comput. complex. **20** (2011), 755 – 786
© Springer Basel AG 2012
1016-3328/11/040755-32
published online February 7, 2012
DOI 10.1007/s00037-011-0029-x

computational complexity

MORE ON AVERAGE CASE VS APPROXIMATION COMPLEXITY

Michael Alekhnovich

Abstract. We consider the problem to determine the maximal number of satisfiable equations in a linear system chosen at random. We make several plausible conjectures about the average case hardness of this problem for some natural distributions on the instances, and relate them to several interesting questions in the theory of approximation algorithms and in cryptography. Namely we show that our conjectures imply the following facts:

- Feige's hypothesis about the hardness of refuting a random 3CNF is true, which in turn implies inapproximability within a constant for several combinatorial problems, for which no **NP**-hardness of approximation is known.

- It is hard to approximate the NEAREST CODEWORD within factor $n^{1-\epsilon}$.

- It is hard to estimate the rigidity of a matrix. More exactly, it is hard to distinguish between matrices of low rigidity and random ones.

- There exists a secure public-key (probabilistic) cryptosystem, based on the intractability of decoding of random binary codes.

Keywords. Cryptographic primitives, hardness of approximation.

Subject classification. 68Q10, 68Q17, 68Q30, 68W25, 03F20.

Our conjectures are strong in that they assume cryptographic hardness: no polynomial algorithm can solve the problem on any non-negligible fraction of inputs. Nevertheless, to the best of our knowledge no efficient algorithms are currently known that refute any of our hardness conjectures.

1. Introduction

Since the discovery of the PCP theorem in the beginning of 90s (Arora & Safra 1998; Arora 1998), there has been much progress in proving the hardness of approximating optima for various classes of combinatorial problems. This research lead to many brilliant results, for many problems the optimal hardness of approximation that matches the upper bounds given by approximation algorithms has been achieved. However in some cases there is still a large gap between known upper and lower bounds. In some cases (see Håstad 1988; Lagarias *et al.* 1990; Goldreich & Goldwasser 1998) it is unlikely to show NP-hardness of approximation within factor beyond a certain barrier. Thus, it is a natural goal to investigate the hardness of such optimization problems in some other frameworks different from **NP**-completeness.

Recently, Feige (2002) suggested to use cryptographic conjectures for proving interesting inapproximability results. His method is based on the observation that if an instance of certain NP-complete problems looks "like random" then more approximation preserving reductions can be constructed, that do not work on the arbitrary instance. It was assumed in Feige (2002) that it is hard to refute in polynomial time a random 3CNF with linearly many clauses. On one hand this hypothesis is much stronger than the usual worst case hardness assumptions. In particular, by itself it immediately implies the hardness of approximating MAX-3SAT within the optimal constant $8/7 - \epsilon$. On the other hand, this hypothesis implies inapproximability for problems, for which no NP-hardness of approximation is known, which makes it a promising and interesting direction for the further investigation of tractability of NP-complete problems.

In this paper we continue the research initiated by Feige and show more relations between the average case complexity and the complexity of approximation. In order to investigate this direction further it is convenient to define a uniform framework that would embrace both the average case and the worst case complexity. For this we suggest to use the notion of a promise problem generalized for the probabilistic case. In this new concept the instance of the problem is "promised" to be chosen according to one of random distributions, that belong to some given family. The most important example of such a problem is the classical cryptographic task to distinguish two distributions with non-negligible success probability.

In the base of our considerations lies the problem of maximizing the number of simultaneously satisfied equations in a linear system. Due to its high symmetry over GF_2 field the linear mapping possesses some nice "pseudorandom" and pointwise independence properties. The powerful gaussian elimination procedure can invert a linear mapping, however if one adds small non-linear noise to the system, the resulting function becomes hard to invert or to decode in general (one exception is efficiently decodable linear error correcting codes, however only few codes have known polynomial decoding algorithms). The complexity of a linear mapping (sometimes augmented with small number of "non-linear" errors) was considered by many researchers and successfully used in several theoretical and practical applications. To name just a few, this includes

- Hastad's PCP (Håstad 2001). His construction in particular shows that it is NP-hard to approximate MAX-3LIN within the optimal factor $1/2 - \epsilon$. This implies that nothing better than a random guessing is possible to maximize the number of satisfiable linear constraints over three variables.

- Classical construction of (almost) pointwise independent families using linear codes, see for example Naor & Naor (1993).

- Tseitin tautologies for propositional calculus. Since the seminal paper (Tseĭtin 1968) there has been proved a lot of lower bounds on refuting an unsatisfiable linear system for many propositional proof systems.

- Property testing (Ben-Sasson 2003). This recent work provides examples of 3CNF properties based on linear functions that are hard to test.

- Practical cryptographic applications: linear feedback shift registers. In these practical constructions an output of finite automata computing a linear operator augmented by some nonlinear transform is used as a very fast generator of pseudorandom bits, see for example Gollman & Chambers (1989).

In this paper we consider the following problem: distinguish vectors located within hamming distance k from the linear space $\mathrm{im}(A)$ from those located within distance $k+1$ from $\mathrm{im}(A)$, where $A : \{0,1\}^n \to \{0,1\}^m$ is a linear operator over GF_2. We make several conjectures about the hardness of this problem on average for various types of A. As a corollary we derive the following hardness results.

First, if the problem to compute the distance to $\mathrm{im}(A)$ is hard on average for a random sparse matrix A (which does not contradict to the current state of the art in efficient algorithms) then it is hard to refute a random 3CNF on average, i.e. Feige's assumption is true. We believe that this result brings more evidence for Feige's hypothesis.

Under a similar (strong) conjecture we prove that it is hard to approximate the NEAREST CODEWORD problem within factor $n^{1-\epsilon}$. As a consequence of this result we show the hardness of estimating the rigidity of a given matrix. Since this is one of our main motivations we would like to elaborate more on this concept.

As defined by Valiant (1977), the rigidity $\mathcal{R}_M(r)$ of $(0-1)$-matrix M is the minimal number of entries of M that have to be changed to reduce its rank below r. This notion is tightly connected to the linear circuit complexity, namely Valiant showed that for any sequence of matrices M_n s.t. $\mathcal{R}_{M_n}(\epsilon n) > n^{1+\delta}$, the vector multiplication by M_n cannot be performed by linear circuits of linear size and logarithmic depth. Since then a lot of research has been done towards the understanding the notion of rigidity (Razborov 1989; Pudlak & Z. Vavřin 1991; Friedman 1993; Shokrollahi *et al.* 1997; Kashin & Razborov 1998; Lokam 2001), however still there are no known explicit constructions of matrices with high rigidity. We try to explain the intricate difficulty to find such explicit matrices from the point of the natural proofs approach by Razborov & Rudich (1997). We show that it is not likely to prove lower bounds for the rigidity by constructing an efficiently computable property $\mu(M)$ which separates matrices of low rigidity from random ones.

We hope that this result may give some evidence that new ("non-natural") ideas are necessary for proving lower bounds on matrix rigidity.

Finally, modulo stronger hypothesis that it is not feasible to decode $n^{1/2-\epsilon}$ errors for a random linear error correcting code we construct two public key cryptosystems, based on binary codes. The first cryptosystem was inspired by Ajtai-Dwork lattice cryptosystem (Ajtai & Dwork 1997), which gives a brilliant reduction from the worst-case to the average case complexity. Our second cryptosystem is similar to McElice cryptosystem (McEliece 1978), however in our case the choice of the underlying error correcting code is arbitrary and the security is based solely on the assumed hardness of decoding a *random* code. We do not claim (although do not exclude either) any practical significance of the constructed cryptosystems, however we hope that they might be interesting from the theoretical point of view, and probably might have some applications in complexity theory.

The paper is organized in the following way. Section 2 contains some basic notation and the definition of probabilistic promise problems. Section 3 proves our main reduction, which we use for applications in Section 4. We finish our paper with discussion and open questions in Section 5.

2. Preliminaries

2.1. Probability and indistinguishability. We use Greek letters for random variables and capital letters for distributions. For a distribution D we write $\xi \in_{\mathrm{R}} D$ to indicate that random variable ξ is chosen according to D. We denote the uniform distribution on the set $\{0,1\}^n$ by $\mathcal{U}_n$. A function $\mu : \mathbb{N} \to \mathbb{R}$ is negligible if it tends to zero faster than any inverse polynomial, that is, for every constant $c > 0$ $\mu(n) < 1/n^c$, for all sufficiently large n's. A *probability ensemble* (or just ensemble for short) is a sequence of random variables $\{\xi_n\}_{n\in I}$ where I is a countable set and ξ_n is a random variable. Sometimes when clear from the context we omit the lower index n in the asymptotic notation of random sequences (as well as other objects parameterized by the length of the input).

