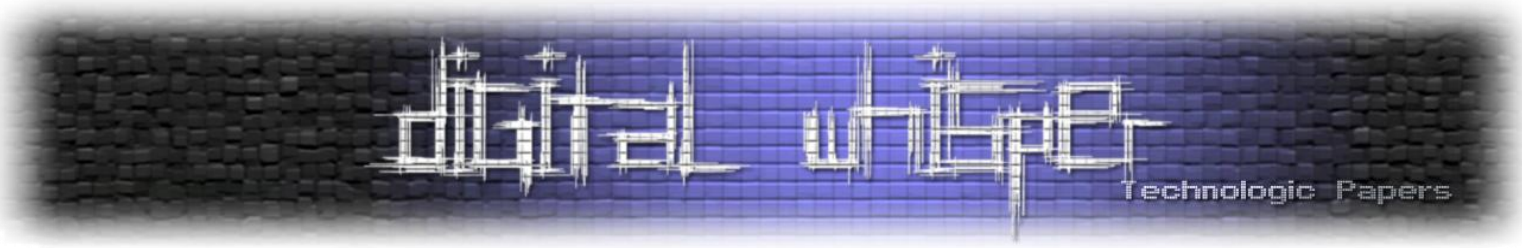

הרקע המתימטי של RSA, או: איך הצפנת RSA עובדת?

מאת: בעז (tsabar)

הקדמה

הצפנת RSA תופסת מקום מכובד ביותר בחיינו במאה ה-21. אם ננסה לדמיין את עולמנו כיום בלי היכולת של הצפנה ע"י מפתח ציבורי, כנראה שהעולם היה אחר בתכלית. כל נקודה פיזית על הקו, כל שרת או נתב בדרך, היה יכול להפוך מאוד בקלות לנקודת האזנה לכל פרט מידע אשר יוצא ונכנס בשער. בין אם זה אימיילים, סיסמאות, מצב חשבון הבנק או מספר כרטיס האשראי ששלחנו בטופס המקוון ל-Ebay.

סקירה היסטורית קצרה

הצפנה סימטרית היא הצפנה מאוד קלה: שני הצדדים נפגשים מעל תווך מוצפן (בד"כ פגישה פיזית), ומתאמים מפתח ואלגוריתם הצפנה. הבעיה ה"רקורסיבית" היא שבשביל ליצור תווך מוצפן, צריך תווך מוצפן. הצפנות סימטריות קיימות כבר אלפי שנים, בין אם באלגוריתמים פשוטים כמו של יוליוס קיסר (צופן ההזזה), או הצפנת ויז'נר שנפוצה במאה ה-16. המחשבים ויכולת החישוב האלקטרוני הקפיצו לגבהים חדשים את המודעות להצפנה חזקה, "אמיתית" מבחינה מתימטית, ולא כזו שהסוד שלה הוא צורת ההצפנה (האלגוריתם עצמו).

עד לתחילת שנות השבעים היה עוד אגוז קשה לפיצוח: כל ההצפנות, גם הטובות שבהם, היו סימטריות. בזמנו כבר היה מדובר באלגוריתמים קצת יותר מתוחכמים מהצפנים שפוענחו ידנית, אבל המחשבה על כך שצריך תווך מוצפן (או שליח שניתן לסמוך עליו ב-100%) היא מחשבה מטרידה בהתחשב בעובדה שלא תמיד יכולים שני הצדדים לתאם מפתח מוסכם מראש, כזה שאף אחד אחר לא ידע מהו.

