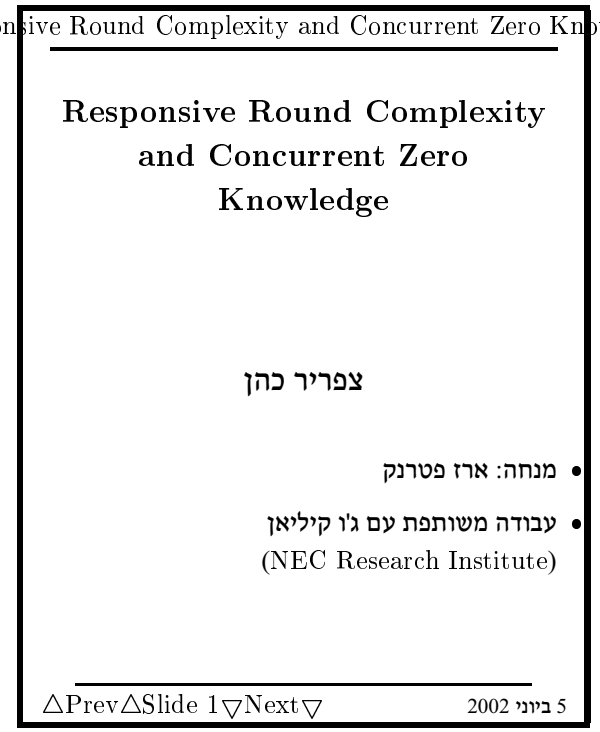


שקף 1

זמן נוכחי: 0:0 | השקף הזה: 1:0

זוהי הרצאה על Responsive Round Complexity and Concurrent Zero Knowledge, מושג אשר אני מקווה שתבינו בסופה.

אני צפריר, וזוהי עבודת המסטר שלי שנעשתה בהנחיית ארז פטרנק וביחד עם ג'ו קיליאן מ-(NEC Research Institute)

רשימת השקפים

1 Responsive Round Complexity and Concurrent Zero Knowledge
2 בתוכנית
3 הוכחות אינטראקטיביות [Goldwasser-Micali-Rackoff]
4 דוגמה: (GI) Graph Isomorphism
5 דוגמה: GI: הפרוטוקול
7 דוגמה: GI: נכונות הפרוטוקול
8 דוגמה: GI: סימולציה
9 דוגמה: GI: סימולטור ראשון
10 דוגמה: GI: סימולציית Black Box:
11 דוגמה: GI: סימולטור Black Box:
12 הגדרה: אפס מידע (Zero-Knowledge), גירסת Black-Box
13 המשפט היסודי של Zero-Knowledge
14 בתוכנית
15 דוגמה: GI: מיקבול:
16 התחייבות Commitment
17 דוגמה: GI: מיקבול: הסימולטור
18 הרכבה אסינכרונית של הוכחות
19 הבעיה: תיזמון "פירמידה"
20 החסם התחתון הנוכחי
21 התמודדות עם תזמון "פירמידה"
22 מודלים מוגבלים
23 בתוכנית
24 Responsive-Round Complexity
25 Responsive-Round Complexity: הגדרה
26 בתוכנית
27 תוצאה עיקרית
28 תוצאה עיקרית (המשך)
29 הפרוטוקול הבסיסי
30 הפרוטוקול המלא
31 הוכחת מערכת ההוכחה
32 מערכת ההוכחה: הסימולציה
33 מערכת ההוכחה: הסימולציה: הדגמה
34 הערות
35 סיכום
36 מקורות

שקף 2

בתוכנית

- **הוכחות באפס-מידע**:
  אפס-מידע, קופסה שחורה, דוגמה
- אפס-מידע בסביבה אסינכרונית:
  מקבול ההוכחה, המודל האסינכרוני, מה הבעיות?
  אלטרנטיבות?
- מדד סיבוכיות חדש: Responsive-Round-Complexity
- מערכת ההוכחה שלנו
- סיכום

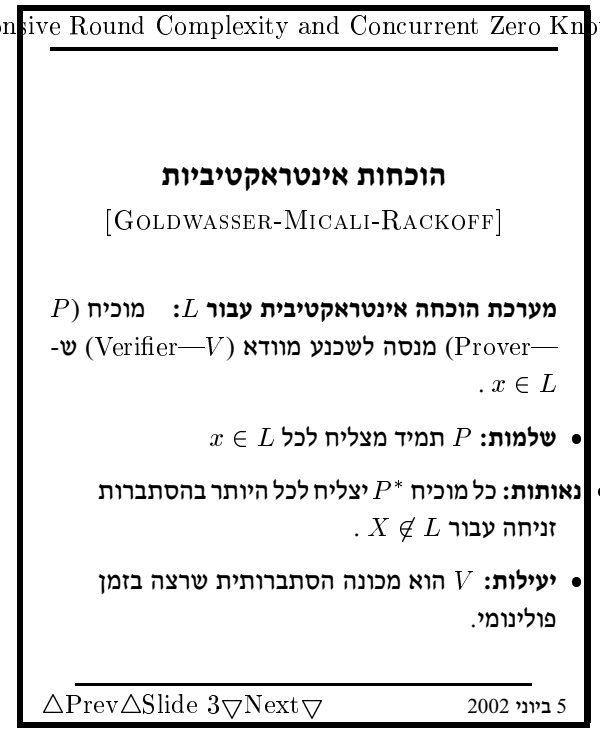
△Prev△Slide 2▽Next▽ 5 ביוני 2002

זמן נוכחי: 1:0 השקף הזה: 0:30

אני אתחיל ברקע על הוכחות באפס מידע. נראה גם סוג ספציפי של הוכחות באפס מידע: הוכחות black-box.

לאחר מכן נראה אילו בעיות יש כאשר מנסים להריץ את הדברים הללו בסביבה אסינכרונית.

בהמשך אני אציג את המושג *responsive round complexity* (סיבוכיות זמן תגובה) ואציג מערכת הוכחה חדשה ל-$\mathcal{NP}$ אשר משיגה סיבוכיות טובה לפי מדד זה.



שקף 3

זמן נוכחי: 1:30	השקף הזה: 1:20

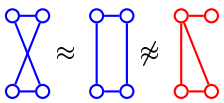
אני מניח שהמונח "הוכחה אינטראקטיבית" לא זר לרובכם, אבל תזכורת קצרה:
הוכחה אינטראקטיבית לשפה L היא פרוטוקול בין שני צדדים: המוכיח (P—Prover) מנסה לשכנע את המוודא (V—Verifier) בנכונות של טענה מסויימת: שהקלט ששניהם קיבלו, x שייך לשפה.
לשכנע כאן פירושו שאם x הוא ב-L הטענה תמיד תתקבל (שלמות), ואילו אם x אינו ב-L הקלט יידחה כמעט תמיד, לא משנה מה יעשה המוכיח (נאותות). ל-V לא יכולים להיות שום כוחות מיוחדים. הוא צריך להיות פולינומיאלי-הסתברותי רגיל.

במערכת הוכחה טיפוסית המוודא שולח כל מיני אתגרים למוכיח, והמוכיח עונה עליהם. המוכיח האמיתי יכול תמיד לענות עליהם, ואילו למוכיח מתחזה יש סיכוי משמעותי לטעות.
ברור שאם $L \notin \mathcal{BPP}$, P חייב להיות בעל כוחות מיוחדים כלשהם (ולו עד לטענה לנכונות הטענה, אם $L \in \mathcal{NP}$.

שקף 4

דוגמה: (GI) Graph Isomorphism

- בהינתן גרפים $G_1 = (V, E_1)$, $G_2 = (V, E_2)$
  האם יש פרמוטציה $\pi : V \to V$ כך ש-$(u, v) \in$
  $(\pi(u), \pi(v)) \in E2 \Leftrightarrow E_1$?
- סימון: $G_1 \approx G_2$. יחס שקילות.
- השפה: $GI \triangleq \{(G_1, G_2) : (G_1 \approx G_2)\}$



△Prev△Slide 4▽Next▽ 5 ביוני 2002

זמן נוכחי: 2:50	השקף הזה: 1:20

נפתח בדוגמה לפרוטוקול של הוכחה באפס מידע.

השפה GI (Graph Isomorphism) היא שפה של זוגות הגרפים האיזומורפים, כלומור זוגות הגרפים כך שיש פרמוטציה על צמתי הגרף שהופכת את הגרף הראשון לשני.

לא ידוע אלגוריתם כללי להכרעה בשאלה האם שני גרפים הם איזומורפיים בזמן פולינומי[1] *[עד כמה היא קשה לקירוב?]* אבל בהינתן שני גרפים ופרמוטציה קל לקבוע האם הפרמוטציה הזו על צמתי הגרף מעבירה את הגרף הראשון לשני. לכן הבעיה הזו נמצאת ב-$\mathcal{NP}$.

באופן פורמלי הקלט הוא זוג סדור של גרפים, G_1, G_2. הקלט שייך לשפה אם יש פרמוטציה מצמתי G_1 לצמתי G_2 אשר לאחר הפעלתה מתקבל גרף זהה לגרף G_2.

נשים לב שהאיזומורפיזם בין גרפים הוא יחס שקילות. בפרט אם G_1 איזומורפי ל-G_2 ו-G_3 איזומורפי ל-G_1 אזי גם G_2 איזומורפי ל-G_3.

