Multicollisions in iterated hash functions. Application to cascaded constructions

Antoine Joux

רז חילו

Introduction

- פונקציות תמצות הן פונקציות קריפטוגראפיות, אשר פועלות על קלטים בגדלים גדולים מאוד (לדוגמא 2^{64}) ופלט בגודל קבוע.
- האבטחה העיקרית של פונקציות תמצות קריפטוגראפיות היא שאין תוקף שיוכל למצוא שתי הודעות M, M' שמובילים לערכי תמצות זהים.
- ידוע כי אין פונקצית תמצות החסינה מפני פרדוקס יום ההולדת, כלומר בהינתן פונקצית תמצות עם פלט של n ביטים ניתן למצוא התנגשות ב- $2^{n/2}$ הודעות שונות.

Introduction- Cont.

- מסיבה זאת פונקציות תמצות עם פלטים קטנים מ- 160 ביט נחשבות מיושנים, למרות שבעבר הוצאו הרבה פונקציות תמצות עם פלטים בגודל 128 ביט וחלקם עדיין בשימוש (לדוגמא SHA, MD_4, MD_5).
- אנחנו נעסוק בפונקציות תמצות שנבנו עם איטרציה של פונקציות דחיסה.
- מטרת המאמר- לפתור בעיה: **"האם שרשור של שני ערכי פונקציות תמצות עצמאיות, יותר מאובטח מערך תמצות יחיד?"**

Basic facts about iterated hash functions

- פונקצית תמצות איטרטיבית נבנית על ידי חזרה על פונקצית דחיסה פשוטה.
- פונקצית הדחיסה f מקבלת שני קלטים : ערך משורשר ובלוק הודעה ומוציאה כפלט את הערך המשורשר הבא.
- לפני העיבוד ההודעה "מרופדת" ומחולקת למספר בלוקים השווים בגודלם.
- "ריפוד" ההודעה מתבצע על ידי הוספת '1' בתחילת ההודעה ושרשור מספר לא מוגבל של '0'-ים (בחיזוק נוסף ניתן להוסיף גם את אורך ההודעה).
- שיטה זאת נקראת Merkle-Damgard Strengthening.

Basic facts about iterated hash functions- Cont.

- לאחר מכן ההודעה מחולקת ל- ℓ בלוקים, מציבים ערך התחלתי ראשון והאיטרציה מתחילה.
- האלגוריתם נראה כך :

-Pad the message and split it into blocks $M_1,....M_\ell$.

-Set Ho to the initial value IV.

-For i from 1 to ℓ, let $H_i=f(H_{i-1},M_i)$.

-Output $H(M)=H_\ell$.

Basic facts about iterated hash functions- Cont.

- מושגי יסוד:
  - התנגשות (collision)- כאשר שתי הודעות שונות M, M' מביאות אל אותו ערך, כלומר נקבל H(M)=H(M'). מתקפה קלה למציאת התנגשות זה הגרלת הודעות בצורה רנדומאלית וחישוב ערכם עד שמקבלים התנגשות.
  - Preimage resistance- כאשר מקבלים ערך y ומוצאים הודעה M כך ש H(M)=y.
  - Second preimage resistance- כאשר מקבלים הודעה M מוצאים עוד הודעה כך ש H(M)=H(M').
- התקפות מול שלושת מטרות אלו עולות בערך 2^n .

Basic facts about iterated hash functions- Cont.

- הגדרה:

r-way collision (r-collision)- כאשר יש סדרת הודעות $M_1,\ldots,M_r$, כך ש $H(M_1)=\ldots=H(M_r)$.

בהנחה שערכי התמצות מתנהגים בצורה כמעט רנדומאלית, מציאת r-collision יכולה להתבצע על ידי תמצות בערך $2^{n\cdot\frac{(r-1)}{r}}$ הודעות.

Basic facts about iterated hash functions- Cont.

- הגדרה:

r-way collision (r-collision)- כאשר יש סדרת הודעות $M_1,\ldots,M_r$, כך ש $H(M_1)=\ldots=H(M_r)$.

בהנחה שערכי התמצות מתנהגים בצורה כמעט רנדומאלית, מציאת r-collision יכולה להתבצע על ידי תמצות בערך $2^{n\cdot\frac{(r-1)}{r}}$ הודעות.

