# שיעור 3

בהרצאה קודמת הגדרנו מערכת הצפנה סימטרית.

* דרשנו נכונות מושלמת כחלק מהגדרת מערכת הצפנה
* כלומר, לכל הודעה m

Pr(dec(k,enc(k,m))=m)=1

* הגדרנו מתי מערכת הצפנה מקיימת בטיחות מושלמת אם לכל m0,m1 ולכל cמתקיים ש pr(enc(k,m0)=c) = pr(enc(k,m1)=c)

ראינו שמערכת ההצפנה ODP מקיימת את הדרישה הזו, אולם היא איננה בטוחה עבור הצפנה של מעל 2 הודעות. (כלומר היא לא מקיימת את הדרישה שעבור כל 2 זוגות ווקטורים של הודעות לא ניתן להבחין בין ההצפנות של 2 הווקטורים המוצפנים, כי לשניהם אותה ההתפלגות).

התחלנו לראות שכל מערכת הצפנה דטרמיניסטית (הכוונה בדטרמניסטית שפונקציית הenc שלה היא דטרמיניסטית, פונקציית gen נשארת רנדומלית) לא מסוגלת לספק בטיחות להצפנה של מעל 2 הודעות.

**הוכחה:** תהי (gen,enc,dec) מערכת הצפנה עם פונקציית enc דטרמיניסטית. נתבונן בוקטורים

M1=(m0,mo) M2=(m0,m1), m0!=m1

במקרה הראשון נקבל תמיד הצפנה מהצורה c0,c0, זה נובע מהדטרמיניסטיות של ההצפנה. במקרה השני ההצפנה תהיה מהצורה enc(k,m0)=c0, enc(k,m1)=c1. בגלל נכונות מערכת ההצפנה בהכרח c0!=c1,שאם לא כן בתהליך הdec של c1 ששווה לc0 האלגוריתם לא ידע להחליט לאיזו הודעה (m0 או m1) שייכת ההצפנה, והוא לא ידע מה לפלוט.

לכן, תמיד ניתן להבחין בין הצפנה של M1 לבין הצפנה של M2 (הsupports שלהם זרים)

(הגדרה: supports אלו המקרים שההסתברות שלהם איננה אפס)

בעצם, אפשר לדעת בוודאות אם ההצפנה שלפנינו היא של M1 או של M2.

נחזור לכמה דוגמאות הצפנה עתיקות שראינו ונראה כיצד הן לא מקיימות את דרישת הבטיחות של 2 הודעות ומעלה.

**צופן קיסר Ceaser Cypher:**

כזכור הצופן הוגדר בתור M=C=Z26^l כאשר l>1 לדוגמא l=3 (l הוא מספר תווים בהודעה. אם ארוכה מדי מחלקים קצרה מדי שמים padding כלומר תוספות)

K=Z26 (כלומר מרחב המפתחות הוא 26)

:Gen(r) פולט את k שהוא המרכיב הרנדומלי, שייך לZ26

Enc(k,m) : יהי m הודעה שנראית כך m=m1m2m3…ml לדוגמא m מחרוזת באורך 3 תווים.

הפלט c יראה כך c=(m1+k)(m2+k)(m3+k)…(ml+k) כאשר הסכום הוא מודולו 26

Dec(k,c): הפלט יהיה m שיראה כך m=(c1-k)(c2-k)(c3-k)…(cl-k)

קל לראות שהמערכת מקיימת נכונות.

נראה אינה מקיימת בטיחות מושלמת (לשתי הודעות ומעלה):

לצורך הפשטות ניקח l=3. נראה שקיימים m0,m1 וc כך שההסתברות לקבלת c מתוכם איננה שווה

Pr(enc(m0,k)=c)!=pr(enc(m1,k)=c)

לדוגמא – m1=dog, m2=cat, c=cat

Pr(enc(m2,k)=c)=1/26 מכיוון שמתוך 26 אפשרויות לk קיימת אפשרות אחת שתעביר את הההודעה cat לצופן cat, כאשר k=0

לעומת זאת

Pr(enc(m1,k)=c)=0. זה נכון כי אף אחד מ26 האפשרויות למפתח לא הופך את dog לcat. ניתן לראות זאת גם מבלי לעבור על כל האפשרויות, כי c1-c2=m1-m2 (ההפרש בין dog לcat לא אחיד בין כל תו ותו)

**צופן החלפה Subtitustion Cypher:**

המפתח מגדיר לכל אות בא"ב האנגלי אות חלופית שתהווה הצפנה שלה. לכל אות תהיה הצפנה אחרת משיקולי נכונות (אחרת לא נדע לאן לחזור במהלך אלגוריתם dec)

M=C=Z26^l עבור l כלשהו (כזכור l הוא גודל ההודעה)

K=Z26,26 כלומר מחליפים לכל אות מה26 אות אחרת מה26)

Get(r): פולט את kשהוא מרכיב האקראיות

Enc(m,k) : עבור k=k1,k2,k2… נפלוט c=c1,c2,c3 ע"פ המפתח

Dec(k,c) נפלוט את m כאשר m יחליף כל אות בc ע"פ המפתח.

קל לראות שהמערכת מקיימת נכונות.

כעת נראה שהיא איננה מקיימת בטיחות ברמה מושלמת ל2 הודעות ומעלה:

(בפועל, אפשר לשבור את הבטיחות יותר חזק ממה שנכתוב באמצעות אנליזה של התפלגות אותיות בשפה האנגלית. ברור למשל שהאות e מופיעה יותר פעמים מx, אך נסתפק בהוכחה פשוטה בשלב זה)

ההתקפה הקודמת לא תעבוד כי מהמילה dog בהחלט ניתן לקבל cat ע"י מפתח מתאים.