הצפנה א-סימטרית, כזו שהמפתח לפענוח שונה ממפתח ההצפנה, היא לא דבר מובן מאליו. לפני שאנחנו מדברים על הצפנה א-סימטרית, צריך להוכיח שזה בכלל אפשרי - שניתן לעשות תרגילי לוליינות

מתימטית על מערכת מספרים ומשוואות, כך שניתן להצפין בצורה אחת ולפענח בצורה אחרת, ושאי אפשר לפענח בעזרת מידע המופק מתוך המפתח שאיתו מצפינים. את ההוכחה שאפשר ליצור תווך מוצפן בפרוטוקול שעובר כולו בתווך בלתי מוצפן (במקרה של RSA זה נוצר ע"י מפתח פומבי להצפנת הודעות ומפתח פרטי לפענוח, ושליחת המפתח הפומבי בלבד), נתנו החוקרים ויטפילד דיפי (Whitfield Diffie) ומרטין הלמן (Martin Hellman) במאמרם המהפכני "New Directions in Cryptography" משנת 1976. במאמר זה הם הוכיחו שניתן ליצור מפתח פרטי להצפנה סימטרית מעל תווך לא מוצפן, כך שמי שמאזין להודעות העוברות ברשת לא יוכל לגלות את המפתח. המאמר לא דיבר על RSA, ואני לא אדבר על המאמר.

בסיס מתימטי: הצפנה למתחילים

למען הדוגמא ניקח את פונקצית ההעלאה בריבוע:

$$f(x) = x^2$$

אם זוהי ההצפנה שלי, הרי שפונקצית הפענוח תהיה הוצאת שורש ריבועי:

$$f^{-1}(x) = \sqrt{x}$$

זוהי דוגמא פשוטה ומצוינת ל"הצפנה" שבה דרך הפענוח שונה מההצפנה. זו כמובן לא באמת הצפנה, ולא רק משום שאפשר בקלי-קלות "לפענח" אותה, אלא שגם אין לה מפתח הצפנה (לחילופין, ניתן לומר שמרחב המפתחות מכיל מפתח יחיד).

באלגוריתמים של הצפנה, מקובל לכתוב את פונקצית ההצפנה ב-E (קיצור של Encrypt) ואת פונקציית הפענוח ב-D (קיצור של Decrypt). בנוסף, הודעות להצפנה מסמנים בד"כ ב-m (קיצור של message), והודעות מוצפנות מסמנים ב-c (קיצור של cipher), וכך זה יסומן בהמשך. על פי הדוגמא הקודמת, נסמן את פונקצית ה"הצפנה" שלנו כך:

$$E(m) = m^2 = c$$

ואת פונקצית הפענוח המתאימה:

$$D(c) = \sqrt{c} = m$$

כמו שבוודאי שמתם לב, הפונקציה D היא ההופכית של E, ולא בכדי. התנאי הראשון והבסיסי שצריך להתקיים, הוא שפענוח הודעה מוצפנת יהיה זהה להודעה המקורית לפני ההצפנה. בסימנים:

$$D(E(m)) = m$$

אם התנאי הזה לא מתקיים, אי אפשר לפענח הודעה מוצפנת.

> לפעמים לא צריך לפענח הודעה מוצפנת. לדוגמא: שמירת סיסמאות של משתמשים על שרת נעשית עם פונקצית hash, שדומה במאפייניה לפונקצית הצפנה שאין לה פונקצית פענוח מתאימה. בכל אימות סיסמא, מצפינים את הסיסמא שנשלחה ומשווים את הערך המוצפן לערך השמור בשרת. ככה גם אם פורצים לשרת, אי אפשר לגנוב סיסמאות.

על הודעות מוצפנות ישנן הגבלות כאלו ואחרות מפאת הפרקטיקה היישומית של אמצעי המיחשוב העומדים לרשותינו: פרוטוקול התקשורת, כמות הביטים במנת נתונים, כמות הביטים של המפתחות, יכולת חישובית מסוימת וכו'. לא תמיד אנחנו יכולים להשאיר את הנתונים האלו "לגדול כרצוננו". לדוגמא: אם ההצפנה שלי היא העלאה בריבוע, ואני מוגבל ב-4 ספרות, אז לא תמיד אני באמת יכול להעלות את המספר "הגולמי" בריבוע, ואני נאלץ לקצר אותו לרוחב של 4 ספרות בשיטת המודולו. לדוגמא:

$$E(12) = 144$$
$$E(200) = 200^2 = 40000 \equiv 0(mod 10000)$$

ננתח את מה שקיבלנו: ההצפנה של הערך 12 עובדת מצוין ושווה ל-144. ההצפנה של הערך 200 תהיה שווה ל-0. ננסה כעת לפענח את ההצפנה:

$$D(0) = \sqrt{0} = 0$$

אופס...

