



מרצה: ד"ר קלים יפרמנקו

סמסטר סתיו תשפייא

10/01/2021

:22 הרצאה

Local codes

<u>: מבוא</u>

קודים מקומיים, הם קודים לתיקון שגיאות המצוידים באלגוריתמים מקומיים. קודים אלה הם בעלי היסטוריה ארוכה של אינטראקציה עם תיאוריית הסיבוכיות, עם יישומים בולטים להוכחות אינטראקטיביות, הוכחות ניתנות לבדיקה באופן הסתברותי (PCP) , hardness amplification , (PCP) אחזור מידע פרטי(private information retrieval) ותאוריית הסיבוכיות האלגברית .

מה היה לנו עד עכשיו!

 ${
m C}$: כך ש-C הוא קוד לתיקון שגיאות, והתרחיש שדיברנו עליו הוא כזה C כך ב- $C: \{0,1\}^k o \{0,1\}^n$

אליס שולחת מכן היא שולחת את הקידוד את האודעה (m) אליס שולחת לבוב, אז ראשית היא מקודדת את מקודדת את לבוב. אליס שולחת מגיע אליו בתוספת רעש c(m)+e. ואז בוב מפענח את ההודעה שקיבל.

היום נדבר על המצב שבו בוב לא מעוניין לקרוא את כל ההודעה שקיבל מאליס כי הוא מתעניין רק בביט אחד מתוך ההודעה שהוא מקבל מאליס, למשל אליס היא המרצה והיא מפרסמת ציונים, אז בוב מתעניין רק בציון שלו ואין לו עניין לדעת את כל הציונים. קיימים הרבה מקרים כאלו, שבהם מעניין אותנו רק דברים לוקליים.

איך עושים את זה!

נציג כמה סוגים של קודים לוקליים:

ור מקומי: locally recoverable codes .1

קוד על אלפבית סופי נקרא בר שחזור (קוד LRC) אם כל ביט בקידוד הוא פונקציה של מספר קטן (לכל היותר קוד על אלפבית סופי נקרא בר שחזור (קוד LRC) אם כל ביט בקידוד הוא פונקציה של מילת הקוד. באופן רשמי יותר, (n, k, r) קוד הניתן לשחזור מקומי (LRC). המשמעות שהקוד מפיק n ביטים מהמילה מתוך הודעה באורך k, ולכל ביט ממילת הקוד, קיימים לכל היותר r ביטים אחרים כך שניתו לשחזר מהם את ערך הביט הנייל.

אנחנו אומרים שלקוד יש מקום/סביבה (r, d) אם לקוד יש מרחק לביט בהודעה (במילת קוד) יכול היות משוחזר מ- r ביטים אחרים.

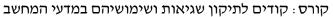
הערה: שפמא local recoverability - הערה: חלשים לשחזור מקומי לשחזור מקומי הניתנים לשחזור מקומי חלשים וחזקים ביטים אורים מניתנים לשחזור מקומי על ידי r ביטים אחרים ממילת קוד. ב- strong local recoverability עבור מההודעה ניתנים לשחזור מקומי על ידי r ביטים אחרים ממילת קוד.

: locally recoverable codes-שימושים ב

דוגמה מספר 1:

חברה גדולה מחזיקה cloud ענק והמון דיסקים שבהם נשמר מידע רב כלשהו. עם הזמן חלק מהדיסקים מתקלקלים או נשרפים כי יש בלאי. כלומר יש בעיה מקומית. נרצה להביא למצב שאם דיסק אחד התקלקל, עדין נוכל לקרוא כמה דיסקים אחרים, (כלומר קצת מידע, ולא את כל הדיסקים) ובאמצעותם לשחזר את מה שהיה כתוב בדיסק שהתקלקל. מכיוון שזו בעיה שצריך לפתור בתדירות גבוהה, <u>צריך לעשות את זה מהר</u>.

