A Thirty Year Old Conjecture about Promise Problems*

Andrew Hughes[†] Debasis Mandal[‡] A. Pavan[§] Nathan Russell[¶]
Alan L. Selman[‖]

Abstract

Even, Selman, and Yacobi [ESY84, SY82] formulated a conjecture that in current terminology asserts that there do not exist disjoint NP-pairs all of whose separators are NP-hard via Turing reductions. In this paper we consider a variant of this conjecture—there do not exist disjoint NP-pairs all of whose separators are NP-hard via bounded-truth-table reductions. We provide evidence for this conjecture. We also observe that if the original conjecture holds, then some of the known probabilistic public-key cryptosystems are not NP-hard to crack.

1 Introduction

Even, Selman, and Yacobi [ESY84, SY82] conjectured that there do not exist certain promise problems all of whose solutions are NP-hard. Specifically, there do not exist disjoint NP-pairs all of whose separators are NP-hard. This conjecture has fascinating (and largely believable) consequences, including that NP differs from co-NP and NP is not equal to UP. Even though this conjecture is 30 years old, we do not know of any concrete evidence in support of the conjecture. (We don't know hypotheses that imply all of its consequences.) In this paper, we report some exciting progress on this conjecture. We consider variants of the conjecture and show that under some reasonable hypotheses, these variants of the conjecture hold.

A promise problem can be thought of as a disjoint pair—a pair of disjoint sets (Π_y, Π_n), Π_y is called the set of "yes" instances and Π_n is the set of "no" instances. Their union $\Pi_y \cup \Pi_n$ is called the *promise*. The motivation to study disjoint pairs or promise problems

*A preliminary version of this paper appeared in the Proceedings of the 39th International Colloquium on Automata, Languages, and Programming [HPRS12].

[†]University at Buffalo. ahughes6@buffalo.edu.

[‡]Iowa State University. debasis@iastate.edu. Research supported in part by NSF grants CCF-0916797 and CCF-1421163.

[§]Iowa State University. pavan@cs.iastate.edu. Research supported in part by NSF grants CCF-0916797 and CCF-1421163.

[¶]University at Buffalo. nrussell@buffalo.edu.

[‖]University at Buffalo. selman@buffalo.edu.

stems from their connections to a wide range of questions from diverse areas such as public-key cryptosystems, propositional proof systems, study of complete problems for semantic classes, and approximation algorithms. For a recent survey on promise problems, we refer the reader to a survey by Goldreich [Gol06].

For a promise problem (Π_y, Π_n), one is interested in the following computational question: Is there an efficient algorithm that tells whether an instance x lies in Π_y or not, under the promise that x is in $\Pi_y \cup \Pi_n$. The algorithm may give an arbitrary answer if the promise does not hold, i.e., $x \notin \Pi_y \cup \Pi_n$. More formally, a *solution/separator* of a promise problem is any set S that includes Π_y and is disjoint from Π_n. A promise problem is considered easy if it admits a solution in P and is hard if every solution is computationally difficult. The ESY conjecture concerns the computational difficulty of disjoint NP-pairs.

The ESY conjecture has some interesting implication regarding the hardness of public-key cryptosystems. Even, Selman, and Yacobi [ESY84] observed that the problem of cracking a public-key cryptosystem may not formalize as a straightforward decision problem, and it is more natural to formulate it as a promise problem. They associated a promise problem (Π_y, Π_n) to a model of public-key cryptosystems such that both Π_y and Π_n are in NP. A public-key cryptosystem that fits the model cannot be deemed secure, if the underlying promise problem admits at least one efficient solution. On the other hand, if every solution is NP-hard then the system is NP-hard to crack. Thus the ESY conjecture implies that public-key cryptosystems that fit the model are not NP-hard to crack. (We will discuss this further in a later section.)

The ESY conjecture is also related to the study of propositional proof systems [Raz94, Pud01]. Razborov observed that every propositional proof system f can be identified with a canonical disjoint NP-pair $(\mathrm{SAT}^*, \mathrm{REF}_f)$ where REF_f is the set of all formulas that have short proofs of unsatisfiability with respect to f. Conversely, Glaßer, Selman, and Zhang [GSZ07] showed that for every disjoint NP-pair (A, B) there is a proof system f such that (A, B) is many-one equivalent to $(\mathrm{SAT}^*, \mathrm{REF}_f)$. Because of this equivalence between propositional proof systems and disjoint NP-pairs, several interesting questions regarding propositional proof systems are related to the structure of disjoint NP-pairs. One of the open questions on propositional proof systems is whether optimal proof systems exist and the belief is that they do not exist. This question is related to the ESY conjecture. It is known that if optimal proof systems do not exist, then a variant of the ESY conjecture holds [GSSZ04].

In addition to connections with public-key cryptosystems and propositional proof systems, the ESY conjecture has several believable consequences in complexity theory. It is known that this conjecture implies NP differs from co-NP, NP differs from UP, and satisfying assignments of boolean formulas cannot be computed by single-valued NP machines (NPSV) [ESY84, GS88].

Given its relation to public key cryptosystems, propositional proof systems, and complexity theory, it is important to understand the power of the ESY conjecture. Is there a reasonable hypothesis that implies the conjecture? To date we do not know any reasonable hypotheses that imply the ESY conjecture. However, the analogue of the ESY conjecture

to the c.e. sets is a known theorem [Sch60]. It seems to be difficult to formulate reasonable hypotheses that imply the ESY conjecture, because of its wide range of consequences. Any hypothesis that implies the ESY conjecture immediately implies that NP $\neq$ co-NP, NP $\neq$ UP, and satisfying assignments of boolean formulas cannot be computed by single-valued NP machines (NPSV). None of the standard hypotheses used in complexity theory, such as PH is infinite, E has high circuit complexity, the measure of NP is not zero, and so on, are known to imply all of the above mentioned consequences. This seems to be the root difficulty.

In this paper we make progress toward this question. We consider variants of the ESY conjecture and show that under some reasonable hypotheses these variants follow. Note that the ESY conjecture states that every disjoint NP-pair has a solution that is not NP-hard via *adaptive reductions*. We can obtain variants of the conjecture by replacing adaptive reductions with less restrictive reductions. Given a reduction type r, the ESY-r conjecture states that every disjoint NP-pair has a solution that is not NP-hard via r-reductions. We know already that if we take r to be many-one reductions, then the ESY-m conjecture is equivalent to NP $\neq$ co-NP [GSSZ04]. What if we take r to be truth-table reductions or bounded-truth-table reductions?

We first observe that the ESY conjecture for truth-table reductions also has the same set of complexity theoretic consequences such as NP $\neq$ co-NP, NP $\neq$ UP, and satisfying assignments of boolean formulas cannot be computed by single-valued NP machines (NPSV). This suggests that obtaining evidence for the ESY-*tt* conjecture could be as hard as obtaining evidence for the original conjecture. In this paper we consider bounded-truth-table reductions; these are nonadaptive reductions that make a fixed number of queries.

The first main result of this paper is that if NP $\neq$ co-NP, then every disjoint NP-pair has a solution that is not NP-hard via length-increasing bounded-truth-table reductions (i.e., the ESY conjecture for *btt* length-increasing reductions holds). By using stronger hypotheses, we remove the length-increasing restriction. We show two sets of results. The first result shows that if NP contains certain type of generic sets, then every disjoint NP-pair has a solution that is not NP-hard via bounded-truth-table reductions. The second result concerns with the existence of one-way functions. We show that if there exist one-one, one-way functions that are hard to invert via circuits having an NP $\cap$ co-NP-oracle, then the ESY conjecture for $\leq^{\mathrm{P}}_{btt}$ reductions holds.

As noted earlier, one of the motivations for introducing the ESY conjecture was its relation to NP-hardness of public key cryptosystems. The analysis of Even, Selman, and Yacobi pertained to the deterministic public-key cryptosystems of that time. In the final section we observe that the ESY conjecture remains relevant to some of the current probabilistic public-key cryptosystems. If the cracking problem of a current public-key cryptosystem also can be formulated as a disjoint NP-pair, and the ESY conjecture holds, then these cryptosystems are not NP-hard to crack.

2 Preliminaries

We assume the standard lexicographic order on strings. We use $x-1$ to denote the immediate predecessor of x in this order. Given a language L and a string x, $L(x)$ denotes the characteristic function of L, and $L|x$ is defined as $L(\lambda)L(0)L(1)\cdots L(x-1)$.

The class QuasiNP is the set of languages that can be accepted by nondeterministic Turing machines running in quasi-polynomial-time, i.e., $\text{QuasiNP} = \cup_{c>0}\text{NTIME}(2^{\log^c n})$.

A language A is k-truth-table reducible to a language B (denoted $A \leq^{\text{P}}_{ktt} B$) if there exist two polynomial-time computable functions f and t such that for every x,

$$f(x) = \langle q_1, \cdots, q_k \rangle \text{ and } t(x, B(q_1), \cdots, B(q_k)) = A(x).$$

We say that A is bounded-truth-table reducible to B (denoted $A \leq^{\text{P}}_{btt} B$) if there exists a constant $k > 0$ such that $A \leq^{\text{P}}_{ktt} B$.

Definition 2.1. A function $f\colon \Sigma^* \to \Sigma^*$ is SNP *computable* if there is a nondeterministic polynomial-time bounded Turing machine M such that for every x at least one path of $M(x)$ outputs $f(x)$ and no path outputs a wrong answer. Some paths may output $\perp$.

We will also consider strong nondeterministic reductions. These reductions were originally defined by Adleman and Manders [AM77]. We slightly modify their definition to suit our purposes.