למרות שפונקצית הפענוח היא הופכית לפונקצית ההצפנה, קיבלנו שהפענוח, בגלל אילוצים מסוימים, לא נותן תוצאה נכונה. זה מוביל אותנו לתנאי החשוב הבא בתורת ההצפנה: פונקצית ההצפנה צריכה להיות חד-חד-ערכית. בדוגמא לעיל, הבעיה בפענוח נוצרה כי:

$$E(200) = E(0) = 0$$

זה הזמן לעבור לשלב מתקדם יותר - השלב ה"בעייתי" מבחינה מתימטית - חיפוש אחר פונקציה שתענה לדרישות מאוד מסוימות, שפונקצית הצפנה במפתח פומבי צריכה לענות עליהן:

- שלא קשה לחשב את ערכה (כי אנחנו לא רוצים שזה יקח לנו חודשים של חישוב),
- שתפעל נכון על הודעות בתחום נתון כלשהו (לדוגמא: תחום רצוי של מספרים שלמים אי-שליליים).
- שתהיה חד-חד-ערכית (ולכן יש לה פונקציה הפיכה, או פונקצית פענוח),
- שיהיה קשה מאוד לחשב את הערך המקורי לפני שעבר בפונקציה (לפי הדוגמא לעיל: בהינתן מספר כלשהו, נניח 9, יהיה קשה לחשב את השורש הריבועי שלו),
- ובנוסף להכל, עם מעט מידע נוסף על המספר (על המספר עצמו!!! לא על הפונקציה), ניתן בקלות לחשב את ערכו לפני שעבר דרך הפונקציה (בהמשך לדוגמא שלנו: בהינתן ש-9 הוא כפולה של 3, יהיה ניתן בקלות לחשב את השורש הריבועי שלו).

פונקציה כזו נקראת "פונקציה חד כיוונית עם דלת סתרים". חד כיוונית - כי חישוב ההופכי הוא משימה מורכבת ביותר (פונקציות hash הן במובן מסוים פונקציות חד כיווניות, למרות שאינן חד-חד-ערכיות וכלל אין להן פונקציה הופכית). דלת סתרים - אפשרות למידע נוסף, שבזכותו ניתן לחשב בקלות את הכיוון ההופכי של הפונקציה.

מידע מוצפן מעל מספרים ראשוניים

מהו פירוק לגורמים כולנו יודעים מכיתה ב': בהינתן מספר n, צריך למצוא את הגורמים הראשוניים שלו. ע"פ המשפט היסודי של האריתמטיקה, יש רק מכפלה אחת של מספרים ראשוניים שתתן את n כתוצאה. נרצה גם איכשהו לשלב "עליהם" מידע להצפנה. בשלב זה אנחנו אמורים להתחשב בתנאים המקדימים לכל הצפנה: פענוח נכון וחד-חד-ערכיות. הכפלה של שני מספרים ראשוניים גדולים מאוד היא במובן מסוים פונקציה חד כיוונית, ודלת הסתרים האפשרית כאן היא, באופן טבעי, אחד הגורמים של המכפלה (ראוי לציין שדלת הסתרים של RSA היא דווקא ההעלאה בחזקה מודולרית ופונקצית אוילר שמחושבת בקלות בעזרת גורמי המכפלה. תודה לגדי אלכסנדרוביץ' על התיקון).

