קריפטוגרפיה- הרצאה 5  
  
בשיעור הקודם: התאכזבנו סופית מבטיחות סטטיסטית, וחישבנו מסלול מחדש.

הרעיון הכללי: נפתח מערכות הגנה שבטוחות רק נגד תוקפים "מוגבלים בזמן".

* הבטיחות תסתמך על הנחות קושי שאין להן הוכחה. אבל הנחות קושי יהיו הנחות לגבי קושי אלגוריתמי של בעיות מתמטיות מסוימות (יותר "נקי" מהנחה ישירה שמע' ההצפנה שלנו היא בטוחה). בהינתן הנחות הקושי, נוכיח בצורה מדויקת בטיחות של מע' ההצפנה – ע"י רדוקציות. נראה דרך לפתור את הבעיה המתמטית B שאין לה פתרון יעיל ע"י שימוש בתוקף (היעיל) של מע' ההצפנה. כלומר זו רדוקציה מהבעיה המתמטית לבעיה של שבירת מע' ההצפנה.  
  כיוון שהנחנו שלא ניתן לפתור את הבעיה המתמטית, נובע שלא נתן לשבור את מע' ההצפנה. (אם ההנחה נופלת, אז הבטיחות גם).

הערה: בתחומים אחרים בתיאוריה של מדעי המחשב – חישוביות , תיאוריה של NP-שלמות, משתמשים ברדוקציות כדי להראות קושי של בעיות על סמך קושי של בעיה אחרת שהקושי שלה ידוע/יותר "אמין".

נתחיל בהגדרות של אבחנה שמחלישות במובן שאנחנו רוצים את ההגדרה של בטיחות סטטיסטית.   
תזכורת:

זוג התפלגויות הן ɛ-קרובות סטטיסטית ( ( ɛ -statistically indistinguishable

אם לכל A ɛ>=|Pr(A(d)=1)-Pr(A(d)=1)|

1D-> d d<-D0

אנחנו נרצה לדבר על קושי אבחנה עבור תוקפים מוגבלים בזמן החישוב שלהם.

(מכונות טיורינג המבצעות ≤t צעדים (מס' קונקרטי כרגע)

ניסיון הגדרה 1 (גם משתמשים, אבל אנחנו נעבור בד"כ לדבר על בטיחות במובן אסימפטוטי)

נקח לדוגמא ɛ=10-6 , t=2400 , D0,D1 התפלגויות על 2048{0,1} .

נאמר ש D0,D1 (ɛ,t)- computationally indistinguishable או בקיצור (ɛt,)-indisinguishable

אם לכל A שעל קלטים באורך 2048 עושה ≤ t צעדים מתקיים

ɛ>=|Pr(A(d)=1)-Pr(A(d)=1)|

1D-> d d<-D0

הגדרת בטיחות קונקרטית כמו זו שראינו בדוגמא (עם פרמטרים קבועים) אכן שימושית בפועל, נרצה הגדרה אסימפטוטית – כלומר נדבר על סדרה של התפלגויות מעל תחומים הולכים וגדלים. אחת הסיבות היא שזה יותר נח מתמטית. נדבר במקום על זוג התפלגויות D0,D1 על סדרות אינסופיות של התפלגויות (ensembles of distributions) D0,1,D0,2…. D1,1,D1,2….

וt(n), ɛ(n) והתווך lm){0,1} של D0,n, D1,n כלשהו יהיו פונק' של n.

כעקרון lm) ילך לאינסוף ככל ש n יגדל. נגדיר מה זה computational indistinguishability של זוג (סדרות) של התפלגויות כנ"ל.

הגדרה : נאמר ששני ensembles (סדרות אינסופיות) של התפלגויות

D0,n…. D0=D0,n….

D1,n…. D1=D1,1, D1,2…

כאשר D0,n , D1,n הן מעל הקב' lm){0,1} ->פונק' l:N+-> N+ הם computationally indistinguishable (בלתי ניתנים לאבחנה חישובית)

אם לכל אלגוריתם {0,1}<-\*{0,1}A שרץ בזמן פולינומי (כלומר קיים פולינום p(n) כך שלכל קלט x, A מבצע לכל היותר p(n) צעדים)

מתקיים: לכל n≤1

(n)ɛ>=|Pr(A(d)=1)-Pr(A(d)=1)|

1D-> d d<-D0

כאשר (n)ɛ היא פונ' זניחה בn.

