מה הקשר בין מספרים ראשוניים וקריפטוגרפיה?

מאת ד"ר גדי אלכסנדרוביץ'

הקדמה

החודש נתגלה המספר הראשוני הגדול ביותר שנתגלה אי פעם (זה קורה אחת לכל כמה שנים), וגילוי שכזה תמיד מעורר שאלות מתבקשות של "בשביל מה זה טוב?". אם ניכנס לטוקבקים במאמר על התגלית ב-Ynet נגלה שבציבור הרחב, התחום שאליו מספרים ראשוניים מתקשרים ישירות הוא קריפטוגרפיה: "למספר יש חשיבות גדולה בהצפנה" אומר אחד. זה לא נכון ואסביר זאת בהמשך. אחר אומר "אין אלגוריתם או תוכנה שמסוגלת לחשב מספרים ראשוניים. לכן זאת נחשבת ההצפנה הטובה ביותר". גם זה לא נכון ואסביר זאת בהמשך. שלישי אומר "כל מערכת ההצפנה במחשבים עובדים על מספרים ראשוניים" שזה קצת יותר נכון אבל עדיין ממש לא נכון, ואסביר זאת בהמשך. מישהו אחר מנסה לצנן את ההתלהבות עם "שיטות הצפנה מבוססים על מספרים ראשוניים אבל לשם כך יש מספיק" - גם כן לא נכון, אבל יותר קרוב לתיאור מצב העניינים. במאמר הזה אני רוצה להבהיר את העניינים ככל הניתן. נתחיל מהשורה התחתונה - מספרים ראשוניים מהווים כיום מרכיב חשוב בחלק ממערכות ההצפנה שלנו; הם בשום פנים ואופן לא המרכיב היחיד ויש מערכות הצפנה שבהן אין כל חשיבות לראשוניים; ובכל הנוגע למציאת ראשוניים יש לנו אלגוריתמים נפלאים כיום שעובדים היטב ובלעדיהם לא הייתה שום הצפנה שמבוססת על ראשוניים, והכי חשוב: לא, לראשוני שנתגלה זה עתה אין כל קשר לכל זה.

מהו מספר ראשוני קל מאוד להגדיר: זה מספר טבעי גדול מ-1 שמתחלק רק ב-1 ובעצמו. למשל 2, או 17, או 131. לעומת זאת 57 אינו ראשוני כי הוא המכפלה של 3 ו-19. למה הראשוניים אמורים לעניין מישהו? ובכן, יש מספר סיבות. המיידית מביניהן היא שכל מספר טבעי גדול מ-1 ניתן להציג בתור מכפלה של ראשוניים באופן שהוא פחות או יותר יחיד. כך למשל את 57 אפשר לתאר בתור 3 כפול 19 או בתור 19 כפול 3, אבל פרט להיפוך הסדר הזה אין שום דבר שאפשר לעשות. כדי להבין מה מיוחד כאן כדאי לחשוב על מספר כמו 60, שאפשר להציג בתור 2 כפול 30 וגם בתור 4 כפול 15 - כלומר, שתי מכפלות שונות - אבל עדיין, הפירוק של 60 למכפלה של ראשוניים בלבד הוא יחיד (2 כפול 2 כפול 3 כפול 5). בשל התכונה הזו נהוג לומר על הראשוניים שהם "אבני הבניין" של כל המספרים הטבעיים. נראה בהמשך עוד תכונות של הראשוניים שהן מעניינות.