איך נעשה את זה? ראשית, נניח שכל דיסק ניתן לשחזר מ-r דיסקים אחרים, ולכן פשוט נאסוף את המידע ואחרי כל r דיסקים שמכילים מידע נוסיף דיסק r+1 שישמור את תוצאת ה-XOR של r הדיסקים הקודמים (סוג של קוד Parity שלמדנו).





מרצה: דייר קלים יפרמנקו

סמסטר סתיו תשפייא

10/01/2021

כך, במידה ודיסק אחד נהרס, נצטרך לקרוא את r הדיסקים האלה ונחשב את ה-XOR שלהם וכך נקבל בדיוק את הדיסק הזה שנהרס לנו ונצליח לשחזר אותו. כלומר, קוד זה יודע להתמודד עם נתונים שנפגעו ולשחזר אותם כהלכה.

הערה: באופן עקרוני ניתן לפתור את זה עם הקוד שלמדנו של ההעתקה, וכך אפילו נצטרך לקרוא רק דיסק אחד(ולא r דיסקים), אך הבעיה שהאחסון של זה יקר מאוד ולכן, בפועל לא נשתמש בזה כי זה לא מומלץ.

<u>: 2 דוגמה מספר</u>

מתקפת סייבר על חברה גדולה אשר מחזיקה cloud ענק והמון דיסקים שבהם נשמר מידע רב כלשהו. המתקפה מביאה למצב שכמות רבה של דיסקים התקלקלו בבת אחת (ולא אחד) . מכיוון שמקרה זה הוא חריג ואינו קורה בתדירות גבוהה, פה לא נתעקש על כך שזה יהיה מהיר, אך עדין לא נרצה לקרוא את כל המידע (המידע הוא רב וזה לא ריאלי). אז למעשה, לשחזר את המידע במצב הזה אנחנו כבר יודעים לעשות : ניקח את המידע ואותו נקודד עם הקוד לתיקון שגיאות ולאחר מכן נבדוק מה המרחק של הקוד הזה. אם הוא d אז נוכל לשחזר כל עוד מספר הדיסקים שהתקלקלו לנו הוא לכל היותר d-1. אם נעשה את זה עם קוד Reed- Solomon אז נצטרך לקרוא את כל המקומות.

-locally testable codes .2 קודים הניתנים לבדיקה מקומית:

קוד לבדיקה מקומית הוא סוג של קוד לתיקון שגיאות שעבורו ניתן לקבוע אם מחרוזת היא אכן מילת קוד על ידי קריאת מספר קטן (קבוע לרוב) של ביטים מהמחרוזת (ביטים אלו יבחרו באופן אקראי, וכך ייהיריביי לא יוכל לצפות באילו ביטים נבחר). כלומר, הכוונה ב-"בדיקה מקומית" היא שבאמצעות דגימה של מספר קטן מאוד (סופי) של ביטים ממילת הקוד ניתן יהיה לזהות בהסתברות טובה האם היא חוקית או לא. במצבים מסוימים, נרצה לדעת אם הנתונים פגומים מבלי לפענח את כל המידע, כך שנוכל לבצע פעולות מתאימות בתגובה. ולכן נרצה לדעת להפריד בין שתי אופציות(עיימ להבחין מה נכון ומה לא): a זאת מילה קוד נכונה וליתר דיוק - אם היא נכונה, תמיד נזהה זאת ואם היא לא נכונה, אז ככל שהיא רחוקה ממילת קוד נכונה. ליתר דיוק - אם היא נכונה, תמיד נזהה זאת תגדל משמעותית (מובן מאליו שהיא רחוקה מלהיות מילת קוד חוקית/נכונה ההסתברות שלנו לזהות זאת תגדל משמעותית (מובן מאליו שאם נפלה שגיאה בביט בודד במילת הקוד אין סיכוי לזהות זאת בהסתברות גדולה על ידי בדיקה של מספר סופי של ביטים - הרי חייבים לקלוע בדיוק לביט שהתקלקל כדי שיהיה בכלל סיכוי להבין שמשהו השתבש).

: locally testable codes-שימושים ב

משתמשים בהם בעיקר שרוצים לבנות מטבעות וירטואליים או שרוצים לבנות הוכחות, נרצה להיות מסוגלים לבדוק מהר אם מה שמוכיחים לנו זאת טענה נכונה או שמנסים לרמות אותנו.