Definition 2.2. Let A and B be two languages. We say that A is *strong nondeterministic k-truth-table reducible* to B (denoted $A \leq^{\text{SNP}}_{ktt} B$), if there is a polynomial-time computable function f and an SNP computable function t such that for every x, $f(x) = \langle q_1, \cdots, q_k \rangle$, and $t(x, B(q_1), \cdots, B(q_k)) = A(x)$. We say that A is *strong nondeterministic bounded-truth-table reducible* to B ($A \leq^{\text{SNP}}_{btt} B$) if there exists a $k > 0$ such that $A \leq^{\text{SNP}}_{ktt} B$.

Remark. Note that in this definition, the reduction does not use nondeterminism to produce the queries. The original definition of Adleman and Manders [AM77] allows the query generator f also to be SNP computable. Similarly, we can define the notion of strong nondeterministic quasi-polynomial-time reductions.

Definition 2.3. We say that A is reducible to B via *length-increasing, strong nondeterministic, k-truth-table reductions (denoted $\leq^{\text{SNP}}_{ktt,li}$)* if $A \leq^{\text{SNP}}_{ktt} B$ and the length of every query is bigger than the length of the input.

Notions of length-increasing are defined similarly for $\leq^{\text{P}}_{ktt}$, $\leq^{\text{P}}_{btt}$, and $\leq^{\text{SNP}}_{btt}$-reductions.

2.1 ESY Conjecture

A *disjoint* NP*-pair* is a pair (A, B) of nonempty, disjoint sets A and B such that both A and B belong to NP. We let DisjNP denote the collection of all disjoint NP-pairs. We say that a set S is a *separator* for (A, B) if $A \subseteq S$ and $B \subseteq \overline{S}$. We now state the original conjecture of Even, Selman, and Yacobi [ESY84].

ESY Conjecture. For every pair of disjoint sets in NP, there is a separator that is not Turing hard for NP.

Although the original conjecture talks about Turing hardness, we can generalize it to arbitrary reductions. Let r be a reduction.

ESY-r Conjecture. For every pair of disjoint sets in NP, there is a separator that is not r-hard for NP.

Although the ESY conjecture stipulates a condition about arbitrary pairs of sets in NP, the following observation tells us that we can always take one of the sets to be SAT.

Observation 2.4. *The* ESY-r *conjecture is equivalent to the following statement: For every set B in* NP *that is disjoint from* SAT*, there is a separator of (B, SAT) that is not r-hard for* NP*.*

Proof. The direction from left to right is trivial. We will show the other direction holds, that the statement implies the ESY-r conjecture holds.

Assume that for every pair $(B, \mathrm{SAT}) \in \mathrm{DisjNP}$, there is a separator that is not r-hard for NP. Let $(C, D) \in \mathrm{DisjNP}$. Let f be a one-one, length-increasing polynomial-time reduction from D to SAT. Consider the pair $(f(C), \mathrm{SAT})$. Since f is length-increasing and polynomial-time computable, $f(C) \in \mathrm{NP}$. Since $x \in D \Leftrightarrow f(x) \in \mathrm{SAT}$, $f(C) \cap \mathrm{SAT} = \emptyset$. So $(f(C), \mathrm{SAT}) \in \mathrm{DisjNP}$. Let S be a separator for $(f(C), \mathrm{SAT})$ that is not r-hard for NP. Then $f(C) \subseteq S$ and $\mathrm{SAT} \subseteq \overline{S}$. Let

$$S' = \{y \mid f(y) \in S\}.$$

Observe that $f(C) \subseteq S$ gives $C \subseteq S'$ and $\mathrm{SAT} \subseteq \overline{S}$ gives $D \subseteq \overline{S'}$. Thus S' is a separator for (C, D).
Assume S' is r-hard for NP. Suppose $A \in \mathrm{NP}$. Then $A \leq_r S'$ and $x \in S' \Leftrightarrow f(x) \in S$, so we get $A \leq_r S$. This implies that S is also r-hard for NP. This contradicts our assumption that S was not r-hard for NP. Therefore S' is not r-hard for NP. Since, for any pair $(C, D) \in \mathrm{DisjNP}$ there is a separator that is not r-hard for NP, the ESY-r conjecture holds. □

Given any two types of reductions, if one reduction is stronger than the other, then there is a simple relation between the reductions and the ESY conjecture for those reductions.

Observation 2.5. *Let r and r', such that r'-hardness for* NP *implies r-hardness for* NP*. If the* ESY-r *conjecture holds then the* ESY-r' *conjecture holds.*

Proof. Suppose r'-hardness for NP implies r-hardness for NP. Suppose the ESY-r' conjecture does not hold. Then there is a disjoint NP-pair (A, B) such that all separators are r'-hard for NP. Then every separator of (A, B) must also be r-hard for NP. Therefore the ESY-r conjecture does not hold. □

The next observation is that the ESY-tt conjecture has the same set of consequences as the original ESY conjecture.

Observation 2.6. *The* ESY-*tt conjecture implies that* NP $\neq$ UP*,* NP $\neq$ co-NP*, and satisfying assignments of Boolean formulas cannot be computed by single-valued* NP *machines.*

Proof. Assume that NP = UP. Thus SAT is in UP and so let R be a relation that witnesses that SAT is in UP. Consider the following two disjoint languages in NP:

$$A = \{\langle x, i\rangle \mid \exists w\ R(x, w) = 1 \text{ and the } i\text{th bit of } w = 1\}$$

and

$$B = \{\langle x, i\rangle \mid \exists w\ R(x, w) = 1 \text{ and the } i\text{th bit of } w = 0\}.$$

Let S be any separator for (A, B). Below is a truth-table reduction from SAT to S. On input x, produce queries $\langle x, 1\rangle, \cdots \langle x, m\rangle$ (where m is the number of Boolean variables in the propositional formula x). If $\langle x, i\rangle \in S$, then set $a_i = 1$, else set $a_i = 0$. Accept x if and only if $R(x, a_1a_2 \cdots a_m) = 1$. Therefore SAT is $\leq^p_{tt}$-reducible to every separator of (A, B), so the ESY-*tt* conjecture does not hold.

A similar proof shows that if the ESY-*tt* conjecture holds, then satisfying assignments of Boolean values cannot be computed by single-valued NP-machines.

Lastly, assume that NP = co-NP. Then it follows that the ESY-*m* conjecture does not hold [GSSZ04], so the ESY-*tt* conjecture also does not hold, by Observation 2.5. □

The above observation suggests that providing evidence for the ESY-*tt* conjecture could be as difficult as providing evidence for the original conjecture. In this paper we consider the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture.

2.2 Unpredictability

Our results make use of the notion of *unpredictability*, which is similar to the notion of *genericity*.

Definition 2.7. We say that a nondeterministic machine M is *strong* if for every input x, exactly one of the following conditions hold: 1) at least one path of M accepts x and no path rejects, 2) at least one path of M rejects x and no path accepts. Some paths of the machine may output $\perp$.

Definition 2.8. Let M be a strong nondeterministic machine and L be a language. We say that M is a *predictor* for L if for every $x \in L$, M accepts $\langle x, L|x\rangle$ and for every $x \notin L$, M rejects $\langle x, L|x\rangle$.

Definition 2.9. Let $t(n)$ be any time bound. We say that a language L is SNTIME($t(n)$)-*unpredictable* if for every strong nondeterministic machine M that predicts L, every path of M runs for more than $t(n)$ time for *all but finitely many* inputs of the form $\langle x, L|x\rangle$.

Remark. The running time $t(n)$ of the predictor is in terms of the length of the input, which is $\langle x, L|x\rangle$. When measured in terms of the length of x, this time is roughly $t(|x| + 2^{|x|})$. The notion of unpredictability is very similar to the notion of *genericity* [ASFH87, ASNT96]. In

fact it is known that for deterministic computations, these two notions are equivalent [BM95]. Note that the definition of unpredictability requires "almost everywhere hardness", i.e., any predictor takes more than $t(n)$ time on all but finitely many inputs. Thus to show that a language is predictable, we only need exhibit a predictor that runs in time $t(n)$ only on infinitely many inputs.

Definition 2.10. Let A and L be two languages. We say that L is $\mathrm{SNTIME}(t(n))$-*unpredictable within* A if $L \subseteq A$ and for every strong nondeterministic machine M that predicts L, for all but finitely many x from A, M runs for more than $t(n)$ time on inputs of the form $\langle x, L|x\rangle$.

Our main results use the following theorem concerning the existence of unpredictable sets within $\overline{\mathrm{SAT}}$. This theorem is proved using strong diagonalization techniques. In particular, we can obtain this result by an application of Theorem 2.3 from [ASNT96] (by taking $f(n) = n$, $B = \overline{\mathrm{SAT}}$ and $t'(n) = 2^{2n}$). For the sake of completeness we provide a proof.

Theorem 2.11. *For every $k > 0$, there is a set R such that R is* $\mathrm{SNTIME}(2^{\log^k n})$-*unpredictable within* $\overline{\mathrm{SAT}}$.

Proof. We will proceed by considering unpredictability via deterministic machines. Consider definitions 2.8 and 2.9. We can make analogous definitions using deterministic machines and define a notion of $\mathrm{DTIME}(t(n))$ unpredictable. Let L be a language and say M is a strong nondeterministic predictor for L such that M runs in time $t(n)$ on infinitely many inputs of form $\langle x, L|x\rangle$. We can easily convert M into a deterministic predictor M' for L such that M' runs in $2^{t(n)}$ time for infinitely many inputs of the form $\langle x, L|x\rangle$. Thus L is not $\mathrm{DTIME}(2^{t(n)})$ unpredictable. So if a language L is $\mathrm{DTIME}(2^{t(n)})$ unpredictable, then it is $\mathrm{SNTIME}(t(n))$ unpredictable.

We will first note that known results imply the existence of $\mathrm{DTIME}(2^{2^{\log^k n}})$ unpredictable languages. It is known that if a language L is $t(n)$-generic, then it is $t(n)$-unpredictable [BM95, PS02]. In this paper we will not define the notion of genericity and refer the readers to the papers by Ambos-Spies *et al.* [ASFH87, ASNT96]. These papers also show that for every $t(n)$, there exist decidable languages that are $t(n)$-generic. Thus for every $k > 0$, there exist languages that are $\mathrm{DTIME}(2^{2^{\log^k n}})$ unpredictable. By our previous observation, for every $k > 0$, there exist languages that are $\mathrm{SNTIME}(2^{\log^k n})$ unpredictable.

