



פרויקט גמר

הנדסאי

הנדסת-תוכנה

בהתמחות: מחשבים

נושא: תוכנת מסרים המצפינה ומפענחת מסרים בתוך תמונה
בשיטת RSA.

מגיש: זיו בן סימון

תעודת זהות: [REDACTED]

מנחים: מוטי פניקשווילי ואודי מלכה

2020

התש"פ

שנה"ל

תוכן העניינים:

2	מטרת התוכנה
3	תקציר
3	ציוד חומרה ותוכנה הנדרשים לביצוע הפרויקט
4	תהליכים עיקריים
4	דרישות מערכת
5	שלבים בפיתוח הפרויקט
5-8	רקע תאורטי
9-17	הפעלת התוכנה
18	תרשים UML
19-30	פירוט מחלקות
31-101	נספח א' - קודי מחלקות
102	נספח ב' - ביבליוגרפיה

מטרת התוכנה:

מטרת התוכנה היא הצפנה של מסר סודי בתוך תמונה ופענוח מסר מתוך תמונה, התוכנה גם תכלול צ'אט עם חיבור מידי (על המשתמש יהיה להכניס כינוי בכדי להיכנס לצ'אט) והוא ישמש לתקשורת בין משתמשי התוכנה ובנוסף בתכונה יש אפשרות לשליחת אימייל עם קבצים\מסמכים.

בפרויקט זה אשתמש בטכניקה הנקראת סטגנוגרפיה לשם הצפנת המסר בתמונה. סטגנוגרפיה היא האמנות והמדע של הסתרת מסרים באופן שאף אחד מלבד המקבל לא יוכל לראות אותם או לדעת על קיומם. בדרך כלל מסר סטגנוגרפי נראה על פניו כמשהו תמים אחר, כגון תמונה, קטע עיתונות, רשימת קניות או כל דבר אחר שאינו מעורר חשד, המשמש ככיסוי למסר האמיתי.

המסר הסודי יוצפן יחד עם מפתחות הצפנה (מפתח ציבורי ומפתח פרטי) בהצפנת RSA.

RSA היא שיטת הצפנה אסימטרית דטרמיניסטית, האלגוריתם מבוסס על רעיון המפתח הפומבי ומסתמך על בעיית פירוק לגורמים. בהצפנה זו, בניגוד להצפנה סימטרית, נעשה שימוש בשני מפתחות, האחד נקרא מפתח פרטי איתו מצפינים את המסר. בתוכנה זו המסר יהיה מוצפן בשיטת RSA בתוך כמה מהביטים של התמונה אשר תהווה כמפתח הפרטי. לא ניתן לשחזר את המסר המוצפן באמצעות שימוש במפתח זה. המפתח השני נקרא מפתח ציבורי שעליו להישמר בסוד בידי המקבל ורק באמצעותו ניתן לפענח את המסר המוצפן.

תקציר:

התוכנה כוללת בתוכה צ'אט, שרת ולקוח, ומטרתה היא לאפשר למשתמשים לתקשר ביניהם בעזרת הרשת, ישנה אפשרות לשליחת אימייל ישר מתוך התוכנה ובנוסף לאפשר הצפנת ופענוח מסר סודי שנכתב על ידי משתמש במהלך השימוש בתוכנה.

בעת הפעלה, המשתמש יוכל לבחור לעצמו כינוי אשר יוצג בצ'אט, וכשירצה יוכל המשתמש להצפין מסר סודי בתוך תמונה שיבחר ולשלוח למשתמש אחר (דרך האימייל המובנה בתוכנה או בדרך שליחה חיצונית), או לפענח צופן סודי מתוך תמונה שנשלחה אליו ממשתמש אחר.

בתהליך ההצפנה המשתמש יקליד את המסר אותו ירצה להצפין בתמונה שיבחר, תוך כדי התוכנה תצור מפתח ציבורי ופרטי לשם אבטחה שישמשו את תהליך הפענוח של המסר.

בתהליך הפענוח המשתמש יבחר את התמונה בה המסר המוצפן ואת המפתח הציבורי של המשתמש ששלח לו את התמונה (תפקידו של המפתח הציבורי נועד לשם אבטחה, על מנת לוודא שאכן המסר המוצפן הגיע מהמשתמש ששלח לו את התמונה ולא נפגע ושונה על ידי מישהו אחר).

ציוד חומרה ותוכנה הנדרשים לביצוע הפרויקט:

עמדת פיתוח:

- סביבת פיתוח Eclipse (4.15.0)
- גרסה [11.0.7] JavaSE
- ווינדוס 10

עמדת משתמש:

- מחשב
- עכבר
- מערכת הפעלה ווינדוס 10

תהליכים עיקריים:

התכנה תאפשר למשתמש את הפעולות העיקריות הבאות:

- התחברות לצ'אט התוכנה המאפשרת תקשורת בין המשתמשים.
- אפשרות הצפנה של מסר סודי בתוך תמונה שהמשתמש יבחר בה, תוך יצירה אוטומטית של מפתחות הצפנה (או יצירה של מפתחות חדשים התקבע על ידי המשתמש בעזרת התפריט).
- אפשרות פענוח של מסר סודי בתמונה, באמצעות התמונה שבה מוצפן המסר ומפתח ציבורי של המשתמש שממנו הגיעה התמונה.
- אפשרות להתחבר לחשבון ה-gmail ולשלוח אימייל ישר מתוך התוכנה עם האפשרות לצרף לאימייל קבצים\מסמכים\תמונות.

דרישות מהמערכת:

- יצירה של מפתחות הצפנה (מפתח ציבורי ומפתח פרטי) לשם אבטחה, על מנת לוודא כי אכן זהו האדם שאליו שייכת התמונה בעלת המסר הסודי
- הפיכת התמונה, המסר והחתימה (ההצפנה עם המפתח הפרטי) למערך בתיים, והצפנת המסר הסודי והחתימה בתוך הפיקסלים של התמונה.
- בדיקת חיבור סוקטים (משתמשים) חדשים וסגירתם בעת הצורך (כאשר אבד החיבור או כאשר המשתמש סוגר את הצ'אט)
- פענוח המסר המוסתר בתוך התמונה על ידי הפיכת התמונה בעלת המסר המוצפן למערך בתיים, וקבלת אישור שאכן המפתח הציבורי תואם למפתח הפרטי.
- בניית ממשק נוח למשתמש עבור צ'אט התוכנה, עבור שליחת האימייל וחלון ההצפנה או פענוח.

שלבים בפיתוח הפרויקט:

1/12/19 - 20/12/19 - תכנון מבנה נתונים.
21/12/19 - 31/1/20 - בנייה ופיתוח אלגוריתם להצפנה.
1/2/20 - 15/2/20 - פיתוח הצ'אט והאימייל.
16/2/20 - 25/3/20 - פיתוח ממשק משתמש.
26/3/20 - 26/4/20 - בדיקת תקינות התכנה וכתיבת חוברת עבודה.
20/5/20 - 27/4/20 - עיצוב התוכנה

רקע תאורטי:

אלגוריתם ההצפנה הוא LSB (Least Significant Bit), כלומר, אנו מחליפים את הביט הכי פחות משמעותי, וזה משום שנרצה לגרום לשינוי המזערי ביותר בתמונה בעת ההצפנה ולכן המקום הכי מתאים לשינוי זה הוא הביט הכי פחות משמעותי (הביט הכי ימני).

דוגמא:

סיבית ה-LSB

1 0 0 1 0 0 1 1

אם נתייחס לצבע הכחול שיוצגו ב-RBG למשל (230, 11, 246) כאשר נשנה את הביט הכי פחות משמעותי (במקרה זה הספרה 6), לא נוכל בעזרת העין האנושית להבחין בהבדל השוני וזאת משום שהשינוי כל כך מזערי שלעין האנושית אין יכולת להבדיל בין גווני הצבע שאותו שינינו במספר כה מזערי, ומכיוון שהשינוי מזערי ביותר לא יהיה ניתן לראות הבדל בתמונה ששונתה עקב ההצפנה.

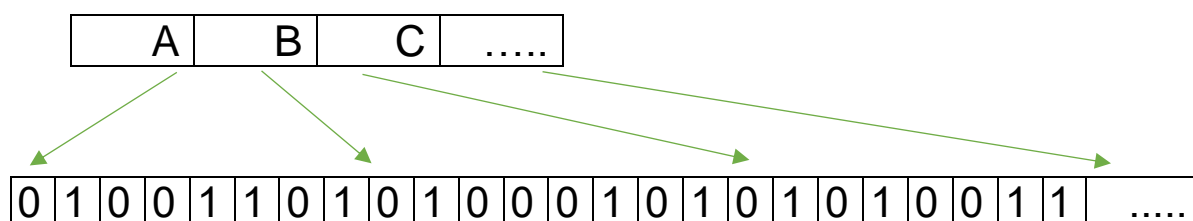
בדפים הבאים אסביר את הרעיון של הביט הכי פחות משמעותי (LSB), יש לשים לב שבמערכי ה-"Bytes Array" הייצוג הוא בבתים (Bytes) ולא בבינארי (ביטים), התצוגה הבינארית נועדה לשם המחשה והבנה של תהליכי ההצפנה והפענוח בלבד.

איור 1 מערך התווים

במערך העליון מוצג המסר בתווים (Char), כאשר כל תא הוא בגודל בית.

במערך התחתון מוצג המסר בבתים (ייצוג בינארי), כאשר כל 8 תאים הם בית.

מערך התווים:

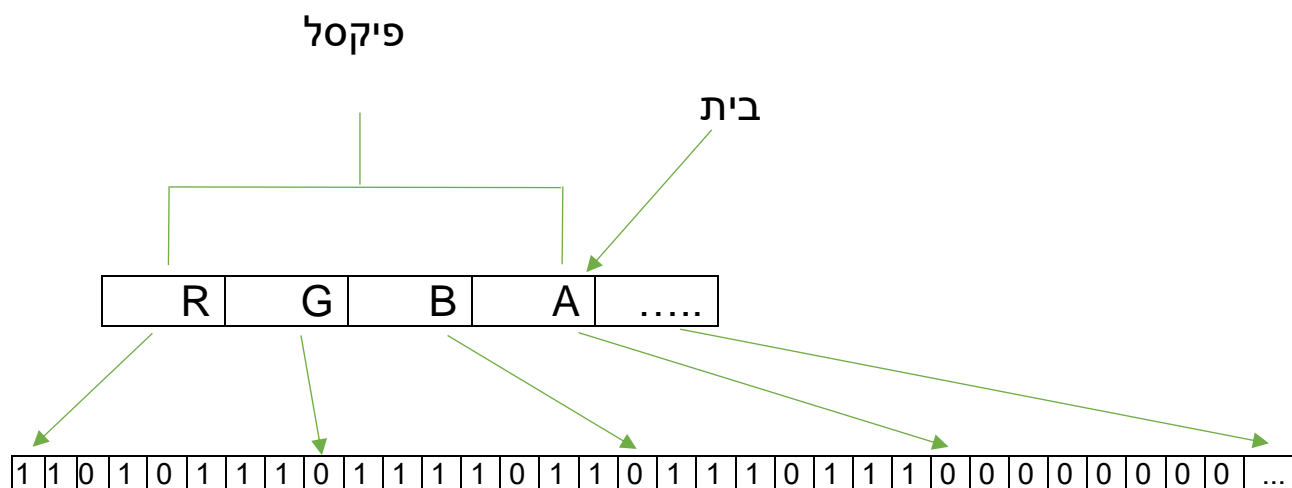


איור 2 מערך הפיקסלים

במערך העליון מוצגים הפיקסלים של תמונת המקור כמספרים שלמים, כאשר כל תא מייצג צבע למעט תא A וכל תא הוא בגודל בית (כל 4 תאים הם פיקסל).

במערך התחתון מיוצגים הפיקסלים של התמונה בבתים, כאשר כל 8 תאים הם בית.

מערך הפיקסלים:



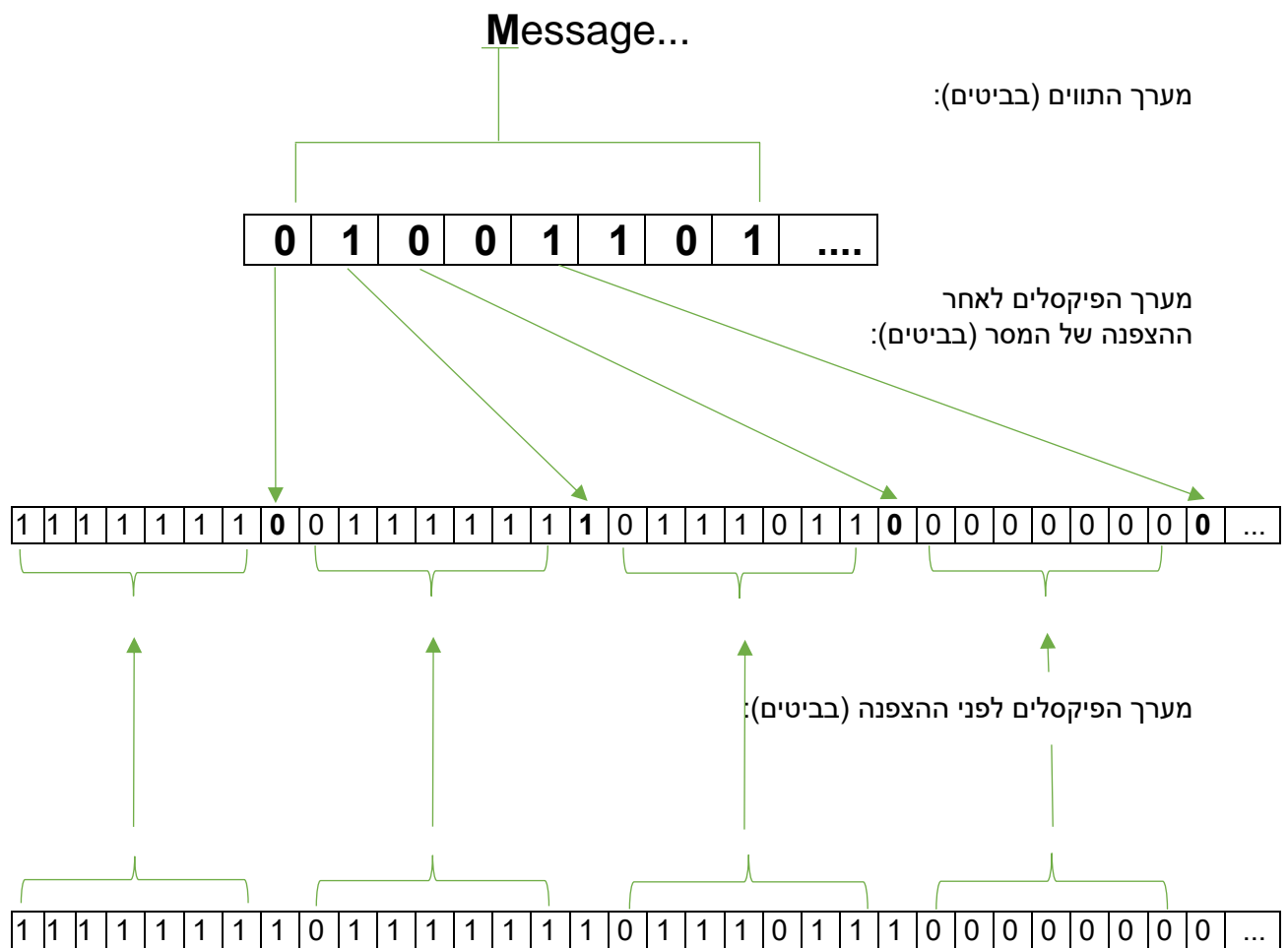
איור 3 אלגוריתם הצפנה

במערך העליון מוצג המסר בבתים.

במערך התחתון מוצגים הפיקסלים של תמונת המקור בבתים.

במערך האמצעי מוצגים הפיקסלים החדשים שנוצרו לאחר ההצפנה בבתים.

המסר:



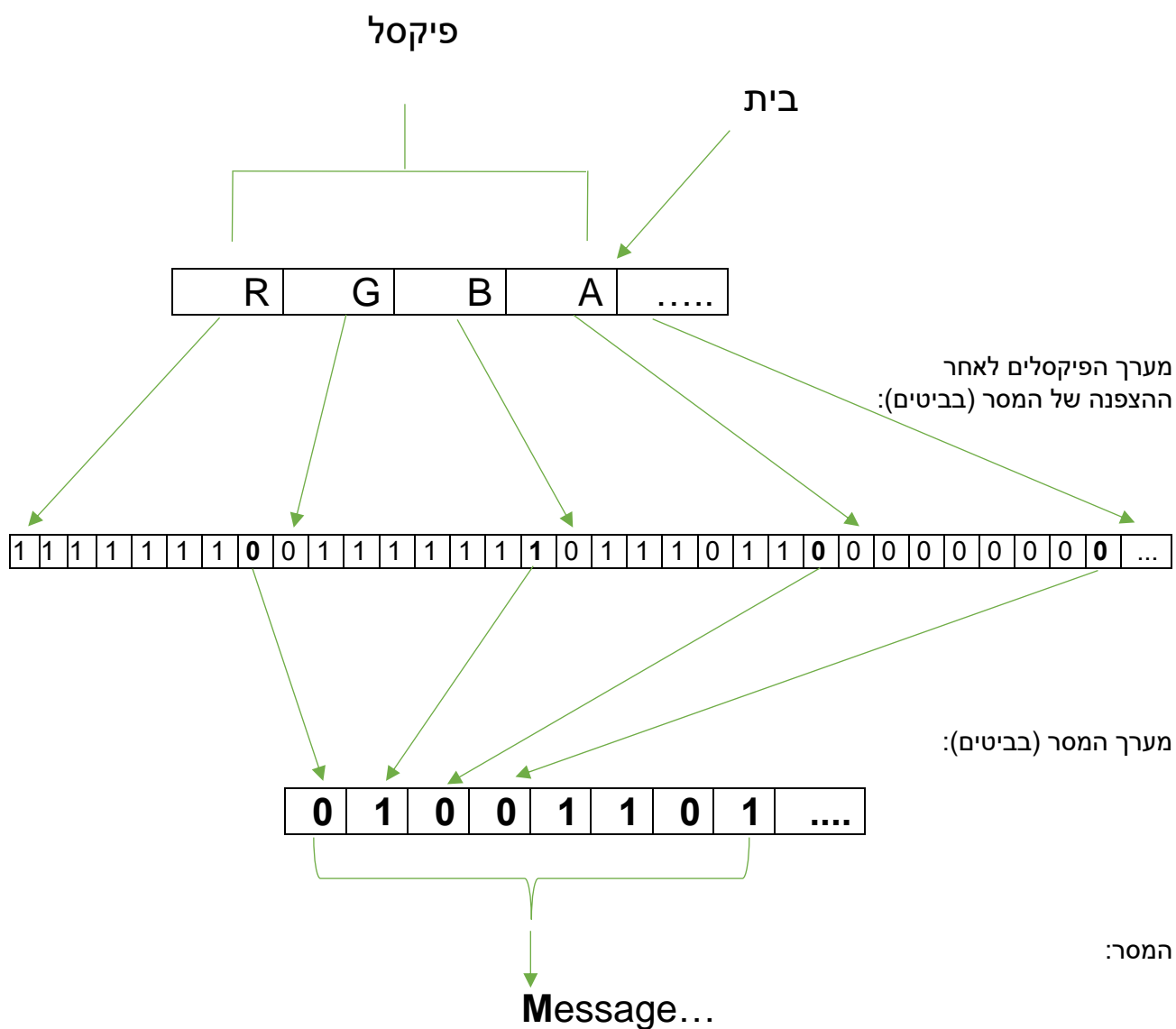
איור 4 אלגוריתם פענוח

במערך העליון מוצגים הפיקסלים החדשים שנוצרו לאחר ההצפנה, כאשר כל תא בגודל בית.