:1 דוגמה מספר

מטבעות וירטואליים – אדם כלשהו נותן הוכחה וטוען שהמטבע הזה הוא המטבע שלו. בשביל לבדוק את החוכחה הזאת, לא נרצה לקרוא את ה-log של כל הטרנס אקסונים של כל המטבעות, אלא יספיק לנו לקרוא log של כמה כאלה וככה לדעת האם ההוכחה הזאת נכונה או לא.

: 2 דוגמה מספר

: הרעיון של המשפט probabilistically checkable proof (PCP) משפט

רוצים להוכיח טענה כלשהי, ההוכחה לטענה היא ארוכה מאוד ובנוסף, ניתן להציג אותה כשרשרת של גרירות. הבעיה בסוג הוכחה הנייל שמספיק שמישהו ייירמהיי רק במקום אחד בהוכחה וזה יגרום לכך שכל החוכחה תשתבש ותהפוך להיות לא נכונה. משפט PCP מיועד לסוג הוכחות כאלה והוא אומר שבשביל שההוכחה תהיה לא נכונה מישהו חייב יילרמותיי בהרבה מקומות בהוכחה, כך נוכל לבדוק מהר מאוד את ההוכחה. איך! נקבל הוכחה, ואז כדי לבדוק אותה נבחר כמה מקומות ונקרא את המידע במקומות הללו, אם ההוכחה בטוח לא נכונה.

מרצה: דייר קלים יפרמנקו

סמסטר סתיו תשפייא

10/01/2021

-locally decodable codes .3

קוד לפענוח מקומי (LDC) הוא קוד לתיקון שגיאות המאפשר פענוח של ביט בודד מהחודעה המקורית בסבירות גבוהה על ידי בחינה (או שאילתה) של מספר קטן של ביטים של מילת קוד שאולי מקולקלת. במילים אחרות, קודים אלו משתמשים במספר קטן של ביטים של מילת הקוד כדי לשחזר את המידע המקורי באופן הסתברותי. התדירות של השגיאות קובעת את הסבירות שהמפענח ישחזר נכון את הביט המקורי.

מאפיין זה יכול להיות שימושי, למשל, בהקשר שבו מידע מועבר בערוץ רועש, ורק תת-קבוצה קטנה של הנתונים נדרשת בזמן מסוים ואין צורך לפענח את ההודעה כולה בבת אחת.

אליס שולחת הודעה לבוב, ואת בוב מעניין רק ביט בודד מההודעה, הביט ה- i, ולכן הוא לא מעוניין לקרוא את ההודעה כולה. נציין שבקודים רגילים, בוב יצטרך לקחת את כל ההודעה, לפענח אותה ורק אז יוכל לשחזר את הביט ה-i, אך עם קודים מקומיים אנחנו מעוניינים שבוב יקרא מספר קטן של ביטים ויהיה מסוגל לשחזר מתוכם את הביט ה-i.

שימו לב: עלולה להיות בעיה השיטה הזאת- אם בוב קורא למשל 3 ביטים, שאותם קבע מראש ולא בחר אותם באופן אקראי, ויש לנו רעש אדברסרילי. אז שלושת הביטים האלה בדיוק יכולים להיות משובשים, בגלל שהיה ידוע באילו ביטים הוא יבחר, ולכן המטרה היא לבחור את הביטים שנקרא באופן אקראי, ככה שגם אם יהי לנו רעש כזה, בסיכוי גבוהה נוכל לשחזר את הביט ה-i.

הערה: קודים לפענוח מקומי אינם תת קבוצה של קודים הניתנים לבדיקה מקומית, אם כי קיימת חפיפה מסוימת בין השניים.

: locally decodable codes-שימושים ב

שימוש רחב בבניית פרוטוקולים קריפטוגרפים, המאפשרים לנו לשמור על הפרטיות שלנו.

הערה: בהמשך השיעור נחזור לעסוק בקודים מסוג זה.