Let L' be a language that is $\mathrm{SNTIME}(2^{\log^k n})$ unpredictable and let $L = L' - \mathrm{SAT}$. We will now claim that L is $\mathrm{SNTIME}(2^{\log^{k-1} n})$ unpredictable within $\overline{\mathrm{SAT}}$. Suppose not. Let M be a strong nondeterministic predictor for L such that for infinitely many strings $x \in \overline{\mathrm{SAT}}$, M on input $\langle x, L|x\rangle$ runs in time $2^{\log^{k-1} n}$. We will use M to build a predictor for L'. Note that L and L' only differ on strings from SAT. More precisely, if $y \notin \mathrm{SAT}$, then $y \in L$ if and only if $y \in L'$. If $y \in \mathrm{SAT}$, then by definition $y \notin L$. Thus given $L'|x$, one can construct $L|x$ by cycling through all strings $y < x$ and deciding their membership in SAT. This process takes $O(|\langle x, L'|x\rangle|)$ time. Let N be a strong nondeterministic machine that accepts L'.

1. Input $\langle x, L'|x\rangle$.

2. If $x \in \mathrm{SAT}$, then run $N(x)$ and output the result.

3. Construct $L|x$.

4. Run $M(\langle x, L|x\rangle)$.

5. Accept $\langle x, L'|x\rangle$ if $M(\langle x, L|x\rangle)$ accepts. Otherwise reject $\langle x, L'|x\rangle$.

For every $x \in \mathrm{SAT}$, the above predictor runs $N(x)$ and is thus correct. For every $x \notin \mathrm{SAT}$, $x \in L'$ if and only if $x \in L$. Since M is a correct predictor for L, the above predictor is correct on all $x \notin \mathrm{SAT}$. Thus the above machine is a predictor for L'.

By our assumption, for infinitely many x from $\overline{\mathrm{SAT}}$, M runs in $2^{\log^{k-1} n}$ time on input $\langle x, L|x\rangle$. Note that for every such x from $\overline{\mathrm{SAT}}$, the above predictor constructs $\langle x, L|x\rangle$ and runs $M(\langle x, L|x\rangle)$. Since construction of $\langle x, L|x\rangle$ (from $\langle x, L'|x\rangle$) takes $O(|\langle x, L'|x\rangle|)$ time, the above predictor runs in time $2^{\log^k n}$ for infinitely many x. This contradicts unpredictability of L'. □

2.3 Secure One-Way Functions and Pseudorandom Generators

An O-oracle circuit C, denoted by C^O, is a Boolean circuit with access to oracle O. Then we define the notion of secure one-way functions against O-oracle circuits.

Definition 2.12. A family of functions $\{f_n\}_{n\geq 1} : \Sigma^n \to \Sigma^{\ell(n)}$ is *one-way, $s(n)$-secure against oracle O*, if f is uniformly computable in polynomial time and for every nonuniform circuit C^O of size at most $s(n)$ and for sufficiently large n,

$$\Pr_{x\in\Sigma^n}[C^O(f(x)) \in f^{-1}(f(x))] \leq \frac{1}{s(n)}.$$

Given any function $f : \{0,1\}^* \to \{0,1\}$ and oracle O, the *circuit complexity of f relative to O-oracle at length n*, denoted by $C_f^O(n)$, is the size of the smallest O-oracle circuit that computes f on every input of size n. We say that a function $f : \{0,1\}^* \to \{0,1\}$ has circuit complexity $s(n)$ relative to an oracle O, if for all but finitely many n, $C_f^O(n) \geq s(n)$.

Definition 2.13. A *pseudorandom generator* is a function $G : \Sigma^{m(n)} \to \Sigma^n$ such that for every circuit C of size at most $O(n)$,

$$\left| \Pr_{x\in\Sigma^n}[C(x) = 1] - \Pr_{y\in\Sigma^{m(n)}}[C(G_n(y)) = 1] \right| \leq \frac{1}{8}.$$

Given an oracle O, G is said to be *secure against O-oracle* if the above inequality holds for all O-oracle circuits C^O of size at most $O(n)$, for almost all n.

All known constructions of pseudorandom generators are based on some hardness assumptions on the circuit complexity of a function. We need the following result due to Klivans and van Melkebeek [KvM02] (Theorem 3.4 in their paper.)

Theorem 2.14 ([KvM02]). *There exists a positive constant c such that the following holds for any oracle O, any function $f \in \mathrm{EXP}$, and any time constructible function $\ell : \mathbb{N} \to \mathbb{N}$: If for every n, the O-oracle circuit complexity of f at length $\ell(n)$ is at least n^c, then there is a pseudo-random generator $G : \Sigma^{O(\ell^2(n))} \to \Sigma^n$ that is secure against O-oracle. The running time of G is $2^{\ell^{2d}(n)}$ for some constant $d > 0$.*

We use the following instantiation of the above result obtained by taking $\ell(n)$ to be $(c \log n)^{1/\epsilon}$.

Theorem 2.15 ([KvM02]). *Let O be any language. If there is a constant $\epsilon > 0$ and a function f in* EXP *with circuit complexity at least 2^{n^ϵ} relative to O-oracle, then there is a constant $a > 0$ and a pseudorandom generator $G : \Sigma^{\log^a n} \to \Sigma^n$ that is secure against O-oracle. The running time of G is $2^{\log^b n}$ for some constant $b > 0$.*

3 ESY Conjecture for Bounded-Truth-Table Reductions

In this section we provide evidence for the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture. Before we present our results, we describe the ideas and intuition behind our proofs. Let (B, SAT) be a disjoint NP-pair. Our goal is to exhibit a separator S that is not NP-hard. One trivial way to achieve this is by making S to be an easy set—a set in P. However, this approach is not feasible because if NP differs from UP or P does not equal NP $\cap$ co-NP, then (B, SAT) does not have separators in P (for some $B \in \mathrm{NP}$) [GS88]. Thus we look for a separator that is not in P. Our first observation is that there exist "computationally difficult" sets that are not NP-hard, thus we can achieve our goal by taking S to be a difficult set.

It is known that if H is an unpredictable set, then H does not reduce to $H \cup B$ [LM96, ASB00, PS04]. This suggests that we can take $H \cup B$ as our separator and claim that it is not NP-hard. However, we run into at least two major problems. The set $H \cup B$ may not be disjoint from SAT and thus cannot be a separator. In fact, one can show that an unpredictable set H must have an infinite intersection with SAT. We get around this problem by taking H as an unpredictable set within $\overline{\mathrm{SAT}}$. This ensures that S is a separator.

The second, and the more serious problem, is that showing H does not reduce to $H \cup B$ does not imply that $H \cup B$ is not NP-hard as the set H may not be in NP. Instead of working with H, we will argue that SAT does not reduce to S. This argument makes a critical use of nondeterminism. We will show that if SAT does reduce to S, then either we can get a predictor for H or, by making use of nondeterminism, we can reduce the number of queries. Our first observation is that any reduction from SAT to S must infinitely often produce *relevant queries*—these are queries whose answers, given answers to all other queries, uniquely determine the output of the reduction. We then show that these relevant queries must lie outside of $B \cup \mathrm{SAT}$. If not, we can reduce the number of queries by making use of *strong nondeterminism*. Next we argue that if a query q is relevant, then knowing answers to all other queries help us determine the membership of $q \in S$, and if q lies outside of $B \cup \mathrm{SAT}$, then this contradicts the unpredictability of the set H.

3.1 Length-Increasing Reductions

In this subsection we prove that if NP does not equal co-NP, then the ESY conjecture holds for length-increasing bounded-truth-table reductions. In fact, we will show that the conjecture holds even for reductions that use nondeterminism.

Theorem 3.1. *If* NP $\neq$ co-NP*, then the* ESY-$\leq^{\mathrm{SNP}}_{btt,li}$ *conjecture is true.*

Proof. Suppose NP $\neq$ co-NP. Suppose the ESY-$\leq^{\mathrm{SNP}}_{btt,li}$ conjecture is false. Let (B, SAT) be a disjoint NP-pair. By observation 2.4, every separator of (B, SAT) is $\leq^{\mathrm{SNP}}_{btt,li}$-hard for NP. Let Q_1 and Q_2 be two polynomial-time computable relations for SAT and B respectively. Assume that the length of witnesses (for positive instances in SAT and in B) is bounded by n^r, $r > 0$. By Theorem 2.11 there is a set R that is $\mathrm{SNTIME}(2^{\log^{2r} n})$-unpredictable within $\overline{\mathrm{SAT}}$. Consider the separator $S = R \cup B$. Suppose that S is $\leq^{\mathrm{SNP}}_{ktt,li}$-hard for NP for some $k \geq 0$. We will achieve a contradiction to our hypothesis NP $\neq$ co-NP and to the fact that R is $\mathrm{SNTIME}(2^{\log^{2r} n})$-unpredictable within $\overline{\mathrm{SAT}}$. This will give us that S is not $\leq^{\mathrm{SNP}}_{ktt,li}$-hard for any $k \geq 0$ and therefore not $\leq^{\mathrm{SNP}}_{btt,li}$-hard for NP.

We prove this by induction. The base case is when the number of queries is zero. This means that there is an SNP computable function t such that $t(x) = \mathrm{SAT}(x)$. This implies that NP = co-NP, a contradiction.