ובכן - רונלד ריבסט (Ronald Rivest), עדי שמיר (Adi Shamir) ולאונרד אדלמן (Leonard Adelman) עשו זאת, והצליחו להלביש מידע להצפנה בצורה מחוכמת על בעית הפירוק לגורמים. הם הנציחו את שמותיהם בשם האלגוריתם - RSA - ששמו מורכב מהאותיות הראשונות של שמות המשפחה שלהם.באלגוריתם ההצפנה משתמשים בחזקות שמחושבים מתוך המספרים הראשוניים, ומודולו n, כאשר n הוא מכפלת שני מספרים ראשוניים, וממנו קשה לדעת מהם אותם מספרים ראשוניים.

המתמטיקה

האלגוריתם מתבסס על האבחנה שאם n הוא מספר כלשהו, אז עבור מספר m זר ל-n וקטן ממנו (כזה שהמחלק המשותף המקסימלי שלו עם n הוא 1- $(GCD(n,m)=1$), מתקיים $m^{\varphi(n)} \equiv 1(modn)$ [1]. בעברית: m בחזקת $\varphi(n)$ (פונקצית אוילר על n), לחלק ב-n תניב שארית 1. פונקצית אוילר של n היא כמות המספרים הזרים ל-n וקטנים ממנו.

כעת, אנחנו רוצים לחשב את פונקצית אוילר על מספרים מסוימים. פנקצית אוילר אינה קשה לחישוב אם יודעים מהם הגורמים הראשוניים של המספר עליו אנו רוצים לחשב את הפונקציה. דוגמא אחת היא מספר ראשוני: מכיוון שלכל מספר p ראשוני, כל מספר קטן ממנו (ואינו 1) הוא זר לו, תוצאת פונקצית אוילר היא p-1. דוגמא נוספת היא מכפלת שני מספרים ראשוניים: אם n הוא מכפלת שני מספרים ראשוניים שונים זה מזה, p ו-q, אז ישנם q-1 מספרים קטנים ממנו שמתחלקים ב-p, וישנם p-1 מספרים קטנים ממנו שמתחלקים ב-q.

דוגמא מספרית להמחשה:

$$n = 15, p = 3, q = 5$$

נבחן מהם המספרים **שאינם** זרים ל- 15: 3,6,9,12,5,10. יש כאן 5-1=4 מספרים שמתחלקים ב-3, ו-3-1=2 מספרים שמתחלקים ב-5.

נפתח את הנוסחא: [2]

$$\varphi(n) = (n-1) - (p-1) - (q-1) = n - 1 - p + 1 - q + 1 = \\ = n - p - q + 1 = pq - p - q + 1 = (p-1)(q-1)$$

אסביר מה הצבתי בהתחלת המשוואה:

- n-1 - כמות המספרים הקטנים מ-n (המועמדים הפוטנציאליים).
- p-1 - כמות המספרים הקטנים מ-n שהם כפולה של q (ולכן לא זרים ל-n).
- q-1 - כמות המספרים הקטנים מ-n שהם כפולה של p (ולכן לא זרים ל-n).

בסה"כ, הנוסחא הזו היא מקרה פרטי של נוסחת אוילר למציאת ערך הפונקציה.

מ-[1] ומ-[2], אפשר להסיק שעבור n שהוא מכפלה של 2 מספרים ראשוניים, p ו-q, ועבור m שאינו p או q (ולכן זר ל-n. זה אפשרי: פשוט בוחרים p ו-q שונים מ-m), מתקיים:

$$m^{\varphi(n)} \equiv 1(modn)$$

ואם נכפול את שני צידי המשוואה ב-m, נקבל כי:

$$m^{\varphi(n)+1} \equiv m(modn)$$

וזה תקף לכל מכפלה של $\varphi(n) = (p-1)(q-1)$, כי:

$$[3]\ m^{\varphi(n)} \equiv 1(modn) \Rightarrow (m^{\varphi(n)})^x = m^{\varphi(n)\cdot x} \equiv 1^x(modn) \equiv 1(modn)$$

במילים אחרות, אפשר לרשום זאת כך: יש לנו מספר כלשהו, $\varphi(n)$, שאם מעלים את m בחזקה שהיא כפולה כלשהי שלו פלוס 1, ולוקחים את שארית החלוקה ב-n, מקבלים את m עצמו. זה אכן מקיים את דרישת החד-חד-ערכיות שקבענו בתור תנאי התחלתי: אם m זה המסר שאנו רוצים להצפין, זה מבטיח שיש לנו איך לפענח אותו; אין שתי הודעות שונות שלאחר הצפנה ופענוח יתנו את אותו ערך.

