מטלה 2 : שימוש בדיפי-הלמן, צופנים, ופונקציות ערבול

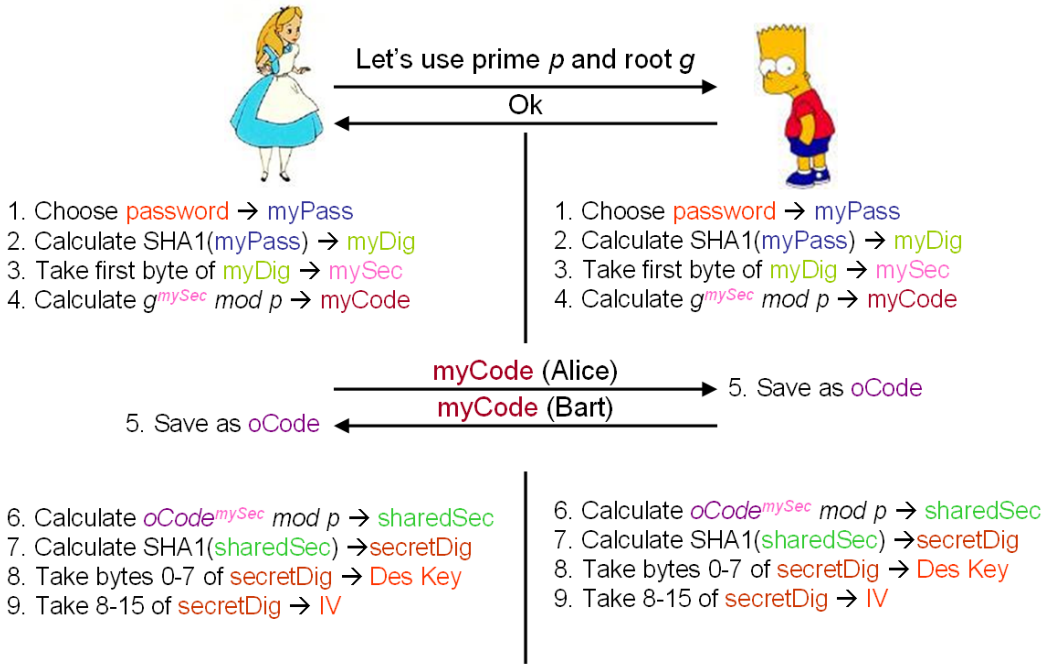
קורס 1-02-328 : אבטחת תקשורת ומסחר אלקטרוני
מועד אחרון להגשה: **10 אפריל 2011 ב 08:30**

במטלה זו נשתמש בשלושה מהכלים שהוסברו בהרצאות הקודמות – צופן סימטרי DES, פונקציות ערבול קריפטוגרפיות, והקמת מפתחות בפרוטוקול דיפי-הלמן.
יש לעשות שימוש ביישומי האלגוריתמים ב.NET על מנת לבנות אפליקציה, להצפין ולפענח קבצים.

חלק 1: העברת קבצים באופן מאובטח

בעיה שכיחה בשליחת קבצים דרך דוא"ל היא אבטחה.
נניח כי ברצונכם לשלוח קובץ עם מידע רגיש למישהו דרך אימייל, אבל אין לכם סוד או סיסמא משותפת שתאפשר לכם לשלוח אותו באופן מאובטח.
הכלי שתפתחו במטלה הזאת יענה על הבעיה הזאת ויאפשר לכם להצפין קבצים להעברה באימייל מבלי שיהיה לכם סיסמא או קוד משותף מראש.

השלבים הבסיסיים להקמת מפתח משותף ל DES ו IV מופיעים באיור הבא:



איור 1: שלבים בסיסיים בהקמת מפתח ו IV משותפים ל DES על ידי החלפת המפתחות של דיפי-הלמן

כעת נסביר את השלבים של הקמת המפתח וה IV המשותפים כמו שהם מתוארים באיור.
הפרוצדורה הינה סימטרית, כלומר, שני הצדדים מבצעים את אותן פעולות ובסופו של דבר הם מפתחים את אותו מפתח סודי ו IV.