As the inductive hypothesis, assume that S is not $\leq^{\mathrm{SNP}}_{(\ell-1)tt,li}$-hard. Now assume that SAT $\leq^{\mathrm{SNP}}_{\ell tt,li}$-reduces to S via $\langle f, t\rangle$, for contradiction. Given x, let $f(x) = \langle q_1, \cdots, q_\ell\rangle$. We assume that q_ℓ is the largest query and denote it with b_x. We say that a query q_i is *relevant* if the following holds:

$$t(x, S(q_1), \cdots, S(q_i), \cdots, S(q_\ell)) \neq t(x, S(q_1), \cdots, \overline{S(q_i)}, \cdots, S(q_\ell)).$$

In other words, if q_i is relevant then knowing answers to all the other queries still does not help us determine $\mathrm{SAT}(x)$.

Observation 3.2. *There exist infinitely many x such that b_x is relevant.*

Proof. Suppose not. Then for all but a finite number of x we can remove b_x from the list of queries, giving an $\leq^{\mathrm{SNP}}_{(\ell-1)tt,li}$-reduction from SAT to S, and this contradicts the induction hypothesis. □

Let

$$T = \{x \mid \ b_x \text{ is relevant}\}.$$

Lemma 3.3. *There exist infinitely many $x \in T$ such that $b_x \notin B \cup \mathrm{SAT}$.*

Proof. Suppose not. For all but finitely many $x \in T$, the query b_x is relevant and belongs to $B \cup \mathrm{SAT}$. Now consider the following reduction $\langle f', t'\rangle$ from SAT to S: on input x, f' will first compute $f(x) = \langle q_1, q_2, \cdots, q_{\ell-1}, b_x\rangle$ and outputs the queries $\langle q_1, \cdots, q_{\ell-1}\rangle$. We now describe t':

1. Let $b_1 = S(q_1), \cdots, b_{\ell-1} = S(q_{\ell-1})$.

2. Determine whether b_x is relevant or not by comparing $t(x, b_1, \cdots, b_{\ell-1}, 0)$ with $t(x, b_1, \cdots, b_{\ell-1}, 1)$. If b_x is not relevant, then output $t(x, b_1, \cdots, b_{\ell-1}, 0)$.

3. Guess a witness $w \in \Sigma^{n^r}$. If $Q_1(b_x, w)$ holds, then output $t(x, b_1, \cdots, b_{\ell-1}, 0)$.

4. If $Q_1(b_x, w)$ does not hold, then guess a witness $u \in \Sigma^{n^r}$. If $Q_2(b_x, u)$ holds then output $t(x, b_1, \cdots, b_{\ell-1}, 1)$, else output $\perp$.

We claim that the above is an $\leq^{\mathrm{SNP}}_{(\ell-1)tt,li}$-reduction from SAT to S. Clearly f' produces only $\ell - 1$ queries. If b_x is not relevant, then the reduction is correct. Suppose that b_x is relevant. By our assumption $b_x \in B \cup \mathrm{SAT}$. If $b_x \in \mathrm{SAT}$, then $b_x \notin S$. Thus $t(x, b_1, \cdots, b_{k-1}, 0) = \mathrm{SAT}(x)$. If $b_x \in B$, then $b_x \in S$. Thus $t(x, b_1, \cdots, b_{k-1}, 1) = \mathrm{SAT}(x)$. Thus the reduction is always correct.

It remains to show that this is an SNP-reduction. Clearly all queries are produced by a deterministic polynomial-time process. Step 2 computes the function t. However t is SNP-computable. So this step can be done via an SNP-machine. Suppose $b_x \in \mathrm{SAT}$. Then there is a $w \in \Sigma^{n^r}$ such that $Q_1(b_x, w)$ holds, and thus this path outputs the correct answer. Since SAT is disjoint from B, for every $u \in \Sigma^{n^r}$, $Q_2(b_x, u)$ does not hold. Thus no path outputs the wrong answer. A similar argument shows that when $b_x \in B$, at least one path outputs the correct answer and no path outputs the wrong answer.

Thus SAT $\leq^{\mathrm{SNP}}_{(\ell-1)tt,li}$-reduces to S. This contradicts our induction hypothesis. This completes the proof of the lemma. □

Now, we return to the proof of the theorem. Lemma 3.3 has the following corollary.

Corollary 3.4. *There exist infinitely many $y \notin B \cup \mathrm{SAT}$ with the following property: There exists an x, $|x| < |y|$ such that $y = b_x$ and y is relevant.*

This enables us to build the following predictor for R. Let M be a strong nondeterministic algorithm that decides R.

1. Input $\langle y, R|y \rangle$.

2. If $y \in B \cup \mathrm{SAT}$, then run $M(y)$ and output the result.

3. Search for an x such that $|x| < |y|$ and $b_x = y$. If no such x is found run $M(y)$ and output the result.

4. Let $f(x) = \langle q_1, \cdots, q_{\ell-1}, y \rangle$. Compute $b_i = S(q_i)$, $1 \leq i \leq \ell - 1$ by

   (a) Decide the membership of $q_i \in B$, by running a brute force algorithm for B

   (b) Decide the membership of $q_i \in R$, by looking at $R|y$.

5. Check if y is relevant or not by comparing $t(x, b_1, \cdots, b_{\ell-1}, 0)$ and $t(x, b_1, \cdots, b_{\ell-1}, 1)$. If y is not relevant, then run $M(y)$ and output the result.

6. Now we know that y is relevant. Compute $\mathrm{SAT}(x)$. Find the unique bit b such that $\mathrm{SAT}(x) = t(x, b_1, \cdots, b_{\ell-1}, b)$.

7. Accept if and only if b equals 1.

Claim 3.5. *The above predictor correctly predicts R and for infinitely many strings from $\overline{\mathrm{SAT}}$ runs in time $2^{\log^{2r} n}$.*

Proof. Let I be the set of all y for which the conditions of Corollary 3.4 holds. The above predictor runs $M(y)$ on any y that is not in I and thus is correct on all such y. Let $y \in I$. We know that $\mathrm{SAT}(x) = t(x, b_1, \cdots, b_{\ell-1}, S(y))$. Since y is relevant $\mathrm{SAT}(x) \neq t(x, b_1, \cdots, b_{\ell-1}, \overline{S(y)})$. Thus $b = S(y)$. Since $y \notin B \cup \mathrm{SAT}$, $y \in S$ if and only if $y \in R$. Thus the above predictor correctly decides every y in I.

Now we will show that for every $y \in I$, the above predictor halts in quasi-polynomial time. Let $|y| = m$, note that the length of x found in step 3 is at most m. Checking for membership of y in $B \cup \mathrm{SAT}$ takes $O(2^{m^r})$ time. Since $y = b_x$ is the largest query produced, $|q_i| \leq m$, $1 \leq i \leq \ell - 1$. Since B can be decided in time 2^{n^r}, Step 4a takes $O(2^{m^r})$ time. Since $y > q_i$, $1 \leq i \leq \ell - 1$, Step 4b takes polynomial time. Computing $\mathrm{SAT}(x)$ takes $O(2^m)$ time. The predictor computes the function t. However t is SNP computable. Thus the total time taken is $O(2^{m^{r+1}})$. Note that the run time of the predictor is measured in terms of length of $\langle y, R|y\rangle$ which is at least 2^m. Thus for every $y \in I$, the predictor runs in time $2^{\log^{2r} n}$ time. Since I is an infinite set and by definition is a subset of $\overline{\mathrm{SAT}}$, the claim follows. □

We have shown that S is not $\leq^{\mathrm{SNP}}_{\ell tt,li}$-hard for NP. This completes the induction step. Thus S is not $\leq^{\mathrm{SNP}}_{btt,li}$-hard for NP. This completes the proof of the Theorem. □

Our main result of this subsection is a corollary of the above theorem.

Theorem 3.6. NP $\neq$ co-NP *if and only if the* ESY-$\leq^{\mathrm{P}}_{btt,li}$ *conjecture holds.*

Proof. Suppose NP $\neq$ co-NP. Since every length-increasing bounded-truth-table reduction is trivially an $\leq^{\mathrm{SNP}}_{btt,li}$-reduction, by Theorem 3.1 our result holds.

Suppose NP = co-NP. Then $(\mathrm{CNFSAT}, \overline{\mathrm{CNFSAT}}) \in \mathrm{DisjNP}$. The only separator is CNFSAT. We know that CNFSAT is $\leq^{\mathrm{P}}_{m\text{-}li}$-complete for NP. This can be shown via the Cook-Levin reduction, which given input x outputs a formula ϕ of length $\Theta(p_i(|x|) \log p_i(|x|))$, where p_i is the runtime of N_i from the standard enumeration of NP machines. Therefore the ESY-$\leq^{\mathrm{P}}_{m\text{-}li}$ conjecture doesn't hold, so the ESY-$\leq^{\mathrm{P}}_{btt,li}$ conjecture also doesn't hold by Observation 2.5. □

3.2 General Reductions

In this subsection we relax the length-increasing requirement and consider general bounded-truth-table reductions. We show two results. The first result shows that if NP contains unpredictable sets, then the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture holds without the length-increasing restriction. The second result states that if certain one-way functions exist, then the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture holds.

3.2.1 Unpredictable Sets

Theorem 3.7. *If* NP *has a* SNTIME(n^2) *unpredictable set, then the* ESY-$\leq^{\mathrm{P}}_{btt}$ *conjecture holds.*

Proof. Let (B, SAT) be a disjoint NP-pair. As before, let Q_1 and Q_2 be polynomial-time computable relations associated with SAT and B, and say that the length of a witness is bounded by n^r. Let G' be a set in NP that is SNTIME(n^2) unpredictable. Let

$$G = \{x \mid 0^{n^{2r}}x \in G', |x| = n\}.$$

Clearly, G is in NP. Using the reverse padding trick [ASTZ97], we can show that G is SNTIME($2^{\log^{2r} n}$) unpredictable. For the sake of completeness, we provide a proof sketch. Assume that G is SNTIME($2^{\log^{2r} n}$) predictable and let P be a predictor. The predictor P' for G' works as follows. Let y be a string of the form $0^{n^{2r}}x$ ($|x| = n$). On input $\langle y, G'|y\rangle$, it first extracts $\langle x, G|x\rangle$ and runs P on $\langle x, G|x\rangle$. It accepts if and only if P accepts. Note that the length of $\langle y, G'|y\rangle$ is $m = O(2^{n+n^{2r}} + n + n^{2r})$ and the length of $\langle x, G|x\rangle$ is $O(2^n + n)$. Since P is $2^{\log^{2r} n}$ time-bounded, it runs for $O(2^{2n^{2r}})$ time on input $\langle x, G|x\rangle$. Thus the running time of P' is bounded by $O(2^{2n^{2r}})$ and this is linear in the input length (which is $O(2^{n+n^{2r}} + n + n^{2r})$). Thus G' is n^2 predictable.