Definition 2.1 (statistical distance). *For two random variables ξ^1 and ξ^2 their statistical distance is defined as*

$$\rho(\xi^1, \xi^2) = \max_C \left|\Pr[C(\xi^1) = 1] - \Pr[C(\xi^2) = 1]\right|,$$

where C is an arbitrary Boolean function. The ensembles $\xi^1 = \{\xi^1_n\}$ and $\xi^2 = \{\xi^2_n\}$ are statistically indistinguishable *if their statistical distance is negligible in n.*

Definition 2.2 (computational distance). *Two probability ensembles ξ^1_n and ξ^2_n are said to be* computationally $f(n)$-close *if for any probabilistic polynomial time C there exists N s.t. for all $n > N$*

$$|\Pr[C(\xi^1_n) = 1] - \Pr[C(\xi^2_n) = 1]| \le f(n).$$

The ensembles ξ^1_n and ξ^2_n are computationally indistinguishable *if for any probabilistic polynomial time C the distinguishing advantage $|\Pr[C(\xi^1_n) = 1] - \Pr[C(\xi^2_n) = 1]|$ is negligible in n.*

It is well known that statistical and computational distance have metric-like properties with respect to the space of random variables. In particular, the following fact holds.

Proposition 2.3. *Assume that ξ^1, ξ^2, ξ^3 are random sequences satisfying $\rho_c(\xi^1, \xi^2) \le f_1(n)$ and $\rho_c(\xi^2, \xi^3) \le f_2(n)$. Then*

$$\rho_c(\xi^1, \xi^3) \le f_1(n) + f_2(n).$$

2.2. Linear codes. We will mainly work in the field GF_2, and in some cases, by abuse of notation, identify binary strings as vectors over GF_2. For $0-1$ vector x, its *Hamming weight* is the number of ones in x. A linear binary error correcting code $\mathcal{C}$ is a linear subspace of GF_2^m. As any linear space, $\mathcal{C}$ can be specified as an image of a linear operator $G \in \mathrm{GF}_2^{m\times n}$, i.e., $\mathcal{C} = \mathrm{im}(G) = \{Gx | x \in \mathrm{GF}_2^n\}$. In this case, G is called the *generator matrix* of the code $\mathcal{C}$. Similarly, $\mathcal{C}$ can be specified as a kernel of a linear operator $H \in \mathrm{GF}_2^{m-n\times m}$: $\mathcal{C} = \ker(H) = \{c | Hc = 0^{m-n}\}$. In this case, H is called the *parity check matrix* for $\mathcal{C}$. A code $\mathcal{C}$ is *good* if its distance (i.e., the minimal Hamming weight of a non-zero codeword) is at least δm for some constant $\delta > 0$.

2.3. Probabilistic promise problems. Promise problems are a useful formalism for proving gaps in approximation of **NP**-hard problems. In a promise problem the instance is "promised" to be taken out of specific subclass of all instances. It is convenient to use the following generalized definition of this notion for the statement of our results.

Definition 2.4 (probabilistic promise problem). *Let Ω be a probability space, Σ a finite alphabet, and Σ^n be the set of all words of length n over Σ. A* probabilistic promise problem *is a sequence* $\Pi = (\Pi_n^{\text{yes}}, \Pi_n^{\text{no}})$,

$$\Pi_n^{\text{yes}}, \Pi_n^{\text{no}} \subseteq (\Sigma^n)^{\Omega},$$

such that for every n, Π_n^{yes} and Π_n^{no} contain only measurable functions $\xi_n : \Omega \to \Sigma^n$.

For a (possibly probabilistic) algorithm C and the probabilistic promise problem $(\Pi_n^{\text{yes}}, \Pi_n^{\text{no}})$ define its completeness *as*

$$c_n = \min_{\xi \in \Pi_n^{\text{yes}}} \Pr_{\Omega}[C(\xi) = 1]$$

and its soundness *as*

$$s_n = \max_{\xi \in \Pi_n^{\text{no}}} \Pr_{\Omega}[C(\xi) = 1)],$$

where in both cases the probability is also taken over the internal randomness of C. We define the success probability *of C to be*

$$\text{success}(C, \Pi) \overset{\text{def}}{=} c_n - s_n.$$

One can think of this definition in the following way. Assume that the instance of the promise problem is generated according to some probabilistic distribution, which belongs to some general family (e.g. normal distributions). Then the success probability of an algorithm is defined as the worst among all distributions in the family. The next example shows that this definition is indeed a generalization of usual promise problems.

Example 2.5. Given a pair of non-intersecting languages L^{yes} and L^{no} that describe the standard promise problem choose $\Omega = \{0\}$ and let Π_n^{yes} contain the functions $\xi : 0 \mapsto x_n$ for all $x_n \in \Sigma^n \cap L^{\text{yes}}$

and Π_n^{no} contain the functions $\xi : 0 \mapsto y_n$ for all $y_n \in \Sigma^n \cap L^{\mathrm{no}}$. There is no randomness, the admissible distributions coincide with *yes* and *no* instances and, when A is deterministic, c_n and s_n are always either 0 or 1. ◊

Another example is the problem considered in Feige (2002).

Example 2.6 (refuting a random 3CNF). Let Π_n^{no} contain only one distribution that chooses a random 3CNF with n variables and Δn clauses, where Δ is a large constant. Π_n^{yes} consists of all distributions for which the generated 3CNF is always satisfiable.

Clearly every deterministic algorithm that solves this probabilistic promise problem with high success probability should always say "yes" on every satisfiable CNF (as Π_n^{yes} in particular contains all singleton distributions) and say "no" with high probability on a random CNF. ◊

In the special case where both Π^{yes} and Π^{no} contain only one distribution, it is easy to show that solving the probabilistic promise problem is equivalent to the standard cryptographic task to distinguish *yes* and *no* distributions:

Proposition 2.7. *Let* $\Pi = (\Pi^{\mathrm{yes}}, \Pi^{\mathrm{no}})$, *be a probabilistic promise problem where* $\Pi_n^{\mathrm{yes}} = \{\xi_n^{\mathrm{yes}}\}, \Pi_n^{\mathrm{no}} = \{\xi_n^{\mathrm{no}}\}$. *Suppose that there exists a probabilistic polynomial-time algorithm* C *for which*

$$|\Pr[C(\xi^{\mathrm{yes}}) = 1] - \Pr[C(\xi^{\mathrm{no}}) = 1]| > f(n),$$

for infinitely many n*'s and for some function* $f(n)$. *Then there exists a probabilistic polynomial-time algorithm* B *that solves the problem* Π *with success probability* $f(n)$ *for infinitely many input lengths* n.

Proof. Fix B to be either C or its negation. □

3. Main reduction

In the core of our reductions lies the following **NP**-optimization problem called MAXIMUM SATISFYING LINEAR SUBSYSTEM (MAX-LIN-SAT for short).

Problem 3.1 (MAXIMUM SATISFYING LINEAR SUBSYSTEM).

- *INSTANCE: System $Ax = b$ of linear equations, where A is $m \times n$ matrix over* GF_2*, and b is a vector in* $\{0, 1\}^m$.
- *SOLUTION: A vector* $x \in \{0, 1\}^n$.
- *OBJECTIVE FUNCTION: The number of equations satisfied by x.*

Below we define an average case version of this problem, in which the random system is generated by choosing a planted solution and adding a number of errors.

Definition 3.2. *Let A be $m \times n$ matrix over* GF_2*. Let* $\mathrm{D}_k(A)$ *be the distribution of the random vector*

$$\eta_k(A) = Av + e,$$

where $v \in_{\scriptscriptstyle R} \mathcal{U}_n$ is random and $e \in \mathrm{GF}_2^m$ is randomly chosen from the vectors of Hamming weight k.

Thus, $\mathrm{D}_k(A)$ is the distribution of a random vector located within distance k from $\mathrm{im}(A)$. We are interested in the complexity of maximizing the number of satisfied equations in the system

$$Ax = \eta.$$

More exactly, we want to distinguish between distributions $D_k(A)$ and $D_{k+1}(A)$. Below we show that if this problem is hard then the distribution $D_k(A)$ is computationally indistinguishable from the uniform distribution, i.e. $\eta_k(A)$ is a good pseudorandom generator.

