Εισαγωγή στα συστήματα αποδείξεων μηδενικής γνώσης

Δημήτρης Βυτινιώτης
dvitin@softlab.ntua.gr

19 Ιουνίου 2003

1 Διαλογικά συστήματα αποδείξεων

Τα διαλογικά συστήματα αποδείξεων (*interactive proof systems*) είναι συστήματα με δύο εμπλεκόμενα μέρη, τον *prover* P και τον *verifier* V. Ο P προσπαθεί, μέσω του διαλογικού πρωτοκόλλου, να αποδείξει στον V ότι ξέρει τη λύση σε κάποιο πρόβλημα.

Έστω ότι έχουμε μια γλώσσα L κάποιου προβλήματος, $L \subseteq \{0,1\}^*$, π.χ. τη γλώσσα του 3-SAT, και έστω ότι ο P γνωρίζει για ένα x, ότι $x \in L$ και θέλει να αποδείξει και στον V ότι $x \in L$. Τα δύο μέρη εμπλέκονται σε μια διαλογική διαδικασία, που μπορεί να έχει πολλούς γύρους. Σε κάθε γύρο ανταλλάσονται μια σειρά απο μηνύματα. Στο τέλος ο V αποδέχεται ή απορρίπτει την απόδειξη με βάση κάποια κριτήρια. Θα συμβολίζουμε ένα τέτοιο πρωτόκολλο μεταξύ P και V ως (P, V).

Σε αυτά τα πρωτόκολλα, το υπολογιστικό μοντέλο είναι το ζευγάρι αλληλοδραστικών μηχανών Turing (*pair of interactive Turing Machines*) [11]. Το μοντέλο αυτό φαίνεται στην Εικόνα 1. Θεωρούμε ότι ο P έχει απεριόριστες υπολογιστικές δυνατότητες, ενώ ο V έχει δυνατότητες ενός probabilistic polytime αλγορίθμου. Έτσι έχει νόημα ο P να θέλει να αποδείξει κάτι, που μπορεί να υπολογίσει μόνο αυτός, αλλά όχι ο V, στον "αδύναμο" υπολογιστικά V.

Ένα interactive protocol, για να χαρακτηριστεί ως interactive proof system, δηλαδή να έχει αποδεικτικό χαρακτήρα, θα πρέπει να ικανοποιεί δύο συνθήκες:

- **Completeness:** $\forall x \in L$, είσοδο στο (P, V), $prob(V\ accepts) \geq 2/3$
- **Soundness:** $\forall x \notin L$, για κάθε "ανέντιμο" prover P', αν το (P', V) τρέξει με είσοδο το x, τότε ισχύει ότι $prob(V\ accepts) \leq 1/3$

Η πρώτη συνθήκη λέει ότι ο V "πείθεται" με πολύ μεγάλη πιθανότητα για κάθε yes instance του προβλήματος. Η δεύτερη συνθήκη λέει ότι, αν ένας "ανέντιμος" P', προσπαθήσει να πείσει τον V για ένα no instance του προβλήματος, η πιθανότητα να πειστεί ο V είναι πολύ μικρή[1]. Οι γλώσσες που δέχονται τέτοιες αποδείξεις ανήκουν στην κλάση πολυπλοκότητας **IP**. Ισχύουν τα ακόλουθα:

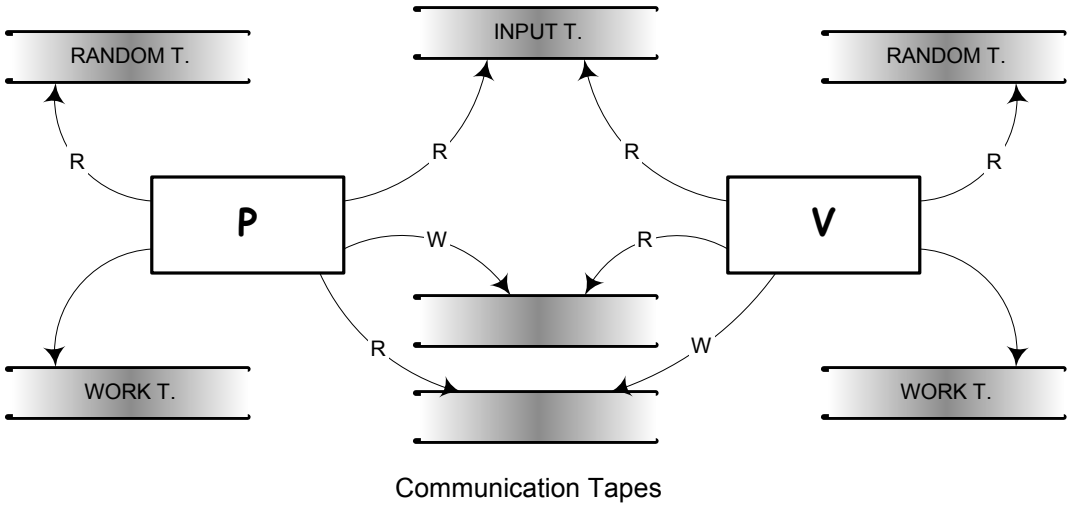
- $NP \subseteq IP$, γιατί ο P μπορεί να στείλει στον V το computation που αντιστοιχεί στο πρόβλημα, π.χ. ένα assignment που ικανοποιεί το 3-SAT, και ο V έχει την υπολογιστική δυνατότητα να το ελέγξει. Ελέγχεται πολυωνυμικά, αφού η γλώσσα είναι NP, και ο V δεν χρειάζεται καν να χρησιμοποιήσει randomization.
- $BPP \subseteq IP$, γιατί ο V, χωρίς κανένα interaction με τον P, έχει τη δυνατότητα να αποφανθεί για τα στοιχεία μιας γλώσσας στο BPP.

2 Zero Knowledge

Μας ενδιαφέρει, η πληροφορία την οποία αποκτά ο V μετά το τέλος του πρωτοκόλλου, να του είναι άχρηστη για υπολογισμούς οι οποίοι υπερβαίνουν τις δυνατότητές του.

[1] Οι τιμές $2/3$ και $1/3$ παραπάνω δεν έχουν ιδιαίτερη σημασία, γιατί τρέχοντας τα πρωτόκολλα αυτά διαφορετικό αριθμό επαναλήψεων αλλά πολυωνυμικό ως προς το μέγεθος της εισόδου, μπορεί κανείς να φράξει την πιθανότητα λάθους πολυωνυμικά όσο θέλει. Στα αρχικά άρθρα μάλιστα αυτές οι πιθανότητες λάθους είναι φραγμένες από $1/|x|^k$.

Εικόνα 1: Υπολογιστικό μοντέλο αλληλοδραστικών πρωτοκόλλων.



Έστω ένα IP σύστημα (P, V) για μια γλώσσα L. Θα λέμε, άτυπα, ότι είναι ένα σύστημα *μηδενικής γνώσης (zero knowledge)* αν, ό,τι μπορεί να υπολογιστεί αποδοτικά από τον V, μετά την αλληλεπίδραση με τον P, μπορούσε επίσης να υπολογιστεί αποδοτικά χωρίς τη χρήση του πρωτοκόλλου, έχοντας μόνο την κοινή είσοδο.

Για παράδειγμα ένα IP πρωτόκολλο, το οποίο αποδεικνύει μια NP γλώσσα με τον τρόπο που αναφέρθηκε παραπάνω, δεν είναι μηδενικής γνώσης. Γιατί ο V μπορεί να χρησιμοποιήσει την πληροφορία που του έστειλε ο P – ένα computation – για να λύσει στο μέλλον το NP πρόβλημα αυτό.

Μιας και ο V, αντλεί τις πληροφορίες του από τα δεδομένα που βλέπει πάνω στις ταινίες επικοινωνίας, την αρχική είσοδο, καθώς και τα coin tosses του, μας ενδιαφέρει η "εικόνα" του πρωτοκόλλου για τον V. Το σύνολο όλων των μηνυμάτων που ανταλλάσσονται κατά τη διάρκεια του πρωτοκόλλου με είσοδο x, συμπεριλαμβανομένων των coin tosses του V και του x, θα αντιπροσωπεύεται από μια τυχαία μεταβλητή που ακολουθεί κατανομή πιθανότητας $VIEW_V^P(x)$. Το σύνολο των μηνυμάτων αυτών, της εισόδου και των coin tosses του V θα ονομάζουμε *transcript* του πρωτοκόλλου.

Στόχος είναι αυτό το transcript να μην αποκαλύπτει τίποτε παραπάνω παρά την εγκυρότητα της απόδειξης. Δηλαδή ένας randomized polytime αλγόριθμος M (*simulator*), θα πρέπει να μπορεί μόνος του να δημιουργεί transcripts, τα οποία θα "μοιάζουν" σαν πραγματικά. Για μια δεδομένη είσοδο x, το $M(x)$ θα εκφράζει την κατανομή πιθανότητας που ακολουθούν τα transcripts που παράγει ο simulator.

Ο simulator μπορεί να παράγει transcripts τα οποία είναι αρκετά όμοια με τα πραγματικά. Η "διαφορά" τους καθορίζει το ποσό της *γνώσης* που διακινείται από το πρωτόκολλο και ορίζει το *knowledge complexity* του πρωτοκόλλου. Τα zero-knowledge πρωτόκολλα έχουν μηδενικό knowledge complexity.

Συνοψίζοντας, τα εμπλεκόμενα μέρη μπορούν να "λοξοδρομήσουν" από το πρωτόκολλο ως εξής:

1. Ένας P' προσπαθεί να πείσει τον V για ένα false instance. Αυτή η περίπτωση αποκλείεται εξαιτίας του soundness.

2. Ένας V^*, προσπαθεί να αυξήσει τις υπολογιστικές του δυνατότητες, αλληλεπιδρώντας με τον P. Αυτή η περίπτωση καλύπτεται λόγω του zero knowledge.

3 Perfect Zero Knowledge

Θα παρουσιάσουμε ένα παράδειγμα μιας τέτοιας απόδειξης για το πρόβλημα *GRAPH ISOMORPHISM*. Το πρόβλημα είναι: Έστω δύο γράφοι n κόμβων, $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$. Υπάρχει συνάρτηση f, 1-1 και επί, τ.ώ. για κάθε $\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$; Μια απόδειξη μηδενικής γνώσης για αυτό φαίνεται στον Πίνακα 1.

input: G_1, G_2 on vertex set $\{1, \ldots, n\}$
repeat n times

1. P chooses **random permutation** π of $\{1, \ldots, n\}$, computes $H := \pi(G_1)$ and sends H to V.
2. V chooses random integer $i \in \{1, 2\}$ and sends it to P.
3. P computes: **if** $i = 1$ **then** $\rho := \pi$ **else** $\rho := \pi * \sigma$, where $\sigma(G_2) = G_1$.
4. V checks if $H = \rho(G_1)$.

V accepts if last step accepts in each of the n rounds.