[1] בגודל הגרף

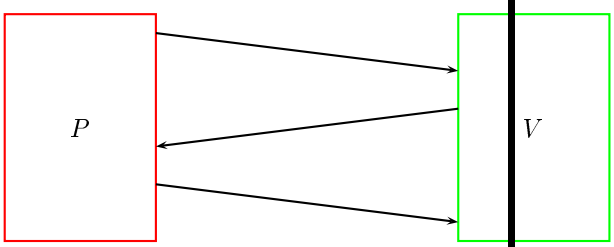
Slide 5

דוגמה: GI: הפרוטוקול

פרוטוקול הוכחות באפס מידע עבור GI (שלב בסיסי):

1. P יוצר פרמוטציה אקראית, π. שולח $G = \pi(G_1)$
2. V בוחר (באקראי) $b \in 1, 2$ ושולח את b ל-P
3. P שולח π_b כך ש-$\pi_b(G) = G_b$
4. V מקבל אם אכן מתקיים $\pi_b(G) = G_b$

שקף 6



זמן נוכחי: 4:10 השקף הזה: 2:0

נניח שאני טוען שזוג גרפים מסויימים הם איזומורפיים. איך אני יכול להוכיח לכם שאני לא טועה (או משקר)?

אני יכול להראות לכם את הפרמוטציה. אבל הפרמוטציה הזו יכולה להיות היא הסוד המקצועי שלי שעליו אני מגונן בקנאות רבה. האם אפשר להוכיח את זה בלי לחשוף את הפרמוטציה עצמה?

כדי להוכיח את יכולתי אני אאפשר לכל אחד ליצור לי מבחן. במבחן הזה אני תמיד יכול להצליח אם אמנם אני מכיר פרמוטציה כזו. אם אני לא מכיר פרמוטציה כזו, יש לי סיכוי לא רע להכשל במבחן.[1]

מהו המבחן?

אני (המוכיח, P) יוצר פרמוטציה אקראית (π) ושולח לבודק (מוודא, V את תוצאות הפעלתה על G_1. כזכור, אם G_1 איזומורפי ל-G_2 אז הגרף החדש יהיה איזומורפי גם ל-G_2.

בשלב (2) הבודק מנסה להפתיע אותי: הוא מבקש ממני פרמוטציה מאחד מהגרפים (G_1 או G_2). בשלב (3) אני מחזיר לא את הפרמוטציה המבוקשת. אם זו אמנם הפרמוטציה במעבירה את הגרף שהוא ביקש ל-G_1, הוא אמור להשתכנע.

[1] לגבי הרצת המבחן מספר פעמים: בהמשך

שקף 7

דוגמה: נכונות הפרוטוקול GI

- **שלמות**: אם $G_1 \approx G_2$ (כלומר: $G_2 = \pi'(G_1)$), אזי P יכול לענות כיאות על כל אתגר b *[ששולח המוודא]*
- **נאותות**: אם $G_1 \not\approx G_2$ (כלומר: π' כזה לא קיים) אזי מרגע שנשלח G יש רק b אחד עליו יכול P לענות כיאות.
  - $\Leftarrow$ אם $(G_1, G_2) \notin \mathrm{GI}$, אזי לאחר n צעדים בסיסיים P מצליח לרמות בהסתברות של לכל היותר 0.5^n
- **אפס-מידע** (אינטואיטיבית): V לא לומד שום דבר רלוונטי: רק כמה גרפים ופרמוטציות אקראיות.

△Prev△Slide 7▽Next▽ 5 ביוני 2002

זמן נוכחי: 6:10	השקף הזה: 1:45

אם טוענים שפרוטוקול מסויים הוא *מערכת הוכחה באפס מידע* אזי טוענים שבנוסף לתנאים הרגילים שנדרשים ממערכת הוכחה, נדרש גם שהמוודא לא ילמד כלום (למעט מה שרצינו להוכיח). נבדוק האם הפרוטוקול הוא אמנם מערכת הוכחה כזו.

- שלמות: אם אני מוכיח אמיתי, אני יכול להוכיח כל דבר נכון. במקרה שלנו: אם הקלט שייך לשפה ("נכון"), אם אני יודע להוכיח, ואם הבודק מקיים את הפרוטוקול, התוצאה בהכרח תהיה חיובית.

  זה אכן מתקיים: אם הבודק בחר ב-G_1 אני אחזיר לו את π^{-1}. אם הוא בחר ב-G_2 ואני יודע מהי הפרמוטציה π' בין צמתי G_1 אני אחזיר לו $\pi' \circ \pi^{-1}$ *[אולי הפוך?]*

- נאותות: אם אני מתחזה יש לי סיכוי סביר להכשל: אם אין פרמוטציה כזו (או שלא הצלחתי למצוא אותה) אזי מראש יש לי יכולת לענות רק על אחת משתי השאלות האפשריות של הבודק. אני קובע איזו שאלה ע"י כך שאני בוחר לאיזה משני הגרפים הגרף החדש יהיה איזומורפי.

  זה נותן לי הסתברות הצלחה ברמאות של בערך חצי. אם נחזור על זה n פעמים, נוריד את ההסתברות לבערך 2^{-n}. כלומר אחרי מספר חזרות סביר אפשר להגיע להסתברות די נמוכה להצלחה ברמאות.

מה לגבי אפס מידע? כמו שבוודאי ניתן היה לראות, המוודא לא קיבל כאן אינפורמציה מעניינת. מה הוא כבר יכול לעשות עם כמה גרפים אקראיים ופרמוטציות אקראיות?
וכמו שניתן לראות, הטיעון הזה נטען בשיטה החביבה של נפנופי ידיים. צריך דרך יותר מסודרת לנסח ולהוכיח את הטענה ש-"המוודא לא למד שום דבר מעניין"

שקף 8

דוגמה: GI: סימולציה

- נרצה להראות ש-V היה יכול לייצר תעתיק הוכחה (בהתפלגות מתאימה) בעצמו ללא P
- נבנה סימולטור S (יעיל) אשר מייצר תעתיק שנראה "בדיוק כמו" שיחה בין P ל-V

△Prev△Slide 8▽Next▽ 5 ביוני 2002

זמן נוכחי: 7:55 השקף הזה: 0:40

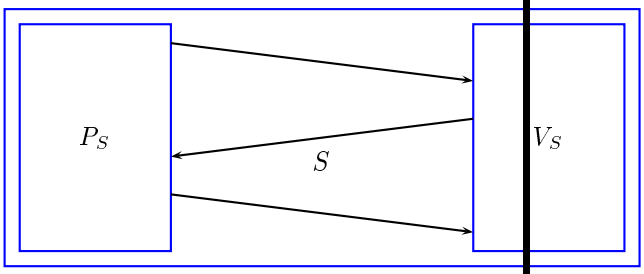
קצת קשה לנסח בדיוק את הטענה ש-"לא נלמד שום דבר מועיל מלבד העובדה שהטענה נכונה". אז ננסה להראות טענה קצת יותר חזקה ויותר ברורה: *כל מה שעובר בפרוטוקול ניתן לייצור ע"י משתתף חיצוני*. אם אני אצליח להראות לכם מכונה שמייצרת שיחות שלא ניתן להבדיל בינן לבין מה שעובר בפרוטוקול, והמכונה הזו היא מכונה פולינומית (הסתברותית) רגילה, אז מי שהאזין מהצד בוודאי לא למד שום דבר מועיל: הוא היה יכול באותה מידה להריץ את המכונה בעצמו.

שקף 9

דוגמה: GI: סימולטור ראשון

נסיון ראשון (מדמה בודק הגון):

1. S בוחר (באקראי) $b \in \{1,2\}$ ופרמוטציה, π.
   מחשב $G = \pi(G_b)$ וכותב "$P \to V : G$"
2. כותב "$V \to P : b$"
3. כותב "$P \to V : \pi$"



△Prev△Slide 9▽Next▽ 5 ביוני 2002

זמן נוכחי: 8:35	השקף הזה: 1:50

נראה הצעה למכונה כזו עבור הפרוטוקול: המכונה הזו בעצם משחקת את שני התפקידים. בשלב ראשון היא מגרילה מראש את האתגר של הבודק. מרגע שהאתגר ידוע, אין כל קושי בהכנת פרמוטציה רק לגרף הזה ולא לגרף השני, ואכן מכאן היא ממשיכה ורושמת את "תפקידיהם" של שני המשתתפים.