נסמן comp.ind. ב D0cD1. (D0D1, D0sD1 ->(התפלגויות זהות))

↓

קרבה סטטיסטית (A לא מוגבל חישובית)

נותר להגדיר מהי פונ' זניחה. אינטואיטיבית זו פונקציה "ששואפת" ל0 "יותר מהר" מכל פולינום.

הגדרה 5.2 : נאמר שהפונקציהU{0} +R<-+N:(n)ɛ היא זניחה אם לכל פולינום p(n) קיים קבוע כך שלכל n≤n0 מתקיים>(n)ɛ

דוגמאות לפונקציות זניחות/לא זניחות

,,=(n)ɛ

↓

גם זניחה כי lglgn "גדול" מהקבוע, אבל בפועל היא גדלה מאד לאט.

לדוגמא =(n)ɛ אינן זניחות.

שימו לב שפונק' זניחה לא "מיד" קטנה מפולינום מסויים, אלא הn0 הרלוונטי יכול להיות גדול. לדוגמא לעומת p(n)=n10 , נדרש n>n0=.

נעבור להגדרה שך מע' הצפנה עם בטיחות חישובית.

בפרט נדבר על משפחות(ensembles) של הצפנות שיאפשרו לייצר מפתחות הולכים ומתארכים.

הגדרה 5.3:

משפחה של מע' הצפנה: נתונה ע"י שלשה של אלגוריתמים:

Gen(1n),Enc(1n,k,mjr),Dec(1n,k,mjr)

↓

פרמטר בטיחות.

האלגוריתמים יהיו פולינומים (ירוצו בזמן פולינומי בקלט שלהם)

Gen(1n)- מייצר מפתח kɛKn . (בדר"כ Kn={0,1}n אבל לא בהכרח).

Enc(1n,k,mjr)*- מייצר הצפנה c של m בהינתן קלט m*ɛMn (cɛCn).

*בניגוד להגדרה המקורית שלנו, נרשה (ואפילו נעדיף) ש Mn,Cn יהיו איסופיות(\*{0,1})*

Dec(1n,k,mjr)- כמו קודם, אלג' פענוח דטרמיניסטי.

מהו פרמטר הבטיחות – קובע את אורך המפתח(ואת הבטיחות הקונקרטית של מע' ההצפנה).

מייצרים מפתח באורך התלוי בn ומשם משתמשים בו כל הזמן. (בסופו של דבר, נרצה להשתמש בו להרבה הודעות).

כמו קודם נדרוש נכונות מושלמת. כלומר לכל n, kɛKn*, m*ɛMn*,*

*Pr[Dec(1n,k,Enc(1n,k,m,r))=m]=1*

*כעת נגדיר בטיחות חישובית של מע' הצפנה סימטרית.*

*הגדרה 5.4 (בטיחות חישובית) נאמר שמע' הצפנה סימטרית n*ɛN+ *(Genn­,Encn,Decn)(כל n מגדיר שלשה של אלג')*

*נאמר שמע' מקיימת בטיחות חישובית (במובן החלש ביותר של הודעה בודדת)*

*Comp.indistinguishability , אם הensembles הבאים הם Comp.indistinguishability:*

*עבור n נגדיר M0,M1*ɛMn ו

D0,n : Enc(1n,k,M0jr)

K<-Gen(1n,r') 🡪r,r'שנקבעת ע"י ,c0התפלגות על פני הצפנות

*r for Enc(Rn)*

*r' for Gen(R'n)*

*D1,n : m1אלא שמצפינים את ,D0,n  כמו*

נדגיש שבהגדרה לכל תוקף פולינומי A יש את כל הפונקציה (n)ɛ שלו.