Theorem 3.3. *Let $k(n), m(n)$ be integer parameters, $\epsilon(n)$ be a positive real and $\{A_n\}$ be a sequence of $m(n) \times n$ matrices over* GF_2*. Assume that the ensembles*

$$(A_n, \mathrm{D}_{k(n)}(A_n)) \quad \textit{and} \quad (A_n, \mathrm{D}_{k(n)-1}(A_n))$$

are computationally $\epsilon(n)$-close. Further, assume that the ensembles

$$(A_n, \mathrm{D}_{k(n)}(A_n)) \quad \textit{and} \quad (A_n, \mathrm{D}_{k(n)+1}(A_n))$$

are also computationally $\epsilon(n)$-close. Then the ensembles

$$(A_n, \mathrm{D}_{k(n)}(A_n)) \quad \textit{and} \quad (A_n, \mathcal{U}_{m(n)})$$

are computationally $t(n)\epsilon(n)/2 + m(n)e^{\Omega(-t(n)/m(n))}$-close for any polynomial $t(\cdot)$.

Let ξ_n be a random binary vector of length m which is distributed as follows: with probability half ξ_n equals to the all zero vector 0^m; and with probability half, ξ_n is uniformly distributed over all vectors of weight 1. The proof of Theorem 3.3 is based on the following Lemmas.

Lemma 3.4. *Under the assumption of Theorem 3.3, we have*

$$\rho_c((A_n, \eta_{k(n)}(A_n)), (A_n, \eta_{k(n)}(A_n) + \xi_n)) < \epsilon(n)/2.$$

Proof. Let C be a probabilistic polynomial algorithm. Fix n and let $k = k(n), m = m(n), \epsilon = \epsilon(n), \xi = \xi_n$ and $A = A_n$. We will show that

$$|\Pr[C(A, \eta_k(A) + \xi) = 1] - \Pr[C(A, \eta_k(A)) = 1]| < \epsilon/2.$$

The variable $\eta_k(A) + \xi$ results from $\eta_k(A)$ by flipping a random bit with probability 1/2. Recall that $\eta_k(A) = Av + e_k$, thus

$$\eta_k(A) + \xi = Av + (e_k + \xi).$$

With probability 1/2 the variable $e_k + \xi$ is a uniform vector of weight k, with probability $1/2 \cdot k/m$ it is a uniform vector of weight $k-1$ and with probability $1/2 \cdot (m-k)/m$ it is of weight $k+1$. Let p_r denote $\Pr[C(A, \eta_r(A)) = 1]$. Thus, we can write

$$\Pr[C(A, \eta_k(A) + \xi) = 1] = \frac{1}{2}p_k + \frac{k}{2m}p_{k-1} + \frac{m-k}{2m}p_{k+1}$$

which implies that

$$\begin{aligned} &\Big|\Pr[C(A, \eta_k(A) + \xi) = 1] - p_k\Big| \\ &\quad = \left|\frac{k}{2m}(p_{k-1} - p_k) + \frac{m-k}{2m}(p_{k+1} - p_k)\right| < \epsilon/2. \end{aligned}$$

The lemma follows. □

For an integer i, let ξ_n^i be the sum (over GF_2) of i independent samples from ξ_n. (If $i = 0$ then ξ_n^i is fixed to be the all zero string.)

LEMMA 3.5. *Under the assumption of Theorem 3.3, we have*

$$\rho_c((A_n, \eta_{k(n)}(A_n)), (A_n, \eta_{k(n)}(A_n) + \xi_n^{t(n)})) < \epsilon(n)t(n)/2,$$

for every polynomial $t(\cdot)$.

PROOF. The proof uses a standard hybrid argument. Assume, towards a contradiction, that there exists an efficient algorithm C for which $\Pr[C(A_n, \eta_{k(n)}(A_n)) = 1] - \Pr[C(A_n, \eta_{k(n)}(A_n) + \xi_n^{t(n)}) = 1]$ is greater than $\epsilon(n)t(n)/2$ for infinitely many n's. We construct an algorithm B which distinguishes $(A_n, \eta_{k(n)}(A_n))$ from $(A_n, \eta_{k(n)}(A_n) + \xi_n)$ and derive a contradiction to Lemma 3.4. Given A_n and an $m(n)$-bit input z, the algorithm B chooses a random $i \in \{0, \ldots, t(n) - 1\}$, samples ξ_n^i adds it to z and outputs $C(A_n, z + x_i)$.

We analyze the success probability of B. Fix n and let $k = k(n), m = m(n), \epsilon = \epsilon(n), t = t(n), \xi = \xi_n$ and $A = A_n$. We define the following hybrid distributions: Let $H_i = (A, \eta_k(A) + \xi^i)$ and let $p_i = \Pr[C(H_i) = 1]$. By our assumption $p_t - p_0 > \epsilon t/2$. It follows that when i is chosen randomly from $\{0, \ldots, t(n) - 1\}$ the expectation

$$\mathbb{E}_i[p_{i+1} - p_i] = (p_t - p_0)/t$$

is greater than $\epsilon/2$. On the other hand, the probability of $B(A, \eta_k(A)) = 1$ is equal to $\mathbb{E}_i[p_i]$ and the probability of $B(A, \eta_k(A) + \xi^i) = 1$ is exactly $\mathbb{E}_i[p_{i+1}]$. It follows, by the linearity of expectation, that B has distinguishing advantage greater than $\epsilon/2$, and the lemma follows. □

PROOF. Theorem 3.3 Fix a polynomial $t(n)$. By Lemma 3.5, the distribution $(A_n, \mathrm{D}_{k(n)}(A_n))$ is computationally $t(n)\epsilon/2$-close to $(A_n, \eta_{k(n)}(A_n) + \xi_n^{t(n)})$. We argue that the latter distribution is $m(n)e^{\Omega(-t(n)/m(n))}$ statistically close to uniform. Indeed, it is well known (see for example Pak & Vu 2001) that the distribution of ξ_n^t converges to the uniform very fast, namely the statistical distance $\rho(\xi_n^t, \mathcal{U}_m)$ is smaller than $me^{-\Omega(t/m)}$. Hence, the theorem follows from Proposition 2.3. □

4. Applications

In this section we apply the general result of Theorem 3.3 to show the intractability of several optimization problems. Our results in this section are based on three different conjectures about the average hardness of MAXIMUM SATISFYING LINEAR SUBSYSTEM. We are unaware about efficient algorithms that refute any of these assumptions. Formally, all conjectures are independent and have different implications in the theory of approximation algorithms and in cryptography. Conjecture 4.5 may be of independent interest as it assumes a mixture of average case and worst-case hardness.

Problem 4.1 (Average-3LIN(m, ϵ)).

- *INPUT: Parameters n, m and $\epsilon = \epsilon(n) > 0$.*
- *YES INSTANCE: A random pair (A_n, b_1), where A_n is a random $m \times n$ $(0-1)$-matrix in which every row contains exactly three ones and $b_1 \in_{R} D_{\lceil \epsilon n \rceil}(A_n)$.*
- *NO INSTANCE: A random pair (A_n, b_2), where A_n is a random $m \times n$ matrix in which every row contains exactly three ones and $b_2 \in_{R} D_{\lceil \epsilon n \rceil + 1}(A_n)$.*

We conjecture that Average-3LIN(m, ϵ) cannot be solved with success probability greater than $1/(n \ln^2 n)$ as long as m is linear in n and ϵ is bounded away from zero. Formally,

Conjecture 4.2. *For every $m(n) = O(n)$, for every function $0 < \epsilon(n) < 1/2$ which is bounded away from zero, and for every efficient algorithm C*

$$\text{success}(C, \text{Average-3LIN}(m, \epsilon)) < 1/(n \ln^2 n),$$

for all sufficiently large n's.

Remark 4.3. *It can be shown that if the matrix A_n happens to be "degenerate" (for example, contains two equal rows, which occurs with probability $1/n$) then one can distinguish vectors b_1 and b_2*

with probability roughly $1/n$. This gives an algorithm that distinguishes (A, b_1) and (A, b_2) with success probability $1/n^2$. We believe that no algorithm can do substantially better than this bound, and if A_n is an expander (which occurs with probability $1 - 1/n$) then the distributions of b_1 and b_2 are indistinguishable. Thus, we could specify in Problem 4.1 that A is chosen uniformly from the set of good expanders and assume its $1/n^{\omega(1)}$-intractability (but this would sacrifice the property of being samplable).