המתמטיקאים התעניינו במספרים ראשוניים כבר משחר המתמטיקה; אחת ההוכחות הידועות ביותר במתמטיקה היא ההוכחה של אוקלידס לכך שיש אינסוף ראשוניים (נניח שיש מספר סופי שלהם, אז בואו נכפול את כולם ביחד ונוסיף 1; קיבלנו מספר שאינו מתחלק על ידי אף אחד מהראשוניים במכפלה ומכאן שהוא חייב להתחלק על ידי ראשוני חדש, שונה מכולם) ואחת התוצאות המפורסמות ביותר במתמטיקה היא משפט המספרים הראשוניים, שמתאר במובן מסויים את ה"צפיפות" של המספרים הראשוניים בתוך קבוצת המספרים הטבעיים. גם הבעיה הפתוחה המפורסמת במתמטיקה, השערת רימן, קשורה בקשר בל ינתק למספרים הראשוניים (היא שקולה לטענה שמהווה חיזוק רב עוצמה של משפט המספרים הראשוניים). עם זאת, לאורך כל תולדותיה של המתמטיקה העיסוק במספרים ראשוניים נותר בגדר עיסוק פנים-מתמטי בלבד, שמטרתו העיקרית היא לספק את סקרנותם של המתמטיקאים. ציטוט ידוע של המתמטיקאי ג'. ה. הארדי, מהמתמטיקאים הבולטים שעסקו בתורת המספרים בחצי הראשון של המאה ה-20, על כך שהוא שמח שהתחום שבו הוא עוסק לא מועיל לשום דבר מעשי (בהנגדה לתורת היחסות שהובילה לפצצת האטום).

זה השתנה בצורה מוחלטת עם הקריפטוגרפיה של שנות השבעים. אבל קריפטוגרפיה היא תחום עתיק יומין, ותורת המספרים היא שחקן חדש יחסית בו. איך זה קרה?

ראשית, חשוב להעיר שיש תחומים רחבים בקריפטוגרפיה שאינם עושים שימוש במספרים ראשוניים או בתורת המספרים. הדוגמה הבסיסית ביותר היא אלגוריתם ההצפנה AES - דה פקטו אחד מאלגוריתמי ההצפנה הנפוצים בעולם היום, שבו ההצפנה מתבצעת על ידי ביצוע שוב ושוב של סדרה של פעולות פשוטות ביותר על ההודעה שרוצים להצפין. התחום העיקרי (אם כי לא היחיד) שבו תורת המספרים נכנסת לתמונה היא עם שיטות הצפנה ששונות מהותית באופיין מאשר AES. הצפנת AES היא מה שמכונה "הצפנה סימטרית" - כדי לפתוח קובץ שהוצפן עם AES, צריך לדעת את אותה סמא שבאמצעותה הקובץ הוצפן. זה שימושי מאוד במקרים רבים, אבל לא כאשר רוצים לתקשר עם שרת מרוחק שמעולם לא היה לך קשר אליו עד כה ובוודאי שאין לכם סמא משותפת. כדי לפתור את הבעיה הזו הומצאו שיטות הצפנה שונות, א-סימטריות: "הצפנת מפתח פומבי". בהצפנה כזו ישנן שתי סמאות - הפומבית והפרטית. אני מגלה לכל העולם את הסמא הפומבית שלי וכל מי שרוצה להצפין משהו ולשלוח לי עושה זאת באמצעות הסמא הפומבית; אבל כדי לפתוח קובץ שהוצפן באמצעות הסמא הפומבית חייבים את הסמא הפרטית, שאותה יש לי ולי בלבד. בהערת אגב, בעולם האמיתי הצפנות א-סימטריות והצפנות סימטריות עובדות יחד בהרמוניה - משתמשים בהצפנה א-סימטרית כדי להסכים על סמא משותפת, ואז שאר התקשורת מתנהלת בהצפנה סימטרית (שכן השיטות הסימטריות כיום מהירות ואמינות משמעותית יותר מאלו הא-סימטריות).

