אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

The Department of Information Systems Engineering

Security of Computers and Communication Networks

Assignment #1

Submission guidelines

- **Please answer all questions**.
- Your answers should be full, short as possible, and address the question asked.
- Answers should be submitted in a word document called **answers.pdf**
- If you need more information **Google it!** Still have questions? Use the course forum on Moodle (do not use any other ways to ask questions (e.g., e-mails).
- Submission is allowed only in pairs. Please find yourself a partner from your department.
- Make sure your zip file is not corrupt – download it and extract it from Moodle
- Please submit the assignment to Moodle.
- Postponements will not be given (except of special cases such as Miluim, and etc.)
- Each day of delay will result in reduction of 5 points to the assignment's grade.

חלק א – חלק מעשי

שאלה מס' 1 (25 נק):

בשאלות הבאות תלמדו לחקור קובץ PCAP שמכיל הקלטה של משתמש ולהסיק מה המשתמש עשה בגלישתו. לטובת השאלה הנ"ל יש להתקין WIRESHARK על המחשב. WIRESHARK הוא כלי ניתוח פופולארי לתעבורת רשת.

אנא הורד את הקובץ MYSTERY-21. טענו את הקובץ PCAP המצורף לעבודת בית ב - WIRESHARK. הקובץ הנ"ל מכיל הקלטה של תעבורת רשת.

יש מס' דרכים לענות על השאלות הבאות. אנו ממליצים לכם לפעול בדרך הבאה:

- מיצאו את הפילטר הרצוי שעונה על השאלה והכניסו אותו לשורה הרלוונטית (דוגמאות לכתיבת ביטויים לפילטרים ב WIRESHARK ניתן לראות בקישור הבא ובחיפוש באינטרנט).
- יצאו את התוצאה שהתקבלה מהפילטר לקובץ CSV ע"י:
  File->Export Packet Dissections-> As CSV
- כתבו קוד שמנתח את הקובץ CSV או נתחו בעזרת EXCEL.
- בכל סעיף אתם מתבקשים לכתוב את הפילטר שהשתמשתם בו ואת התשובה.

- שאלות פילטרים (כל שאלה 2 נקודות):
  א. כמה פקטות TCP נשלחו ממכשיר שה IP שלו הוא 216.58.210.234 בשניה ה 31.
  ב. מה החלון זמנים הקצר ביותר בין שתי פקטות בפרוטוקול TLSv1.2 שנשלחו למכשיר שה IP שלו הוא 216.58.210.234
  ג. מה גודל הפקטה המקסימלי שנשלח ממכשיר שה IP שלו הוא 216.58.210.234
  ד. מה גודל המינימלי המקסימלי שנשלח ממכשיר שה IP שלו הוא 216.58.210.234
  ה. באילו אתרים גלש המשתמש (כתוב לפחות 2)?

- שאלות גרפים (כל שאלה 3 נקודות)
  בכל השאלות אנו מבקשים ליצר את ציר X כציר הזמן:
  א. יצרו גרף של כמות פרוטוקולים שונים של פקטות שנשלחו כפונקציה של הזמן (בשניות) ממכשיר ש IP שלו הוא 216.58.204.10 (לדוגמא אם בשניה השלישית נשלחו רק פקטות TCP אז הכמות בשניה הזאת היא 1).
  ב. יצרו גרף של נפח פקטות UDP שנשלחו כפונקציה של הזמן (בשניות) למכשיר ש IP שלו הוא 10.100.102.6
  ג. יצרו גרף של כמות פקטות DNS שנשלחו כפונקציה של הזמן (בשניות) למכשיר ש IP שלו הוא 10.100.102.1
  ד. יצרו גרף של כמות היעדים השונים אליהם נשלחה הודעה ממכשיר ש IP שלו הוא 216.58.204.10 כפונקציה של הזמן (בשניות).
  ה. יצרו גרף של נפח תעבורה כפונקציה של הזמן (בשניות) של הודעות שנשלחו ברשת לכל אחד מהפרוטוקולים הבאים:
     a. TLSv1.3
     b. DNS
     c. TCP
     d. UDP

שאלה מס' 2 (25 נק):

בשאלה זו אתם צריכים לשבור ווריאציה מסוימת של צופן AES.