Problem 4.4 (Average-LIN(A, ϵ) with the worst case choice of the matrix A).

- *INPUT: Parameters n, m, k, and a sequence of $m \times n$ binary matrices $A = \{A_n\}$.*
- *YES INSTANCE: A random vector $b_1 \in_{R} D_{\lceil k \rceil}(A_n)$.*
- *NO INSTANCE: A random vector $b_2 \in_{R} D_{\lceil k \rceil + 1}(A_n)$.*

We conjecture that the problem is hard for every $m = O(n)$ and $k = n^\epsilon$ for arbitrary $\epsilon > 0$. This problem resembles the question how to decode a linear error correcting code from n^ϵ errors. However, the matrix A_n is *not necessarily* a generator of a good code. Thus, a priori Problem 4.4 may be more difficult than the unique decoding of the random codeword with n^ϵ errors.

Conjecture 4.5. *For every $m(n) = \Theta(n)$, there exists a family of $m(n) \times n$ matrices $A = \{A_n\}$, such that for every function $k(n)$ which satisfies $n^\epsilon < k(n) < n^{1-\epsilon}$ for some constant $\epsilon > 0$, and for every efficient algorithm C, the success probability*

$$\text{success}(C, \textsc{Average-LIN}(A, k(n)))$$

is negligible in n.

Note that this conjecture combines the worst case assumption (in the choice of A_n) with the average case assumption in the random choice of n^ϵ unsatisfied equations. Finally for our cryptographic applications in Section 4.4 we need the following conjecture.

Problem 4.6 (Average-NEAREST-CODEWORD(m, k)).

- *INPUT: Parameters n, m and $k = k(n) > 0$.*
- *YES INSTANCE: A random pair (A_n, b_1), where A_n is a random $m \times n$ binary matrix and $b_1 \in_{R} D_{\lceil k(n) \rceil}(A_n)$.*
- *NO INSTANCE: A random pair (A_n, b_2), where A_n is a random $m \times n$ binary matrix and $b_2 \in_{R} D_{\lceil k(n) \rceil + 1}(A_n)$.*

Conjecture 4.7. *For any $m = O(n)$ and $k = \Theta(\sqrt{n})$, and for every efficient algorithm C, the success probability*

$$\text{success}(C, \text{Average-NEAREST-CODEWORD}(m, k))$$

is negligible in n.

4.1. Average MAX-3LIN and Feige's Refute-3SAT hypothesis. In his paper on average case hardness versus hardness of approximation (Feige 2002), Feige assumes that the following problem (that we call Refute-3SAT) is hard on average and infers the hardness of approximation for several interesting problems, for which it is not known whether the approximation is NP-hard.

Problem 4.8 (Refute-3SAT(m, ϵ)).

- *INPUT: Parameters n, m and $\epsilon > 0$.*
- *YES INSTANCE: 3CNF φ with n variables and m clauses, for which at least $(1 - \epsilon)m$ clauses are satisfiable.*
- *NO INSTANCE: A random 3CNF φ with n variables and m clauses.*

Note, that since nothing is said about the distribution of *yes* instances it is assumed to be chosen in the worst case. Thus in order to solve this probabilistic promise problem the algorithm should always output "yes" on every *yes* instance and w.h.p. say "no" on every *no* instance. Stated in our terms, Hypothesis 2 in Feige (2002) assumes that for every constant $\epsilon > 0$ and sufficiently large constant Δ, no polynomial algorithm can solve Refute-3SAT$(\Delta n, \epsilon)$ with constant success probability. One can similarly define Refute-3LIN problem:

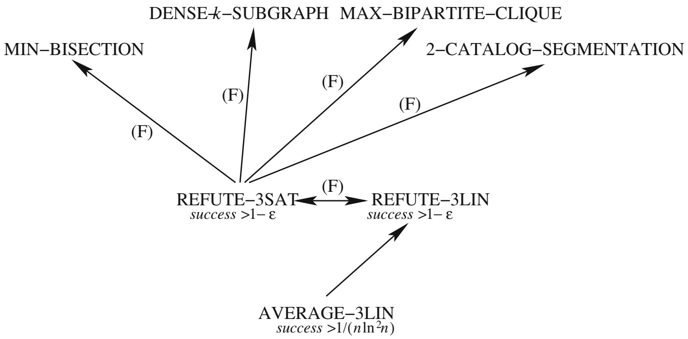

Figure 4.1: The graph of average-case reductions

PROBLEM 4.9 (REFUTE-3LIN(m, ϵ)).

- *INPUT: Parameters n, m and $\epsilon > 0$.*
- *YES INSTANCE: A set of m linear constraints on three variables, such that at least $(1 - \epsilon)m$ constraints are satisfiable.*
- *NO INSTANCE: A set of m linear constraints on three variables, each of which is chosen at random from the set of all constraints over n variables.*

It was shown in Feige (2002) that Refute-3SAT and Refute-3LIN are essentially equivalent, and if they are hard then the following problems cannot be approximated within some constant: Min Bisection, Dense k-subgraph, Max Bipartite Clique, 2-Catalog Segmentation. Below we show that if AVERAGE-3LIN (with success probability $1/(n \ln^2 n)$) is hard for polynomial algorithms then REFUTE-3LIN is hard (and hence the other hardness results hold too, see Figure 4.1). This result may be interesting as

- it gives more evidence that Hypothesis 2 in Feige (2002) is true.
- AVERAGE-3LIN is a samplable problem, thus more natural from the cryptographic point of view.

THEOREM 4.10. *Suppose that Conjecture 4.2 holds. Then, for every $m(n) = O(n)$, for every fixed $0 < \epsilon < 1/2$, and for every efficient algorithm C,*

$$\text{success}(C, \text{REFUTE-3LIN}(m, \epsilon)) < o(1).$$

PROOF. Assume, towards a contradiction, that there exists an $m(n) = O(n)$, a fixed $0 < \epsilon < 1/2$, and a probabilistic polynomial algorithm C, that solves REFUTE-3LIN(m, ϵ) with success probability greater than some fixed $\delta > 0$, for infinitely many n's. Let Y_n be the distribution of the pair (A, y) where $A \in \mathrm{GF}_2^{m(n)\times n}$ is a random binary matrix in which each row contains exactly three ones and y is chosen from $D_{\lceil \epsilon n \rceil}(A)$. Let X_n be the distribution of the pair (A, y) where A is distributed as before, and $y \in_{\mathrm{R}} \mathcal{U}_{m(n)}$ is uniformly chosen. Then,

$$\Pr[C(Y_n) = 1] - \Pr[C(X_n) = 1] > \delta, \tag{4.11}$$

for infinitely many n's. We will show that this contradicts Conjecture 4.2.

Let Y_n^+ (respectively, Y_n^-) be the same as Y_n except that y is chosen from $D_{\lceil \epsilon n \rceil + 1}(A)$ (respectively, $D_{\lceil \epsilon n \rceil - 1}(A)$). Under Conjecture 4.2, we know that AVERAGE-3LIN$(m(n), \epsilon(n))$ cannot be solved with success probability $1/(n \ln^2 n)$, and therefore, by Proposition 2.7, Y_n is $1/(n \ln^2 n)$-computationally close to Y_n^+. Similarly, since AVERAGE-3LIN$(m(n), \epsilon(n) - 1)$ is also $1/(n \ln^2 n)$-intractable, we get that Y_n^- is $1/(n \ln^2 n)$-computationally close to Y_n. Now we can apply Theorem 3.3 with $t = \delta n \ln^2 n$, and conclude that Y_n is δ-computationally close to X_n. This however contradicts (4.11). □

4.2. Nearest codeword. In this section we study the limitations on approximability of the following problem.

PROBLEM 4.12 (Nearest codeword).

- *INSTANCE: A linear binary code given by its generator $m \times n$ matrix A and a vector b.*
- *SOLUTION: A vector $x \in \{0, 1\}^n$ that specifies a codeword Ax.*
- *OBJECTIVE FUNCTION: The Hamming distance $d(Ax, b)$.*

The best known NP-hard lower bound on the factor of approximation for the Nearest Codeword is $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$ due to Arora *et al.* (1997).

Theorem 4.13. *Conjecture 4.5 implies that the Nearest Codeword is hard to approximate within a factor of $n^{1-\epsilon}$ for every fixed $\epsilon > 0$.*

We will need the following simple lemmas.