במערך האמצעי מוצג הפיקסלים עם המסר המוצפן, כאשר כל תא הוא בגודל בית.

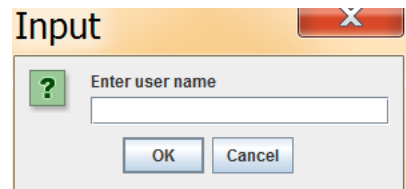
במערך התחתון מוצג המסר בבתים.

מערך הפיקסלים:

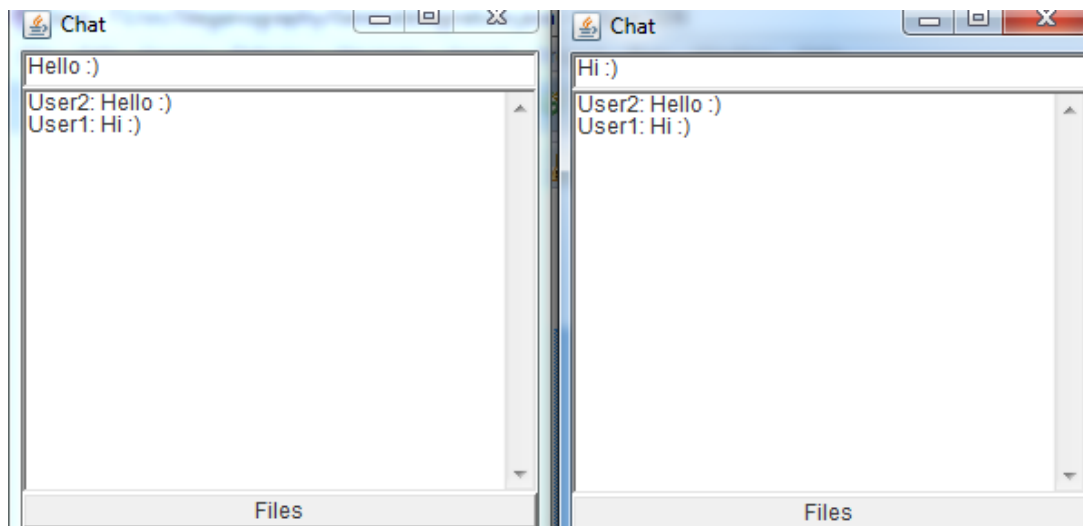


הפעלת התוכנה:

בעת הפעלת התוכנה יופיע חלון בו המשתמש יוכל להכניס את כינוי שיוצג בצ'אט.



לאחר מכאן חלון הצ'אט יפתח ויהיה ניתן להתכתב בצ'אט.

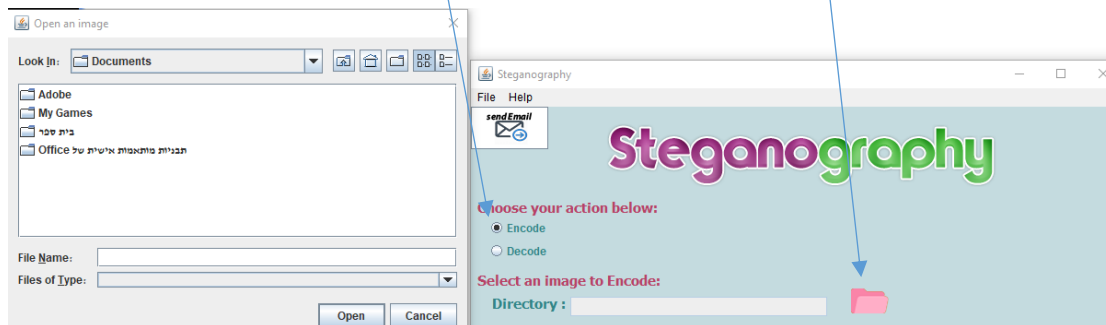


הצפנה ופיענוח של מסר בתמונה:

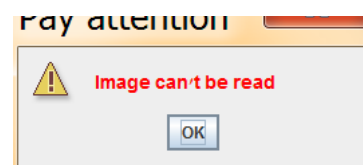
אם ברצון המשתמש להצפין או לפענח מסר בתוך תמונה, ילחץ על כפתור "Files" הנמצא בתחתית חלון הצ'אט.

תהליך ההצפנה:

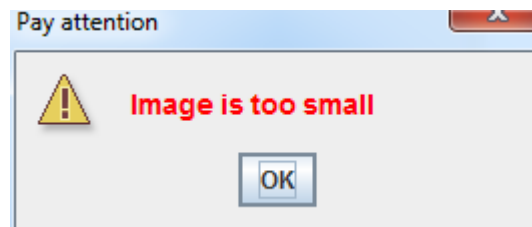
בכדי להצפין יבחר המשתמש בכפתור ה-"Encode" ומיד לאחר מכן יפתח כפתור "Brose" המאפשר למשתמש לבחור בתמונה בה ירצה להצפין את המסר. (יש לשים לב כי פורמטי התמונה האפשריים הם GIF ו-PNG).



אם נבחר קובץ תמונה ריק תופיע ההודעה:



אם נבחרה תמונה קטנה מדי בכדי להצפין בה מסר תופיע ההודעה:

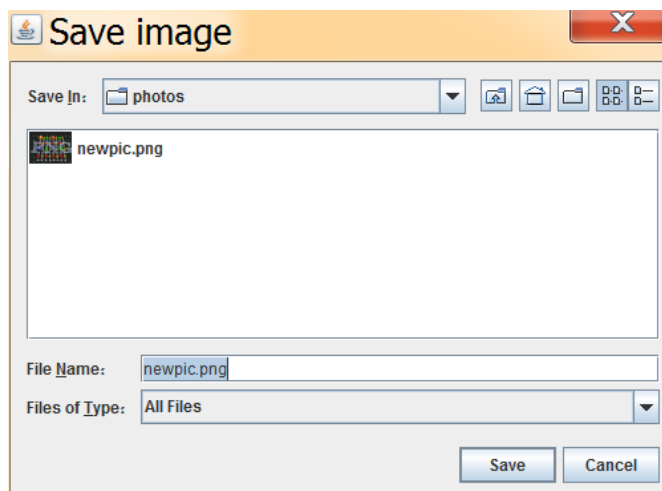


לאחר בחירת התמונה בה המשתמש ירצה להצפין את המסר, תפתח תיבת טקסט ובה המשתמש יקליד את המסר הסודי שירצה אשר לא יעלה על מספר התווים המקסימלי הנקבע על פי גודל התמונה.

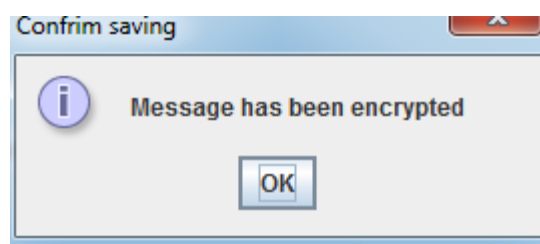


ברגע שיקליד המשתמש לפחות תו אחד בתוך תיבת הטקסט יופיע כפתור "Encode", על המשתמש ללחוץ על כפתור זה בכדי להצפין את המסר בתמונה.

לאחר לחיצה על כפתור זה, יפתח חלון לשם שמירת התמונה בה מוצפן המסר.



לאחר מכן תוצג הודעה אשר תודיע למשתמש האם המסר הוצפן בתמונה בהצלחה.

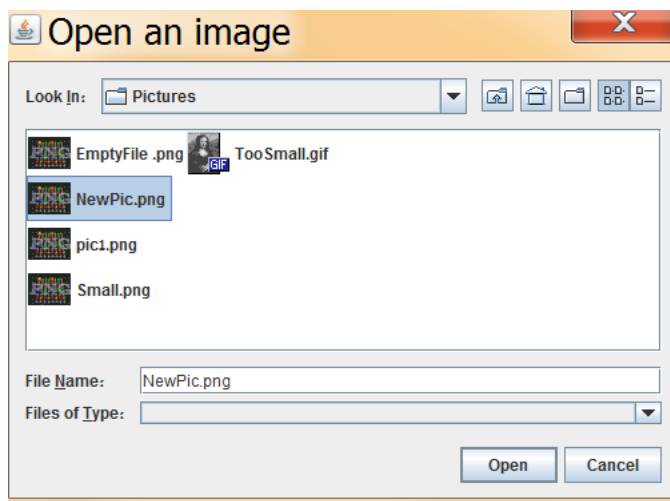


תהליך הפענוח:

בכדי לפענח את המסר הסודי שהוצפן בתמונה יבחר המשתמש בכפתור ה-"Decode" ומיד לאחר מכן יפתח כפתור "Brose" המאפשר למשתמש לבחור בתמונה שבה המסר המוצפן.



ולאחר מכן יפתח חלון המאפשר את בחירת התמונה (כמו בתהליך ההצפנה).



לאחר שהמשתמש יבחר את התמונה בה מוצפן המסר הסודי, יפתח כפתור עיון נוסף ובו המשתמש יתבקש לטעון את המפתח שאיש הקשר שלו (או אותו אדם שהצפין את המסר) שלח לו הנקרא "Public Key.dat".



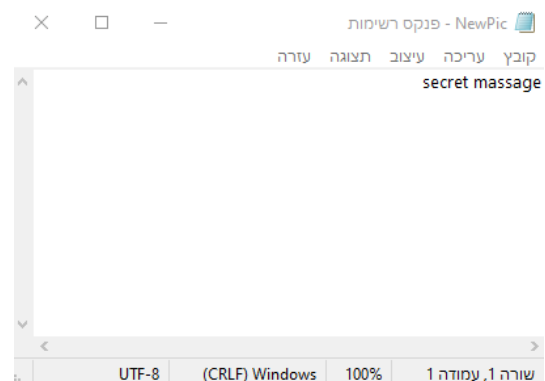
חלון בחירת המפתח:



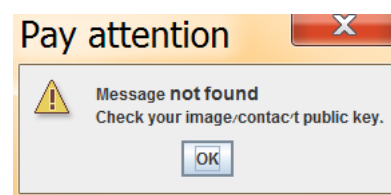
לאחר טעינת המפתח יופיע כפתור ה-"Decode", על המשתמש ללחוץ על כפתור זה בכדי לפענח את המסר הסודי החבוי בתמונה. אם המשתמש עשה את הפעולות האלו נכון, תופיע הודעה שנמצא מסר



ולאחר מכן יפתח Notepad עם המסר הסודי שהתגלה.



במקרה ואין מסר חבוי בתמונה או במקרה והמפתח לא מתאים/פגום תופיע ההודעה הבאה:

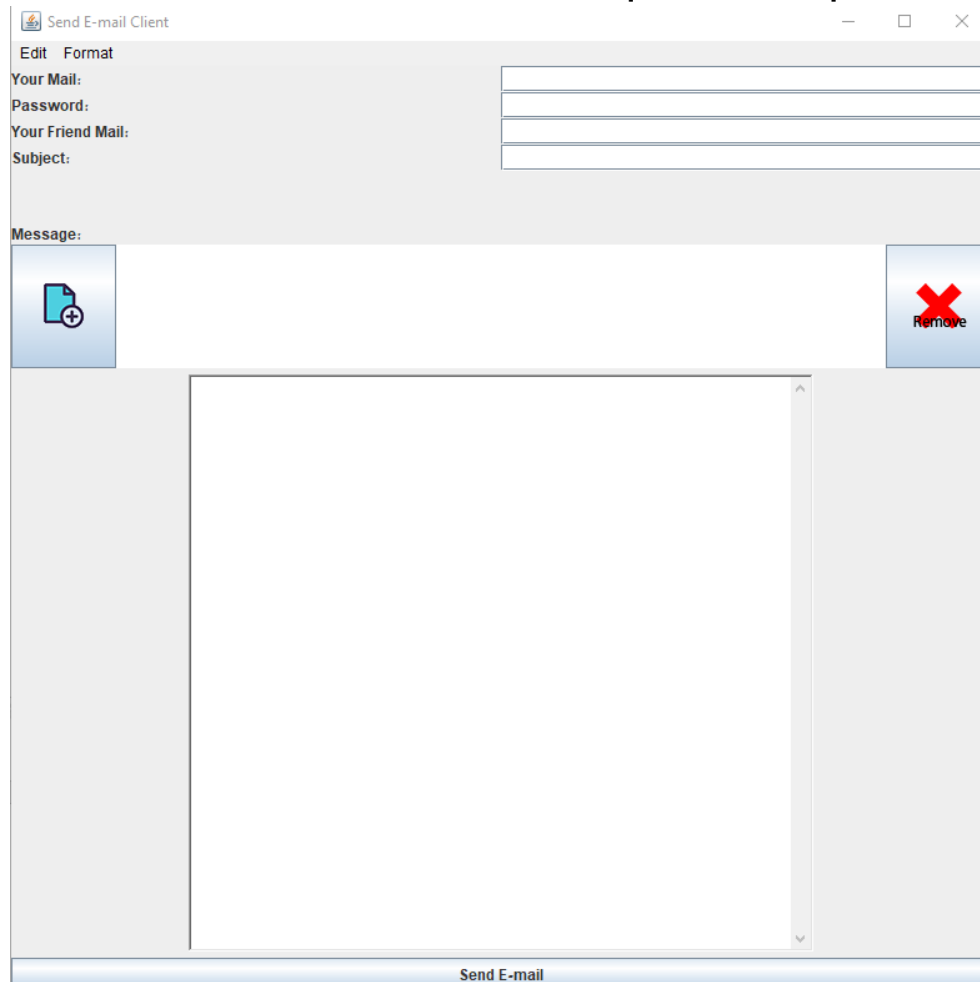


תהליך שליחת אימייל:

ניתן ללחוץ על כפתור שליחת אימייל שנמצא במסך הראשי



לאחר מכן יפתח החלון הבא

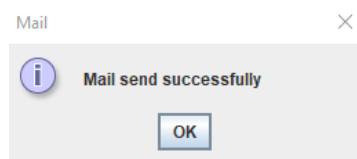


בחלון זה ממלאים את הפרטים של החשבון גימייל שלך (מייל וסיסמא), המייל של המשתמש השני, הנושא של המייל וניתן להוסיף קבצים ולכתוב דברים נוספים בגוף של ההודעה

לאחר לחיצה על הכפתור השליחה של המייל נקבל את ההודעה הבאה

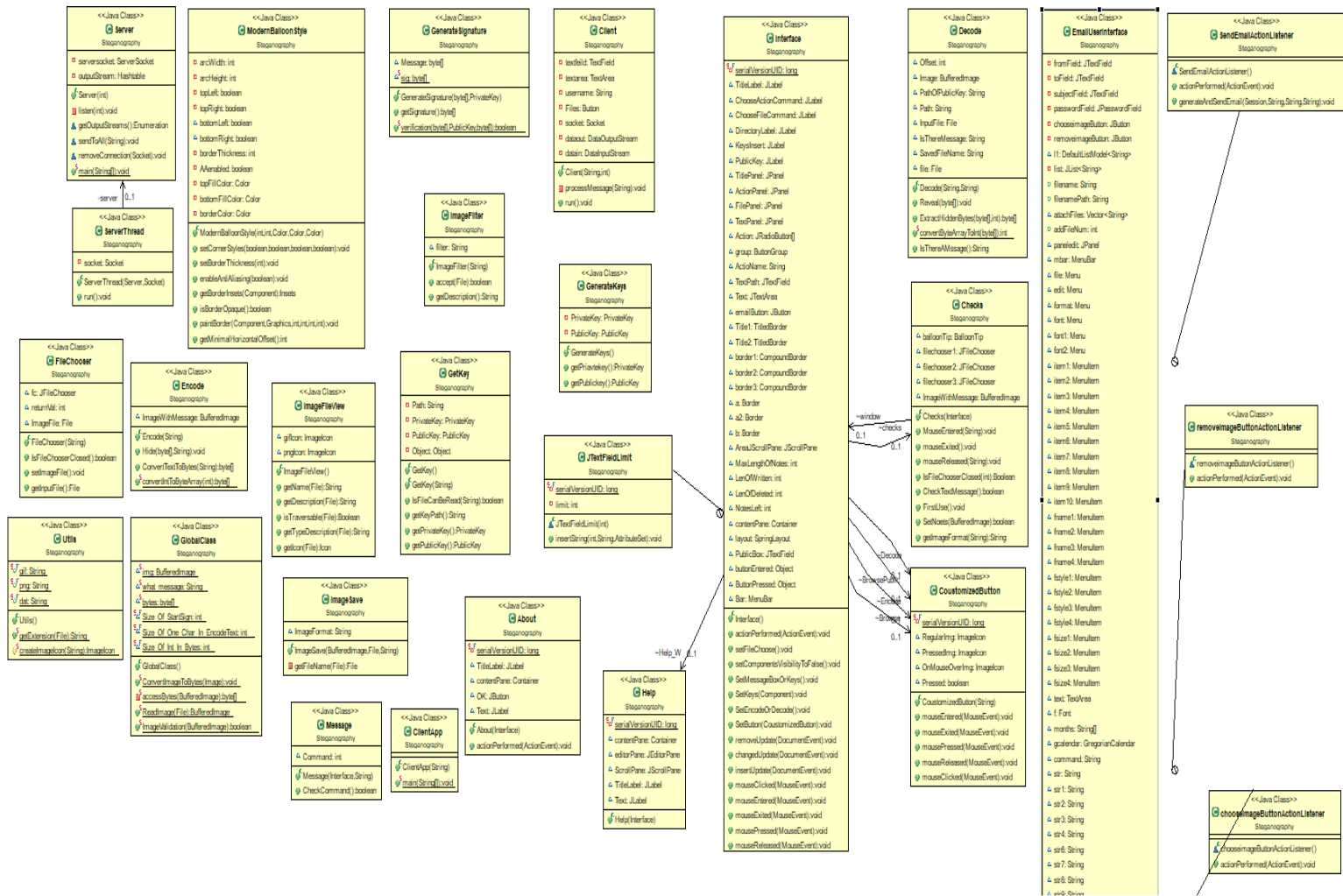


אם המייל לא נשלח בהצלחה ושיש בעיה בשם משתמש או בסיסמא ואת ההודעה הזאת



אם האימייל נשלח בהצלחה

תרשים UML:



פירוט מחלקות:

מחלקת About

מחלקה זו מסבירה אודות התוכנה.