נגדיר גרסה פשוטה של AES ונקרא לה AES_1^* . נגדיר את הגרסה הנ"ל בצורה הבאה:

M – הודעה לא מוצפנת Plain-text

C – הודעה. מוצפנת Cipher-text

K – מפתח הצפנה/פענוח

Formal Definition of AES_1*

AES_1* is a single round implementation of AES that is defined as follows:

- $AES_1^* (M)_K = AddRoundKey(SwapIndexes (M),K) = C$
- $AES_1^{*-1} (C)_K = SwapIndexes^{-1}(AddRoundKey(C,K)) = M$

הפונקציה SwapIndexes (M) היא פונקציה אשר מחליפה כל תא בהודעה בתא הנמצא באינדקסים ההפוכים. כלומר, התא הנמצא במיקום (I, J) יקבל את הערך הנמצא בתא (J, I) ולהפך. לדוגמא:
התא באינדקס (0,0) נשאר זהה.
התא באינדקס (1,0) מקבל את הערך שנמצא בתא (0,1).
התא באינדקס (0,1) מקבל את הערך שנמצא בתא (1,0).
וכו'....

$SwapIndexes^{-1}(M)$ אמורה לבצע את הפעולה ההפוכה, אך מכיוון שהפעולה המקורית היא החלפת אינדקסים, הפעולה ההפוכה שלה תהיה גם החלפת אינדקסים. כלומר:

$SwapIndexes (M) = SwapIndexes^{-1}(M)$

כעת נגדיר אלגוריתם הצפנה נוסף המבוסס על הגרסה הפשוטה שהגדרנו זה עתה. נגדיר אלגוריתם הנקרא AES_2^*, אלגוריתם זה נעזר ב2 מפתחות שונים, K1,K2 ובעזרתם הוא מבצע 2 איטרציות של האלגוריתם AES_1^*.
ולכן הגדרה של AES_2^* היא:

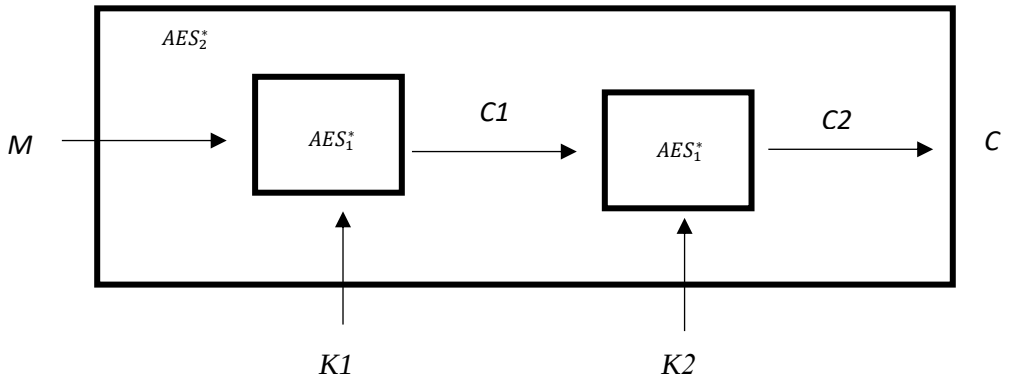
Formal Definition of AES_2*

AES^*_2 is the application of AES^*_1 twice with two different keys: K_1, K_2.

- $AES^*_2\{M\}_{K1, K2} = AES_1^*\{AES_1^*\{M\}_{K1}\}_{K2} = C$
- $AES^{*-1}_2\{C\}_{K1, K2} = AES_1^{*-1}\{AES_1^{*-1}\{M\}_{K2}\}_{K1} = M$

בהינתן הודעה M והודעה מוצפנת C כך ש: $C = AES_2^*\{M\}_{k1,k2}$, הנכם צריכים לממש שיטה יעילה למציאת 2 מפתחות K1,K2 המקיימת: $C = AES_2^*\{M\}_{k1,k2}$.