4. self correctable codes-קודים הניתנים לתיקון מקומי:

דיברנו על כך, שכאשר נרצה לשחזר איזשהו ביט מהר אז אנחנו מתמודדים עם זה שרק דיסק אחד נהרס. כעת נעסוק במצב בו למשל מישהו אדברסילי, שיבש לנו כמות קבועה של דיסקים, והביא להרס של כל הרצף הזה של r הדיסקים ובשאר לא נגע, אז כבר לא נוכל לשחזר. אך עדין נרצה במקרה כזה לבחור לקרוא כמה מוקמות באופן אקראי ומתוכם לשחזר את המידע.

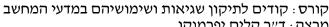
קודים אלו, עובדים בצורה דומה ל-locally decodable codes , רק שבמקרה זה תחשבו שיש חלק מסוים של הודעות שהן משובשות ומעניין אותנו לשחזר ביט אחד מהקוד. כלומר, ההבדל ביניהם הוא שב- (LDC) אנחנו רוצים לשחזר אחד מ ${\bf n}$ ביטים של מילת הקוד, באופן רוצים לשחזר את אחד מתוך ${\bf k}$ הביטים של ההודעה ופה נרצה לשחזר אחד מ- ${\bf n}$ ביטים של מילת הקוד, באופן לוקלי ובצורה מהירה.

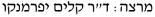
הערה: סוג זה של קוד נחשב למושג חזק יותר מ- (LDC), כי כשיש לנו קוד לינארי בדרך הזאת נוכל תמיד לשמור את ההודעה כחלק מהקוד, בביטים הראשונים שלו. ואז אם נרצה לשחזר ביט כלשהו פשוט נעשה self correction לחלק הזה של ההודעה (להתחלה של הקוד).

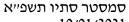
: locally decodable codes נתעמק כעת ב-

<u>: הגדרה</u>

$$(q, \delta, \varepsilon) - \text{LDC}$$
 : וזה נקרא. $P_r(D_i(w) = m_i) = 1 - \varepsilon$







10/01/2021

הוא הפרמטר החשוב, וערכו הוא כמה q שאילתה הכוונה, איזה מקום במילת קוד הוא מבקש. qשאילתות הקוד הזה עושה לתוך מילת הקוד המשובשת.

. שימו המילה המילה של i-ו את הביט לנו את המילה המקודדת $D_i(w)$ אמור לעו

: הסבר

 i_1,i_2,\ldots,i_q : אינדקסים אינדקסים, עבל הוא יכול לבחור איזשהם אינדקסים, אלגוריתם הפענוח אינדקסים, אלגוריתם הפענוח אינדקסים וכך לקבל את כל לקרוא את כל מילת אלגוריתם הפענוח אלגוריתם $w[i_1], w[i_2], \ldots, w[i_a]$ וכך לקבל את הוא קורא איזשהם q מקומות מתוך המילה(לא בהכרח ברצף) ובעזרתם הוא מצליח לשחזר ביט אחד מסוים מההודעה שמעניין אותנו.

שימו לב: ביטים ממילת קוד הם אינם ביטים מהודעה.

מושג: אלגוריתם פענוח D_i נקרא חלק אם כאשר מסתכלים על כל שאילתה שלו i_i אז היא מתפלגת באופן אחיד (כלומר, היא יכולה לקבל את אחד הערכים מ-1 עד n באותו סיכוי).

, אך אם אלגוריתם פענוח אז הוא מפענח נכון מילות לא משבשות (זה תמיד). אך אם אלגוריתם פענוח חלק, ארכון מילות לא משבשות אז הוא מפענח נכון מילות לא משבשות או הוא מפענח פענוח אז הוא מפענח נכון מילות לא משבשות או הוא מפענח פענוח אז הוא מפענח נכון מילות לא משבשות או הוא מפענח פענוח אז הוא מפענח נכון מילות לא משבשות או הוא מפענח פענוח אז הוא מפענח נכון מילות לא משבשות או הוא מפענח פענוח אז הוא מפענח נכון מילות לא משבשות או הוא מפענח פענוח אז הוא מפענח מילות מילות משבשות או הוא מפענח מילות מילות משבשות מעוד מילות מפענח מילות מפענח מילות מילות משבשות משבשות מילות משבשות מילות מפענח מילות מילות מפענח מילות מילות משבשות מילות מ . $P_r(D_i(w) \neq m_i) \leq q \cdot \delta$: ונניח שקיימות $\delta \cdot n$ שגיאות אז

