פרויקט הסתרת מידע בתוך תמונה – ממ"ן 16

תוכן עניינים

[**רקע כללי – סטגנוגרפיה** 2](#_Toc76600242)

[האלגוריתם הממומש בפרויקט 3](#_Toc76600243)

[תרשים של הפרויקט 5](#_Toc76600244)

[המחלקה 7](#_Toc76600245)

[הטמעת הודעה בתמונה 7](#_Toc76600246)

[שליפת הודעה מתוך תמונה 8](#_Toc76600247)

[המחלקה 13](#_Toc76600248)

[אלגוריתם לקביעת חוזק הסיסמא 13](#_Toc76600249)

[המחלקה 14](#_Toc76600250)

[המחלקה 14](#_Toc76600251)

[המחלקה 15](#_Toc76600252)

[המחלקה 16](#_Toc76600253)

[המחלקה 16](#_Toc76600254)

[המחלקה 16](#_Toc76600255)

[מחלקות 18](#_Toc76600256)

# **רקע כללי – סטגנוגרפיה**

סטגנוגרפיה היא צירוף מילים מיוונית שמשמעותן "כתיבה בסתר", והיא האומנות של הסתרת העובדה שמתקיימת תקשורת באמצעות הסתרת מידע במידע אחר. היא נעשית על ידי הסתרת ההודעה שרוצים להעביר בתוך תמונה, קובץ אודיו, קובץ וידיאו, בטקסט או ב של פרוטוקול.

סטגנוגרפיה וקריפטוגרפיה הן דרכים שונות להגן על מידע מגורמים לא רצויים, אף אחת לא מושלמת לבדה. קריפטוגרפיה מתמקדת בהסתרת התוכן של ההודעה, בעוד ידוע שהודעה עוברת. סטגנוגרפיה מתמקדת בהסתרת העובדה שמעבירים הודעה. החוזק של סטגנוגרפיה מוגבר כשמשולב עם הצפנה.

הקבצים הדיגיטליים הטובים ביותר להעברת הודעה הם אלו המכילים מידע יתיר. יתירות יכולה להיות מוגדרת על ידי ביטים של אובייקט המספקים דיוק גדול הרבה יותר מהנדרש לשימוש ולהצגה שלו. הביטים היתירים של אובייקט הם אלו שניתן לשנות מבלי שהשינוי שלהם יכול להיות מאותר בקלות. קבצי תמונה וקול מתאימים במיוחד לדרישה זו. בהתחשב ברמה בה תמונות נפוצות באינטרנט, אין זה פלא שהן הדרך הנפוצה ביותר להעביר הודעות.

תמונה היא אוסף של מספרים שמייצרים חוזק אור שונה (וצבע שונה) באזורים שונים של התמונה. הייצוג המספרי נעשה באמצעות מטריצה בה כל תא נקרא פיקסל. עומק הביט הוא מושג המתאר כמה ביטים מרכיבים פיקסל. עומק הביט של תמונות דיגיטליות מכיל 24 ביטים: 8 עבור הרכיב האדום, 8 עבור הרכיב הירוק ו8 עבור הרכיב הכחול (פורמט ). באמצעות שילוב שלושת הרכיבים ניתן להציג 16 מיליון צבעים.

על-מנת לצמצם את זמן השליחה וההצגה של תמונות שעומק הביט שלהם הוא 24 משתמשים בטכניקות של דחיסה כך שעומק הביט של כל פיקסל יהיה 8 ביטים. לתמונה יש 2 סוגים של דחיסות: ו. דחיסה מסוג מסירה מהתמונה פרטים שהם קטנים מדי מכדי שהעין האנושית תבחין בהם ובכך הדחיסה יוצרת קירוב של התמונה המקורית. דוגמא לדחיסה מסוג זה היא .

דחיסה מסוג לא מסירה מידע מהתמונה אך מייצגת אותו באמצעות נוסחאות מתמטיות. השלמות של התמונה המקורית נשמרת והתוצאה זהה ברמת הביט לתמונה המקורית. דחיסה מסוג זה מתקיימת בפורמטים .

הטכניקות לסטגנוגרפיה בתמונות מתחלקת ל2 קטגוריות: . ב ההודעות מוסתרות בתמונה ישירות בעוד ב מעבירים את התמונה לפורמט אחר ואז מסתירים הודעה.

ב משתמשים בשיטות להסתרת המידע באזורים משמעותיים יותר של התמונה ולכן ההודעה המוסתרת יותר עמידה. טכניקות רבות בקטגוריה זו לא תלויות בפורמט של התמונה וההודעה תוכל לשרוד המרה בין דחיסות .

הפורמטים המתאימים ביותר לשימוש ב הם בדחיסת והטכניקות בהם משתמשים כוללות פעולות על ביטים , לדוגמא .

טכניקת היא דרך פשוטה לשלב הודעה סודית בתוך תמונה. הביטים שהכי פחות משפיעים על מראה התמונה הם אלו שנמצאים בסוף ייצוג הפיקסל: לדוגמא הביט במיקום ה8 בתמונה עם עומק ביט של 8. בתמונה עם עומק ביט של 24 ניתן להחליף 3 ביטים בכל פיקסל – הביט האחרון בכל ייצוג של צבע: ירוק, כחול, אדום.

הדרך הפשוטה ביותר היא למקם את ההודעה ב מהבית הראשון בתמונה עד לבית האחרון הנדרש להודעה ברצף. אולם, גישה זו קלה מאוד לזיהוי. גישה משופרת כוללת את הצפנת ההודעה ושיתוף מפתח בין השולח למקבל המכיל את המיקומים בהם מוסתרים ביטים מההודעה (כאשר המפתח משותף באופן נפרד).

