# שיעור 2

המיוחד בקריפטוגרפיה מודרנית לעומת עתיקה הוא שהמודרנית מבוססת על הגדרות מתמטיות מדויקות. ניתן להוכיח בטיחות (לפעמים מניחים הנחות שלא יודעים להוכיח כמו קושי של בעיית פירוק מספרים (כלומר מניחים שהבעיה לא יכולה להיפתר בזמן פולינומי אלא מעריכי, כי זה מה שהאנושות הצליחה לעשות עד עכשיו), אבל לפחות בדיוק יודעים מה מניחים ומה מקבלים).

הבדל נוסף, כיום הקריפטוגרפיה משומשת בכל מקום, כל התעבורה באינטרנט מוצפנת, כל הודעות בין בנקים למשתמשים וכדומה. בעבר השתמשו בה רק בהקשר צבאי.

יש כמה סוגי תוקפים (מסודרים מהפשוט למתוחכם):

1. (הכי פשוט) תוקף שרואה רק את אותה ההודעה שהוא מעונין לפרוץ
2. (יותר מתוחכם) known plaintext attack – התוקף כבר ראה הודעות שלנו תחת מפתח K הנוכחי, ועכשיו הוא מעוניין לפרוץ הודעה אחרת (אם מצפינים הרבה הודעות עם אותו מפתח חייבים רמת בטיחות שיודעת להתמודד עם תוקפים מהסוג הזה)
3. (יותר חזק) chosen plaintext attack כמו הקודם, רק שהוא בוחר אילו הודעות מוצפנות הוא יראה
4. (הכי חזק) Chosen cyphertext attack לתוקף יש גישה לאלגוריתם הפיענוח והוא יכול לבקש ממנו לפענח כל הודעה שהוא רוצה, פרט להודעה שהוא מעוניין לפרוץ (ממדל מצב שעובד סורר בחברה קיבל גישה לפיענוח הודעות מוצפנות, ובנקודת זמן מסוימת איבד את הגישה והוא מעוניין להמשיך לפענח הודעות). במצב כזה הוא לא ראה את המפתח אלא רק הודעות מוצפנות ולא מוצפנות (אחרת, אם יש לו גם האלגוריתם וגם את המפתח, הקרב נגדו אבוד).

מה רמת הבטיחות שנרצה נגד התוקף, מי שיהיה?

1. נרצה שC- ההודעה המוצפנת- לא תחשוף את המפתח הסודי (כי אחרת גם m שהוצפן יתגלה וגם כל הודעה שתוצפן ע"י אותו מפתח k תיחשף). זה לא מספיק טוב, לדוגמה, אלגוריתם הצפנה שמתעלם מהמפתח ופשוט שולח C=M עומד בהגדרה זו וזה לא מספיק. (C הוא הודעה מוצפנת, M הוא ההודעה המקורית c=enc(m,k)))).
2. נרצה שC לא יחשוף מהו בדיוק m. זה עדיין לא מספיק טוב, למשל אולי C חושף את השליש האחרון, נניח, של ההודעה.
3. נדרוש שC לא תכיר אף ביט מההודעה M המקורית. זה עדיין לא טוב, כי אם הוא יכול להבין את xor של 2 ביטים (כלומר אם הם שווים או שונים) זה כבר נחשב מידע סודי חשוב שדלף.
4. C לא מאפשר ללמוד שום דבר שימושי על m – עדיין מסוכן כי אנחנו לא יודעים מה שימושי לתוקף.
5. נדרוש שהתוקף לא ילמד כלום על m דרך c- זו הרמה הנדרשת שנרצה לפרמל (להפוך לפורמלית).

