


ספר הפרויקט

מגיש: עומר דגרי

מנחה: ניר דוויד

DES Encryption & Decryption & 3DES2 Encryption & Decryption



Contents

מדריך למשתמש	3
Stage 1	3
Stage 2	3
תרשים זרימה של הפרויקט	5
רשימת הפרוצדורות	6
permutation	6
printstring	6
readstring	6
shift_keys	6
shift_keys_2	6
connect_arr	6
connect_arr_run4last	7
copy_arr	7
xor_arr	7
S_Box_1 S_Box_2 S_Box_3 S_Box_4 S_Box_5 S_Box_6 S_Box_7 S_Box_8	7
S_Box1to8	7
Final_Stage	7
hex_to_ascii	8
DES_Encryption	8
ascii_to_hex	8
reversed_permutation	8
DES_Decryption	8
clean_data	9
DES3_Encryption	9
DES3_Dcryption	9
רשימת מקורות חיצוניים	10

מדריך למשתמש

Stage 1

לאחר ההרצה המשתמש מתבקש להכניס את **mode** שהוא רוצה להיות בו.

enter your desired mode !!! DES Encryption - 1E, DES Decryption - 1D, DES3 Encryption - 3E, DES3 Decryption - 3D !!!

Stage 2

לאחר שהמשתמש הכניס את **mode** הרצוי הוא יתבקש בהתאם ל**mode** שבחר להכניס את הנתונים שצריך, לאחר שהוא יכניס את הנתונים הוא יקבל את המידע המוצפן או המפוענח תלוי ב**mode** שבחר.

```
enter your desired mode !!! DES Encryption - 1E, DES Decryption - 1D, DES3 Encry
ption - 3E, DES3 Decryption - 3D !!!
1D
-----

enter encrypted data !! IN HEX !!
21C60DA534248BCE
-----

enter the encrypted data - key:
abcdefgh
-----

Decrypted data:
12345678
-----
```

Mode 1D

```
enter your desired mode !!! DES Encryption - 1E, DES Decryption - 1D, DES3 Encry
ption - 3E, DES3 Decryption - 3D !!!
1E
-----

enter data (8 chars):
12345678
-----

enter your key (8 chars):
abcdefgh_
-----

Encrypted Data:
21C60DA534248BCE
-----
```