Πίνακας 1: ZK απόδειξη για το GI

Το παραπάνω είναι IP πρωτόκολλο. Για το completeness, φαίνεται ότι για κάθε $x \in L$, ο V αποδέχεται με πιθανότητα 1. Για το soundness, αν ο G_1 δεν είναι ισομορφικός με τον G_2, δηλαδή $x \notin L$, ο μόνος τρόπος, ένας "ανέντιμος" P', να πείσει τον V, είναι να μάντευε σωστά κάθε φορά τα coin tosses του V, και να έστελνε, αντί για H, ισομορφικά αντίγραφα του εκάστοτε γράφου. Αυτό μπορεί να γίνει με πιθανότητα $1/2$ σε κάθε γύρο και άρα για n γύρους είναι $1/2^n$.

Είναι το παραπάνω zero knowledge; Αρκεί να βρούμε έναν simulator που μπορεί να παράγει παρόμοια transcripts με το πρωτόκολλο. Τα transcripts του συγκεκριμένου πρωτοκόλλου μπορεί να θεωρηθούν ότι έχουν τη μορφή: $T = ((G_1, G_2); (H_1, i_1, \rho_1); \ldots (H_n, i_n, \rho_n))$. Ο αλγόριθμος που ακολουθεί στον Πίνακα 2 [2] μπορεί να αναπαράγει τα transcripts του πρωτοκόλλου με την ίδια ακριβώς κατανομή πιθανότητας.

input: isomorhic G_1, G_2 on vertex set $\{1, \ldots, n\}$
initialize transcript $T := (G_1, G_2)$.
for $i := 1$ **to** n do

1. choose random $i_j \in \{1, 2\}$.
2. choose ρ_j random permutation on vertex set.
3. compute $H_j = \rho_j(G_{i_j})$.
4. concatenate to T the tuple (H_j, i_j, ρ_j)

Πίνακας 2: Simulator για "έντιμο" V

Ορισμός 1 (Perfect Zero Knowledge για "έντιμο" V) *Έστω ένα IP πρωτόκολλο (P, V) και είσοδος ένα yes instance x μιας γλώσσας L. Έστω M ένας randomized polytime simulator και $M(x)$ η κατανομή των transcripts που παράγει αυτός με είσοδο x. Το πρωτόκολλο είναι Perfect Zero Knowledge (PZK) για τον "έντιμο" V, αν $VIEW_V^P(x) \equiv M(x)$.*

Η παραπάνω απόδειξη **είναι PZK για τον "έντιμο"** V. Η κάθε τριάδα είναι ανεξάρτητο ενδεχόμενο από κάθε άλλη και επομένως αρκεί οι τριάδες να δημιουργούνται με την ίδια πιθανότητα. (i) Στον simulator έχω $1/2$ πιθανότητα στο πρώτο βήμα και $1/n!$ στο δεύτερο, και επομένως η πιθανότητα για κάθε transcript είναι $1/2n!$. (ii) Στο πρωτόκολλο ο P επιλέγει με πιθανότητα $1/n!$ ένα permutation, ο V επιλέγει με πιθανότητα $1/2$ έναν ακέραιο και άρα ανά τριάδα έχουμε πάλι $1/2n!$, και άρα και συνολικά: $VIEW_V^P(x) \equiv M(x)$.

Αλλά τι γίνεται αν ο V δεν ακολουθήσει το πρωτόκολλο; Μήπως μπορεί να μάθει κάτι παραπάνω; Στην περίπτωσή του GI π.χ. μπορεί να διαλέγει τους ακέραιους αριθμούς όχι τυχαία αλλά με κάποιο μη ομοιόμορφο τρόπο.

Ορισμός 2 (Perfect Zero Knowledge) *Έστω πάλι (P, V) για τη γλώσσα L, και τρέχει σε yes instance x. Αν για κάθε V^*, υπάρχει **expected polytime** randomized αλγόριθμος $M^* = M^*(V^*)$, τ.ώ. $VIEW_{V^*}^P(x) \equiv M^*(x)$ τότε το πρωτόκολλο είναι perfect zero knowledge.*

Πώς μπορεί κανείς να κατασκευάσει έναν τέτοιο simulator; Αυτό μπορεί να γίνει "κλέβοντας" π.χ. το randomization του V^* [2]. Στον Πίνακα 3 φαίνεται αυτός ο simulator:

---

input: isomorhic G_1, G_2 on vertex set $\{1,\ldots,n\}$
initialize transcript $T := (G_1, G_2)$.
for $i := 1$ **to** n do

- $oldstate := state(V^*)$
- **repeat**
  1. choose random $i_j \in \{1,2\}$.
  2. choose ρ_j randomly, compute $H_j := \rho_j(G_{i_j})$.
  3. call V^* with input H_j obtaining challenge i'_j.
  4. **if** $i_j = i'_j$ **then** concatenate to T the tuple (H_j, i_j, ρ_j) **else** $state(V^*) := oldstate$
- **until** $i_j = j'_j$

Πίνακας 3: Simulator για κάθε V^*

Μπορεί κανείς να παρατηρήσει ότι ο παραπάνω είναι ένας expected polytime αλγόριθμος, γιατί μαντεύοντας ακέραιους θα πετυχαίνει το σωστό κάθε 2 επαναλήψεις. Η απόδειξη που δόθηκε για το GI **είναι PZK για κάθε** V^*. Η απόδειξη είναι επαγωγική πάνω στον αριθμό των γύρων. (i) Αν δεν τρέξει κανένας γύρος τότε τα transcripts έχουν μόνο τους γράφους, άρα έχουμε ίδια πιθανότητα. (ii) Έστω ότι για όλους τους γύρους πριν το γύρο j, οι πιθανότητες των κανονικών και των "πλαστών" transcripts ταυτίζονται. (iii) Στον j γύρο ο κανονικός αλγόριθμος επιλέγει στο πρώτο βήμα permutation με πιθανότητα $1/n!$ και στο επόμενο ο V^* επιλέγει έστω 1 με πιθανότητα p_1, 2 με πιθανότητα $1-p_1$. Δηλαδή $prob(T_j = (H,1,\rho)) = p_1/n!$. Ο simulator τώρα σε κάθε repeat υπολογίζει ένα permutation με πιθανότητα $1/n!$ και μαντεύει με πιθανότητα $1/2$ να είναι σωστός, έναν αριθμό. Η πιθανότητα να γραφτεί ένα $(H,1,\rho)$ στον j γύρο, στο πρώτο repeat είναι $p_1/2n!$, η πιθανότητα να γραφτεί ένα $(H,1,\rho)$ στο δεύτερο repeat είναι $p_1/2^2n!$ κ.ο.κ. Δηλαδή η πιθανότητα στον j γύρο να έχουμε τριάδα $(H,1,\rho)$ είναι $(1+1/2+...)p_1/2n! = p1/n!$. Όμοια για τριάδες $(H,2,\rho)$. Άρα ταυτίζονται οι πιθανότητες και λόγω επαγωγής όσο και να τρέξουν οι αλγόριθμοι, τα δύο transcripts θα έχουν την ίδια κατανομή, δηλαδή $VIEW^P_{V^*}(x) \equiv M^*(x)$.

Ένα άλλο παράδειγμα είναι το *QUADRADIC RESIDUOSITY*. Με δεδομένο έναν ακέραιο $n = pq$, όπου p,q πρώτοι, αλλά χωρίς να δίνονται οι p,q (άγνωστη παραγοντοποίηση), και ένα x, είναι αυτό τετραγωνικό υπόλοιπο του n; Στον Πίνακα 4 φαίνεται μια PZK απόδειξη για το πρόβλημα αυτό.

Το completeness του πρωτοκόλλου είναι trivial. Για το soundness μπορεί κανείς να παρατηρήσει ότι η πιθανότητα να ξεγελαστεί ο V είναι $1/2^{logn}$. Τέλος είναι zero knowledge. Τα transcripts του πρωτοκόλλου είναι της μορφής (y,i,z). Ο simulator θα επέλεγε ένα i[2], θα επέλεγε τυχαία ένα z, και θα υπολόγιζε $y := z^2x^{-i} mod n$.

4 Άλλα είδη Zero Knowledge

Είδαμε παραπάνω ότι τα transcripts τα οποία παράγει το πρωτόκολλο και ο simulator έχουν την ίδια ακριβώς κατανομή για δεδομένη είσοδο. Τώρα μας ενδιαφέρει να εξετάσουμε περιπτώσεις που οι κατανομές αυτές είναι αρκετά "κοντά" για κάποιον "κριτή".

[2]Είτε τυχαία είτε εξομοιώνοντας το randomization κάποιου V^*

input: n with inknown factorization, $x \in QR(n)$.
repeat *logn* times

- P chooses random $v \in \mathbb{Z}_n^*$, and sends $y := v^2 mod n$ to V.
- V chooses random $i \in \{0,1\}$ and sends i to P.
- P computes $z := u^i v mod n$, where $x = u^2 mod n$.
- V checks that $z^2 \equiv x^i y (mod n)$

V accepts if last step accepts in each of the *logn* rounds.

Πίνακας 4: PZK για το QR

4.1 Σύνολα κατανομών και βαθμοί διακρισιμότητας

Ορισμός 3 (Κατανομή πιθανότητας) *Μια κατανομή πιθανότητας (probability distribution) θα είναι μια συνάρτηση* $\pi : \{0,1\}^* \to (0,1)$, *τ.ώ.* $\sum_{\forall a} \pi(a) = 1$.

Ορισμός 4 (Σύνολο κατανομών με δείκτες στοιχεία μιας γλώσσας) *Έστω* I *μια γλώσσα. Τότε το σύνολο* $\Pi = \{\pi_i\}_{i \in I}$ *θα είναι ένα σύνολο κατανομών (distribution ensemble) με δείκτες τα στοιχεία της γλώσσας* I.

Για παράδειγμα έστω ότι έχουμε μια PPT μηχανή Turing M για γλώσσα L, που για είσοδο x παράγει κατανομή εξόδου $M(x)$. Τότε μπορούμε να ορίσουμε το $\{M(x)\}_{x \in L}$, που θα είναι η ακολουθία όλων των κατανομών που παράγονται για τα yes instances της εισόδου. Επίσης έστω ένα Interactive Protocol (P,V), που για ένα yes instance x της γλώσσας L, δίνει κατανομή $VIEW_V^P(x)$ στα transcripts. Όμοια μπορεί να οριστεί το $\{VIEW_V^P(x)\}_{x \in L}$.

Έστω ότι έχουμε δύο τέτοια σύνολα κατανομών για την ίδια γλώσσα I. Αυτό μπορεί να αντιστοιχεί σε δύο διαφορετικούς αλγόριθμους, όπως ακριβώς στην περίπτωσή μας για τον simulator και το (P,V). Ας πούμε τα δύο αυτά σύνολα

$$\Pi_1 = \{\pi_{1,i}\}_{i \in I}, \Pi_2 = \{\pi_{2,i}\}_{i \in I}$$

Μπορούμε επίσης να θεωρήσουμε έναν αλγόριθμο $\mathcal{A}$, ο οποίος θα ονομάζεται κριτής και ο οποίος έστω ότι μπορεί να έχει έξοδο 1, αν πάρει ένα στοιχείο που ανήκει σε μια κατανομή π_i και το i. Τότε ορίζουμε την παρακάτω πιθανότητα:

$$p_1^{\mathcal{A}}(i) = \sum_a \pi_{1,i}(a) \times prob(\mathcal{A}(i,a) = 1)$$

Η παραπάνω πιθανότητα είναι η πιθανότητα να έχει έξοδο 1 ο αλγόριθμος, αν πάρει δείγματα από την πρώτη κατανομή. Ομοίως ορίζουμε την πιθανότητα ο αλγόριθμος να φέρει πάλι 1, δηλαδή **να συμπεριφερθεί με τον ίδιο τρόπο** αν πάρει δείγματα από τη δεύτερη κατανομή ως εξής:

$$p_2^{\mathcal{A}}(i) = \sum_a \pi_{2,i}(a) \times prob(\mathcal{A}(i,a) = 1)$$

Στις παραπάνω σχέσεις τα αθροίσματα είναι πάνω σε δείγματα πολυωνυμικού μήκους ως προς το μήκος του εκάστοτε input i του αλγόριθμου. Τώρα εξετάζουμε αν αυτές οι μέσες τιμές του αλγόριθμου για την κάθε κατανομη διαφέρουν αρκετά.