בהתחלה, אליס וברט מגיעים להסכמה על מספר ראשוני p ושורש פרימיטיבי g. (המספרים לא חייבים להיות סודיים).

(הערה: g אמור להיות שורש פרימיטיבי ל p, אבל האלגוריתם יעבוד אפילו אם הוא לא באמת שורש פרימיטיבי.)

- השלבים לבחירת p ו g לא חייבים להיות מאובטחים, ולכן ניתן לשלוח את המספרים באימייל רגיל (לא מאובטח).

ברגע שהם הגיעו להסכמה על p ו g, הם יכולים להתחיל את הצעדים לחשב את המפתח הסודי וה IV ל DES:

1. אליס בוחרת סיסמא (טקסט) סודי שנקרא "myPass"

2. אליס מחשבת את הדייג'סט (פלט digest-) של פונקצית ערבול SHA1 של myPass (ערבול myPass על ידי SHA1) ושומרת את הדייג'סט בשם "myDig" .

3. אליס לוקחת את הבייט הראשון של myDig ושומרת אותו בשם "mySec".

   - **<u>הערה</u>**: ההחלטה לקחת רק הבייט הראשון מקטינה את מספר המפתחות האפשריים ל 256, ומחלישה את הפרוטוקול בצורה משמעותית. בתרגיל זה הסיבה לכך היא על מנת להקל עליכם בדיבוג הקוד והאפליקציה שלכם.

4. אליס מחשבת $g^{mySec} \bmod p$ ושומרת את התוצאה בשם "myCode". אליס שולחת את myCode לברט.

5. ברט גם עשה את הפעולות הנ"ל (1-4), ולכן אליס מקבלת קוד מברט. אליס שומרת את הקוד שהיא קבלה מברט בשם "oCode".

6. אליס מחשבת $oCode^{mySec} \bmod p$ ושומרת את התוצאה בשם "sharedSec". הוא הקוד הסודי שמשותף בין אליס וברט.

7. אליס מחשבת את הדייג'סט של sharedSec (ערבול sharedSec על ידי SHA1) ושומרת אותו בשם "secretDig".

8. אליס לוקחת את בייטים 0-7 מ secretDig ושומרת אותם בתור המפתח ל DES.

9. אליס לוקחת את הבייטים 8-15 מ secretDig ושומרת אותם בתור ה IV.

ברט, כמו אליס, גם כן יבצע את התהליך הנ"ל ויקבל אותו מפתח ל DES ואותו IV. מעכשיו, הם יכולים להשתמש במפתח DES וב IV להצפין ולפענח קבצים.

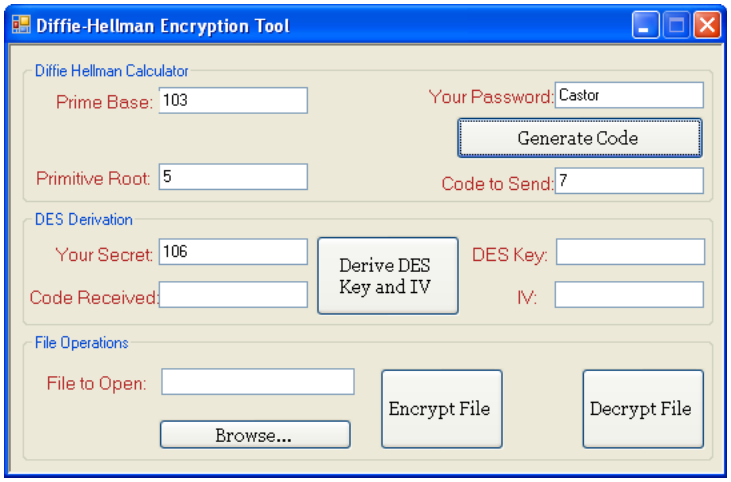
חלק 2: מה עליכם לעשות

תפקידכם לפתח אפליקציה שתאפשר לאליס וברט לבצע את התהליך הנ"ל.
נראה דוגמה לריצה של התהליך עבור p=103 ו g=19. אליס בוחרת הסיסמא "Castor" וברט בוחר את הסיסמא "Pollux".

