הרצאה 13 במבוא לקריפטוגרפיה מודרנית: וידוא מהיר של חישובים ארוכים

6 בינואר 2014

מוטיבציה

- ציור.
- משתמש מעוניין בחישוב פונ' $f(x)$, אך אין לא את הזמן הדרוש לכך (למשל, סמרטפון) .
- מעוניין להעזר בשרת בעל משאבים לחישוב $f(x)$ (למשל אמזון).
- כיום בעולם: המשתמש פשוט שולח את x (ותשלום כלשהו) וסומך על השרת שיחזיר את התשובה $y = f(x)$ הנכונה.
- בעולם טוב יותר: השרת "יוכיח" למתשמש כי כי חישב את התשובה הנכונה y.
- רווח נדרש: הזמן הנחוץ למשתמש לוידוא ההוכחה קצר משמעותית מהזמן הנחוץ לחישוב $f(x)$ בעצמו.

הגדרה

הגדרה 0.1 תהי $F : \{0,1\}^n \to \{0,1\}$ ניתנת לחישוב בזמן $T_F = poly(n)$. מערכת הוכחה (אינטרקטיבית או לא) עם וידוא מהיר עבור F כוללת מוודא הסתברותי V ומוכיח P. P ו־V מריצים פרוטוקול $(P,V)(x,y)$ שמטרתו להוכיח ל־V, כי $y = F(x)$ ומקיים:

- completeness: בהנתן קלט "חוקי" $(x, F(x))$, V תמיד מקבל את ההוכחה.
- soundness: בהנתן קלט "לא חוקי" (x, y^*), כך ש $y^* \neq F(x)$, V דוחה בהס' 99%.
- fast verification: זמן הריצה של V: $T_F \gg T_V$, למשל $T_V = n \cdot polylog(T_F)$.

- <u>דרישות אפשריות נוספות</u>
  - פרטיות של קלט המשתמש x.
  - הוכחה לא־אינטרקטיבית.
  - יעילות גם של המוכיח P, אידאלית $T_P \approx T_F$, למשל $T_P = T_F \cdot polylog(T_F)$.
- סיטואציה כללית יותר: גם לשרת יש קלט w העשוי להיות ארוך כ־T_F - מאפשר להוכיח שייכות לשפת NP עם וידוא מהיר.

כיצד בונים מערכת לוידוא מהיר?

- ראינו בניה מ־FHE:
  - preprocessing: V דוגם מפתח sk ל־FHE , והצפנה של $c_0 \leftarrow E_{sk}(0^n)$ ומחשב $\hat{c}_0 = Eval(F, c_0)$.
  - online: V מצפין את הקלט שלו $c_x \leftarrow E_{sk}(x)$ ושולח ל־P את c_0, c_x בסדר אקראי. P מקבל c, c' ומחשב $\hat{c} = Eval(F, c)$, $\hat{c}' = Eval(F, c')$
  - אפשר למחזר את c_0 ע"י שימוש בשכבה נוספת של FHE.
- מגבלות:
  - preprocessing ארוך כמו החישוב כולו.
  - תשובת V חייבת להשאר סודית - לאחר למידת $|c_0|$ תגובות, ניתן לשחזר את c_0! (חשבו, כיצד). חישוב c_0 חדש בכל פעם יקר מדי.
  - בנוסף, אינו תומך בסיטואציה בה למוכיח יש קלט (לא מאפשר להוכיח שייכות בשפת NP).

מתכון עתיק יומין לוידוא מהיר

- מצרך עיקרי: Probabilistically-Checkable Proofs
- אחד ההישגים המדהימים של מדעי המחשב.
- סיפור מעניין

courses.cs.washington.edu/courses/cse533/05au/pcp-history.pdf

- מתחיל בסוף שנות ה־80 עם נסיון לקבל יותר מושג חזק יותר של ZK IPs.
- מודל ה־ MIP. "יותר קל לחקור שני חשודים מחשוד יחיד".
- מסתבר: $MIP = NEXP$, יותר מכך מספיק לשאול כל אחד מהמוכיחים שאלה אקראית אחת.
- הוביל למודל ה־PCP.

הגדרה 0.2 (מקרה מיוחד) מערכת PCP עבור פונ' $F : \{0,1\}^n \to \{0,1\}$ כוללת מוכיח $\bar{P}$ ומוודא הסתברותי $\bar{V}$. בהנתן קלט (x,y), P מייצר הוכחה (מחרוזת) $\pi \in \{0,1\}^\ell$ לכך ש־$y = F(x)$. המוודא $\bar{V}^\pi(x,y)$ דוגם $q \leq \ell$ מקומות, קורא רק את הביטים המתאימים בהוכחה, ומחליט אם לקבל או לדחות (נניח כי $\bar{V}$אינו אדפטיבי). דורשים:

- completeness: בהנתן קלט "חוקי" $(x, F(x))$, $\bar{V}^\pi$ תמיד מקבל את ההוכחה π ש־$\bar{P}$ מייצר.
- soundness: בהנתן קלט "לא חוקי" (x, y^*), כך ש $y^* \neq F(x)$, **ולכל** הוכחה π^* המוודא $\bar{V}^{\pi^*}(x, y^*)$ דוחה בהס' 99%.
- fast verification: זמן הריצה של $\bar{V}$: $\ell \geq T_F \gg T_{\bar{V}}$.

משפט 0.3 PCP (מקרים מיוחדים)

- לכל פונ' F כנ"ל קיימת מערכת PCP עם הוכחה באורך $\ell = poly(T_F)$, עם מס' שאילתות $q = O(1)$.
- לכל פונ' F כנ"ל קיימת מערכת PCP שבה $T_{\bar{V}} = n \cdot polylog(T_F)$ וכן $T_{\bar{P}} = T_F \cdot polylog(T_F)$.