Ορισμός 5 (Computationally Indistinguishable) *Τα σύνολα* Π_1 *και* Π_2 *θα λέγονται computationally indistinguishable αν:* $\forall$ *randomized polytime* $\mathcal{A}$, $\forall c > 0$, $\forall i$ *αρκετά μεγάλο ισχύει:* $|p_1^{\mathcal{A}}(i) - p_1^{\mathcal{A}}(i)| < 1/|i|^c$

Πρακτικά αυτό σημαίνει ότι κανένας πολυωνυμικός αλγόριθμος δεν μπορεί να αποφασίσει (δηλ. να διαφοροποιήσει τη συμπεριφορά του) αν πάρει δείγματα από τον έναν αλγόριθμο ή τον άλλο, ξέροντας την είσοδο που έδωσε τις δύο κατανομές. Αν ο αλγόριθμος κριτής $\mathcal{A}$ έχει τη δυνατότητα να κάνει μη πολυωνυμικούς υπολογισμούς, τότε έχουμε τα εξής:

Ορισμός 6 (Statistically Indistinguishable) *Τα σύνολα* Π_1 *και* Π_2 *θα λέγονται statistically indistinguishable αν:* $\forall c > 0$, $\forall i$ *αρκετά μεγάλο,* $\sum_a |\pi_{1,i}(a) - \pi_{2,i}(a)| < 1/|i|^c$. *όπου και πάλι τα αθροίσματα παίρνονται πάνω σε δείγματα πολυωνυμικού μεγέθους.*

Αυτό σημαίνει ότι οι δύο κατανομές δίνουν ίδια "μάζα πιθανότητας" σε πολυωνυμικά φραγμένα υποσύνολα του χώρου δειγμάτων και επομένως, όσο ισχυρός και να ήταν ο αλγόριθμος δεν θα μπορούσε παίρνοντας πολυωνυμικά δείγματα να αποφανθεί για την προέλευσή τους.

Ορισμός 7 (Perfectly Indistinguishable) *Τα παραπάνω σύνολα θα λέγονται perfectly indistinguishable αν για κάθε είσοδο* i, *οι κατανομές* $\pi_{1,i}$ *και* $\pi_{2,i}$ *ταυτίζονται.*

Για να ελέγξει κανείς το τελευταίο χρειάζεται να έχει κάποιον αλγόριθμο ο οποίος θα μπορεί να τρέξει πάνω σε μη πολυωνυμικά δείγματα παίρνοντας αναπόφευκτα μη πολυωνυμικό χρόνο ως προς το μέγεθος της εισόδου.

4.2 Ορισμός zero knowledge

Ορισμός 8 (Zero Knowledge) *Ενα αλληλοδραστικό πρωτόκολλο* (P, V) *είναι perfect/ statistical/ computational zero knowledge για μια γλώσσα* L, *αν για κάθε PPT* V^* *υπάρχει ένας expected PPT αλγόριθμος* $\mathcal{M}$ *(simulator), τ.ώ. τα σύνολα κατανομών* $\{VIEW^P_{V^*}(x)\}_{x\in L}$ *και* $\{\mathcal{M}(x)\}_{x\in L}$ *να είναι perfect/ statistically/ computationally indistinguishable.*

Πρακτικά ενδιαφέρει να εξασφαλίζεται το computational zero knowledge, το οποίο αναφέρεται και απλά ως zero knowledge, δηλαδή να υπάρχει simulator του οποίου τα transcripts να μην μπορούμε να τα διακρίνουμε από τα αυθεντικά χρησιμοποιώντας πολυωνυμικό αλγόριθμο. Το statistical zero knowledge μας εξασφαλίζει απόλυτα, μιας και τα πρωτόκολλά μας παράγουν πολυωνυμικά transcripts. Το perfect zero knowledge έχει μόνο θεωρητική σημασία. Επιπλέον παρατηρεί κανείς ότι ο ορισμός για το perfect zero knowledge είναι ταυτόσημος με αυτόν που δόθηκε σε προηγούμενη ενότητα. Οι γλώσσες που έχουν perfect/ statistical/ computational zero knowledge αποδείξεις ανήκουν αντίστοιχα στις κλάσεις PZK, SZK και CZK ή απλά ZK. Ισχύει:

$$BPP \subseteq PZK \subseteq SZK \subseteq CZK \subseteq IP$$

5 Zero-Knowledge αποδείξεις για κάθε NP σύνολο

Είδαμε σε προηγούμενες ενότητες ότι $GI \in \mathcal{PZK}$. Αλλά δεν είναι γνωστό αν το *GRAPH ISOMORHISM* ανήκει στο NP. Θα μας ενδιέφερε να εξετάσουμε αν μπορούμε να βρούμε zero knowledge αποδείξεις (οποιουδήποτε είδους) για γλώσσες στο NP. Η ιδέα είναι να κατασκευαστεί ένα τέτοιο σύστημα για ένα NP-complete πρόβλημα. Πρώτα όμως χρειάζεται μια συζήτηση πάνω στην κρυπτογράφηση και τα commitment schemes.

5.1 Κρυπτογράφηση και commitment schemes

Ορισμός 9 (One way functions (strong)) *Μια συνάρτηση* $f : \{0,1\}^* \to \{0,1\}^*$ *λέγεται one-way [1] αν:*

1. $\forall x \in \{0,1\}^*, |x|^{1/k} \leq |f(x)| \leq |x|^k$ *για κάποιο* k, *δηλαδή η* f *δεν συμπιέζει την είσοδό της περισσότερο από πολυωνυμικά.*
2. $\exists$ *polytime TM, τ.ώ. με είσοδο* x *υπολογίζει το* $f(x)$, *δηλαδή η συνάρτηση υπολογίζεται αποδοτικά.*
3. $\forall PPT$ *Turing Machine* M', $\forall c > 0$, $\forall x$ *αρκετά μεγάλο ισχύει:*

$$prob(M'(f(x)) \in f^{-1}(f(x))) < 1/|x|^c$$

Δηλαδή η συνάρτηση δεν αντιστρέφεται αποδοτικά.

Οι συναρτήσεις αυτές[3] έχουν την ιδιότητα επομένως, ότι όταν σταλεί η υπολογισμένη τιμή της συνάρτησης είναι υπολογιστικά πολύ δύσκολο να ανακτήσει κανείς από την τιμή αυτή, το όρισμα της συνάρτησης (preimage). Παρόλα αυτά είναι πολύ συχνά δυνατό να ανακτήσει κανείς **μέρος** του preimage. Για παράδειγμα έστω ότι έχουμε τη συνάρτηση $f'(x,y) = (f(x),y)$, όπου η f είναι one-way και $|x| = |y|$. Τότε και η f' είναι one-way, αλλά τα μισα τελευταία bits του preimage της περνούν όπως είναι, κάνοντας έτσι δυνατή την μερική ανάκτηση του preimage. Μας ενδιαφέρει δηλαδή το μέρος του preimage το οποίο **δεν μπορεί να ανακτηθεί εύκολα**.

Ορισμός 10 (Hardcore predicates) *Έστω $b : \{0,1\}^* \to \{0,1\}$. Το b θα λέγεται hardcore predicate (ή hardcore bit) για μια OWF f αν [3]:*

1. *Το $b(x)$ μπορεί να υπολογιστεί πολυωνυμικά.*
2. *$\forall$ $PPTA$, $\exists\epsilon$ αμελητέα[4] συνάρτηση, τ.ώ. $prob(A(f(x)) = b(x)) \leq 1/2 + \epsilon(|x|)$.*

Δηλαδή με δεδομένη μια τιμή της f, το hardcore bit της είναι μια τιμή που δεν μπορούμε να υπολογίσουμε αποδοτικά με καλή πιθανότητα. Η πιθανότητα να βρούμε το bit αυτό από την υπολογισμένη τιμή της f θα είναι πάντοτε όχι πολύ πιο ψηλή από $1/2$.

Το ερώτημα είναι: πώς μπορούμε να κατασκευάσουμε hardcore predicates έχοντας OWF; Μπρούμε να χρησιμοποιήσουμε το εξής θεώρημα:

Θεώρημα 1 (Goldreich-Levin) *Έστω f OWF. Τότε και η $f'(x,r) = (f(x),r)$ είναι OWF ($|x| = |r|$). Η συνάρτηση $b(x,r) = \langle x,r\rangle = \oplus_{i|r_i=1}x_i$ είναι hardcore bit για την f'.*

Επομένως αν έχουμε OWF, τότε μπορούμε να κατασκευάζουμε harcore predicates, και θα δούμε τώρα ότι χρησιμοποιώντας τέτοια predicates μπορούμε να κωδικοποιούμε bits χωρίς υπάρχει κίνδυνος να αποκαλυφθούν από την τιμή της συνάρτησης.

Έστω ότι έχουμε μια OWF f, με γνωστό hardcore predicate b. Έστω ότι θέλουμε να κωδικοποιήσουμε ένα bit m. Επιλέγουμε τυχαία ένα $k \in \{0,1\}^k$ και υπολογίζουμε το:

$$(y,z) = (f(k), b(k) \oplus m)$$

Δηλαδή "κλειδώνουμε" το bit μας με την τιμή του hardcore predicate της f κάνοντας αδύνατη την αποκάλυψή του από την τιμή των y, z με πιθανότητα σημαντικά καλύτερη του $1/2$. Το παραπάνω είναι τυπικό παράδειγμα ενός bit commitment scheme, και βλέπουμε ότι στηρίζεται στην ύπαρξη OWF.