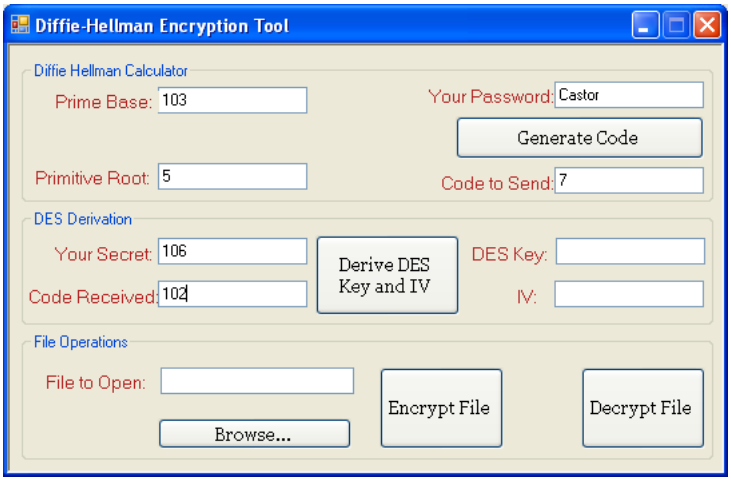
1. אליס מכניסה את המספר הראשוני 103, השורש הפרימיטיבי 5, והסיסמא שלה "Castor". היא לוחצת על הלחצן "Generate Code" ומייצרת שתי תוצאות:

   a. הסיסמא שלה mySec.
   הסיסמא נוצרת על ידי חישוב הקידוד יוניקוד (Unicode) של הסיסמא, חישוב פונקצית ערבול (הש), ושליפת הבייט הראשון. הערך הסופי (כלומר, הבייט הראשון) מופיע בקופסת-טקסט (textbox) שכותרתה "Your Secret".
   ניתן לראות באיור כי הערך של אליס הינו 106.

   b. הקוד שאליס אמורה לשלוח לברט. הוא מחושב על ידי $5^{106} \bmod 103 = 7$. הערך מופיע בקופסת-טקסט עם שכותרתה "Code to Send".



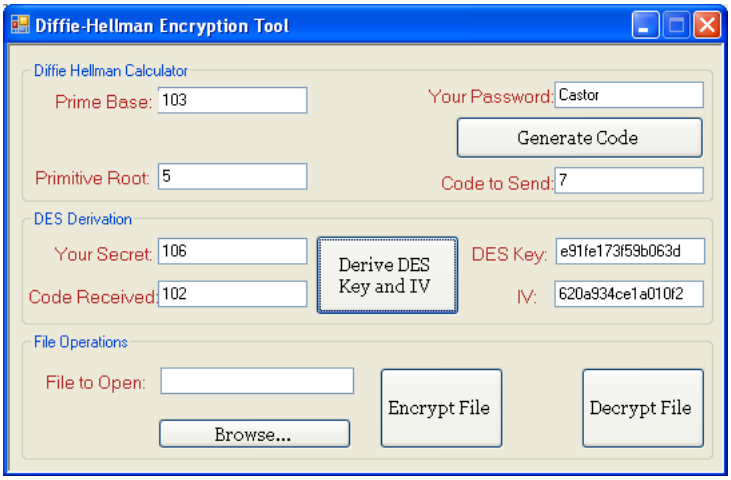
2. אליס מקבלת את הערך 102 מברט (זהו הערך שברט קיבל כשהוא הכניס הסיסמא "Pollux" לכלי שלו).שימו לב כי במטלה זו אין עליכם לממש את קבלת הערך, ויש להכניסו ידנית לתוכנה.
היא מכניסה את הערך (102) לקופסת-טקסט שכותרתה "Code Received".



