Technologic Papers

# פונקציות תמצות קריפטוגרפיות ותחרות ה-SHA3

מאת ד"ר אור דונקלמן, אוניברסיטת חיפה

## הקדמה

פונקציות תמצות (Hash function) הן כלי תכנותי שימושי. היכולת של פונקציה לקבל כל אורך קלט, ולייצר תמיד את אותו אורך פלט, מאפשרת בניית מבני נתונים יעילים (Hash tables), אשר בתורם מאפשרים להאיץ ביצועי מחשבים, ולשפר אלגוריתמים (חפשו את הביטוי "Cuckoo hashing" לקרוא על אחד ממבני הנתונים החדשים והמרתקים שיש לעולם התיאוריה של מדעי המחשב להציע).

במקביל לפונקציות התמצות הרגילות, שעיקר תפקידן הוא לייצר מחרוזות באורך קבוע, קיימות פונקציות תמצות קריפטוגרפיות, שאכן מייצרות מכל קלט מחרוזת באורך קבוע, אבל באופן בטוח. הבעיה העיקרית העומדת בפני מי שבא להגדיר את פונקציות התמצות הקריפטוגרפית, היא מה פירוש הדרישה שהתמצית הנוצרת תהיה בטוחה. בעוד הקהילה הקריפטוגרפית מתחבטת בסוגיה הקריטית הזו, נהוג לציין שלוש דרישות בנוגע לבטיחות של פונקציית תמצות קריפטוגרפית h(M):

- (בטיחות מקור) בהנתן h(M), קשה למצוא 'M כך שמתקיים h(M) = h(M')
- (בטיחות מקור שני) בהנתן M, קשה למצוא 'M כך שמתקיים h(M) = h(M').
- (בטיחות התנגשות) קשה למצוא M ו-'M כך שמתקיים h(M) = h(M').

התפישה הרווחת בנוגע לפונקציות תמצות קריפטוגרפיות בטוחות היא שהן מייצרות ערכי תמצית שנראים אקראיים למדי. תכונה זו, יחד עם יצירת הפלט באורך קבוע, הפכה את פונקציות התמצות הקריפטוגרפיות לפופלריות מאוד: הן בשימוש בחתימות אלקטרוניות (כאשר הנוהג הוא לחתום על תמצית של ההודעה, ולא על ההודעה המקורית), בקבצי סיסמאות (כאשר נשמר בקובץ ערך התמצית של הסיסמא, מה שמונע קריאת הסיסמא מהקובץ), בגזירת מפתחות קריפטוגרפיים לשימוש (לדוגמא, בפרוטוקול IPsec), ובעוד שורה ארוכה של יישומי אבטחת מידע. לפיכך, יש רבים המדמים את פונקציות התמצות הקריפטוגרפיות לאולר השיווצרי הקריפטוגרפי - כלי שיכול לעשות הכל.

יש לציין כי פונקציות תמצות קריפטוגרפיות משמשות גם לדברים "לא אבטחתיים". לדוגמא, מכיוון שהן נורא רגישות לשינויים בקלט, אחת הדרכים לדעת האם קובץ השתנה (או התקבל בצורה לא נכונה), היא

לשמור בצד את תוצאת הפעלת פונקציית התמצות עליו, ולבדוק האם הקובץ החדש נותן את אותה תמצית. זאת הסיבה שבלא מעט מקרים ניתן לקבל לא רק את הקובץ עצמו, אלא גם את ה-md5sum שלו (שהוא פשוט תוצאת הפעלת פונקציית התמצות MD5 על הקובץ).