Ορισμός 11 (Bit Commitment Scheme) *Ένα σχήμα bit commitment μπορεί να μοντελοποιηθεί σαν ένας PPT αλγόριθμος $Commit(1^{|k|}, m, k)$ όπου m είναι bit και k ακέραιος[5] με δύο ιδιότητες:*

1. *$\forall m, m', \forall A$ PPT, $|prob\{A((Commit(1^{|k|}, m, k)) = 1\} - prob\{A(Commit(1^{|k'|}, m', k')) = 1\}| < \epsilon(k)$ όπου ϵ αμελητέα για οποιαδήποτε k,k' συνάρτηση. Η ιδιότητα αυτή ονομάζεται **concealing**.*
2. *$\nexists m' \neq m$ τ.ώ. $Commit(1^{|k|}, m, k) = Commit(1^{|k'|}, m', k')$ για οποιαδήπτε k, k'. Η ιδιότητα αυτή ονομάζεται **binding**.*

Ο παραπάνω ορισμός είναι ένας uniform ορισμός (βλέπε και παράρτημα για nonuniform complexity). Αυτό δυστυχώς για τους σκοπούς μας δεν αρκεί. Χρειαζόμαστε έναν nonuniform ορισμό για το concealing (ο λόγος φαίνεται στην απόδειξη για το computational zero knowledge στο παράρτημα για το G3C). Ένας nonuniform ορισμός για το concealing θα ήταν: $\forall$ polysize circuit family $C = \{C_n\}$:,

$$|prob\{C_n((Commit(1^{|k|}, m, k)) = 1\} - prob\{C_{n'}(Commit(1^{|k'|}, m', k')) = 1\}| < \epsilon(k)$$

[3] Οι συναρτήσεις αυτές σχετίζονται με την κλάση πολυπλοκότητας $\mathcal{UP}$. Έστω μια non-deteministic polynomial time Turing Machine (NDTM). Θα λέμε ότι είναι unambiguous αν για κάθε είσοδο υπάρχει το πολύ ένα computation που αποδέχεται την είσοδο. Ορίζουμε σαν $\mathcal{UP}$ το σύνολο των γλωσσών που αναγνωρίζονται από unambiguous NDTM. Ισχύει $\mathcal{P} \subseteq \mathcal{UP} \subseteq \mathcal{NP}$. Αποδεικνύεται ότι $UP \equiv P$ ανν δεν υπάρχουν one-way συναρτήσεις[1].

[4] $\forall c > 0$, $\exists n_0$, τ.ώ. $\forall n > n_0$, $\epsilon(n) < 1/n^c$.

[5] Το $1^{|k|}$ απλώς σηματοδοτεί ότι ο αλγόριθμος τρέχει σίγουρα πολυωνυμικά ως προς το k

όπου n, n' το μέγεθος του output του αλγόριθμου Commit. Αντίστοιχα ο προηγούμενος ορισμός για τις OWF μπορεί να γίνει nonuniform. Παρακάτω θα χρησιμοποιούμε τους non-uniform ορισμούς.

Τα πρωτόκολλα αυτά περιλαμβάνουν δύο φάσεις μεταξύ δύο μερών. Στην πρώτη φάση (**commit**) το ένα μέρος εκτελεί τον αλγόριθμο και στέλνει στο άλλο μέρος το αποτέλεσμα. Η concealing ιδιότητα εξασφαλίζει ότι το άλλο μέρος δεν μπορεί σε εκείνη τη φάση να ανακτήσει κάποια πληροφορία για το bit που κωδικοποίησε το πρώτο μέλος. Στην επόμενη φάση (**reveal**) το πρώτο πάλι μέλος στέλνει όλες τις παραμέτρους που πέρασε στον αλγόριθμό Commit και ο άλλος επιβεβαιώνει υπολογίζοντας πάλι τον Commit ότι αυτές ήταν όντως οι τιμές που είχε χρησιμοποιήσει αρχικά ο πρώτος. Η ιδιότητα binding εξασφαλίζει ότι ο πρώτος δεν μπορεί να εξαπατήσει τον δεύτερο στη φάση αυτή στέλνοντας του άλλες παραμέτρους.

Παρατήρηση: το παραπάνω σχήμα που χρησιμοποιεί hardcore predicate είναι bit commitment scheme. Η φάση commit είναι ο υπολογισμός του ζεύγους και η αποστολή του στον άλλο, και η φάση reveal είναι ο επανυπολογισμός από τον δεύτερο του ζεύγους, όταν ο πρώτος του έχει στείλει τις παραμέτρους.

Μια άλλη τεχνική που μπορεί να χρησιμοποιηθεί σαν bit commitment scheme είναι το probabilistic encryption των Goldwasser και Micali [10]. Τα μέλη είναι ο A και ο B και τα δύο μέλη γνωρίζουν από την αρχή εναν ακέραιο n. Ο A θέλει να κωδικοποιεί bits b. Το πρωτόκολλο έχει ως εξής:

Ορισμός 12 (Probabilistic Encryption) *Οι δύο φάσεις του σχήματος είναι:*

- *Commit: Ο A υπολογίζει το $f(b,x) = m^b x^2 mod n$ όπου το m είναι pseudo-square modulo n. Στέλνει το αποτέλεσμα (καλείται blob) στον B.*
- *Reveal: Ο A στέλνει τα b, x και ο B υπολογίζει εκ νέου το blob, εξετάζοντας αν είναι το ίδιο.*

Το σχήμα αποδεικνύεται ότι είναι concealing και binding. Για το binding ειδικά, αν υποθέσουμε ότι δεν ήταν, θα υπήρχαν x_1, x_2, ώστε $mx_1^2 \equiv x_2^2 (mod n) \Rightarrow m = (x_2/x_1)^2 mod n$ που είναι άτοπο, γιατί το m δεν είναι τετραγωνικό υπόλοιπο modulo n.

5.2 Το πρόβλημα του Graph 3 Coloring είναι CZK

Μετά από όλα αυτά μπορούμε να δώσουμε ένα πρωτόκολλο computational zero knowledge για το πρόβλημα αυτό [9].

Ορισμός 13 (GRAPH 3 COLORING) *Δίνεται ένας γράφος $G = (V, E)$ με κόμβους $V = \{1, 2, \ldots, n\}$. Μπορούμε να χρωματίσουμε τους κόμβους του με 3 χρώματα, ώστε γειτονικοί κόμβοι να μην έχουν το ίδιο χρώμα; Υπάρχει δηλαδή $\phi : V \to \{1,2,3\}$, τ.ώ. $\forall (u,v) \in E \Rightarrow \phi(u) \neq \phi(v)$;*

Ο αλγόριθμος αυτός χρησιμοποιεί το σχήμα των Goldwasser-Micali αλλά αυτό μπορεί να γενικευτεί χρησιμοποιώντας κάθε nonuniformly secure συνάρτηση (βλέπε παράρτημα για nonuniform complexity).

Η απόδειξη αυτή είναι IP. Το πρωτόκολλο αποφαίνεται σωστά για yes instances του προβλήματος (completeness). Για το soundness, αν δεν υπάρχει 3 coloring, σημαίνει ότι υπάρχει τουλάχιστον ένα $\{u,v\} \in E$, τ.ώ. $\phi'(u) = \phi'(v)$. Η πιθανότητα να επιλέξει "κακή" πλευρά ο V είναι τουλάχιστον $1/m$ και άρα η πιθανότητα ο P να ξεγελάσει τον V είναι το πολύ $1 - 1/m$. Επαναλαμβάνοντας για m^2 γύρους, η πιθανότητα να ξεγελαστεί ο V σε όλους είναι φραγμένη από $(1-1/m)^{m^2} \to e^{-m}$ καθώς το m μεγαλώνει. Άρα και το soundness ικανοποιείται.

Είναι το πρωτόκολλο CZK; Θα πρέπει να είναι, γιατί σε κάθε γύρο ο V βλέπει μόνο permutations του αρχικού coloring, τα οποία permutations είναι, λόγω των ιδιοτήτων των commitment schemes, computationally indistinguishable για διαφορετικά input. Η μορφή των transcripts εδώ είναι $(G, A_1, \ldots A_{m^2})$, όπου:

$$A_i = ((coloring\ encryption), (u,v), (c_{u,1}c_{u,2}, r_{u,1}, r_{u,2}), (c_{v,1}c_{v,2}, r_{v,1}, r_{v,2}))$$

Πραγματικά το πρωτόκολλο είναι CZK και υπάρχει simulator ο οποίος μπορεί να βγάζει transcripts παρόμοια με τα αυθεντικά (computationally indistinguishable). Αυτός δίνεται στο παράρτημα, μαζί με την ολοκληρωμένη απόδειξη. Το συμπέρασμα είναι ότι υπάρχει computationally zero knowledge απόδειξη, με δεδομένο ότι υπάρχουν συναρτήσεις που να είναι nonuniformly secure.

input: $G = (E, V)$ on vertex set $\{1, \ldots, n\}$, $|E| = m$
repeat m^2 times

1. Let ϕ be the 3-coloring. P picks permutation π of $\{1, 2, 3\}$ and computes $c_i = \pi(\phi(i))$ for each vertex i. Since there are 3 colors, P needs 2 bits and encodes each c_i as sequence of 2 bits: $c_i = c_{i,1}c_{i,2}$. For each vertex P picks random $r_{i,1}$, $r_{i,2}$ and computes $R_{i,1} = f(c_{i,1}, r_{i,1})$ and $R_{i,2} = f(c_{i,2}, r_{i,2})$. P sends $(R_{1,1}, R_{1,2}, \ldots, R_{n,1}, R_{n,2})$ to V. f is the Goldwasser-Micali probabilistic encryption function.

2. V chooses random edge $\{u, v\} \in E$ and sends it to P.

3. P reveals commitment sending $(c_{u,1}c_{u,2}, r_{u,1}, r_{u,2})$ and $(c_{v,1}c_{v,2}, r_{v,1}, r_{v,2})$ to V.

4. V checks that $c_{v,1}c_{v,2} \neq c_{u,1}c_{u,2} \neq 00_{(2)}$, $R_{u,1} = f(c_{u,1}, r_{u,1})$, $R_{u,2} = f(c_{u,2}, r_{u,2})$, $R_{v,1} = f(c_{v,1}, r_{v,1})$, $R_{v,2} = f(c_{v,2}, r_{v,2})$.

V accepts if last step accepts in each of the m^2 rounds.

Πίνακας 5: CZK απόδειξη για το G3C

5.3 CZK αποδείξεις για κάθε NP γλώσσα

Το G3C είναι NP-complete πρόβλημα. Μια ιδέα θα ήταν επομένως για κάθε NP πρόβλημα, να ανάγεται αυτό στο G3C και στη συνέχεια να γίνεται η απόδειξη του G3C. Αν και αυτό φαίνεται διαισθητικά σωστό, υπάρχει μια τεχνική λεπτομέρεια που μπορεί να δημιουργήσει προβλήματα.

Ας θεωρήσουμε μια NP γλώσσα L και μια αναγωγή της στο G3C, t. Δηλαδή:

$$\forall x \in \{0,1\}^*, x \in L \Leftrightarrow t(x) \in G3C$$

Αρχικά ο P και ο V^*[6] υπολογίζουν την αναγωγή t, η οποία λόγω των ιδιοτήτων των αναγωγών είναι μια πολυωνυμική αναγωγή αλλά και αναστρέψιμη πολυωνυμικά. Ο P αρχικοποιεί σαν ρουτίνα το σύστημα απόδειξής του για το G3C, έστω P_{G3C} με είσοδο $t(x)$. Ο V^* με τη σειρά του χρησιμοποιεί τον verifier που έχει ενσωματωμένο για το G3C, έστω V^{**}. Δηλαδή έχουμε την εξής ισοδυναμία:

$$(P_{G3C}, V^{**}(x))(t(x)) \equiv (P, V^*)(x)$$

Το πρόβλημα έγκειται στο ότι ο V^{**} μπορεί να χρησιμοποιήσει εκτός από την κανονική του είσοδο $t(x)$ και την αρχική είσοδο x, μαθαίνοντας έτσι πιθανώς παραπάνω πληροφορίες για το πρόβλημα. Στην παραπάνω σχέση δεν μπορούμε να χρησιμοποιήσουμε την απόδειξη που κάναμε για το G3C καθώς δεν είχαμε συνυπολογίσει την "βοηθητική" είσοδο του verifier.