הואיל ואנו רוצים להגיע למצב של העלאה בחזקה שהיא מכפלה (כלשהי) של $\varphi(n)$ ועוד 1, אנחנו צריכים לחפש פתרון לנוסחא:

$$[4]\ x \cdot y = \varphi(n) \cdot z + 1$$

נשמע קשה? ובכן - לא ממש. אוקלידס היווני כתב לזה אלגוריתם לפני 2,300 שנים, ובגירסה המורחבת של האלגוריתם למציאת מחלק משותף מקסימלי, GCD, שעליו התבססנו מקודם בשביל לוודא ששני מספרים הם זרים (המחלק המשותף המקסימלי שלהם הוא 1), נוכל למצוא בקלות ערכי y ו-z שיפתרו את המשוואה עבור ערך x נתון.

אלגוריתם אוקלידס המורחב מוצא לא רק את המחלק המשותף המקסימלי של שני מספרים a ו-b, אלא גם את הערכים שבהכפלתם ב-a ו-b יתנו את המחלק הזה. במילים אחרות, הוא מוצא גם את הערכים c ו-d במשוואה:

$$a \cdot c + b \cdot d = GCD(a,b)$$

אז קודם כל נגריל מספר כלשהו x, ונוודא שהוא זר ל-$\varphi(n) = (p-1)(q-1)$, כלומר שמתקיים $GCD(x, \varphi(n))$. אחרי שינוי קל למשוואה [4] נקבל:

$$x \cdot y - \varphi(n) \cdot z = 1$$

כעת נמשיך ונמצא בעזרת אלגוריתם אוקלידס המורחב את שאר המספרים החסרים לנו: y ו-z (ניתן להחליף ב- z-, זה לא משנה את הערך המספרי, רק את הסימן).

בשלב זה הכל מוכן: המפתח הפומבי יהיה n ו-x, ואילו המפתח הפרטי יהיה p, q ו-y. שווה לשים לב: מעכשיו אין עוד צורך בגורמים הראשוניים p ו-q בשביל הצפנה או פענוח של הודעות. המספרים האלו היו מספרי עזר לתשתית - יצירת המפתחות, אבל לא משמשים להצפנה עצמה.

ההצפנה תהיה:

$$E(m) = m^x (mod\ n)$$

הפענוח יהיה בדומה:

$$D(c) = c^y (mod\ n)$$

כעת, בהתבסס על משוואה [4], והצבתה אל תוך משוואה [3], נקבל את המשוואה שמסבירה למה פענוח של הודעה מוצפנת בשיטת RSA אכן יניב את ההודעה המקורית:

$$D(c) = D(E(m)) = D(m^x) = (m^x)^y = m^{x \cdot y} = m^{\varphi(n) \cdot z + 1} \equiv m^1 (mod n) = m$$

עד עכשיו ראינו למה זה נכון מתימטית - כלומר, למה פענוח של הודעה מוצפנת אכן מניב את ההודעה המקורית, ולמה אין שתי הודעות שונות שהפענוח שלהן יהיה זהה (תחת אותו מפתח הצפנה).

כעת נעבור לחלק האחרון, והוא החלק המעניין של הקשר המתימטי בין פירוק לגורמים לשבירת צופן RSA.