<u>הערות (לידע כללי)</u>

- המשפטים למעשה נכונים עבור NP (או באופן כללי יותר $Ntime$), כלומר פונ' $F(x,w)$, בה ל־$\bar{P}$ יש עד.
- המשפט הראשון (בגרסת ה־NP) הנו הבסיס לכמעט כל תוצאות הקושי ל"בעיות קירוב".

מאיפה באים $PCPs$ לעולם? (רמזים שטחיים)

- שני סוגים עיקריים:
  - ה־PCP הראשונים התבססו על טכניקות אלגבריות - תכונות של פולינומים מעל שדות סופיים.
  - כיום יש גם בניות קומבינטריות לחלוטין.
- PCP אלגבריים:
  - מקודדים את החישוב בכמה פולינומים מ"דרגה נמוכה".
  - וידוא מתרגם לבדיקת תכונות מסוימות של הפולינומים וקונסיסטנטיות זה עם זה.
  - בעיה יסודית: איך בודקים ששני פולינומים $P, P' : \mathbb{F} \to \mathbb{F}$ מדרכה נמוכה זהים עם מס' קטן של שאילתות? $|\mathbb{F}| \gg d$

מ־PCP לוידוא מהיר של חישובים

- מה חסר? האם $PCPs$ אינם מבטיחים כבר חישוב מהיר?
- פתרון נאיבי:
  - V מריץ את מוודא ה־PCP, $\bar{V}$, המייצר שאלות $i_1, \ldots, i_q \in [\ell]$ ושולח אותן ל־P.
  - P מריץ את מוכיח ה־PCP, $\bar{P}$, מייצר הוכחה π ומחזיר ל־V את $\pi[i_1, \ldots, i_q]$.
  - V שוב מריץ את $\bar{V}$ על־מנת להשלים את הוידוא.
- הבעיה: ניתן להבטיח $soundness$ רק כאשר ההוכחה π^* נקבעת לפני וללא תלות בשאילתות של $\bar{V}$.
- אינטואיציה: חשבו על מודל ה־MIP, אם המוכיחים יכולים לבחור אסטרטגיה משותפת לאחר השאלות של החוקר (בטרם הכניסו אותם לחדרים נפרדים) , קל לרמות.
- צריך "לאכוף" את מודל ה־PCP כיצד ניתן לפתור את הבעיה בעזרת קריפטו?

פרוטוקול אינטרקטיבי - $Kilian$

- קלט משותף: (x, y) כך ש-$y = F(x)$,
- נסיון ראשון
  1. $P : P \to V$ מחשב PCP., π, ושולח ל-V.
- נסיון שני
  1. $P : P \to V$ מחשב PCP., π, ושולח ל-V האש קצר $d = H(\pi)$.
  2. $V : P \leftarrow V$ דוגם שאילתות PCP $i_1, \ldots, i_q$ ושולח ל-P.
  3. $P : P \to V$ שולח ל-V $\pi[i_1, \ldots, i_q]$ ומראה קונסיסטנטיות עם d. איך?
- תזכורת: Merkle Tree Hashing.
- נסיון שלישי
  1. $P : P \to V$ מחשב PCP., π, מחשב $root = MT_H(\pi, \epsilon)$ ושולח ל-V.
  2. $V : P \leftarrow V$ דוגם שאילתות PCP $i_1, \ldots, i_q$ ושולח ל-P.
  3. $P : P \to V$ שולח ל-V $\pi[i_1, \ldots, i_q]$ ומראה קונסיסטנטיות עם $root$: שולח מסלולי האש $path(i_j) = MT_H(\pi, i_j)$.
- כיצד מוכיחים $soundness$? רדוקציה ל-$soundness$ של ה-PCP ול $collision - resistance$. (במקור - complexity leveraging, לאחר מכן - Barak-Goldreich-$\pi^*[i] = \lfloor \mathbb{E}[\pi[i] : i \in i_1, \ldots i_q] \rceil$ ערך שכיח בממוצע. דרושות גם תכונות דגימה מסוימות.)

פרוטוקול לא-אינטרקטיבי?

- כיצד ניתן להפוך את הפרוטוקול של $Kilian$ לפרוטוקול בן שתי הודעות?
- אמרנו כי אינטואיטיבית הבעיה היא ש-P יכול לבחור את ה-PCP π^* באופן התלוי בשאילתות.
- מה אם במקום לשלוח ל-P את השאילתות נשלח אותן מוצפנות? האם אפשר להשתמש למשל ב-PIR עם סיבוכיות תקשורת נמוכה?
- נסיון ראשון. תהי (E, A, D) מערכת PIR.

1. V: דוגם שאילתות PCP $i_1, \ldots, i_q$ ושולח ל־P שאילתות מוצפנות $\{c_j \leftarrow E(i_j)\}_{j\in[q]}$.
2. P: מחשב PCP π, ושולח ל־Vתשובות $\{a_j = A(\pi, c_j)\}_{j\in[q]}$.

- האם זה בטוח? דמיינו שה־PIR ממומש ע"י FHE למשל. כיצד ניתן לתקוף?
- אפשר עדיין לשלוח $root = MT_H(\pi, \epsilon)$ ולשים ב־DB גם את מסלולי ההאש הרלבנטיים.
- האם עכשיו בטוח? לא ברור ש־collision-resistance מספיק.

משפט 0.4 (Gentry-Wichs) אי אפשר להוכיח בטיחות בעזרת רדוקציה שפשוט "מריצה את היריב" ולא משתמשת בקוד שלו.

- כל הוכחות הבטיחות שראינו עובדות כך.
- איך מוכיחים משפט שכזה?