Lemma 4.14. *Conjecture 4.5 implies that for every $m(n) = \Theta(n)$, there exists a family of $m(n) \times n$ matrices $A = \{A_n\}$, such that for constant $0 < \epsilon < 1$ the distribution $(A_n, \mathrm{D}_{n^\epsilon}(A_n))$ is pseudo-random—i.e., $(A_n, \mathrm{D}_{n^\epsilon}(A_n))$ is computationally indistinguishable from $(A_n, \mathcal{U}_m)$.*

Proof. Let $m = \Theta(n)$ and let $k = \lceil n^\epsilon \rceil$ and $k' = \lceil n^\epsilon \rceil - 1$. Let A_n be a sequence of $m \times n$ matrices for which both, Average-LIN(A_n, k) and Average-LIN(A_n, k'), cannot be solved with more than negligible probability. (The existence of such A_n follows from Conjecture 4.5.) By Proposition 2.7, it follows that $(A_n, \mathrm{D}_{k(n)}(A_n))$ is computationally indistinguishable from both, $(A_n, \mathrm{D}_{k(n)+1}(A_n))$ and $(A_n, \mathrm{D}_{k(n)-1}(A_n))$. Hence, by Theorem 3.3, it follows that $(A_n, \mathrm{D}_{k(n)}(A_n))$ cannot be distinguished from the uniform distribution $(A_n, \mathcal{U}_m)$. □

Lemma 4.15. *Let A be an arbitrary $m \times n$ matrix A where $m \geq 2n$. Then, the Hamming distance of the uniformly distributed vector $y \in \{0,1\}^m$ and the set $\mathrm{im}(A)$ is greater than $m/10$, except with exponentially small probability.*

Proof. By a simple counting argument. The number of different vectors in $\mathrm{im}(A)$ is at most 2^n, the ball of radius $m/10$ contains $\binom{m}{m/10}$ points, the union of these balls covers at most

$$2^n 2^{\mathrm{H}(1/10)m} < 2^{0.97m}$$

points, which consists an exponentially small fraction of the whole space. □

Proof. Theorem 4.13 The proof is similar to that of Theorem 4.10. Assume, towards a contradiction, the existence of an approximation algorithm C which approximates the Nearest Codeword within a factor of $n^{1-\epsilon}$ for some $\epsilon > 0$ and for infinitely many input lengths. Let $m = 2n$ and $k = \lceil n^{\epsilon/2} \rceil$. By Lemma 4.14, there exists a sequence A_n of $m \times n$ matrices for which $(A_n, \mathrm{D}_{k(n)}(A_n))$ is pseudorandom, i.e., cannot be distinguished from the uniform distribution $(A_n, \mathcal{U}_m)$. However, the approximation algorithm C can be easily used to distinguish between these distributions. Indeed, consider an algorithm B which given $A \in \mathrm{GF}_2^{m \times n}$ and $z \in \mathrm{GF}_2^m$ outputs 1 if and only if $C(A, z)$ is smaller that $n^{1-\epsilon/2}$. Then, by Lemma 4.15, C has distinguishing advantage of $1 - 2^{\Omega(n)}$ for infinitely many n's, and the theorem follows. □

4.3. Matrix rigidity. Recall that for a matrix M over GF_2 its rigidity $\mathcal{R}_M(r)$ is the minimal number of entries of M that have to be changed to reduce its rank below r. Consider the following approximation problem.

Problem 4.16 (Approximating matrix rigidity).

- *INPUT: Parameters m, $\epsilon, \delta > 0$.*
- *YES INSTANCE: Any matrix $M \in \mathrm{GF}_2^{m \times m}$ for which*
$$\mathcal{R}_M(\epsilon m) < m^{1+\delta}$$
- *NO INSTANCE: A random matrix M uniformly chosen from $\mathrm{GF}_2^{m \times m}$.*

Theorem 4.17. *Conjecture 4.5 implies that for every fixed $\epsilon, \delta > 0$, Problem 4.16 cannot be solved with more than negligible success probability.*

Proof. Fix some constants $\epsilon > 0$ and $\delta > 0$, we will show that there exists a distribution M_m over $m \times m$ matrices with low rigidity $\mathcal{R}_{M_m}(\epsilon m) < m^{1+\delta}$ that is computationally indistinguishable from a random $m \times m$ matrices $\mathcal{U}_{m \times m}$.

Choose $n = \epsilon m$ and denote $k = m^{\delta}$. By Lemma 4.14, there exists a sequence A_m of $m \times n$ matrices for which $(A_m, \mathrm{D}_{k(m)}(A_m))$ is computationally indistinguishable from the distribution $(A_m, \mathcal{U}_m)$.

For $t = 0 \ldots m$, define a random matrix $M_m^t = (M_1 | M_2 | \cdots | M_m)$ whose first t columns M_i are chosen independently with distribution $D_k(A_m)$ and last $m - t$ columns are chosen uniformly at random. Note that M_m^0 is just the uniform distribution over $m \times m$

matrices. Also, all matrices M in the support of M_m^m have low rigidity $\mathcal{R}_M(\epsilon m) < m^{1+\delta}$. Indeed, every column in M can be represented as a sum of a vector in $\mathrm{im}(A_m)$ and a vector of weight k. Thus we can flip km entries in M_m to decrease its rank to $\dim(\mathrm{im}(A_m)) = \epsilon m$.

We argue that the probability ensemble (A_m, M_m^m) is computationally indistinguishable from the ensemble (A_m, M_m^0). This follows by a standard hybrid argument. Assume, towards a contradiction, that an algorithm C distinguishes between these distribution with non-negligible advantage $\alpha(m)$. Then, we can construct an algorithm B which distinguishes $(A_m, \mathrm{D}_{k(m)}(A_m))$ from $(A_m, \mathcal{U}_m)$ with advantage $\alpha(m)/m$, which is also non-negligible and therefore contradicts Conjecture 4.5.

The algorithm B gets A_m and an m-bit string z as an input, and then, it invokes C on the pair A_m and $T_m = (V_1|V_2|\cdots|V_i|z|R_{i+2}|\cdots|R_m)$ where i is randomly chosen from $\{0, \ldots, m-1\}$, the V_j's are independently chosen from $D_k(A_m)$, and the R_j's are independently chosen from the uniform distribution $\mathcal{U}_m$. Let $p_i = \Pr[C(A_m, M_m^i) = 1]$. Clearly,

$$\Pr[B(A_m, \mathrm{D}_{k(m)}(A_m)) = 1] = \frac{1}{m}\sum_{i=1}^{m} p_i,$$

and similarly,

$$\Pr[B(A_m, \mathcal{U}_m) = 1] = \frac{1}{m}\sum_{i=0}^{m-1} p_i,$$

it follows that the distinguishing advantage of B is

$$\frac{1}{m}\left|\sum_{i=1}^{m} p_i - \sum_{i=0}^{m-1} p_i\right| = (p_m - p_0)/m \geq \alpha(m)/m,$$

which completes the proof. □

4.4. Two public-key cryptosystems. We have seen in Section 3 that any of our conjectures implies the existence of a simple pseudorandom distribution. Below we give two constructions of another cryptographic primitive: public key cryptosystem. Our first cryptosystem is easier to analyze, however it has a large computational and communication overhead per encrypted bit. The second system is more efficient and it encrypts n bits by an $O(n)$-bit ciphertext. Both systems are secure unless Conjecture 4.7 is false.

The rest of this section is organized as follows. We begin by presenting a formal definition of public-key encryption scheme (Section 4.4.1), together with several useful facts (Section 4.4.2), and then describe and analyze our new cryptosystems (Section 4.4.3 Section 4.4.4).

4.4.1. Definition of public-key cryptosystem.

DEFINITION 4.18. ***(Public-key encryption scheme)*** *A* public-key encryption scheme *(PKE) consists of three probabilistic polynomial-time algorithms: A key generation algorithm Gen which takes as an input a security parameter 1^n, and outputs a pair of public/private keys $(\mathsf{pk}, \mathsf{sk})$; An encryption algorithm Enc which given a public-key pk and a plaintext w as inputs, outputs a ciphertext y; and a decryption algorithm Dec which uses the private key sk to decrypt a ciphertext y. The scheme should satisfy correctness: For every plaintext $x \in \{0,1\}^*$, the error probability*

$$\Pr_{(\mathsf{pk},\mathsf{sk}) \in_R \mathsf{Gen}(1^n)}[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, w)) \neq w)],$$

taken over the internal randomness of Enc, Dec and Gen, should be negligible in n.

