קריפטוגרפיה - חלק ג'

מאת אופיר בק

הקדמה

כמנהג הסדרה, לפני שנתקדם לתוכן המאמר, אכריז על הפותר של החידה מהמאמר הקודם: **אדווין כהן**. הטקסט המוצפן היה L'albatros - שיר בצרפתית, דבר אשר הקשה מעט על הפענוח. אדווין הוא הפותר הראשון שהצליח לאחר **ש-16 לפניו נכשלו**!

וכעת, בחזרה לענייננו, בחלק הקודם עסקנו בצפנים מעט חדישים יותר מהצפנים המונואלפבתיים הבסיסיים, אם כי גם בהם מצאנו נקודות תורפה. במאמר זה אנו נתקדם לתקופה יותר מודרנית. החלטתי לדלג על מלחמת העולם השנייה, למרות שהיא שלב מדהים בהיסטוריה של ההצפנה, ובכלל.

על כן, אני אסכם את הדברים במספר פסקאות קצרות.

המחשב הראשון

גרמניה הנאצית השתמשה בצופן מכני שנקרא אניגמה לתקשורת הבסיסית. הפענוח שלו בבסיס הבריטי ארך שבועות לכל הודעה, וגרם לכך שההודעות היו מיושנות ולא רלוונטיות כשפוצחו. פריצת דרך נעשתה על ידי המדען הגאון אלן טיורינג, אשר הצליח לפתח מכונות שיעשו את מלאכת החישוב. לא נכנס לפרטים בנוגע לפעולה המדויקת של המכונות אלא נתקדם מעט הלאה:

לאחר שצופן האניגמה נפרץ, הבריטים החלו לעבוד על פיצוח הצופן המתקדם יותר של הגרמנים: **הלורנץ**. הלורנץ פעלה בדומה לאניגמה אך הייתה מסובכת הרבה יותר. ג'ון טילטמן וביל טוטה גילו חולשה בצופן והבינו איך לפצח הודעה ביד.

בדומה להתפתחות פיצוח האניגמה, גם כאן פיצוח כל הודעה ארך שבועות בתחילה, עד שהמתמטיקאי המוכשר מקס ניומן הגה דרך להשתמש במיכון גם כדי לפצח את צופן הלורנץ. לצערו, רבים טענו כי הפיתרון שלו לא ישים, ולכן בניית המכונה מעולם לא תוקצבה, אך מהנדס בשם טומי פלאוורס, שהיה מעורב בדיון על התכנון של ניומן, החליט לעבוד על מכונה משלו, ולאחר עשרה חודשים הציג את מכונת הקולוסוס.

היא הכילה 1500 מתגים חשמליים שהיו מהירים באופן משמעותי מהמתגים האלקטרומכניים במכונת הבומבס של טיורינג, והיא הייתה ניתנת לתכנות, מה שהפך אותה למחשב בר התכנות הראשון.

לאחר המלחמה, הפרויקט נסגר וכל המעורבים נאסרו לשתף את המידע אודות מעשיהם, כך שמכונה שפותחה שנתיים אחרי זה, באוניברסיטת פנסילבניה, בשם ENIAC נחשבה למחשב הראשון במשך שנים, מבלי שטומי פלאוורס ומקס ניומן זכו לכבוד הראוי להם על הפיתוח המוקדם יותר.

הצפנה בעידן המחשבים

הצפנה במחשב והצפנה מסורתית דומת זו לזו מבחינות רבות, אך מספר הבדלים משמעותיים מבחינים ביניהן:

- **הראשון**: ייצוג המידע במחשב נעשה בסופו של דבר על ידי ביטים - כלומר, כל המידע במחשב מיוצג על ידי שורות על גבי שורות הבנויות משני תווים - 0 ו-1.
- **השני**: המהירות - רכיבים אלקטרוניים נעים במהירות גבוה בהרבה ביחס לרכיבים מכניים.
- **השלישי**: יכולת ערבול. בעוד אנחנו מוגבלים במכונות למה שמעשי לבנות (ולכן לאניגמה לדוגמה היו שלושה או ארבעה מערבלי אותיות), במחשב אנחנו יכולים להניח גם מאה מערבלים וירטואליים מבלי שנפח המקום שהמחשב יתפוס יגדל.

בשל שלושת ההבדלים הללו, ניתן לבנות מכונת הצפנה תיאורטית עם מורכבות עצומה, ואין את הצורך לבנות מכונה פיזית שתממש זאת.