Αυτό που κάνουμε είναι: Κατασκευάζουμε έναν V^{***}, ο οποίος με είσοδο $t(x)$ υπολογίζει το x (η t είναι πολυωνυμικά αντιστρέψιμη) και εφαρμόζει τον V^{**}. Τότε το πρωτόκολλο $(P_{G3C}, V^{***})(t(x))$ είναι ZK[7]. Επομένως υπάρχει ένας simulator $M_{V^{***}}$, ώστε τα $\{M_{V^{***}}\}$ και $\{VIEW^{P_{G3C}}_{V^{***}}\}$ να είναι computationally indistinguishable. Αν θέσουμε $M_{V^*} := M_{V^{***}}$, τότε θα ισχύει και ότι $\{VIEW^{P}_{V^*}\}$ είναι computationally indistinguishable από το $\{M_{V^*}\}$.

Το συμπέρασμα είναι ότι **όλες οι γλώσσες στο NP έχουν computationally zero knowledge αποδείξεις υπό την προυπόθεση ότι υπάρχουν nonuniformly secure συναρτήσεις** [9] (ισοδύναμα: nonuniformly secure bit commitment schemes).

6 Ανοικτά προβλήματα και σχετικά αποτελέσματα

Τα εξής προβλήματα παραμένουν ανοικτά:

- Υπάρχουν CZK συστήματα αποδείξεων για σύνολα στο NP, χωρίς τις προαναφερθείσες υποθέσεις;

[6] Ο V^* γενικά μπορεί να είναι οποιοσδήποτε ακόμη και μη έντιμος verifier που θέλει να αποκομίσει γνώση από τον prover

[7] Το zero knowledge των πρωτοκόλλων καθορίζεται στην ουσία από τον prover. Ο verifier μπορεί να είναι όπως είδαμε οποιοσδήποτε, οσοδήποτε πονηρός θέλει.

- Υπάρχουν PZK συστήματα αποδείξεων για σύνολα στο NP, και αν ναι με ποιες προϋποθέσεις; (φαίνεται απίθανο)

Επιπλέον έχει αποδειχθεί το παρακάτω:

Θεώρημα 2 *Αν μια γλώσσα L έχει perfect ή zero knowledge σύστημα απόδειξης, τότε $L \in AM \cap coAM$.*

Ας πούμε ότι έχουμε μια NP-complete γλώσσα, η οποία ας υποθέσουμε ότι έχει perfect zero knowledge σύστημα απόδειξης. Τότε από το παραπάνω, θα ανήκει στο $coAM$ [8] και επομένως το συμπλήρωμά της θα πρέπει να ανήκει στο AM. Το συμπέρασμα είναι ότι οι NP-complete γλώσσες δεν μπορούν να έχουν perfect ή zero knowledge συστήματα, εκτός αν όλες οι coNP γλώσσες ανήκουν στο AM, κάτι που θεωρείται απίθανο.

Έχει επίσης αποδειχτεί ότι, ό,τι αποδεικνύεται από ένα IP σύστημα, μπορεί να αποδεικτεί σε ένα σύστημα zero knowledge, με τις ίδιες προϋποθέσεις που ισχύουν και για τις NP γλώσσες.

Θεώρημα 3 *Κάθε γλώσσα στο IP, ανήκει και στο CZK, αν υπάρχουν nonuniformly secure encryption functions.*

Η απόδειξη που παρουσιάζεται στο [5] χρησιμοποιεί AM παιχνίδια.

7 Σύνθεση αποδείξεων

Επίσης στο σχεδιασμό κρυπτογραφικών πρωτοκόλλων μας ενδιαφέρει η σύνθεση αποδείξεων μηδενικής γνώσης [3]. Για παράδειγμα μπορεί να χρειάζεται σε ένα πρωτόκολλο να γίνουν πολλές "αποδείξεις" που η κάθε μια να είναι zero knowledge.

Ορισμός 14 (Sequential Composition) *Στην περίπτωση αυτή το πρωτόκολλο εκτελείται σειριακά, πολυωνυμικό αριθμό φορών, η μία μετά την άλλη. Αν όλες οι εκτελέσεις του αποδέχονται τότε το sequential composition αποδέχεται.*

Ορισμός 15 (Parallel Composition) *Στην περίπτωση αυτή, το πρωτόκολλο εκτελείται πολλές φορές (πολυωνυμικά) παράλληλα, δηλαδή κάθε γύρος του εκτελείται σε όλα τα instances του πρωτοκόλλου. Αποδεχόμαστε αν αποδεχτούν όλα τα πρωτόκολλα.*

Πρόταση 1 (Sequential Composition Lemma) *Κάθε zero knowledge πρωτόκολλο, όπου επιτρέπεται στον verifier να πάρει βοηθητικό input (το οποίο γνωρίζει και ο simulator) το οποίο είναι από πριν καθορισμένο και μπορεί να τον βοηθάει να αποκτήσει γνώση είναι κλειστό υπό το sequential composition.*

Και στις δύο περιπτώσεις μπορούμε να αναφερόμαστε σε πολλούς διαφορετικούς πιθανά provers και verifiers.

Πρόταση 2 (Parallel Composition Lemma) *To zero-knowledge δεν είναι γενικά κλειστό υπό το parallel composition*[8].

Η ιδέα της απόδειξης για το τελευταίο είναι η εξής: Έστω ένας prover P_1, ο οποίος στέλνει ένα "μυστικό" στον verifier αν και μόνο αν ο τελευταίος απαντήσει σωστά σε μια "δύσκολη" ερώτηση του prover. Επίσης οι απαντήσεις αυτές ας υποθέσουμε ότι είναι ολές indistinguishable (μοιάζουν pseudorandom). Ο P_1 μπορεί να επαληθεύσει αν είναι σωστές, αλλά κανένας probabilistic polytime verifier δεν μπορεί να τις απαντήσει. Οποτε το πρωτόκολλο είναι zero knowledge. Από την άλλη έστω ένας prover P_2, ο οποίος απαντά τέτοιες δύσκολες ερωτήσεις στον verifier. Το πρωτόκολλο αυτό είναι πάλι zero knowledge γιατί τα strings που στέλνει φαίνονται πάλι pseudorandom. Παρόλα αυτά και με τον κατάλληλο συγχρονισμό, η παράλληλη σύνθεση δεν είναι zero knowledge: Ένας verifier μπορεί να απαντήσει την ερώτηση του P_1, στέλνοντάς την στον P_2 , παίρνοντας την απάντησή του και δίνοντας την στον P_1, παίρνοντας το "μυστικό", δηλαδή κάτι που δεν μπορούσε να υπολογίσει μόνος του.

[8] Αλλά υπό προϋποθέσεις intractability για προβλήματα όπως το factoring κάθε NP-σύνολο έχει ένα parallel-zero-knowledge proof, το οποίο μάλιστα έχει σταθερό αριθμό γύρων.

8 Zero Knowledge Arguments

Μια αθενέστερη μορφή πρωτοκόλλων είναι τα zero knowledge arguments. Εδώ ο prover δεν είναι απεριόριστων δυνατοτήτων, αλλά είναι και αυτός probabilistic polynomial time, ο οποίος όμως παίρνει τα παραπάνω δεδομένα που χρειάζεται από μια βοηθητική ταινία εισόδου, της οποίας τα περιεχόμενα είναι γνωστά μόνο σε αυτόν πριν ξεκινήσει το πρωτόκολλο. Στην ουσία "χαλαρώνουμε" την απαίτηση για το soundness: Να μην μπορεί οποιοσδήποτε **PPT** P^* με βοηθητικό input να "ξεγελάσει" τον verifier για false instances. Ισχύει το παρακάτω:

Θεώρημα 4 (PZK Arguments for NP) *Αν υπάρχουν nonuniform one way συναρτήσεις, τότε κάθε γλώσσα στο NP έχει perfect zero knowledge argument [6].*

Η αρχική ιδέα διατυπώθηκε με την προϋπόθεση του intractability του *QUADRADIC RESIDUOSITY*.

Παρατηρεί κανείς ότι γίνεται μια "ανταλλαγή" στο είδος του zero knowledge και τις ικανότητες του prover. Για κάθε γλώσσα στο NP μπορούμε είτε να πετύχουμε μόνο computational zero knowledge με απόλυτο soundness, είτε απόλυτο zero knowledge με computational soundness.

9 Αποδείξεις γνώσης μηδενικής γνώσης

Η έννοια των αποδείξεων γνώσης (proofs of knowledge) [4] είναι λίγο διαφορετική. Στα παραπάνω παραδείγματα είχαμε ένα στοιχείο x και ο prover αποδείκνυε ότι $x \in L$. Για παράδειγμα αν είχαμε μια φόρμουλα CNF, ο prover αποδείκνυε ότι είναι satisfiable. Εδώ ενδιαφέρει να αποδεικνύεται ότι **ο prover "γνωρίζει" ένα satsifying assignment για την φόρμουλα** . Για παράδειγμα ο prover θα μπορούσε να είναι και αυτός PPT, και να παίρνει το satisfying assigment σαν είσοδο σε μια βοηθητική ταινία. Τότε σκοπός του θα ήταν να αποδείξει ότι ξέρει όντως το satisfying assigment. Επιπλέον ενδιαφέρει η απόδειξη αυτή να είναι zero knowledge, με την έννοια που αναφέραμε παραπάνω.

Το ότι ο prover "γνωρίζει" ένα τέτοιο στοιχείο (π.χ. ένα 3-coloring για ένα γράφο), σημαίνει ότι υπάρχει μια μηχανή που θα λέγεται "knowledge extractor", η οποία χρησιμοποιώντας τον prover σαν oracle, μπορεί να κατασκευάσει το στοιχείο αυτό αποδοτικά.

Ορισμός 16 (Witness Relation) *Έστω $R \subseteq \{0,1\}^* \times \{0,1\}^*$ μια δυαδική σχέση. Τότε ορίζουμε μια γλώσσα $L_R := \{x | \exists s.(x,s) \in R\}$. Αν υπάρχει τέτοιο s, το s ονομάζεται witness για το x. Για παράδειγμα μια γλώσσα στο NP ορίζεται ως μια L_R, όπου η συμμετοχή στην R μπορεί να ελεγχθεί πολυωνυμικά και για το witness s που αντιστοιχεί στο πρόβλημα x ισχύει $|s| \leq |x|^k$ για κάποιο k.*

Ορισμός 17 (Συνάρτηση μηνυμάτων του P) *Έστω $P_{x,y,r,}(\overline{m})$ το μήνυμα που στέλνει ο P όταν έχει αρχική είσοδο x, βοηθητική είσοδο y, και τυχαία είσοδο r, όταν λάβει στο πρωτόκολλο τα μηνύματα $\overline{m}$. Η $P_{x,y,z}$ θα λέγεται συνάρτηση μηνυμάτων του P.*

Σκοπός μας είναι οι knowledge extractors να είναι oracle machines που έχουν πρόσβαση στη συνάρτηση αυτή. Ο χρόνος εκτέλεσης των knowledge extractors θα είναι αντίστοφα ανάλογος (κατά έναν πολυωνυμικό παράγοντα) της πιθανότητας με την οποία αποδέχεται ο verifier.