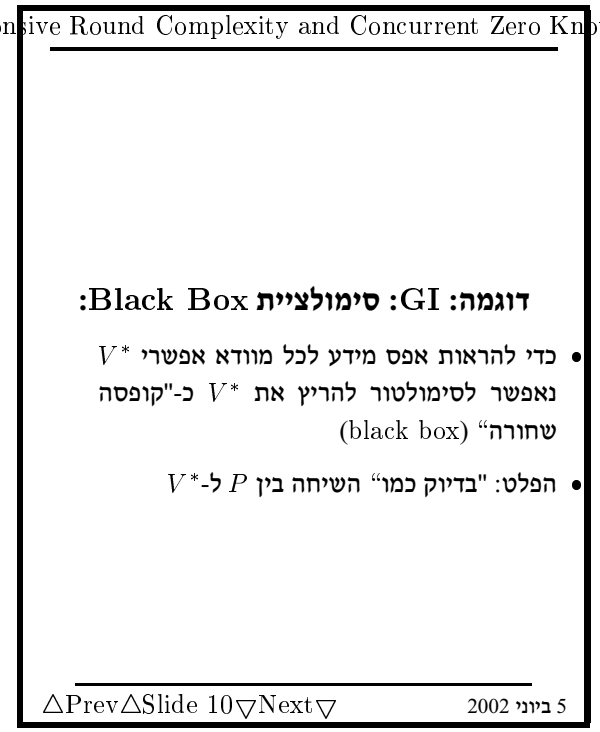
נשים לב שההתפלגות של התעתיקים שייווצרו היא בדיוק כמו ההתפלגות של שיחות בין מוודא למוכיח אשר פועלים לפי הפרוטוקול. אז מה לא בסדר כאן?

הבעיה היא שבנסיבות המתאימות התפלגות השיחות שהסימולטור הזה ייצר לא תהיה טובה. יכול להיות, לדוגמה, שמוודא מסויים, V^*, לא בוחר את המטבעות האקראיים שלא ממש באקראי אלא כפונקציה של הערך של π שהתקבל בשלב 1. יכול להיות שהבודק גם מחליט לבחור תמיד 2. בשני המקרים הוא סוטה מהפרוטוקול, אולם למוכיח האמיתי לא אמורה להיות שום בעיה לענות לו.

אולם נקבל שההתפלגות של השיחות בין P ל-V^* החדש אינה כמו זו שמוציא הסימולטור שלנו! מכאן שהסימולטור צריך לדעת איך יתנהג הבודק כדי לסמלץ את התנהגותו בפרוטוקול.

[לנסח מחדש את הפסקה האחרונה]

שימו לב ש-V עדיין יוצא משוכנע. זה בגלל שהוא השתתף בפרוטוקול וקבע מראש מבחנים. ***להרחיב.*** ***למצוא איפה למקם את זה]***



שקף 10

זמן נוכחי: 10:25	השקף הזה: 0:30

איך סימולטור יוכל להתאים לכל התנהגות אפשרית של המוודא? ברור שהוא לא יכול לחזות מראש את ההתפלגות בלי להכיר את האלגוריתם של המוודא. גם אם נתון לו האלגוריתם של המוודא, לא ברור אם הוא יוכל לנתח אותו ולהוציא ממנו משהו מועיל.

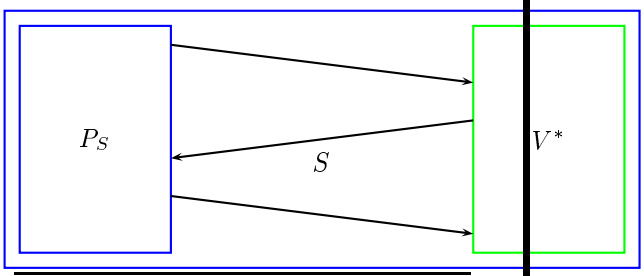
משהו שבוודאי אפשר לעשות הוא להשתמש ב-"מכונת מוודא" לשם הרצת הסימולציה.מכיוון שמשתמשים במכונת המוודא כקופסה שחורה, מבלי התחשבות בדרך פעולתו. קוראים להוכחות כאילו *אפס מידע black box*.

דוגמה: GI: סימולטור Black Box:

1. S בוחר (באקראי) $b \in \{1,2\}$. ופרמוטציה , π
ומחשב $G = \pi(G_b)$. מריץ V^* על סרט אקראי,
הקלט (G_1, G_2) וההודעה הראשונה: "G"

2. מקבל את התשובה $\sigma \in \{0,1\}$ מ-V^*. אם $b \neq$
σ חזור ל (1). אחרת החזר: "$G : V \to P$",
"$b : P \to V$", "$\pi : P \to V$".

שקף 11



זמן נוכחי: 10:55 השקף הזה: 2:10

נראה אלגוריתם סימולטור כזה:

הסימולטור הזה מקבל מכונה V^*. הוא יכול להריצה (כולל הרצה חלקית). מטרתו היא לרשום לפלט שיחה שתראה כמו שיחה בין V^* לבין המוכיח.

הוא פועל בצורה די דומה לסימולטור הקודם. הוא מגריל מבחן, ולפיו יוצר גרף לשלב (1) שעליו הוא יכול לענות. כעת הוא מריץ את המוודא ושואל אותו מהו המבחן שלו. נשים לב שבניגוד לסימולטור הקודם, כאן הוא לא יכול לצפות מראש את תשובתו של V^*.

לכן בנקודה זו יכול להיות שהניחוש שלו היה נכון, ואז הוא יכול להמשיך ולרשום את השיחה הזו כשיחת הפלט. אולם אם הניחוש לא היה נכון הוא זורק הכל, ומתחיל מהתחלה (הוא עדיין לא כתב כלום לפלט!).

ברור שכאשר הוא יוציא שיחה היא תהיה נכונה. כמוכן ניתן לראות שהתפלגות השיחות שיוצאות מכאן היא כמו התפלגות השיחות שהיו נוצרות משיחות בין V לבין המוכיח המקורי, P.

שקף 12

הגדרה: אפס מידע (Zero-Knowledge), גירסת Black-Box

אפס מידע: [Goldwasser-Micali-Rackoff], **גרסת** Black-Box: [Goldreich-Oren]

הוכחה אינטראקטיבית היא (P,V) באפס מידע black-box אם קיים סימולטור יעיל S כך שעבור כל מוודא V^*:

אם $x \in L$, קשה להבדיל בין $(P,V^*)(x)$ לבין $S_{V^*}(x)$

- S מתקשר עם *[מפעיל את(?)]* V^* כ-"קופסה שחורה"
- S קובע את סרט הטלות המטבע של V^*
- S יכול להחזיר אחורה את V^* ולנסות תשובות שונות.

זמן נוכחי: 13:5 השקף הזה: 2:30

אז הנוסח הפורמלי שלנו ל"הוכחה לא מדליפה מידע" הוא: "יש אלגוריתם סימולטור אשר עבור כל מוודא נתון מייצרת בעזרתו שיחות שנראות בדיוק כמו השיחות בין המוודא למוכיח אמיתי".

כאן המקום להבהיר כאן נקודה בעייתית:
אם הסימולטור יכול לרמות, איך ייתכן שהמערכת הזו היא נאותה? איך סומכים על ההוכחות שלה?
התשובה לשאלה היא שהסימולטור לא מרמה מודא אמיתי. מוודא אמיתי משתכנע בשל העובדה שהוא בחר בעצמו את המבחנים למוכיח. הסימולטור מצליח לרמות כאן כי יש לו יכולת להתחרט ולהתחיל מהתחלה (או מהאמצע, כמו שנראה בהמשך).
נשים לב שהסימולטור מריץ את V^* וקובע את כל תנאי ההתחלה שלה, כולל סרט המטבעות האקראיים. לכן יש ביכולתו להקליט את המצב בנקודה נתונה, ולהתחיל את ההרצה ממנה מאוחר יותר. השלייטה על סרט המטבעות האקראיים מבטיחה ששתי הריצות יתקדמו באותו המסלול אם יינתן אותו הקלט.

Zero-knowledge proofs are proofs in which the verifier learns nothing new except the fact that was proven in the protocol. This is formulated by showing that the verifier could have created a convincing session without the help of the all-mighty prover. That is: For any verifier we can build a simulator, S, that produces interaction which is indistinguishable from the interaction between P and V.

In almost all the zero-knowledge proof systems, the proofs for separate verifiers don't differ much: there is one simulator algorithm that uses the verifier algorithm as a sub-procedure, a **black box**.

Formally, an interactive proof protocol is black-box zero-knowledge if there exists an efficient simulator algorithm S such that for any possible algorithm V^* the interaction between P and V^* is indistinguishable from the output of the simulator that uses V^*.

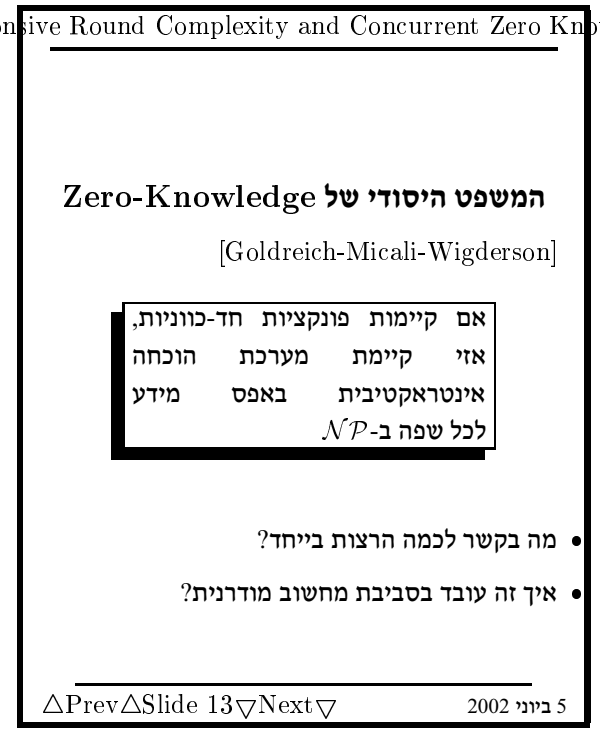
The simulator needs to provide a "cheating" prover. The simulator will simulate the interaction using this prover and the provided verifier. When this simulator's prover is presented with a challenge from the verifier, it will give a simple answer (that by itself is not enough to convince the verifier). However, now that it knows what the verifier is going to ask, it will re-run the simulation, with the same random coins to the verifier.