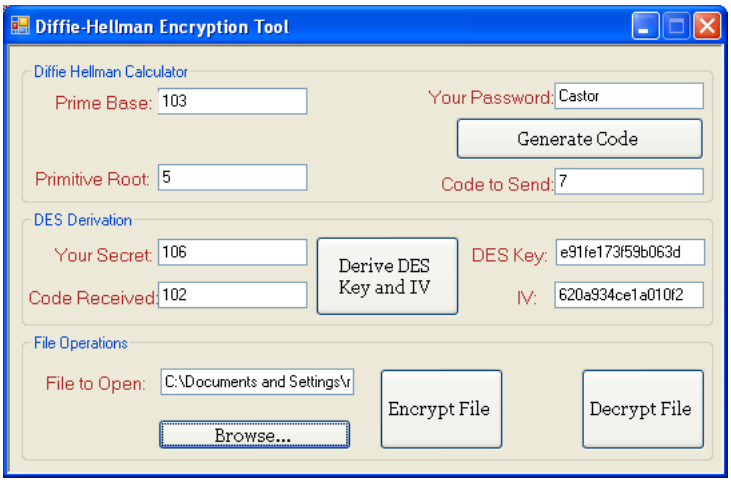
3. אליס לוחצת על הלחצן שכותרתו "Derive DES Key and IV" לחשב את הסוד המשותף

$$102^{106} \bmod 103 = 1$$

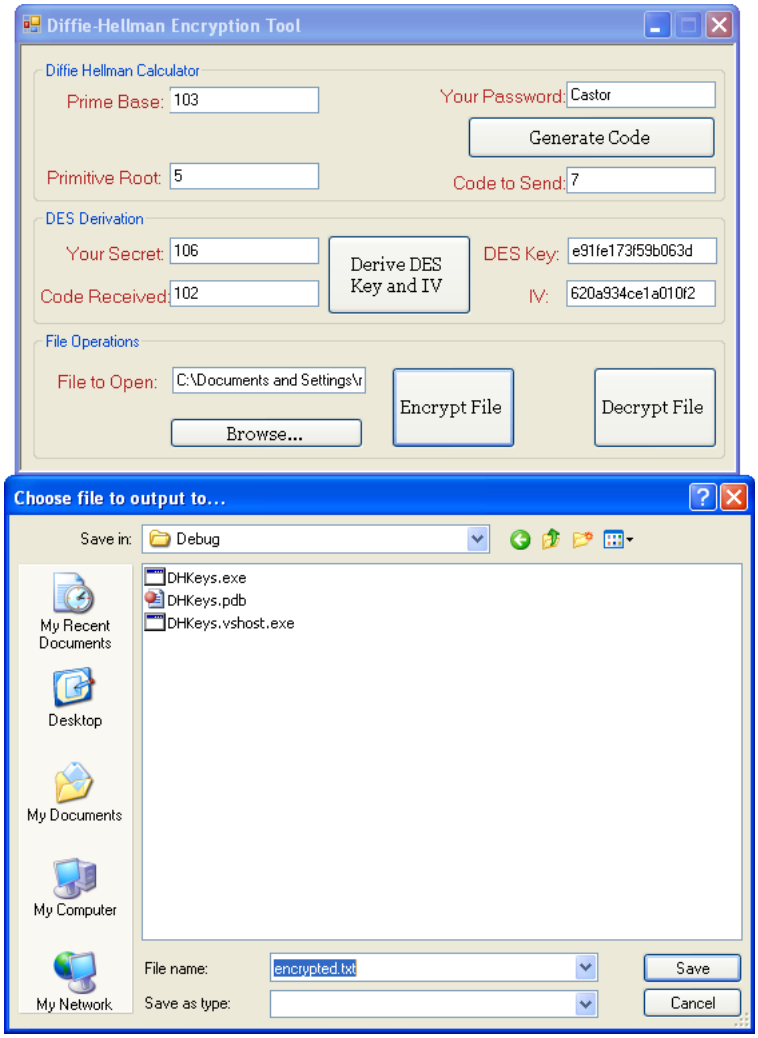
הכלי מחשב את הדייג'סט של 1 (בתור מחרוזת יוניקוד).
שמונת הבייטים הראשונים מהפלט מופיעים (בקידוד הקסדצימלי) בקופסת-טקסט שכותרתה "DES Key" (הערך הינו e91fe173f59b063d).
שמונת הבייטים הבאים מופיעים בקופסת-טקסט שכותרתה "IV" (הערך הינו 620a934ce1a010f2).



