TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

אבטחת מערכות מידע (0571-4180)

עבודה מס' 1 - הצפנה ופענוח טקסט

הוראות כלליות:

- הגשה בזוגות או בשלשות בלבד.
- מועד הגשת המטלה מעודכן באתר ה-Moodle.
- העבודה תוגש באופן אלקטרוני לאתר ה-Moodle בקובץ ZIP.
- חשוב שרק אחד בלבד מבני הזוג לפרויקט יגיש את חלקי העבודה באתר.
- שאלות על העבודה יש לשאול רק בפורום המתאים ב-Moodle על מנת שכל הסטודנטים יוכלו לראות את התשובות.

הוראות כלליות:

- בעבודה זו יהיה עליכם לממש אלגוריתם ההצפנה העובד במצב הפעלה CBC יחד עם שתי התקפות על טקסטים המוצפנים באלגוריתם זה. **חשוב להבין את אלגוריתם ההצפנה, מצב ההפעלה והקלט שלו עוד לפני תחילת העבודה.**
- **הגשת המטלה:**
  - יש להגיש את המטלה בקובץ ZIP יחיד המכיל **רק** את שלושת קבצי ההרצה הבאים (ללא תת-תיקיה בתוך ה-Zip):
    - cbc.py
    - CipherTextAttack.py
    - PlainTextAttack.py
  - שם קובץ ה-ZIP יכיל את ת"ז של המגישים בפורמט ID1_ID2_ID3 כאשר מספרי הת"ז מסודרים בסדר עולה.
- נא בדקו את הקוד שלכם באמצעות שורות הפקודה המצורפות בסעיפים השונים (מופיע עם רקע אפור). את שורות הפקודה יש להריץ דרך Command Prompt (CMD). לצורך כך, מומלץ להשתמש בחבילת argparse.
- הניקוד יעשה על פי רמת הדמיון של המפתח המוחזר למפתח המקורי. מפתחות זהים לגמרי יזכו בציון המרבי.
- שימו לב שהעבודה מורכבת משלושה חלקים, כאשר גודל הבלוק משתנה מחלק לחלק.
- כל הטקסטים בהם העבודה עוסקת הינם בשפה האנגלית בלבד (מספרים ותווים מיוחדים).
- אם הטקסט לא מתחלק בגודל הבלוק, ירופד בתווי null character (**\0**) (לא התו שמתאים לסימבול '0'). שימו לב ש **\0** הוא תו יחיד (כמו **\n** המסמל שורה חדשה).
- לעבודה מצורפים קבצי דוגמא:
  - plainMsg_example – הודעה קצרה להצפנה.

- key_example – מפתח לדוגמא (יש לפתוח באמצעות notepad++ או wordpad).
- IV_example, IV_longExample – וקטור אתחול לדוגמא.
- cipherMsg_example – הודעה מוצפנת באמצעות המפתח ו-IV_example.
- התיקיה KeyGenerator מכילה תוכנה ליצירת מפתחות (*KeyProducer (JDK1.6).jar*)
  - התוכנה מבוססת Java לכן חייב להתקין JRE על המחשב שלכם. http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
  - על מנת להפעילה השתמשו בקובץ המצורף *GenerateKey.bat*.

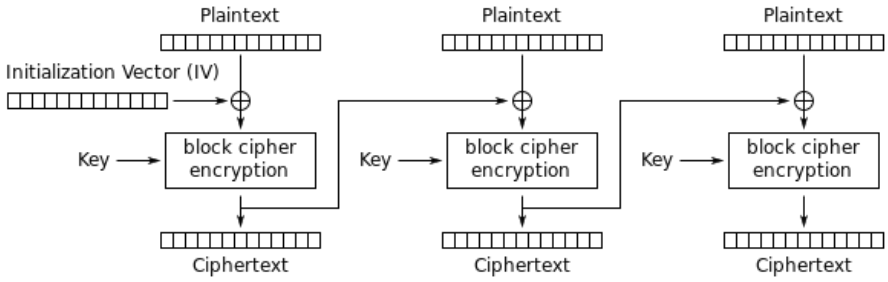
- **דגשים חשובים ביותר**:
  - שפת התכנות היא Python בלבד. השתמשו בגירסה 3.5 ומעלה (לא ב-2.7).
  - **יש לכתוב קוד ברור וקריא:**
    - הקפדה על חלוקה לפונקציות.
    - כתיבת הסברים לאורך כל הקוד ותיעוד של הפונקציות.