נתן הגדרה חלופית שקולה לבטיחות חישובית של מע' הצפנה שתהיה יותר נוחה להרחבה. (בסגנון הגדרה 3 לבטיחות סטטיסטית)

**הגדרה 5.5** (הגדרה חלופית לבטיחות חישובית של מע' הצפנה):

תהי n(Genn,Encn,Decn) מע' הצפנה. נאמר שהמערכת מקיימת בטיחות חישובית אם לכל תוקף שרץ בזמן פולינומי (אפילו תוקף הסתברותי – אפשר להראות שזה לא עוזר)

ההסת' שלו לנצח במשחק הבא חסומה ע"י פונק' זניחה (n)ɛ (הפונקציה ɛ תלויה בתוקף)

A(1n) | Challenger(1n)

1. A מייצר זוג הודעות m0,mnϵMN ושולח לchallenger

M0,m1-----------------------------------------🡪

1. K->Gen(1n)

b->U1 (התפלגות אחידה על {0,1})

C0<-Enc(1n,k,mb)

Cb🡨-----------------------------------

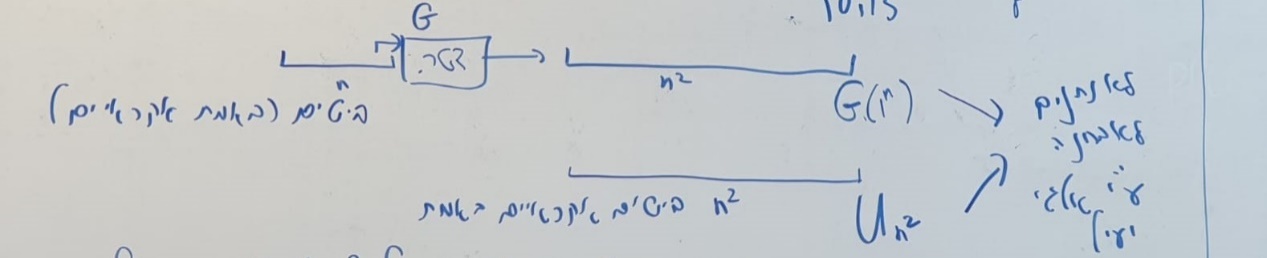
1. פולט ביט b' (מנסה לנחש את b)

A מצליח אם b'=b קורה בהסת' ≤ - זו הסתברות הנצחון שלו.

כיצד אפשר לבנות מע' הצפנה כזו?

נשתמש ב"יצור" שנקרא PRG- pseudo random generator . זהו אלגו' דטרמיניסטי שעבור קלט אקראי קצר מייצר פלט ארוך יותר ש"נראה" אקראי לכל מבחין חסום פולינומית.

לדוגמא: יכול להיות נחמד לבנות PRG G שמצפה מ n{0,1} ל n^2{0,1}



איך זה יכול לעזור בהצפנה? הרעיון הוא לקצר את המפתח ולהשתמש בו כדי להצפין הודעות ארוכות.

נזכר שOTP עובד: אם נצפין הודעות באמצעות Enc(k,m) =m xor k = c

נקבל בטיחות מושלמת.

הרעיון הוא להשתמש במקום k ב G(k) (k באורך n ביטים)

Enc'(k,m)=G(k)xor m

(k בגודל n , m בגודל n2, G(k) בגודל n2)

ונראה שG(k) אקראי לכל תוקף יעיל

נתחיל מהגדרה פורמלית של PRG .

הגדרה 5.6: גנרטור פסאודו-אקראי (PRG) הוא אלגוריתם דטרמיניסטי G:{0,1}+->{0,1}+ כך ש:

1. לכל x , l(x)=|G(x)| כאשר l:N->N היא פונקציה המקיימת: לכל n l(n)>=n+1
2. G נתן לחישוב ע"י אלג' יעיל(רץ בזמן פולינומי)
3. (פסאודו-אקראיות): לכל תוקף יעיל A , A לא מצליח להשיג יתרון אבחנה "טוב". כלומר הensembles

D0=D0,1,D0,2……..,D0,n = G(x),….

(גדול שווה nלכל ) X<-Un

D1=D1,1,D1,2……..,D01n = Ul(n),….

(נזכיר, במילים אחרות לכל n ולכל A פולינומי קיים Aɛ זניח )

(|Pr(A(d)=1)-Pr(A(G(d))=1)|≤ɛA(n)

d<-Ul(n) d<-Un (פונקציה זניחה)

הערה: ה SD בין G(Un) לUl(n) עלול להיות מאוד קרוב ל1.

לדוגמא עבור l(n)=n2. n2=>|support(G(Un))|