Mode 1E

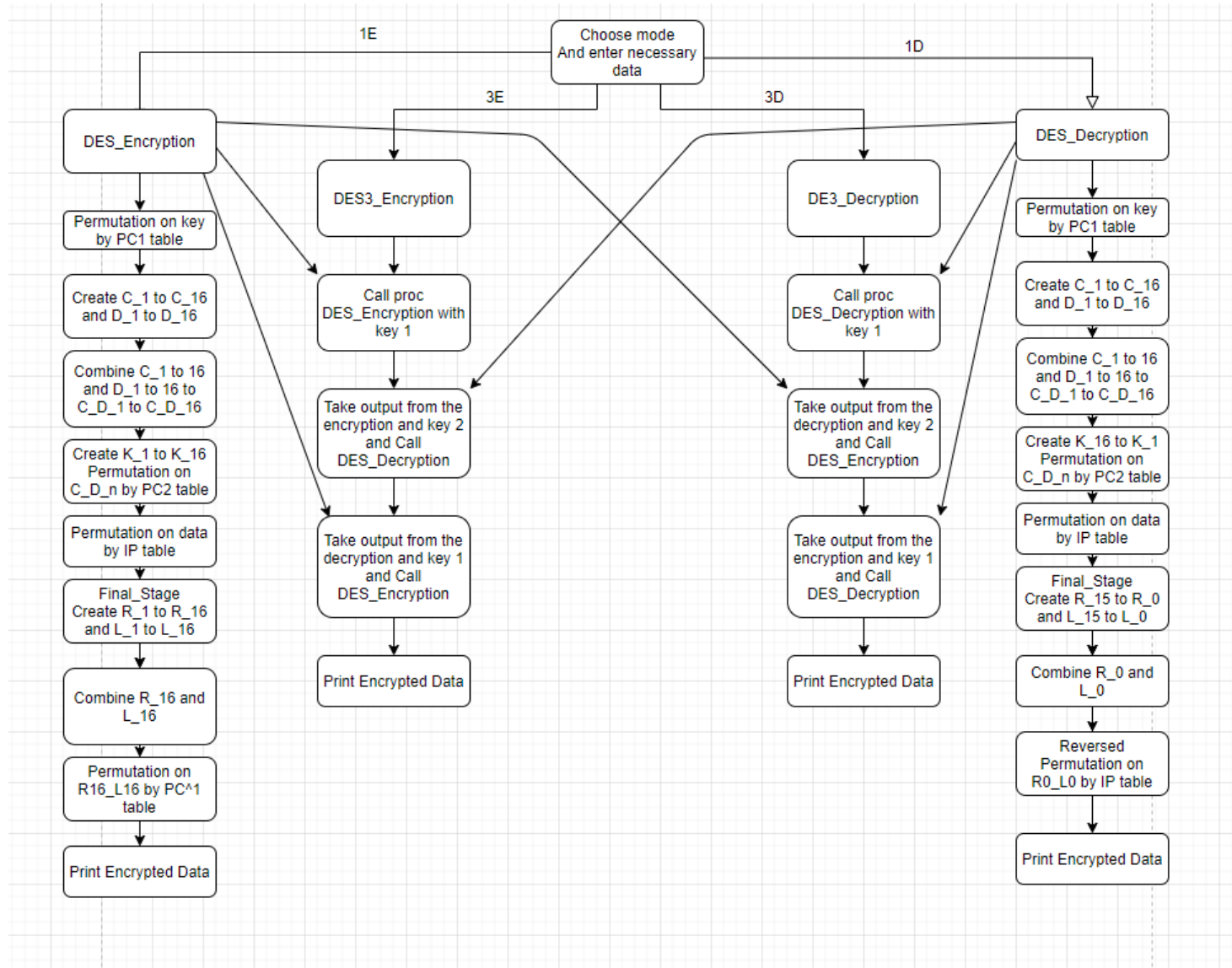
```
enter your desired mode !!! DES Encryption - 1E, DES Decryption - 1D, DES3 Encry
ption - 3E, DES3 Decryption - 3D !!!
3D
-----
enter encrypted data !! IN HEX !!
30A20DEBC1CF5438
-----
enter the encrypted data - key 1
abcdefgh
-----
enter the encrypted data - key 2
87654321
-----
Decrypted data:
12345678
-----
```

Mode 3D

```
enter your desired mode !!! DES Encryption - 1E, DES Decryption - 1D, DES3 Encry
ption - 3E, DES3 Decryption - 3D !!!
3E
-----
enter data (8 chars):
12345678
-----
enter your first key (8 chars):
abcdefgh
-----
enter your second key (8 chars):
87654321
-----
Encrypted Data:
30A20DEBC1CF5438
-----
```

Mode 3E

תרשים זרימה של הפרויקט



רשימת הפרוצדורות

permutation

offset של טבלה לפרמוטציה וoffset של מה שעושים עליו פרמוטציה וoffset של הפלט ועל כמה ביטים מבצעים את הפרמוטציה.

הפרוצדורה לוקחת את הערך הראשון מהטבלה שהוא מייצג מיקום של ביט מסוים ואז הולכת לביט המסוים בoffset שעליו עושים פרמוטציה ומעתיקה אותו למיקום של הביט הראשון בoffset החדש ואז לוקחת את הערך השני הולכת לביט ומכניסה אותה למיקום של הביט השני וכך הלאה.

printstring

מקבלת offset של משהו ומדפיסה מהoffset הזה עד שמגיע ל'\$'.

readstring

מקבלת offset של מקום לאחסן קלט ומכניסה את הקלט למקום זה (במערך צריך להיות 2 בתים ריקים בהתחלה, הראשון אומר כמה בתים ניתן לקלוט והשני אומר כמה נקלטו, את הראשון צריך לקבוע מראש).

shift_keys

מקבלת offset של 2 דברים, אחד שהיא עושה לו שיפט מעגלי ואחד שהיא תשמור אליו את התוצאה.

shift_keys_2

מקבלת offset של 2 דברים, אחד שהיא עושה לו שני שיפטים מעגליים ואחד שהיא תשמור אליו את התוצאה, פרוצדורה זו קוראת פעמיים ל[shift_keys](#).

connect_arr

מקבלת offset של שני מערכים ואת אורכם וoffset של מערך ריק בגודל פי 2 משני המערכים האחרים ומחברת את שני המערכים לתוך המערך היחיד הגדול פי 2.