פירוט	שדה
כותרת החלון	JLabel <code>TitleLabel</code>
הטקסט בחלון	JLabel <code>Text</code>
החלון הראשי של המחלקה About	Container <code>ContentPane</code>
כפתור אישור	JButton <code>OK</code>

פירוט	שיטה
בנאי מחלקת About אשר נועד ליצירת החלון בו נראה את המידע אודות התוכנה	<code>public</code> <code>About(Interface i)</code>
פונקציה אשר סוגרת את חלון ה-About בעת לחיצה על כפתור OK	<code>public void</code> <code>actionPerformed(ActionEvent e)</code>

מחלקת Checks

מחלקה העוסקת בבדיקות עם העכבר.

פירוט	שדה
כפתורי בחירת קבצים עבור המפתחות והתמונות	JFileChooser <code>filechooser1,filechooser2,filechooser3</code>
הממשק	Interface <code>window</code>
התמונה עם המסר	BufferedImage <code>ImageWithMessage</code>

פירוט	שיטה
חלון הקופץ כשעומדים עם העכבר על הקובץ	<code>public void</code> <code>MouseEntered(String name)</code>
שיטה הסוגרת את החלון כשמזיזים את העכבר מהקובץ	<code>public void</code> <code>mouseExited()</code>

<code>public boolean IsFileChooserClosed(int returnVal)</code>	שיטה הבודקת האם הקובץ שנבחר על ידי המשתמש נסגר
<code>public boolean CheckTextMessage()</code>	שיטה הבודקת את הודעת הטקסט
<code>public void FirstUse()</code>	שיטה היוצרת מפתחות אוטומטית
<code>public boolean SetNotes(BufferedImage image)</code>	שיטה המודיעה על תמונה קטנה מדי במקרה שהתמונה קטנה מדי
<code>public String getImageFormat(String path)</code>	שיטה לקבלת פורמט התמונה

מחלקת Client

מחלקה המשתמש עבור צ'אט התוכנה.

שדה	פירוט
<code>private TextField textfeild</code>	שדה לכתיבת הודעה למשתמש בצ'אט
<code>private TextArea textarea</code>	שדה המציג את השיחה
<code>private String username</code>	שם המשתמש
<code>private Button Files</code>	כפתור להצפנה ופענוח
<code>private Socket socket</code>	הסוקט שמחבר בין הלקוח לשרת
<code>private DataOutputStream dataout</code>	תקשורת עם הסרבר כלפי חוץ
<code>private DataInputStream datain</code>	תקשורת עם הסרבר כלפי פנים

שיטה	פירוט
<code>private void processMessage(String message)</code>	שיטה הנקראת כאשר המשתמש מקליד
<code>public void run()</code>	שיטה המראה הודעה שהתקבלה מחלון אחר

מחלקת CoustomizedButton

מחלקה האחראית לשינוי תמונת הכפתור בהתאם לפעולות העכבר השונות כגון, לחיצה או מעבר עם העכבר.

פירוט	שדה
תמונת הכפתור במצב המקורי	ImageIcon <code>RegularImg</code>
תמונת הכפתור בעת לחיצת העכבר עליו	ImageIcon <code>PressedImg</code>
תמונת הכפתור במעבר עם העכבר עליו	ImageIcon <code>OnMouseOverImg</code>
שדה לבדיקת לחיצה של העכבר	<code>boolean pressed</code>

פירוט	שיטה
שיטה הבודקת אם המשתמש לוחץ על הכפתור, אם כן, הכפתור משנה את תמונתו לכפתור לחוץ, אחרת עובר למצב עכבר על הכפתור	<code>public void mouseEntered(MouseEvent e)</code>
שיטה הבודקת האם המשתמש מעביר את העכבר מהכפתור, אם כן, תמונת הכפתור חוזרת למצב כפתור רגיל	<code>public void mouseExited(MouseEvent e)</code>
שיטה הבודקת האם המשתמש לוחץ על הכפתור, אם כן הכפתור עובר למצב כפתור לחוץ	<code>public void mousePressed(MouseEvent e)</code>
שיטה המשמשת את השיטה <code>mouseEntered</code>	<code>public void mouseReleased(MouseEvent e)</code>
	<code>public void mouseClicked(MouseEvent e)</code>

מחלקת Decode

מחלקה אשר אחראית על פענוח התמונה בה נמצא המסר הסודי.

פירוט	שדה
מכיל את התמונה המוצפנת	BufferedImage Image;
מכיל ומייצג את המסלול של המפתח הציבורי	String PathOfPublicKey
מכיל ומייצג את המסלול של התמונה	String Path
ייבוא הקובץ שאליו מגיע הטקסט המוצפן	File InputFile
משמש לבדיקה האם יש מסר	String IsThereMessage
שומר את הקובץ עם הטקסט המוצפן	String SavedFileName
קובץ הטקסט שאליו נשמר המסר הסודי	File file
משמש למיקום היסט בבתי התמונה	int Offset

פירוט	שיטה
השיטה בודקת האם קיימת הודעה, אם כן, מחזירה את המסלול, אחרת מחזירה שקר	public String IsThereAMessage()
השיטה חושפת את הצופן	public void Reveal(byte[] bytes)
השיטה מוציאה את הביתים המוחבאים	public byte[] ExtractHiddenBytes(byte[] bytes, int Size)
השיטה הופכת את הביתים למספרים שלמים (int)	public static int convertByteArrayToInt(byte[] integerInBytes)

מחלקת EmailUserInterface

מחלקה המשמשת לשליחת אימייל.

פירוט	שדה
שדה לכתיבת האימייל של המשתמש ששולח	<code>private TextField fromField</code>
שדה לכתיבת האימייל של המשתמש שמקבל	<code>private TextArea toField</code>
שדה לכתיבת הנושא של האימייל	<code>private TextArea subjectField</code>
שדה לכתיבת הסיסמא של האימייל של המשתמש ששולח	<code>private JPasswordField passwordField</code>
כפתור לבחירת מסמך/קובץ	<code>private JButton chooseImageButton</code>
כפתור להסרת קובץ נבחר	<code>private JButton removeImageButton</code>
רשימה שמכילה את הקבצים המצורפים לאימייל	<code>private DefaultListModel<String> l1</code>
רשימה שמאכסנת ומציגה את l1	<code>private JList<String> list</code>
השם של הקובץ	<code>public String filename</code>
המסלול של המיקום של הקובץ	<code>public String filenamePath</code>
וקטור המכיל את הקבצים המצורפים	<code>Vector<String> attachFiles</code>
משתנה בוליאני שבודק עם צורפו קבצים לאימייל	<code>public boolean ImageChoose</code>
סופר את מספר הקבצים המצורפים לאימייל	<code>public int addFileNum</code>

פירוט	שיטה
שיטה המסדרת את כל האובייקטים על המסך	<code>private void InitializeUI()</code>
שיטה שיוצרת ושולחת אימייל	<code>private void generateAndSendEmail()</code>
שיטה המריצה את תכונה של האימייל	<code>public void run()</code>

מחלקת Encode

המחלקה האחראית על הצפנת המסר בתמונה, המחלקה לוקחת תמונה במצב המקורי שלה ומצפינה בה את המסר הסודי.

פירוט	שדה
שומר את התמונה המוצפנת	BufferedImage ImageWithMessage

פירוט	שיטה
השיטה מחביאה את המסר בתמונה	<code>public void Hide(byte[] bytes, String Text)</code>
השיטה הופכת את הטקסט לבתים	<code>public byte[] ConvertTextToBytes(String Text)</code>
השיטה הופכת את המספרים לבתים	<code>public static byte[] convertIntToByteArray(int integer)</code>
השיטה הופכת את אורך החתימה לבתים	<code>public byte[] ConvertLengthOfSigToByte(int integer)</code>

מחלקת FileChooser

מחלקה המשמשת לטעינת קובץ התמונה.

פירוט	שדה
מאפשר את חלון הבחירה	JFileChooser fc
מצב הפעולה (בודק אם אושר)	<code>int returnVal</code>
שומר את קובץ התמונה	File ImageFile

פירוט	שיטה
שיטה לבחירת הקובץ	<code>public FileChooser(String Operation)</code>
שיטה הבודקת אם החלון נסגר	<code>public boolean IsFileChooserClosed()</code>
שמירת התמונה לתוך הקובץ	<code>public void setImageFile()</code>
שיטה המחזירה את הקובץ	<code>public File get inputFile()</code>

מחלקת GenerateKeys

המחלקה מייצרת מפתחות רנדומליים לצורך הצפנה.

פירוט	שדה
מפתח פרטי	<code>private PrivateKey PrivateKey</code>
מפתח ציבורי	<code>private PublicKey PublicKey</code>

פירוט	שיטה
שיטה המחזירה את המפתח הפרטי	<code>public PrivateKey getPrivatekey()</code>
שיטה המחזירה את המפתח הציבורי	<code>public PublicKey getPublickey()</code>

מחלקת GenerateSignature

המחלקה יוצרת חתימה של איש הקשר ובדיקה האם התהליך תקין.

פירוט	שדה
ההודעה של איש הקשר	<code>byte[] Message</code>
"החתימה" של איש הקשר	<code>static byte [] sig</code>

פירוט	שיטה
השיטה מחזיר את החתימה של איש הקשר	<code>public byte[] getSignature()</code>
השיטה בודקת ומחזירה האם יש התאמה בין החתימה למפתח והכל תקין או שיש שגיאה עם איש הקשר והתהליך שובש	<code>static public boolean verification(byte[] Sig, PublicKey PublicKey, byte [] message)</code>

מחלקת GetKey

מחלקה המקבלת מפתח ובודקת את נתוניו.

פירוט	שדה
מסלול המפתח הפרטי	<code>private String Path = "../Keys/Private Key.dat"</code>
מפתח פרטי	<code>private PrivateKey PrivateKey = null</code>
מפתח ציבורי	<code>private PublicKey PublicKey = null</code>
משמש לטעינה ממסלול הקובץ	<code>private Object Object = null</code>

פירוט	שיטה
שיטה הבודקת האם הקובץ יכול להיקרא	<code>public boolean IsFileCanBeRead(String Path)</code>
שיטה המחזירה את מסלולו של המפתח	<code>public String getKeyPath()</code>
שיטה המחזירה מפתח פרטי	<code>public PrivateKey getPrivateKey()</code>
שיטה המחזירה מפתח ציבורי	<code>public PublicKey getPublicKey()</code>

מחלקת GlobalClass

המחלקה אחראית להפיכת תמונה למערך פיקסלים בבתים

פירוט	שדה
התמונה	<code>static BufferedImage img</code>
מערך הבתים	<code>static byte[] bytes</code>
גודל המערך StartSign	<code>static final int Size_Of_StartSign = 2</code>
גודל של תו אחד	<code>static final int Size_Of_One_Char_In_EncodeText = 8</code>
גודל של מספר שלם (int) בבתים	<code>static final int Size_Of_Int_In_Bytes = 4</code>

פירוט	שיטה
השיטה הופכת את התמונה למערך פיקסלים בבתים	<code>public static void ConvertImageToBytes(Image i)</code>
פונקציית עזר ההופכת תמונה טעונה למערך בתים	<code>private static byte[] accessBytes(BufferedImage image)</code>
שיטה הטוענת את התמונה	<code>static public BufferedImage ReadImage(File Path)</code>
שיטה המאמתת אם קיימת תמונה או לא	<code>public static boolean ImageValidation(BufferedImage img)</code>

מחלקת ImageView

המחלקה משמשת ליצירת חלון של פתיחת קבצי תמונה.

פירוט	שדה
אייקון לפורמט GIF	<code>ImageIcon gifIcon = Utils.createImageIcon("../Icon Format/gifIcon.gif");</code>
אייקון לפורמט PNG	<code>ImageIcon pngIcon = Utils.createImageIcon("../Icon Format/pngIcon.png");</code>

פירוט	שיטה
שיטה המחזירה את שם הקובץ	<code>public String getName(File f)</code>
שיטה המחזירה את תיאור הקובץ	<code>public String getDescription(File f)</code>
שיטה הבודקת אם המליח או לא הצליח	<code>public Boolean isTraversable(File f)</code>
מחזיר את תיאור הפורמט	<code>public String getTypeDescription(File f)</code>
מחזיר את סוג האייקון	<code>public Icon getIcon(File f)</code>

מחלקת ImageFilter

סינון סוגי התמונות שאפשר לשים בתמונה.

פירוט	שדה
מקבל מחרוזת לבדיקה של סוג קובץ	<code>String filter</code>

פירוט	שדה
השיטה מסננת את סוג הקובץ אם תמונה מפורמטי PNG, GIF או מפתח (DAT) מחזירה אמת, אחרת שקר	<code>public boolean accept(File f)</code>
השיטה מחזירה את תיאור הקובץ	<code>public String getDescription()</code>

מחלקת ImageSave

מחלקה האחראית על שמירת התמונה המוצפנת.

פירוט	שדה
מחרוזת המשמשת לידיעת סיומת התמונה	String <code>ImageFormat</code>

פירוט	שיטה
השיטה שומרת את התמונה כולל סיומת	<code>public ImageSave(BufferedImage img, File path, String format)</code>
שיטה המונעת מהמשתמש להכניס פורמטים השונים מ-GIF ו-PNG	<code>private File getFileName(File p)</code>

מחלקת Messages

מחלקה האחראית על ההודעות התראה המוצגות למשתמש במהלך שימוש בתוכנה.

פירוט	שדה
מייצג פקודה "כן" או "לא"	int <code>Command</code>

פירוט	שיטה
השיטה האחראית על הודעות בהתאם לאירועים	<code>public Message(Interface MainWindow, String Operation)</code>
השיטה בודקת במה בחר המשתמש "כן" או "לא" ליצירת מפתחות חדשים, ויוצרת מפתחות חדשים בהתאם לבחירת המשתמש	<code>public boolean CheckCommand()</code>

מחלקת Server

שרת עבור צ'אט התוכנה.

פירוט	שדה
סוקט לחיבורים חדשים	<code>private ServerSocket serversocket</code>
טבלת האש שמשמשת למיון מסוקטים לסטרימים	<code>private Hashtable outputStream = new Hashtable()</code>

פירוט	שיטה
השיטה מאזינה לסטרימים חדשים שמתקבלים מהסוקטים, שולחת לסרבר ומשם (בעזרת פונקציות נוספות) נשלחות הודעות מהמשתמש	<code>private void listen(int port) throws IOException</code>
השיטה מחזירה את מספר הסטרימים לכל סוקט	<code>Enumeration getOutputStreams()</code>
השיטה מקבלת דטה (הודעה) מהמשתמש ושולח לשאר המשתמשים	<code>void sendToAll(String message)</code>
השיטה מוחקת חיבור של סוקט וסוגרת אותו	<code>void removeConnection(Socket socket)</code>

מחלקת ServerThread

מחלקת עזר של השרת היוצרת תהליך לשליחת הודעות משתמשים.

פירוט	שדה
מקבל את הסרבר	<code>private Server server</code>
סוקט שמחובר ללקוח	<code>private Socket socket</code>

פירוט	שיטה
השיטה יוצרת תהליך לשליחת הודעות שהתקבלו מהסוקט אל השרת ושולחת את הודעות המשתמש לשאר המשתמשים	<code>public void run()</code>

מחלקת Utils

מחלקה העוזרת במציאת סיומות קבצים ויצירה של אייקוני שמירה לתמונה.