קיימת דרך בטוחה יותר לעבודה המבוססת על המאמר של המציגה את הרעיון של [2]. העבודה תתבסס על מאמר זה.

## האלגוריתם הממומש בפרויקט

האלגוריתם המוצע של :

1. יצירת מפתח סימטרי.

**מימוש בפועל**: קבלת סיסמא מהמשתמש ובדיקתה לפי האלגוריתם של . לאחר מכן מימוש של האלגוריתם ליצירת מפתח סימטרי מתוך הסיסמא.

2. כיווץ ההודעה המקורית.

**מימוש בפועל**: כיוון באמצעות .

3. הצפנת ההודעה המכווצת באמצעות המפתח הסימטרי.

**מימוש בפועל**: הצפנה באמצעות אלגוריתם .

4. יצירת מפתח פסאודו-אקראי.

**מימוש בפועל**: מימוש אלגוריתם לגזירת מפתח פסיאודו אקראי מתוך . <https://en.wikipedia.org/wiki/Linear-feedback_shift_register>

5. משתמשים ב הביטים הראשונים של המפתח הסודו-אקראי כאינדקס. אם האינדקס גדול או שווה ל זורקים אותו ויוצרים חדש. שומרים מערך של האינדקסים בהם ביקרנו על-מנת לוודא שלא מבקרים באותו אינדקס פעמיים. עבור האינדקס השני ומעלה ה של הבית שנערך משמשים כ2 הביטים הראשונים של האינדקס הבא.

הטמעת המידע: במקום לבצע החלפה של ה מעלים את ערך הבית באחד או מפחיתים ב1 כדי להתאים את הערך של ה לערך המתאים מההודעה הסודית. הבחירה להגדיל או להקטין את הערך היא אקראית. שיטה זו מונעת התקפה סטטיסטית שאפשרית על השיטה הרגילה.

**מימוש בפועל**: כתיבה של הפונקציה המתוארת לעיל. הפורמט הנתמך בעבודה לתמונות .

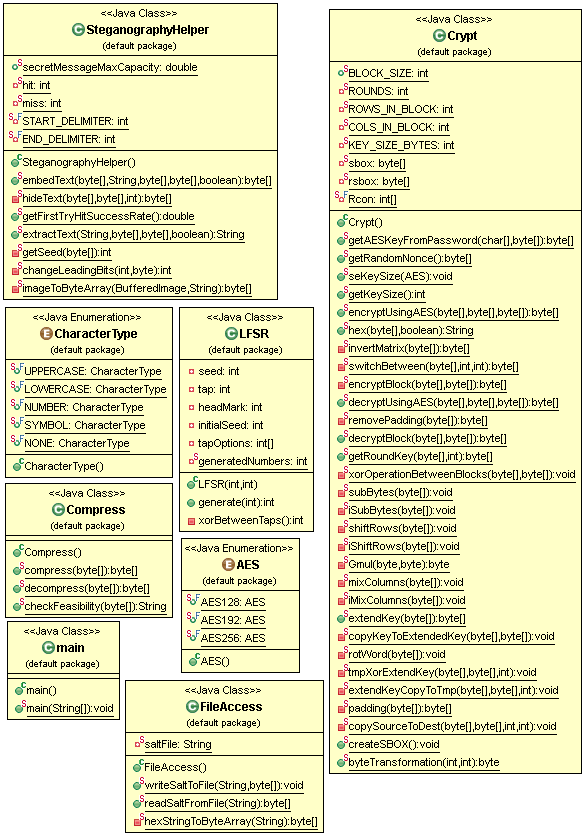
מימוש התרגיל בשפת .

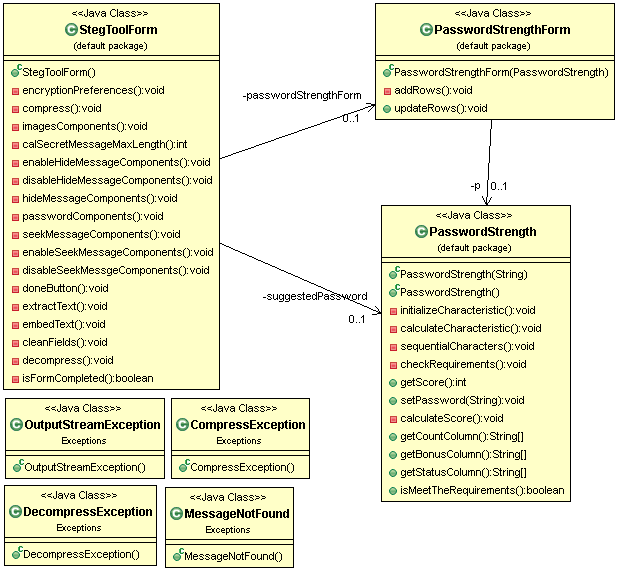
**מקורות**

[1] https://d1wqtxts1xzle7.cloudfront.net/30900669/stegoverview-with-cover-page-v2.pdf?Expires=1624296429&Signature=Je2mzo2WvaQ5Yagv4YQAuzVYCIwKSj9BcmSq-iw1V2TKOp~yduWa0REsIXaRJelgGpkVGw1gl9banrAHy4O8iMaqulL7~m01h7ekPhDifs05-zbuPJdMj4sYEOlBJ~Jc5xppYUfHSCKXtENWNnvcdL1PcG2wY2FLO-XKdFFANtZ1sG5pGVE-yY-3vdaHdY2zRPPHTc7PbFYfgIqr0fjxZPb0YPmpdf1flNB-4IgncGqLjp4kpWKMwvjeB4VcwLuortUtlVT0WwGFy7DSnTUEsd2uKem1prnpSRooeFHPyi1ePeIA0YXRCbSiedsQoK9vMoazIK9Yj3CING2S6ibqAw\_\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[2] An Implementation of Key-Based Digital Signal Steganography | Toby Sharp.