קוד מסורבל ולא קריא עלול להוביל לקושי בבדיקתו ואף להורדת ציון.

- אין להשתמש בשום חבילת Python חיצונית שאינה חלק מה-Python הבסיסי (למעט חבילות להעברת ארגומנטים ב-CMD). קבוצה שתעשה שימוש בחבילות חיצוניות מסתכנת בכך שעבודתה לא תיבדק ושהציון יהיה 0.
- כאשר יש צורך לכתוב קובץ לדיסק, כתבו רק את שם הקובץ בלבד (לדוגמא: *plainText_encrypted.txt*), **ולא** Path מלא (לדוגמא: C:*plainText_encrypted.txt*). הדבר חשוב ביותר מאחר ואינכם יכולים לדעת אילו כוננים קיימים במחשב שבו תתבצע הבדיקה.
- קריאה של קבצי עזר תתבצע ע"י ציון של הקובץ בלבד, ללא Path מלא.
- קבצי הפלט שלכם צריכים להיכתב לאותה התיקיה שבה נמצא קובץ ההרצה (לדוגמא cbc.py). חשוב ביותר!
- הטמיעו בתוך קובץ ה-Python שלכם (Hard Coded) את מילון המילים שישמש אתכם למתקפות.
- אין להשתמש בקבצים חיצוניים כלשהם לפתרון המטלה (פרט לקבצים שהקוד שלכם מקבל כקלט).
- שימו לב כמה פרמטרים צריך לקבל כל קובץ קוד שהונחתם לכתוב. שגיאה בכמות הפרמטרים או באופיים עלול לגרום להורדת ציון משמעותית.
- זמן הריצה חייב להיות סביר - החזרת **פתרון תוך פחות מדקה!**

רקע

לחברת Miracles & Priest יש שני סניפים והיא מעוניינת להעביר מידע מאובטח בין סניפיה המרוחקים: אחד בסיליקון ואלי (Palo Alto) והשני בסיליקון ואדי (BGU). החברה בחרה להצפין את המידע שלה באמצעות אלגוריתם הצפנה המבוסס על "צופן החלפה" אשר עובד במצב הפעלה Cipher-block chaining (CBC) שלמדתם בכיתה:



Cipher Block Chaining (CBC) mode encryption

חלק א: מימוש אלגוריתם CBC

אלגוריתם הצפנה: ההצפנה מחליפה רק 8 אותיות ראשונות מתוך 26 האותיות בשפה האנגלית: (a…h). טבלת ההחלפה, המהווה את מפתח ההצפנה (ראה טבלה מס' 1) תהיה בת 8 כניסות, כלומר מפתח ההצפנה הוא בגודל 8 תווים, כאשר תחום הערכים שווה לטווח הערכים, [a-h]-> [a-h] (ראה דוגמה בטבלה מס' 2). על הפונקציה להיות הפיכה, כלומר היא חייבת להיות חד-חד ערכית ועל.

Index	0	1	2	3	4	5	6	7
Source	a	b	c	d	e	f	g	h
Destination	?	?	?	?	?	?	?	?

טבלה מס' 1 – מבנה מפתח הצפנה. המפתח הוא השורה השלישית, כלומר שורת ה-Destination.

Index	0	1	2	3	4	5	6	7
Source	a	b	c	d	e	f	g	h
Destination	b	a	c	d	e	g	h	f

טבלה מס' 2 – מפתח הצפנה כלשהו. נתן לראות כי כל מופע של האות a הופכת בכתב הסתרים להיות b.

כאשר: אורך IV וגודל הבלוק הינם 10 תווים (אותיות או מספרים). IV ידוע לכולם.

לדוגמה: ה-plain text "ABCDQRSTAB" יעבור xor עם וקטור אתחול בין 10 **התווים**, למשל "0000000000" ויתקבל "**qrstabcdqr**" $(65('A') \oplus 48('0') = 113('q'))$ ואז בעזרת המפתח (בטבלה מס' 2) יהפוך ל- cipher text "qrst**ba**cdqr".

1.1 **(7 נקודות)** ממש/י את סכמת ההצפנה CBC המקבלת קובץ טקסט, וקטור אתחול ומפתח הצפנה וכותבת את הטקסט המוצפן לקובץ בשם "*plainText_encrypted.txt*" כאשר *plainText* שם קובץ הטקסט הקריא.

