

נמשיך עם עוד דוגמה להצפנת RSA שבמהלכה נלמד גם איך מעלים במהירות בחזקה כשמדובר בחשבון מודולארי.

**מה זה חשבון מודולארי?**

**חשבון מודולרי** (הידוע גם כ**חשבון קונגרואנציות**) הוא שיטה **מתמטית**, בה מחליפים מספרים בשארית החלוקה במספר קבוע. לדוגמה, בחשבון מודולו 7 מתקיים השוויון (מודולו 7)  $5 + 6 = 4$ , מכיוון ש:  $5 + 6 = 11$ , ו**שארית החלוקה** של 11 ב-7 היא 4.

חישוב שעות כדוגמה  
חישוב שעות נעשה מודולו 24 או 12. נניח שכעת השעה היא 20:00 (או 8 בערב).  
מה תהיה השעה בעוד 9 שעות?  
 $20 + 9 \equiv 5 \pmod{24}$  (גם 8 פלוס 9 מודולו 12 נותן 5)

**פעולת הקונגרואציה:**  
 $a \equiv b \pmod{n}$

אומרים שהמספר השלם  $a$  הוא קונגרואטיבי למספר השלם  $b$  אם שניהם נותנים אותה שארית בחלוקה ב- $n$  (וכך זה נרשם)

לדוגמה:

$$10 \equiv 17 \pmod{7}$$

$$10 \pmod{7} = 3$$

$$17 \pmod{7} = 3$$

כלומר 10 הוא קונגרואנטי ל 17 מודולו 7 שזה גם קונגרואנטי ל 3 מודולו 7.  
אגב מספרים שהם ראשוניים ביחס אחד לשני (נקראים גם קו-פריים), הם מספרים שאין להם מחלק משותף מלבד 1.

וכעת לדוגמה: נבחר שני ראשוניים שירכיבו את  $N$ .

$$p=23, q=41$$

$$p \cdot q = 943$$

$$(p-1) \cdot (q-1) = 22 \cdot 40 = 880$$

קו-פריים ל 880, למשל 7. לכן נפרסם לעולם את המפתח להצפנה:

(7,943)

נניח ששולחים הודעה שהיא למשל המספר 35

$$35^7 \pmod{943} = 545$$

נמצא כעת  $d$  (לפיענוח) שמקיים את המשוואה  $d \cdot 7 = 1 \pmod{880}$   
יש אינספור כאלה, למשל: 503 או 1383. ננסה את שניהם:

$$545^{503} \pmod{943} = 35$$

אל חשש מהמספרים הגדולים, יש לנו תוכנית 'פלא' שמחשבת במהירות מדהימה.  
cryptosystem.py

עכשיו ננסה את מפתח הפיענוח 1383

$$545^{1383} \pmod{943} = 35$$

זה  $d$  נקרא בשפה מתמטית: **הופכי מכפילי מודולו**. הופכי מכפילי רגיל זה מספר  
שם נכפיל בו את המקורי נקבל 1. למשל ההופכי המכפילי הרגיל של 40 זה  $1/40$   
באנגלית: multiplicative inverse

חישבנו את החזקות הגבוהות בחשבון מודולרי די בקלות הודות לכלל הבא:

$$a \cdot b \pmod{n} = a \pmod{n} \cdot b \pmod{n}$$

איך זה עוזר? זה מקטין את המספרים המוכפלים. למשל:

$$1750 \cdot 890 \pmod{55} = 1750 \pmod{55} \cdot 890 \pmod{55} = 45 \cdot 10 \pmod{55} = 10$$

שנית, אם נייצג את החזקה כחזקות של 2, אפשר להקטין את מספר המכפלות  
(העלאות בריבוע) כפי שאנו יודעים אפשר לייצג כל מספר כבינארי (סכום חזקות  
של 2)

נראה זאת על ידי דוגמה: יש לחשב את התוצאה של

$$7^{66} \pmod{101}$$

החזקה 66, בבינארי זה: 10000010 (או  $2^6 + 2^1$ ) ולכן אפשר לכתוב:

$$7^{66} = 7^{(2^6)} \cdot 7^2 = 7^{64} \cdot 7^2$$

כמה פעולות של העלאה בריבוע נדרשות?