לצורך הנוחות, אנחנו נשתמש בפרוטוקול ASCII (הפרוטוקול הסטנדרטי ביותר להפיכת תווים לקבוצה של ביטים) כדי לייצג את האותיות, באופן הבא (בטבלה מוצגים הקודים של האותיות הגדולות באנגלית):

A	1000001	**N**	1001110
B	1000010	**O**	1001111
C	1000011	**P**	1010000
D	1000100	**Q**	1010001
E	1000101	**R**	1010010
F	1000110	**S**	1010011
G	1000111	**T**	1010100
H	1001000	**U**	1010101
I	1001001	**V**	1010110

J	1001010	**W**	1010111
K	1001011	**X**	1011000
L	1001100	**Y**	1011001
M	1001101	**Z**	1011010

בנוסף, משתמשים בפעולות לוגיות על ביטים האלה, אותן פירטתי בעבר גם באסמבלי, אך למען הנוחות אפרט אותן מחדש גם פה:

1. **AND** - משווה בין ביטים, ורק כאשר שני שווים ל-1 רושמת 1, בכל מקרה אחר - 0. כלומר שימוש בפקודה AND על 10000000 ביחד עם 11111111 תחזיר לנו 10000000 - את התחום המשותף בו לשניהם יש את המספר 1.
2. **OR** - משווה בין ביטים, ואם לפחות אחד מהם שווה ל-1, רושמת 1. בכל מקרה אחר - 0. כלומר - השימוש בפקודה הלוגית הזאת על 10000000 ביחד עם 11111111 תחזיר לנו 11111111 - התחום המשותף בו לפחות לאחד מהם יש את המספר 1.
3. **XOR** - משווה בין ביטים, ורק אם אחד מהם בלבד שווה ל-1, רושמת 1. בכל מקרה אחר - 0. כלומר - השימוש בפקודה הלוגית הזאת על 10000000 יחד עם 11111111 תחזיר לנו 01111111 - התחום המשותף בו לאחד האיברים יש 0 ולאחד האיברים יש 1.
4. **NOT** - הופכת ביטים. הפקודה הזאת מקבלת רק איבר אחד, ובכל מקום שיש 1 רושמת 0, ולהפך. כלומר - השימוש בפקודה הלוגית הזאת על 10000000 תחזיר לנו 01111111.

בעזרת הכלים הבסיסיים האלו כבר ניתן להתחיל לכתוב הצפנות בסיסיות, בעזרת שימוש במפתח כמובן.

הצפנה בסיסית ביותר לדוג' תהיה לקחת את הטקסט, לצורך העניין HELLO, ולרשום 1 בכל פעם שהביטים שונים, ו-0 כשהם שווים. נבחר לו את המפתח DAVID. את השלבים אראה מיד:

הודעה:	HELLO
מפתח:	DAVID
הודעה ב-ASCII ביטים:	1001000100010110011001001100111
מפתח ב-ASCII ביטים:	1000100100000110101101001001100100
הודעה מוצפנת:	0001100000010000110100000101000101

כעת ניתן לשדר את הרצף של 35 הביטים האלו, כשבצד השני, המקבל ישתמש באותו המפתח כדי לפענח את ההודעה.

אין ספק שמדובר בצופן יחסית בסיסי, ועם זאת, הוא סך הכל די יעיל, אך צצה בעיה בסיסית ביותר: ההצפנה הזו מוגבלת לאנשים שיש להם מחשבים, ובאותה תקופה, מדובר היה רק בממשלות וצבאות. סדרה של פריצות דרך מדעיות, טכנולוגיות והנדסיות הביאו להפצה רחבה בהרבה של מחשבים, וכתוצאה מכך, גם של הצפנה ממוחשבת.

ב-1947, המצאת הטרנזיסטור, תחליף זול למתג החשמלי הוזילה מעט את הייצור. ב-1951, החלו חברות לייצר מחשבים לפי הזמנה, וב-1953 הציגה IBM את המחשב הראשון שלה, וארבע שנים מאוחר יותר גם את שפת התכנות הנוחה (יחסית לאסמבלי) הראשונה - Fortan. ב-1959 הייתה פריצת דרך משמעותית גם כן, עם המצאת המעגל המשולב.

כך יצא שבשנות השישים מחשבים גם התחזקו מחד, וגם נעשו זולים יותר ויותר מאידך, כך שיותר ויותר עסקים החלו לרכוש מחשבים, כדי להצפין תקשורת חשובה, כדוגמת העברת כספים. בדיוק מכאן התפתחה בעיה נוספת: סטנדרטיזציה. חברה עשויה להשתמש במערכת הצפנה כלשהי כדי להצפין תקשורת נתונים פנימי, אך כיצד תוכל לשלוח הודעה חיצונית לארגון אחר, שאינו משתמש באותה שיטת הצפנה?