נבחר m0=target m1=strong c=target

Pr(enc(k,mo)=c)=21!/26!

כדי שההצפנה של target תביא לנו הודעה מוצפנת target אנחנו זקוקים למפתח שמעביר את האותיות t.a.r.g.e אל עצמם. יש 21! מפתחות כאלה מתוך כלל 26! המפתחות. (נדמיין 26 פסים \_ \_ \_ \_ \_ שכל אחת מייצגת אות. ב5 האותיות t,a,r,g,e ישנה אפשרות אחת, שהאות תעבור לעצמה. בפס הבא יש לאות 21 אפשרויות, לבאה אחריה 20, אחריה 19 וכו').

לעומת זאת

Pr(enc(k,m1)=c)=0. כלומר אין התכנות שהמילה strong תוצפן לצופן target. כי בצופן taget האות הראשונה והאחרונה זהות.

**(detour) – מבוא לאלגוריתמים הסתברותיים**

ראינו שהשימוש באקראיות הוא קריטי במערכות הצפנה (אחרת אין בטיחות למעל הודעה בודדת). אך לאקראיות יש שימוש נוסף, בדרך כלל בייעול אלגוריתמים ולפעמים כאמצעי ליישום נכון.

לדוגמה, עבור בדיקת ראשוניות היו ידועים בשנות ה70 אלגוריתמיים הסתברותיים שמאפשרים לכל קלט סיכויי הצלחה גבוהים אך לא מושלמים שרצו בסיבוכיות O(nlogn) .

רק ב2001 מצאו אלגוריתם דטרמיניסטי שמצליח לנבא מספרים ראשוניים ב100% הצלחה בסיבוכיות O(n^6) ומסתמך על אלגברה יותר כבדה.

דוגמה קלה לאלגוריתם הסתברותי (נזכיר שאלגוריתם כזה מקבל קלט r ומניח שהוא נדגם אקראית מתוך קבוצה סופית של קלטים אפשריים בד"כ מניחים ש R={0,1}^l):

הקלט הוא מערך של ביטים באורך n, כשאר חציות אפסים וחציו אחדות. צריך למצוא ביעילות אינדקס מסוים שהביט שלו הוא 1. אלגוריתם דטרמיניסטי ירוץ על חצי מהקלט, ברגע שימצא 1 יחזיר את האינדקס שלו ואם הוא לא ימצא אחד יחזיר את האינדקס הn/2+1.

אלגוריתם הסתברותי יעיל יותר פשוט יקח את 1000 ביטים ויבדוק אותם. אם ימצא 1 יחזיר את האינדקס, ואם לא, יחזיק אינדקס שהוא לא בדק. מכיוון שההסתברות לקבל 0 הוא 0.5 ההסתברות שהאלגוריתם יכשל הוא 2^-1001

כעת נראה שדרישת הבטיחות למערכת הצפנות הבטוחות למעל הודעה אחת דורשת מפתחות מאוד ארוכים. מרחב המפתחות חייב להיות בגודל של מרחב ההודעות. זו מגבלה משמעותית שנרצה לעקוף בעתיד.

**משפט Shannon:**

תהי (gen,enc,dec) מערכת הצפנה סימטרית עם בטיחות מושלמת(סימטרית הכוונה שלשתי הצדיים אותו המפתח) אזי |K|>=|M|

כלומר מרחב המפתחות גדול לפחות כמו מרחב ההודעות.

הוכחה: נניח בשלילה שקיימת מערכת הצפנה כזו עם בטיחות מושלמת ש |K|<|M|

ניקח m1 ששייך לM, r ששייך לR וk ששייך לK, c1=enc(k,m1,r)

לכן pr(enc(m1,k,r)=c1)>0 שכן הצלחנו לקבל c1

נתבונן בקבוצת ההודעות mEM עבורן קיימים k,r כך ש c1=enc(k,m,r)

לכל מפתח יש לכל היותר הודעה אחת שהופכת לc1. לא ייתכן שיהיה יותר מהודעה אחת אחרת הנכונות תפגע (שכן dec לא ידע לאיזה מן ההודעות עליו לחזור עם המפתח והצופן שבידו). כמות המפתחות היא קטנה מכמות ההודעות, לכן ישנה הודעה אחת לפחות m0 שאין לה מפתח להגיע לc1 (שובך היונים). וכבר הגענו למסקנה שיש הודעה m1 שיש לה האפשרות להגיע לc1.

P1(enc(m1,r,k)=c)!=pr(enc(m0,r,k)=c)=0<0

לכן הבטיחות איננה מושלמת, סתירה.

**הצעות לשיפור:**

נחליש במידה מסוימת את הבטיחות. כיוון מתבקש הוא שנאפשר לכל תוקף ללמוד מעט. כל תוקף בעולם יוכל ללמוד לכל היותר אפסילון מידע. נצטרך להגדיר מה הכוונה "אפסילון מידע". אינטואיטיבית נרצה שההתפלגות של הצפנות m0,m1 יהיו דומות ככל האפשר. ננסה להשתמש בהגדרה הסטנדרטית למרחק סטטיסטי בין התפלגויות, בתקווה שהוא ישקף לנו את מידת היכולת של תוקף ללמוד מידע.

הגדרה 3.1:

יהיו D0,D1 התפלגויות על קבוצה סופית המרחק הסטטיסטי ביניהם יוגדר

SD(D0,D1)=0.5\* sigma[pr(x=d)-pr(x=d)]

כאשר הסיגמא רצה על כל האפשרויות בהתפלגויות ומחשבת את הפרשי ההסתברות ביניהם. הx הראשון שייך להתפלגות D0 הx השני שייך להתפלגות D1