**מערכת הצפנה סימטרית – הגדרה פורמלית:** מערכת הצפנה סימטרית (עם מפתח סימטרי) היא שלשה של אלגוריתמים. אלגוריתם Gen לייצור מפתחות עם תחום סופי של אקראיות (הקלט שלו הוא תחום סופי של אקראיות, והפלט הוא מפתח מתוך תחום סופי של מפתחות). אלגוריתם נוסף הוא אלגוריתם הצפנה enc הקלט יהיה מפתח הודעה מתוך קבוצה סופית של הודעות אפשריות ומחרוזת אקראית מתוך קבוצה סופית של מחרוזות אקראיות, והפלט יהיה הודעה מוצפנת (שקראנו לה C). אלגוריתם פענוח דטרמיניסטי dec הקלט הוא המפתח וההודעה המוצפנת, והפלט הוא ההודעה המקורית.

שימוש אופייני במערכת הצפנה כזו:

1. Alice ו- bob נפגשים (בהמשך נראה שיטה יותר טובה) מריצים את gen ומקבלים מפתח k ששניהם מודעים אליו.
2. בכל פעם שירצו לדבר ישתמשו בK כדי להצפין ולפענח הודעות

מה נדרוש ממערכת הצפנה כזו? נדרוש תמיד נכונות מושלמת, בלי קשר למת הבטיחות, כלומר: לכל m הודעה שיכולה להיות מוצפנת (מתוך המאגר הסופי) ההסתברות שבתהליך הבא

1. K מיוצר ע"י אלגוריתם gen(r) פעם אחת
2. מצפינים את m באמצעות פונקצית enc(k,m,r`)
3. מפענחים את c באמצעות פונקציית dec(k,c) ומקבלים m'

M יהיה שווה לm' שווה ל1.

כאן ההסתברות נוגעת לאקראיות שנכנסת כקלט לפונקציית gen ולפונקציית enc (המחרוזת הרנדומלית).

מה לגבי בטיחות? נתחיל מדרישת בטיחות מינימלית עבור הודעה מוצפנת אחת.

הבהרה: ערכים קונקרטיים מסומנות באותיות קטנות, משתנים מקריים באותיות גדולות, וקבוצות מהן דוגמים גם באותיות גדולות (לפניהם יהיה E שמסמן "שייך ל-")

הגדרה: מערכת הצפנה סימטרית מקיימת בטיחות מושלמת אם לכל זוג הודעות m1, m2 E M מתקיים שההתפלגות של ההודעה המוצפנת c היא אותה התפלגות כלומר לכל cEC

pr(C=c|M=m1)= pr(C=c|M=m2) במילים אחרות ההצפנה מתפלגת אותו דבר בין אם הגרלנו m1 או m2 כלומר היא לא עוזרת לנו להבחין בין אף זוג הודעות.

הסבר: M,C משתנים מקריים. ההסתברות שההודעה המוצפנת c (מיוצגת בC גדולה) היא הc שקיבלנו (c קטנה) מייצגת את ההצפנה של ההודעה הראשונה m1, שווה להסתברות שההודעה המוצפנת שקיבלנו היא ההצפנה ההודעה השנייה m2.

דוגמה 1: **one time pad -otp**

זו מערכת הצפנה פשוטה שתקיים את דרישת הבטיחות הזו. נקבל פרמטר n (למשל n=100) שתסמל את כמות ההודעות שנרצה להצפין.

Rgen={0,1)^n -- gen(r') ,כלומר הטווח של המחרוזות האקראיות הוא {0,1)^n וgen(r) פולט k=r'

Enc(m,k)—M={0,1}^n' כלומר מאגר ההודעות הוא שייך לקבוצה הזו. פולט C=k xor m כאשר הxor עובר ביט ביט (זה אלגוריתם ההצפנה). R(enc)={epsilon}

Dec(k,c) – C={0,1}^n האלגוריתם עושב שוב xor ומציג את m.