 i_i מקומות, שכל אחד מהם מתפלג באופן אחיד. נרצה לדעת מה הסיכוי ש ${
m q}$ שנבחר, γ נופליי במקום לא נכון, כלומר משובשn נזכיר שיש $\delta \cdot n$ מקומות משובשים מתוך מקומות ולכן , ומכאן נובע שהסיכוי ייליפוליי על מקום אחד כזה הוא הוא אומכאן ומכאן נובע הייליפוליי על מקום אחד כזה הוא החיכוי $\frac{\delta \cdot n}{n} = \delta$ היותר $q \cdot \delta$. בנוסף, אם כל המוקמות לא משובשים אז פשוט נחזיר תשובה נכונה, כי הנחנו שכאשר האלגוריתם הפענוח שלנו קורא מילה נכונה/לא משובשת אז הוא מחזיר תשובה נכונה. ולכן:

:בשביל לבנות קוד פיענוח מקומי צריך לבנות אלגוריתם פענוח D_i לכל כך ש

- 1. השאילתות מתפלגות בצורה אחידה
- מחזיר תשובה נכונה על מילת קוד לא משובשת D_i .2

למעשה, ראינו כבר קוד כזה ולא הוכחנו שהוא כזה, מדובר בקוד Hadamard.

תוכורת: קוד Hadamard הוא קוד לתיקון שגיאות על שם Hadamard המשמש לזיהוי ותיקון שגיאות בעת העברת הודעות בערוצים רועשים מאוד או לא אמינים. קוד זה הוא דוגמה לקוד לינארי באורך מעל האייב הבינארי. 2^m

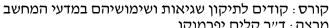
קוד למילת באורך \mathbf{k} למילת באורך אוות הממפה מקומי, הממפה לקוד פשוט הניתן לפענוח אוות Hadamard קוד באורך $C: \{0,1\}^k \to \{0,1\}^{2^k}$: איך קורה המיפויי

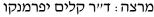
 \cdot מקבלים הודעה f m ושולחים את כל הפונקציונליים הלינאריים של ההודעה הנ"ל

$$m \to (\langle m, x \rangle)_{x \in \{0,1\}^k}$$

 $(2, \delta, 2 \cdot \delta)$ – LDC טענה: קוד Hadamard טענה

, ולכל ביט ממילת הקוד, הודעה באורך $k=\delta$ ולכל ביט ממילת הקוד, משמעות הטענה: הקוד מפיק n=2. ביטים אחרים ערך מהם את ערך לשחזר מחרים ביטים $r=2\cdot\delta$ ביטים לכל היותר







סמסטר סתיו תשפייא 10/01/2021

 m_i ובאמצעותם לחשב את (m,y) ואת (m,x) ולקבל את (x,y) ובאמצעותם לחשב את במילים אחרות, הוא יכול לבחור m_i נבחר כדי לקבל את x ,y אז איזה x

היינו צריכים לבחור (שתי השאילתות) x (שתי השאילתות) במקרה בו לא היה חשוב לנו ש- x (שתי השאילתות) במקרה בו לא היה חשוב לנו שאילתה אחת ,אשר ממנה נקבל $(m,x) = \sum m_i \cdot x_i = \langle m,x \rangle$ והיינו רוצים למצוא את שהי ממנה נקבל m_i יש ונוכל לדעת מהו , v את ונוכל לדעת מהו יש i-

 \cdot יהיה: איריתם באופן אחיד ולכן, אלגוריתם הפענוח (אשר מפענח את הביט הi יתפלגו באופן אחיד ולכן, אלגוריתם הפענוח