Ορισμός 18 (Knowledge Verifier) *Έστω σχέση R. Ένας interactive αλγόριθμος V θα λέγεται knowledge verifier για τη σχέση R, αν ισχύουν:*

- *Υπάρχει interactive αλγόριθμος P, τ.ώ $\forall (x,y) \in R$, όλα τα interactions (P,V) με είσοδο x και βοηθητική είσοδο y για τον P αποδέχονται.*
- *Υπάρχει μια σταθερά m, και μια oracle machine K (universal knowledge extractor), ώστε $\forall x \in L_R$, $\forall y, r$, η K κάνει τα εξής: Αν η πιθανότητα ο V να αποδέχεται στα interactions με τον P είναι $p(x)$, η μηχανή αυτή εξάγει ένα s, ώστε $(x,s) \in R$ και τερματίζει σε expected αριθμό βημάτων:*

$$\frac{|x|^m}{p(x)}$$

Ορισμός 19 (Σύστημα αποδείξεων γνώσης) *Το πρωτόκολλο (P,V), όπου ισχύει η πρώτη συνθήκη και όπου ο V είναι knowledge verifier, είναι ένα σύστημα αποδείξεων γνώσης.*

Τα συστήματα που περιγράψαμε σε προηγούμενες ενότητες είναι συστήματα αποδείξεων γνώσης.

9.1 Εφαρμογή στην ταυτοποίηση χρηστών

Τα συστήματα αυτά μπορούν να χρησιμοποιηθούν για την ταυτοποίηση χρηστών. Η ιδέα είναι η εξής: Κάθε χρήστης αποθηκεύει ένα δύσκολο πρόβλημα, π.χ. ένα θεώρημα σε ένα δημόσιο αρχείο. Την απόδειξη του θεωρήματος ξέρει μόνο αυτός (π.χ. του έχει δοθεί αρχικά). Για να αναγνωριστεί ο χρήστης από το σύστημα αποδεικνύει ότι γνωρίζει την απόδειξη στο θεώρημα αυτό. Αν η απόδειξη είναι πειστική, τότε αποκτά πρόσβαση. Αν το πρωτόκολο είναι μηδενικής γνώσης, αυτό εγγυάται ότι κανένας eavesdropper που παρακολουθεί τη διαδικασία δεν μπορεί να μάθει αρκετά για να αποκτήσει και αυτός πρόσβαση.

Το σχήμα ταυτοποίησης Fiat-Shamir [7] είναι ένα τέτοιο σύστημα μηδενικής γνώσης. Στον Πίνακα 6 φαίνεται η εγκατάσταση του πρωτοκόλλου και στον Πίνακα 7 φαίνεται η διαδικασία ταυτοποίησης.

- trusted authority publishes n with unknown factorization $n = p \cdot q$.
- P chooses random S, where $1 < S < n$ and $gcd(S, n) = 1$.
- P computes $I = S^2 mod n$, publishes I.

Purpose: P has to convince V thah he knows the secret S.

Πίνακας 6: Αρχικοποίηση του πρωτοκόλλου FS

V must be convinced that P knows the secret S, corresponding to (I, n).
repeat t times (t is a security parameter)

- P chooses at random R, where $1 < R < n$, and computes $X = R^2 mod n$.
- P sends X to V.
- V guesses a random $i \in \{0, 1\}$.
- if $i = 0$ then P sends R, else $(R \cdot S) mod n$.
- V checks that $R^2 \equiv X (mod n)$ if $i = 0$, else checks that $(R \cdot S)^2 \equiv X \cdot I (mod n)$.

V accepts if last step accepts in each of the t rounds.

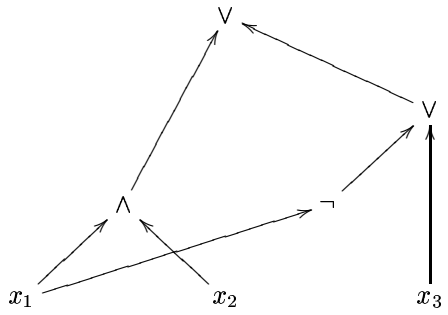
Πίνακας 7: Πρωτόκολλο FS

Παράρτημα Α
Boolean Circuits και Nonuniform Complexity

Οι οικογένειες κυκλωμάτων πολυωνυμικού μεγέθους αναφέρονται πολύ συχνά σαν κριτές σε συστήματα μηδενικής γνώσης. Αναφέρουμε τους σχετικούς ορισμούς και μερικές χρήσιμες προτάσεις. Για περισσότερες λεπτομέρειες μπορεί κανείς να αναφερθεί στο [1].

Ορισμός 20 (Boolean Circuit) *Ένα Boolean Circuit είναι ένας ακυκλικός κατευθυνόμενος γράφος $C = (V, E)$, όπου τα στοιχεία του V καλούνται πύλες του κυκλώματος. Όλοι οι κόμβοι του κυκλώματος έχουν πρόσβαθμο 0, 1 ή 2. Κάθε πύλη είναι ενός είδους. Τα είδη είναι* ***true***, ***false***, $\vee$, $\wedge$, $\neg$, x_1, x_2, ..., *όπου x_i μεταβλητές. Πύλες με πρόσβαθμο 0 ονομάζονται πύλες εισόδου του κυκλώματος και το είδος τους είναι πάντα* ***true***, ***false*** *ή x_i. Οι πύλες $\vee$, $\wedge$ έχουν πρόσβαθμο 2, η πύλη $\neg$ έχει 1. Υπάρχει τέλος μοναδική πύλη που έχει απόβαθμο 0 και ονομάζεται πύλη εξόδου του κυκλώματος.*

Σαν **μέγεθος** ενός κυκλώματος αναφέρουμε το πλήθος των πυλών. Τα κυκλώματα αντιστοιχούν σε υπολογισμούς λογικών συναρτήσεων, όπου στις εισόδους αναθέτουμε σταθερές ή μεταβλητές. Στις πύλες του κυκλώματος αντιστοιχίζουμε τις κατάλληλες λογικές συναρτήσεις και στην έξοδο παίρνουμε την έξοδο της συνάρτησης.



Εικόνα 2: Boolean Circuit για τη συνάρτηση $f(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee (x_3 \vee \neg x_1)$

Ορισμός 21 (Οικογένεια κυκλωμάτων) *Ορίζουμε σαν οικογένεια κυκλωμάτων (circuit family) την (πιθανά άπειρη) ακολουθία $C = \{C_0, C_1, \ldots\}$, όπου C_i κύκλωμα με i εισόδους.*

Ορισμός 22 (Πολυωνυμικού μεγέθους οικογένεια κυκλωμάτων) *Έστω γλώσσα $L \subseteq \{0,1\}^*$. Τότε θα λέμε ότι η γλώσσα έχει πολ/κού μεγέθους οικογένεια κυκλωμάτων (polynomial size circuit family) αν υπάρχει μια οικογένεια κυκλωμάτων C, ώστε:*

- $size(C_n) \leq p(n)$ *για κάποιο σταθερό πολυώνυμο* p.
- $\forall x \in \{0,1\}^*, x \in L \Leftrightarrow C_{|x|}(x) = true$.

Οι γλώσσες που έχουν πολυωνυμικές οικογένειες κυκλωμάτων ανήκουν στην κλάση πολυπλοκότητας $\mathbf{P/poly}$.

Η κλάση πολυπλοκότητας αυτή είναι πιο ισχυρή από την κλάση των πολυωνυμικών αλγορίθμων, γιατί η κατασκευή των κυκλωμάτων μπορεί να χρειάζεται μη φραγμένη υπολογιστική ισχύ. Έχοντας το μήκος ενός κυκλώματος επιτρέπεται δηλαδή να χρειάζεται μη φραγμένη ισχύ για να κατασκευαστεί το κατάλληλο κύκλωμα που θα αναγνωρίζει τις εισόδους αυτού του μήκους. Αν υπάρχει *logn*-space μηχανή Turing η οποία με είσοδο 1^n, δίνει το αντίστοιχο C_n, τότε λέμε ότι η οικογένεια C είναι **uniform**. Αν για κάποιο n χρειαζόμαστε μη φραγμένη ισχύ για να παράγουμε το C_n, τότε η οικογένεια χαρακτηρίζεται σαν **non-uniform**. Υπάρχουν ορισμένες χρήσιμες προτάσεις για την κλάση *P/poly*:

1. $P \subseteq P/poly$. Αυτό προκύπτει από ότι το *CIRCUIT VALUE* είναι P-complete πρόβλημα, η κατασκευή των κυκλωμάτων μάλιστα γίνεται σε *logn*-space και άρα κάθε γλώσσα στο P έχει uniformly polynomial size circuit family.
2. $BPP \subseteq P/poly$.
3. Εικασία: Τα NP-complete προβλήματα δεν έχουν uniformly polynomial size circuits.
4. $P/poly \not\subseteq NP$. Η κλαση *P/poly* αποφασίζει ακόμη και για κάποιες undecidable(!) γλώσσες. Π.χ. $L = \{1^n | (M_n; x) \in HALTING\}$. Προφανώς η L είναι undecidable. Οι συμβολοσειρές της γλώσσας είναι τα 1^n, και επομένως υπάρχει μόνο μια συμβολοσειρά για δεδομένο μήκος. Έστω για ένα n, A_n ένα κύκλωμα που δίνει 1 αν η είσοδός του είναι η 1^n. Επίσης έστω B_n ένα κύκλωμα που δίνει 0 πάντα. Προφανώς ένα από τα δύο είναι το σωστό για το μήκος n. Επιλέγουμε C_n να είναι το σωστό από αυτά τα δύο[9].

[9] Δεν υπάρχει παρόλα αυτά μηχανή Turing που θα μας έδινε το σωστό από αυτά τα δύο.

Παράρτημα Β
Το Graph-3-Coloring είναι ZK

Συναρτήσεις κρυπτογράφησης και hybrid arguments

Εδώ δεν θα χρησιμοποιήσουμε τα bit commitments που χρησιμοποιήσαμε παραπάνω, αλλά για απλότητα θα μιλήσουμε για συναρτήσεις ο οποίες κωδικοποιούν ολόκληρο το χρώμα ενός κόμβου, έχοντας παρόμοιες ιδιότητες ασφάλειας. Οι δύο προσεγγίσεις είναι ισοδύναμες.