בקיצור- ההצפנה היא xor וגם הפענוח (כי xor מבטל xor) ופונקציית enc לא משתמשת במחרוזת רנדומלית (קבוצת המחרוזות שלה ריקה)

נראה לפני הבטיחות שהמערכת מקיימת נכונות, כנדרש:

תהי m הודעה כלשהי ו k מפתח כלשהו:

C=enc(k,m)=k xor m

Dec(k,c)=k xor c=k xor (k xor m)= k xor k xor m

החלק הראשון k xor k =0 כי כל דבר xor עם עצמו הוא אפס (זה ההגדרה של xor)

החלק השני יוצא 0 xor m שזה m. 1 xor 0 שווה 1

0 xor 0 שווה 0. הוכחנו נכונות.

כעת נוכיח בטיחות לפי הרמה שדרשנו:

[בהגדרת הבטיחות העובדה שההתפלגות של m על m1,m2 אחידה היא לא קריטית ומספיק לבחור התפלגות שבה כל אחת מההודעות m1,m2 גדולה מ0. ]

יהיו m1,m2 זוג הודעות. M1,m2 E {0,1}^n=M נראה שההתפלגות של ההודעה המוצפנת היא אותה ההתפלגות עבור m1 ועבור m. אכן:

K הוא תוצר של gen, c תוצר של enc ו M הוא מתפלג באופן אחיד על .m1,m2

**Pr(C=c|M=m1) while [gen(r')=k,C=enc(k,m) and M=U{m1,m2}]**

**כלומר, ההסתברות שההודעה המוצפנת היא של ההודעה הראשונה היא שווה ל:**

**pr(C=c)[ while k=gen(r') and c= enc(k,m1)]**

**השוויון מתקיים כי אמרנו את אותו הדבר בדרכים שונות. אמרנו שההסתברות של C בתנאי שM הוא ההודעה הראשונה שווה להסתברות שC הוא ההודעה המוצפנת כאשר לפונקציית enc נכנסה ההודעה m1.**

**המשך, וזה משתווה ל:**

**Pr(C=c) while [c=k xor m]=pr(C=c) while [kE {0,1)^m]**

**...שווה להסתברות שההצפנה היא c כאשר k הוא חלק מקבוצה {0,1}^n ששווה ל:**

**Pr(c=k xor m1) (כי זה האלגוריתם) ששווה ל:**

**Pr[k=c xor m1] (העברת אגפים בXOR) ששווה ל:**

**1/2^n.**

**כי ההסתברות שk יהיה k מסוים מתוך קבוצה של 2^n אפשרויות (כי k שייך לקבוצה {0,1}^n ) הוא 1 חלקי גודל הקבוצה).**

**באופן דומה ההסתברות שההודעה המוצפנת היא m2 גם שווה ל1/2^n.**

הערה: בהגדרת הבטיחות (בפרט בotp) חשוב לא להחזיק את מע' ההצפנה עצמה בסוד אלא רק את המפתח המיוצר בכל פעם. אנחנו רוצים (!) שמערכת ההצפנה תהיה גלויה כדי שנוכל למצוא בה פרצות באמצעות אחרים.

שאלה: מה קורה אם מצפינים מספר הודעות עם אותו מפתח עם otp ? (עוד לא הגדרנו בטיחות עבור מספר הודעות, בניסוח לא פורמלי הכוונה היא שעבור 2 ווקטורים של הצפנות c1,c2

c'1,c'2, לא נוכל להבחין בין כל זוג שונה של ווקטורים).

השאלה היא האם 2 הצפנות שונות של 2 זוגות הודעות,הצפנה ראשונה M1 של הזוג m1,m1

הצפנה שנייה M2 של הזוג m1,m2 הם בעלי אותה ההתפלגות? תשובה: לא. ההצפנה הראשונה M1 יוצרת בעצם וקטור עם 2 מחרוזות זהות והשנייה עם 2 מחרוזות שונות (וההסתברויות של 2 אלה שונות). מבחינה אינטואיטיבית- דרך 2 הודעות אלה התוקף כן למד משהו, ששתי הההודעות הראשונות זהות והבאות אחריהן שונות).

נשים לב שנרצה בטיחות אפילו בשביל 2 הודעות, אנחנו נהיה מוכרחים להשתמש בקראיות. זה לא נכון רק בotp אלא באופן כללי.