We consider schemes that are secure against chosen plaintext attacks as formulated by the definition of semantic security (Goldwasser & Micali 1984).

DEFINITION 4.19. ***(Security of public-key encryption)*** *Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Consider the following indistinguishability game between a challenger and an adversary C:*

- *Common input: security parameter 1^n.*
- *The challenger chooses $(\mathsf{pk}, \mathsf{sk}) \in_R \mathsf{Gen}(1^n)$ and sends pk to the adversary C.*

- *C outputs two messages $w_1 \neq w_2$ of the same length.*
- *The challenger chooses a random bit σ and sends the ciphertext $\mathsf{Enc}_{\mathsf{pk}}(w_\sigma)$.*
- *The adversary outputs σ' in an attempt to guess σ.*

We say that the scheme is secure *if every probabilistic polynomial time algorithm C guesses σ with probability at most $\frac{1}{2} + \nu(n)$, where $\nu(n)$ is some negligible function that may depend on C.*

We mention that a scheme that supports encryption of messages of length ℓ can be easily transformed to support messages of arbitrary polynomial lengths by simple concatenation (see Goldreich 2004, Chapter 5).

4.4.2. Useful facts. Before we describe our cryptosystem we put forward the following useful facts. Let $\mathcal{U}_{m\times n}$ be the uniform distribution over $m \times n$ matrices, $\mathcal{U}_{m\times n,\mathrm{rk}=n}$ be the restriction of $\mathcal{U}_{m\times n}$ to matrices of rank n, and $\mathcal{V}_{m\times n}$ be the result of the following random process: choose a random matrix U from $\mathcal{U}_{m\times(m-n)}$ and then choose V to be a random $m \times n$ matrix that spans $\ker(U)$. It is not hard to verify (see, e.g., Feldman 2007) that all three distribution are statistically close.

Fact 4.20. *The distribution $\mathcal{U}_{m\times n}$ is 2^{-m+n}-statistically close to $\mathcal{U}_{m\times n,\mathrm{rk}=n}$, and $\mathcal{U}_{m\times n,\mathrm{rk}=n}$ is 2^{-n}-statistically close to $\mathcal{V}_{m\times n}$.*

Let us extend Definition 3.2 and use $\mathrm{D}_k^\ell(A)$ to denote the distribution of ℓ independent samples from $\mathrm{D}_k(A)$ (i.e., the distribution over ℓ independent random vectors located within distance k from $\mathrm{im}(A)$). Then, the following proposition is a simple implication of Conjecture 4.7.

Proposition 4.21. *Suppose that Conjecture 4.7 holds. Then, for every $m = O(n)$, $k = \Theta(\sqrt{n})$ and a polynomial $\ell(\cdot)$, the ensemble*

$$(A_n, \mathrm{D}_k^\ell(A_n)),$$

where $A_n \in_{\mathrm{R}} \mathcal{U}_{m(n)\times n}$, is computationally indistinguishable from $(A_n, \mathcal{U}_{m(n)})$.

Proof. The case of $\ell = 1$ follows from Theorem 3.3 and Conjecture 4.7 similarly to the proof of Lemma 4.14. The case of a polynomial ℓ follows from the $\ell = 1$ case, via the following fact (which can be proven by a simple hybrid argument): If the ensemble $(X, g(X, Y))$ is pseudorandom then, as long as the function g is efficiently computable, the ensemble $(X, g(X, Y_1), \dots, Y_\ell)$ is also pseudorandom where Y_i is an independent copy of Y and ℓ is polynomial. □

4.4.3. The first cryptosystem. We describe a basic cryptosystem with noticeable decryption error; this error can be reduced to negligible by using standard techniques (see Remark 4.24).

Parameters.
Let $k = \sqrt{\frac{n}{2}}$, and $m = 2n$.

Generation of public/private keys.
Choose $A \in_{\text{R}} \mathcal{U}_{m \times n}$. Generate a random vector $b \in \{0,1\}^m$ within distance k from the image of A: $b = Ax + e$, where $x \in_{\text{R}} \mathcal{U}_n$ and $e \in \{0,1\}^m$ is a random vector of weight k. Let $M = (b|A)$ be $m \times (n+1)$ matrix that results from A by adding the column b, and let B be a random $m \times (m-n-1)$ matrix which spans $\ker(M^{\mathbf{T}})$, i.e., $M^{\mathbf{T}} B = \mathbf{0}^{n+1 \times m-n-1}$.[1]

The *public key* is the matrix B. The *private key* is the pair (B, e).

Encryption.
To encrypt the bit *one* we output a uniform random vector $\mathcal{U}_m$.
To encrypt the bit *zero* we output the random vector $Bx' + e'$, where $x' \in_{\text{R}} \mathcal{U}_{n-1}$ is a random vector and $e' \in \{0,1\}^m$ is a random vector of weight k.

Decryption.
Given a ciphertext $\xi \in \{0,1\}^m$, output *zero* if $e^{\mathbf{T}}\xi = 0$; otherwise, output *one*.

The correctness and security of this cryptosystem are provided by the following lemmas.

[1] Such a matrix can be generated by first finding an arbitrary matrix B_0 that spans $\ker(M^{\mathbf{T}})$ and then multiplying B_0 by a random invertible $(m-n-1) \times (m-n-1)$ matrix T.

Lemma 4.22. *The decryption algorithm returns* zero *on the encrypted zero-message with probability* $2/3$, *and returns* one *on the encrypted one-message with probability* $1/2$, *where in both cases the probability is taken over the randomness used in the encryption process.*

Proof. Let $\xi^1 = \mathsf{Enc}_B(1)$ and $\xi^0 = \mathsf{Enc}_B(0)$. Obviously, the probability that $e^{\mathbf{T}}\xi^1 = 0$ is exactly $1/2$. To estimate the probability that $e^{\mathbf{T}}\xi^0 = 0$ notice that any vector $y = Bx$ in $\ker(A_1^{\mathbf{T}})$ satisfies $e^{\mathbf{T}}y = 0$ since $e^{\mathbf{T}}$ belongs to the span of the rows of $A_1^{\mathbf{T}}$. Thus for $\xi^0 = Bx + e'$ we have

$$e^{\mathbf{T}}\xi^0 = e^{\mathbf{T}}(Bx + e') = e^{\mathbf{T}}e'.$$

Fix a vector e of weight k. Recall that e' is a random vector of weight k, and let p_i denote the probability that e' misses the first i non-zero entries of e. Hence, the probability (over the choice of e') that e and e' have a common one can be written as

$$\sum_{i=0}^{k-1} p_i \frac{k}{m-i} < \frac{k^2}{m-k} < \frac{k^2}{m-k^2} = \frac{1}{3},$$

and the claim follows. □

The security of the scheme is established via the following lemma.

Lemma 4.23. *Conjecture 4.7 implies that the distributions of* $(B, \mathit{Enc}_B(1))$ *and* $(B, \mathit{Enc}_B(0))$ *are computationally indistinguishable.*

Proof. Consider the imaginary experiment in which the matrix M is chosen randomly from $\{0,1\}^{m\times(n+1)}$, and the rest of the key generation and encryption process remains the same. Formally, let $M_1 \in_{\mathrm{R}} \mathcal{U}_{m\times(n+1)}$, let B_1 be a random matrix that spans $\ker(M_1^{\mathbf{T}})$. Then, by Proposition 4.21, the ensemble $(B, \mathsf{Enc}_B(\sigma))$ is computationally indistinguishable from $(B_1, \mathsf{Enc}_{B_1}(\sigma))$ for both $\sigma = 0$ and $\sigma = 1$.

Now consider an experiment in which B_2 is chosen randomly from $\{0,1\}^{m\times(n-1)}$. By Fact 4.20, the ensemble $(B_2, \mathsf{Enc}_{B_2}(\sigma))$ is statistically indistinguishable from $(B_1, \mathsf{Enc}_{B_1}(\sigma))$ for $\sigma \in \{0,1\}$.

