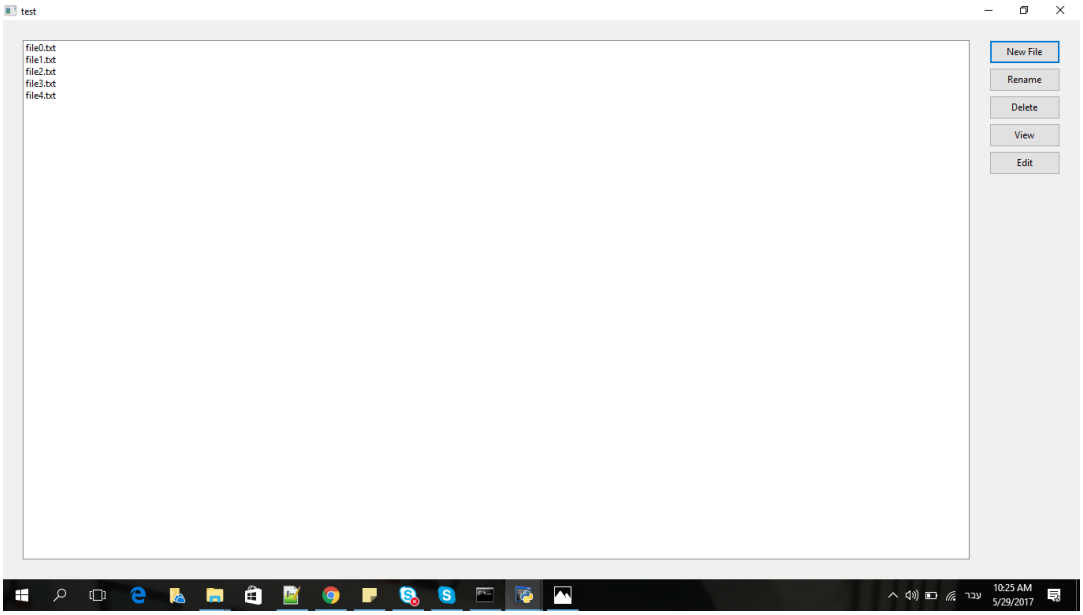


תפעול - Directory Mode

את מצב זה ניתן להריץ על ידי קליק ימני על תיקייה במחשב ולחיצה על 'Directory Mode'.

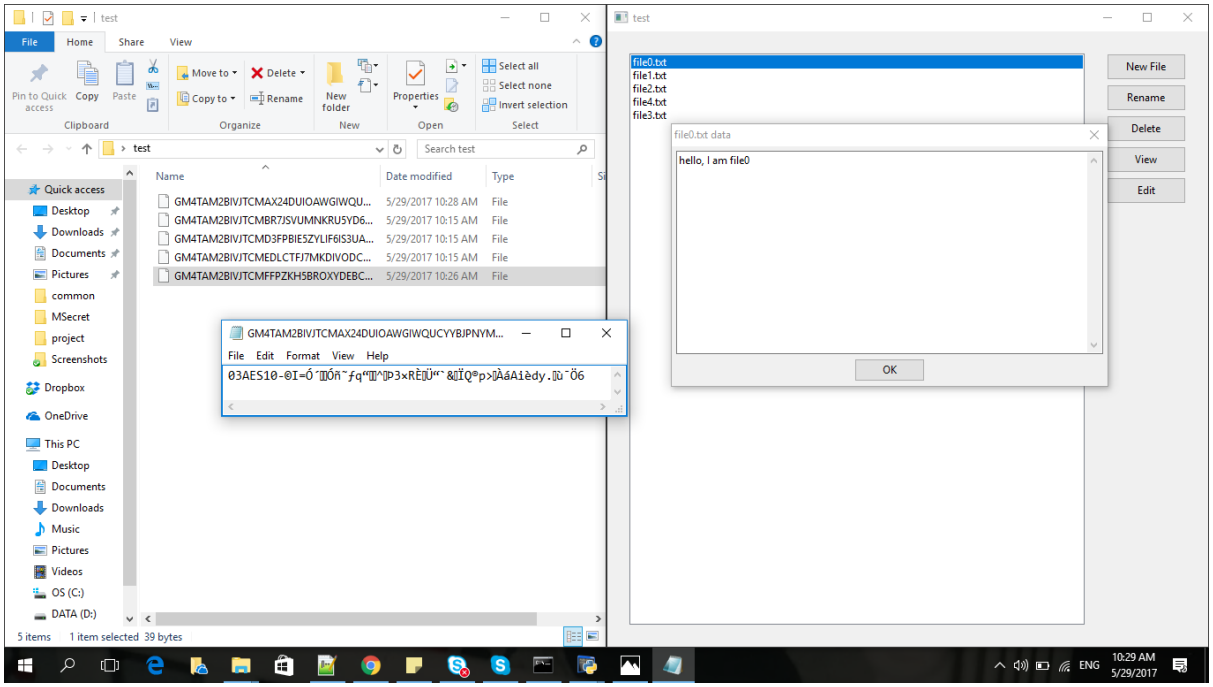


לאחר הלחיצה עולה חלון בו אתם צריכים להכניס את הסיסמא של התיקייה, לבחור הצפנה (הקבצים שאתם עורכים או יוצרים בכניסה זו יהיו מוצפנים בהצפנה זו) וללחוץ על done. אם אין לא השתמשתם במצב זה בעבר ואין לכם תיקייה ממצב זה או שאתם רוצים ליצור עוד אחת, אתם פותחים תיקייה ריקה ועבורה עושים את זה. יפתח החלון הבא :



בתוך קופסת הטקסט ישנה רשימת קבצים (של השמות המפוענחים) של הקבצים.

ניתן דרך ממשק זה ליצור קובץ חדש, לראות את המידע בקבצים, לערוך את הקבצים למחוק מחיקה בטוחה ולשנות את שמם.



תכניות לעתיד

- להוסיף תמיכה בעוד הצפנות שונות.
- להוסיף מצב בו ניתן לבחור קובץ מהממשק משתמש ומיקום, ולהצפין\לפענח את הקובץ למיקום.
- להוסיף לתיקייה מוצפנת אפשרות שלא רק קבצי טקסט יהיו בה.

פרק אישי

כתיבת הפרויקט הייתה חוויה מעניין ומשמעותית עבורי, למדתי המון דברים חדשים במהלך עשייתו. למרות שהיו לי אתגרים במהלך עשייתו כמו מימוש אלגוריתם ההצפנה, הבנה של איך להשתמש בwxpython, הבנה של איך עובד הregistry של windows, למידה על מה זה PYTHONPATH ואיך ניתן להגדיר אותו, כמו כן גם הלמידה של המידע התיאוריטי שמאחורי הפרויקט כמו איך הדיסק הקשיח עובד וכדומה. לבסוף הצלחתי להגיע לתוצר ונהניתי תוך כדי, בזכות ההתמדה והלימוד העצמי ובזכות עזרת המורים שליוו אותי לאורך כל הדרך. למדתי מהפרויקט המון נושאים שיהיו שימושיים לי גם בעתיד. תובנותיי מהפרויקט הן שהתמדה ועבודה קשה משתלמת ושלא צריך לפחד מללמוד בעצמך נושאים חדשים אלא צריך לרצות ללמוד נושאים חדשים.