Ορισμός 23 (Encryption Function) *Μια συνάρτηση* $f : \{0,1,2,3\} \times \{0,1\}^* \to \{0,1\}^*$ *θα λέγεται encryption function αν είναι πολυωνυμικά υπολογίσιμη και ισχύει ότι*

$$\forall x \neq y \in \{0,1,2,3\}, \forall r,s \in \{0,1\}^* f(x,r) \neq f(y,s)$$

Ορισμός 24 (Probabilistic Encryption) *Αν τα δεύτερα ορίσματα παίρνονται τυχαία από το* $\{0,1\}^n$*, τότε ορίζεται η τυχαία μεταβλητή* $f_n(x) = f(x,r)$ *όπου* r *τυχαίος.*

Ορισμός 25 (Secure Function) *Η* f *λέγεται secure αν* $\forall x \neq y$, $\{f_n(x)\}$ *και* $\{f_n(y)\}$ *είναι computationally indistinguishable.*

Ορισμός 26 (Nonuniformly Secure) *Η* f *λέγεται nonuniformly secure αν* $\forall x \neq y$, $\{f_n(x)\}$ *και* $\{f_n(y)\}$ *είναι indistinguishable από πολυωνυμικού μεγέθους οικογένειες κυκλωμάτων. Δηλαδή* $\forall C = \{C_n\}$ *polysize circuits,* $\forall c > 0$, $\forall n$ *αρκετά μεγάλο:*

$$|prob(C_n(f_n(x)) = 1) - prob(C_n(f_n(y)) = 1)| < 1/n^c$$

Λήμμα 1 (Hybrid Argument) *Έστω* $a^1, a^2, \ldots$ *και* $b^1, b^2, \ldots$*, ακολουθίες ώστε το κάθε στοιχείο να είναι ακολουθία με πολυωνυμικά φραγμένο μήκος, δηλαδή* $a^n = a_1^n \ldots a_{n^k}^n$ *και αντίστοιχα για το* b^n*. Επίσης συμβολίζουμε με* $\overline{f}_n(a^n)$ *τα encryptions κάθε ενός στοιχείο του* a^n*, δηλ. τα* $f_n(a_1^n) \ldots f_n(a_{n^k}^n)$ *και όμοια για το* b^n*. Τότε* $\forall C = \{C_n\}$ *polysize circuits,* $\forall c > 0$, $\forall n$ *αρκετά μεγάλο:*

$$|prob(C_n(\overline{f}_n(a^n)) = 1) - prob(C_n(\overline{f}_n(b^n)) = 1)| < 1/n^c$$

Απόδειξη: Με απαγωγή σε άτοπο. Έστω ότι υπάρχει τέτοια οικογένεια κυκλωμάτων. Θεωρούμε την ακολουθία c (hybrid sequence) για την οποία ισχύει:

$$\begin{aligned} c_0^n &= a_1^n \ldots a_{n^k}^n \equiv a^n \\ c_1^n &= a_1^n \ldots a_{n^k-1}^n b_{n^k}^n \\ &\ldots \\ c_{n^k-1}^n &= a_1^n b_2^n \ldots b_{n^k}^n \\ c_{n^k}^n &= b_1^n b_2^n \ldots b_{n^k}^n \equiv b^n \end{aligned}$$

Όμοια με πριν μπορούμε να υπολογίσουμε για κάθε τέτοια ακολουθία το encryption της π.χ. για το c_0^n, έχουμε $\overline{f}_n(c_0^n)$. Τώρα αφού υποθέσαμε ότι υπάρχει τέτοια οικογένεια κυκλωμάτων, με την παρατήρηση ότι η αρχική και η τελική ακολουθία ταυτίζονται με τις a_n και b_n, θα ισχύει ότι:

$$|prob(C_n(\overline{f}_n(c_0^n)) = 1) - prob(C_n(\overline{f}_n(c_{n^k}^n)) = 1)| > 1/n^c$$

Από αυτό συμπεραίνουμε ότι υπάρχει ένα ζεύγος διαδοχικών στοιχείων της ακολουθίας c^n, τα οποία έχουν διακρίσιμη nonuniform απόσταση, γιατί αλλιώς, αν όλα τα ζευγάρια είχαν αμελητέα απόσταση, και οι αρχικές και τελικές τιμές θα είχαν αμελητέα απόσταση. Δηλαδή υπάρχει l, τ.ώ:

$$|prob(C_n(\overline{f}_n(c_l^n)) = 1) - prob(C_n(\overline{f}_n(c_{l+1}^n)) = 1)| > 1/n^c$$

Όμως έχουμε ότι:

$$\begin{aligned}\overline{f}_n(c_l^n) &= f_n(a_1^n)\ldots\mathbf{f_n(a^n_{n^k-l})}f_n(b^n_{n^k-l+1})\ldots f_n(b^n_{n^k})\\ \overline{f}_n(c_{l+1}^n) &= f_n(a_1^n)\ldots\mathbf{f_n(b^n_{n^k-l})}f_n(b^n_{n^k-l+1})\ldots f_n(b^n_{n^k})\end{aligned}$$

Πρακτικά αυτό είναι σαν να έχουμε ένα σύνολο τυχαίων μεταβλητών X_i που αντιστοιχούν στα $f_n(a_i^n)$ και Y_i, που αντιστοιχούν στα $f_n(b_i^n)$ και ισχύει ότι:

$$|prob(C_n(X_1,\ldots X_{n^k-l}\ldots Y_{n^k})=1)-prob(C_n(X_1,\ldots Y_{n^k-l}\ldots Y_{n^k})=1)|>1/n^c$$

Αν θεωρήσω ανεξαρτησία, τότε:

$$\begin{aligned}&prob(C_n(X_1,\ldots X_{n^k-l}\ldots Y_{n^k})=1)-prob(C_n(X_1,\ldots Y_{n^k-l}\ldots Y_{n^k})=1)=\\ &\sum_{y_{n^k}}\cdots\sum_{y_{n^k-l+1}}\sum_{x_{n^k-l}}\cdots\sum_{x_1}prob(C_n(X)=1|X=X_1\ldots X_{n^k-l}\ldots Y_{n^k})\\ &prob(X_1=x_1)\ldots prob(X_{n^k-l}=x_{n^k-l})\ldots prob(Y_{n^k}=y_{n^k})-\\ &\sum_{y_{n^k}}\cdots\sum_{y_{n^k-l+1}}\sum_{y_{n^k-l}}\cdots\sum_{x_1}prob(C_n(X)=1|X=X_1\ldots Y_{n^k-l}\ldots Y_{n^k})\\ &prob(X_1=x_1)\ldots prob(Y_{n^k-l}=y_{n^k-l})\ldots prob(Y_{n^k}=y_{n^k})\end{aligned}$$

Χρησιμοποιώντας ένα γενικευμένο argument για μέσο όρο[10] μπορούμε να συμπεράνουμε ότι υπάρχουν σταθερά $(x'_1,\ldots x'_{n^k-l-1}$ και $y'_{n^k-l+1}\cdots y_{n^k}$, τ.ώ.

$$|prob(C_n(x'_1\ldots X_{n^k-l}\ldots y'_{n^k})=1)-prob(C_n(x'_1\ldots Y_{n^k-l}\ldots y'_{n^k})=1)>1/n^c$$

Αν θεωρήσουμε μια νέα οικογένεια κυκλωμάτων $C'=\{C'_n\}$ ώστε

$$C'_n(X)=C_n(x'_1\ldots X\ldots y'_{n^k})$$

τότε θα ισχύει:

$$|prob(C'_n(X_{n^k-l})=1)-prob(C'_n(Y_{n^k-l})=1)|>1/n^c$$

Δηλαδή:

$$|prob(C'_n(f_n(a^n_{n^k-l}))=1)-prob(C'_n(f_n(b^n_{n^k-l}))=1)|>1/n^c$$

Αυτό όμως είναι **άτοπο** γιατί η f είναι nonuniformly secure. □

Πρωτόκολλο για το G3C

Εδώ παρουσιάζουμε το πρωτόκολλο για το G3C βασισμένο όχι σε bit commitment αλλά σε nonuniformly secure encryption συνάρτηση f [9].

Παρακάτω είναι ο simulator για το πρωτόκολλο αυτό.

Μερικοί συμβολισμοί. Θα συμβολίζουμε με $msg_{PV}^{\phi,\pi}$ το $(\pi(\phi(1)),\ldots\pi(\phi(n)))$, όπως αυτά δίνονται σε κάποιο γύρο από το πρωτόκολλο. Επίσης θα συμβολίζουμε με $msg_M^{(u,v)(a,b)}$ το $(c_1,\ldots,c_n)$, όπου $c_i=0$ για $i\neq u,v$, $c_u=a$, $c_v=b$.

Πρόταση 3 (Ο simulator είναι expected polytime)

[10] Όταν $\sum_a\phi(a)>\epsilon$, τότε υπάρχει ένα a', τ.ώ. $\phi(a')>\epsilon$.

input: $G = (E, V)$ on vertex set $\{1, \ldots, n\}$, $|E| = m$
repeat m^2 times

1. P chooses permutation π of coloring and random r_v for each vertex v. P computes: $F_v = f(\pi(\phi(v)), r_v)$, and sends $F_1, F_2 \ldots F_n$.
2. V chooses randomly edge e and sends it to P.
3. if $e = \{u, v\} \in E$, P sends $(\pi(\phi(u)), r_u)$, $(\pi(\phi(v)), r_v)$, else sends zero as permutation.
4. V checks that $F_u = f(\pi(\phi(u)), r_u)$, $F_v = f(\pi(\phi(v)), r_v)$, $\pi(\phi(u)) \neq \pi(\phi(v))$, and $\pi(\phi(u)) \in \{1, 2, 3\}$, $\pi(\phi(v)) \in \{1, 2, 3\}$.

V accepts if last step accepts in each of the m^2 rounds.

Πίνακας 8: CZK απόδειξη για το G3C

input: $G = (E, V)$ on vertex set $\{1, \ldots, n\}$, $|E| = m$
repeat m^2 times

1. Pick randomly an edge $(u, v) \in E$ and random different colors a, b.
2. For each vertex $i \in V$, pick random r_i and compute:
$$F_i = \begin{cases} f(0, r_i) & \forall i \neq u, v \\ f(a, r_i) & i = u \\ f(b, r_i) & i = v \end{cases}$$
3. Call V^* with input F_i vector, and the rest of the tape history, and get challenge e.
4. if $e = \{u, v\}$ append $(F_i, e, (a, r_u), (b, r_v))$ to tape, else goto previous step.

V Stop after m^2 rounds.