Now let

$$G_0 = \{x \mid x0 \in G\} \text{ and } G_1 = \{x \mid x1 \in G\}.$$

Observation 3.8. *Since* G *is* SNTIME($2^{\log^{2r} n}$) *unpredictable neither* G_0 *nor* G_1 *is in* NP $\cap$ co-NP.

Proof. Suppose G_0 is in NP $\cap$ co-NP. This lets us build the following predictor for G. Let M be a strong nondeterministic algorithm that decides G, let R_1, R_2 be polynomial relations associated with G_0 and $\overline{G_0}$, respectively, and say lengths of the witnesses are bounded by $n^s, s > 0$, and the running time is bounded by $n^k, k > 0$, for R_1 and R_2.

1. Input $\langle y, G|y\rangle$.
2. If $y = x1$ or $y = \lambda$, then run $M(y)$ and output the result.
3. Otherwise $y = x0$.
4. Guess a witness $w \in \Sigma^{n^s}$. If $R_1(x, w)$ holds then accept $\langle y, G|y\rangle$.
5. If $R_1(x, w)$ does not hold, then guess a witness $u \in \Sigma^{n^s}$. If $R_2(x, u)$ holds then reject $\langle y, G|y\rangle$, else output $\perp$.

Clearly if y is of the form $x1$ or λ, the algorithm runs correctly. If $y = x0$, then the algorithm nondeterministically chooses a witness for each of the polynomial-time relations and computes correctly the membership in G_0, and therefore G, in time $O(n^k)$, or outputs $\perp$ if the choice of witness doesn't provide a distinct solution. Therefore the predictor is correct and on infinitely many inputs runs in time $O(n^k)$, which is clearly within $2^{\log^{2r} n}$, contradicting that G is SNTIME($2^{\log^{2r} n}$)-unpredictable.

A similar argument shows that G_1 cannot be in NP $\cap$ co-NP. $\square$

Let $G' = G_1 - \mathrm{SAT}$. Let S be the separator $G' \cup B$. We now claim that S is not $\leq^{\mathrm{P}}_{btt}$-hard for NP. Our proof again proceeds by induction and we will actually prove a stronger claim. We will prove that for every $k > 0$, G_0 is not $\leq^{\mathrm{SNP}}_{ktt}$ reducible to S. Since G_0 is in NP, this would show that S is not $\leq^{\mathrm{SNP}}_{ktt}$-hard for NP.

The base case is when the number of queries produced is zero. In this case there exists an SNP-computable function t and $G_0(x) = t(x)$ for every x. This implies that G_0 is in $\mathrm{NP} \cap \text{co-NP}$ and this is a contradiction.

As the inductive hypothesis, assume that G_0 does not $\leq^{\mathrm{SNP}}_{(\ell-1)tt}$ reduce to S. We will now prove that G_0 does not $\leq^{\mathrm{SNP}}_{\ell tt}$ reduce to S. Assume $G_0 \leq^{\mathrm{SNP}}_{\ell tt}$ reduces to S. Let $\langle f, t\rangle$ be one such reduction from G_0 to S. Given x, let Q_x denote the set of queries produced by $f(x)$ and let b_x be the largest query.

Claim 3.9. *For all but finitely many x, $x0 < b_x1$ in the standard lexicographic order.*

Proof. Suppose there exist infinitely many strings such that $x0 > b_x1$. Since b_x is the largest query this means that for every $q \in Q_x$ we have $x0 > q1$. This yields the following predictor for G: The input is $\langle y, G|y\rangle$. If y is not of the form $x0$ run a deterministic algorithm that decides G. Say $y = x0$, compute the set $Q_x = \{q_1, \cdots, q_\ell\}$. Consider a $q \in Q_x$. We can determine its membership in B by running a brute force algorithm. Since $q \leq q_\ell$, $q_\ell = b_x$, and $b_x1 < x0$, we have $q1 < x0$. Thus for every $q \in Q_x$, we can decide the membership of $q \in G_1$, by looking at $G|y$. Finally for every $q \in Q_x$ decide the membership of $q \in \mathrm{SAT}$. This enables us to compute the membership of $q \in S$ for every $q \in Q_x$. Now compute $G_0(x)$ by computing $t(S(q_1), \cdots S(q_\ell))$. This will tell whether x in G or not. If y is of length m, since t is SNP computable, then the total time taken is at most $2^{m^{2r}}$. Thus for infinitely many strings the predictor runs in time $2^{\log^{2r} n}$ time and this contradicts the unpredictability of G. □

As before we consider whether a query is relevant or not and we have the following observation.

Observation 3.10. *There exist infinitely many x such that b_x is relevant.*

Proof. Suppose not, then there is a $\leq^{\mathrm{SNP}}_{(\ell-1)tt}$ reduction from G_0 to S and this contradicts the inductive hypothesis. □

Let

$$T = \{x \mid b_x \text{ is relevant}\}.$$

Lemma 3.11. *There exist infinitely many $x \in T$ such that $b_x \notin B \cup \mathrm{SAT}$.*

The proof is very similar to the proof of Lemma 3.3. This gives the following corollary.

Corollary 3.12. *There exist infinitely many $y \notin B \cup \mathrm{SAT}$ with the following property: There exists an x, $x < y$, such that $b_x = y$ and y is relevant.*

We will now describe a predictor for G. Let M be a strong nondeterministic algorithm that decides G. The predictor gets $\langle z, G|z\rangle$ as input. Let $z = yb$. If $b = 0$ or $y \in B \cup \mathrm{SAT}$, or if there is no $x < y$ with $b_x = y$, then run $M(z)$. From now we assume that $z = y1$ and $y \notin B \cup \mathrm{SAT}$. Let $x < y$ be a string such that $b_x = y$.

1. Let $f(x) = \langle q_1, \cdots, q_{\ell-1}, y\rangle$.
2. Compute $b_i = S(q_i)$, $1 \le i \le \ell - 1$ by
   (a) Decide the membership of $q_i \in B$, by running a brute force algorithm for B.
   (b) If $q_i \in \mathrm{SAT}$, then $q_i \notin G'$. If $q_i \notin \mathrm{SAT}$, then by the definition of G', $q_i \in G'$ if and only if $q_i 1 \in G$. Since $q_i < y$, $q_i 1 < y1$. Decide the membership of $q_i 1 \in G$ by looking at $G|y1$. This determines the membership of q_i in G'.
3. Check if y is relevant or not by comparing $t(x, b_1, \cdots, b_{\ell-1}, 0)$ and $t(x, b_1, \cdots, b_{\ell-1}, 1)$. If y is not relevant, then run $M(z)$ and output the result.
4. Compute $G_0(x)$ (by looking at $\langle z, G|z\rangle$) and find the unique bit b such that $G_0(x) = t(x, b_1, \cdots, b_{\ell-1}, b)$.
5. Accept if and only if b equals 1.

We now claim that this predictor is correct and runs in $2^{\log^{2r} n}$ time on infinitely many strings. The proof is very similar to the proof of Claim 3.5, so we omit the details. This concludes the proof of Theorem 3.7. □

Power of the Hypothesis. We will now make a few remarks about the hypothesis in the above theorem and connect it to the earlier used hypotheses. We will first make a few informal observations. The notion of unpredictability attempts to capture the difficulty of a language, given some auxiliary information: For a language L how easy/difficult is it to determine membership of $x \in L$, *given* $L|x$ *as auxiliary information*? Many natural problems turn out to be very easy in this model. For example, consider SAT. We believe that there is no polynomial-time algorithm that decides SAT. However, we can easily decide the membership of a formula $\phi(x_1, \cdots, x_n)$ if we know the memberships of $\phi(x_1, \cdots, x_{n-1}, 0)$ and $\phi(x_1, \cdots, x_{n-1}, 1)$. Thus for *every* formula ϕ, given access to the partial characteristic sequence $\mathrm{SAT}|\phi$, we can decide the membership of ϕ in time that is polynomial in the length of ϕ (and logarithmic in the length of $\langle \phi, \mathrm{SAT}|\phi\rangle$). Do there exist languages that are difficult even when the partial characteristic sequence is given as auxiliary input? It turns out that EXP, somewhat surprisingly, contains such languages. Our hypothesis asserts that NP also contains such languages.

More formally, we can connect this hypothesis to known hypotheses. Say that a language is NP∩co-NP bi-immune if every strong nondeterministic machine that decides L takes more than polynomial time on all but finitely many inputs. It is easy to see that if our hypothesis holds, then NP contains NP ∩ co-NP bi-immune sets.

Our hypothesis is similar to, but stronger than, the genericity hypothesis of Ambos-Spies et al. The genericity hypothesis asserts that NP contains n^2-generic languages. This hypothesis is shown to have several interesting and believable consequences [ASFH87] [ASNT96]. In the definition of unpredictability, if we replace the strong nondeterministic machines with deterministic machines, then it coincides with genericity [BM95]. That is, the statements "L is DTIME($t(n)$) unpredictable" and "L is $t(n)$-generic" are equivalent. Since our hypothesis concerns strong nondeterministic predictors, our hypothesis can be taken as "NP contains SNTIME(n^2) (or simply NP $\cap$ co-NP) generic sets."