פירוט	שדה
מייצג פורמט GIF	<code>public final static String gif = "gif"</code>
מייצג פורמט PNG	<code>public final static String png = "png"</code>
מייצר פורמט DAT (למפתחות)	<code>public final static String dat = "dat"</code>

פירוט	שיטה
השיטה מחזירה את סיומת הקובץ	<code>public static String getExtension(File f)</code>
השיטה משנה את האייקוני שמירה על פי סוג התמונה	<code>protected static ImageIcon createImageIcon(String path)</code>

נספח א' - קודי המחלקות:

```
package Steganography;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class About extends JDialog implements ActionListener
{
    private static final long serialVersionUID = 1L;
    JLabel titleLabel;
    Container contentPane;
    JButton OK;
    JLabel Text;
    public About(Interface i)
    {
        //Set window properties
        super(i,"About Steganography",true);
        Point location=i.getLocation();
        setLocation(location.x+50, location.y+70);
        setSize(416,405);
        setResizable(false);
        setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        //Set Layout
        SpringLayout layout = new SpringLayout();
        setLayout(layout);

        //Set the About Title
        ImageIcon icon = new ImageIcon(getClass().getResource("../Interface Images/AboutTitle.png"));
        //create JLabels
        titleLabel=new JLabel();
        //set icon "AboutTitle" to JLabel
        titleLabel.setIcon(icon);
        Text=new JLabel();
        Text.setFont(new Font("Cambria",Font.PLAIN,14));
        Text.setForeground(Color.black);

        //face=Cambria
```



```

        Text.setText("<Html><font >"+
                    "Steganography is the art and the sciense of <br>"+
                    "hiding a secret messages in a way that <br>"+
                    "no body besides the one who recive (in this case) <br>"+
                    "this picture know about <br>"+
                    "the secret message.<br>"+

                    "<hr width=70cm>"+
                    "</font><Html>");

    OK=new JButton("OK");
    OK.addActionListener(this);

    //Get Main Panel
        contentPane=getContentPane();

        //Set Components to window
        layout.putConstraint(SpringLayout.WEST, titleLabel,0, SpringLayout.WEST, contentPane);
    layout.putConstraint(SpringLayout.NORTH, titleLabel,0, SpringLayout.NORTH, contentPane);
    contentPane.add(titleLabel);

        layout.putConstraint(SpringLayout.WEST, Text,35, SpringLayout.WEST, contentPane);
    layout.putConstraint(SpringLayout.NORTH, Text,20, SpringLayout.SOUTH, titleLabel);
    contentPane.add(Text);

        layout.putConstraint(SpringLayout.WEST, OK,330, SpringLayout.WEST, contentPane);
    layout.putConstraint(SpringLayout.SOUTH, OK,35, SpringLayout.SOUTH, Text);
    contentPane.add(OK);

        setVisible(true);
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==OK)
        {
            dispose();
        }
    }
}

```

```

package Steganography;

import java.awt.Color;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.swing.JFileChooser;

import net.java.balloontip.BalloonTip;
import net.java.balloontip.styles.ModernBalloonStyle;
import net.java.balloontip.utils.TimingUtils;

public class Checks
{
    BalloonTip balloonTip;

    JFileChooser filechooser1,filechooser2,filechooser3;

    Interface window;

    BufferedImage ImageWithMessage;

    public Checks(Interface w)
    {
        window=w;
    }

    public void MouseEntered(String Name)
    {
        Color c=new Color(7,75,118);

        ModernBalloonStyle Modern = new ModernBalloonStyle(10,10,c,Color.white,Color.black);

        Modern.setBorderThickness(2);

        Modern.enableAntiAliasing(true);

        if(Name=="Browse")

            //Click this button in order to Choose an image

            balloonTip = new BalloonTip(window.Browse, "<html><font color=black> Choose an
image.</font></html>",Modern,false);

```

```

else

    if(Name=="BrowsePublic")

        balloonTip = new BalloonTip(window.BrowsePublic, "<html><font
color=black> Click this button in order to choose a public key.</font></html>",Modern,false);

    else

        balloonTip.setPadding(0);

        TimingUtils.showTimedBalloon(balloonTip, 5000);

}

public void mouseExited()

{

    balloonTip.closeBalloon();

}

public void mouseReleased(String Name)

{

    int returnVal=0;

    File path;

    if(Name=="Browse")

    {

        if(filechooser1==null)

        {

            filechooser1=new JFileChooser();

            System.out.println("NEW");

        }

        filechooser1.setAcceptAllFileFilterUsed(false);

        filechooser1.setFileFilter(new ImageFilter("Images"));

        filechooser1.setFileView(new ImageFileView());

        filechooser1.setDialogTitle("Open an image");

        returnVal=filechooser1.showOpenDialog(window);

        if(!IsFileChooserClosed(returnVal))

        {

            path=filechooser1.getSelectedFile();

            //path.exists()

            if(path.canRead())

```

```

        {
            ImageWithMessage=GlobalClass.ReadImage(path);
            if(GlobalClass.ImageValidation(ImageWithMessage))
            {
                GlobalClass.ConvertImageToBytes(ImageWithMessage);

                window.TextPath.setText(path.toString());
                if(window.ActioName=="Encode")
                {
                    //check if the image is not too small,if it does

                    if(SetNoets(ImageWithMessage))
                        window.setSize(726,457);
                    else
                        return;
                }
                else
                    window.setSize(726,404);
                window.SetMessageBoxOrKeys();
            }
            else
            {
                window.setComponentsVisibilityToFalse();
                window.setSize(726,298);
                new Message(window,"ImageError");
                System.out.println("File Is Damaged");
                return;
            }
        }
        else

            System.out.println("File not exsistence");
    }
}
else

    if(Name=="BrowsePublic")
    {
        if(filechooser2==null)

```

than don't set the notes and the text box

```

        filechooser2=new JFileChooser("");

        filechooser2.setAcceptAllFileFilterUsed(false);
        filechooser2.setFileFilter(new ImageFilter("PublicKey"));
        //?
        filechooser2.setDialogTitle("Open a public key");
        returnVal=filechooser2.showOpenDialog(window);
        if(!IsFileChooserClosed(returnVal))
        {
            path=filechooser2.getSelectedFile();
            window.PublicBox.setText(path.toString());

            window.SetEncodeOrDecode();
            window.setSize(726,404);
        }
    }
    else
        if(Name=="Encode")
        {
            if(filechooser3==null)
                filechooser3=new JFileChooser();
            filechooser3.setDialogTitle("Save image");
            filechooser3.setFileView(new ImageFileView());
            returnVal=filechooser3.showSaveDialog(window);
            if(!IsFileChooserClosed(returnVal))
            {
                FirstUse();
                String
                new File(window.TextPath.getText());
                new Encode(window.Text.getText());
                path=filechooser3.getSelectedFile();

                if(GlobalClass.what_message=="Encode")
                    new
                    new
                Message(window,GlobalClass.what_message);
            }
        }
    }
    ImageForamt=getImageFormat(window.TextPath.getText());

```

```

        }
        else
            if(Name=="Decode")
            {
                Decode decode=new
                Decode(window.TextPath.getText(),window.PublicBox.getText());
                String
                MessageExistent=decode.IsThereAMessage();

                new Message(window,MessageExistent);
                if(MessageExistent!="false")
                try
                {

                    Runtime.getRuntime().exec("notepad"+" "+"../Decoded Messages/"+decode.SavedFileName+".txt");

                }
                catch (IOException e)
                {

                    e.printStackTrace();

                }

            }

    }

    public Boolean IsFileChooserClosed(int returnVal)
    {
        if (returnVal == JFileChooser.APPROVE_OPTION)
            return false;
        else
            System.out.println("Attachment cancelled by user.");
        return true;
    }

    public boolean CheckTextMessage()
    {
        int MsgLength=window.Text.getText().trim().length();
        if(MsgLength==0)
        {
            System.out.println("sapce");
            return false;
        }
        return true;
    }

```

```

    }

    public void FirstUse()
    {
        File PublicKey=new File("../Keys/Public Key.dat");
        if(PublicKey.canRead())
        {
            System.out.println("Keys already created");
        }
        else
        {
            System.out.println("First use: Keys created automatically");
            new GenerateKeys();
        }
    }

    public boolean SetNoets(BufferedImage img)
    {
        int AmountOfNoets=(GlobalClass.bytes.length-456)/8;

        if(AmountOfNoets<=0)
        {
            window.setComponentsVisibilityToFalse();
            window.setSize(726,303);
            new Message(window,"SmallImage");

            System.out.println("Image is too small");
            return false;
        }
        else
        {
            if(AmountOfNoets<5000000)
            {
                window.NotesLeft=AmountOfNoets;
                window.Title2.setTitle("Notes left: "+window.NotesLeft);
                window.Text.setDocument(window.new JTextfieldLimit(AmountOfNoets));
                window.Text.getDocument().addDocumentListener(window);
                window.AreaJScrollPane.setBorder(window.b);
                window.AreaJScrollPane.setBorder(window.border1);
                window.validate();
            }
        }
    }

```

```

        return true;

    }

    public String getImageFormat(String p)
    {
        String format;

        String ImageName = p; //the whole path

        int period=ImageName.indexOf('.');

        format=ImageName.substring(period+1, ImageName.length());

        System.out.println("Format is: "+format);

        return format;
    }

}

```

```
package Steganography;
```

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;
import javax.swing.*;

```

```
public class Client extends Panel implements Runnable
```

```

{
    //Components for the visual display of the chat window
    private TextField textfield = new TextField(); //שדה לכתובת הודעה למשתמש בצ'אט
    private TextArea textarea = new TextArea(); //שדה המציג את השיחה
    private String username; //סטרינג להצגת המשתמש
    private Button Files; //כפתור שיעביר את המשתמש לחלון ההצפנה/פיענוח
    private Socket socket; //the socket connecting us to the server
    //the streams we communicate to the server, these come from the socket
    private DataOutputStream dataout; //תקשורת עם הסרבר כלפי חוץ
    private DataInputStream datain; //תקשורת עם הסרבר כלפי פנים

    public Client(String host, int port)

```



```

{
    username = JOptionPane.showInputDialog("Enter user name");// כאשר מעפילים את התוכנה יוצג חלון
    להכנסת שם משתמש

    if(username.equals(""))//במקרה שלא נכתב שם
        username = "Anonymous";//יוצג אנונימי בצ'אט

    Files = new Button("Files");//יוצר את כפתור ההעברה להצפנה/פיענוח

    //set up the screen

    setLayout(new BorderLayout());//ארגון החלון
    add("North", textfield);//שורת הכתיבה תהיה למעלה
    add("Center", textarea);//חלון השיחה יוצג באמצע
    add(BorderLayout.SOUTH, Files);//כפתור ההצפנה/פיענוח יופיע למטה

    //Anonymous class is used as callback to see the message when someone types a line and
    press send

    Files.addActionListener(new ActionListener());//האזנה לכפתור
    {
        public void actionPerformed(ActionEvent e)//אם בוצעה לחיצה עליו
        {
            new Interface();//יכנס לחלון ההצפנה/פיענוח
        }
    });

    textfield.addActionListener(new ActionListener());//ההזנה לאיזור ההקלדה
    {
        public void actionPerformed(ActionEvent e)//בוצעה פעולה
        {
            processMessage(e.getActionCommand());//מעבד את המסר שהוקלד
        }
    });

    //connect to the server

    try
    {
        //initiate the connection

        socket = new Socket(host, port);//אתחול סוקט החיבור

        //connection confirmed

        System.out.println("connected to: " + socket);

        //create DataInput and DataOutput

        datain = new DataInputStream(socket.getInputStream());
        dataout = new DataOutputStream(socket.getOutputStream());

        //start a background thread for message receiving

        new Thread(this).start();
    }
}

```

```

    }
    catch(IOException ex)
    {
        System.out.println(ex);
    }
}

//gets called when the user types something
private void processMessage(String message)//
{
    try
    {
        //send it to the server
        dataout.writeUTF(username + ": " + message); //שולח לשרת את הקלט שהתקבל
        //clear out text input field
        textfeild.setText(""); //מנקה את שורת ההקלדה אחרי שליחת הודעה
    }
    catch(IOException ex)
    {
        System.out.println(ex);
    }
}

//show message from the other window (runs by the background thread)
public void run()//מציג הודעות ממשתמשים אחרים
{
    try
    {
        //Receive messegges one-by-one, forever
        while(true)
        {
            //get the next message
            String message = datain.readUTF(); //מקבל את ההודעות שנשלחו לשרת
            //print it to the next window
            textarea.append(message + "\n"); //ומוסיף אותם לחלון השיחה
        }
    }
    catch(IOException ex)
    {
        System.out.println(ex);
    }
}

```

```
package Steganography;

import java.applet.*;
import java.awt.*;
import java.io.*;
import java.net.*;
import javax.swing.JFrame;

public class ClientApp extends JFrame
{
    public ClientApp(String caption)
    {
        super(caption);
        setLayout(new BorderLayout());
        add("Center", new Client("LocalHost", 6000));
        setSize(500, 500);
        setVisible(true);
    }

    public static void main(String args[])
    {
        new ClientApp("Chat");
    }
}
```

```

package Steganography;

import java.awt.event.*;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

public class CoustomizedButton extends JLabel implements MouseListener
{
    private static final long serialVersionUID = 1L;

    ImageIcon RegularImg=null; // ללא לחיצה או מעבר עכבר
    ImageIcon PressedImg=null; // העכבר ע"י
    ImageIcon OnMouseOverImg=null; // מעבר עכבר
    boolean Pressed; // לבדיקת לחיצה
    public CoustomizedButton(String Name)
    {
        if(Name=="Browse")// אם נבחר לחיצה/מעבר עכבר לכפתור עיון
        {
            RegularImg=new ImageIcon(getClass().getResource("../Interface
            Images/BrowseRegular.png")); // רגיל
            OnMouseOverImg=new ImageIcon(getClass().getResource("../Interface
            Images/BrowseOnMouseOver.png")); // מעבר עכבר
            PressedImg=new ImageIcon(getClass().getResource("../Interface
            Images/BrowsePressed.png")); // לחיצה
        }

        if(Name=="Encode")// אם נבחר לחיצה/מעבר עכבר לכפתור הצפנה
        {
            RegularImg=new ImageIcon(getClass().getResource("../Interface
            Images/EncodeRegular2.png")); // רגיל
            OnMouseOverImg=new ImageIcon(getClass().getResource("../Interface
            Images/EncodeOnMouseOver2.png")); // מעבר עכבר
            PressedImg=new ImageIcon(getClass().getResource("../Interface
            Images/EncodePressed.png")); // לחיצה
        }

        if(Name=="Decode")// אם נבחר לחיצה/מעבר עכבר לכפתור פיענוח
        {
            RegularImg=new ImageIcon(getClass().getResource("../Interface
            Images/DecodeRegular.png")); // רגיל
            OnMouseOverImg=new ImageIcon(getClass().getResource("../Interface
            Images/DecodeOnMouseOver.png")); // מעבר עכבר
        }
    }
}

```

```

        PressedImg=new ImageIcon(getClass().getResource("../Interface
Images/DecodePressed.png")); //לחץ
    }

    אם נבחר לחיצה/מעבר עכבר לכפתור מפתחות חדש/new keys"
    {

        RegularImg=new ImageIcon(getClass().getResource("../Interface
Images/NewKeysRegular2.png")); //רגיל

        OnMouseOverImg=new ImageIcon(getClass().getResource("../Interface
Images/NewKeysOnMouseOver2.png")); //מעבר עכבר

        PressedImg=new ImageIcon(getClass().getResource("../Interface
Images/NewKeysPressed.png")); //לחץ

    }

    מצב תמונה התחלתי = רגיל/
    setIcon(RegularImg);

    הוסף מאזין לכפתורים/
    addMouseListener(this);

}

public void mouseEntered(MouseEvent e) //בדיקת לחיצה/מעבר העכבר
{

    אם נלחץ//
    if(Pressed)

        שנה לתמונה לחיצה/
        setIcon(PressedImg);

    else

        אם לא נלחץ שנה תמונה למעבר עכבר/
        setIcon(OnMouseOverImg);

}

public void mouseExited(MouseEvent e) //בדיקה אם אין מעבר ולחיצה עכבר כלל/
{

    משנה למצב תמונה רגיל/
    setIcon(RegularImg);

}

public void mousePressed(MouseEvent e) //בדיקת לחיצה העכבר/
{

    נלחץ//
    Pressed=true;

    משנה תמונה למצב לחץ/
    setIcon(PressedImg);

}

public void mouseReleased(MouseEvent e) //שחרור לחיצה/
{

    לא לחץ//
    Pressed=false;

}

public void mouseClicked(MouseEvent e) //לאחר לחיצה/
{

    העכבר חוזר למצב רגיל/
    setIcon(RegularImg);

}

}

```

```

package Steganography;

import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileOutputStream;

import java.security.PublicKey;

public class Decode
{
    int Offset;//היסט בבתי התמונה
    BufferedImage Image;//טעינת התמונה
    String PathOfPublicKey;//המפתח הציבורי של המסלול
    String Path;//הווא את מסלול התמונה בעלת המסר הסודי
    File InputFile;//קובץ התמונה המוצפנת
    String IsThereMessage;//לא אם יש מסר בתמונה או לא
    String SavedFileName;//שם לקובץ
    File file;//קובץ הטקסט שאליו ישמר המסר הסודי

    מחלקה מקבלת את מסלול התמונה ואת מסלול המפתח
    הציבורי
    public Decode(String ImagePath,String PathOfPublicKey)
    {
        Path=ImagePath;//מקבל את מסלול התמונה
        InputFile=new File(Path);//מקבל את הקובץ של התמונה על פי המסלול שצויין
        Image=GlobalClass.ReadImage(InputFile);//מחזיר את התמונה שנטענה מהקובץ
        this.PathOfPublicKey=PathOfPublicKey;//מקבל את מסלול המפתח הציבורי
        Offset=0;//0 להיסט את ההיסט
        Reveal(GlobalClass.bytes);
    }

    טענת כניסה: הפונקציה מקבלת את מערך הבתים של התמונה בה מוצפן המסר
    פעולת הפונקציה: בודקת אם קיימת בכלל הודעה, אם כן מחלצת את החתימה ובודקת תאימות בין המפתחות ואז מחלצת
    "את ההודעה לקובץ טקסט, אם לא מחזירה ערך שקר במשתנה "יש הודעה"

    {
        byte[] StartSign=ExtractHiddenBytes(bytes,GlobalClass.Size_Of_StartSign);//מקבל את המערך של
        התמונה המוצפנת ואת גודל התווים הבודקים האם הוצפנה בכלל הודעה

        if(StartSign[0]!='$' && StartSign[1]!='<')//לא אם בדיקה האם הודעה מוצפנת או לא
        (הוגדר במחלקת הצפנה)

        {
            אם לא נמצא בתמונה מערך זה אין כניראה מסר
            IsThereMessage="false";//אין מסר