Πίνακας 9: Simulator M_{V^*} για το G3C

Απόδειξη: Η πιθανότητα ο simulator να πετύχει το challenge του verifier σε κάποια επανάληψη (εσωτερική) είναι η:

$$prob_{(u,v) \in_R E}(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u, v)) = \sum_{(u,v)} (1/m) prob(V * (\overline{f}_n(msg_M^{(u,v)(a,b)}))) = (u, v))$$

Όμως από το λήμμα παραπάνω θα πρέπει να ισχύει:

$$|prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u, v)) - prob(V^*(\overline{f}_n(msg_M^{(r,s)(a,b)})) = (u, v))| < 1/n^2$$

Γιατί αλλιώς ο V^* θα ήταν distinguisher για την $\overline{f}_n(msg_M^{(u,v)(a,b)})$. Επομένως για αρκετά μεγάλα n έχουμε:

$$prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u, v)) > prob(V^*(\overline{f}_n(msg_M^{(r,s)(a,b)})) = (u, v)) - 1/n^2$$

Επομένως:

$$\begin{array}{l} \sum_{(u,v)} \frac{1}{m} prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u, v)) > \\ \sum_{(u,v)} \frac{1}{m} (prob(V^*(\overline{f}_n(msg_M^{(r,s)(a,b)})) = (u, v)) - 1/n^2) \geq \\ \sum_{(u,v)} \frac{1}{m} (prob(V^*(\overline{f}_n(msg_M^{(r,s)(a,b)})) = (u, v)) - 1/2m) > \\ \sum_{(u,v)} \frac{1}{m} (prob(V^*(\overline{f}_n(msg_M^{(r,s)(a,b)})) = (u, v)) - 1/2) = \\ \frac{1}{m} ((prob(V^*(\overline{f}_n(msg_M^{(r,s)(a,b)})) \in E) - 1/2) = \\ \frac{1}{m}(1 - 1/2) = 1/2m \end{array}$$

Επομένως κάθε $2m$ εσωτερικές επαναλήψεις θα πετυχαίνει το challenge του πρωτοκόλλου και επομένως είναι expected polytime. □

Πρόταση 4 (Τα $\{M_{V^*}\}$ και $\{VIEW^P_{V^*}\}$ είναι nonuniformly indistinguishable)

Απόδειξη: Έστω ότι υπάρχει πολυωνυμικού μεγέθους οικογένεια κυκλωμάτων $C = \{C_n\}$ που διακρίνει τα δύο σύνολα. Δηλαδή έστω ότι $\forall c > 0, \forall\ V$ αρκετά μεγάλο:

$$|prob(C_{|V|}(M_{V^*}(G)) = 1) - prob(C_{|V|}(VIEW(G)) = 1)| > 1/|V|^c$$

Ένα τυπικό transcript έχει τη μορφή:

$$(G, r, (R_1, e_1, R'_1) \ldots (R_{m^2}, e_{m^2}, R'_{m^2}))$$

όπου G ο γράφος, r τα coin tosses του verifier, R_i τα εκάστοτε encryptions του permutation του coloring, e_i τα challenges του verifier, R'_i οι αποκαλύψεις του prover.

Κατασκευάζουμε την **υβριδική** ακολουθία Π για την οποία ισχύει ότι Π^i περιέχει τα G, r και τις πρώτες i τριάδες από την $VIEW$ ενώ τις υπόλοιπες $m^2 - 1$ από την M_{V^2}. Τότε θα ισχύει ότι $\Pi^0 \equiv M_{V^*}(G)$ και $\Pi^{m^2} \equiv VIEW(G)$. Όμοια με προηγούμενα αν υπάρχει οικογένεια κυκλωμάτων που αναγνωρίζει τα ακραία στοιχεία, τότε αναγνωρίζει και κάποιο γειτονικό ζεύγος. Δηλαδή:

$$\exists i, |prob(C_n(\Pi^i) = 1) - prob(C_n(\Pi^{i+1}) = 1)| > 1/n^c$$

Με χρήση ενός argument για μέσο όρο και με δεδομένο ότι τα γειτονικά μέλη της ακολουθίας Π διαφέρουν μόνο σε ένα στοιχείο, μπορεί να προκύψει και πάλι ότι υπάρχει μια άλλη οικογένεια κυκλωμάτων που αναγνωρίζει αυτά τα στοιχεία. Θέτουμε κατάλληλα σύμβολα για αυτά ακριβώς τα στοιχεία:

$$\begin{aligned} \Pi_{PV} &= (\overline{f}_n(msg^{\phi,\pi}_{PV}), (u,v), (\pi(\phi(u)), r_u), (\pi(\phi(v)), r_v)) \\ \Pi_M &= (\overline{f}_n(msg^{(u,v)(a,b)}_M), (u,v), (a, r_u), (b, r_v)) \end{aligned}$$

όπου στην πρώτη περίπτωση το (u,v) έχει προκύψει από κλήση στον verifier με το $\overline{f}_n(msg^{\phi,\pi}_{PV})$ενώ στη δεύτερη έχει προκύψει και πάλι από κλήση στον πραγματικό verifier με το $\overline{f}_n(msg^{(u,v)(a,b)}_M)$. Μιλάμε για τις τριάδες που διαφέρουν στα γειτονικά μέλη της ακολουθίας Π. Υπάρχει επομένως μια οικογένεια $C_0 = \{C_{n0}\}$ για την οποία ισχύει:

$$|prob(C_{n0}(\Pi_{PV}) = 1) - prob(C_{n0}(\Pi_M) = 1)| > 1/n^c$$

Επίσης εισάγουμε τους παρακάτω συμβολισμούς:

$$\begin{aligned} msg1 &= 0^{3n} \\ msg2 &= 1^n 2^n 3^n \end{aligned}$$

Δηλαδή κατασκευάζουμε δύο μηνύματα, το ένα από $3n$ μηδενικά και το άλλο από n 1, ακολουθούμενα από n 2, ακολουθούμενα από n 3. Τα encryptions αυτών των μηνυμάτων είναι από το λήμμα indistiguishable από πολυωνυμικές οικογένειες κυκλωμάτων. Εδώ θα δείξουμε ότι μπορούμε να χρησιμοποιήσουμε την $\{C_{n0}\}$ για να κατασκευάσουμε μια τέτοια οικογένεια. Έστω μια οικογένεια κυκλωμάτων τ.ώ. το C'_n να δέχεται εισόδους μεγέθους $3n$. Το κύκλωμα αυτό κάνει τα εξής:

Δύο παρατηρήσεις: Όταν κληθεί το C'_n με είσοδο $\overline{f}_n(msg1)$ μπορεί κανείς εύκολα να δεί ότι το V^* θα κληθεί με την από την κατανομή $\overline{f}_n(msg^{(u,v)(a,b)}_M)$, ενώ όταν κληθεί το C'_n με είσοδο το $\overline{f}_n(msg2)$ θα κληθεί ο V^* με είσοδο από την κατανομή $\overline{f}_n(msg^{\phi,\pi}_{PV})$. Καταρχήν και πάλι λόγω του λήμματος θα πρέπει να ισχύει:

$$|prob(V^*(\overline{f}_n(msg^{(u,v)(a,b)}_M)) = (u,v)) - prob(V^*(\overline{f}_n(msg^{\phi,\pi}_{PV})) = (u,v))| < 1/n^c$$

input: $t_1, \ldots, t_{3n}$ where $t_i \in \{0,1\}^*$

1. Pick randomly $(u,v) \in E$, π random permutation, r_u, r_v random integers in $\{0,1\}^n$.
2. Compute $text := t_{i_1} \ldots t_{i_{u-1}} f(c_u, r_u) t_{i_{u+1}} \ldots t_{i_{v-1}} f(c_v, r_v) t_{i_{v+1}} \ldots t_{i_n}$, where the j-th element of text is the element in the $\pi(\phi(j))$ third of the input, the u-th element is $f(c_u, r_u)$, the v-th element is $f(c_v, r_v)$. $c_u = \pi(\phi(u))$ and $c_v = \pi(\phi(v))$.
3. Run V^* on $text$.
4. **if** $V^*(text) \neq (u,v)$ **then** C'_n stops and outputs 0,
   else runs $C_{n0}(text, (u,v), (c_u, r_u), (c_v, r_v))$.

Πίνακας 10: C'_n κύκλωμα

Τώρα ισχυριζόμαστε ότι η οικογένεια $\{C'_n\}$ διακρίνει τα encryptions των $msg1$, $msg2$. Ισχύει:

$$\begin{aligned} prob(C'_n(\overline{f}_n(msg1)) = 1) &= prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u,v)) \cdot prob(C_{n0}(\Pi_M) = 1) \\ prob(C'_n(\overline{f}_n(msg2)) = 1) &= prob(V^*(\overline{f}_n(msg_{PV}^{\phi,\pi})) = (u,v)) \cdot prob(C_{n0}(\Pi_{PV}) = 1) \end{aligned}$$

Αυτό γιατί, όταν κληθεί ο V^* με τα $\overline{f}_n(msg_M^{(u,v)(a,b)})$και $\overline{f}_n(msg_{PV}^{\phi,\pi})$τότε οι τριάδες που δημιουργούνται ανήκουν στις κατανομές Π_M και $\Pi_P V$ αντίστοιχα! Έχουμε:

$$\begin{aligned} &|prob(C'_n(\overline{f}_n(msg1)) = 1) - prob(C'_n(\overline{f}_n(msg2)) = 1)| = \\ &|prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u,v)) \cdot prob(C_{n0}(\Pi_M) = 1) - \\ &- prob(V^*(\overline{f}_n(msg_{PV}^{\phi,\pi})) = (u,v)) \cdot prob(C_{n0}(\Pi_{PV}) = 1)| \geq \\ &prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u,v)) \cdot |prob(C_{n0}(\Pi_M) = 1) - prob(C_{n0}(\Pi_{PV}) = 1)| - \\ &- prob(C_{n0}(\Pi_{PV}) = 1) \cdot |prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u,v)) - prob(V^*(\overline{f}_n(msg_{PV}^{\phi,\pi})) = (u,v))| \geq \\ &\frac{1}{m} \cdot (1/n^c) \geq 1/n^{c'} \end{aligned}$$

Όπου στο τέλος χρησιμοποιούμε τα προηγούμενα αποτελέσματα και επίσης ότι:

$$prob(V^*(\overline{f}_n(msg_M^{(u,v)(a,b)})) = (u,v)) = 1/m$$

μιας και το (u,v) μπορεί να είναι τυχαίο edge του γραφου. Καταλήξαμε σε άτοπο από το προηγούμενο λήμμα. □

Το nonuniform security για τις συναρτήσεις, δεν μπορεί να γίνει απλό security, δηλαδή οι συναρτήσεις να μη σπάνε από πολυωνυμικούς αλγόριθμους. Ωστόσο έχουν αποδειχτεί διάφορες ασθενέστερες προτάσεις για κατάργηση της απαίτησης για nonuniform κριτές.

Αναφορές

[1] Papadimitriou, C. H. *Computational complexity.* Addison Wesley, 1994

[2] D. Stinson. *Cryptography theory and practice, 1st Edition.* CRC Press, 1995

[3] O. Goldreich. *Foundations of Cryptography (Fragments of a Book).* http://theory.lcs.mit.edu/~oded/frag.html, 1995

[4] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. *In CRYPTO'92,* SpringerVerlag (LNCS 740), pages 390–420, 1992.

[5] M. Ben-Or, O. Goldreich, S. Goldwasser, et al. Everything provable is provable in zeroknowledge. *In Proc. CRYPTO '88*, LNCS vol. 403, p. 40-51,Springer-Verlag, 1988.

[6] Guilles Brassard, David Chaum, and Claude Crepeau. Minimum disclosure proofs of knowledge. *JCSS*, 37(2):156-189, 1988.

[7] A. Fiat and A. Shamir. How to prove yourself: Practical solution to identification and signature problems. *In CRYPTO '86*, vol. 263 of LNCS, pp. 186–194. Springer Verlag, 1987.

[8] Lance Fortnow. The complexity of perfect zero-knowledge. *In Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 204-209, New York City, 25-27 May 1987.

[9] S. Goldwasser, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691-729, 1991.

[10] S. Goldwasser and S. Micali. Probabilistic Encryption. *In JCSS* 28(2):270-299, 1984.

[11] O. Goldreich, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18:186-208, 1989.