# תרשים של הפרויקט





# המחלקה

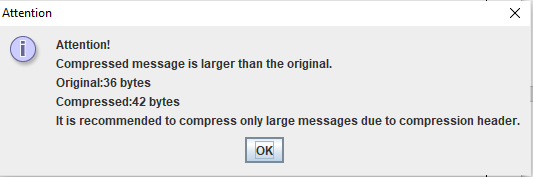
מחלקה זו יורשת מ ומטפלת בממשק הגרפי של האפליקציה.

מאפשרת ביצוע שתי פעולות עיקריות:

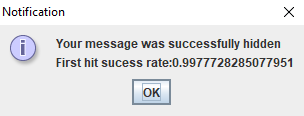
* הטמעה של הודעה סודית לתוך תמונה ושמירת התמונה במיקום רצוי.
* שליפה של הודעה סודית מתוך תמונה.

## הטמעת הודעה בתמונה

* יש ללחוץ על האפשרות .
* **טעינת תמונה** באמצעות לחיצה על הכפתור  *ובחירת תמונה ממיקום רצוי. לפרויקט צירפתי 5 תמונות בפורמט לצורך פשטות, גם תמונות אחרות בפורמט זה יעבדו.*
* ***הזנת ההודעה הסודית*** *בתיבת הטקסט מתחת לתמונה. קיימת הגבלה על מספר התווים שניתן להזין שנגזרת מתוך גודל התמונה (הסתרה של 3 ביטים בכל פיקסל). לא תתאפשר הזנה של מספר תווים גדול יותר.*
* ***הזנת סיסמא בתיבת הטקסט הייעודית****. לאחר הזנה של אות אחת יתאפשר ללחוץ על כפתור על-מנת לראות את הפרמטרים לפיהם חוזק הסיסמא נקבע. ניתן להזין סיסמא ולצפות בהערכה משתנה בחלון שיפתח במקביל. את חוזק הסיסמא בטווח שבין 0 ל100 ניתן לראות ב המופיע בסמוך לסיסמא.*
* ***יש לבחור את גודל המפתח להצפנת הטקסט*** *: .*
* ***יש לבחור את וקטור האתחול למצב***  *( אפשרתי למשתמש לבחור על-אף שמימשתי גם כלים לבחירה פסיאודו-אקראית שיכולים השולח והמקבל להשתמש).*
* ***אופציונאלי: בחירה לדחוס את ההודעה****. עבור הודעות קטנות הכותרת שמתווספת עקב הדחיסה מגדילה את ההודעה למעשה. בעת הלחיצה תוצג למשתמש הודעה שמדווחת על כמות הדחיסה. בנוסף, התגית שמציגה את מספר התווים שהוזנו משתנה בהתאם לדחיסה ותאפשר הזנה של טקסט ארוך יותר.*

**

לסיום, יש ללחוץ על הכפתור (שיתאפשר ללחיצה רק במילוי נכון של כל השדות).

אם ההטמעה בוצעה בהצלחה תופיע הודעה מתאימה. הודעות על אי הצלחה יופיעו עם הסיבה לכך במידת הצורך.

כפי שתואר במבוא, בחירת האינדקס בתמונה מתבססת על יצירת מספר פסיאודו-אקראי ולכן ייתכנו חזרות על מספרים. ההודעה מציגה את אחוז הפעמים בהם נבחר אינדקס פנוי בפעם הראשונה.

על-מנת לשמור את התמונה עם ההודעה הסודית יש ללחוץ על כפתור ולבחור את המיקום הרצוי.