3.2.2 One-Way Functions

Here we show that if there exist one-one, one-way functions that are 2^{n^ϵ}-secure (for some $\epsilon > 0$) against NP $\cap$ co-NP oracle circuits, then the ESY conjecture for bounded-truth-table reductions holds. The starting point for this result is Theorem 3.1, which states that if NP is not the same as co-NP, then the ESY conjecture holds for nondeterministic, length-increasing bounded-truth-table reductions. We first extend the notion of SNP-reductions to strong, nondeterministic, quasi-polynomial-time reductions. We call a function *polynomially-bounded* if the length of the output of the function is bounded by a polynomial in input length.

Definition 3.13. Let A and B be two languages. We say that A is *strong, nondeterministic, quasi-polynomial k-truth table reducible* to B (denoted $A \leq_{ktt}^{\mathrm{SNQP}} B$) if there is a polynomially-bounded, quasi-polynomial-time computable function f and a strong nondeterministic quasi-polynomial-time computable function t such that for every x, $f(x) = \langle q_1, \cdots, q_k \rangle$, and $t(x, B(q_1), \cdots, B(q_k)) = A(x)$.

We call a $\leq_{ktt}^{\mathrm{P}}$-reduction *weakly-length-increasing*, if for every input x, the reduction outputs at least one query whose length is larger than the length of x. Suppose every language that is $\leq_{ktt}^{\mathrm{SNP}}$-hard for NP is hard via weakly-length-increasing $\leq_{ktt}^{\mathrm{SNP}}$-reductions. We first observe that under this assumption, the ESY-$\leq_{btt}^{\mathrm{P}}$ conjecture holds if NP differs from co-NP.

Observation 3.14. *Suppose that every set A that is $\leq_{ktt}^{\mathrm{SNP}}$-hard for* NP *is hard for* NP *via weakly-length-increasing $\leq_{ktt}^{\mathrm{SNP}}$-reductions. Then* co-NP *is a not a subset of* NP *if and only if the* ESY-$\leq_{btt}^{\mathrm{P}}$ *conjecture holds.*

Proof. Consider the proof of Theorem 3.1. The only places where we require the length-increasing part of the $\leq_{\ell tt,li}^{\mathrm{SNP}}$-reduction are in the proof of Lemma 3.3 and in the runtime analysis of Claim 3.5. Suppose we have a $\leq_{\ell tt}^{\mathrm{SNP}}$-reduction from SAT to set S. Then by our assumption, since S is $\leq_{\ell tt}^{\mathrm{SNP}}$-hard for NP, we have a weakly-length-increasing $\leq_{\ell tt}^{\mathrm{SNP}}$-reduction $\langle f, t \rangle$ from SAT to S, such that f is a query generator that generates ℓ queries. Let f' be the query generator that produces all queries of f except the largest query. Thus, there is a t', as described in the proof of Lemma 3.3, such that $\langle f', t' \rangle$ is an $\leq_{\ell-1tt}^{\mathrm{SNP}}$-reduction from SAT to S. However, in this case $\langle f', t' \rangle$ is not necessarily a weakly-length-increasing reduction, but is still an $\leq_{(\ell-1)tt}^{\mathrm{SNP}}$-reduction. This gives us that S is $\leq_{(\ell-1)tt}^{\mathrm{SNP}}$-hard for NP. By our assumption,

there must exist another $\leq^{\mathrm{SNP}}_{(\ell-1)tt}$-reduction $\langle f'', t'' \rangle$, where f'' produces at least one query whose size is larger than the input size. Then with these changes, the proof of Lemma 3.3 goes through, the statement of Corollary 3.4 still holds, and the runtime analysis of predictor M holds in Claim 3.5. □

Our next observation is the following. The proof of Observation 3.14 goes through if we replace strong nondeterministic *polynomial-time* reductions with strong nondeterministic *quasi-polynomial-time* reductions and strengthen the hypothesis to include co-NP is not a subset of QuasiNP.

Observation 3.15. *Suppose that every set A that is $\leq^{\mathrm{SNQP}}_{ktt}$-hard for* NP *is hard for* NP *via weakly-length-increasing $\leq^{\mathrm{SNQP}}_{ktt}$-reductions, and suppose that* co-NP *is not a subset of* QuasiNP*, then the* ESY-$\leq^{\mathrm{P}}_{btt}$ *conjecture holds.*

Next we will show that assuming the existence of a certain kind of one-way function implies both the hypotheses of Observation 3.15 hold.

Observation 3.16. *Suppose that there exist an $\epsilon > 0$ and a one-one, one-way function that is 2^{n^ϵ}-secure against* NP ∩ co-NP *oracle circuits, then* co-NP *is not a subset of* QuasiNP*.*

Proof. Every one-way function can be trivially inverted in polynomial-time with an NP-oracle. Suppose that co-NP is a subset of QuasiNP, then NP is a subset of both QuasiNP and co-QuasiNP. Thus every one-way function can be inverted in polynomial-time with a QuasiNP ∩ co-QuasiNP oracle. We now use a padding argument to show that every one-way function can be inverted by quasi-polynomial size NP ∩ co-NP-oracle circuits. Let f be a one-way function and M be a polynomial-time algorithm that inverts f with access to a language O that is in QuasiNP ∩ co-QuasiNP. Let r be a constant such that both O and $\overline{O}$ can be decided by nondeterministic algorithms running in time $O(2^{\log^r n})$ time. Define

$$O' = \{\langle x, 0^m \rangle \mid |x| = n, m = 2^{\log^r n}, \text{ and } x \in O\}.$$

Clearly O' is in NP ∩ co-NP. Consider the following algorithm M' with O' as oracle: On any input y, simulate $M(y)$. When M makes a query q (of length m) to O, make a query $\langle q, 0^{2^{\log^r m}} \rangle$ to oracle O'. Note that the time taken to form this query is $O(2^{\log^r m})$, since m is polynomial in the input length and M makes at most polynomially many queries. Thus the total running time of M' is $O(2^{\log^{r'} n})$ for some constant $r' > r$. We can convert this algorithm into a circuit of size $O(2^{\log^{r''} n})$ for some constant $r'' > r'$. This contradicts the hypothesis. □

Now we show that the existence of the above 2^{n^ϵ}-secure one-way functions implies the first hypothesis of Observation 3.15.

Theorem 3.17. *Suppose that there exist an $\epsilon > 0$ and a one-one, one-way function that is 2^{n^ϵ}-secure against* NP ∩ co-NP *oracle circuits. Then every $\leq^{\mathrm{SNQP}}_{ktt}$-hard language for* NP *is hard for* NP *via weakly-length-increasing $\leq^{\mathrm{SNQP}}_{ktt}$-reductions.*

The above theorem is the heart of this discussion. We will momentarily postpone the proof of it. The following main theorem follows by Observations 3.15, 3.16 and Theorem 3.17.

Theorem 3.18. *Suppose that there exist an $\epsilon > 0$ and a one-one, one-way function that is 2^{n^ϵ}-secure against* NP $\cap$ co-NP *oracle circuits. Then the* ESY-$\leq^{\mathrm{P}}_{btt}$ *conjecture holds.*

The rest of this section is devoted to the proof of Theorem 3.17. Agrawal [Agr02] and Agrawal and Watanabe [AW09] showed that if one-one, one-way functions exist, then all NP-complete sets are complete via nonuniform, one-one, and length-increasing reductions. Our proof heavily relies on the ideas in those papers. For the sake of simplicity, we prove the theorem for polynomial-time reductions. It can be verified easily that the proof holds for quasi-polynomial-time reductions.

First, we give the proof ideas of Theorem 3.17 for polynomial-time many-one ($\leq^p_m$) reductions. The proof proceeds in three steps. Suppose f_o is a 2^{n^ϵ}-secure one-way function from the hypothesis and L is a $\leq^p_m$-hard language for NP, via a $\leq^p_m$ reduction g. Let A be a language in NP that reduces to L via g. Now informally, define a $\leq^p_m$ reduction to be "sparse" on a set S if the number of strings that are mapped to any single string in L is sufficiently "small" (to be defined formally later). In general, the $\leq^p_m$ reduction g need not be sparse. The first step in our proof is to show that there is a sparse $\leq^p_m$-reduction from A to L, for every language A in NP. The notion of Goldreich-Levin *hardcore bit* [GL89] is critical in this proof. Next, we show that there is a randomized length-increasing $\leq^p_m$-reduction h from A to L, for every language A in NP. To prove this, we pad every input string x with a randomly selected "sufficiently large" (to be defined later) string r and apply the above sparse $\leq^p_m$ reduction on $\langle x, r\rangle$. The sparsity of the $\leq^p_m$ reduction helps us to bound the number of random strings r for which the reduction $h(\langle x, r\rangle)$ is not length-increasing for every input x. Finally, we derandomize this reduction using an appropriate pseudorandom generator.

Now we give the proof details.

Proof of Theorem 3.17. Let $\{f_o\}_{n\geq 1} : \Sigma^n \to \Sigma^{\ell(n)}$ be a family of one-one, one-way functions from the hypothesis. Let L be an $\leq^{\mathrm{SNP}}_{ktt}$-hard language for NP. We will view an $\leq^{\mathrm{SNP}}_{ktt}$-reduction $\langle f, t\rangle$ as a function from Σ^* to $(\Sigma^*)^k \times T_k$, where T_k is the set of all truth-tables for k variables. Let us denote this function by $F_{f,t}$.

Definition 3.19. A truth-table-reduction $\langle f, t\rangle$ is *α-sparse* on a set $S \subseteq \Sigma^n$ if for every x_0 in S,

$$\Big|\{x \in \Sigma^n \mid F_{f,t}(x) = F_{f,t}(x_0)\}\Big| \leq \frac{2^n}{2^{n^\alpha}}.$$

First, we will prove that from every language A in NP, there is a sparse $\leq^{\mathrm{SNP}}_{ktt}$-reduction to L. Define A^n to be the set of strings of length n from the set A.