The original proof was sound because the prover could not have prepared in advance to the challenges sent by the verifier. But in this re-run the "prover" (S) knows the challenges. And this is enough for the simulator to create answers to those challenges.

The simulator vs. a real prover

- The Simulator may always create a "valid proof". any $x \notin L$ is not of its concern.
- It sets the coin flips of the verifier

$\Rightarrow$ therefore it can "rewind" the simulation: The simulator can explore different paths of execution by rerunning the simulation up to a certain point with exactly the same environment.



שקף 13

זמן נוכחי: 15:35	השקף הזה: 0:40

[Goldreich-Micali-Wigderson] הציעו מערכות הוכחה באפס מידע לכל שפה ב-$\mathcal{NP}$, שבהן הדרישה המיוחדת היחידה מהמוכיח היא עד לטענה שהוא מוכיח. זו היתה התפתחות מרשימה שאפשרה להשתמש באפס-מידע בפרוטוקולים קריפטוגרפיים חשובים רבים. ***[להכין כמה דוגמאות]***

אולם תקפות מערכות ההוכחה הללו הוכחה רק עור המקרה שבו המוודא והמוכיח מנותקים לחלוטין מהעולם, ואפילו קשרי נוספים ביניהם אינם מורשים. הדוגמה הקודמת, של הוכחה במקביל, כבר חורגת במקצת ממודל זה. וסביבת רשת נורמלית שבה יכולים לרוץ במקביל כל מיני קשרים עם תזמון לא מתואמלחלוטין, בוודאי שאינה מכוסה על-ידו. אנו בכל זאת רוצי למצוא פרוטוקולים שיאפשרו עבודה ברשת.

שקף 14

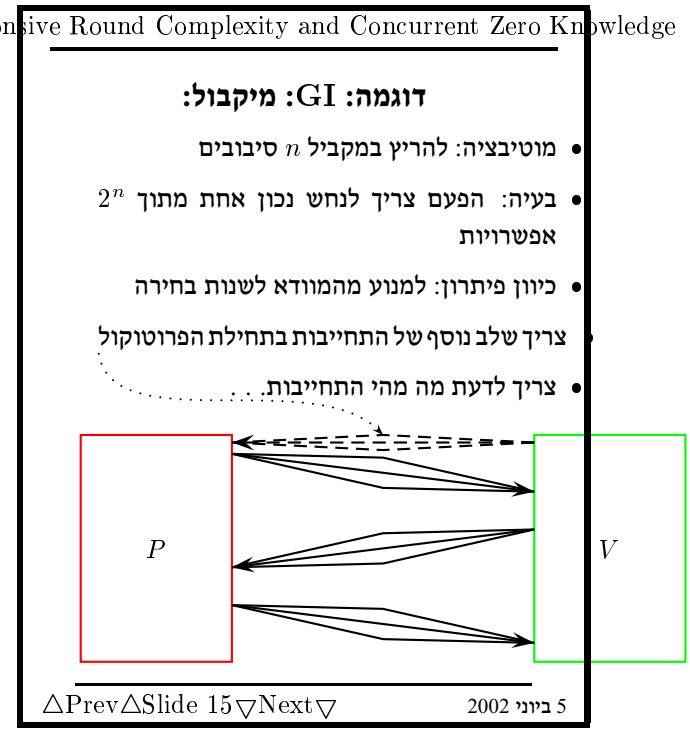
בתוכנית

- הוכחות באפס-מידע:
  אפס-מידע, קופסה שחורה, דוגמה
- **אפס-מידע בסביבה אסינכרונית**:
  מקבול ההוכחה, המודל האסינכרוני, מה הבעיות?
  אלטרנטיבות?
- מדד סיבוכיות חדש: Responsive-Round-Complexity
- מערכת ההוכחה שלנו
- סיכום

△Prev△Slide 14▽Next▽ 5 ביוני 2002

זמן נוכחי: 16:15 השקף הזה: 0:10

שקף 15



זמן נוכחי: 16:25 השקף הזה: 2:0

כדי להוריד את הסתברות הרמאות למשהו נורמלי כמו 2^{-n} צריך להריץ את הפרוטוקול הקודם n פעמים. האם אי-אפשר לעשות את זה במקביל? נריץ ביחד n עותקים של הפרוטוקול. כולם ישלחו ביחד את ההודעות של כל שלב.

הבעיה היא שזה לא עובד. הסימולטור שלנו לא יצליח לדמות ביעילות את האינטרקאקציה עם מוודא כזה: הוא צריך לנחש מה בחר המוודא, ואילו ההסתברות שלו לעשות זאת כאן היא רק 2^{-n}.

מה ששהיה נחמד אם היתה לו דרך לשמור על ניחושיו הנכונים. אם היתה דרך למנוע מהמוודא לשנות את בחירתו, הסימולטור היה יכול לא לנחש כל פעם מהתחלה אלא להסתפק בשניים-שלושה ניחושים.

כדי שזה יעבוד, נצטרך להוסיף בהתחלה עוד שלב: המוודא שולח למוכיח **התחייבות** על השאלה שלו. זה אומר שהו צריך להחליט מראש מהי השאלה שלו, לפני שהוא ראה את הגרף ששולח המוכיח. אבל הגרף הזה הוא ממילא סתם גרף אקראי.

מה שלא ברור מהפסקה הקודמת, זה איך המוודא יכול לשלוח את הניחוש שלו למוכיח מבלי לגלות אותו. לשם כך צריכים להבין מהי סכמת התחייבות:

שקף 16

זמן נוכחי: 18:25	השקף הזה: 1:0

סכמת התחייבות באה לדמות מצב שבו המתחייב מפקיד את הערך שעליו הוא מתחייב אצל שופט אמין, ולאחר מכן (בהסכמת המתחייב) השופט פותח את הערך שהופקד אצלו, ומגלה אותו. המתחייב לא יכול לשנות את הערך המופקד מרגע שהוא הופקד. מוודא ההתחייבות לא יכול לדעת מהו הערך שהופקד עד שהוא לא מגולה.

כמובן ששופטים אמינים על כולם הם דבר די נדיר, ופרוטוקולי מערכות התחייבות מנסים להסתדר בלי צד שלישי. ומצליחים לא רע. בפרוטוקול כזה יש שני שלבים:

1. **התחייבות**: המתחייב מעביר למוודא "התחייבות"
2. **גילוי**: המתחייב מגלה למוודא על מה הוא התחייב

ורוצים באופן כללי להבטיח שתי תכונות:

1. **חד-משמעיות**: מרגע שנתנה ההתחייבות, ישנו רק ערך תקין אחד שלו אפשר לפתוח את ההתחייבות ("ערך ההתחייבות")
2. **סודיות**: החל מההתחייבות ועד הגילוי המוודא לא יכול לקבל שום מידע על ערך ההתחייבות

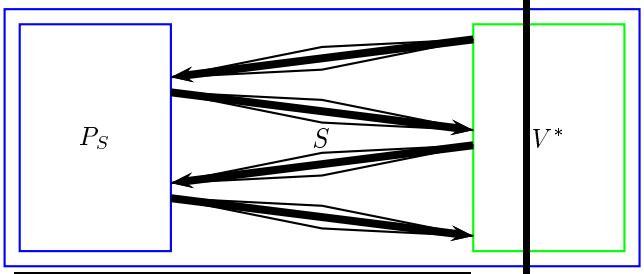
יש כאן נקודה עדינה של מה בדיוק הפרוטוקולים מבטיחים: עמידות מוחלטת (information-theoretic) או חישובית (computeational). מפאת קוצר הזמן רק אעיר שמה שקיים בהחלט מספיק. [1]

[1] פרטים נוספים: ב-*[להיסיף רפרנס על סכמות התחייבות]*

שקף 17

דוגמה: GI: מיקבול: הסימולטור

1. מריץ את V^* על סרט אקראי, מקבל התחייבויות.
2. יוצר (באקראי) n ביטים $b_1, \ldots, b_n$, ולפיהם יוצר n גרפים לסיבוב הראשון, $G^1 \ldots G^n$
3. ממשיך הרצת V^* עם $P_S \rightarrow V^* : G^1 \ldots G^n$
4. אם אחת התשובות היתה לא כצפוי: חוזר ל-2, אחרת: כותב תעתיק השיחה



△Prev△Slide 17▽Next▽ 5 ביוני 2002

זמן נוכחי: 19:25 השקף הזה: 2:0

[השקף לא גמור!!!!]

אז כעת השינוי בפרוטוקול הוא שלפני התחלת הפרוטוקול המקורי המוודא שולח התחייבות על השאלה שהוא ישאל.. כעת בהרצת הסימולציה הסימולטור יכול להתחרט לפני שליחת ההודעה הרביעית. לכן הוא מחזיר את הסימולציה אחורה לשלב של אחרי שליחת ההודעת הסימולטור הראשונה.