## שליפת הודעה מתוך תמונה

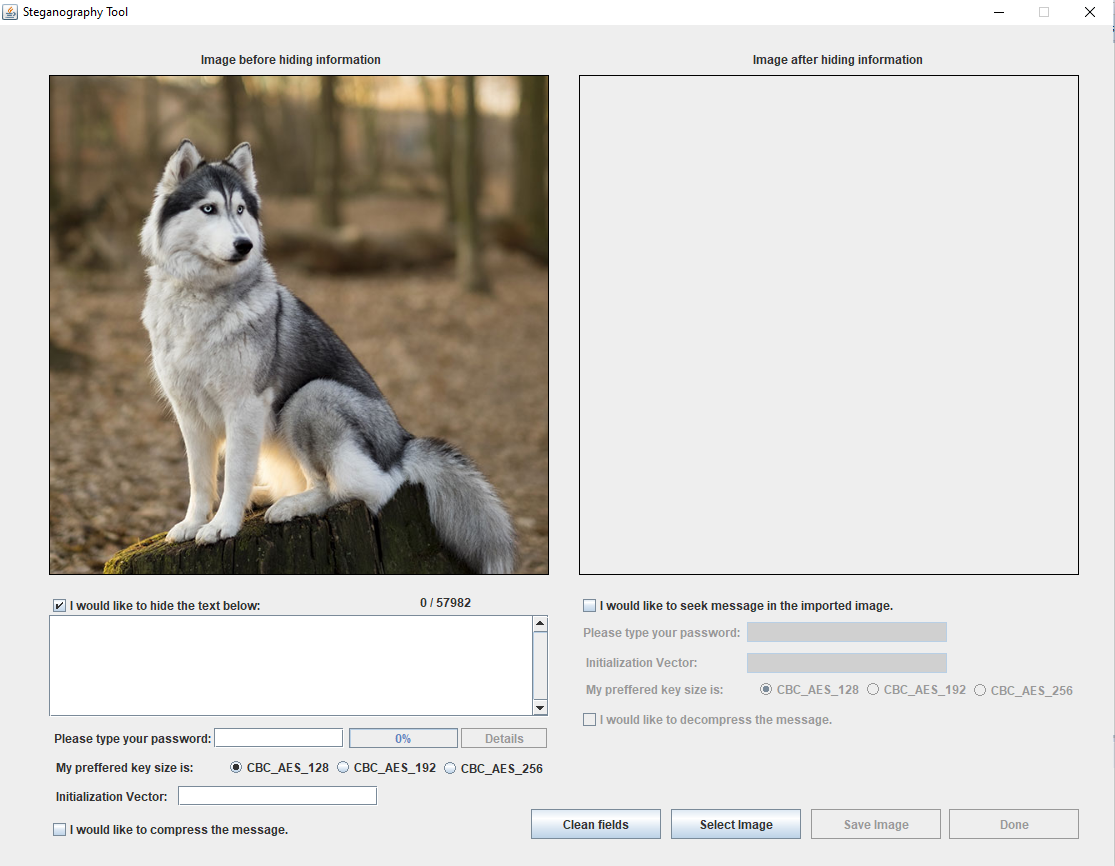
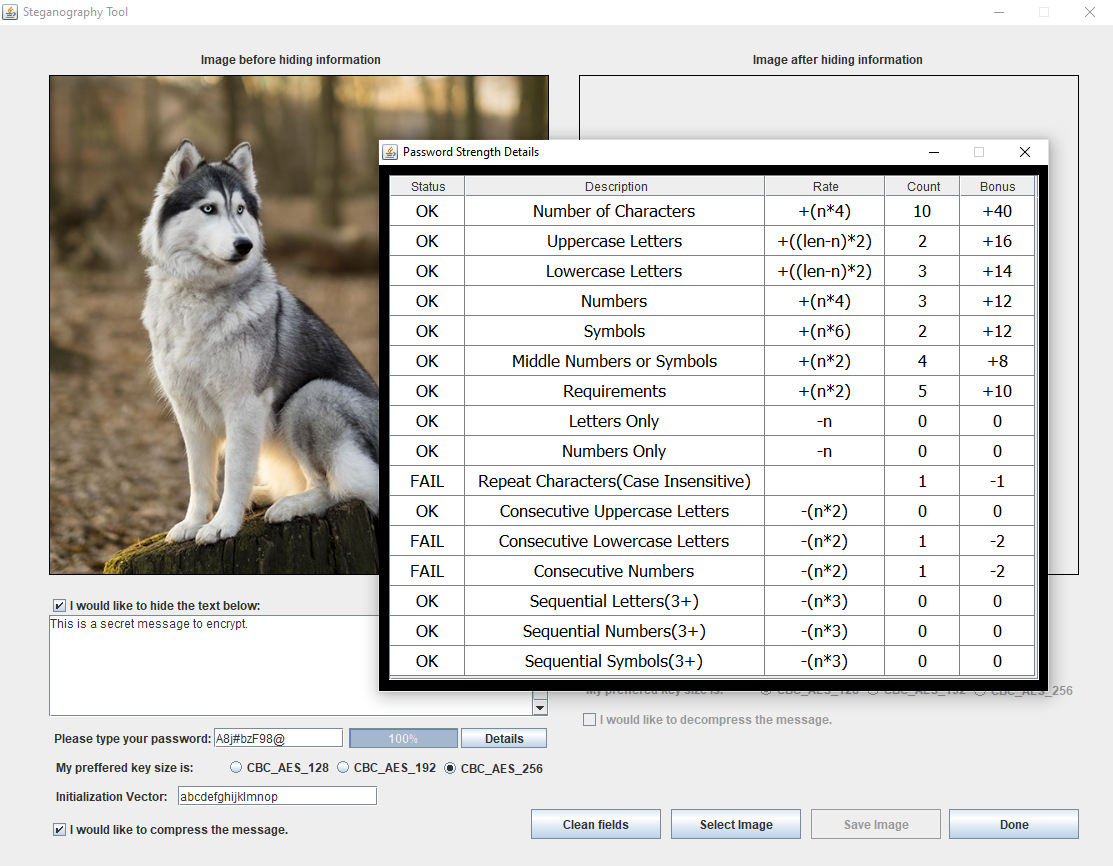
במידה ורוצים לבצע אפשרות זו לאחר הטמעת מידע יש ללחוץ על כפתור המנקה את החלון מכל הקלט שהוזן. אפשרות נוספת היא לסגור את האפליקציה ולפתוח מחדש.