Lemma 3.20. *There exists a $\gamma > 0$ such that for every language A in* NP*, there is an $\leq^{\mathrm{SNP}}_{ktt}$-reduction from A to L that is γ-sparse on A^n for every $n \geq 0$.*

Proof. For the sake of simplicity, we assume that the language A is defined only on even length strings. If that were not the case, then we can work with $A' = A \cdot A$. Using the one-way function f_o, we first define the hard-core function of Goldreich-Levin [GL89]. Given a $2n$ bit string xy such that $x, y \in \Sigma^n$, $f_{gl}(xy) = \langle f_o(x), y, x \oplus y \rangle$. Since f_o is one-one, f_{gl} is one-one. Then the following lemma can be obtained by relativizing the hard-core bit theorem [GL89, HILL99] (In particular Proposition 4.5 from [HILL99]).

Lemma 3.21. *There exists a γ ($< \epsilon$) such that for every sufficiently large n, for every oracle O in* NP $\cap$ co-NP*, and for every O-oracle circuit D of size at most 2^{n^γ},*

$$\left| \Pr_{x,y \in \Sigma^n}[D(f_o(x), y, x \oplus y) = 1] - \Pr_{x,y \in \Sigma^n, b \in \{0,1\}}[D(f_o(x), y, b) = 1] \right| \leq \frac{1}{2^{(3n)^\gamma}}.$$

Let $B = \{f_{gl}(w) \mid w \in A\}$. Since f_{gl} is one-one and is length-increasing, we have that B is in NP. Since B is in NP, there is an $\leq^{\mathrm{SNP}}_{ktt}$-reduction $\langle g, t \rangle$ from B to L. Thus $\langle f, t \rangle$ is a reduction from A to L, where $f = g \circ f_{gl}$. Note that there is a language O in NP $\cap$ co-NP such that f can be computed in polynomial-time with oracle access to O.

Now we will establish that $\langle f, t \rangle$ is γ-sparse on A^{2n}. Suppose not, then there exists a string $w_0 \in A^{2n}$ such that the size of the following set S is bigger than $\frac{2^{2n}}{2^{(2n)^\gamma}}$,

$$S = \{w \in \Sigma^{2n} \mid F_{f,t}(w) = F_{f,t}(w_o)\}.$$

We make the following observation.

Observation 3.22. *Since f_{gl} is one-one, $|S| = |f_{gl}(S)|$ and $F_{g,t}(f_{gl}(S)) = \{F_{f,t}(w_0)\}$.*

Define an O-oracle circuit D that on an input string z of length $\ell(n) + n + 1$ behaves as follows: If $F_{g,t}(z) = F_{f,t}(w_0)$ then accept, otherwise reject. The next observation follows from the definition of S and Observation 3.22.

Observation 3.23. *The circuit D accepts a string $z \in \Sigma^{\ell(n)+n+1}$ if and only if $z \in f_{gl}(S)$.*

By Observations 3.22 and 3.23,

$$\Pr_{x,y \in \Sigma^n}[D(f_o(x), y, x \oplus y) = 1] = p_n \geq \frac{1}{2^{(2n)^\gamma}}.$$

On the other hand,

$$\begin{aligned}
& \Pr_{x,y \in \Sigma^n, b \in \{0,1\}}[D(f_o(x), y, b) = 1] \\
= & \Pr_{x,y \in \Sigma^n, b \in \{0,1\}}[b = x \oplus y] \times \Pr_{x,y \in \Sigma^n, b \in \{0,1\}}[D(f_o(x), y, b) = 1 \mid b = x \oplus y)] \\
& + \Pr_{x,y \in \Sigma^n, b \in \{0,1\}}[b = \overline{x \oplus y}] \times \Pr_{x,y \in \Sigma^n, b \in \{0,1\}}[D(f_o(x), y, b) = 1 \mid b = \overline{x \oplus y}] \\
= & \frac{1}{2} \Pr_{x,y \in \Sigma^n}[D(f_o(x), y, x \oplus y) = 1] + \frac{1}{2} \Pr_{x,y \in \Sigma^n}[D(f_o(x), y, \overline{x \oplus y}) = 1] \\
= & \frac{1}{2} p_n.
\end{aligned}$$

The last equality is as follows: For every x and y, the tuple $\langle f_o(x), y, \overline{x \oplus y}\rangle$ does not belong to $f_{gl}(\Sigma^*)$ and hence does not belong to $f_{gl}(S)$. Thus, by Observation 3.22, $F_{g,t}(\langle f_o(x), y, \overline{x \oplus y}\rangle) \neq F_{f,t}(w_0)$. So D does not accept $\langle f_o(x), y, \overline{x \oplus y}\rangle$. Then

$$\begin{aligned}\left| \Pr_{x,y\in\Sigma^n}[D(f_o(x), y, x \oplus y) = 1] - \Pr_{x,y\in\Sigma^n, b\in\{0,1\}}[D(f_o(x), y, b) = 1]\right| &= \frac{1}{2}p_n \\ &\geq \frac{1}{2}\frac{1}{2^{(2n)^\gamma}}.\end{aligned}$$

Finally, note that D is a polynomial size circuit with access to NP $\cap$ co-NP oracle O. This contradicts the hard-core bit Lemma 3.21. □

We will now show that for any set A in NP there is a randomized, weakly-length-increasing $\leq^{\text{SNP}}_{ktt}$-reduction from A to L. Later we will derandomize this reduction.

Claim 3.24. *Let A be any language in* NP*. There is an $\leq^{\text{SNP}}_{ktt}$-reduction $\langle h, t\rangle$ such that*

1. *For every x and r, if $h(x, r) = \langle q_1, \cdots, q_k, t\rangle$, then $x \in A$ if and only if $t(L(q_1), \cdots, L(q_k)) = 1$.*
2. *For every $x \in A$ of length n,*

$$\Pr_{r\in\Sigma^m}[\textit{there is a } q_i \in Q_{x,r} \textit{ such that } |q_i| > n] \geq 3/4,$$

where $Q_{x,r}$ is the set of all queries produced by $h(x, r)$ and $m > (nk)^{\frac{1}{1-\gamma}}$.

Proof. By Lemma 3.20, there is an $\leq^{\text{SNP}}_{ktt}$-reduction $\langle h, t\rangle$ from $A \times \Sigma^*$ to L that is γ-sparse on $S = A^n \times \Sigma^m$ (for every n and m). Clearly this reduction satisfies property 1. Let R be the set of all tuples $\langle q_1, \cdots, q_k, t\rangle$ where each q_i is of length at most n and t is a truth-table over k variables. The total number of such tuples is at most $2^{(n+1)k} \times 2^{2^k}$. For a string x of length n, let us count the number of strings from $\{x\} \times \Sigma^m$ that are mapped to the elements of R. Note that every tuple from $\{x\} \times \Sigma^*$ can be encoded as a string of length $2(n + m)$. Since h is γ-sparse on S the total number of such strings is at most

$$\frac{2^{2n+2m}}{2^{(2n+2m)^\gamma}} \times 2^{(n+1)k} \times 2^{2^k} < 2^{(2m+2m)^{1-\gamma}+2nk} < 2^{6m^{1-\gamma}},$$

which is bounded by $2^m/4$ for sufficiently large n, $m > (nk)^{\frac{1}{1-\gamma}}$, and $\gamma > 0$. The first inequality in the above equation is due to the fact that $n < m$ and $k + 2^k < nk$ for large n. Thus for every x, the cardinality of the set $\{r \in \Sigma^m \mid h(x, r) \notin R\}$ is at least $\frac{3}{4}2^m$. Hence the claim follows. □

It is now easy to see that we can derandomize the above reduction. Note that our hypothesis implies the existence of a hard language in EXP whose NP $\cap$ co-NP-oracle circuit complexity is 2^{n^δ} (for some $\gamma > \delta > 0$). Then Theorem 2.15 implies that we can construct a

pseudorandom generator $G : \Sigma^{\log^a m} \to \Sigma^m$ that is secure against NP $\cap$ co-NP-oracle circuits of size $O(m)$. Thus for every $x \in A$ of length n, we have the following:

$$\Pr_{r \in \Sigma^{\log^a m}}[\text{at least one query of } h(x, G(r)) \text{ is of length bigger than } n] \geq 1/2. \tag{1}$$

The desired reduction from A to L works as follows: We describe the query generator. Given an input x of length n, set $m > (nk)^{\frac{1}{1-\gamma}}$, cycle through all strings r of length $\log^a m$ and compute $h(x, G(r)) = \langle q_1, \cdots, q_\ell \rangle$. If at least one q_i is of length bigger than n, then output this tuple and stop. If for every r, every query of $h(x, G(r))$ is of length at most n, then by inequality (1), it must be the case that x is not in A. In this case the reduction simply outputs $\langle 0^{n+1}, \cdots, 0^{n+1}, F \rangle$, where F is the truth-table that always evaluates to false. Note that the running time of the query generator is deterministic quasi-polynomial. So this is a strong nondeterministic, quasi-polynomial-time, k-tt reduction. This completes the proof of Theorem 3.17. □

4 Application to Probabilistic Encryption

Although the ESY conjecture was originally formulated to capture the difficulty of cracking deterministic public-key cryptosystems, we observe that if it holds, then certain "probabilistic" encryption schemes including the Goldwasser-Micali [GM84], Gentry [Gen09] and Ajtai-Dwork [AD97] systems also cannot be NP-hard to crack.