$$7^2 = 7 \cdot 7 = 49 \pmod{101} = 49$$

$$7^4 = 7^2 \cdot 7^2 = 49 \cdot 49 \pmod{101} = 2401 \pmod{101} = 78$$

$$7^8 = 7^4 \cdot 7^4 = 78 \cdot 78 \pmod{101} = 6084 \pmod{101} = 24$$

$$7^{16} = 7^8 \cdot 7^8 = 24 \cdot 24 \pmod{101} = 576 \pmod{101} = 71$$

$$7^{32} = 7^{16} \cdot 7^{16} = 71 \cdot 71 \pmod{101} = 5041 \pmod{101} = 92$$

$$7^{64} = 7^{32} \cdot 7^{32} = 92 \cdot 92 = 8464 \pmod{101} = 81$$

ולכן:

$$7^{66} \pmod{101} = 7^{64} \cdot 7^2 \pmod{101} = 81 \cdot 49 \pmod{101} = 3969 \pmod{101} = 30$$

**אם נחזור לדוגמה הנ"ל:**

$$545^{503} \pmod{943}$$

$$503 = 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1 + 2^0$$

503 בבינארי: 111110111

$$545^{(2^8)} \cdot 545^{(2^7)} \cdot 545^{(2^6)} \cdot 545^{(2^5)} \cdot 545^{(2^4)} \cdot 545^{(2^2)} \cdot 545^{(2^1)} \cdot 545$$

$$545^{503} \pmod{943} = 35$$

קיבלנו חזרה את ההודעה המפוענחת.

אל חשש מהמספרים הגדולים, יש לנו תוכנית 'פלא' שמחשבת במהירות מדהימה.

עכשיו ננסה את מפתח הפיענוח 1383

$$545^{1383} \pmod{943} = 35$$

הפלא ופלא, שניהם עובדים והתוכנית אפילו לא מנידה עפעף מהחישובים האסטרונומיים. cryptosystem.py

כמובן ש 943 אפשר לפרק לגורמים בשניות ולמצוא את 23 ו-41. במציאות המספר שמהווה את המכפלה יהיה בעל 200 עד 300 ספרות, כך שגם כל הזמן שעבר מתחילת היקום (13.8 מיליארד שנים) לא יספיק אפילו למחשב העל המהיר ביותר למצוא.

### **תהליך של אותנטיקציה**

איך נדע שאנחנו קונים באמזון ולא אצל מתחזה?

כל הציבור יודע את המפתח הציבורי של אמזון (לרוב נשלח למחשב שלכם כאשר אתם קונים שם בפעם הראשונה).

אמזון שולחת לנו שתי מחרוזות:  $x, y$

ההודעה בטקסט חופשי היא  $x$  ו-  $y$  זו ההודעה המוצפנת. אנחנו מפעילים את המפתח הציבורי של אמזון על  $y$ . אם קיבלנו את  $x$ , אנחנו יודעים שבצד השני נמצאת אמזון (כי היחידה שיכלה להצפין את  $x$  בעזרת המפתח  $d$  שיש רק לה).

איך לייצר את  $x$  ומי שמייצר אותו זה לדיון מעבר לטווח הדיונים שלנו כאן. מי שמעוניין בקריאה נוספת, מוזמן לקרוא את המסמך: Crypto01.pdf בתיקיה של קובץ זה.

משימה

לכתוב תוכנית שמקבלת כקלט ראשון: שני זוגות מספרים. זוג של מפתח ציבורי וזוג של מפתח פרטי.

קלט שני: הודעה כטקסט באנגלית והאות e או ההודעה המוצפנת כרשימת מספרים, והאות d

כלומר הקלט השני זה אות אחת (e או d) ומחרוזת

התוכנית תצפין את ההודעה אם ביקשנו e ותדפיס רשימה שמכילה מספרים שהתקבלו עבור ההודעה המוצפנת. אם ביקשנו d, התוכנית תפענח את ההודעה והפלט יהיה ההודעה לפני ההצפנה (אם התהליך יעבוד).

לדוגמה: קלט ראשון

7,943 , 503,943

קלט שני:

abcd , e

פלט:

[682, 426, 65, 59]

ועבור הקלט:

d , [682, 426, 65, 59]

התוכנית צריכה להדפיס חזרה :

abcd