הרעיון של הצפנת מפתח ציבורי הוצע באופן פומבי לראשונה בשנת 1976 במאמר של דיפי והלמן, אלא שהם לא הצליחו לגלות שיטה מעשית שתאפשר הצפנת מפתח ציבורי. עם זאת, הם הציעו שיטה לשיתוף מפתחות - שיטה שבה שני צדדים מרוחקים בלי ידע מוקדם מסוגלים ליצור ססמא סודית שתהיה משותפת לשניהם ולא תהיה ידועה לאף אחד שמצותת לתקשורת ביניהם (אבל, וזו החולשה הגדולה של האלגוריתם - אם מישהו יצליח להשתלט על קו התקשורת ביניהם הוא יהיה מסוגל להטעות את שני המשתתפים ולגרום להם לשתף מפתח **איתו**). השיטה הזו מעניינת במיוחד מכיוון שהיא משתמשת במספרים ראשוניים, ובאופן שמבהיר יפה את השימוש העיקרי שלהם בקריפטוגרפיה: כפל מודולו p כאשר p הוא מספר ראשוני.

"כפל מודולו p" הוא דרך לתאר פעולת כפל רגילה של שני מספרים, שאחריה מחלקים את התוצאה ב-p ונשארים עם השארית. למשל, אם p הוא 17, אז 8 כפול 5 מודולו p יחזיר 6, שכן 8 כפול 5 הוא 40, וכשמחלקים ב-17 מקבלים מנה 2 ושארית 6. כפל מודולרי שכזה קל מאוד לממש במחשב, ויתרונו בכך שהוא נותן **מבנה** יפה לקבוצת המספרים מ-0 ועד $p-1$, שאסמן מעתה ואילך ב-Z_p. מבחינה מתמטית המבנה הזה נקרא **שדה**, וזוהי דרך אחרת לומר שאפשר להגדיר עליהם פעולות של כפל וחיבור (גם חיבור מוגדר מודולו p) כך שכל כללי החשבון שאנחנו מכירים ואוהבים יתקיימו: כלל החילוף, כלל הקיבוץ וכלל הפילוג, ובנוסף לכך לכל איבר יהיה **נגדי** ביחס לחיבור (מספר שאם מחברים אותו למספר המקורי מקבלים 0; הנגדיים של המספרים הטבעיים ביחס לפעולת החיבור הרגילה הם המספרים השליליים) וחשוב מכל - לכל איבר יהיה **הופכי** ביחס לכפל, כלומר אפשר "לחלק". הנה דוגמה: אם אנחנו עובדים מודולו 17, אז כאשר כופלים את 5 ב-7 מקבלים 35, ואחרי חלוקה ב-17 ולקיחת שארית מקבלים 1. זה אומר ש-7 הוא ההופכי הכפלי של 5, ובמקום "לחלק ב-5" (פעולה שלא באמת מוגדרת עבור מספרים שלמים) אפשר לכפול ב-7.

עוד תכונה רלוונטית היא שלכל מספר ראשוני p קיים מספר g ששייך ל-Z_p בעל התכונה שהחזקות $g^0,g^1,\ldots,g^{p-1}$, כשמסתכלים עליהן מודולו p, הן בדיוק כל האיברים של Z_p (למעט 0). מספר g כזה נקרא **יוצר** של Z_p.

עכשיו אפשר להסביר איך שיטת החלפת המפתחות של דיפי-הלמן עובדת: שני הצדדים, שאקרא להם אליס ובוב, מסכימים ביניהם על p ועל g מתאים עבורו (אין צורך לשמור אותם בסוד). אז אליס מגרילה לעצמה x ובוב מגריל לעצמו y ששניהם מספרים בין 1 ו-p-1 . עכשיו אליס מחשבת ושולחת לבוב את g^x ואילו בוב מחשב ושולח לאליס את g^y. כעת כל אחד מעלה את המספר שהוא קיבל מהשני בחזקת המספר שהוא הגריל. למשל, אליס קיבלה את g^y, אז היא תעלה את זה בחזקת x, ומחוקי החזקות הרגילים, שמתקיימים גם עבור הכפל של Z_p יתקיים $(g^y)^x=g^{xy}$. באופן דומה החישוב של בוב יניב את $(g^x)^y=g^{xy}$.