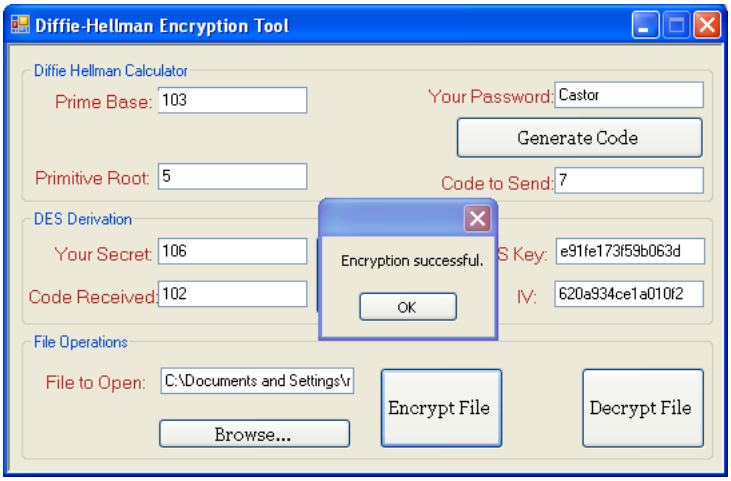
4. אליס בוחרת את הקובץ להצפין בלחיצה על הלחצן שכותרתו "Browse". היא בוחרת קובץ (שכבר קיים) בשם "testing-file.txt" כדי להצפין אותו.



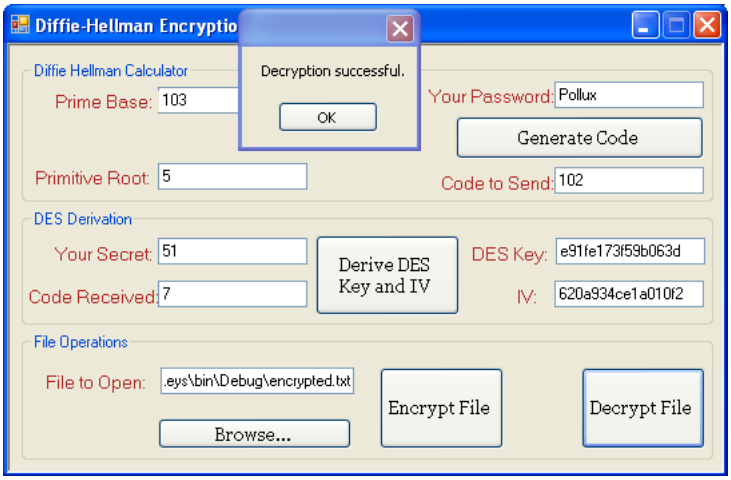
5. אליס לוחצת על "Encrypt File" ובוחרת שם בשביל הקובץ המוצפן שיווצר.



6. לאחר לחיצה על "Save", הכלי מצפין את הקובץ ומציג חלון "הצלחה".



עכשיו אליס יכולה לשלוח את הקובץ לברט באופן מאובטח. כשברט יקבל את הקובץ הוא יכול ללחוץ על "Decrypt File" כדי לפענח את הקובץ כמו שאליס עשתה.



חלק 3: בדיקת האפליקציה

הכלי שלכם אמור להיות מסוגל להצפין ולפענח קבצי טקסט וגם קבצים בינאריים. כדי לנהל קבצים בינאריים, יש להשתמש במחלקות BinaryReader ו BinaryWriter לנהל את הזרימות ב C#.
ניתן לקרוא על המחלקות בתיעוד באינטרנט.