למרות העובדה שאכן פונקציות תמצות קריפטוגרפיות משמשות במגוון רחב של פרוטוקולים ויישומים, אנשי התיאוריה של הקריפטוגרפיה טרם הצליחו להגדיר את כל הדרישות מהן (בשונה מצפנים, עבורם המודלים התיאורטיים מוגדרים היטב). במשך שנים התחושה בקהילה הקריפטוגרפית הייתה שמדובר בבעיה של אנשי התיאוריה, מכיוון שאנשי המעשה, מתכנני פוקנציות התמצות, הציגו סדרה של פונקציות שהיו חזקות ומהירות כאחת, דוגמת MD5 ו-SHA1.

## תור הזהב

יש לציין כי מרבית פונקציות התמצות מורכבות משני חלקים - פונקציות דחיסה (compression functions) ואופן שרשור (mode of iteration). עם הצגתה של פונקציית התמצות MD4 ע"י רון ריבסט ב-1989, התפתח הנוהג לבנות את פונקציות הדחיסה על בסיס שילוב פעולות XOR, חיבור (של מספרים בני 32-ביט), והזזות ציקליות, ואת פונקצית השרשור לבסס על אופן בשם Merkle-Damgard. לראיה, אחרי הצגת MD5 בשנת 1991 (שהיוותה שיפור של MD4 שסבל מכמה בעיות אבטחה קלות), לא הצליחו חוקרים ל"שבור" את הפונקציה במשך זמן רב.

יתר על כן, בשנת 1993, הכריזה ממשלת ארה"ב על תקן פונקצית תמצות קריפטוגרפית בשם SHA, המבוסס על אותם עקרונות כמו MD5 (עם כמה שינויים קטנים), ואחרי שנתיים, הם עדכנו את התקן, והציגו את SHA1. שתי הפונקציות - SHA1 ו-MD5, שלטו בכיפה. הן היו מהירות, נחשבו בטוחות, והקהילה הקריפטוגרפית סברה שבכך נגמר הדיון על פונקציות תמצות. מלבד אורכי הפלט הקצרים משהו (128-ביט ל-MD5, ו-160-ביט ל-SHA1), שהביא להצגתן גם של פונקציות התמצות ממשפחת SHA2 (עם אורכי פלט של 224, 256, 384 או 512 ביט), התחושה הרווחת הייתה שזהו, הינה דוגמא לבעיה שפתרנו היטב.

## הגירוש מגן עדן

ואז הגיעה שנת 2004. באותה שנה, נחשפה התקפה למציאת התנגשויות בפונקציית התמצות MD5. ההתקפה, שהוצגה ע"י פרופ' שיאוון וונג (Xiaoyun Wang, משמשת כיום כחברת סגל באוניברסיטת Shandong היוקרתית) הראתה כי ניתן למצוא התנגשויות בפונקציות תמצות כגון MD4, MD5, ואפילו SHA1. ההתקפות של פרופ' וונג על MD4 היו כל כך יעילות, שאת כל החישובים הדרושים אפשר ניתן

לבצע בחישוב ידני! ההתקפה על MD5 לקחה כשעת ריצה, ויצרה התנגשויות, וההתקפה על SHA1 הראתה שבטיחותה כפונקציית תמצות איננה מושלמת.

מאז הצגת ההתקפות של וונג, ההתקפות על פונקציות תמצות רק הלכו והשתכללו. היום, אנחנו יודעים למצוא התנגשויות ב-MD5 במספר שניות (אפילו על המחשב ששימש את וונג למצוא התנגשות בשעה), ובטיחותה של SHA1 מוטלת בספק לנוכח קיומן של מספר התקפות (אם כי, אף אחת מאלה לא מומשה לחלוטין).

מלבד ההתקפות הללו, החל מחקר אינטנסיבי ביותר בנוגע למה לעשות עם אותן התקפות. יש להבין שהתנגשויות בפונקציות הדחיסה בעצמן נחשבות הרסניות ביותר, אבל הן לא תמיד מתרגמות להתקפות שימושיות כנגד פונקציית התמצות, מכיוון שההתנגשות מבוססת לרוב על ערכים שנראים אקראיים למדי. לכן פרוטוקולים הנסמכים על פונקציות התמצות יכולים עדיין להיות בטוחים לשימוש. לדוגמא, התנגשות בפונקציית הדחיסה, לא מקלה על מציאת שני מסמכי PDF בעלי אותו ערך תמצות...