האלגוריתם הצפנה AES_2^* ממומש כך:



שימו לב כי במימוש שלכם לפריצה של AES_2^* עליכם להתייחס ל AES_2^* כקופסא שחורה המקבלת הודעה M ו2 מפתחות כקלט ומוציאה כפלט הודעה מוצפנת C לפי ההגדרה נ"ל. **אינכם יכולים להשתמש בהודעות הביניים C1! כמו כן מפתחות K1 K2 חייבים להיות שונים אחד מהשני.**

א. רשום פתרון תיאורטי לשיטה שאתה מציע **(5 נק')**.
ב. ממש את הפתרון שהצעת ב- JAVA לפי הדגשים הבאים **(20 נק')**:

דגשים למימוש:

- הודעה M יכולה להיות ארוכה יותר מ 128 ביט, המימוש שלכם צריך לקחת בחשבון שאורך הודעה יכול להיות יותר ארוך מ 128 ביט, לחלק את ההודעה לבלוקים של 128 ביט ולהפעיל את האלגוריתם על כל בלוק, לשם הפשטות ניתן להניח כי אורך ההודעה היא מכפלה של 128 ביט.
- עליכם לממש ממשק (interface) הצפנה/פענוח כדלקמן:

  - –e: instruction to encrypt the input file
  - –d: instruction to decrypt the input file
  - –k <path>: path to the keys, the key should be 256 bits (128*2) for AES_2^*. and should be divided into 2 separate keys.
  - –i <input file path>: a path to a file we want to encrypt/decrypt
  - –o <output file path>: a path to the output file
  - Usage examples:
    - **Java –jar aes.jar -e/-d –k <path-to-key-file > -i <path-to-input-file> -o <path-to-output-file>**
    - **Java –jar aes.jar –e –k key.txt –i message.txt –o cypher.txt**

- עליכם לממש ממשק (interface) לשבירה של ההצפנה כדלקמן:

  - –b : instruction to break the encryption algorithm
  - –m <path>: denotes the path to the plain-text message
  - –c <path>: denotes the path to the cipher-text message
  - –o <path>: a path to the output file with the key(s) found.
  - Usage: **Java –jar aes.jar -b –m <path-to-message> –c <path-to-cipher> -o < output-path>**

- פורמט הפלטים והקלטים:
  - הנכם מתבקשים לכתוב ולקרוא מקבצים **בבתים Bytes ולא כטקסט.**
  - **שימו לב לסדר בתים (Endianness) ,ניתן לוודא את סדר הבתים בשקופיות של ההרצאה.**
  - השתמשו בקבצי בדיקה שסופקו לכם ביחד עם התרגיל על מנת לבדוק את התוכנית שלכם.

- שימו לב כי זמן ריצה של התוכנית צריך להיות בזמן סביר **הלא עולה מעל דקה אחת**.
- אין להשתמש ב brute force.
- עליכם להגיש את כל קבצי המקור וקובץ jar מקומפל של התוכנית שלכם.
- הבדיקה מתבצעת בתוכנה אוטומטית, אנא בדקו היטב כי התוכנית שלכם עונה על כל הדרישות הנמצאות בקובץ הזה.
- שימו לב כי תוכנה אוטומטית תצליב בין כל קבצי המקור לזיהוי קוד דומה, אנא הימנעו מהעתקות.
- ההגשה היא במודל, יש להגיש קובץ zip בלבד בפורמט הבא: ass1_id1_id2.zip בתוך הקובץ יש לשים את כל קבצי המקור וקובץ jar. קובץ jar חייב להיות בשם aes.jar
- קבצי בדיקה ניתן להוריד במודל

חלק ב – חלק תיאורטי

שאלה מס' 3 (15 נק'):