        }

        אם כן נמצאו סימנים אלו במערך שחולץ מהתמונה יש מסר מוצפן
    }
}

```

```

{

    System.out.println("Therre is a message");//מודיע על מציאת מסר

    byte[]
    SignLengthInBytes=ExtractHiddenBytes(bytes,GlobalClass.Size_Of_Int_In_Bytes);//מחלץ את גודל מערך החתימה

    int SignLength=convertByteArrayToInt(SignLengthInBytes);//מכניס למשתנה זה את הגודל

    byte [] Signature=ExtractHiddenBytes(bytes,SignLength);//מחלץ את החתימה מהתמונה

    byte[]
    LengthInBytes=ExtractHiddenBytes(bytes,GlobalClass.Size_Of_Int_In_Bytes);//חלץ את גודל מערך המסר
    int Length=convertByteArrayToInt(LengthInBytes);//מכניס למשתנה זה את הגודל

    byte Message[]=ExtractHiddenBytes(bytes,Length);//מחלץ מהתמונה את המסר הסודי

    GetKey Read_Key;//
    PublicKey PublicKey;//פה נציב את המפתח הציבורי שיתקבל
    if( (Read_Key=new GetKey(PathOfPublicKey)).IsFileCanBeRead(PathOfPublicKey)
        בדוק האם המסלול חוקי והאם קיים מפתח)
    {
        PublicKey=Read_Key.getPublicKey();//הצב את המפתח הציבורי שנמצא
    }
    else//מסלול לא תקין/
    {
        new Message(null,"Public Key Error");//הודעה שיש בעיה במפתח הציבורי
        return;
    }

    if(GenerateSignature.verification(Signature,PublicKey,Message))//בודק שהמפתחות
    תואמים ומדפיס את המסר
    {
        IsThereMessage="true";//יש מסר
        System.out.print("The Encode Message is: ");//והמסר הוא
        for(int i=0; i<Length; i++)
        {
            System.out.print((char)Message[i]);//ממיר את הבתים לתווים ומדפיס
        }
        אותם לפי סדר הופעתם

        //Save Message To File
    }
}

```

```

try
{
    int EndIndex=InputFile.getName().indexOf(".");// מחשב את כמות
    התווים מהנקודה וסימת התמונה

    SavedFileName=InputFile.getName().substring(0,
    EndIndex);//קובץ התמונה בשם קובץ הטקסט
    מוריד את הנקודה והסימת בשביל לשמור את שם קובץ הטקסט בשם קובץ התמונה

    file=new File("../Decoded
    Messages/"+SavedFileName+".txt");//יוצר את קובץ הטקסט לתיקיה של מסרים שפוענחו ושומר אותו על פי שם התמונה שהתקבלה

    FileOutputStream OutputStream = new
    FileOutputStream("../Decoded Messages/"+SavedFileName+".txt" );//יוצר את ההזרמה לכתיבת תוכן לקובץ

    OutputStream.write(Message);//כותב את ההודעה לתוך הקובץ

    OutputStream.close();//סגירה

}

catch( Exception e )//שגיאה במהלך יצירת הקובץ או כתיבה אליו
{
    e.printStackTrace( System.out );
    System.out.println("Could not save the decoded message to the txt file");
}

else
    IsThereMessage="false";//אין הודעה

}

}

public byte[] ExtractHiddenBytes(byte[] bytes,int Size)// מחלץ את הסיבית הכי פחות משמעותית בכל בית בתמונה
ומחזיר מערך בתים של מה שחולץ
{
    byte[] HiddenBytes=new byte[Size];//מערך הבתים שיוחזר

    for(int i=0; i<Size; i++)//מיקום במערך שיוחזר
        for(int j=0; j<8; j++)//הכנסת סיביות
        {
            HiddenBytes[i] = (byte) ((HiddenBytes[i] << 1) | (bytes[Offset] & 1)); //this
            part: (b[i] << 1) comes in order to mov to bit to the higher place

            Offset++;//התקדמות לבית הבא
        }

    return HiddenBytes;//מחזיר את מערך הבתים שחולץ
}

public static int convertByteArrayToInt(byte[] integerInBytes)
{
    int[] byteToInt;
    int result;

```



```

byteToInt = new int[GlobalClass.Size_Of_Int_In_Bytes];

byteToInt[0] = ( integerInBytes[0] << 24 );
byteToInt[1] = ( (integerInBytes[1]& 0xff) << 16);
byteToInt[2] = ( (integerInBytes[2]& 0xff) << 8 );
byteToInt[3] = ( integerInBytes[3]& 0xff );


System.out.println("byteToInt[0]="+byteToInt[0]);
System.out.println("byteToInt[1]="+byteToInt[1]);
System.out.println("byteToInt[2]="+byteToInt[2]);
System.out.println("byteToInt[3]="+byteToInt[3]);


result=byteToInt[0]|byteToInt[1]|byteToInt[2]|byteToInt[3];
return result;
}

public String IsThereAMessage()
{
    if(IsThereMessage=="true")
        return file.getAbsolutePath(); ;
    return "false";
}
}

```

```

package Steganography;

import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.Authenticator;
import javax.mail.BodyPart;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Multipart;
import javax.mail.PasswordAuthentication;
import javax.mail.SendFailedException;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Properties;
import java.util.Vector;

public class EmailUserInterface extends JFrame implements ActionListener{
    private JTextField fromField = new JTextField(); //שדה לכתיבת האימייל של המשתמש ששולח
    private JTextField toField = new JTextField(); //שדה לכתיבת האימייל של המשתמש שמקבל
    private JTextField subjectField = new JTextField(); //שדה לכתיבת הנושא של האימייל
    private JPasswordField passwordField = new JPasswordField(); //שדה לכתיבת הסיסמא של האימייל של המשתמש ששולח

```

```

private JButton chooseImageButton = new JButton();//קובץ לבחירת מסמך
private JButton removeImageButton = new JButton();//קובץ להסרת
DefaultListModel<String> l1 = new DefaultListModel<>(); //רשימה שמכילה את הקבצים המצורפים לאימייל
private JList<String> list = new JList<>(l1); //רשימה שמאכסנת ומציגה את l1
public String filename = "Empty file \n";//השם של הקובץ
public String filenamePath = "Empty file \n";//המסלול של המיקום של הקובץ
Vector<String> attachFiles = new Vector<String>();//וקטור המכיל את הקבצים המצורפים
public int addFileNum = 0;//סופר את מספר הקבצים המצורפים לאימייל
public boolean ImageChoose = false;//משתנה בוליאני שבודק עם צורפו קבצים לאימייל

JPanel paneedit;
MenuBar mbar;
Menu file,edit,format,font,font1,font2;
MenuItem item1,item2,item3,item4;
MenuItem item5,item6,item7,item8,item9,item10;
MenuItem fname1,fname2,fname3,fname4;
MenuItem fstyle1,fstyle2,fstyle3,fstyle4;
MenuItem fsize1,fsize2,fsize3,fsize4;

TextArea text;
Font f;
String months[] = { "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };
GregorianCalendar gcalendar;
String command = " ";
String str = " ";
String str1 = " ", str2 = " ", str3 = " ";
String str4 = " ";
String str6 = " ";
String str7 = " ", str8 = " ", str9 = " ";
int len1;
int i = 0;
int pos1;
int len;

EmailUserInterface() {
    InitializeUI();
}

public static void main() {
    SwingUtilities.invokeLater(new Runnable() {

```

```

        @Override
        public void run() {
            EmailUserInterface client = new EmailUserInterface();
            client.setVisible(true);
        }
    });
}

private void InitializeUI() {
    setTitle("Send E-mail Client");
    setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
    setSize(new Dimension(800, 800));
    setLocation(300, 100);
    getContentPane().setLayout(new BorderLayout());

    // Header Panel
    JPanel headerPanel = new JPanel();
    headerPanel.setLayout(new GridLayout(6, 2));
    headerPanel.add(new JLabel("Your Mail:"));
    headerPanel.add(fromField);

    headerPanel.add(new JLabel("Password:"));
    headerPanel.add(passwordField);

    headerPanel.add(new JLabel("Your Friend Mail:"));
    headerPanel.add(toField);

    headerPanel.add(new JLabel("Subject:"));
    headerPanel.add(subjectField);

    // Body Panel
    JPanel bodyPanel = new JPanel();
    bodyPanel.setLayout(new BorderLayout());
    bodyPanel.add(new JLabel("Message:"), BorderLayout.NORTH);
    bodyPanel.add(list, BorderLayout.CENTER);
    //add file button
    chooseImageButton.setIcon(new ImageIcon(getClass().getResource("../Interface
Images/AddFile.png"))));
    //chooseImageButton.setText("Add File");

```

```

bodyPanel.add(chooseImageButton, BorderLayout.WEST);

chooseImageButton.addActionListener(new chooseImageButtonActionListener());

//remove file button

removeImageButton.setIcon(new ImageIcon(getClass().getResource("../Interface
Images/Remove.png")));

//removeImageButton.setText("Remove");

bodyPanel.add(removeImageButton, BorderLayout.EAST);

removeImageButton.addActionListener(new removeImageButtonActionListener());


JPanel footerPanel = new JPanel();
footerPanel.setLayout(new BorderLayout());

JButton sendMailButton = new JButton("Send E-mail");
sendMailButton.addActionListener(new SendEmailActionListener());
footerPanel.add(sendMailButton, BorderLayout.SOUTH);

////////////////////////////////////

paneledit=new JPanel();
paneledit.setLayout(new FlowLayout());


mbar=new MenuBar();
setMenuBar(mbar);

edit=new Menu("Edit");
format=new Menu("Format");
font=new Menu("Font");
font1=new Menu("Font Style");
font2=new Menu("Size");

edit.add(item5=new MenuItem("Cut (Ctrl+X)"));
edit.add(item6=new MenuItem("Copy (Ctrl+C)"));
edit.add(item7=new MenuItem("Paste (Ctrl+V)"));
edit.add(item8=new MenuItem("Delete"));
edit.add(item10=new MenuItem("Select All (Ctrl+A)"));
edit.add(item9=new MenuItem("Time/Date"));
mbar.add(edit);

format.add(font);
format.add(font1);
format.add(font2);

```

```

font.add(fname1=new JMenuItem("Arial"));
font.add(fname2=new JMenuItem("Calibri"));
font.add(fname3=new JMenuItem("David"));
font.add(fname4=new JMenuItem("Georgia"));

font1.add(fstyle1=new JMenuItem("Regular"));
font1.add(fstyle2=new JMenuItem("Bold"));
font1.add(fstyle3=new JMenuItem("Italic"));
font1.add(fstyle4=new JMenuItem("Bold Italic"));

font2.add(fsize1=new JMenuItem("12"));
font2.add(fsize2=new JMenuItem("18"));
font2.add(fsize3=new JMenuItem("24"));
font2.add(fsize4=new JMenuItem("28"));

mbar.add(format);

item5.addActionListener(this);
item6.addActionListener(this);
item7.addActionListener(this);
item8.addActionListener(this);
item9.addActionListener(this);
item10.addActionListener(this);
fname1.addActionListener(this);
fname2.addActionListener(this);
fname3.addActionListener(this);
fname4.addActionListener(this);
fstyle1.addActionListener(this);
fstyle2.addActionListener(this);
fstyle3.addActionListener(this);
fstyle4.addActionListener(this);
fsize1.addActionListener(this);
fsize2.addActionListener(this);
fsize3.addActionListener(this);
fsize4.addActionListener(this);

text=new TextArea(26,60);

```

```

paneledit.add(text);

f=new Font("Arial",Font.PLAIN,18);
text.setFont(f);

getContentPane().add(headerPanel, BorderLayout.NORTH);
getContentPane().add(bodyPanel, BorderLayout.CENTER);
footerPanel.add(paneledit);
getContentPane().add(footerPanel, BorderLayout.SOUTH);
}

private class SendEmailActionListener implements ActionListener {
    SendEmailActionListener() {
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        final String sourceEmail = fromField.getText(); // requires valid Gmail id
        final String password = passwordField.getText(); // correct password for Gmail id
        final String toEmail = toField.getText(); // any destination email id

        Properties props = new Properties();
        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.port", "587");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");

        System.out.println("\n2nd ==> create Authenticator object to pass in
Session.getInstance argument..");

        Authenticator authentication = new Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(sourceEmail, password);
            }
        };

        if(text.getFont().getStyle()==1) {
            Session session = Session.getInstance(props, authentication);

```

```

        generateAndSendEmail(session, toEmail, subjectField.getText(), "<font
face=\""+text.getFont().getName()+"\" style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-
10)+"px\"><b> + text.getText() + "</b></font>"); // הוהדעה בגוף של
        System.out.println("<font face=\""+text.getFont().getName()+"\"
style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-10)+"px\"> + text.getText() + "</font>");
        // המייל

    }

    else if(text.getFont().getStyle()==2) {

        Session session = Session.getInstance(props, authentication);

        generateAndSendEmail(session, toEmail, subjectField.getText(), "<font
face=\""+text.getFont().getName()+"\" style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-
10)+"px\"><i> + text.getText() + "</i></font>"); // הוהדעה בגוף של
        System.out.println("<font face=\""+text.getFont().getName()+"\"
style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-10)+"px\"> + text.getText() + "</font>");
        // המייל

    }

    else if(text.getFont().getStyle()==3) {

        Session session = Session.getInstance(props, authentication);

        generateAndSendEmail(session, toEmail, subjectField.getText(), "<font
face=\""+text.getFont().getName()+"\" style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-
10)+"px\"><b><i> + text.getText() + "</i></b></font>"); // הוהדעה בגוף של
        System.out.println("<font face=\""+text.getFont().getName()+"\"
style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-10)+"px\"> + text.getText() + "</font>");
        // המייל

    }

    else{

        Session session = Session.getInstance(props, authentication);

        generateAndSendEmail(session, toEmail, subjectField.getText(), "<font
face=\""+text.getFont().getName()+"\" style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-
10)+"px\"> + text.getText() + "</font>"); // הוהדעה בגוף של
        System.out.println("<font face=\""+text.getFont().getName()+"\"
style=\""+text.getFont().getStyle()+"\" size=\""+(text.getFont().getSize()-10)+"px\"> + text.getText() + "</font>");
        // המייל

    }

}

public void generateAndSendEmail(Session session, String toEmail, String subject, String body)
{

    try {

        //if mail have attachments

        if (ImageChoose && addFileNum != 0) {

            System.out.println("\n3rd ==> generateAndSendEmail() starts..");

            MimeMessage MainMessage = new MimeMessage(session);

            MainMessage.addHeader("Content-type", "text/HTML;

charset=UTF-8");

            MainMessage.addHeader("format", "flowed");

```



```

        MainMessage.addHeader("Content-Transfer-Encoding", "8bit");

        MainMessage.setFrom(new InternetAddress(fromField.getText(),
subjectField.getText())); // נושא של המייל

        MainMessage.setReplyTo(InternetAddress.parse(fromField.getText(), false));

        MainMessage.setSubject(subject, "UTF-8");
        MainMessage.setSentDate(new Date());
        MainMessage.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(toEmail, false));

        // creates message part
        MimeBodyPart messageBodyPart = new MimeBodyPart();
        messageBodyPart.setContent(body, "text/html");

        // creates multi-part
        Multipart multipart = new MimeMultipart();
        multipart.addBodyPart(messageBodyPart); //set message text

        // adds attachments
        if (attachFiles != null && addFileNum > 0) {
            for (int i = 0; i < addFileNum; i++) {
                MimeBodyPart attachPart = new MimeBodyPart();

                try {
                    attachPart.attachFile(attachFiles.elementAt(i));
                } catch (IOException ex) {
                    ex.printStackTrace();
                }

                multipart.addBodyPart(attachPart);
            }
        }

        DataSource source = new FileDataSource(filenamePath); //file
        path

        messageBodyPart.setDataHandler(new DataHandler(source));
        messageBodyPart.setFileName(filename); // file name
        // Trick is to add the content-id header here
        messageBodyPart.setHeader("Content-ID", "image_id");
        multipart.addBodyPart(messageBodyPart);

```

```

the email body..");

System.out.println("\n4th ==> third part for displaying image in

messageBodyPart = new MimeBodyPart();
messageBodyPart.setContent("<br>" + "<img
src='cid:image_id'>", "text/html");

multipart.addBodyPart(messageBodyPart);
MainMessage.setContent(multipart);
System.out.println("\n4th ==> third part for displaying image in

the email body..");

System.out.println("\n5th ==> Finally Send message..");

// Finally Send message
try {
    Transport.send(MainMessage);

    System.out.println("\n6th ==> Email Sent
Successfully With Image Attachment. Check your email now..");

    System.out.println("\n7th ==>
generateAndSendEmail() ends..");

    JOptionPane.showMessageDialog(null, "Mail send
successfully", "Mail", JOptionPane.INFORMATION_MESSAGE);

} catch (Exception e) { //error
    System.err.println("Error Sending: ");

    JOptionPane.showMessageDialog(null, "Wrong
username or password", "Error", JOptionPane.ERROR_MESSAGE);
    e.printStackTrace();
}

}

//if mail have not attachments
else {

    System.out.println("\n3rd ==> generateAndSendEmail() starts..");

    MimeMessage MainMessage = new MimeMessage(session);
    MainMessage.setFrom(new InternetAddress(fromField.getText(),
subjectField.getText())); // נושא של המייל
    MainMessage.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(toEmail, false));

    MainMessage.setSubject(subject);
    //MainMessage.setText(body, "text/html");

```

```

        MainMessage.setContent(body,"text/html");

        System.out.println("\n4th ==> third part for displaying image in
the email body..");

        System.out.println("\n5th ==> Finally Send message..");

        try {
            Transport.send(MainMessage);

            System.out.println("\n6th ==> Email Sent
Successfully. Check your email now..");

            System.out.println("\n7th ==>
generateAndSendEmail() ends..");

            JOptionPane.showMessageDialog(null, "Mail send
successfully", "Mail", JOptionPane.INFORMATION_MESSAGE);

        } catch (Exception e) { //error
            System.err.println("Error Sending: ");

            JOptionPane.showMessageDialog(null, "Wrong
username or password", "Error", JOptionPane.ERROR_MESSAGE);

            e.printStackTrace();
        }
    }

    } catch (MessagingException e) {
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}

}

private class chooseImageButtonActionListener implements ActionListener { // add item to attached to mail
    chooseImageButtonActionListener() {
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        JFileChooser chooser = new JFileChooser(); // בחירת קובץ שאותו רוצים לשלוח במייל
        chooser.setCurrentDirectory(new java.io.File(".");

```

```

        chooser.setDialogTitle("choosertitle");
        // chooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
        chooser.setAcceptAllFileFilterUsed(true);

        if (chooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {
            System.out.println("getCurrentDirectory(): " +
chooser.getCurrentDirectory());

            System.out.println("getSelectedFile() : " + chooser.getSelectedFile());
            ImageChoose = true;
            filenamePath = chooser.getSelectedFile().toString(); // get file path
            filename = chooser.getSelectedFile().getParent().toString(); // get file path

            without the name

            filename= filenamePath.replace(filename, ""); //get only the filename
            filename = filename.replace("\\", ""); //get only the filename(remove /)
            attachFiles.add(filenamePath);
            l1.add(addFileNum, filename);
            addFileNum++;