השם "החזרה" קצת מטעה. אפשר היה להבין ממנו שבשלב מסויים של הסימולציה חוזרים משלב 3 לשלב 1 [להדגים חזרה משלב 3 לשלב 1]. אבל בעצם מתחילים מחדש את הסימולציה עם אותם תנאי פתיחה [לחזור לשקף.4, לחזור לשקף הקודם, ולהגיע לשקף .2]

הרכבה אסינכרונית של הוכחות

Concurrent: (אסינכרוני) מוודאים רבים (V_i) מתחברים למוכיח (P) בשיחות שונות ולא תלויות.

מוכיח פרנואידי: כל המוודאים מתואמים ע"י תוקף יחיד. מטרתו היחידה של התוקף היא להוציא מידע מ-P.

Concurrent Zero-Knowledge: להבטיח שתכונת אפס המידע תישמר גם בסביבה אסינכרונית

שקף 18

זמן נוכחי: 21:25 השקף הזה: 0:50

במקרה הכללי יש לנו סביבה *אסינכרונית*. המונח המקובל בתחום לסביבה כזו בתחום הוא concurrent, אולם מילה זו אינה מוצלחת במיוחד, כי היא רומזת שמדובר רק על הרצה במקביל. למעשה סביבה אסנכרונית לחלוטין.

יש לנו מוכיח. מדי פעם ניגשים אליו מוודאים שונים ובזמנם החופשי פותחים איתו ב-*שיחות* (sessions). לכן ייתכנו שיחות רבות שרצות במקביל. לכולן יש את אותו המוכיח, *[לנסח תשובה לשאלה "" מה עם כמה מוכיחים?]* אולם המוודא יכול להיות שונה.

קיימת גם האפשרות שכל השיחות מנוהלות למעשה ע"י תוקף אחד. המוכיח שלנו לא יכול לשלול את האפשרות שכל שאר העולם מנוהל ע"י תוקף אחד, וכל מטרתו של התוקף הזה היא לגלות קצת מסודותיו.

מערכת הוכחה היא zero Knowledge **concurrent** אם היא מצליחה לשמור על תכונת אפס המידע גם באותה סביבה בעייתית.

ותיכף נראה שהסביבה אמנם בעייתית.

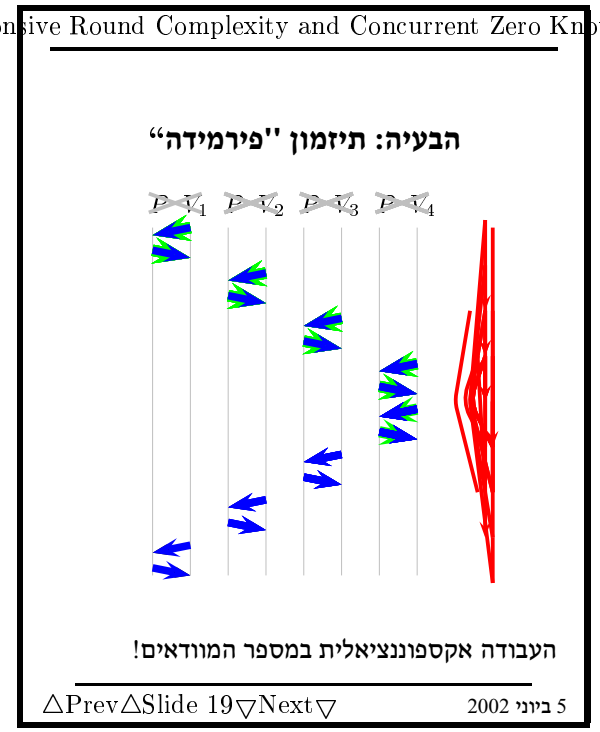
More general settings are the concurrent (or asynchronous) settings. In concurrent settings we may have multiple **sessions** of the protocol running alongside. All of them have the same prover[1], but may have different verifiers. It is also possible that all the different verifiers are actually controlled by a single adversary whose sole purpose is to extract some information from the prover.

A protocol is **concurrent**-zero-knowledge if the zero-knowledge property still preserved in such concurrent settings.

This has proven to be a non-trivial task.

[1] provers don't coordinate sessions

שקף 19



זמן נוכחי: 22:15	השקף הזה: 4:30

הנה דוגמה שמראה איך מוודאים מתואמים יכולים לנוע סימולציה יעילה עם סימולטור כמו זה מהדוגמה הקודמת. אנו רואים כאן האינטראקציה בין מוכיח אחד לארבעה מוודאים. כל האינטראקציה הזו צריכה להיות מסומלצת ע"י סימולטור.

כל פס אנכי מסמן כאן שיחה בין המוכיח לבין אחד מהמוודאים. חץ $\leftarrow$ מסמל הודעה מהמוודא למוכיח, וחץ $\rightarrow$ מסמל הודעה מהמוכיח למוודא.

הפרוטוקול של מערכת ההוכחה הוא זה שהוצג בדוגמה האחרונה, של הרצה במקביל. הוא כולל ארבע הודעות: התחייבות של המוודא, גרף של המוכיח, אתגר של המוודא, ופרמוטציה של המוכיח. אולם במקרה זה המוודאים השונים משהים במכוון את שליחת השאלה. למוכיח האמיתי אין בעיה אם זה, כמובן. אולם נזכור שהמוכיח של הסימולטור, P_S, צריך לדעת את השאלהמראש כדי לתת גרף בשלב (1) שלו יוכל לספק פרמוטציה בשלב (3).

על V_3. בפרט הסימולציה חזרה לזמן שהוא לפני ההתחייבות של V_4 ומסלולה השתנה. V_3 נפתרה, אבל . . .

שקף 5: . . . צריך לטפל ב-V_4 מחדש.

שקף 6: OK. שני אילו פתורים והגיע V_2. החזרה אחורה כאן פותרת את V_2 אך מקלקלת את השיחות הפנימיות. ולכן צריך שוב. . .

שקף 7: לטפל ב-V_4 מחדש

שקף 8: עכשיו אפשר לטפל ב- V_3 . . .

שקף 9: וב-V_4. להמשיך עם V_1?

שקף 10: העבודה אקספוננציאלית בכמות המוודאים!

שקף 2: הסימולטור מתקדם. הוא מנחש (באקראי) את האתגרים של המוודאים ולפיהם יוצר גרפים. כאחד מהם, V_4, סוף-סוף שואל את שאלתו, הסימולטור יודע איך לפתור אותה.

שקף 3: כעת S מחזיר לאחור את הסימולציה ושולח גרף מתאים ל-V_4. הוא לא החזיראותו יותר מדי אחורה, ולכן V_4 עדיין כבול להתחייבותו וחייב לשאול את אותה השאלה. לאחר שנשלח הגרף המתאים אנו אומרים ש-V_4 הוא *פתוד*.

שקף 4: עכשיו V_3 עונה והסימולטור משתמש בהחזרה לאחור כדי לפתור אותו. זה פותראמנם את V_3, אך פעולת ההחזרה פעלה על כל *הסימולציה*, ולא רק

שקף 20

החסם התחתון הנוכחי

כפי שהראו [Canetti-Killian-Petrank-Rosen]:

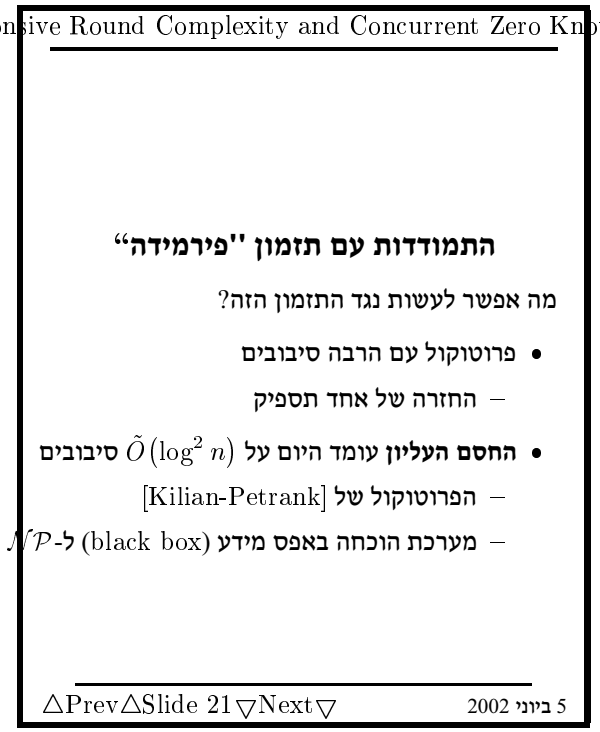
> סימולציית **black-box** של הוכחה לשפה שאינה ב-$\mathcal{BPP}$ דורשת $\tilde{\Omega}(\log n)$ סיבובים

(החסם התחתון תקף רק עבור הוכחות אפס מידע black-box)

זמן נוכחי: 26:45 השקף הזה: 0:40

הבעיה שהדגמתי היא לא בעיה ייחודית לפרוטוקול הזה של graph isomorphism ואי אפשר לעקוף אותה בקלות ע"י תזמונים קצת שונים של הסימולטור. [Canetti-Killian-Petrank-Rosen] הראו שכל שפה לא טריוויאלית, לא ב-$\mathcal{BPP}$, שיש לה מערכת הוכחה שהיא black-box אשר עמידה בתנאים אסינכרוניים, במערכת יש לפחות $\tilde{\Omega}(\log n)$ סיבובים.

