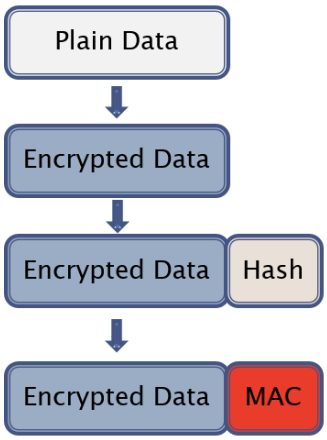
רשתות מחשבים מתקדם - תרגיל יצירת סוקט מוצפן

ברק גונן

בתרגיל זה נממש את ארבעת המנגנונים הנדרשים ליצירת סוקט מוצפן:

1. הצפנה סימטרית מבוססת מפתח שידוע רק לשני בצדדים
2. מנגנון קביעת מפתח הצפנה משותף
3. פונקציית גיבוב (Hash)
4. יצירת חתימה MAC באמצעות מפתח פרטי

התרגיל ידריך כיצד לבנות את המנגנונים. להלן איור של השלבים:



הצפנה סימטרית מבוססת מפתח הצפנה

מסיבות פדגוגיות, נעשה שימוש בהצפנת בלוק פשוטה, המדמה את העקרונות של AES אך הרבה יותר פשוטה למימוש. מפתח ההצפנה שלנו יהיה בגודל 16 ביטים בלבד. לא מורכב לפיצוח באמצעות בדיקת כל האפשרויות, אך ממחיש את העקרון ולא קשה למימוש.

בצד המשדר: כל 4 בתים שעומדים להשלח בסוקט יוצפנו בתור בלוק. אם כמות הבתים שצריכה להשלח לא מתחלקת ב-4, הוסיפו ריפוד באפסים לגודל של 4 בתים.

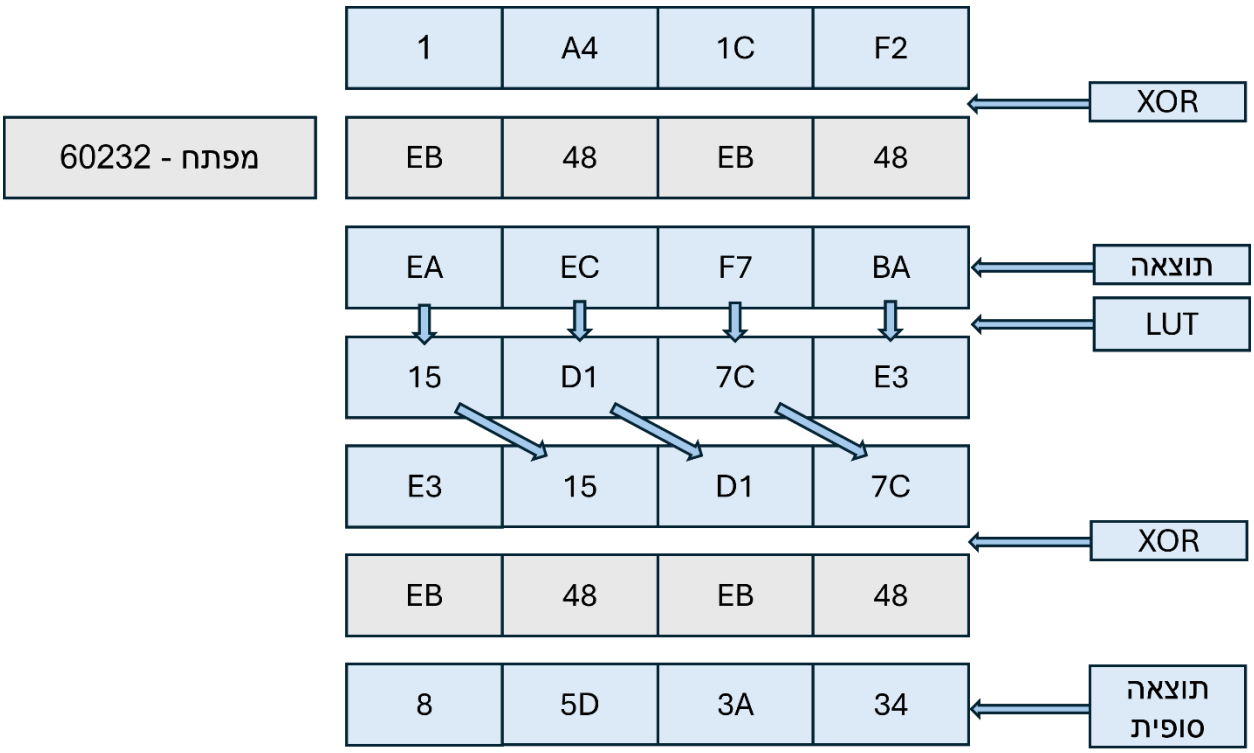
בשלב הראשון יבוצע XOR של כל שני בתים עם מפתח ההצפנה.