- עבחר היא הביט ה-i ולבקש את הערך היא הכוונה היא הכוונה $y=x+e_i$ את הערד את נבחר באופן ${f x}$ הקוד במקומות האלה..
- : כלומר אנחנו מקבלים את הסכום את מילות הקוד $\langle m, x + e_i \rangle$, $\langle m, x \rangle$: את מילות מילות מילות מילות הקוד נוסיף i-נוסיף במקום הסכום אותו הסכום $\sum m_i \cdot x_i + m_i = \langle m, x + e_i \rangle$ ובנוסף ובנוסף $\sum m_i \cdot x_i = \langle m, x \rangle$ w[x]-w[y] את נמצא את שקיבלנו ונמצא המשוואות נחסיר בין שתי החסיר החסיר m_i את הקבל את וכעת, כדי לקבל את

. באופן אחיד $\{0,1\}^k$ - אנחנו עדין לא נדע כלום על ה- e_i . כי vיכול לקבל כל ערך ב-v אנחנו עדין לא נדע כלום אויד.

 $1-2\cdot\delta$ מקומות משובשים אז הסיכוי ש-x ,y-שימו משובשים מקומות משובשים א מקומות משובשים אז מסיכוי ש-x ,y-שימו

w[x] את אואל את . $y=x+e_i$ -ו הפענום אלגוריתם באופן בוחר באופן בוחר באופן אלגוריתם הפענוח: אלגוריתם הפענוח: w[x] - w[y] ומחזיר: w[y] ואת

ההוכחה לטענה שהקוד הזה הוא קוד פיענוח לוקלי היא טריוויאלית מכיוון ש x ,y שניהם מתפלגים באופן אחיד ואם לוקחים מילה אקראית לחלוטין והפוכים ביט אחד בה אז עדין מקבלים משהו אקראי לחלוטין וב-[*] הוכחנו שזה מחזיר דבר נכון.

. ארוך מדיי. בעיות המרכזיות בבנייה כזאת, היא שאורך הקוד הוא 2^k (אקספוננציאלי) כלומר ארוך מדיי.

: private information retrieval - LDC-דוגמה נוספת ל

.data base- ויש לו איזשהו מספר i ובנוסף ישנם כמה שרתים שלכולם יש גישה לu user יש משתמש, נעסוק כעת בבעיה בה המשתמש שלנו לא רוצה שידעו מהו ה-i הזה שהוא מחזיק, שהוא מידע פרטי שלו. המטרה שלו היא לגשת בדרך כלשהי ל-DB ולקבל משם את ה-i הזה, מבלי לגלות את ה-iהזה לאף אחד מהשרתים. נצא מנקודת הנחה שהשרתים שלנו לא מתקשרים אחד עם השני והמטרה שלנו היא לעשות את זה עם כמה שפחות חיבורים.

. i מבלי לגלות את D[i] מבלי לגלות את במילים אחרות, אנחנו רוצים

הפתרון הטריוויאלי זה לבקש את כל ה-DB ולסנן את המקום ה-i. אבל אנחנו יכולים להשתמש גם ב-וסכבווy decodable code כדי לשחזר את המקום ה-i הזה. איך נעשה זאתי

אז הוא \mathbf{x} , אז הוא \mathbf{x} ולמקום ה- \mathbf{y} , אלגוריתם הפענוח D_i , אלגוריתם שלו ניגש למקום ה- \mathbf{x} יכול לבקש את לשלוח את c(D[y]) ומ- s_2 לשלוח את לשלוח את לשלוח את לשלוח את יכול לבקש את לשלוח את

נזכיר ש-y, א מתפלגים באופן אחיד, כלומר כל אחד מהשרתים מקבל אוסף של ביטים אקראיים ומתוכם הוא יכול להסיק c(D[y]) ו-c(D[x]) יכול להסיק אני הביטים מתוך שני המשתמש מתוך אבל . x ,y-יכול c(D[x]) - c(D[y]): מהו הD[i] עייי החישוב





