מימוש מפתח הצפנה RSA

6 בנובמבר 2020

תזכורת מתמטית כיצד פועל מפתח ההצפנה: (סימון: $U_K = \{a \in \mathbb{Z}_K : (a, K) = 1\}$, φ - פונקציית אוילר: $\varphi(K) = |U_K|$)

- מגרילים שני ראשוניים p, q גדולים. מגדירים $N = pq$, מחשבים $\varphi(N) = (p-1)(q-1)$ ומגרילים $e \in U_{\varphi(N)}$ (מספר הפיך מוד $\varphi(N)$) . בעזרת האלגוריתם האוקלידי מחשבים את ההופכי של e כלומר $d \in U_{\varphi(N)}$ כך ש־$ed \equiv 1 \pmod{\varphi(N)}$.
- מפרסמים את המפתח הציבורי (N, e) ושומרים את d בסוד.
- הצפנה של הודעה x: $y = E(x) = x^e \pmod N$.
- פענוח של הודעה מוצפנת y: $x = D(y) = y^d \pmod N$.
- נכונות:
  - מכיוון ש־$ed \equiv 1 \pmod{\varphi(N)}$ אפשר לכתוב $ed - 1 = k \cdot \varphi(N)$.
  - לכל $x \in U_N$ מתקיים $x^{\varphi(N)} = 1 \pmod N$ כי $\varphi(N) = |U_N|$.
  - מכאן שלכל $x \in U_N$ מתקיים $x^{ed-1} = \left(x^k\right)^{\varphi(N)} = 1$,
  - כלומר $x^{ed} = x \pmod N$.
  - על כן: $D(E(x)) = x^{ed} \pmod N = x \pmod N$.

מטרת הפרויקט:

לממש בקוד את העקרונות המתמטיים לעיל ולפתור בעזרת המימוש לפחות 3 מתוך 4 החידות למטה (הגשת המימוש ופיתרון החידות במודל).
שימו לב שחלק מהחידות ניתנות לפיתרון גם לפני שמימשתם את כל הפונקציות, אז אתם יכולים לממש/לפתור באיזה סדר שאתם רוצים.

תהליך העבודה:

במודל נמצאים 3 קבצים:

- number_theory_functions.py כאן נמצאות הגדרות הפונקציות המיועדות למימוש אלגוריתם אוקלידס מוכלל, העלאה בחזקה מודולו וכו'
- rsa_functions.py כאן נמצאת הגדרת ה class של מערכת rsa המשתמשת בפונקציות מהקובץ הקודם

- test_rsa.py כאן נמצאים טסטים אותם אתם יכולים להריץ כדי לוודא שהקוד שלכם עובד כמו שצריך

למה כדאי לשים לב?

- בקובץ number_theory_functions.py יש 3 פונקציות האחראיות ליצירת ראשוניים (תוך כדי שימוש באלגוריתם מילר רבין) שכבר ממומשות. אתם יכולים להשתמש בהן או בכל פונקצית ספריה אחרת לשם יצירת הראשוניים.
- בבואכם לממש את פונקצית modular_exponent (העלאה בחזקה מודולו) שימו לב שעבור מספרים גדולים העלאה ישירה בחזקה ואז לקיחת מודולו לא תסתיים בזמן אנושי סביר. לכן כדאי להשתמש בטריק הבא: אם ברצוננו לחשב $a^d \pmod n$ נוכל להביט על הייצוג הבינארי $d = \sum_{i=0}^{m} b_i 2^i$ ואז לחשב

$$a^d = a^{b_0 2^0 + \ldots + b_m 2^m} = a^{b_0 2^0} \cdot \ldots \cdot a^{b_m 2^m} \equiv_n a^{b_0 2^0} \pmod n \cdot \ldots \cdot a^{b_m 2^m} \pmod n$$

לדוגמה עבור $d = 45$ הייצוג הבינארי הוא 101101 כלומר $b_0 = b_2 = b_3 = b_5 = 1$, $b_1 = b_4 = 0$ ואז במקום לחשב a^{45} ישירות נחשב

$$a^1 \pmod n \cdot a^4 \pmod n \cdot a^8 \pmod n \cdot a^{32} \pmod n = a^{45}$$

כאשר כמובן כדאי בנוסף לבצע את המודולו n לאחר כל פעולה, דהיינו $a^{32} \equiv_n (a \pmod n)^{32}$.

הערות נוספות:

- מסתבכים עם פייתון ורוצים לממש בשפה אחרת? בסדר גמור! אומנם פייתון נוחה לחישובים עם מספרים גדולים, אך אם אתם מעדיפים להשתמש בשפת תכנות אחרת (סי/מטלב וכו) אתם מוזמנים.
- אל תשכחו לפתור את החידות ולהגיש את הפתרונות והקוד שלכם במודל. אפשר להגיש את הקבצים כזיפ או לינק לגיטהאב/דרופבוקס/דרייב וכדומה שבו הקבצים נמצאים.
- אם נשאר לכם זמן וכוח מוזמנים ליצור ממשק למערכת rsa שמיממשתם ונשמח להציג את התוצר המוגמר במודל (זוהי רשות למי שרוצה. לא חלק מדרישות הפרוייקט)

חידות (לקבלת achievment unlock ומגן יש להגיש במודל לפחות 3 מתוך 4 החידות ואת הקוד שלכם)

1. לוקי רוצה שישלמו לו בדיוק מיליון דולר. לאיירון מן יש רק שטרות שכל אחד מהם בשווי 5279 דולר. לוקי מציע לתת עודף עם מטבעות שכל אחד מהם בשווי 797. האם יש סיכוי לעסקה כזו? כיצד?
2. מהי ספרת המאות של $23539673^{3434462}$?
3. נתון המפתח הציבורי $N = 12215009, e = 3499$, ונתונה ההודעה המוצפנת 42. פצחו את הקוד (רמז: עליכם לנחש קודם מהו המפתח הפרטי, מותר להשתמש בכדור בדולח/וולפראם אלפא וכדומה לשם כך)
4. נתונים הראשוניים $p = 7919, q = 6841$. בחרו הודעה ומפתח ציבורי e והצפינו את ההודעה. (הגישו למודל את ההודעה המקורית שלכם, המפתח הציבורי וההודעה המוצפנת).