connect_arr_run4last

מקבלת offset של שני מערכים ואת אורכם וoffset של מערך ריק בגודל פי 2 משני המערכים האחרים פחות בית (לדוגמה 2 מערכים של 4 אז זה יהיה מערך של 7) ומחברת את שני המערכים לתוך המערך היחיד, אך דורסת בבית האחרון בכל מערך את ה-4 ביטים האחרונים מאחר ולא הייתי צריך אותם (ולכן פי 2 פחות בית).

copy_arr

מקבלת offset של שני מערכים ואת אורכם ומעתיקה את המערך הראשון לשני.

xor_arr

מקבלת offset של שני מערכים ואת אורכם ועושה ביניהם xor.

S_Box_1 S_Box_2 S_Box_3 S_Box_4 S_Box_5 S_Box_6 S_Box_7 S_Box_8

כל הפרוצדורות האלו מקבלות offset של שני מערכים וכל אחת לוקחת את ה-6 ביטים שהיא צריכה לקחת במערך הראשון ולפי כל s-box הופכות את ה-6 ביטים ל-4 ביטים ומעתיקות את ה-4 ביטים האלו למערך השני.

S_Box1to8

מקבלת offset של שני מערכים וקוראת ל-S_Box_1 עד S_Box_8.

Final_Stage

מקבלת offset של 9 מערכים ומבצעת את expansion ואז את xor ואז את s_boxes ואז permutation ואז עוד xor ובסוף מקבלים את R_1 עד R_16 ואת L_1 עד L_16.

hex_to_ascii

מקבלת offset של מערך ואורכו ומדפיסה את הערך עצמו שיש בזיכרון ולא את הערך האסקי של הערך שבזכרון, לדומה יש בזכרון 31 היא תדפיס 31 ולא את הספרה 1.

DES_Encryption

לאחר שקוראים לפרוצדורה היא תדאג לקבל קלט לבד (אלא אם זה דרך des3, ואז הקלט כבר קיים).
הפרוצדורה מבצעת את כל תהליך ההצפנה של DES בעזרת כל הפרוצדורות שהיו עד עכשיו ומדפיסה את התוצאה.

ascii_to_hex

מקבלת offset של שני מערכים ואת האורך של המערך פלט, לוקחת מהמערך הראשון את הערכים שיש (אמורים להיות באסקי, בשלב של פענוח קולטים את הטקסט שצריך לפענח בהקסהדצימלי ואז זה הופך להיות אסקי בזיכרון אז צריך להמיר את זה להקסהדצימלי בזיכרון) וממירה אותם להקסהדצימלי ושומרת במערך השני.

reversed_permutation

מקבלת offset של שני מערכים וoffset של טבלת פרמוצטיה ואת כמות הביטים שיש, והופכת את התהליך שהפרוצדורה [permutation](#) עושה.

DES_Decryption

לאחר שקוראים לפרוצדורה היא תדאג לקבל קלט לבד (אלא אם זה דרך des3, ואז הקלט כבר קיים).
הפרוצדורה מבצעת את כל תהליך הפענוח של DES בעזרת כל הפרוצדורות שהיו עד עכשיו ומדפיסה את התוצאה.

clean_data

מנקה את כל המידע שצריך להשתמש בו שוב, רק ב3D mode, מאחר ורק בהם מצפינים ומפענחים כמה פעמים.

DES3_Encryption

לאחר שקוראים לפרוצדורה היא תדאג לקבל קלט לבד.

הפרוצדורה מבצעת את כל תהליך ההצפנה של 3DES2 בעזרת [DES Encryption](#) ו- [DES Decryption](#) | [clean_data](#)

DES3_Dcryption

לאחר שקוראים לפרוצדורה היא תדאג לקבל קלט לבד.

הפרוצדורה מבצעת את כל תהליך הפענוח של 3DES2 בעזרת [DES Encryption](#) ו- [DES Decryption](#) | [clean_data](#)

רשימת מקורות חיצוניים

Source 1: <https://he.wikipedia.org/wiki/3DES>

Source 2: <https://paginas.fe.up.pt/~ei10109/ca/des.html>

Source 3: <http://xor.pw#/>

Source 4: <https://he.wikipedia.org/wiki/DES>

Source 5: <https://pdfcoffee.com/des-example-encryption-decryption-pdf-free.html>

Source 6: <http://des.online-domain-tools.com/>