נשים לב שהחסם הזה לא תקף לגבי מערכות הוכחה שבהן הסימולציה אינה black box, אולם אין הרבה מערכות כאילו.



שקף 21

זמן נוכחי: 27:25	השקף הזה: 2:0

כיוון אחד להתמודדות עם התזמון הזה הוא ע"י הוספת סיבובים.
יהיה כאן פרוטוקול שבו יהיו m סיבובים. במהלך הסימולציה מספיק לסימולטור לבצע החזרה של הסימולציה מסביב לאחד מהם בלבד כדי להצליח.
כעת, אם ארבעת המוודאים הללו מנסים להציג את אותו תזמון, הסימולטור לא מתרגש. הוא מטפל רק באחד מהם עכשיו, והיתר יחכו לסיבובים הבאים. הדבר נותן לו חופש פעולה רב בהרבה.
בעזרת פרוטוקול כזה הצליחו [Kilian-Petrank] לבנות מערכת הוכחה ל-$\mathcal{NP}$ ע אפס מידע אסינכרוני שעובדת ב-$\tilde{O}(\log^2 n)$. (שימו לב ל-tilde כאן, כלומר $O\left(\log\log n \cdot \log^2 n\right)$ סיבובים למשל, יעבוד.

שקף 22

מודלים מוגבלים

- **תזמון**: Timing Assumptions
  - **מגבלה על זמן של שלב בפרוטוקול**
  - **ר', לדוגמה** [Dwork-Naor-Sahai]
- Settings Assumptions
  - **הנחות נוספות על המודל** (כגון PKI)
  - **ר', לדוגמה** [Damgård], [Canetti-Goldreich-Goldwasser-Micali]
- Bounded Concurrency
  - **מספר המשתתפים בו-זמנית מוגדר מראש**
  - [Barak]

△Prev△Slide 22▽Next▽ 5 ביוני 2002

זמן נוכחי: 29:25	השקף הזה: 3:0

מכיוון שיש בעיה להשיג אפס-מידע אסינכרוני במספר מועט של סיבובים במודל הכללי, נעשה מחקר גם במודלים יותר מוגבלים. לדוגמה:

תזמון: Timing Assumptions המאמר המקורי של [Dwork-Naor-Sahai] שהציג את תזמון הפירמידה גם הציג פרוטוקול שבו לאחד השלבים יש אורך קבוע. פרוטוקול כזה מאפשר למוכיח ולסימולטור לסדר את המוודאים אחד אחרי השני: הם מסיימים באותו סדר שבו הם התחילו, ולכן לא יכול להיות תזמון פירמידה.

להקלות דומות על המודל קוראים בספרות timing assumptions

אבל אם אורך של שלב מסויים בפרוטוקול הוא קבוע, הפרוטוקול לא יתאים למוודאים בעלי השהיות רשת יותר גבוהות. או לחילופין: גם מוודאים המחוברים בחיבור טוב יצטרכו להתעכב כמו מוודאים המחוברים בחיבור הגרוע ביותר.

Settings Assumptions אפשרות אחרת היא לבחור סביבה אשר תאפשר למוכיח לקבל מראש פרטים על המוודא. המידע הזה מקל על הסימולטור לרמות את המוודא. ישנן כמה עבודות שעובדות במודלים כאילו (לדוגמה: מודל של public key infrastructure) ומשיגות פרוטוקול עם מספר סיבובים קבוע.

אבל לא תמיד אפשר להשתמש בהנחות הללו, ובכלל רצוי לא להגביל את עצמנו למודל כזה בלבד.

[לגמור לכתוב]

Bouded Concurrency עבודה די מעניינת מהשנה האחרונה הוא העבודה של בועז ***[זה השם?]*** ברק ממכון וויצמן. מה שמעניין בה במיוחד היא שזהו פרוטוקול שאינו black box. כלומר בפרוטוקול זה הסימולטור יודע להפיק מידע על דרך ריצת מכונת מוודא מתוך תאור המכונה עצמה. כמובן גם כאן יש פרוטוקול עם מספר סיבובים קבוע.

אבל זה עדיין לא בא בחינם: הפרוטוקול דורש הנחה מראש על מספר המשתתפים שכנגדם הפרוטוקול עמיד. כמובן גודל ההודעות בפרוטוקול תלוי במספר זה.

שקף 23

בתוכנית

- הוכחות באפס-מידע:
  אפס-מידע, קופסה שחורה, דוגמה
- אפס-מידע בסביבה אסינכרונית:
  מקבול ההוכחה, המודל האסינכרוני, מה הבעיות?
  אלטרנטיבות?
- **מדד סיבוכיות חדש**: Responsive-Round-Complexity
- מערכת ההוכחה שלנו
- סיכום

△Prev△Slide 23▽Next▽ 5 ביוני 2002

זמן נוכחי: 32:25 השקף הזה: 0:10

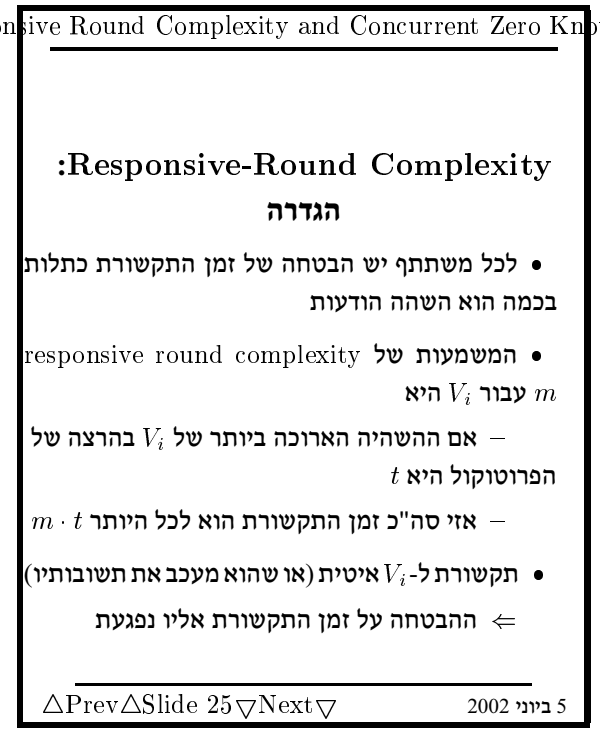
שקף 24

זמן נוכחי: 32:35	השקף הזה: 1:40

כאן הגענו סוף-סוף לעבודה הזו!

בעבודה הזו אנו עובדים במודל הכללי. אולם בתור התחלה אנו מציעים שינוי לממדד הסיבוכיות. במקור מודדים סיבוכיות מספר הסיבובים בפרוטוקול כדי לקבל הערכה מסויימת לזמן הריצה שלו. זה באמת נכון עבור המקרה הטיפוסי שבו עיקר הזמן מתבזבז על תקשורת, וכן אם אי כל מיני השהיות מוזרות. אבל יש מקרים שבהם סיבוכיות מספר הסיבובים לא מספקת הערכה טובה. לדוגמה, אם אחד הסיבובים בפרוטוקול ארוך מכל יתר שאר ביחד, מספר הסיבובים אינו מדד רלוונטי במיוחד.

ההצעה שלנו היא למדוד את מספר הסיבובים בעזרת זמן. *סיבוב* יהיה הזמן של הסיבוב הארוך ביותר, סיבוכיות הסיבובים החדשה תהיה מספר הסיסובים כנ"ל שנכנסים בזמן הביצוע.



שקף 25

זמן נוכחי: 34:15	השקף הזה: 1:10

Responsive-round complexity, כלומר סיבוכיות זמן תגובה, נמדדת בנפרד לכל משתתף, כי לא רוצים שעיכובים של משתתף אחד ישפיעו על המשתתפים האחרים. רוצים לתת לכל משתתף הבטחה על סה"כ זמן התקשורת איתו, שתהיה תלויה רק בכמה המשתתף הזה מעכב הודעות.

באופן פורמלי *[לתקן]* המשמעות של m responsive round complexity עבור V_i היא:

- אם ההשהיה הארוכה ביותר של V_i בהרצה של הפרוטוקול היא t
- אזי סה"כ זמן התקשורת הוא לכל היותר $m \cdot t$

אם התקשורת ל-V_i איטית, או שהוא בוחר לעכב משלוח הודעות, זמן התגובה שלו יעלה ולכן הפרוטוקול יבטיח זמן תקשורת יותר גבוה.

ההגדרה המקורית:

Response time: We say that the response time of a party A in round i of a specific run σ of a protocol Π is t, if the protocol Π in run σ tells A to send a message in round i, and t is the length of the time interval starting from the time B sent its message in round $i-1$ and ending at the time B received A's response of round i. (If A is not supposed to send a message in round i, then its response time is 0 for round i.) The response time of A in run σ of protocol Π is the maximum over all rounds i of A's response time in round i.

Responsive time complexity: We say that a protocol Π has responsive round complexity m with respect to Party A, if for any possible run σ of Protocol Π, the overall communication time does not exceed $t \cdot m$ where t is A's response time in Run σ of Protocol Π.