Finally, again by Proposition 4.21, $(B_2, \mathsf{Enc}_{B_2}(0))$ is computationally indistinguishable from $(B_2, \mathsf{Enc}_{B_2}(1))$. Hence, by Proposition 2.3, the proof is completed. □

REMARK 4.24. *As shown in Lemma 4.22 the decryption algorithm errs with constant probability. This probability can be reduced by simple repetition. Specifically, we define a new scheme* $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}')$ *in which* $\mathsf{Enc}'_B(b)$ *consists of* k *independent copies of* $\mathsf{Enc}_B(b)$ *and* $\mathsf{Dec}'_{B,e}$ *applies* Dec *to all the copies and output 0 if at least* $5/8$ *of the ciphertexts decrypts to 0. By Chernoff bound, the decryption of the scheme errs with probability* $2^{-\Omega(k)}$*. Since semantic security is preserved under multiple encryptions with the same key, the new scheme remains secure (see Goldreich (2004)).*

4.4.4. The second cryptosystem. Parameters.

Let $m = 2n$, and $H \in \{0,1\}^{0.1m\times m}$ be a parity check matrix of an asymptotically good error correcting code for which there exists an efficient decoding algorithm that corrects up to δm errors, for some constant $0 < \delta < 1/2$. We assume that the rank of $\ker(H)$ is exactly $0.9m$. Let $k = \sqrt{\frac{\delta n}{2}}$.

Generation of public/private keys.

The *public key* consists of a matrix $T \in \mathrm{GF}_2^{m\times r}$ where r is an integer in the interval $[0.4m, 0.5m]$. The *private key* (T, M, E) consists of the public-key matrix T together with two additional matrices $M \in \mathrm{GF}_2^{m\times m}$ and $E \in \mathrm{GF}_2^{m\times m}$. These matrices are generated by the key-generation algorithm Gen as follows:

1. Choose a random matrix A from $\mathcal{U}_{m\times n}$. Choose a random matrix X from $\mathcal{U}_{n\times m}$ and a random $m \times m$ matrix E, in which every column contains exactly k ones.

2. Let $M = AX + E$. If M is not invertible, repeat the process again, unless n unsuccessful iterations has already occurred. In this case, output the all zero matrices, abort and refer to this event as failure.

3. Let $C = HM^{\mathbf{T}}$, let $\mathcal{C} = \ker(C)$ and $\mathcal{A} = \ker(A^{\mathbf{T}})$. Let $\mathcal{T} = \mathcal{A} \cap \mathcal{C}$ and let r be the dimension of $\mathcal{T}$ which, by linear algebra, satisfies $0.4m \leq r \leq 0.5m$. Let T be a random $m \times r$ matrix which spans $\mathcal{T}$.

Encryption.

To encrypt an $0.2m$-bit message w, define an r-bit vector y by appending to w a random suffix $x \in_{\mathrm{R}} U_{r-0.2m}$. Then output $Ty + e$ where $e \in \{0,1\}^m$ is a random vector of weight k.

Decryption.

To decrypt the vector ξ, compute $\hat{v} = E^{\mathbf{T}}\xi$. Apply the efficient decoding algorithm for H to find the nearest codeword v to $\hat{v}$. Find y by solving the linear system $v = M^{\mathbf{T}}Ty$ and output the first $0.2m$ bits of y.

Lemma 4.25 and Lemma 4.26 show that our cryptosystem is a valid encryption scheme that satisfies Definition 4.18.

LEMMA 4.25. *Assuming Conjecture 4.7, the key-generation algorithm fails with negligible probability.*

PROOF. It suffices to show that the success probability is a single iteration is at least $1/3$. It is well known than a matrix which is uniformly chosen from $U_{m \times m}$ is invertible with probability at least $1/4$. Hence, if a single iteration of the key generation succeeds with probability smaller than $1/3$ (for infinitely many n's), we can distinguish M from $U_{m \times m}$ with noticeable advantage in contradiction to Proposition 4.21. □

LEMMA 4.26. *The decryption is correct except with negligible probability.*

PROOF. Fix a vector $y \in \mathrm{GF}_2^r$ and a vector $e \in \mathrm{GF}_2^m$ of weight k. We will show that $\mathsf{Dec}_{(M,T,E)}(Ty + e)$ errs with no more than negligible probability, where the probability is taken over the random choice of E. First we argue that, except with negligible probability, $E^{\mathbf{T}}(Ty + e)$ is δ-close to $M^{\mathbf{T}}Ty$. Indeed, we can write

$$\begin{aligned} E^{\mathbf{T}}(Ty + e) - M^{\mathbf{T}}(Ty) &= (E^{\mathbf{T}} - M^{\mathbf{T}})(Ty) + E^{\mathbf{T}}e \\ &= X^{\mathbf{T}}A^{\mathbf{T}}(Ty) + E^{\mathbf{T}}e = E^{\mathbf{T}}e, \end{aligned}$$

where in the last equality we used the fact that $Ty \in \ker(A^{\mathbf{T}})$. Recall that e contains exactly k ones, and each row of $E^{\mathbf{T}}$ is a random vector of Hamming weight k. Hence, each coordinate of $E^{\mathbf{T}}e$ is 1 independently of the other coordinates with probability at most $k^2/(m-k) < k^2/(m-k^2) < \delta/2$, and therefore, by a Chernoff bound, with all but negligible probability, the Hamming weight of $E^{\mathbf{T}}e$ is smaller than δm.

Now observe that $M^{\mathbf{T}}Ty$ is a codeword since $\mathrm{im}(M^{\mathbf{T}}T)$ is a subset of the linear subspace $M^{\mathbf{T}}(\ker(HM^{\mathbf{T}}) = \ker(H)$. Hence, as long as $E^{\mathbf{T}}(Ty+e)$ is δ-close to $M^{\mathbf{T}}Ty$, the decoding algorithm of H will output $v = M^{\mathbf{T}}Ty$. We complete the proof by noting that $M^{\mathbf{T}}T$ is a matrix of rank r and therefore there exists a unique y which satisfies the system $v = M^{\mathbf{T}}Ty$. □

We proceed with the security proof. We will prove that the scheme is secure even if the adversary sees the matrices M, A and T (but does not see E). That is, from now on let us assume that the key generation algorithm Gen outputs (M, A, T) as a public information. We refer to this scheme as the modified version of Cryptosystem 2.

In order to prove security, we define several hybrid versions of the key generating algorithm. First we consider the algorithm Gen_1 in which we modify the second step of Gen and choose the matrix M uniformly at random from $U_{m\times m}$. Proposition 4.21 implies the following claim.

Claim 4.27. *Assuming Conjecture 4.7, the output of Gen_1 is computationally indistinguishable from the output of Gen.*

In our second hybrid Gen_2 we modify the first step of Gen_1 and choose the matrix A from the distribution $\mathcal{U}_{m\times n,\mathrm{rk}=n}$, defined in Fact 4.20. We also modify the third step and redefine the matrix M as follows: first we choose a random matrix C from $\mathcal{U}_{0.1m\times m,\mathrm{rk}=0.1m}$ and then we let M be the (unique) invertible matrix for which $C = HM^{\mathbf{T}}$. By Fact 4.20, these modifications only cause an exponentially small deviation. Hence, we have,

Claim 4.28. *The output of Gen_2 is statistically indistinguishable from the output of Gen_1.*

Let $\mathcal{R}$ be the distribution of the dimension r in Gen_2. In Gen_3 we modify the third step of Gen_2 as follows. Choose r according to $\mathcal{R}$ and choose a random $m \times r$ matrix T from $\mathcal{U}_{m\times r}$. Choose A uniformly from all the $m \times n$ matrices of rank n for which $\mathrm{im}(T) \subseteq \ker(A^{\mathbf{T}})$, and similarly choose C uniformly from all the $0.1m \times m$ matrices of rank $0.1m$ for which $\mathrm{im}(T) \subseteq \ker(C)$. As in Gen_2, let M be the (unique) invertible matrix for which $C = HM^{\mathbf{T}}$.

Claim 4.29. *The output of Gen_3 is statistically indistinguishable from the output of Gen_2.*

Proof. By Fact 4.20, we may assume that T is chosen from $\mathcal{U}_{m\times r,\mathrm{rk}=r}$ as this causes only negligible deviation in the statistical distance.