- שאלה:

למראית עין נראה שקשה יותר למצוא r-collision, האם זה נכון?

Constructing multicollision

- נראה שיצירת multicollision בפונקצית תמצות איטרטיבית יכולה להעשות בצורה די יעילה.
- אם נדייק יצירת $2^t - collision$ עולה t פעמים בניית 2-collision פשוטה.
- הערה:

  ניתן להתעלם מתהליך ה"ריפוד" אם נניח שמדובר על הודעות באותם אורכים.
- על מנת לפשט את הבעיה, נניח שגודל בלוקי ההודעה גדול יותר מגודל הערך המוחזר מפונקצית התמצות, אך הפיתרון תקף גם למקרה השני.

Constructing multicollision-Cont.

- הנחות:
  - גודל בלוקי ההודעה גדול יותר מגודל הערך המוחזר מפונקצית התמצות (הפיתרון תקף גם למקרה שלא).
  - קיימת מכונה C שבהינתן כקלט ערך המוחזר מפונקצית תמצות h, מחזיר שני בלוקים שונים B,B' כך ש: $f(h,B)=f(h,B')$.

הערה:

התכונה הכי חשובה של C היא ש- C תפעל עבור כל ערך משורשר.

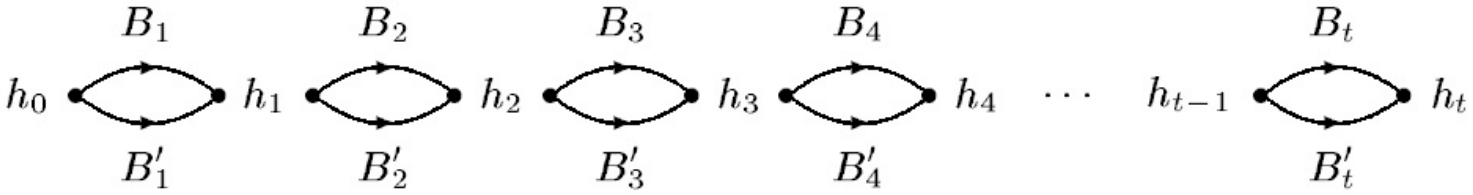
Constructing multicollision-Cont.

- נראה מציאת 4-collision בעזרת C :
  - נגדיר ערך התחלתי IV.
  - נקרא ל- C בפעם הראשונה על מנת להשיג שני בלוקים B_0,B'_0 שיש ביניהם התנגשות, כלומר $f(IV,B_0)=f(IV,B'_0)$.
  - נסמל את הערך המוחזר ב- z ונפעיל את C שוב ונמצא עוד שני בלוקים B_1,B'_1 שיש ביניהם התנגשות, כלומר $f(z,B_1)=f(z,B'_1)$.
  - אם נשים את שני השלבים האלו ביחד נקבל את ה- 4-collision שרצינו :

$$f(f(IV,B_0),B_1)=f(f(IV,B_0),B'_1)=f(f(IV,B'_0),B_1)=f(f(IV,B'_0),B'_1)$$.

Constructing multicollision-Cont.

- באותו רעיון נראה את האלגוריתם לבניית $2^t - collision$:
- Let h_0 be equal to the initial value of IV of H.
- For i from 1 to t do:
  - Call C and find B_i and B'_i such that $f(h_{i-1}, B_i) = f(h_{i-1}, B'_i)$.
  - Let $h_i = f(h_{i-1}, B_i)$.
- Pad and output the 2^t messages of the form $(b_1,...,b_t, Padding)$ where b_i is of the two blocks B_i or B'_i.



On the security of cascaded hash functions

- שיטה קלה לבניית פונקצית תמצות גדולה היא לבצע שרשור של פונקציות תמצות קטנות יותר.

לדוגמא:

בהינתן שתי פונקציות תמצות F ו- G והודעה M, ניתן ליצור ערך תמצות גדול $F(M)||G(M)$.

במבנה זה F ו- G יכולים להיות פונקציות תמצות שונות לגמרי או שני אינסטנסים שונים במקצת של אותה פונקצית תמצות.

On the security of cascaded hash functions-Cont.