*python **cbc.py** Encryption [plainText.txt] [key.txt] [iV.txt]*

1.2. **(7 נקודות)** ממש/י את סכמת הפענוח של CBC המקבלת קובץ מוצפן, וקטור אתחול ומפתח הצפנה וכותבת את הטקסט המקורי לקובץ בשם "*plainText_decrypted.txt*" כאשר *cipherText* שם קובץ הטקסט המוצפן.

*python **cbc.py** Decryption [cipherText.txt] [key.txt] [iV.txt]*

חלק ב: Cipher-Text Only Attack

סוכנות הביון PloughedField & ForeverFather מעוניינת לפענח את המידע שהחברה מעבירה בין סניפיה לצורך כך היא משתמשת בהתקפת Cipher-Text Only Attack לצורך גילוי מפתח ההצפנה:

2. **(21 נקודות)** ממש/י התקפת Cipher-Text Only Attack המקבלת כקלט טקסט מוצפן ווקטור אתחול ומחזירה את מפתח ההצפנה כפלט. הפלט ירשם לתוך קובץ טקסט בשם "*cipherText*_key.txt" כאשר *cipherText* הוא שם קובץ הטקסט המוצפן שמתקבל כקלט. בקובץ זה יש לרשום רק את המפתח לפי הסדר! בפורמט הדוגמה המצורפת (פתחו עם wordpad או ++notepad). **יש לשים לב שמפתח ההצפנה ומפתח הפענוח הפוכים!**

*python **CipherTextAttack.py** Decryption [cipherText.txt] [iV.txt]*

חלק ג: Known Plain-Text Attack

חברת Miracles & Priest הבחינה כי בבדיקות אבטחה שגרתיות ניתן לפענח את הטקסט המוצפן בקלות ע"י מתקפת Cipher-Text Only Attack, לכן החליטה החברה להגדיל את מפתח ההצפנה (ראה טבלה מס' 3) כדי לקבל הצפנה חזקה יותר שתבטל את היכולת להשתמש במתקפה זו, כלומר מפתח ההצפנה מחליף יותר אותיות (ראה דוגמה בטבלה מס' 4). על פונקצית ההצפנה להיות הפיכה, כלומר היא חייבת להיות חד-חד ערכית ועל.

Index	0	1	…	25	26	27	…	51
Source	a	b	…	z	A	B	…	Z
Destination	?	?	…	?	?	?	…	?

טבלה מס' 3 - מבנה מפתח הצפנה. המפתח הוא השורה השלישית, כלומר שורת ה-Destination

Index	0	1	…	25	26	27	…	51
Source	a	b	…	z	A	B	…	Z
Destination	g	J	…	f	s	F	…	H

טבלה מס' 4 – מפתח הצפנה כלשהו. נתן לראות כי כל מופע של האות Z הופכת בטקסט המוצפן להיות H וכל תו 'a' הופך ל 'g'.

<u>כאשר:</u> אורך IV וגודל הבלוק הינם 8,128 תווים (אותיות או מספרים). IV ידוע לכולם.

3.1 **(5 נקודות)** ממש את ההצפנה לפי המתואר לעיל.

3.2 **(30 נקודות)** סוכנות הביון N&A הבחינה כי אין באפשרותה לפענח את ההצפנה החדשה באמצעות המתקפה בחלק ב' אך יש באפשרותה להשיג חלקי מידע קטנים מפוענחים ומוצפנים. על כן, החליטה סוכנות הביון לשנות אסטרטגיה ולהשתמש בהתקפת Known-Plain-Text על סכמת ההצפנה CBC.

ממש התקפת Known Plain-Text Attack ע"י כתיבת תכנית המקבלת ארבעה קבצי טקסט המכילים: וקטור אתחול, טקסט מוצפן, הודעת טקסט קריא בגודל עד 8,128 תווים (P_{known}) והודעה מוצפנת המתאימה להודעת הטקסט הקריא $C_{known}=E_k(P_{known})$. מפתח ההצפנה של שני הטקסטים המוצפנים בקלט זהה. יש לשים לב כי הודעת הטקסט הקריא המשמש את התוקף מכילה רק חלק מהאלפבית, כלומר, אחת או יותר מהאותיות באלפבית האנגלי לא נכללות בה. התוצאה המתבקשת בסעיף זה היא מפתח ההצפנה. כמו בסעיף 2, יש לרשום את המפתח המתקבל לתוך קובץ : "*cipherText*_key.txt". דוגמה להודעת טקסט קריא והטקסט המוצפן המתאים מצורפת.

```
python PlainTextAttack.py [plainMsg.txt] [cipherMsg.txt] [cipherText.txt] [iV.txt]
```

בהצלחה!