מרצה: ד"ר קלים יפרמנקו

סמסטר סתיו תשפייא

10/01/2021

היינו x אינו רוצים לשלוח את אחד הייתה בעיה, כי בשביל לשלוח את אחינו צריכים הערה: אם היינו רוצים להשתמש פה בקוד Hadamard ביטים ואז למעשה לא עשינו פה כלום, כי זה בעצם כמו שאחד השרתים ישלח לנו את אותם k ביטים מתוך ה-DB ואנחנו פשוט נשחזר אותו מתוכם.

locally decodable code כעת נראה בנייה של קודים לתיקון שגיאות שהם הרבה יותר קצרים. יש בנייה של קודים לתיקון שגיאות שהם הרבה יותר קצרים. כד ש $n=2^{2^{\sqrt{\log k}}}$: כד ש

אם נבנה איזשהו קוד locally decodable אם נבנה איזשהו לאטר (אשר locally decodable אם נבנה איזשהו קוד אומר שיכולנו לבנות וסכמווען לאטר וועך פיטים וכך private information retrieval כך שהמשתמש היה שולח ($\log n$) ביטים, ומקבל בחזרה private information retrieval מצליח לשחזר את ה-i .

נצטרך, i נצטרן, i בכל אופן חייב לשלוח אפילו במקרה שלא ביטים כי אפילו ביטים לפיטים לשלוח אכפת למשתמש וו $\log n$ ביטים כדי לשלוח או וו $\log n$

. אז בקצרה איך הבניה עובדתי . $\log k \ll 2^{\sqrt{\log k}} \ll k$ פה אנחנו עושים בבנייה משהו באמצע, כי

- S matching vectors .1
- קוד locally decodable לבנות S- matching vectors איך בעזרת ה-S-
 - 3. אלגוריתם פענוח

: אם מתקיים אם $s\subseteq\{m\}$ תואם עבור $\{u_i\}_{i=1}^k\subseteq Z_m^h$ אם מתקיים אבדרה קבוצה קבוצה

- $\sum_{i=1}^m u_i[j]^2\%m$: הכוונה $\langle u_i, u_i \rangle = 0$: i לכל
 - $\langle u_i, u_i \rangle \in S$.2
 - $0 \notin S$.3

 $S=\{1,3,4\}$ נניח, |S|=3 ו- m=6 עבור m=6 כך ש $S=\{1,3,4\}$ נניח, S- matching vectors למה: קיימים

$$h \le 2^{\tilde{O}(\sqrt{\log n})}$$

 \cdot שהוא מספר ראשוני אז אפשר להוכיח שהוא הערה: הלמה הזאת מפתיעה כי אם היינו בוחרים לכל

(זאת Grolmusz פעם לאורך מישורים, ואי ש-m הוא מכפלה של שני מישורים, ואי ארוכה האמינו ש-m הבנייה שלו) למעשה בנה את ה-S – matching vectors, ככה שאנחנו מקבלים משהו הרבה יותר טוב מפולונימלי ב-n.

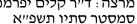
בנייה של הקוד:

איך הבנייה הזאת תעבודי

יעבודי ואיך הוא יעבודי $C: F^k o F^{m^h}:$ הקוד שלנו

$$C(m_1,\ldots,m_k) = \left(\sum m_i \gamma^{\langle u_i,x\rangle}\right)_{x\in Z_6^h}$$

 ${\bf x}$ הסבר: יש m^h קורדינאטות. לכל קורדינאטה נתאים איזשהו מקום ב- Z_m^h כלומר עבור כל קורדינאטה m^h נחשב את הסכום: $m_i \gamma^{\langle u_i, x \rangle}$.