A probabilistic public-key cryptosystem consists of three publicly known, polynomial-time computable functions (E, D, G): encryption function E, decryption function D and key generator G. For a randomly generated string X, $G(X)$ generates the pair (k_1, k_2), where k_1 is the public key and k_2 is the private key. To encrypt a plain text m, randomly pick a string r and generate cipher text $E(m, r, k_1) = c$. The decryption function D has the property that $D(c, k_2) = m$, if c is a valid cipher text for m. We say that the cryptosystem is *error-free* if whenever m and m' are two distinct messages, then for every r and public key k_1, $E(m, r, k_1) \neq E(m', r, k_1)$.

We will now observe that the cracking problem of every error-free probabilistic public-key cryptosystem can be formulated as a disjoint NP-pair.

Theorem 4.1. *If the ESY conjecture holds, then any error-free probabilistic public-key cryptosystem is not* NP*-hard to crack.*

Proof. Given an error-free probabilistic public-key cryptosystem (E, D, G), let $\Pi_n = \{\langle c, k_1, m' \rangle \mid \exists m, r, X$ and k_2 such that $E(m, r, k_1) = c$ and $G(X) = \langle k_1, k_2 \rangle$ and $m < m'\}$, and let $\Pi_y = \{\langle c, k_1, m' \rangle \mid \exists m, r, X$ and k_2 such that $E(m, r, k_1) = c$ and $G(X) = \langle k_1, k_2 \rangle$ and $m \geq m'\}$. Since the cryptosystem is error-free we have that $\Pi_y \cap \Pi_n = \emptyset$ and both Π_y and Π_n are in NP. Thus (Π_y, Π_n) is a disjoint NP-pair. Clearly a separator for this pair can be used to crack the cryptosystem. Thus if the ESY conjecture holds, then this problem has no separator that is NP-hard. □

Since Goldwasser and Micali proved that their cryptosystem is error-free [GM84], and Gentry's homomorphism cryptosystem is also error-free [Gen09], we have the following corollary.

Corollary 4.2. *If the ESY conjecture holds, then the Goldwasser-Micali cryptosystem, as well as Gentry's homomorphic cryptosystem, cannot be* NP*-hard to crack.*

Now we consider the Ajtai-Dwork [AD97] cryptosystem. This cryptosystem has the property that 0 is encrypted as a lattice point near one of the hidden hyperplanes that constitute the private key. The bit 1 is encrypted as a random lattice point so that, with low probability, 1 might be encrypted as a cyphertext that is also a valid encryption of 0, and thus the system is not error-free. We now formulate the cracking problem in terms of detecting encryptions of 0. We let $\Pi_y = \{\langle k_1, c\rangle \mid \exists X$ such that $G(X) = \langle k_1, k_2\rangle$ for some k_2 and $D(c, k_2) = 0\}$ and $\Pi_n = \{\langle k_1, c\rangle \mid \exists X$ such that $G(X) = \langle k_1, k_2\rangle$ for some k_2 and $D(c, k_2) = 1\}$. Clearly, both Π_y and Π_n are in NP. The pair is disjoint since no message decrypts to both 0 and 1. Thus we have the following result.

Theorem 4.3. *If the ESY conjecture holds, then in the Ajtai-Dwork cryptosystem, it is not* NP*-hard to determine whether a given cypher text is a valid encryption of* 0*.*

We note that Nguyen and Stern [NS98] showed that if the polynomial-time hierarchy is infinite, then the Ajtai-Dwork cryptosystem is not NP-hard to crack.

5 Discussion

In this paper we provide evidence that the ESY conjecture holds when we restrict the reduction types. We showed that the ESY-conjecture for length-increasing, $\leq^{\mathrm{P}}_{btt}$-reductions is equivalent to NP $\neq$ co-NP. Theorems 3.7 and 3.18 remove the length increasing restriction by using stronger hypotheses. An obvious question is whether we can weaken the hypotheses. The hypothesis of 3.18 requires the existence of one-one, one-way functions that are hard for circuits with NP $\cap$ co-NP oracles. The one-way functions studied in most of the literature (henceforth called standard one-way functions) only require hardness against subexponential-size circuits (having no oracles). Can we show that if standard one-way functions exist, then the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture holds? We note that this is unlikely. Recall that the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture implies NP $\neq$ co-NP (equivalently, NP differs from NP $\cap$ co-NP). Thus a positive answer to our question immediately shows that if standard-one way functions exist, then NP differs from NP $\cap$ co-NP. Intuitively, the existence of standard one-way functions imply that NP is hard against subexponential time/circuits, not hard against NP $\cap$ co-NP. We cannot hope to prove that the existence of standard one-way functions separates NP from co-NP.

Note that the existence of one-way functions that are hard against NP $\cap$ co-NP-oracle circuits, implies that NP is *hard on average* for NP $\cap$ co-NP. This raises the following interesting question: If NP is average-case hard for NP $\cap$ co-NP, then does the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture hold? Another question is whether we can replace $\leq^{\mathrm{P}}_{btt}$-reductions with reductions

that make $O(\log n)$ (or n^ϵ) nonadaptive queries. We believe that the techniques used in this paper can be extended to work for $O(\log n)$ queries.

As noted in the preliminaries, the ESY conjecture and the ESY-*tt* conjecture both imply that NP differs from UP, and we believe that the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture does not imply NP $\neq$ UP. Is there an oracle relative to which the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture holds and NP = UP? As noted in the introduction, the ESY-*m* conjecture is equivalent to NP $\neq$ co-NP, and there is an oracle relative to which NP $\neq$ co-NP and NP = UP [GW03]. Relative to this oracle, not only does the ESY-*m* conjecture hold and NP = UP, but also the ESY-$\leq^{\mathrm{P}}_{btt,li}$ conjecture holds by Theorem 3.6. Can we show that the ESY-$\leq^{\mathrm{P}}_{btt}$ conjecture is also equivalent to NP $\neq$ co-NP or is there an oracle against it? We mention that there exists an oracle relative to which the ESY conjecture holds [GSSZ04].

6 Acknowledgements

We thank the anonymous reviewers for several helpful comments. These comments greatly increased the quality of the presentation. We also thank Lane Hemaspaandra for providing valuable suggestions.

References

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 284–293, 1997.

[Agr02] M. Agrawal. Pseudo-random generators and structure of complete degrees. In *17th Annual IEEE Conference on Computational Complexity*, pages 139–147, 2002.

[AM77] L. Adleman and K. Manders. Reducibility, randomness, and intractability. In *Proceedings of the 9th ACM Symposium on Theory of Computing*, pages 151–163, 1977.

[ASB00] K. Ambos-Spies and L. Bentzien. Separating NP-completeness under strong hypotheses. *Journal of Computer and System Sciences*, 61(3):335–361, 2000.

[ASFH87] K. Ambos-Spies, H. Fleischhack, and H. Huwig. Diagonalizations over polynomial time computable sets. *Theoretical Computer Science*, 51:177–204, 1987.

[ASNT96] K. Ambos-Spies, H. Neis, and A. Terwijn. Genericity and measure for exponential time. *Theoretical Computer Science*, 168(1):3–19, 1996.

[ASTZ97] K. Ambos-Spies, A. Terwijn, and X. Zheng. Resource bounded randomness and weakly complete problems. *Theoretical Computer Science*, 172(1):195–207, 1997.

[AW09] M. Agrawal and O. Watanabe. One-Way Functions and the Berman-Hartmanis Conjecture. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, pages 194–202, 2009.

[BM95] J. Balcazar and E. Mayordomo. A note on genericity and bi-immunity. In *Proceedings of the Tenth Annual IEEE Conference on Computational Complexity*, pages 193–196, 1995.

[ESY84] S. Even, A. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, May 1984.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, pages 169–178, 2009.

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[Gol06] O. Goldreich. On promise problems: A survey. In *Theoretical Computer Science - Essays in Memory of Shimon Even*, volume 3895 of *Lecture Notes in Computer Science*, pages 254–290. Springer, 2006.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–355, April 1988.

[GSSZ04] C. Glaßer, A. Selman, S. Sengupta, and L. Zhang. Disjoint NP-pairs. *SIAM Journal on Computing*, 33(6):1369–1416, 2004.

[GSZ07] C. Glaßer, A. Selman, and L. Zhang. Canonical disjoint NP-pairs of propositional proof systems. *Theoretical Computer Science*, 370(1):60–73, 2007.

[GW03] C. Glaßer and G. Wechsung. Relativizing function classes. *Journal of Universal Computer Science*, 9(1):34–50, 2003.

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HPRS12] A. Hughes, A. Pavan, N. Russell, and A. Selman. A thirty year old conjecture about promise problems. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 473–484, 2012.

[KvM02] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31:1501–1526, 2002.

[LM96] J. H. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating completeness notions if NP is not small. *Theoretical Computer Science*, 164:141–163, 1996.

[NS98] P. Nguyen and J. Stern. Cryptanalysis of the Ajtai-Dwork cryptosystem. In *Advances in Cryptology—CRYPTO '98. 18th Annual International Cryptology Conference, Santa Barbara, California.*, volume 1462 of *Lecture Notes in Computer Science*, pages 223–242. Springer, 1998.

[PS02] A. Pavan and A. Selman. Separation of NP-completeness notions. *SIAM Journal on Computing*, 31(3):906–918, 2002.

[PS04] A. Pavan and A. Selman. Bi-immunity separates strong NP-completeness notions. *Information and Computation*, 188:116–126, 2004.

[Pud01] P. Pudlák. On reducibility and symmetry of disjoint NP-pairs. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS), Marianske Lazne, Czech Republic*, volume 2136 of *Lecture Notes in Computer Science*, pages 621–632. Springer, 2001.

[Raz94] A. Razborov. On provably disjoint NP pairs. Technical Report 94-006, Electronic Colloquium on Computational Complexity, 1994.

[Sch60] J. Schoenfield. Degrees of models. *Journal of Symbolic Logic*, 25:233–237, 1960.

[SY82] A. Selman and Y. Yacobi. The complexity of promise problems. In *Proceedings of the 8th International Colloquium on Automata, Languages, and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 502–509, Berlin, 1982. Springer-Verlag.