בסופו של דבר, באמצע שנת 1973, מכון הסטנדרדים הלאומי האמריקני החליט לפתור את הבעיה, ויצא באופן רשמי בבקשה לקבל הצעות פיתוח למערכת הצפנה סטנדרטית, שכל העסקים ישתמשו בה. מוצר בשם "לוציפר" (גם היום הרבה מתוכנות הצפנה מקבלות שמות הקשורים לגיהנום או השאול במיתולוגיות שונות, כדוגמת תוכנת "קרברוס" - כלב השאול מהמיתולוגיה היוונית), שהיה אחד האלגוריתמים הטובים ביותר להצפנה באותו הזמן, היה שייך ל-IBM.

הוא פותח על ידי מהגר גרמני בשם הורסט פייסטל (Feistel) שהגיע לאמריקה לפני מלחמת העולם השנייה, בשנת 1934. הוא הושם במעצר בית עד כניעתה של גרמניה, בשל חשד לשיתוף פעולה (שלא היה בו דבר, אבל באותה תקופה האמריקנים לא היו נאורים כפי שהם טוענים שהם). כשהוא החל סוף סוף לחקור צפנים, ה-NSA האמריקני (הסוכנות לביטחון לאומי) סיבך אותו, וגרם לכך שהפרוייקט שלו יבוטל.

בשנות ה-60 הוא עבר לעבוד בחברה בשם MITRE, אבל גם שם ה-NSA רדפה אותו והכריחה אותו לעזוב. לבסוף הוא הגיע למעבדות IBM, שם הצליח לחקור במשך כמה שנים באין מפריע, ושם פיתח את "לוציפר". לוציפר השתמשה ב"פונקציית מטחנה", דבר שעד היום אין לו הגדרה מדוייקת, אך ניתן להסביר זאת במעין מטאפורה: לוקחים גוש בצק עליו כתובה ההודעה, לשים אותו באופן מסויים מספר פעמים, כך שההודעה מעורבלת, לאחר מכן, מבצעים פעולה הפוכה בצד השני כדי לפענח.

למרות זאת, ה-NSA שוב פעם התערב בפרוייקט, וכשהוא הפיץ אותו לציבור, הוא איפשר שימוש "רק" ב-100,000,000,000,000,000 מפתחות (56 ביטים), מכיוון שה-NSA ידע שהוא לא יוכל לחדור ליכולת המלאה של לוציפר, והוא רצה לשמור לעצמו את היכולת להגיע למידע של המשתמשים במידת הצורך. ב-1976 לוציפר נבחר לשמש בתור הצופן הלאומי, ונקרא מאז DES, ראשי תיבות ל- Data Encryption Standard. גם כיום, 40 שנה לאחר מכן, הוא עדיין אחד הצופנים הרשמיים.

כעת, חברות רבות החלו להשתמש בהצפנה ממחושבת, מכיוון שהמתחרים העסקיים, ואף עסק אזרחי, לא יכלו לפענח את ההודעה המוצפנת, מכיוון שמספר וגודל המתפחות היה גדול דיו. ה-NSA לעומת זאת, היה בעל כוח חישובי מספיק גדול כדי לעשות כן.

סיכום

התקדמנו יותר לכיוון העת המודרנית, ואל ההצפנה הממחושבת, אשר גם היום אנו עושים בה שימוש (אם כי באופן מעט שונה, עליו ארחיב בפעמים הבאות). בפעם הבאה נתעסק בבעיות נוספות שצצו בהצפנה הממוחשבת, ואיך חוקרים שונים שקדו כדי לפתור אותן.

על המחבר

שמי אופיר בק, בן 17 מפתח תקווה. אני לומד בתוכנית "גבהים" של מטה הסייבר הצה"לי וב- C-Security. קשה למצוא חומר מעודכן יחיסת בעברית, ומאחר והמגזין הזה העניק לי כל כך, הרגשתי חובה לתרום בחזרה. מקווה שנהניתם. ניתן ליצור עמי קשר בכתובת: ophiri99@gmail.com.

קריאה נוספת

- אלן טיורינג:

https://en.wikipedia.org/wiki/Alan_Turing

- צופן הלורנץ:

https://en.wikipedia.org/wiki/Lorenz_cipher

- הקלוסוס:

https://en.wikipedia.org/wiki/Colossus_computer

- מחשב ENIAC:

https://en.wikipedia.org/wiki/ENIAC