כך קרה שאליס ובוב מחזיקים כעת שניהם במספר משותף - g^{xy}, אבל האם מישהו שציתת לתקשורת ביניהם יודע מהו? הוא יודע מהו g^x ומהו g^y, אבל לא ברור איך לגלות מכך מהו g^{xy}. על פניו, אפשר אולי לחשוב שאם התוקף יודע מהו g (זה הרי מידע פומבי) ויודע מהו g^x הוא יוכל לגלות מכך את x, אבל זו בעיה **קשה** מבחינה חישובית, ואפילו יש לה שם - בעיית הלוגריתם הדיסקרטי. אפשר, כמובן, לנסות את כל הערכים האפשריים של x עד שמגיעים לאחד הנכון (להעלות את g בחזקה שלהם ולראות אם קיבלנו את g^x) ולכן חשוב שיהיו **המון** ערכים אפשריים של x; כמו כן צריך להתגונן בפני שיטות חיפוש מחוכמות יותר (ויש כאלו) ולכן כדי שהשיטה של דיפי-הלמן תהיה בטוחה חייבים לעבוד עם מספר ראשוני p שהוא גדול יחסית - בן מאות ספרות (מספרים כמו 2048 ביטים או 4096 ביטים הם סדרי הגודל הנפוצים בימינו בדיבורים על ראשוניים בקריפטוגרפיה).

דיפי-הלמן ממחיש יפה איך ראשוניים עוזרים לנו בקריפטוגרפיה. לב-לבו של האלגוריתם הוא בכך שיש פונקציה **שקל לחשב** אבל **קשה להפוך** - העלאת g בחזקה, במקרה שלנו. התכונה היפה הזו קיימת ב-Z_p אבל היא לחלוטין לא קיימת במספרים שלמים או ממשיים "רגילים". זו בדיוק הסיבה שהקריפטוגרפים נדחפו להשתמש במשהו כמו Z_p - זה התגלה בתור "שדה משחק" מתאים לצרכים של הקריפטוגרפיה.

שנה אחרי דיפי והלמן התפרסם מאמר של ריבסט, שמיר ואדלמן (RSA) שהציג מערכת הצפנה פומבית של ממש. הרעיון של RSA היה שימוש בפונקציה מסוג שנקרא Trapdoor Function: פונקציה שקל לחשב ובאופן כללי קשה להפוך, אבל אם יש לך מידע (סודי) נוסף, היפוך שלה הופך לקל. באופן די מעניין, RSA עובד מעל Z_n עבור n שאינו ראשוני, מה שאומר ש-n אינו שדה - לא תמיד אפשר לבצע בו חלוקה - אבל דווקא בגלל שהוא קצת "שבור" יש בו פונקציית מלכודת.

אם כן, הרעיון הוא כזה: נניח שאני רוצה להקים מערכת מפתח פומבי שבה כל העולם יוכל לשלוח לי דברים מוצפנים אבל רק אני אוכל לפענח. מה שאני עושה ראשית כל הוא למצוא שני מספרים ראשוניים גדולים p,q. כעת אני כופל אותם ומקבל $n=pq$. אחר כך אני מוצא זוג מספרים e,d בעלי התכונה ש-$ed-1$ מתחלק ב-$(p-1)(q-1)$. לא אסביר כעת את המתמטיקה המדויקת שמאחורי העניין, אך התכונה הזו של e,d מבטיחה שיתקיים הדבר הבא: $(M^e)^d=M$, כאשר החשבון מבוצע מודולו n.

כעת, אני מפרסם לעולם כולו את n ואת e, אבל מותיר את d סודי. אם מישהו רוצה להצפין ולשלוח לי הודעה M, הוא מחשב את M^e מודולו n ושולח לי. כדי לפענח, אני מעלה בחזקת d את מה שקיבלתי. פשוט להחריד. כאן פונקציית ה-Trapdoor היא פשוט העלאה בחזקת e, וה"מידע נוסף" שהופך אותה לקלה להיפוך הוא d.