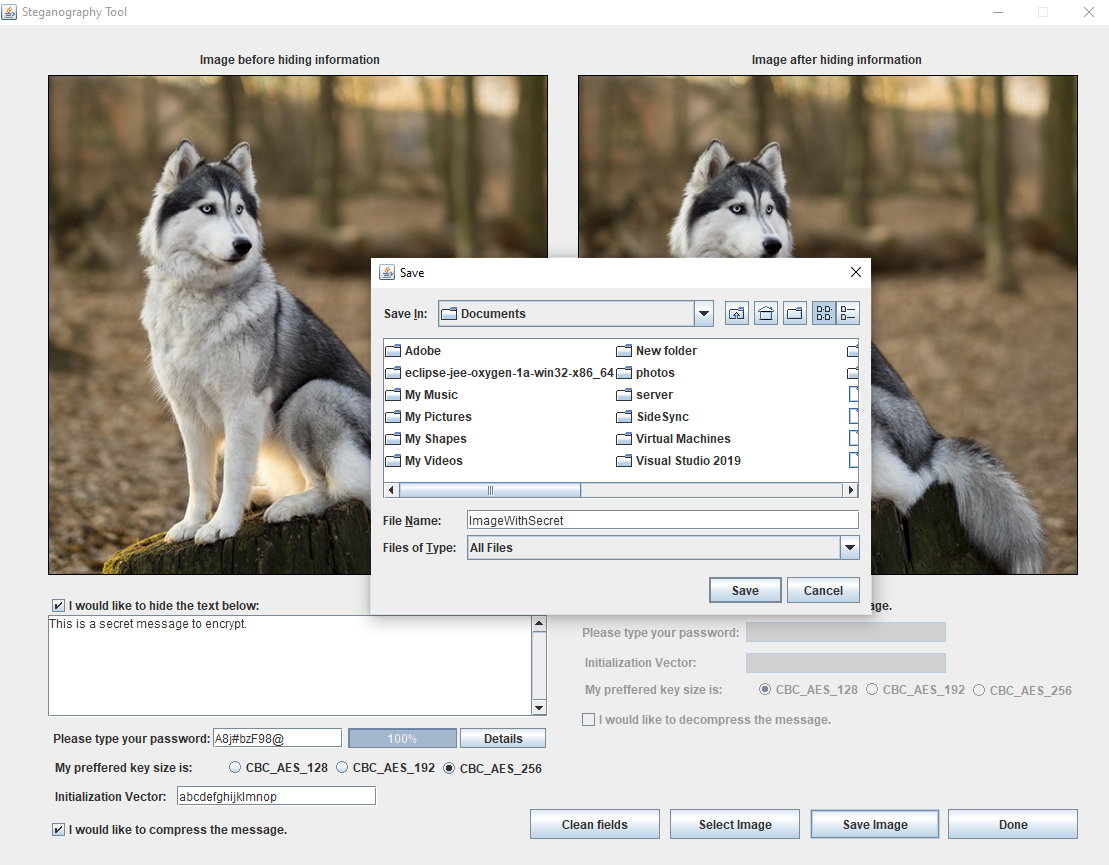
* בחירת האפשרות .
* הזנת הסיסמא בתיבה הייעודית. כאן לא ניתן לצפות בחוזק הסיסמא מתוך הבנה שהסיסמא כבר נקבעה בשלב ההטמעה.
* בחירת ווקטור האתחול (גודל 16 בתים).
* בחירת גודל המפתח של ההצפנה מבין האפשרויות: .
* בחירה האם לבצע .

בסיום, במידה והוזנו כל הפרטים יתאפשר ללחוץ על כפתור . בשלב זה ייתכנו מספר אפשרויות:

* **זיהוי הודעה** – הודעה סודית זוהתה ותוצג בהודעה קופצת.
* **אי זיהוי הודעה** – להודעה הסודית משורשר מזהה ייחודי בהתחלה ובסוף. אי-הימצאות של מזהה זה יגרום להצגת ההודעה.
* **זיהוי הודעה לא קריאה** – במידה וזוהה המזהה הייחודי אך המידע שסופק לא אפשר להציג את ההודעה באופן קריא. לדוגמא, אם ההודעה נדחסה ע"י יוצר ההודעה ולא בוצע . לחלופין אם לא נבחר וקטור אתחול נכון או גודל מפתח נכון.

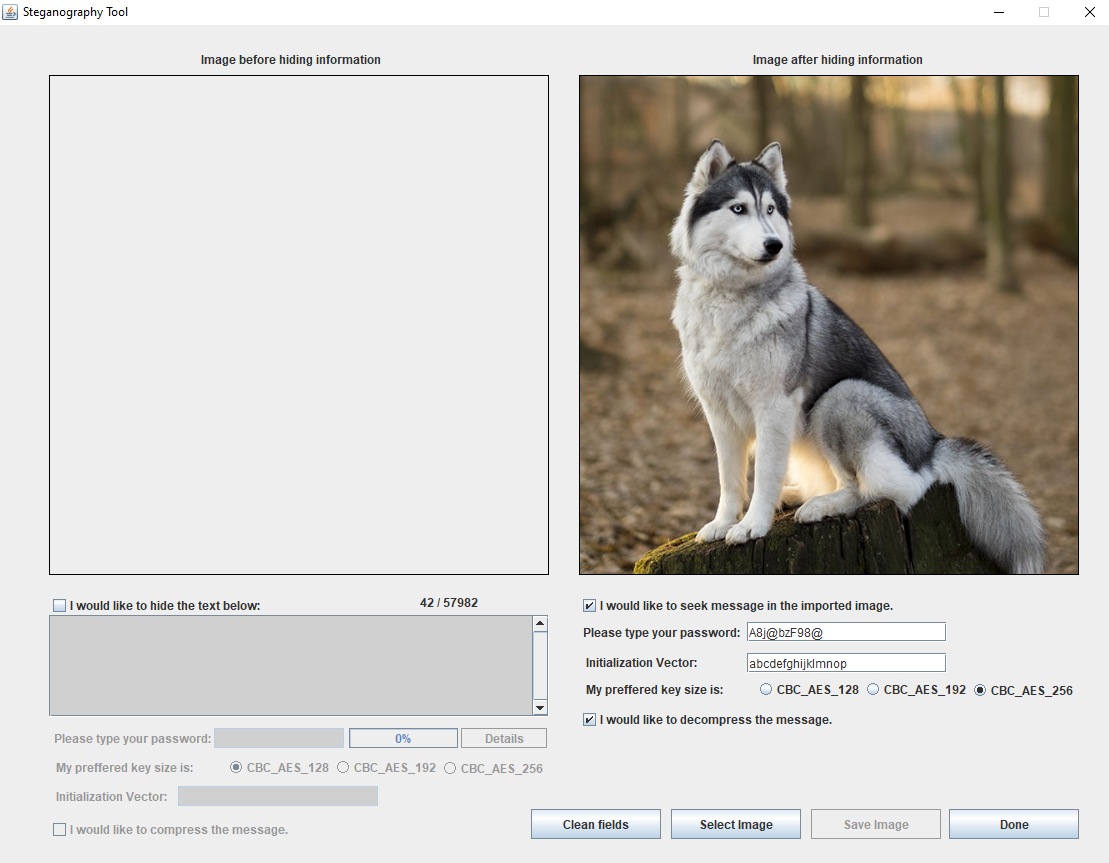
תמונות מתוך התהליך המתואר לעיל מצורפות בהמשך.