שקף 26

בתוכנית

- הוכחות באפס-מידע:
  אפס-מידע, קופסה שחורה, דוגמה
- אפס-מידע בסביבה אסינכרונית:
  מקבול ההוכחה, המודל האסינכרוני, מה הבעיות?
  אלטרנטיבות?
- מדד סיבוכיות חדש: Responsive-Round-Complexity
- **מערכת ההוכחה שלנו**
- סיכום

שקף 27

תוצאה עיקרית

Theorem: There exists a concurrent black-box zero-knowledge interactive proof for $\mathcal{NP}$ with responsive round complexity m=$\tilde{O}(\log n)$.

- לפרוטוקול קיליאן-פטרנק יש responsive round complexity של $m = \tilde{O}(\log^2 n)$
- החסם התחתון של [Canetti-Killian-Petrank-Rosen] תקף גם ל-responsive round complexity

זמן נוכחי: 35:35 השקף הזה: 1:0

בעבודה הזו הצגנו פרוטוקול למערכת הוכחה לכל שפה ב-$\mathcal{NP}$ להוכחות באפס-מידע בסביבה אסינכרונית. לפרוטוקול זה יש סיבוכיות זמן תגובה של $\tilde{O}(\log n)$ (כלומר, לדוגמה, $\log n \cdot \log\log n$ יתאים).

נשים לב שהחסם התחתון שהזכרתי קודם ([Canetti-Killian-Petrank-Rosen]) תקף גם לגבי סיבכיות זמן תגובה. הדבר הזה נובע מהתזמון הכללי שבו הם משתמשים בהוכחה כדי "לשבור" כל סימולטור black-box. *[להמשיך]*

כמוכן מניתוח הפרוטוקול של החסם העליון שהוצג קודם ([Kilian-Petrank]) ניתן לראות שגם סיבוכיות זמן התגובה שלו היא $\tilde{O}\left(\log^2 n\right)$, לא רק סיבוכיות הסיבובים שלו.

שקף 28

תוצאה עיקרית (המשך)

Theorem: There exists a concurrent black-box zero-knowledge interactive proof for $\mathcal{NP}$ with responsive round complexity m=$\tilde{O}(\log n)$.

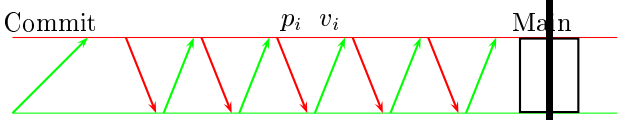
- הוכחה ע"י בניה
- הפרוטוקול הבסיסי: גירסה של הפרוטוקול של [Richardson-Kilian] עם $\tilde{O}(\log n)$ סיבובים
- למוודא יש זמן השהיה (קטן, בהתחלה).
- אם מוודא לא עונה "בזמן", הוא מתחיל מהתחלה את הפרוטוקול הבסיסי עם זמן השהיה כפול

אנו משתמשים בואריציה של האלגוריתם מ-[Kilian-Petrank], עם $\tilde{O}(\log n)$ סיבובים. כמוכן בפרוטוקול שלנו יש לכל שיחה זמן השהיה משלה. אם המוודא לא עונה תוך פרק זמן של זמן-השהיה זה, השיחה איתו מתחילה מחדש אולם עם השהיה כפולה.

שקף 29

הפרוטוקול הבסיסי

- **הקדמה:** m זוגות סיבובים
  - V מתחייב על m מחרוזות אקראיות: $v_1 \ldots v_m$
  - For i=1 to m
    - P מתחייב על מחרוזת אקראית p_i
    - V מגלה v_i
- **גוף:** הוכחה WI שאחת מן השתיים:
  - $x \in L$ (המוכיח יראה את זה)
  - $\exists i : \ v_i = p_i$ (הסימולטור יראה את זה)



זמן נוכחי: 37:20	השקף הזה: 1:30

לפרוטוקול יש שני חלקים: הקדמה וחלק ראשי. למעשה ההקדמה תופסת את עיקר הזמן.

להקדמה יש m סיבובים. היא מתחילה בכך שהמוודא בוחר m מספרים אקראי ומתחייב עליהם. כעת בכל סיבוב המוכיח ינסה לנחש את אחד מהמספרים, ולאחר מכן המוודא יגלה אותו. אולם הניחוש של המוכיח כולל התחייבות על ערך הניחוש, ולאורך כל הפרוטוקול ההתחייבות הזו אינה נפתחת.

בחלק הראשי המוכיח יוכיח שהוא יודע להוכיח את הטענה המקורית **או** שאחד מניחושיו (שעליהם הוא רק התחייב, אך הוא לא פתח) היה נכון.

כל מה שסימולטור צריך בשביל לפתור את זה הוא "ניחוש" נכון אחד, כלומר החזרה אחת. הוא די חופשי לשחק עם המיקום של ההחזרה: רק צריך להזהר לא להחזיר אל מעבר להתחייבות בהתחלה.

להוכחה הזו יש את התכונה שהיא witness-indistingishable: אין אפשרות לדעת איזה עד היה למוכיח: האם העד של הטענה המקורית או שניחוש נכון של אחד ממספרי מוודא. לגבי התכונה הזו ניתן להוכיח שהיא משתמרת גם בהרכבות אסינכרוניות של הפרוטוקול (בניגוד לתכונת אפס המידע, שלא בהכרח משתמרת אף בהרכבה סדרתית).

לפעם השניה

הפרוטוקול המלא

d_s - השהיית המוודא בשיחה s. בהתחלה 1.

אם V בשיחה s לא עונה תוך פרק-זמן d_s אזי $d_s := 2d_s$ ומתחילים מחדש.

אבחנות

1. מוודא נכשל לכל היותר $O(\log(n))$ פעמים
2. הזמן של הפרוטוקול הבסיסי האחרון דומיננטי בסה"כ זמן הפרוטוקול. יש בו את כל $\tilde{O}(\log n)$ הסיבובים
3. שלמות: כמו בפרוטוקול המקורי
4. נאותות: סיכויי המוכיח לנחש עדיין זניחים

שקף 30

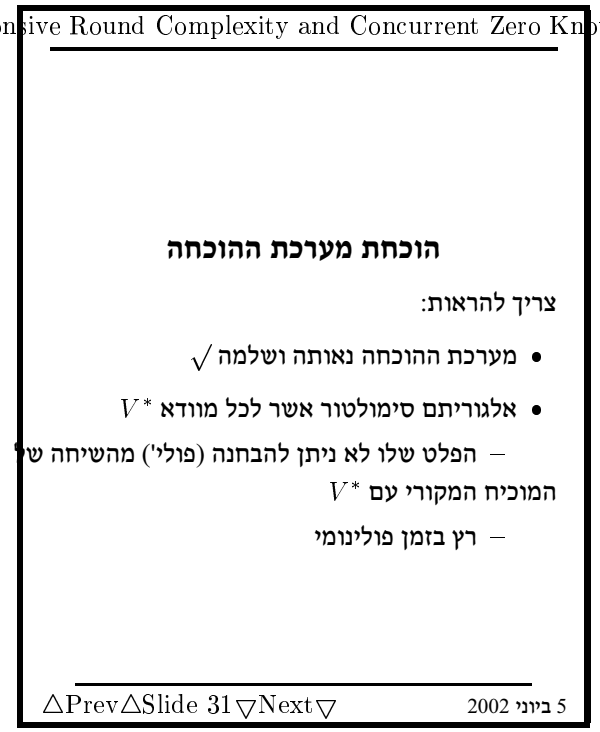
זמן נוכחי: 38:50 השקף הזה: 2:0

כאשר מתחילה שיחה s המוכיח מסמן $d_s = 1$. כדי לסמל שהזמן הצפוי לתשובה הוא יחידת זמן אחת. אם וכאשר מוכיח לא יענה תוך פרק זמן זה הוא המוכיח יכפיל את הערך הזה: כעת הוא יצפה לתגובה תוך פרק-זמן כפול, וכמוכן השיחה תתאפס: היא תתחיל מחדש להריץ את הפרוטוקול הבסיסי.

When a session s begins, the prover assigns $d_s = 1$ to signify that it expects an answer from the verifier within time d_s. Whenever the verifier fails to respond within d_s, the prover sets $d_s = 2 \cdot d_s$ and starts the basic protocol from the beginning.

Observations

1. The number of times a verifier fails is at most logarithmic, because a protocol is restarted only after a delay of d_s, and at each restart d_s is doubled. After $O(\log n)$ restarts the length of d_s is $2^{O(n)}$.
2. In the last level the basic protocol is performed to the end: all of the $\tilde{O}(\log n)$ rounds. Any previous level has no more rounds perfomred. Furthermore, the length of each round is at most half of the length of the final round.
3. Completeness like in original protocol.
4. Soundness: it is negligible in the original protocol and we run at most $O(\log n)$ independent tries.



שקף 31

זמן נוכחי: 40:50	השקף הזה: 1:30

ראינו שמערכת ההוכחה נאותה ושלמה. נשאר להוכיח את תכונת אפס-המידע.
כלומר צריך להראות סימולטור אשר בהינתן מכונת מוודא, V^* ידע ליצור פלט שנראה בדיוק כמו שיחה בין V^* למוכיח האמיתי.