בעקבות מחקריו של מרקוס דאום וסטפן לוקס (Markus Daum, Stefan Lucks), חוקרים הצליחו להראות כיצד ניתן לבנות שני מסמכי postscript אשר חולקים ערך תמצית זהה (תוך ניצול התנגשות שיוצרה ע"י הטכניקה של וונג). מאוחר יותר, התוצאות הללו הורחבו לשני מסמכי וורד, PDF, ואפילו תמונות TIFF. התקפה זו היא אלגנטית ביותר, וניתן בקלות להעביר אותה לכל מבנה קבצים אשר מאפשר בשפה שלו מבני if-else. לדוגמא, ההתקפה על postscript מבוססת על הרעיון הבא: נניח כי מצאנו זוג ערכים X ו-Y, כך שהם מתנגשים עבור פונקציית הדחיסה של MD5 (לא פונקציית התמצות!). במקרה שכזה, ניתן לייצר שני קבצי postscript, אשר ידפיסו למסך (ולמדפסת) שני מסמכים שונים למראה, אך חולקים ערך MD5 (של פונקציית התמצות המלאה).

המסמך הראשון יהיה מהצורה (בכתיב C, הצורה המדויקת של postscript שונה במעט, אבל בעלת אותו עקרון):

```
temp = X;
if (temp == X)
    print (Document 1)
else
    print (Document 2)
```

בעוד שהקובץ השני יהיה מהצורה:

```
temp = Y;
if (temp == X)
    print (Document 1)
else
    print (Document 2)
```

קל לראות, כי עבור הקובץ הראשון, המסמך שיודפס יהיה Document1, בעוד שהקובץ השני ידפיס את Document2. עם זאת, מהרגע ש-X ו-Y גורמים להתנגשות בפונקציית הדחיסה, מכיוון שהמשך הקובץ זהה, הרי שבהכרח גם תווצר התנגשות בערך פונקציית התמצות המלאה, מה שמבטיח ששני הקבצים יהיו בעלי ערך MD5 זהה.

סדרת מחקרים אחרת הראתה כיצד לנצל את הטכניקות של וונג (ושל רבים אחרים), יכולה לשמש לייצור סרטיפיקטים מזויפים. המחקר (שהרוח החיה בו היא מארק סטיבנס) הראה בהתחלה כיצד לייצר שני סרטיפיקטים עבור אותה יישות, אבל עם מפתחות פומביים שונים. סרטיפקט אחד נחתם ע"י ה-CA, והחתימה (בגלל צורת החתימה) תקפה מיידית גם לסרטיפיקט הזה. לאט לאט שוכללו השיטות, עד שמארק הצליח לייצר שני סרטיפיקטים - האחד, על שמו, והשני סרטיפיקט מסוג מיוחד, שמצהיר כי בעליו הוא CA בעצמו! כלומר, לאחר השגת החתימה על הראשון, היה למארק סרטיפיקט שהוא עצמו CA, ולכן הוא יכל לייצר סרטיפיקטים כאוות נפשו! (יש לציין שהדרך בה סרטיפיקטים משמשים, מאפשרת ל-CA שכולם מכירים, דוגמת Verisign, לתת לגופים אחרים סרטיפיקט שהם עצמם CA-ים). לפרטים נוספים אתם מוזמנים להציץ באתר של מארק - http://www.win.tue.nl/hashclash/rogue-ca.