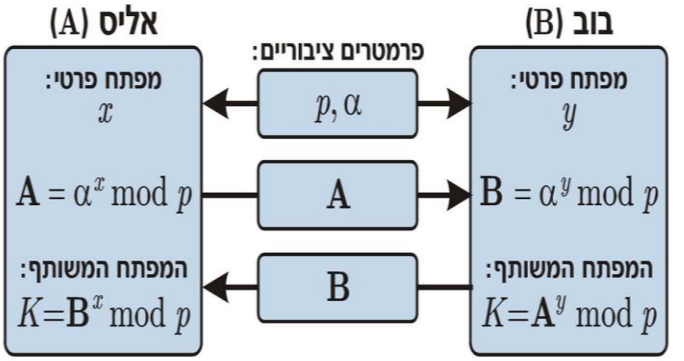
א. הסבר את המודלי ההתקפה הבאים על מערכת קריפטוגרפית:
   a. Ciphertext-only attack
   b. Known-plaintext attack
   c. Chosen-plaintext attack
   d. Chosen-ciphertext attack

ב. הסבר כיצד תוקף שרוצה לתקוף אלגוריתם הצפנה RSA ומחזיק בשתי הודעות ובקריפטוגרמות שלהן (p1,c1) ו (p2,c2) יכול להגדיל את סט ההודעות והקריפטוגרמות שלו ב O(1).
בתשובתך, הסבר באיזה תנאים התוקף לא יכול להגדיל את סט ההודעות והקריפטוגרמות שלו.
בנוסף, הסבר כיצד ניתן למנוע מתוקף להגדיל את סט ההודעות שלו במקרה הנ"ל.

שאלה מס' 4 (20 נק'):

באונ' בן גוריון חילקו לכל העובדי מנהלה מפתח פרטי ומפתח פומבי.

כמו כן, נקבע כי בכל התקשרות (סשן) חדשה בין שני עובדים יופעל תחילה פרוטוקול דיפי-הלמן לסיכום על מפתח סימטרי לסשן, כאשר המפתחות הפרטיים x.y יהיו המפתחות הפרטיים של העובדים שחולקו להם. לאחר שמסכמים על מפתח סימטרי לסשן, העובדים מבצעים את ההתקשרות ע"י הצפנה/פענוח של ההתקשרות בעזרת אלגוריתם AES.



א. הסבירו מה הבעייתיות בפרוטוקול שהוצע?
ב. באונ' בן גוריון ניסו לפתור את הבעיה ע"י שינוי הפרוטוקול לסיכום על מפתח, אך החברה שכתבה עבורה את הקוד פשטה רגל וקבצי ה SOURCE CODE לא נשלחו לאוניברסיטה.
מבלי לשנות את הפרוטוקול הנ"ל לסיכום על מפתח סימטרי בעזרת המפתח הפרטי של עובדים, כיצד ניתן להתגבר על הבעייתיות שנוצרה בצורה שקלה לשליטה ותחזוקה?

שאלה מס' 5 (15 נק'):

מפתחי קוד לראסברי פאי (RP) רצו לפתח ספריה אשר ממשמשת את ההצפנה 5-DES בעזרת 5 מפתחות K1-K5.

5-DES-Encrypt (K1,K2,K3,K4,K5,M)

1. C1 = DES(K2,DES^{-1} (K1,M))
2. Store C1 in C:\DES\tmp
3. Return DES(K5,DES(K4,DES (K3,C1)))

הפעולת פענוח היא הפעולת מראה של הפונ' לעיל.

א. הצע התקפה יעילה ככל האפשר על המנגנון הנ"ל (מבחינת זמן ריצה). נתח סיבוכיות זמן ריצה וזיכרון.

ב. שמעון פיתח קוד בעזרת הספריה הנתונה והחליט לעטוף את הפונקציה בצורה הבאה:

5-DES-Encrypt (K1,K2,K3,M)

Return 5-DES-Encrypt (K1,K2,K1,K2,K5,M)

הצע התקפה עם זמן ריצה יעיל ככל האפשר על המנגנון של שמעון. נתח סיבוכיות זמן ריצה וזיכרון.

ג. יוסי פיתח קוד בעזרת הספריה הנתונה והחליט לעטוף את הפונקציה בצורה הבאה:

5-DES-Encrypt (K1,K2,K3,K4,M)

Return 5-DES-Encrypt (K1,K1,K2,K3,K4,M)

הצע התקפה יעילה ככל האפשר על המנגנון של יוסי.
נתח סיבוכיות זמן ריצה וזיכרון.