כדי לעזור לכם בבדיקת הקוד, ישנם שני קבצים מוצפנים באתר של הקורס. אחד הקבצים הינו קובץ טקסט והשני הינו קובץ PDF (בינארי). שניהם מוצפנים במפתחות של אליס וברט כמו שהוסבר בדוגמה הקודמת (p=103, g=5, אליס בוחרת סיסמא "Castor" וברט בוחר סיסמא "Pollux"). ניתן להשתמש בהם לבדוק את הקוד שלכם לפענוח קבצים.

בדקו את האפליקציה שלכם במספרים ראשוניים ובשורשים פרימיטיביים אחרים.
אתר אינטרנט עם אפליקציית Java שימושית שמגלה שורש פרימיטיבי לכל מספר ראשוני שמזינים לה:

http://www.math.mtu.edu/mathlab/COURSES/holt/dnt/quadratic4.html.

חלק 4: הערה על חזקות מדולריות

תמצאו שהטיפוסים הרגילים בC# לא מספיק רחבים כדי לחשב חזקות מדולריות גדולות בדרך הפשוטה ביותר (כלומר, g^a ואחר כך חישוב מדולו p). ולכן בקלות תקבלו בעיות של הצפה (overflow).

קיים אלגוריתם יותר יעיל שעובד במערכים של סיביות (ניתן לקרוא עליו יותר באינטרנט), אבל מפני שאנחנו משתמשים רק בחזקות (יחסית) קטנות (16 סיביות) ניתן לחשב החזקות באלגוריתם רקורסיבי בזמן סביר.
באתר הקורס תמצאו פונקציה בC# לחשב חזקות בצורה רקורסיבית. ניתן להעתיק אותה לתוך האפליקציה שלכם ולהשתמש בה כמו שהיא.

חלק 5: מה להגיש עד 10 אפריל 2011 ב08:30

כל קבוצה חייבת להגיש הדברים הבאים (הגשה אחת לכל קבוצה). הגשות אלקטרוניות יתקבלו בטלמ. בדיסק-און-קי אישי, או בשליחה במייל לכתובת של הקורס (ise328@gmail) בקובץ אחד בפורמט ZIP או RAR.

1. דוח של המטלה בשם README.txt שכולל:
   - שמות כל הסטודנטים בקבוצה. ומשפט אחד או שנים לתאר מה כל סטודנט בקבוצה תרם לעבודה וכמה שעות עבדתם על המטלה.
   - הקבוצה תקבל ציון אחד על ההגשה. המידע על התרומות האישיות יעזור לי למדוד את הקושי והיעילות של המטלה. הוא לא ישפיע כלל על הציון שלכם.
   - הנחיות או הוראות מיוחדות להפעלת האפליקציה
   - **במידה והמידע הנ"ל חסר, ייחול קנס של 5 נקודות**

2. הגשה אלקטרונית של הקוד. הקוד חייב להיות בתיקייה אחת דחוסה בפורמט ZIP או RAR. התיקייה גם חייבת להכיל גרסה מקומפלת (compiled) של האפליקציה שניתן להריץ כמו שהיא. האפליקציה חייבת להבנות בהתאם לממשק המתואר בקובץ הזה וכתוב בC# בלבד.

   - **הערה:** עקב בדיקות של אבטחה בGmail, לא ניתן לשלוח קבצי EXE במייל, אפילו בתוך קבצי ZIP. כדי להגיש קובץ EXE, ניתן להגישו אותו בטלמ, דחסו את הקובץ בפורמט RAR, או להגיש את הEXE אליי אישית בדיסק-און-קי.

שיתוף פעולה בין קבוצות, העתקה מהאינטרנט, מקורות מקוונות אחרות, אתרי קוד, או משאבים אחרים אסורים בהחלט. אשתמש בכלים אוטומטיים לזיהוי העתקות או שיתוף קוד.

חלק 5.1: חישוב הציון

בדיקת הכלים שלכם בהצפנות, מפתחות, וקבצים תיעשה לפי ראות עיני.
הציונים לעבודה יינתנו לפי הקריטריונים הבאים:

- חישובי דיפי-הלמן: 30%
- יצירת מפתחות של DES ו30%: IV
- הצפנה ופענוח קבצים 40% (קבצי טקסט וקבצים בינאריים)