בשלב השני באמצעות Look Up Table יוחלף כל ערך בגודל בית, בערך אחר שגודלו בית. לדוגמה בית שערכו 0 יהפוך להיות 14, בית שערכו 1 יהפוך להיות 251 וכן הלאה.

בשלב השלישי יוחלף סדר הבתים בבלוק של הרביעיה, בצורה מעגלית. כך הבית ה-0 יהפוך להיות ה-1, ה-1 יהפוך להיות ה-2, ה-3 יחזור להיות ה-0.

יבוצע XOR נוסף של כל שני בתים עם מפתח ההצפנה וזה הבלוק שישודר

בצד הקולט: כל ארבעה בתים שמתקבלים מהסוקט יעברו פענוח לאחר הקליטה. הפענוח הוא הפעולה ההפוכה להצפנה: ביצוע XOR עם המפתח, החלפת סדר הבתים בצורה מעגלית (לכיוון השני), ביצוע Look Up Table (עם מילון שבו הערכים והמפתחות הפוכים מאשר המילון ששימש להצפנה, לדוגמה 251 הופך להיות 1), ביצוע XOR נוסף.



<u>מנגנון קביעת מפתח הצפנה משותף</u>

קביעת המפתח, הסוד המשותף, תתבצע באמצעות אלגוריתם Diffie Hellman. הסוד המשותף צריך להיות בגודל 16 ביטים, בהתאם לשיטת ההצפנה שנבחרה לעיל.

שלב א: בחרו שני מספרים P, G הקטנים מ-65535 (הערך המקסימלי של 16 ביט). P צריך להיות ראשוני.

שלב ב: כל צד יבחר מפתח פרטי

שלב ג: כל צד יחשב את המפתח הציבורי של עצמו וישלח לצד השני, באמצעות הסוקט שפתוח ביניהם. בשלב זה המידע שעובר בסוקט אינו מוצפן עדיין וכולל רק את המפתחות הציבוריים של שני הצדדים.

שלב ד: כל צד יחשב את הסוד המשותף, מספר בגודל 16 ביט, כפי שמתחייב מכך שהחישוב לוקח רק את השארית של החלוקה ב- P

<u>פונקציית גיבוב</u>

כיתבו פונקציית גיבוב כיד הדמיון הטובה עליכן. הפונקציה תיקח את המסר המוצפן כולו, תבצע עליו פעולות מתמטיות שונות ותהפוך אותו למספר בן 16 ביט.

<u>חתימה (MAC) באמצעות מפתח פרטי</u>

נשתמש באלגוריתם RSA לטובת החתימה, Message Authentication Code.

א. בחרו מספרים ראשוניים P, Q שהמכפלה שלהם P*Q גדולה מ-2 בחזקת 16. הסיבה לכך היא, שאחרת עלול לקרות מצב שבו התוצאה של פונקציית הגיבוב (שהיא בתחום של 16 ביט) תהיה ערך שלא ניתן להגיע אליו בעזרת החתימה, המבצעת מודולו במספר נמוך יותר.
ב. בחרו מפתחות ציבוריים לשני הצדדים
ג. חשבו את המפתחות הפרטיים המתאימים למפתחות הציבוריים שבחרתם

החתימה תתבצע באופן הבא:

שלב א: הצד השולח יחשב את התוצאה של פונקציית ה-Hash על המסר המוצפן

שלב ב: הצד השולח יחשב את הערך של החתימה: ה-Hash בחזקת המפתח הפרטי, מודולו P*Q

ניתן להשתמש בפונקציה pow, לדוגמה:

```
signature = pow(hash, private_key, P*Q)
```

שלב ג: הצד השולח ישלח את המסר המוצפן ובסיומו את החתימה

שלב ד: הצד המקבל ישתמש במפתח הציבורי של הצד השולח כדי להוציא את ה-Hash המקורי, לדוגמה:

```
Received_hash = pow(signature, public_key, P*Q)
```

שלב ה: הצד המקבל יחשב באופן עצמאי את ה-Hash מיתר המידע שנשלח אליו

שלב ו: הצד המקבל יבדוק אם ה-Hash המחושב זהה ל-Received_hash ואם לא- יזרוק את ההודעה שהתקבלה, כיוון שאינה מקורית.

<u>שלבי התקשורת על גבי הסוקט</u>

בשלב הקמת הסוקט המוצפן יתבצעו התיאומים הדרושים בין שני הצדדים:

- יועברו המפתחות הציבוריים של Diffie Helman
- יועברו המפתחות הציבוריים של RSA

<u>בשלב התקשורת המוצפנת ההודעות יעברו מוצפנות ועם MAC.</u>

כיתבו צאט בין שרת ולקוח (אין צורך לעבוד עם מספר לקוחות), ההודעות יועברו בצורה מוצפנת ועם חתימה.