כעת אנו מגיעים לנקודה שלדעתי גורמת לבלבול הגדול ביותר בקרב הטוקבקיסטים שציטטתי לעיל. כדי לבנות את מערכת ה-RSA, אחרי שמחליטים על n ועל e אפשר לחשב את d מתוך e ומתוך $(p-1)(q-1)$. במילים אחרות, מי שמכיר את $(p-1)(q-1)$ ואת e יכול לפרוץ את ההצפנה. קרוב לודאי שאתם מקבלים

תחושה ש-RSA מאוד פגיעה בשל כך, אבל כדאי לזכור שב-RSA משתמשים כל הזמן, בכל מקום. אז למה זה עובד? כי גם אם יש לי את $n=pq$, זה לא אומר שאני יכול לחשב מתוכו בקלות את $(p-1)(q-1)$; הדרך הברורה לעשות זאת היא קודם כל לפרק את n לגורמים, כלומר למצוא את p,q, אבל **בעיית הפירוק לגורמים היא בעיה קשה**. שימו לב: בעיית הפירוק לגורמים, לא בעיית בדיקת הראשוניות אלו שתי בעיות שונות, ובעיית הפירוק לגורמים מאז ומעולם נחשבה לקשה יותר.

במבט ראשון לא כל כך ברור למה הבעיות הללו שונות. לכאורה, כדי להראות שמספר הוא לא ראשוני צריך להציג פירוק שלו לגורמים. השיטה הנאיבית הידועה לבדיקת ראשוניות ("עד השורש") פשוט עוברת על המחלקים הפוטנציאליים של המספר אחד אחד עד שהיא מוצאת אחד. אלא שבמתמטיקה יש שיטות מחוכמות הרבה יותר לבדיקת ראשוניות, שיטות שמאפשרות לגלות שמספר אינו ראשוני לא בגלל שמצאנו גורם שלו, אלא בגלל שמשהו ב"עולם" לא מתנהג כמו שצריך - יש איזה שהוא גליץ' במטריקס. אתן דוגמה קטנה לאופן שבו דברים יכולים להשתבש (אם כי בפני עצמה התכונה הזו לא מספיקה כדי לבדוק ראשוניות - צריך לשלב אותה עם עוד משהו).

התכונה נקראת "המשפט הקטן של פרמה" וקובעת שאם p הוא ראשוני ו-a הוא מספר כלשהו ב-Z_p, אז $a^{p-1}=1$ (כשהחשבון הוא מודולו p). אם נתונים לנו p,a אז קל ומהיר למדי לבצע את החישוב של a^{p-1} (איך? זה עניין לפעם אחרת). אם נקבל משהו ששונה מ-1, אז **מובטח** לנו ש-p לא היה ראשוני, למרות שאין לנו שום מחלק שלו. וזו רק תכונה אחת מני רבות. אנחנו מסתמכים כאן בצורה חזקה על כך שראשוניים הופכים מבנים מתמטיים ל"יפים", במובן זה שיש תכונות נחמדות מסויימות שמתקיימות בהם, ואם משהו משתבש לעתים קרובות קל לגלות זאת.

אם כן, אם נחזור לטוקבקיסט שאמר "אין אלגוריתם או תוכנה שמסוגלת לחשב מספרים ראשוניים. לכן זאת נחשבת ההצפנה הטובה ביותר", הנה הטעות שלו: דווקא יש אלגוריתמים מצויינים שיודעים למצוא מספרים ראשוניים. יתר על כן: בלעדי אלגוריתמים שכאלו מספרים ראשוניים לא היו בעלי ערך רב בקריפטוגרפיה. שימו לב שבשביל RSA מי שמייצר את המערכת חייב לעבוד עם שני ראשוניים "סודיים" - אסור שיהיה קל למישהו לנחש עם איזה ראשוניים הוא בחר לעבוד. לכן לא נכון לומר ש"יש מספיק" ראשוניים - כל מי שרוצה לבנות מערכת הצפנה צריך להגריל ראשוניים גדולים אחרת הוא מסתכן בכך שיהיה קל לפרוץ אותו. למרבה המזל יש **המון** ראשוניים בסדרי הגודל המתאימים.