במקביל, הבטיחות של Merkle-Damgard החלה להתערער. סדרה של מאמרים הראתה כל מיני בעיות בפונקציות המתבססות על אופן השרשור הזה, כולל התקפות למציאת מקור שני. אם נוסיף לכך את העובדה שכמעט כל פונקציות התמצות שהוצגו לפני שנת 2000 נשברו (נכון להיום, רק משפחה אחת של פונקציות תמצות מלפני שנת 2000 נחשבות בטוחות - Ripemd), והעובדה שמשפחת SHA2 תוכננה תחת אותם עקרונות של SHA1 (שנחשבת לא בטוחה מספיק), ומשתמשת ב-Merkle-Damgard, הביאה לכך שהעולם נהפך על פיו (אוקי, לפחות מי שמתעסק בפונקציות תמצות).

## למקומות, היכון, רוץ!

מאז הצלחת התחרות לבחירת המחליף של ה-Data Encryption Standard (תחרות ה-AES שנערכה בשנים 1997-2000, ובסיומה זכה Rijndael והפך ל-Advanced Encryption Standard), נחשבות תחרויות קריפטוגרפיות כדרך למצוא פונקציות קריפטוגרפיות טובות. הרעיון הבסיסי הוא שכל מומחה מכין את ההצעה שלו, אשר משולחת לחופשי. כל ההצעות עוברות הן אנליזת חוזק והן אנליזת ביצועים (בחומרה ובתוכנה). לכל מומחה יש אינטרס לנסות למצוא בעיות אבטחה אצל המתחרים, וכמובן שמומחי מימוש מתחרים ביניהם מי יצליח לממש את הפונקציות בצורה מהירה יותר.

לכן, החליט ה-National Institute of Standards and Technology, לערוך את תחרות SHA3. כל מי שרצה התבקש לשלוח את ההצעה שלו לתחרות, ואכן, 64 הצעות שונות נשלחו ל-NIST, אשר בתורו קיבל כ-51 כעומדות בדרישות מינימום (קוד ב-C, הסבר ברור דיו, ועוד כמה סעיפים טכניים). אותן 51 הצעות פורסמו באתר של NIST, באתרים של המחברים, ומיד נבחנו. חלק גדול מההצעות נשבר כבר בכמה חודשים הראשונים לאחר מועד השליחה לתחרות (27 הצעות בסה"כ נשברו ברמה זו או אחרת). במקביל, זמני הריצה של המועמדים נמדדו, ולאחר שנה של ניתוחים ובדיקות, בחרו ב-NIST (מתוך 24 ההצעות הבטוחות) רשימה של 14 פונקציות שעברו לשלב השני. כעת, משצומצמה רשימת המועמדים, ניתן היה להתרכז יותר, ולדון באופן יותר פרטני במועמדים, הן מבחינת אבטחה, והן מבחינת מימוש.

השלב השני של התחרות ארך כשנה וחצי, והוא הסתיים בדצמבר האחרון. בסיומו, מתוך 14 המועמדים שנותרו (שמתוכם אחד נשבר, כנגד 8 היו התקפות שאינן שוברות את הפונקציה עצמה, אבל לא אמורות להיות שם), נבחרו 5 הצעות (חלקן מה-8), ועברו לשלב השלישי והאחרון של התחרות. יש לציין שבכל מעבר בין השלבים, הורשו המגישים לעדכן קלות (tweak) את האלגוריתם שלהם בדרכים שמשפרות את האבטחה (או הביצועים, או שניהם). ואכן, ארבעה מתוך חמשת המועמדים בסיבוב האחרון אכן שינו את האלגוריתם שלהם.

בעוד כשנה בערך, תערוך NIST כנס אחרון בנושא מי מחמשת המועמדים האחרונים הוא הכי מתאים להיות SHA3, ולאחר מכן, תבחר אחד מ-Blake, Grostel, JH, Keccak או Skein, כפונקציית התמצות הבאה. כמובן, שכל אחת מהפונקציות הללו יש לה יתרונות וחסרונות ביחס לאחרות, ונראה כי כמעט כל בחירה ש-NIST תעשה תהיה סבירה (מלבד JH, לכל אחת מהפונקציות האחרות יש מספיק יתרונות להפוך אותן לבחירה טובה). עד אז, יאלצו מי שמחליפים את המערכות שלהם להחליט האם לעבור ל-SHA2 (שנחשבת בטוחה עדיין), או לחכות כבר ל-SHA3.