תמונה שמכילה טקסט, כלב, צילום מסך

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיהזנות שגויות בפיענוח:

# המחלקה

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטימחלקה זו מקבלת מחרוזת המכילה סיסמא ובודקת את החוזק שלה בהתאם לאלגוריתם המובא בהמשך.

המחלקה חושפת את המתודות המופיעות בתמונה המשמשות את המחלקה בכדי לקבל מידע להצגה למשתמש.

## אלגוריתם לקביעת חוזק הסיסמא

האלגוריתם מבוסס על האתר <http://www.passwordmeter.com/> .

הבדלים קיימים בתנאי 10 שלא מפורט דרך חישובו באתר.

סיסמא תאושר אם היא בת 8 תווים לפחות ועומדת ב3 מתוך 4 הדרישות הבאות:

* אות גדולה .
* אות קטנה .
* מספרים.
* סמלים.

האפליקציה מראה למשתמש מד חוזק סיסמא הנקבע לפי הקריטריונים הבאים, כאשר מסמל את מספר התווים שעומדים בתנאי ו מסמלת את אורך הסיסמא:

|  |  |  |
| --- | --- | --- |
| **מס"ד** | **תיאור** | **שינוי בניקוד** |
| 1 | מספר תווים בסיסמא |  |
| 2 | מספר אותיות גדולות |  |
| 3 | מספר אותיות קטנות |  |
| 4 | כמות הספרות בסיסמא |  |
| 5 | כמות הסמלים בסיסמא |  |
| 6 | מספרים וסמלים פנימיים בסיסמא (לא תו ראשון ולא תו אחרון) |  |
| 7 | עמידה בדרישות לעיל |  |
| 8 | מספרים בלבד |  |
| 9 | אותיות בלבד |  |
| 10 | אותיות שחוזרות על עצמן (רגיש לאותיות קטנות/גדולות) | מפחית 1 עבור כל אות שחוזרת על עצמה. לדוגמא, 5 אותיות זהות הן |
| 11 | רצפים של אותיות גדולות |  |
| 12 | רצפים של אותיות קטנות |  |
| 13 | רצפים של מספרים |  |
| 14 | אותיות עוקבות (רצף של 3 ומעלה) |  |
| 15 | מספרים עוקבים (רצף של 3 ומעלה) |  |
| 16 | סמלים עוקבים לפי מיקומם במקלדת ליד מספרים 1-9 (רצף של 3 ומעלה) |  |

כל נקודה היא אחוז במד הסיסמא. סיסמא שקיבלה מעל 100 נקודות – זה 100 אחוז. סיסמא שקיבלה 50 נקודות – 50 אחוז במדד החוזק.

# המחלקה

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטימחלקה זו יורשת מ ומהווה את החלון האחראי להצגת הפרמטרים לחוזק הסיסמא. חלון זה נפתח על ידי לחיצה על כפתור בחלון .

מופע של המחלקה קיים כשדה ב. כאשר מתבצע עדכון של הסיסמא משתמשים במתודה על-מנת לעדכן את התצוגה (הערכת הסיסמא החדשה).

# תמונה שמכילה טקסט התיאור נוצר באופן אוטומטיהמחלקה

מחלקה זו אחראית לכל נושא ההצפנה באפליקציה. לא נדרש לצור מופע של המחלקה מכיוון שהמתודות שלה סטטיות.

בכדי לוודא תוצאות הצפנה נכונות לכל אחד מהשלבים השתמשתי בכלי: <https://www.cryptool.org/en/cto/aes-step-by-step>

סקירה של מתודות מרכזיות:

**public** **static** **byte**[] getAESKeyFromPassword(**char**[] password, **byte**[] salt)

מתודה זו יוצרת מפתח מהסיסמא שהמשתמש בוחר בשילוב עם ערך שנשלח אליה. בכדי לעשות זאת נעשה שימוש באלגוריתם PBKDF2 (פונקציית גיבוב להסקת מפתח) באמצעות המחלקה הקיימת SecretKeyFactory . מבוצעות 65536 איטרציות של פונקציות הגיבוב בכדי להגדיל את חוזק המפתח המתקבל ולהאט תוקף.

**public** **static** **byte**[] getRandomNonce()

מתודה זו מגרילה מספר אקראי בגודל 16 בתים. האפליקציה עושה שימוש במתודה זו ליצירת .