פירוק לגורמים

לפני שבכלל אמשיך, חשוב להזכיר כאן את השאלה הידועה והנושנה, שיתכן שמתקרבת לידי פתרון: האם P=NP? הפתרון המסתמן הוא "לא, הם שונים זה מזה". אם הם היו שווים, הסיפור היה נגמר כאן, והכל היה מתמוטט. אבל... זה בכלל לא חשוב לענייננו, למרות הסיכוי שאולי לא נמצא לעולם אלגוריתם יעיל לפירוק לגורמים (פולינומיאלי בכמות הביטים למפתח). אלגוריתם RSA לא הוכח כקשה כמו בעית הפירוק לגורמים, אלא זו השערה בלבד. ייתכן - והסיכוי הוא לא מאוד גדול - שבעית הפירוק לגורמים היא לא פולינומיאלית ופענוח RSA הוא כן (בעית הפירוק לגורמים היא בעיה ב-NP, וככל הנראה אינה ב--NP

Complete. לפיכך, אם P≠NP, ייתכן שהיא ב-P וייתכן שלא). אסור לשכוח שאולי קיימת שיטה אחרת (שעוד לא מצאנו) לשבור את צופן RSA בלי שימוש בגורמים הראשוניים p ו-q.

ועכשיו נחזור לענייננו: למה צופן RSA נחשב לצופן בטוח. הנחת העבודה היא, כמובן, שבשביל לשבור את צופן RSA צריך לפרק את n לגורמיו הראשוניים, ושהפירוק הזה הוא סיפור קשה.

נסתכל על המפתח הפומבי ונראה מה אפשר ללמוד ממנו: המפתח הפומבי הוא n ו-x. נשים לב ש-x הוא מספר שבחרנו בצורה רנדומלית, והקשר (העקיף) שלו ל-n הוא דרך הכפלתו ב-y. אבל - y הוא חלק מהמפתח הפרטי, ואין לנו אותו. לכן x לא מסגיר איתו שום מידע על איך לפענח.

אם אנחנו מצליחים לפרק את n לגורמיו - p ו-q, אז נוכל למצוא בקלות את $\varphi(n) = (p-1)(q-1)$, להציב בנוסחא של אוקלידס, למצוא את y, ולפענח את ההודעה המוצפנת. אבל אם אין לנו דרך לפרק את n לגורמיו (ושוב - נניח שזו הדרך היחידה לפענח את ההודעה המוצפנת), אז לא נוכל למצוא את$\varphi(n)$, ולכן לא נוכל למצוא (בזמן סביר) את המספר y, שהוא תוצאת חישוב פשוט, אם אנחנו יודעים את x ואת $\varphi(n)$. כאן, $\varphi(n)$ משמש גם כדלת הסתרים של הפונקציה.

סיכום

במאמר זה הסברתי על קצה המזלג את הפן המתימטי של RSA, והשתדלתי ככל האפשר להשאיר את הנוסחאות בחוץ, כדי לא להבריח קוראים פוטנציאליים (למי שנזכר עכשיו בסטיבן הוקינג: כן...).

המאמר תיאר את הדרישות הבסיסיות מכל הצפנה, ובפרט מהצפנה א-סימטרית. בנוסף, המאמר תמצת את הנכונות המתימטית שמאחורי RSA, מהם הרעיונות המתימטיים העומדים בבסיסה של הצפנה זו, ואיך בתוך כל הבלגן הזה, אלגוריתם בן 2,300 שנים חובר לפונקציה מהמאה ה-18 כדי לשמש תשתית להצפנה הפופולרית של תחילת המאה ה-21.

המאמר נכתב בעזרת הספר "Cryptography - Theory and Practice" של דאגלס ר. סטיבנסון.

לקריאה נוספת (ויקיפדיה)

- RSA
- פונקצית אוילר
- מחלק משותף מקסימלי
- פונקציה חד כיוונית
- פירוק לגורמים
- שאלת P=NP

על המחבר

בעז (tsabar) סיים לאחרונה בהצטיינות תואר ראשון במדעי המחשב באוניברסיטה הפתוחה. הוא נכנס לבלוגוספירה די במקרה, וכותב בלוג כבר כמעט 3 שנים. בבלוג שלו, "צבר - בלוג עם קוצים", הוא כותב על נושאים שונים ומגוונים.

שנה טובה.