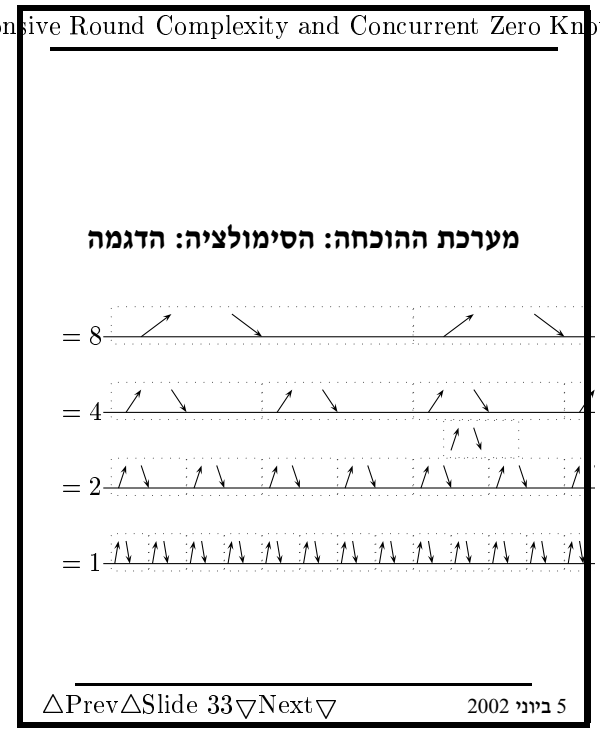
שקף 32

מערכת ההוכחה: הסימולציה

על הסימולטור מוטל למצוא לפחות ערך v_i אחד מהערכים $v_1 \ldots v_m$ של כל מוודא. מבוצע ע"י פעולות החזרה.

- לוח זמנים *[schedule]* קבוע
- רקורסיבי: טפל בשלבים נמוכים, בצע החזרה, חזור לטפל בשלבים נמוכים

שקף 33



זמן נוכחי: 43:20	השקף הזה: 3:0

בשרטוט זה אנו רואים את המערכת שאיתה צריך להתמודד הסימולטור. אנו רואים כאן ארבע שיחות עם ארבעה ערכים שונים של d_s. לכל ערך כזה קראנו **רמה**. כל מסגרת כזו מראה סיבוב אחד של הפרוטוקול. בתחילת הסיבוב המוכיח שולח את ההתחיבות על הניחוש של v_i. תוך פרק זמן של d_s המוודא צריך להחזיר את הערך של v_i. ההודעה הבאה של המוכיח תהיה רק כעבור $2d_s$ מהודעתו הקודמת.

למה העיכוב הזה? נסתכל על סיבוב ששייך לשיחה אחרת מאותו שלב. נשים לב שאפשר לעשות החזרה אחת שתפתור את את שניהם בייחד. באופן כללי התזמון של הסימולטור הזה הוא קבוע. הוא מניח שבמקרה הגרוע תמיד יבואו אליו הרבה מוודאים. התזמון מאפשר לו לסדר את כל המוודאים מאותו שלב בסדר ולטפל בהם בקבוצות.

אבל מה קורה בין השלבים השונים?

הסימולטור מטפל בשלבים בצורה רקורסיבית: מטפל בחצי הראשון, מבצע החזרה, ומטפל בחצי השני. מכיוון שמספר שלבים הוא לכל היותר לוריתמי, לא צפויות צרות מכיוון של תוקפים מרושעים שינסו להפעיל כל מיני דברים כמו תזמון פירמידה.

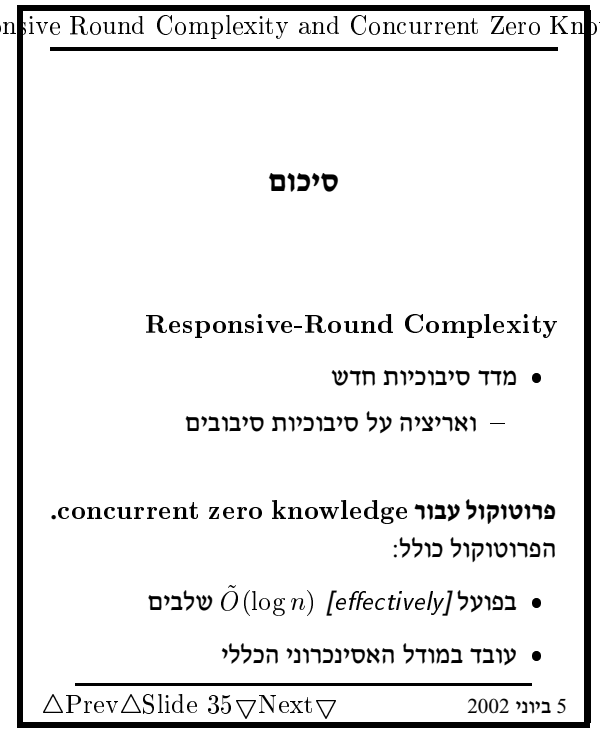
שקף 34

הערות

- שימו לב להבדל בהנחות התזמון בין עבודה זו לבין [Dwork-Naor-Sahai]: אין הגבלה על השהיית הרשת
- סיבוכיות מספר הסיבובים (במקרה הגרוע) היא $\tilde{O}(\log^2 n)$, אולם סיבוכיות זמן תגובה היא $\tilde{O}(\log n)$

Note the difference from timing-assumptions of [Dwork-Naor-Sahai]: In this protocol zero-knowledge is guaranteed even for unbounded network delay.

Round complexity (worst case) is still $\tilde{O}(\log^2 n)$ (as in the upper bound of [Kilian-Petrank]). but responsive round complexity is $\tilde{O}(\log n)$.



שקף 35

זמן נוכחי: 47:5	השקף הזה: 0:45

סיכום ראינו מדד סיבוכיות חדש: responsive-round copmplexity, כלומר סיבוכיות זמן תגובה. זוהי ואריאציה של סיבוכיות מספר הסיבובים אשר מתחשבת גם באורך כל סיבוב, ולכן יכולה לתת הערכ הטובה יותר של סיבוכיות הזמן של הפרוטוקול.

כמוכן ראינו פרוטוקול להוכחות באפס מידע בסביבה אסינכרונית (concurrent zero knowledge proofs) אשר דורש בפועל *[איך לתרגם כאן effectively?]* $\tilde{O}(\log n)$ סיבובים ולוא דורש שום timing assumptions או settings assumptions

References

מקורות

[Barak] Barak, B. (2001). How to get beyond black-box simulator barrier. In *Proceedings of FOCS 2001.*

[Canetti-Goldreich-Goldwasser-Micali] Canetti, R., Goldreich, O., Goldwasser, S., and Micali, S. (1999). Resettable zero-knowledge. Record 99-22, Theory of Cryptography Library. received October 25th, 1999. Supercedes Theory of Cryptography Library Record 99-15.

[Canetti-Goldreich-Goldwasser-Micali] Canetti, R., Goldreich, O., Goldwasser, S., and Micali, S. (2000). Resettable zero-knowledge (extended abstract). In ACM, editor, *Proceedings of the thirty second annual ACM Symposium on Theory of Computing: Portland, Oregon, May 21–23, [2000]*, pages 235–244, New York, NY, USA. ACM Press. see also [Canetti-Goldreich-Goldwasser-Micali].

[Canetti-Killian-Petrank-Rosen] Canetti, R., Killian, J., Petrank, E., and Rosen, A. (2001). Concurrent zero-knowledge requires $\tilde{\Omega}(\log n)$ rounds. In *Proceedings of the thirty third annual ACM Symposium on Theory of Computing.* ACM Press.

[Damgård] Damgård, I. B. (2000). Efficient concurrent zero-knowledge in the auxiliary string model. In Pre-

Slide 36

Slide 37

Slide 38

neel, B., editor, *Advances in Cryptology – EUROCRYPT ' 2000*, Lecture Notes in Computer Science, pages 418–430, Brugge, Belgium. Springer-Verlag, Berlin Germany.

[Dwork-Naor-Sahai] Dwork, C., Naor, M., and Sahai, A. (1998). Concurrent zero-knowledge. In ACM, editor, *Proceedings of the thirtieth annual ACM Symposium on Theory of Computing: Dallas, Texas, May 23–26, 1998*, pages 409–418, New York, NY, USA. ACM Press.

[Goldreich-Micali-Wigderson] Goldreich, O., Micali, S., and Wigderson, A. (1991). Proofs that yield nothing but their validity or all languages in NP have zero-knowledge

Slide 39

proof systems. *JACM*, 38(3):690–728.

[Goldreich-Oren] Goldreich, O. and Oren, Y. (1994). Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32.

[Goldwasser-Micali-Rackoff] Goldwasser, S., Micali, S., and Rackoff, C. (1989). The knowledge complexity of interactive systems. *SIAM Journal of Computing*, 18(1):186–208.

[Kilian-Petrank] Kilian, J. and Petrank, E. (2001). Concurrent zero-knowledge in poly-logarithmic rounds. In *Proceedings of the thirty third annual ACM Symposium on*

Theory of Computing. ACM Press.

[Richardson-Kilian] Richardson, R. and Kilian, J. (1999). On the concurrent composition of zero-knowledge proofs. *Lecture Notes in Computer Science*, 1592:415–431.