**public** **static** **byte**[] encryptUsingAES(**byte**[] plainText, **byte**[] key, **byte**[] IV)

**מתודה זו מקבלת את הטקסט להצפנה, המפתח ווקטור האתחול כמערך בתים, מצפינה את ההודעה באמצעות המפתח במצב ומחזירה את מערך הבתים המוצפן. המתודה מממשת את מצב , יוצרת את טבלת הsbox , מרחיבה את המפתח, מרפדת את ההודעה המקורית, ונעזרת במתודה encryptBlock עבור הצפנה של כל אחד מהבלוקים. גודל המפתח הוא שדה פרטי במחלקה. שינויים של גודל המפתח באמצעות כפתורי ה בממשק ה משתמשים במתודות setKeySize(AES type) ו getKeySize()על-מנת לשנות את גודל המפתח ולקבל את ערכו בהתאמה. המתודה של שינוי המפתח מקבלת ערך של ה *המוגדר באופן הבא:*

*הריפוד של ההודעה המקורית נעשית לפי -* <https://www.cryptosys.net/pki/manpki/pki_paddingschemes.html>

**private** **static** **byte**[] encryptBlock(**byte**[] state, **byte**[] extendKey )

*המתודה מקבלת את מטריצת הstate – ממומשת כמערך של 16 בתים (המדמה מטריצה 4x4 כאשר התא הראשון הוא הפינה השמאלית העליונה והתא האחרון הוא הפינה הימנית התחתונה, כשהסריקה היא לפי עמודות) ואת המפתח המורחב ומצפינה את הבלוק בהתאם לגודל המפתח ומספר הסיבובים המוגדרים כשדות.*

**public** **static** **byte**[] decryptUsingAES(**byte**[] cipherText, **byte**[] key, **byte**[] IV)

מתודה זו מקבלת טקסט מוצפן, מפתח ואת ווקטור האתחול כמערך בתים, מפענחת את ההודעה באמצעות המפתח במצב , ומחזירה את מערך הבתים המפוענח. המתודה מממשת את מצב , יוצרת את טבלת הsbox , מרחיבה את המפתח, נעזרת במתודה decryptBlock עבור הפענוח של כל אחד מהבלוקים ומסירה את הריפוד להודעה המקורית.

**private** **static** **byte**[] decryptBlock(**byte**[] state, **byte**[] extendKey)

*המתודה מקבלת את מטריצת הstate – ממומשת כמערך של 16 בתים (המדמה מטריצה 4x4 כאשר התא הראשון הוא הפינה השמאלית העליונה והתא האחרון הוא הפינה הימנית התחתונה, כשהסריקה היא לפי עמודות) ואת המפתח המורחב ומפענחת את הבלוק בהתאם לגודל המפתח ומספר הסיבובים המוגדרים כשדות.*

**public** **static** **byte**[] extendKey(**byte**[] key)

מתודה זו מרחיבה את המפתח המתקבל כמערך בתים בגודל 16. גודל המערך המוחזר משתנה לפי גודל המפתח המבוקש להצפנה או לפענוח.

**public** **static** **byte**[] getRoundKey(**byte**[] extendKey, **int** round)

מתודה זו מחזירה את מפתח הסיבוב המבוקש מתוך המפתח המורחב.

# תמונה שמכילה טקסט התיאור נוצר באופן אוטומטיהמחלקה

מחלקה זו מבצעת דחיסה וביטול דחיסה של מערך בתים נתון באמצעות אלגוריתם *zlib / deflate. הדחיסה נעשית באמצעות המתודה* compress(**byte**[] data) *וביטול הדחיסה באמצעות המתודה* decompress(**byte**[] data)*.*

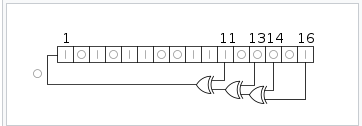
# *תמונה שמכילה טקסט התיאור נוצר באופן אוטומטי*המחלקה

*מחלקה זו מייצרת מספר פסיאודו אקראי מתוך המתקבל כקלט. האפליקציה עושה שימוש במחלקה זו בכדי להגריל אינדקס להטמעה של ביט מההודעה הסודית.*

**public** **int** generate(**int** k)

*מתודה זו מקבלת כקלט את אורך המספר הפסיאודו אקראי המבוקש. כל ביט במספר זה נוצר באמצעות סיבוב הביטים ב מקום אחד ימינה. הביט שנותר "מחוץ" למספר משמש כביט הבא במספר הפסיאודו אקראי. למקום המתפנה כתוצאה מההזזה מוכנסת סיבית המחושבת באמצעות פונקציית XOR בין מספר סיביות מתוך הseed. מיקומי הסיביות ביניהן מבוצעת הפעולה הוא קבוע וכך מייצרים את כל הביטים החדשים.*

*על-מנת לייצר רצף סיביות מקסימלי ללא חזרות משתמשים במיקומי סיביות ספציפיים במחרוזת (בהתאם לגודל הseed) שנמצא ששימוש בהן הוא אידיאלי. מיקומים אלה נקראים והם מוגדרים בצורה מוקשחת במערך המהווה שדה פרטי של המחלקה.*

*מתוך ויקיפדיה:*

# *תמונה שמכילה טקסט התיאור נוצר באופן אוטומטי*המחלקה

*מחלקה זו כותבת לקובץ את הגיבוב של מערך הבתים המכיל את התמונה עם ההודעה המוצפנת יחד עם ערך ה ששימש ליצירת המפתח.*

# *תמונה שמכילה טקסט התיאור נוצר באופן אוטומטי*המחלקה

*מחלקה זו אחראית להטמעת טקסט סודי בתמונה ולשליפת הודעה סודית מתוך תמונה, במידה וקיימת.*

**public** **static** **byte**[] embedText(**byte**[] textToHide, String imagePath, **byte**[] key, **byte**[] IV, **boolean** isCompressed)

*מתודה זו מקבלת מערך של בתים textToHide המכיל את ההודעה להסתרה, מחרוזת הכוללת את נתיב התמונה בה יש להסתיר את המידע, מערך בתים של המפתח, מערך בתים של ווקטור האתחול ומשתנה בוליאני הקובע האם לדחוס את ההודעה.*

*תחילה, ההודעה נדחסת אם נדרש. לאחר מכן מוצפנת ולבסוף מוטמעת באמצעות המתודה* hideText(**byte**[] text, **byte**[] image, **int** seed)*. מבוצע סימון של תחילת ההודעה וסיום ההודעה. הסימון של סיום ההודעה נועד לשלב הפענוח על-מנת לדעת להפסיק את הסריקה. הסימון של תחילת ההודעה נועד על-מנת לזהות בקלות שקיימת הודעה.*