10/01/2021

<u>שאלה- מה האורך של הקוד כפונקציה של k י</u>

נשים לב שהאורך של הקוד שלנו הוא $h \leq 2^{\tilde{O}(\sqrt{\log n})}$ - נשים לב שהאורך של הקוד שלנו הוא $h \leq 2^{\tilde{O}(\sqrt{\log n})}$: בנוסף, אנחנו יודעים לפי הלמה ש $h \leq 2^{\tilde{O}(\sqrt{\log n})}$: עכשיו נסתכל על פולינום מדרגה $h \leq 6^{2^{O(\sqrt{\log n})}}$:

$$p(x) = \prod_{i \in S} (x - \gamma^i) = (x - \gamma^1) \cdot (x - \gamma^3) \cdot (x - \gamma^4) = \sum_{i=0}^3 a_i \cdot x^i$$

 $p(\gamma^i) = 0$: אז $i \in S$ נשים לב:

אלגוריתם פענוח במקרה שלנו:

- $x \in Z_6^h$: בחר את x באופן אקראי 1
- x%6, $(x+u_i)\%6$, $(x+2\cdot u_i)\%6$, $(x+3\cdot u_i)\%6$: עכשיו תדגום 4 מקומות .2
- $c_i = a_0 \cdot w[x] + a_1 \cdot w[x + u_i] + a_2 \cdot w[x + 2 \cdot u_i] + a_3 \cdot w[x + 3 \cdot u_i]$.3
 - $m_i = c_i \cdot \gamma^{-\langle u_i, x \rangle} \cdot p(1)^{-1}$: תחזיר את

שימו לב: מכיוון שאנחנו בוחרים את איקס באופן אקראי אז גם כל אחד מהמקומות שבחרנו:

. מתפלגות באופן אחיד, מתפלגות מתפלגות אחיד, מתפלג באופן אחיד, אופן אחיד, אופן אופן אופן אופן אופן אופן אופן אחיד, אופן אופן אופן אופן אופן אחיד. אופן אחיד, אופן אחיד

נשאר להוכיח: שאם במקום W נציב את מילת הקוד הנכונה אז נקבל את התוצאה הנכונה, כלומר אם במקום C_i נציב את מילת הקוד מה יהיה יהיה C_i

$$c_i =$$

$$\begin{split} &a_0 \cdot \sum_{j=0}^k m_j \cdot \gamma^{\langle u_j, x \rangle} + a_1 \cdot \sum_{j=0}^k m_j \cdot \gamma^{\langle u_j, x + u_i \rangle} + a_2 \cdot \sum_{j=0}^k m_j \cdot \gamma^{\langle u_j, x + 2 \cdot u_i \rangle} + a_3 \cdot \sum_{j=0}^k m_j \cdot \gamma^{\langle u_j, x + 3 \cdot u_i \rangle} \\ &= \sum_{j=0}^k m_j \cdot \left(a_0 \cdot \gamma^{\langle u_j, x \rangle} + a_1 \cdot \gamma^{\langle u_j, x + u_i \rangle} + a_2 \cdot \gamma^{\langle u_j, x + 2 \cdot u_i \rangle} + a_3 \cdot \gamma^{\langle u_j, x + 3 \cdot u_i \rangle} \right) = \\ &= \sum_{j=0}^k m_j \cdot \gamma^{\langle u_j, x \rangle} \left(a_0 + a_1 \cdot \left(\gamma^{\langle u_j, u_i \rangle} \right)^1 + a_2 \cdot \left(\gamma^{\langle u_j, u_i \rangle} \right)^2 + a_3 \cdot \left(\gamma^{\langle u_j, u_i \rangle} \right)^3 \right) = \\ &= * \sum_{j=0}^k m_j \cdot \gamma^{\langle u_j, x \rangle} \left(p(\gamma^{\langle u_j, u_i \rangle}) \right) = * * m_i \cdot \gamma^{\langle u_i, x \rangle} \left(p(\gamma^{\langle u_i, u_i \rangle}) \right) \end{split}$$

 $i\neq j$ אז בעצם באשר אז $p(\gamma^i)=0$ אז אז ישלכל שבחרנו שבחרנו שבחרנו שבחרנו אז לפולינום שבחרנו שלכל אז ישלכל אז ישלכל לענות לתכונה שיש

שווה לאפס ולכן נשארים $p(\gamma^{\langle u_j,u_i \rangle})$ הביטוי הזה הכי לכל $i \neq j$ שווה אפס ולכן נשארים ** ולכן i = j