כעת בואו נחזור לראשוני הגדול ביותר שנתגלה עד כה. האם הוא רלוונטי להצפנה בצורה כלשהי? לחלוטין לא. ראשית, הוא גדול **מדי**. בדוגמאות של דיפי הלמן ושל RSA ראינו שהאופן שבו מנצלים ראשוניים הוא בביצוע פעולות חשבוניות על מספרים שהם בערך מאותו סדר גודל כמו הראשוניים הללו. עכשיו, פעולות חשבוניות פשוטות כמו חיבור, כפל, העלאה בחזקה וכדומה דורשות זמן שהוא פרופורציוני ל**מספר הספרות** של המספרים שעליהם מבצעים אותן (או פרופורציוני בריבוע/בשלישית, תלוי איזו פעולה). כלומר, כדי לחבר שני מספרים בני 100 ספרות נצטרך לבצע בערך רק 100 פעולות - לא רע,

בהתחשב בכמה שהמספרים הללו גדולים. לרוב שיטות ההצפנה בימינו די לנו במספרים של כמה מאות ספרות, מקסימום אלפי ספרות. לעומת זאת, בראשוני החדש יש בערך שבע-עשרה וחצי מיליון ספרות, מה שאומר שמערכת הצפנה שתתבסס עליו תהיה איטית למדי. האם היא גם תהיה בטוחה הרבה יותר? לא בהכרח, כי הנה החסרון הנוסף של המספר הזה - כולם מכירים אותו. אם עכשיו כולם יתאחדו ויחשבו על דרכים מועילות לפרוץ מערכות הצפנה שמבוססות ספציפי על המספר הראשוני הזה, יש סיכוי שהם יצליחו לגלות תכונות או קיצורי דרך שיעזרו להתגבר עליו. זה נכון, כמובן, לכל מספר ראשוני; ולכן עדיף לעבוד עם ראשוניים אקראיים בכל פעם שבה בונים מערכת הצפנה חדשה ולא להסתמך על אחד קיים (גם זה לא בהכרח מדויק - יש מערכות הצפנה שכן מבוססות על ראשוניים "מוכרים", אבל את הבעיה שתיארתי עדיין צריך להביא בחשבון).

אפשר אולי עוד היה לקוות שגילוי הראשוני החדש יעיד על שיפור משמעותי ביכולת שלנו למצוא מספרים ראשוניים, אבל אפילו זה לא נכון. הראשוני הזה, כמו פחות או יותר כל הראשוניים הגדולים שהתגלו בעשורים האחרונים, הוא מצורה מאוד מיוחדת - 2^{n-1} עבור ערך ספציפי של n. ראשוני כזה נקרא **ראשוני מרסן**, וידועים בדיוק 48 כאלו - כמעט כלום. אם כן, איך קרה ה"מזל" הזה שהראשוני שנתגלה היה דווקא מהצורה הזו? כמובן שלא במקרה: יש אלגוריתם יעיל מאוד לבדיקה האם מספר מהצורה 2^{n-1} הוא ראשוני או לא - יעיל משמעותית יותר מאלגוריתמים שמטפלים במספרים "כלליים", ולכן מוצלח יותר במציאת ראשוניים גדולים משמעותית מאלו שהשיטות הכלליות יודעות למצוא. אם כן, כדי למצוא "סתם" ראשוניים אקראים למערכת ההצפנה שלנו אין בכלל טעם להשתמש בו - אם אנחנו רוצים ראשוני שהוא מספר מרסן אנחנו פשוט יכולים לבחור מתוך הרשימה של הראשוניים הידועים, ואם חשוב לנו מספר אקראי, נצטרך להשתמש באלגוריתמים הרגילים.

אז אם שואלים אתכם בשביל מה גילוי הראשוני החדש טוב, תגידו שזה בשביל חדות הגילוי המתמטי. לעומת זאת אם שואלים אתכם בשביל מה מספרים ראשוניים טובים באופן כללי, אני מקווה שכעת יותר ברור לכם כיצד הם רלוונטיים לקריפטוגרפיה.