        }
        else {
            System.out.println("No Selection ");
            ImageChoose = false;
            filenamePath = "No file";
        }

        System.out.println(filenamePath);
        System.out.println(filename);

    }
}

```

```

private class removeImageButtonActionListener implements ActionListener { // remove selected item from
attached

```

```

        removeImageButtonActionListener() {
        }

```

```

@Override

```

```

public void actionPerformed(ActionEvent e) {
    int indexselected =list.getSelectedIndex();
    if(indexselected == -1)

```

```

        System.out.println("Choose File");

        else {

            l1.remove(indexselected);

            attachFiles.remove(indexselected);

            addFileNum--;

            System.out.println("remove " + indexselected );

        }

    }

}

```

```

public void actionPerformed(ActionEvent ae) {

    command = (String) ae.getActionCommand();

    if (command.equals("Cut (Ctrl+X)")) {

        str = text.getSelectedText();

        i = text.getText().indexOf(str);

        text.replaceRange(" ", i, i + str.length());

    }

    if (command.equals("Copy (Ctrl+C)")) {

        str = text.getSelectedText();

    }

    if (command.equals("Paste (Ctrl+V)")) {

        pos1 = text.getCaretPosition();

        text.insert(str, pos1);

    }

    if (command.equals("Delete")) {

        String msg = text.getSelectedText();

        i = text.getText().indexOf(msg);

        text.replaceRange(" ", i, i + msg.length());

    }

    if (command.equals("Time/Date")) {

        gcalendar = new GregorianCalendar();

        String h = String.valueOf(gcalendar.get(Calendar.HOUR));

        String m = String.valueOf(gcalendar.get(Calendar.MINUTE));

        String s = String.valueOf(gcalendar.get(Calendar.SECOND));
    }
}

```

```

        String date = String.valueOf(gcalendar.get(Calendar.DATE));
        String mon = months[gcalendar.get(Calendar.MONTH)];
        String year = String.valueOf(gcalendar.get(Calendar.YEAR));
        String hms = "Time" + " - " + h + ":" + m + ":" + s + " Date" + " - " + date + " " + mon + "
" + year;

        int loc = text.getCaretPosition();
        text.insert(hms, loc);
    }
    if (command.equals("Arial")) {

        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font("Arial", fontStyle, fontSize);
        text.setFont(f);
    }
    if (command.equals("Calibri")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font("Calibri", fontStyle, fontSize);
        text.setFont(f);
    }
    if (command.equals("David")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font("David", fontStyle, fontSize);
        text.setFont(f);
    }

    if (command.equals("Georgia")) {
        String fontName = f.getName();

```

```

        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font("Georgia", fontStyle, fontSize);
        text.setFont(f);
        System.out.println(f.getFamily());
    }

    if (command.equals("Regular")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font(fontName, Font.PLAIN, fontSize);
        text.setFont(f);
    }

    if (command.equals("Bold")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font(fontName, Font.BOLD, fontSize);
        text.setFont(f);
    }

    if (command.equals("Italic")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font(fontName, Font.ITALIC, fontSize);
        text.setFont(f);
    }

    if (command.equals("Bold Italic")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();

```

```

        int fontStyle = f.getStyle();

        f = new Font(fontName, Font.BOLD | Font.ITALIC, fontSize);
        text.setFont(f);
    }

    if (command.equals("12")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font(fontName, fontStyle, 12);
        text.setFont(f);
    }

    if (command.equals("18")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font(fontName, fontStyle, 14);
        text.setFont(f);
    }

    if (command.equals("24")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();

        f = new Font(fontName, fontStyle, 18);
        text.setFont(f);
    }

    if (command.equals("28")) {
        String fontName = f.getName();
        String fontFamily = f.getFamily();
        int fontSize = f.getSize();
        int fontStyle = f.getStyle();
    }

```



```

        f = new Font(fontName, fontStyle, 20);
        text.setFont(f);
    }
    if (command.equals("Select All (Ctrl+A)")) {
        String strText = text.getText();
        int strLen = strText.length();
        text.select(0, strLen);
    }
}
}

```

```

package Steganography;
import java.awt.image.BufferedImage;
import java.security.PrivateKey;

```

```

public class Encode
{
    BufferedImage ImageWithMessage;
    public Encode(String Message)
    {
        System.out.println(GlobalClass.bytes.length);
        Hide(GlobalClass.bytes,Message);
    }

    public void Hide(byte[] bytes,String Text)
    {
        byte[] TextInBytes=ConvertTextToBytes(Text);//הכיל עם החתימה והכל/
        הייתה שגיאה ולא צלחה השיטה//if(TextInBytes==null)
        {
            System.out.println("Error accured while converting text to bytes array");
            return;
        }
        int Offset=0;//מיקום בתמונה בתים
        System.out.println(TextInBytes[Offset]);
        System.out.println(bytes.length+" is the bytes length");
        for (int i = 0; i <TextInBytes.length; i++)

```

```

        for(int j=7; j>=0; j--)
        {
            byte BitValue=(byte)((TextInBytes[j]>>j) & 1); // לוקח סיבית אחת מתוך בית של
            System.out.println(bytes[Offset]);
            bytes[Offset] = (byte)((bytes[Offset] & 0xFE) | BitValue); // מכניס את סיבית
            Offset++; // מתקדם לבית הבא
        }
        System.out.println("Offset="+Offset);
    }

    public byte[] ConvertTextToBytes(String Text) // מחזיר את הטקסט + חתימה בבתים
    {

```

```

        byte[] StartSign="$<".getBytes(); // בפיירוט $=36 +60
        byte[] MessageLengthInBytes=convertIntToByteArray(Text.length()); // אורך המסר בבתים
        byte[] TextInBytes=(Text).getBytes(); // המסר עצמו בבתים

        //read the private key
        //put it in GenerateSignature
        GetKey Read_Key; // מייצג את מסלול המפתח הפרטי
        PrivateKey PrivateKey;

        if( (Read_Key=new GetKey()).IsFileCanBeRead(Read_Key.getKeyPath()) ) // בודק אם המסלול תקין
            PrivateKey=Read_Key.getPrivateKey(); // אם כן מציב את המפתח הפרטי
        else
        {
            GlobalClass.what_message="Private Key";
            return null;
        }

        //Create a signature
        byte [] SigInBytes=new GenerateSignature(TextInBytes,PrivateKey).getSignature(); // מחזיר את
        המסר ומפתח פרטי בתור חתימה

        byte[] SigLengthInBytes=convertIntToByteArray(SigInBytes.length); // אורך החתימה בבתים

        byte[] TotalText=new
        byte[StartSign.length+SigLengthInBytes.length+SigInBytes.length+MessageLengthInBytes.length+TextInBytes.lengt
        h]; // דרך הופעת הנתונים של הטקסט הסופי שיופיע בתמונה

        // סימן ההתחלה והוא $ > ולאחריו יופיע אורך החתימה בבתים ואז החתימה עצמה ואז אורך הטקסט בבתים ואז
        המסר הסודי בעצמו

```

```

System.out.println("TotalText Length="+TotalText.length);//אורך כל מה שציינתי מעל
System.out.println("StartSign.length="+StartSign.length);//אורך סימן ההתחלה והוא תמיד 2 תאים
System.out.println("SigLengthInBytes.length="+SigLengthInBytes.length);//אורך החתימה בבתים
System.out.println("SigInBytes.length="+SigInBytes.length);//אורך החתימה
System.out.println("MessageLengthInBytes.length="+MessageLengthInBytes.length);//אורך
המסר בבתים

System.out.println("TextInBytes.length="+TextInBytes.length);//המסר בבתים
//Combine the StartSign to the TotalText
System.arraycopy(StartSign,0,TotalText,0,StartSign.length);//מוסיף את סימן ההתחלה

//Combine the Length Of Signature to the TotalText
System.out.println("Start Copying Length Signature from="+StartSign.length);

System.arraycopy(SigLengthInBytes,0,TotalText,StartSign.length,SigLengthInBytes.length);//הוספת אורך
החתימה

//Combine the Signature to the TotalText
System.out.println("Start Copying Signature from="+StartSign.length);

System.arraycopy(SigInBytes,0,TotalText,(StartSign.length+SigLengthInBytes.length),SigInBytes.length);//
הוספת החתימה

//Combine the TextLength into TotalText
System.out.println("Start Copying MessageLengthInBytes
from="+ (SigInBytes.length+StartSign.length));

System.arraycopy(MessageLengthInBytes,0,TotalText,(StartSign.length+SigLengthInBytes.length+SigInBy
tes.length),MessageLengthInBytes.length);//הוספת אורך המסר

//Combine the TextInBytes into TotalText
System.out.println("Start Copying TextInBytes
from="+ (StartSign.length+SigInBytes.length+MessageLengthInBytes.length));

System.arraycopy(TextInBytes,0,TotalText,(StartSign.length+SigLengthInBytes.length+SigInBytes.length+
MessageLengthInBytes.length),TextInBytes.length);//הוספת המסר הסודי

return TotalText;//מחזיר את מערך הבתים של כל הטקסט שצריך להיות מוחבא
}

public static byte[] convertIntToByteArray(int integer)
{
    System.out.println("Length of Message "+integer);
    byte[] bytes = new byte[GlobalClass.Size_Of_Int_In_Bytes];
    bytes[0] = (byte)(integer >> 24);
    bytes[1] = (byte)(integer >> 16);

```

```

bytes[2] = (byte)(integer >> 8);
bytes[3] = (byte)(integer);

System.out.println("bytes[0]="+bytes[0]);
System.out.println("bytes[1]="+bytes[1]);
System.out.println("bytes[2]="+bytes[2]);
System.out.println("bytes[3]="+bytes[3]);

return bytes;
    }
}

package Steganography;
import java.io.File;
import javax.swing.JFileChooser;
public class FileChooser //extends JPanel
{
    JFileChooser fc;//מאפשר את חלון הבחירה
    int returnVal;//מצב הפעולה
    File ImageFile;//שומר את קובץ התמונה
    public FileChooser(String Operation)
    {
        ImageFile=null;//איפוס קובץ
        fc = new JFileChooser();//יוצר חלון בחירה
        if(Operation.equals("Open"))//חלון נפתח
            returnVal=fc.showOpenDialog(null);//קבל ערך של הפתיחה
        else//נבחר קובץ
            returnVal=fc.showSaveDialog(null);//קבל ערך שמירה
    }
    public boolean IsFileChooserClosed()//האם נסגר חלון העיון
    {
        System.out.println(returnVal);
        if (returnVal == JFileChooser.APPROVE_OPTION)//אם לא סגור
        {
            return false;//שקר חלון העיון פתוח
        }
        else//סגור
        {
            System.out.println("Attachment cancelled by user.");//נסגר על ידי המשתמש
        }
    }
}

```

```

//Reset the file chooser for the next time it's shown.

//don't have to reset it because it can be more comfortable

return true;
}

public void setImageFile()//מציב את הקובץ הנבחר
{
    ImageFile = fc.getSelectedFile();//זהו המשתמש ומציב אותו במשתנה זה
}

public File getInputFile()//החזר את הקובץ
{
    return ImageFile;
}
}

package Steganography;

import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Security;

import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class GenerateKeys
{
    private PrivateKey PrivateKey;//מפתח פרטי
    private PublicKey PublicKey;//מפתח ציבורי

    public GenerateKeys()
    {
        try
        {
            Security.addProvider(new BouncyCastleProvider());//הספק של הארס אס איי והוא נקרא באונסי קאסל
            KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA", "BC");//מייצר מפתחות בשיטת ארס אס איי
            kpg.initialize(1024, new java.security.SecureRandom());//מאתחל מפתחות

            KeyPair kp = kpg.generateKeyPair();//מייצר את המפתחות
        }
    }
}

```

```

        PublicKey = kp.getPublic();//קבלת מפתח פומבי/
        PrivateKey = kp.getPrivate();//קבלת מפתח פרטי/

        ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("../Keys/Public
        Key.dat",false ) );//יצירת קובץ מפתח ציבורי/

        oos.writeObject(PublicKey);//הכנסת הדאטה לקובץ/
        oos.close();//סגירת קובץ/

        oos = new ObjectOutputStream(new FileOutputStream("../Keys/Private Key.dat", false ) );//יצירת קובץ
        מפתח פרטי

        oos.writeObject(PrivateKey);//הכנסת הדאטה לקובץ/
        oos.flush();//מוודא שהקובץ הולך למקום הנכון/

        oos.close();//סגירת קובץ/

    }
    catch( Exception e )//חריגה/
    {
        e.printStackTrace( System.out );
    }
}

public PrivateKey getPrivatkey();//החזר מפתח פרטי/
{
    return PrivateKey;
}

public PublicKey getPublickey();//חזר מפתח פומבי/
{
    return PublicKey;
}
}

package Steganography;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.Signature;

```

```

public class GenerateSignature
{
    byte[] Message;//הודעה של איש הקשר בבתים
    static byte [] sig;//חתימה של איש הקשר בבתים
    public GenerateSignature(byte[] message,PrivateKey PrivateKey)
    {
    try
    {
        Message=message;//המסר הסודי שהשתמש הקליד
        Signature signer = Signature.getInstance("RSA");//יוצר חתימה בשיטת אר אס איי

        signer.initSign(PrivateKey);//מכניס את המפתח הפרטי לחתימה
        signer.update( Message );//מכנס את הודעת המשתמש לחתימה
        sig = signer.sign();//מכניס את אורך כל החתימה והמסר בבתים
    }
    catch( Exception e )
    {
        e.printStackTrace( System.out );
    }
    }
    public byte[] getSignature();//החזר את החתימה
    {
        return sig;
    }
    static public boolean verification(byte[] Sig,PublicKey PublicKey,byte [] message)//השיטה בודקת התאמה בין
    המפתחות ומודיע בהתאם
    {
        boolean sigIsOK=false;
        try
        {
            Signature verifier = Signature.getInstance("RSA");//יוצר חתימה בשיטת אר אס איי

            verifier.initVerify( PublicKey );//מכניס את המפתח הפרטי לחתימה
            verifier.update( message );//מקבל את המסר הסודי
            sigIsOK = verifier.verify( Sig );//בודק תאימות בין המפתחות
            //SigIsOK ?(if(true))""(else):
            System.out.println( (sigIsOK ? "" : "does not")+ "verify" );//הודע אם אושר או לא
        }
        catch( Exception e )

```

```

    {
        e.printStackTrace( System.out );
    }
    return sigIsOK;
    }
}

```

```

package Steganography;

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.security.PrivateKey;
import java.security.PublicKey;

```

```

public class GetKey
{
    private String Path="../Keys/Private Key.dat";//מסלול של המפתח הפרטי
    private PrivateKey PrivateKey=null;//מפתח פרטי
    private PublicKey PublicKey=null;//מפתח ציבורי
    private Object Object = null;//משמש לטעינה של הקובץ

    כאשר הבנאי נקרא ריק כמו באחת הפונקציות באנקוד, נקרא המפתח הפרטי
    public GetKey(){

    public GetKey(String path)
    {
        Path=path;
    }

    האם הקובץ מפתח קריא
    public boolean IsFileCanBeRead(String Path)
    {
        File file=new File(Path);//הקובץ עם המסלול
    }
}

```



```

if(file.canRead())//האם קריא?
{
    ObjectInputStream inputStream = null;//איפוס של טעינה לקובץ
    try
    {
        //????????
        //there is a problem when i change the content in the private key
        //i'm trying to get a message that points to the problem
        //when the user tries to insert a private key instead of a public key
        inputStream = new ObjectInputStream(new FileInputStream(Path));//טעינה לקובץ
    }
    catch(EOFException e)
    {
        System.out.println("ends");//סוף הקובץ
    }
    catch (Exception e1)
    {
        e1.printStackTrace();//חריגה
    }

    try
    {
        Object = inputStream.readObject();//נסה לקרוא את הקובץ
    }
    catch(Exception e)
    {
        System.out.println("EXCEPTION");//חריגה
        return false;
    }

    if(Path!="../Keys/Private Key.dat")//אם המסלול הוא לא של המפתח
    {
        System.out.println("Public Key!!");//זה מפתח ציבורי
        try
        {
            PublicKey=(PublicKey)Object;//נסה לקבל את
        }
    }
}

```

ולחיצה את האובייקט שלו באובייקט המחלקה

הפרטי

המפתח הציבורי מהקובץ אותו טענו קודם

```

    }
    catch(ClassCastException e)//אין תוכן/
    {
        return false;//הייתה שגיאה במהלך הקאסטינג
    }
}
else
{
    System.out.println("Path="+Path);//הדפס את המסלול/
    PrivateKey=(PrivateKey)Object;//טען את המפתח הפרטי
}
try
{
    inputStream.close();//סוגר את הקובץ/
    return true;
}
catch (IOException e) {e.printStackTrace();}
}
else
{
    System.out.println("A Message window should be open");
    return false;
}
return true;

}

public String getKeyPath()//החזר את מסלולו של המפתח/
{
    return Path;
}

public PrivateKey getPrivateKey()//החזר את המפתח הפרטי/
{
    return PrivateKey;
}

public PublicKey getPublicKey()//החזר את המפתח הציבורי/
{

```

והמפתח לא נמצא

מהקובץ

```

        return PublicKey;
    }
}

```

```

package Steganography;

import java.awt.Image;
import java.awt.image.BufferedImage;
import java.awt.image.DataBufferByte;
import java.awt.image.WritableRaster;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

public class GlobalClass
{
    static BufferedImage img;
    static String what_message="Encode";
    static byte[] bytes;
    static final int Size_Of_StartSign = 2; //תמיד 2 כי משתמש בשני תווים בלבד

    static final int Size_Of_One_Char_In_EncodeText = 8; //the size of one char //1 char (8 bits) stored in 8 bytes. each last bit in each byte presents 1 bit in a char respectively
    static final int Size_Of_Int_In_Bytes = 4; //the size of integer in bytes

    public static void ConvertImageToBytes(Image i)
    {
        img=(BufferedImage)i; //טוען את התמונה
        bytes=accessBytes(img); //הופך את התמונה למערך פיקסלים בבתים
    }

    private static byte[] accessBytes(BufferedImage image)

```

```

    {
        WritableRaster raster = image.getRaster(); // צורת טעינה של פיקסלים מתוך תמונה טעונה
        DataBufferByte buffer = (DataBufferByte) raster.getDataBuffer(); // טעינה של הפיקסלים והפיתום
        למעך בתים על ידי קאסטינג לדטה בית
        return buffer.getData(); // מחזיר את מערך הביתים של הפיקסלים של התמונה
    }

    static public BufferedImage ReadImage(File Path) // המסלול על פי
    {
        BufferedImage ImageRead=null;
        try
        {
            ImageRead=ImageIO.read(Path); // מנסה לטעון את התמונה
        }
        catch (IllegalArgumentException e)
        {
            System.out.println("Path is null!"); // לא נמצאה תמונה במסלול
        }
        catch (IOException e)
        {
            System.out.println("Error occurred during reading!"); // חריגה
        }
        return ImageRead; // מחזיר תמונה טעונה
    }

    public static boolean ImageValidation(BufferedImage img) // בודק אם התמונה ריקה
    {
        if (img!=null)
            return true;
        System.out.println("false");
        return false;
    }
}

```