- Collision resistance:
  - אם נניח ש- F מוציא n_f ביטים כפלט, ו- G מוציא n_g ביטים כפלט.
    לכן רמת האבטחה של F היא $2^{n_f/2}$ ולגבי G $2^{n_g/2}$.
  - אם F||G היה פונקצית תמצות טובה, הסיבוכיות של ההתקפה הטובה ביותר הייתה $2^{(n_f+n_g)/2}$.
  - נראה התקפה עם סיבוכיות של $n_g 2^{n_f/2} + 2^{n_g/2}$ אם $n_f \le n_g$.
  - להלן ההתקפה:
    - בעזרת האלגוריתם שהוסבר מקודם (המשתמש במכונה C), עם t שווה $n_g/2$ (עיגול כלפי מעלה) נבנה $2^t - collision$ על F. זה עולה לנו t קריאות אל אלגוריתם פרדוקס יום ההולדת הבסיסי על פונקצית הדחיסה של f, בערך $t2^{n_f/2}$ פעולות.

On the security of cascaded hash functions-Cont.

- מכאן נקבל 2^t הודעות שונות בעלות אותו ערך בצד של F.
- מכיוון שהגדרנו את t שיהיה שווה ל- $n_g/2$ נוכל לבצע את פרדוקס יום ההולדת על 2^t האלמנטים שיש לנו, ובהסתברות סבירה לצפות שנקבל התנגשות עם n_g ביטים שנמצאים ב- 2^t הודעות של G.

הערה: כדי להגדיל את אחוז ההצלחה ניתן להגדיל את ערך t ולבצע יותר קריאות על ההתקפה על F.

- ניתוח סיבוכיות:
- ביצוע t פעמים את האלגוריתם על F: $t2^{n_f/2}$.
- השוואה של G אל 2^t הודעות בגודל t: $t2^t$ במימוש נאיבי, 2^t במימוש בעזרת סידור ההודעות בעץ.
- סהכ: $t2^{n_f/2}+2^t=n_g 2^{n_f/2}+2^{n_g/2}$.

On the security of cascaded hash functions-Cont.

- Preimage and second-preimage resistance :
  - אם נניח ש- F מוציא n_f ביטים כפלט, ו- G מוציא n_g ביטים כפלט. לכן רמת האבטחה של F היא 2^{n_f} ולגבי G 2^{n_g} בנוגע ל-preimage.
  - האלגוריתם הגנרי הידוע ביותר לשבירת preimage זה לנסות הודעות רנדומליות עד שמקבלים את הערך הרצוי.
  - אם F||G היה פונקצית תמצות טובה, הסיבוכיות של ההתקפה הטובה ביותר הייתה $2^{(n_f+n_g)}$.
  - נראה התקפה עם סיבוכיות של $n_g 2^{n_f/2} + 2^{n_f} + 2^{n_g}$ אם $n_f \le n_g$.

On the security of cascaded hash functions-Cont.

- להלן ההתקפה:
  - בעזרת האלגוריתם שהוסבר מקודם (המשתמש במכונה C), עם t שווה n_g נבנה $2^t - collision$ על F. זה עולה לנו t קריאות אל אלגוריתם פרדוקס יום ההולדת הבסיסי על פונקצית הדחיסה של f, בערך $t2^{n_f/2}$ פעולות.
  - לאחר מכן נחפש בלוק נוסף הממפה את הערך המשורשר האחרון שלנו אל הפלט הסופי של הפונקציה F.
  - לאחר צעד זה אנחנו משיגים 2^t הודעות שונות עם ערך התמצות המצופה בצד של F.
  - מכיוון שהגדרנו את t שיהיה שווה ל- n_g אנחנו מצפים שבהסתברות סבירה לפחות הודעה אחת מ- 2^t הודעות שווה בערכה ל- n_g ביטים בצד של G.

On the security of cascaded hash functions-Cont.

- ראינו כי המתקפה למעלה מוצאת preimage עבור ערך רנדומלי.
- כמו כן התקפה זאת יכולה להיות גם עבור second-preimage ללא שינויים.

- במקרה ונרצה לבצע כל אחת מן המתקפות על יותר משתי פונקציות תמצות נגדיר את מכונה C אשר תמצא התנגשות בשרשור של כל הפונקציות פרק לאחת ונבצע את האלגוריתמים שהוצגו.

  לדוגמא: עבור F,G,H נגדיר כי מכונה C תעבוד על F||G.