## גם אתם יכולים לעזור

מי מכם אשר מתעניין בנושא, יכול לעזור ל-NIST להגיע להחלטה נכונה. התחרות לוקחת בחשבון שני פרמטרים - חוזקן של פונקציות התמצות ויעילותן (בתוכנה ובחומרה). אם אתם מתכנתים טובים (או שולטים בתכן רכיבי ASIC או FPGA), הקהילה הקריפטוגרפית תודה לכם אם תנסו לכתוב מימושים יותר מוצלחים (מהירים/קטנים/חוסכי אנרגיה וכו') עבור הפונקציות הנותרות. גם אם אין לכם את הזמן לכתוב מימושים חדשים, אתם יכולים לעזור ע"י הרצת תוכנה המודדת זמני ריצה של המועמדים השונים. התוכנה זמינה מאתר eBASH בכתובת: http://bench.cr.yp.to/ebash.html.

משם אתם יכולים להוריד את גרסת ה-supercop האחרונה, ולתת לה לרוץ על המעבדים שברשותכם, כדי למדוד את זמני הריצה של מימושים רבים. הגרסא גם מכילה מספר מימושים קודמים של המועמדים, וברור ששיפורם יעזור לזיהוי מי מהמועמדים הוא הכי יעיל.

ברור שאם תצליחו למצוא חולשה באחד מהמועמדים - נשמח לדעת!

## על המחבר

ד"ר אור דונקלמן, מהחוג למדעי המחשב באוניברסיטת חיפה, הוא חוקר פעיל בתחום הקריפטאנליזה (שבירת צפנים), אבטחת מידע, ופרטיות. אור פרסם מספר רב של מאמרים הבודקים את חוזקם של צפנים ומערכות קריפטוגרפיות. מבין מחקריו, בולטות עבודותיו בנוגע לאבטחה של ה- Advanced Encryption Standard (AES), צופן הבלוקים KASUMI (המשמש לצורך הגנה על תעבורת 3G), וכן צופן ה-KeeLoq, המשמש במערכות שלט רחוק לכניסה, כגון מערכות אזעקה.

בנוסף, אור עובד על שיפור התקפות קריפטאנליטיות והמצאת טכניקות חדשות. חלק מפיתוחיו בתחום שבירת הצפנים, איפשרו את ההתקפות האחרונות על צופן ה-AES המלא, וכן מספר צפנים נוספים.

אור סיים את לימודי הדוקטורט בפקולטה למדעי המחשב בטכניון בשנת 2006, ומאז עבד במספר מוסדות מחקר (Katholieke Universiteit Leuven, Ecole Normale Superieure, ומכון וייצמן למדע), טרם הגעתו לאוניברסיטת חיפה.

אתרו:

http://www.cs.haifa.ac.il/~orrd

## כמה אתרים קשורים שעשויים לעניין אתכם:

| | |
|---|---|
| האתר הרשמי של התחרות: | http://csrc.nist.gov/groups/ST/hash/sha-3/index.html |
| אתר SHA3-zoo המרכז תוצאות על המועמדים: | http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo |
| האתר של Blake (המועמד המועדף עלי): | http://www.131002.net/blake |
| האתר של Grostel (קרויה על שם מאכל אוסטרי מפורסם): | http://www.groestl.info |
| האתר של Keccak (שבין מתכנניו, יואן דימן ממציאי ה-AES): | http://keccak.noekeon.org |
| האתר של Skein (בין מתכנניו, ברוס שנייר, איש אבטח מידע המפורסם): | http://www.skein-hash.info |