```

package Steganography;
import java.awt.Color;
import java.awt.Container;

```

```

import java.awt.Font;

import java.awt.Point;


import javax.swing.BoxLayout;
import javax.swing.ImageIcon;
import javax.swing.JDialog;
import javax.swing.JEditorPane;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;


import javax.swing.JScrollPane;

import javax.swing.ScrollPaneConstants;


public class Help extends JFrame //implements ComponentListener
{
    private static final long serialVersionUID = 1L;

    Container contentPane;

    JEditorPane editorPane;

    JScrollPane ScrollPane;

    JLabel titleLabel,Text;

    public Help(Interface i)
    {
        super("Help and support");

        Point location=i.getLocation();

        setLocation(location.x+50, location.y+70);


        setSize(550,423);

        //setResizable(false);

        setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);

        //Get Main Panel

        contentPane=getContentPane();

        //Set Layout

        setLayout(new BoxLayout(contentPane,BoxLayout.Y_AXIS));
    }
}

```

```

//Create ImageIcon for Title
ImageIcon icon = new ImageIcon(getClass().getResource("../Interface Images/Working with
keys.png"));

//create JLabels
titleLabel=new JLabel();
Text=new JLabel();
//set icon "AboutTitle" to titleLabel
titleLabel.setIcon(icon);

Text.setText("<Html><font><br>" +
            "<b>1.<u>What keys are?</u></b><br>" +
            "<b>2.<u>Why do we need keys?</u></b><br>" +
            "<b>3.<u>How to use keys?</u></b><br><br>" +

            "<u>What keys are?</u><br>" +
            "Keys are files with ".dat" extension which are used for security.<br><br>" +

            "<u>Why do we need keys?</u><br>" +
            "Every user have a pair of keys:<br>" +
            "• Public key-used by the user on decode operation.<br>" +
            "• Private key-used by the software on encode operation.<br>" +
            "The two keys are used for security which prevents:<br>" +
            "• Sending a message by impersonation to another person.<br>" +
            "• Decoding a message by a person who doesn't have
permission.<br><br>" +

            "<u>How to use keys?</u><br>" +
            "In order to create a new pair of keys, go to File-->New Keys.<br>" +
            "<b><u color=red>Note</u></b>: Creating a new pair of keys will replace
the previous pair.<br>" +

            "After the keys have been created, inform your contacts about<br>" +
            "the <b>new</b> public key which they will need in order to decode the
message.<br><br>" +

            "<u>Decode operation</u><br>" +
            "You need to have the public key of the user who sent you the picture<br>" +
            "when you are asked to insert a contact's public key.<br><br>"
);

Text.setFont(new Font("Cambria",Font.PLAIN,16));
Text.setForeground(Color.white);
Text.setForeground(Color.black);

```

```

        //Create JPanels
        JPanel p1=new JPanel();
        JPanel p2=new JPanel();
        //Create ScrollPane
        ScrollPane = new JScrollPane(p2);

ScrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER );

        ScrollPane.setBorder(null);
        //Add components to JPanels
        p1.add(TitleLabel);
        p2.add(Text);
        //Add JPanel to window
        add(p1);
        //Add scrollPane to window
        add(ScrollPane);

        setVisible(true);
    }
}

```

```

package Steganography;
import java.io.File;
import javax.swing.*;
import javax.swing.filechooser.*;

public class ImageFileView extends FileView {
    ImageIcon gifIcon = Utils.createImageIcon("../Icon Format/gifIcon.gif");//אייקון לפורמט גיף
    ImageIcon pngIcon = Utils.createImageIcon("../Icon Format/pngIcon.png");//אייקון לפורמט פי אן גי

    public String getName(File f) { //חזר את שם הקובץ
        return null; //let the L&F FileView figure this out
    }
}

```

```
}
```

```
public String getDescription(File f) { //החזר את תיאור הקובץ
```

```
    return null; //let the L&F FileView figure this out
```

```
}
```

```
public Boolean isTraversable(File f) { //האם הצליח או לא הצליח
```

```
    return null; //let the L&F FileView figure this out
```

```
}
```

```
public String getTypeDescription(File f) { //מחזיר את תיאור התמונה כלומר מה הסוג שלה
```

```
    String extension = Utils.getExtension(f); //מחלקת יוטילס
```

```
    String type = null; //יחזיר את סוג התמונה
```

```
    if (extension != null) { //אם קיימת סיומת לקובץ
```

```
    {
```

```
        if (extension.equals(Utils.gif)) { //והוא שווה לפורמט גיף
```

```
        {
```

```
            type = "GIF Image"; //סוג התמונה הוא גיף
```

```
        }
```

```
    } else
```

```
        if (extension.equals(Utils.png)) { //אם הסיומת שווה לפיאנג
```

```
        {
```

```
            type = "PNG Image"; //סוג התמונה הוא פי אנ גי
```

```
        }
```

```
    }
```

```
    return type; //החזר את סוג התמונה
```

```
}
```

```
public Icon getIcon(File f)
```

```
{
```

```
    String extension = Utils.getExtension(f); //מחזיר את סיומת התמונה
```

```
    Icon icon = null; //האייקון שיוחזר
```

```
    if (extension != null) { //אם נמצא סיומת
```

```
    {
```

```
        if (extension.equals(Utils.gif)) { //והיא שווה לפורמט גיף
```

```
        {
```

```
            icon = gifIcon; //האייקון של התמונה יהיה אייקון גיף
```



```

    }
    else

        if (extension.equals(Utils.png)) //ג'י' אן ג'י'
        {
            האייקון של התמונה יהיה פי אן ג'י'
            icon = pngIcon;
        }
    }
    החזר את האייקון;
    return icon;
}
}

package Steganography;
import java.io.File;
import javax.swing.filechooser.*;
public class ImageFilter extends FileFilter
{
    String filter;//תיאור הקובץ
    public ImageFilter(String f)
    {
        filter=f;
    }
    //Accept all directories and all gif, jpg, tiff, or png files.
    public boolean accept(File f) //סוג קובץ
    {
        if (f.isDirectory()) //האם הקובץ הוא תיקיה
            return true; //אם כן החזר אמת
        String extension = Utils.getExtension(f); //החזר את סיומת הקובץ
        if(extension==null) //אם אין כזו
        {
            תודיע לי שאין סיומת לקובץ;
            System.out.println("There is no extention");
        }
        else//אחרת
        {
            תבדוק האם הקובץ שווה לתמונה;
            if(filter=="Images")
            {
                if (extension.equals(Utils.gif) || extension.equals(Utils.png)) //אם היא שווה לפורמט
                {
                    ג'י' או פי אן ג'י'
                    return true; //החזר אמת
                }
                else {}
            }
            else//אחרת
        }
    }
}

```

```

        if(filter=="PublicKey")//מפתח שווה למפתח
            if (extension.equals(Utils.dat)) //האם קיימת סיומת של מפתח לקובץ זה
                return true;//קיים החזר אמת

        return false;//לא תמונה ולא מפתח וגם לא תיקיה החזר שקר
    }

    //The description of this filter
    public String getDescription()
    {

        return null; //"Just Images"}}
package Steganography;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class ImageSave
{
    String ImageFormat;//פורמט של תמונה
    public ImageSave(BufferedImage img,File path,String format)//מקבל תמונה טעונה ומסלול ופורמט
    {
        ImageFormat=format;//מציב את הפורמט במשתנה המחלקה
        File OutputFile=getFileName(path);//מסלול התמונה הכוללת שם התמונה וסיומת
        try
        {
            ImageIO.write(img, ImageFormat, OutputFile);//מנסה ליצור תמונה חדשה בשם שהתקבל וסיומת
            System.out.println("Saved");//הודע לי על הצלחת השמירה
        }
        catch (IOException e) //חריגה
        {
            e.printStackTrace();
            System.out.println("NOT Saved");//התמונה לא נשמרה
        }
    }
    private File getFileName(File p)//מחזיר מסלול תמונה עם סיומת
    {
        File Path; //מסלול חש

```

```

String ImageName = p.toString(); //שם התמונה עם כל המסלול
int period=ImageName.indexOf('.'); //get the index of "." if there is no "." the function returns -1

if(period!=-1)
{
    ImageName=ImageName.substring(0,period);
    System.out.println("Image Name "+ImageName);
}

Path=new File(ImageName+"."+ImageFormat);
return Path;
}
}

package Steganography;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.net.URL;

import javax.swing.*;
import javax.swing.border.Border;
import javax.swing.border.CompoundBorder;
import javax.swing.border.TitledBorder;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;

import javax.swing.text.AttributeSet;
import javax.swing.text.BadLocationException;

import javax.swing.text.PlainDocument;

public class Interface extends JFrame implements ActionListener, MouseListener, DocumentListener
{
    private static final long serialVersionUID = 1L;

    JLabel
    TitleLabel,ChooseActionCommand,ChooseFileCommand,DirectoryLabel,KeysInsert,PublicKey;// כותרת, בחר בפעולה
    שתוצאה, בחר תמונה, תיקיה, הכנס מפתח, מפתח

    JPanel TitlePanel,ActionPanel,FilePanel,TextPanel;

```

```

JRadioButton[] Action;

ButtonGroup group;

String ActionName;

JTextField TextPath;//מסלול תמונה

JTextArea Text;//המסר

CoustomizedButton Browse;//Browse button for image

JButton emailButton;

TitledBorder Title1,Title2;//Message to Encrypt: , Notes left:

CompoundBorder border1,border2,border3;

Border a,a2,b;

JScrollPane AreaJScrollPane;

int MaxLengthOfNotes=5000000,LenOfWritten,LenOfDeleted;

int NotesLeft=MaxLengthOfNotes;


Container contentPane;

SpringLayout layout;

JTextField PublicBox;

CoustomizedButton BrowsePublic,Encode,Decode;

Checks checks;

Object buttonEntered,ButtonPressed;

MenuBar Bar;

Help Help_W = null;

public Interface()
{
    //Window properties
    super("Steganography");
    setSize(726,238);


    //Experiment***
    Toolkit toolkit = Toolkit.getDefaultToolkit();
    Dimension scrnsize = toolkit.getScreenSize();
    int ScreenWidth,ScreenHeight;


    setLocationRelativeTo(null);
    ScreenWidth=(int)getLocation().x;
    ScreenHeight=(int) (scrnsize.getHeight()*0.1);


    setLocation(ScreenWidth, ScreenHeight);
    /**

```

```

        setDefaultCloseOperation(EXIT_ON_CLOSE);

        //Create a menu bar
        Bar = new MenuBar();

        //create headers buttons
        Menu FileMenu = new Menu("File");
        Menu HelpMenu = new Menu("Help");

        FileMenu.add(new MenuItem("New Keys"));
        FileMenu.addSeparator();
        FileMenu.add(new MenuItem("Exit"));

        HelpMenu.add(new MenuItem("View Help"));
        HelpMenu.addSeparator();
        HelpMenu.add(new MenuItem("About Steganography"));

        //add the headers buttons to the bar
        Bar.add(FileMenu);
        Bar.add(HelpMenu);

        this.setMenuBar(Bar);

        FileMenu.addActionListener(this);
        HelpMenu.addActionListener(this);

        layout = new SpringLayout();
        setLayout(layout);

        //Create Background Color
        Color Background=new Color(196,219,223);

        //Create Foreground Color
        Color Forground=new Color(49,133,135);
        //commands foreground color
        Color fg=new Color(184,66,102);

        URL url = getClass().getResource("../Interface Images/Title.png");

```

```

//create ImageIcon for title
ImageIcon icon = new ImageIcon(url);

//create JLabels
titleLabel=new JLabel();
titleLabel.setIcon(icon);

ChooseActionCommand=new JLabel("Choose your action below:");
ChooseFileCommand=new JLabel("");
DirectoryLabel=new JLabel("Directory :");

KeysInsert=new JLabel("Insert a public key of your contact");
PublicKey=new JLabel("Public Key:");

//create font to JLabels
Font f = new Font("tahoma", Font.BOLD, 15);

//Set font to JLabels
ChooseActionCommand.setFont(f);
ChooseFileCommand.setFont(f);
DirectoryLabel.setFont(f);
KeysInsert.setFont(f);
PublicKey.setFont(f);
ChooseActionCommand.setForeground(fg);

emailButton = new JButton(""); // Send email button
emailButton.setBackground(Color.white);
emailButton.setIcon(new ImageIcon(getClass().getResource("../Interface
Images/SendMail.png")));
emailButton.addActionListener(new ActionListener() { // הוּצַפְת מַאֲזִין
    public void actionPerformed(ActionEvent e) {
        new EmailUserInterface().main();; // יצירה של איימיל
    }
});

//Create JRadioButtons

```

```

        Action=new JRadioButton[2];

        Action[0]=new JRadioButton("Encode");

        Action[1]=new JRadioButton("Decode");


        //Create ButtonGroup

        group = new ButtonGroup();

        group.add(Action[0]);

        group.add(Action[1]);


        //Create JTextField

        TextPath=new JTextField(25);

        TextPath.setEditable(false);

        PublicBox=new JTextField(24);

        PublicBox.setEditable(false);


        //Create CoustomizedButton

        Browse=new CoustomizedButton("Browse");

        BrowsePublic=new CoustomizedButton("Browse");

        Encode=new CoustomizedButton("Encode");

        Decode=new CoustomizedButton("Decode");


        //Add MouseListeners


        //Create JTextArea

        Text=new JTextArea("",5,30);

        Text.setLineWrap(true); //go to the next line after there is no more space

        Text.setWrapStyleWord(true);


        //Limit the amount of characters the user can insert

        Text.setDocument(new JTextFieldLimit(4));


        Text.getDocument().addDocumentListener(this);


        //Create JScrollPane for JTextArea

        AreaJScrollPane = new JScrollPane(Text);

        AreaJScrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);

```

```

b = AreaJScrollPane.getBorder();

Title1 = BorderFactory.createTitledBorder(border2, "Message to Encrypt");
Title1.setTitleColor(fg);

Title2=BorderFactory.createTitledBorder(b, "Notes left: "+NotesLeft,5 , 5);
Title2.setTitleColor(fg);

Title1.setTitleFont(f);

a = BorderFactory.createEmptyBorder(0,0,0,0);

border2 = BorderFactory.createCompoundBorder(Title2,a);
border3 = BorderFactory.createCompoundBorder(Title1,a);
border1 = BorderFactory.createCompoundBorder(border3,border2);


AreaJScrollPane.setBorder(border1);


//Add Action Listeners
Action[0].addActionListener(this);
Action[1].addActionListener(this);


//Add Components to JPanels
titleLabel.setOpaque(true);

contentPane = getContentPane(); //Get main panel


//SetComponentes to window

layout.putConstraint(SpringLayout.HORIZONTAL_CENTER,
titleLabel,5,SpringLayout.HORIZONTAL_CENTER, contentPane);

layout.putConstraint(SpringLayout.NORTH, titleLabel,5,SpringLayout.NORTH, contentPane);

contentPane.add(titleLabel);
contentPane.add(emailButton);

ActionPanel=new JPanel();

ActionPanel.setLayout(new BorderLayout(ActionPanel,BoxLayout.Y_AXIS));

JPanel InnerActionPanel=new JPanel();

InnerActionPanel.setLayout(new BorderLayout(InnerActionPanel,BoxLayout.Y_AXIS));

ActionPanel.add(ChooseActionCommand);

```



```

        InnerActionPanel.add(Box.createVerticalStrut(0));

        InnerActionPanel.add(Box.createHorizontalStrut(15));

        InnerActionPanel.add(Action[0]);

        InnerActionPanel.add(Action[1]);

        ActionPanel.add(InnerActionPanel);


        //Set Background to components
contentPane.setBackground(Background);

        TitleLabel.setBackground(Background);

        ActionPanel.setBackground(Background);

        InnerActionPanel.setBackground(Background);

        Action[0].setBackground(Background);

        Action[1].setBackground(Background);

        AreaJScrollPane.setBackground(Background);


        //Set Foreground to components
        ///

        ChooseActionCommand.setForeground(fg);

        Action[0].setForeground(Foreground);

        Action[1].setForeground(Foreground);

        ChooseFileCommand.setForeground(fg);

        DirectoryLabel.setForeground(Foreground);

        KeysInsert.setForeground(fg);

        PublicKey.setForeground(Foreground);


        layout.putConstraint(SpringLayout.WEST, ActionPanel,0, SpringLayout.WEST, contentPane);

        layout.putConstraint(SpringLayout.NORTH, ActionPanel,95, SpringLayout.NORTH, TitleLabel);

        contentPane.add(ActionPanel);


        setVisible(true);

        validate();

        Browse.addMouseListener(this);

        BrowsePublic.addMouseListener(this);

        Encode.addMouseListener(this);

        Decode.addMouseListener(this);

        checks=new Checks(this);

```

```

    }

    public void actionPerformed(ActionEvent e)
    {
        String Command=e.getActionCommand();//מקבל את הפעולה

        if(Command=="Exit")//סוגר חלון
            System.exit(0);
        else
            if(Command=="New Keys")//יצירת מפתחות חדשים
            {
                new Message(this,"Keys");//מודיע עך כך למשתמש
            }
            else
                if(Command=="View Help")//חלון עזרה
                {
                    if(Help_W==null)
                        Help_W=new Help(this);
                    Help_W.setVisible(true);
                }
                else
                    if(Command=="About Steganography")//חלון אודות
                    {
                        new About(this);
                    }