**private** **static** **byte**[] hideText(**byte**[] text, **byte**[] image, **int** seed)

*מתודה זו מקבלת טקסט להטמעה כמערך של בתים, את התמונה בה תוטמע ההודעה כמערך של בתים וערך seed.*

*ההטמעה נעשית בביט האחרון של כל בית בתמונה ( מקסימום 3 בכל פיקסל). אם הביט של ההודעה הסודית והביט של הבית הנבחר להטמעה זהים לא מבוצע שינוי. אחרת, תבוצע העלאה של ערכו של הבית בתמונה או הורדת ערכו של הבית בתמונה ב1 באופן אקראי – דבר המסייע נגד התקפות סטטיסטיות הקיימות על השיטה בה מחליפים את ה בתמונה.*

*הseed המתקבל הוא 3 הבתים הראשונים מהמפתח הסימטרי ומשמש כקלט למופע של המחלקה . ייצור המספר הפסיאודו אקראי נעשה באמצעות המתודה* generate(**int** k) *. המספר הפסיאודו אקראי משמש כאינדקס להטמעה של הביט הבא מההודעה הסודית. הערך שהמתודה מקבלת הוא אורך המספר הפסיאודו אקראי שמעוניינים לקבל בביטים.*

*לפי המאמר כאשר הוא גודל התמונה בבתים. בחירה כזו עם גודל תמונה של (דוגמא המבוססת על גדלי התמונה במימוש שביצעתי) תגרום לבחירה ב כאשר מאפשר אינדקסים בין 0 למיליון. לכן, ההסתברות לקבל מספר פסיאודו אקראי גדול מגודל התמונה הוא גדול. במקרה זה יש לפנות ל להקצאת מספר חדש ובכך מגדיל את זמן ההמתנה להטמעת ההודעה הסודית. על-מנת לשפר את ה"פגיעה" בפעם הראשונה באינדקס מתוך טווח גודל התמונה השתמשתי בפועל ב . כלומר, והאינדקס שיבחר הוא מהטווח 0 עד . הפשרה היא שהביטים המהווים את "סוף" התמונה לא יאוכלסו בהודעה הסודית ולכן תתאפשר הודעה סודית קטנה יותר.*

*באלגוריתם במאמר מומלץ לשנות את 2 הביטים הראשונים של המספר הפסיאודו- אקראי המתקבל מ לשני הביטים שהם ה וה של הבית האחרון מהתמונה בו בוצעה הטמעה. כאשר מימשתי באופן הזה גיליתי שכמות ההצלחות למצוא אינדקס פנוי בפעם הראשונה פוחתת. הסיבה היא שהתדירות של הצירוף 00 לשני הביטים גדולה מהצירופים האחרים וגדלה ככל שעובר הזמן – דבר המביא לכך שלא ניגשים לאינדקסים הגבוהים בתמונה. לכן, החלטתי להשתמש ב על-מנת שהבחירה בכל אחד מ4 האפשרויות תהיה בהסתברות זהה. שינוי זה שיפר את אחוז ההצלחה למציאת אינדקס פנוי בניסיון הראשון.*

*אולם, את התוצאות הטובות ביותר השגתי כאשר השתמשתי במספר הפסיאודו-אקראי ללא כל שינוי.*

*תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיהשארתי את השינוי של 2 הביטים הראשונים כהערה לצורך שלמות העבודה בהתאם למאמר בשגרות של ההטמעה והשליפה.*

**public** **static** String extractText(String imagePath, **byte**[] key, **byte**[] IV, **boolean** isCompressed)

*המתודה מקבלת מחרוזת של מיקום התמונה הנבחרת, את המפתח הסימטרי כמערך בתים, את וקטור האתחול כמערך בתים ומשתנה בוליאני המציין האם קיימת דחיסה על ההודעה המוסתרת. המתודה משתמשת ב3 הבתים הראשונים של המפתח כseed ל ובכך מקבלת את אותם האינדקסים בהם הוטמעה ההודעה הסודית. אם ארבעת הבתים הראשונים בהודעה הם - זוהתה הודעה סודית ולכן ממשיכה להתבצע קריאת ההודעה. במידה ולא נמצא הצירוף לעיל, נזרקת השגיאה ש"נתפסת" ע"י הפונקציה הקוראת מ ומציגה למשתמש הודעה מתאימה.*

*סיום הרכבת ההודעה הסודית מתרחש כאשר מזוהה המחרוזת . בשלב זה מבצעים פענוח של ההצפנה באמצעות המפתח ולאחר מכן ביטול הדחיסה במידה והתקבלה אינדיקציה לכך שההודעה המקורית דחוסה. מערך הבתים מקודד להודעה באמצעות שימוש ב. הודעה זו מוחזרת לקורא.*

# מחלקות

על-מנת שהמשתמש יקבל הודעות ספציפיות על אירועים צפויים ולא צפויים שהתרחשו בתהליך יצרתי מספר מחלקות חריגה כדי שאוכל לזרוק חריגה בשם ספציפי ולטפל בפונקציה הקוראת בהודעה רלוונטית למקרה הספציפי.

המחלקה CompressException מייצגת שגיאה שהתרחשה בשלב הדחיסה.

המחלקה DecompressException מייצגת שגיאה שהתרחשה בשלב ביטול הדחיסה.

המחלקה MessageNotFound היא חריגה בה אני משתמש כאשר לא נמצא הסיפא המזהה את ההודעה הסודית.

המחלקה OutputStreamExcpetion היא חריגה המציינת שהתרחשה שגיאה ב.