        System.out.println(getSize());

        if(e.getSource()== Action[0] || e.getSource()== Action[1])//אם נבחרו אחד מכפתורי הרדיו
        {
            ActionName=e.getActionCommand();//מקבל את שם הפעולה
            ChooseFileCommand.setText("Select an image to "+ActionName+":");//בחר תמונה

            להצפנה/פענוח

            setFileChoose();
            setComponentsVisibilityToFalse();

            TextPath.setText(null);
            PublicBox.setText(null);
            setSize(726,308);
        }
    }

```

```

    }

    validate();
}

public void setFileChoose()//השיטה מסדרת עבור ההצפנה והפענוח את הרכיבים הבאים:
{
    contentPane.remove(ChooseFileCommand);//לייבל של בחר תמונה להצפנה/פענוח
    contentPane.remove(DirectoryLabel);//לייבל תיקיה
    contentPane.remove(TextPath);//מסלול התמונה שנבחרה
    contentPane.remove(Browse);//כפתור עיון לבחירת תמונה

    //the X:      side of contentPane      the distance      side of ChooseFileCommand
    layout.putConstraint(SpringLayout.WEST, ChooseFileCommand,7,SpringLayout.WEST,
contentPane);//מסדר את לייבל בחירת קובץ בצד שמאל של המיכל
    //the Y:      side of ActionPanel      the distance      side of ChooseFileCommand
    layout.putConstraint(SpringLayout.NORTH, ChooseFileCommand,10,SpringLayout.SOUTH,
ActionPanel);//ממקם את לייבל בחירת קובץ מתחת פאנל הפעולה
    contentPane.add(ChooseFileCommand);

    layout.putConstraint(SpringLayout.WEST, DirectoryLabel,23,SpringLayout.WEST,
contentPane);//מסדר את לייבל התיקיה בצידו השמאלי של המיכל
    layout.putConstraint(SpringLayout.NORTH, DirectoryLabel,25,SpringLayout.NORTH, ChooseFileCommand);//
contentPane.add(DirectoryLabel);

    layout.putConstraint(SpringLayout.WEST, TextPath,85,SpringLayout.WEST, DirectoryLabel);
    layout.putConstraint(SpringLayout.NORTH, TextPath,27,SpringLayout.NORTH, ChooseFileCommand);// מסלול
של התמונה בתיקיה
    contentPane.add(TextPath);

    layout.putConstraint(SpringLayout.WEST, Browse,300,SpringLayout.WEST, TextPath);
    layout.putConstraint(SpringLayout.NORTH, Browse,14,SpringLayout.NORTH, ChooseFileCommand);
    contentPane.add(Browse);
}

public void setComponentsVisibilityToFalse()
{
    AreaJScrollPane.setVisible(false);
    KeysInsert.setVisible(false);
    PublicKey.setVisible(false);
    PublicBox.setVisible(false);
    BrowsePublic.setVisible(false);
    Encode.setVisible(false);
}

```

```

        Decode.setVisible(false);
    }

    public void SetMessageBoxOrKeys()//הפעולה שנבחרה על פי
    {

        if(ActionName=="Encode")//נבחר הצפנה
        {

            contentPane.remove(AreaJScrollPane);//מוחק את אזור הגלילה הקיים
            //מסדר רכיבים לאיפשר כתיבת המסר
            layout.putConstraint(SpringLayout.WEST, AreaJScrollPane,23,SpringLayout.WEST,
contentPane);
            layout.putConstraint(SpringLayout.NORTH, AreaJScrollPane,60,SpringLayout.NORTH, Browse);
            contentPane.add(AreaJScrollPane);

            AreaJScrollPane.setVisible(true);

            KeysInsert.setVisible(false);
            PublicKey.setVisible(false);
            PublicBox.setVisible(false);
            PublicBox.setText(null);
            BrowsePublic.setVisible(false);
            Encode.setVisible(false);
            Decode.setVisible(false);

        }
        else//נבחר פענוח
        {

            if(AreaJScrollPane.isVisible())
            {

                AreaJScrollPane.setVisible(false);

            }

            SetKeys(DirectoryLabel);//סדר רכיבים לאיפשר בחירת מפתחות
        }

        validate();

    }

    public void SetKeys(Component comp)
    {

```

```

contentPane.remove(KeysInsert);//Insert a public key of your contact
contentPane.remove(PublicKey);//Public key:
contentPane.remove(PublicBox);//[-----] path of key
contentPane.remove(BrowsePublic);//browse button

layout.putConstraint(SpringLayout.NORTH, KeysInsert,15, SpringLayout.SOUTH, comp);
layout.putConstraint(SpringLayout.WEST, KeysInsert,7, SpringLayout.WEST, contentPane);
contentPane.add(KeysInsert);

```

```

layout.putConstraint(SpringLayout.WEST, PublicKey,23, SpringLayout.WEST, contentPane);
layout.putConstraint(SpringLayout.NORTH, PublicKey,25, SpringLayout.NORTH, KeysInsert);
contentPane.add(PublicKey);

```

```

layout.putConstraint(SpringLayout.WEST, PublicBox,96, SpringLayout.WEST, PublicKey);
layout.putConstraint(SpringLayout.NORTH, PublicBox,27, SpringLayout.NORTH, KeysInsert);
contentPane.add(PublicBox);

```

```

        layout.putConstraint(SpringLayout.WEST, BrowsePublic,288, SpringLayout.WEST, PublicBox);
layout.putConstraint(SpringLayout.NORTH, BrowsePublic, 14, SpringLayout.NORTH, KeysInsert);
contentPane.add(BrowsePublic);

```

```

        KeysInsert.setVisible(true);
        PublicKey.setVisible(true);
        PublicBox.setVisible(true);
        BrowsePublic.setVisible(true);

        validate();
    }
    public void SetEncodeOrDecode()//מסדר כפתור הצפנה/פענוח בהתאם
    {
        if(!PublicBox.getText().isEmpty())
        {

            if(ActionName=="Encode")
                SetButton(Encode);

            else

```

```

        SetButton(Decode);
    }

}

public void SetButton(CoustomizedButton button)//סידור כפתור מעוצב הצפנה/פענוח
{
    contentPane.remove(button);

    Component comp;

    if(ActioName=="Encode")//במצב הצפנה
    {
        comp=AreaJScrollPane;//מקום כתיבת המסר
        layout.putConstraint(SpringLayout.WEST, button,205,SpringLayout.EAST, comp);//מסדר בצד הימני/
        layout.putConstraint(SpringLayout.NORTH, button,25,SpringLayout.SOUTH, comp);//התחתון מתחת
        לאיזור הכתיבה
        contentPane.add(button);
    }
    else//פענוח
    {
        layout.putConstraint(SpringLayout.EAST, button,250,SpringLayout.WEST, BrowsePublic);//מסדר בצד
        הימני
        layout.putConstraint(SpringLayout.NORTH, button,25,SpringLayout.SOUTH, PublicKey);//מתחת
        לשורת המפתח הציבורי
        contentPane.add(button);
    }
    button.setVisible(true);
    validate();
}

class JTextFieldLimit extends PlainDocument
{
    private static final long serialVersionUID = 1L;
    private int limit;

    JTextFieldLimit(int limit)//מקבל את מספר התווים שניתן להכניס/
    {
        this.limit = limit;
    }

    public void insertString(int offset, String str, AttributeSet attr) throws BadLocationException
    {
        String oldText=Text.getText();//המסר שהוקלד בעבר עבור המשתמש/
        System.out.println(str.charAt(0));
    }
}

```

```

        super.insertString(offset, str, attr);
        if(checks.CheckTextMessage())//האם נכתב מסר/
            if (Text.getText().length() > limit) // אם כמות הטקסט גדולה מכמות התווים שמותר
                Text.setText(oldText);//השאר בטקסט הישן/
            else
            {
                if(!Encode.isVisible())//מוצג לא היה מוצג/
                {
                    Encode.setVisible(true);//הצג אותו/
                    SetButton(Encode);
                    setSize(726,508);
                }
            }
        }
    }
}

public void removeUpdate(DocumentEvent e) //כשנמחק תו/
{
    if(Text.getText().length()==0)
    {
        Encode.setVisible(false);
        Decode.setVisible(false);
    }

    System.out.println("remove");

    NotesLeft+=e.getLength();
    Title2.setTitle("Notes left: "+NotesLeft);

    AreaJScrollPane.setBorder(b);
    AreaJScrollPane.setBorder(border1);

    validate();
}

public void changedUpdate(DocumentEvent e) {}
public void insertUpdate(DocumentEvent e) //כשנכתב תו/
{
    System.out.println("add");
}

```

```

        NotesLeft-=e.getLength();

        Title2.setTitle("Notes left: "+NotesLeft);


        System.out.println(NotesLeft);

        AreaJScrollPane.setBorder(b);
AreaJScrollPane.setBorder(border1);
validate();
    }


    public void mouseClicked(MouseEvent e) {}
    //Balloon tool tip
    public void mouseEntered(MouseEvent e)
    {
        buttonEntered=e.getSource();
        if(e.getSource()==Browse)
            checks.MouseEntered("Browse");
        else
            if(e.getSource()==BrowsePublic)
                checks.MouseEntered("BrowsePublic");
    }


    public void mouseExited(MouseEvent e)
    {
        buttonEntered=null;
        checks.mouseExited();
    }

    public void mousePressed(MouseEvent e)
    {
        ButtonPressed=e.getSource();
    }

    public void mouseReleased(MouseEvent e)
    {
        if(ButtonPressed==buttonEntered)
            if(e.getSource()==Browse)
                checks.mouseReleased("Browse");
            else
                if(e.getSource()==BrowsePublic)

```



```

        checks.mouseReleased("BrowsePublic");

    else

        if(e.getSource()==Encode)

            checks.mouseReleased("Encode");

        else

            if(e.getSource()==Decode)

                checks.mouseReleased("Decode");

    }

}

```

```

package Steganography;
import javax.swing.JOptionPane;

```

```

public class Message
{
    int Command;
    public Message(Interface MainWindow,String Operation)//לאירועים בהתאם הודעות
    {

        if(Operation=="ImageError")
            JOptionPane.showMessageDialog(MainWindow,"<html><font color=red size=4>
Image can't be read </font> </html>","Pay attention",JOptionPane.WARNING_MESSAGE);//ניתן ולא ריק כשקובץ לקריאה
        else
            if(Operation=="SmallImage")
                JOptionPane.showMessageDialog(MainWindow,"<html><font color=red size=4>
Image is too small</font></html>","Pay attention",JOptionPane.WARNING_MESSAGE);//בשביל מדי קטנה כשתמונה מסר בה להכניס
            else
                if(Operation=="KeyError")
                    JOptionPane.showMessageDialog(MainWindow,"<html><font color=red
size=4> Check your contact's public key </font> </html>","Pay
attention",JOptionPane.WARNING_MESSAGE);//חוקי לא כשהמפתח
                else
                    if(Operation=="Private Key")
                        JOptionPane.showMessageDialog(MainWindow,"<html><font
color=red size=4> Keys are damaged.</font> <br> Create new ones,go to menu-bar: File-->New keys </html>","Pay
attention",JOptionPane.WARNING_MESSAGE);//פגומים מפתחות
                    else
                        if(Operation=="Keys")
                        {
                            Command=JOptionPane.showConfirmDialog(MainWindow,"<html> <font
size=4 color=red> Creating new keys will replace the previous ones </font> </html> הן Are you sure? ","Pay
Attention",JOptionPane.YES_NO_OPTION);//חדשים במפתחות מפתחות החלפת
                            CheckCommand();//בהתאם חדשים מפתחות של יצירה
                        }
                        else
                            if(Operation=="Encode")
                            {
                                JOptionPane.showMessageDialog(MainWindow,"Message has
been encrypted","Confrim saving",JOptionPane.INFORMATION_MESSAGE);//הצפנה צלחה
                            }
                            else
                                //Decode
                                if(Operation=="false")

                                    JOptionPane.showMessageDialog(MainWindow,"<html> Message <font size=4> not found </font><br>
Check your image/contact's public key.</html>","Pay attention",JOptionPane.WARNING_MESSAGE);//מסר נמצא לא
                                    בתמונה

                                else
                                    {

```

```

        System.out.println(Operation);

        JOptionPane.showMessageDialog(MainWindow,"<html> Message <font size=4> found </font> <br>Path:"
+Operation + " </html>","Pay attention",JOptionPane.WARNING_MESSAGE);//מסר נמצא
    }
}

public boolean CheckCommand()
{
    if(Command==JOptionPane.YES_OPTION)//אם כן יצר מפתחות
    {
        JOptionPane.showMessageDialog(null,"Keys have been created
        successfully","Confrim generation",JOptionPane.INFORMATION_MESSAGE);//בהצלחה נוצרו המפתחות
        System.out.println("New Keys Created");//כך על הודעה
        new GenerateKeys();//יצר מפתחות
        return true;//אמת נוצרו מפתחות
    }
    else
    {
        if(Command==JOptionPane.NO_OPTION)//לא אם
        {
            System.out.println("Keys generation canceled");//את ביטל שהשתמש תודיע
            חדשים מפתחות של היצירה
        }
        return false;//חדשים מפתחות נוצרו לא שקר החזר
    }
}

```

```

package Steganography;

```

```

import java.io.*;
import java.net.*;
import java.util.*;

public class Server
{
    private ServerSocket serversocket;//the ServerSocket we'll use for accepting new connections

    private Hashtable outputStream = new Hashtable();// a mapping from sockets to DataOutputStreams. this
    will help us avoid having to create a DataOutputStream each time we want to write to a stream

    public Server(int port) throws IOException
    {
        listen(port);
    }

    private void listen(int port) throws IOException
    {
        serversocket = new ServerSocket(port);//create the ServerSocket

        System.out.println("Listening on: " + serversocket);//Listening started

        //keep accepting connections forever

        while(true)
        {
            Socket socket = serversocket.accept();//get the next incoming connection

            System.out.println("connection from" + socket);//connection confirmed
        }
    }
}

```

```

        DataOutputStream dataout = new
DataOutputStream(socket.getOutputStream()); //create a DataOutputStream for writing data to other side

        outputStream.put(socket, dataout); //save this stream so we don't need to make it again

        //create a new thread for this connection, and then forget about it
        new ServerThread(this, socket); //send the server and the socket
    }

}

//get an enumeration of all the outputStreams, one for each client connected to us
Enumeration getOutputStreams()
{
    return outputStream.elements(); //החרת מספר הסטרימים
}

//send a message to all client (utility routine)
void sendToAll(String message)
{
    //we synchronize on this because another thread might be calling removeConnection() and this
    //could make problems when trying to walk through the list
    synchronized(outputStream)
    {
        //התחל ממקום ראשון בטבלת ההאש והמשך לסטרים הבא
        for(Enumeration e = getOutputStreams(); e.hasMoreElements(); ) //for each client
        {
            DataOutputStream dataout = (DataOutputStream)e.nextElement(); //get the
output stream

            //and send a message
            try
            {
                dataout.writeUTF(message);
            }
            catch(IOException ex)
            {
                System.out.println(ex);
            }
        }
    }
}

//remove a socket, and it's corresponding output stream, from our list. this is usually called by a connection
//thread that has discovered that the connection to the client is dead
void removeConnection(Socket socket)
{
    synchronized(outputStream)
    {

```

```

        System.out.println("removing connection to: " + socket);
        outputStream.remove(socket);//remove from hashtable
        //make sure it's closed
        try
        {
            socket.close();
        }
        catch(IOException ex)
        {
            System.out.println("error closing: " + socket);
            ex.printStackTrace();
        }
    }

    //main routine
    static public void main(String args[]) throws Exception
    {
        int port = 6000;
        new Server(port);
    }
}

```

```

package Steganography;
import java.io.*;
import java.net.*;

public class ServerThread extends Thread
{

    private Server server;//the server that spawned us
    private Socket socket;//the socket connected to our client

```

```

public ServerThread(Server server, Socket socket)
{
    //save the parameters
    this.server = server;
    this.socket = socket;
    //start up the thread
    start();
}

//this runs in a separate thread when start() is called in the constructor
public void run()
{
    try
    {
        DataInputStream datain = new DataInputStream(socket.getInputStream()); //create a
DataInputStream for communication, the client is using a DataOutputStream to write to the server
        while(true)
        {
            String message = datain.readUTF(); //read the next message
            System.out.println("Sending: " + message); //message confirmed
            //send the message to all the clients
            server.sendToAll(message);
        }
    }
    catch(EOFException ex)
    {
        //no need an error message
    }
    catch(IOException ex)
    {
        ex.printStackTrace();
    }
    finally{
        //the connection is closed for one reason or another, so have the server dealing with it
        server.removeConnection(socket);
    }
}
}

package Steganography;

```

```

import java.io.File;

import java.net.URL;

import javax.swing.ImageIcon;


public class Utils
{

    public final static String gif = "gif";
    public final static String png = "png";
    public final static String dat = "dat";


    public static String getExtension(File f) { //בודק את סיומת הקובץ
        String ext = null; //פה ישמר סוג הקובץ
        String s = f.getName(); //שם הקובץ
        int i = s.lastIndexOf('.'); //כמות התווים שיש מהנקודה
        if (i > 0 && i < s.length() - 1) //אם קיים משהו אחרי הנקודה
        {
            ext = s.substring(i+1).toLowerCase(); //חסר את השם והנקודה והחזר מה שנישאר
        }
        return ext; //מה שנשאר זה בעצם מחרוזת של הסיומת של קובץ או תמונה
    }

    protected static ImageIcon createImageIcon(String path)
    {
        URL imgURL = Utils.class.getResource(path);
        if (imgURL != null) {
            return new ImageIcon(imgURL);
        } else {
            System.err.println("Couldn't find file: " + path);
            return null;
        }
    }
}

```

נספח ב' - ביבליוגרפיה:

- חומרים על RSA: קריפטולוגיה 1, סיכום קורס.
- Balloontip - Balloon tips for Java Swing applications Copyright 2007, 2008 Bernhard Pauler, Tim Molderez – שימוש במחלקה המעצבת בלון טיפ וקובץ Jar.
- [/http://stackoverflow.com](http://stackoverflow.com) לימוד על שימוש בבתים Java.
- bouncy <https://www.bouncycastle.org/java.html>
- castle java, ספק להצפנה מסוג RSA, נלקחו שני קבצי Jar.
- JavaMail – ספק המשמש לשליחת אימייל.