Fix some r. Let (C_2, A_2, T_2) (resp., (C_3, A_3, T_3)) be the distribution of C, A and T in Gen_2 (resp. Gen_3) conditioned on $\dim(\mathcal{T}_2 = r)$ (resp. $\dim(\mathcal{T}_3 = r)$). It suffices to show that (C_2, A_2, T_2) is equivalent to (C_3, A_3, T_3). By linear algebra, $\mathrm{im}(T_2)$ is just a random subspace of dimension r and therefore the marginal distributions T_2 and T_3 are equivalent. Also, it is not hard to verify that for every fixed matrix T of rank r, the conditional distributions $[C_2, A_2|T_2 = T]$ and $[C_3, A_3|T_3 = T]$ are equivalent, as in both cases $\ker(C)$ and $\ker(A^{\mathbf{T}})$ are two independent random subspaces subject to $\mathrm{im}(T) \subseteq \ker(C), \mathrm{im}(T) \subseteq \ker(A^{\mathbf{T}})$ and to $\dim(\ker(C)) = 0.9m$ and $\dim(\ker(A^{\mathbf{T}})) = m - n$. □

We can now prove that the scheme is secure.

Lemma 4.30. *Assuming Conjecture 4.7, Cryptosystem 2 (and its modified version) is semantically secure.*

Proof. By Proposition 2.3 and Claim 4.27–Claim 4.29, it suffices to prove the lemma for (M_m, A_m, T_m) which are chosen from $\mathsf{Gen}_3(1^n)$.

Assume towards a contradiction that there exists an adversary B that breaks the security of the scheme. That is, B wins the indistinguishability game of Definition 4.19 with probability $1/2 + \epsilon(n)$ for some non-negligible function $\epsilon(n)$. We will use B to distinguish between the ensembles

$$(Z, \mathrm{D}_{k(n)}(Z)) \text{ and } (Z, \mathcal{U}_{m(n)}), \quad \text{where } Z \in_{\mathrm{R}} \mathcal{U}_{m(n)\times 0.2m}.$$

This will contradict Conjecture 4.5 via Proposition 4.21. Our distinguisher is defined as follows.

- Input: $Z \in \mathrm{GF}_2^{m\times 0.2m}$ and $z \in \mathrm{GF}_2^m$.
- Choose $r \in_{\mathrm{R}} \mathcal{R}$, and then a matrix $W \in_{\mathrm{R}} \mathcal{U}_{m\times 0.2m}$ and a matrix $V \in_{\mathrm{R}} \mathcal{U}_{m\times r-0.4m}$. Let $T = (V, W, Z)$ and compute M and A from T as in Gen_3. Start the indistinguishability game with B with public-key (T, M, A).
- When B outputs the challenges $w_0, w_1 \in \mathrm{GF}_2^{0.2m}$, choose a random bit $\sigma \in \{0,1\}$ and send the challenge $Ww_\sigma + Vv + z$, where $v \in_{\mathrm{R}} \mathcal{U}_{r-0.4m}$.
- If B guesses σ right output "pseudorandom"; otherwise, output "random".

Suppose that $Z \in_{\mathrm{R}} \mathcal{U}_{m(n)\times 0.2m}$ and $z \in_{\mathrm{R}} \mathrm{D}_{k(n)}(Z)$, i.e., $z = Zx + e$ where $x \in_{\mathrm{R}} \mathcal{U}_{0.2m}$ and $e \in \mathrm{GF}_2^m$ is a random vector of weight k. Then, B interacts with the real scheme (instantiated with Gen_3) as the ciphertext that we send can be written as

$$T \cdot \begin{bmatrix} w_\sigma \\ v \\ x \end{bmatrix} + e.$$

Therefore, in this case, our distinguisher outputs "pseudorandom" with probability $1/2 + \epsilon(n)$.

On the other hand, if $Z \in_{\mathrm{R}} \mathcal{U}_{m(n)\times 0.2m}$ and $z \in_{\mathrm{R}} \mathcal{U}_{m(n)}$ then the bit σ is statistically unpredictable (as Ww_σ is encrypted via a "one-time pad"), and therefore we output "random" with probability exactly $1/2$. It follows that we can break the pseudorandomness of $(Z, \mathrm{D}_{k(n)}(Z))$ with advantage $\epsilon(n)$, which completes the proof. □

5. Conclusion and open problems

We have shown several relations between the average case and the worst case complexity. Our results are based on unproved hardness assumptions that are much stronger than P≠NP. However, since our final goal is not to design a cryptosystem secure in practice but to better understand the complexity of several important problems, we believe that it does make sense to study such reductions. In particular it would be very interesting to see any positive algorithmic results on Problem 4.1–Problem 4.6 as well as to obtain any other reductions that would yield more information on the average complexity of these problems.

Our planted construction in Average-3LIN problem can be used as a challenge for empirical SAT solving algorithms. In particular, we believe that any SAT heuristic that does not invoke Gaussian elimination as a subroutine will fail on this example even if we do not add any noise and the overdetermined linear system is satisfiable. Note that in the case of randomly planted SAT assignment there are non-trivial algorithms that solve it if the density is sufficiently large (Flaxman 2003).

We are also interested in the following question that can be considered as a step toward explicit lower bounds: construct in polynomial time a sequence (A_n, b_n) s.t. $A_n \in \{0,1\}^{m\times n}$ is expander in which every row contains finitely many ones and b_n is a vector for which the system $A_n x = b_n$ has at most $2/3m$ satisfiable equations. In other words, construct any explicit sequence that do not belong to the image of pseudorandom generator (cf. Alekhnovich *et al.* 2000).

Finally, by the analogy with Tseitin tautologies for propositional calculus, one can define Pseudo-Tseitin tautologies, which state that a given linear system is unsatisfiable in the strong sense: there is no satisfiable subsystem that contains almost all linear equations. Formalized as in Alekhnovich *et al.* (2000) Pseudo-Tseitin tautologies may be a curious candidate for proving lower bounds in propositional calculus.

Acknowledgements

We would like to thank Amir Shpilka for introducing to us the notion of matrix rigidity. We are also grateful to Madhu Sudan for helpful discussions.

Supported in part by NSF Awards CCR 0205390 and MIT NTT Award 2001–2004.

References

M. Alekhnovich, E. Ben-Sasson, A. Razborov & A. Wigderson. Pseudorandom generators in propositional complexity. In *Proceedings of the 41st IEEE FOCS*, 2000. Journal version to appear in *SIAM Journal on Computing*.

S. Arora, L. Babai, J. Stern & Z. Sweedy. Hardness of Approximate Optima in Lattices, Codes, and Linear Systems. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

M. Ajtai & C. Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. In *Proc. on 29th Annual ACM Symposium on Theory of Computing*, 284–293, 1997.

S. Arora, C. Lund, R. Motwani, M. Sudan & M. Szegedy. Proof Verification and Hardness of Approximation Problems. *Journal of ACM*, 45(3):501–555, 1998.

S. Arora & S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of ACM*, 45(1):70–122, 1998.

E. Ben-Sasson, P. Harsha & S. Raskhodnikova. Some 3CNF Properties are Hard to Test. To appear in *Proc. on 35th Annual ACM Symposium on Theory of Computing*, 2003.

U. Feige. Relations between average case complexity and approximation complexity. In *Proc. on 34th Annual ACM Symposium on Theory of Computing*, 534–543, 2002.

V. Feldman. Attribute-Efficient and Non-adaptive Learning of Parities and DNF Expressions. *Journal of Machine Learning Research*, 8:1431–1460, 2007.

A. Flaxman. A spectral technique for random satisfiable 3CNF formulas. In *Proc. on 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.

J. Friedman. A note on matrix rigidity. *Combinatorica*, 13(2):235–239, 1993.

D. Gollman & W. Chambers. Clock-controlled shift registers: a review. *IEEE Journal on Selected Areas in Communications*, 7(4):525–533, 1989.

O. Goldreich. *Foundations of Cryptography: Basic Applications.* Cambridge University Press, Cambridge, 2004.

O. Goldreich & S. Goldwasser. On the Limits of Non-Approximability of Lattice Problems. In *Proc. on 30th Annual ACM Symposium on the Theory of Computing*, 1–9, 1998.

S. Goldwasser & S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

J. Håstad. Dual Vectors and Lower Bounds for the Nearest Lattice Point Problem. *Combinatorica* 8 (1):75–81, 1988.

J. Håstad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001.

B. Kashin & A. Razborov. Improved lower bounds on the rigidity of Hadamard matrices. *Mathematical Notes*, 63(4):471–475, 1998.

S. Lokam. Spectral Methods for Matrix Rigidity with Applications to Size-Depth Trade-offs and Communication Complexity. *Journal of Computer and System Sciences*, 63(3):449–473, 2001.

J. Lagarias, H. Lenstra & C. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.

R. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. Deep Space Network Progress Report 42–44, Jet Propulsion Lab., California Institute of Technology, 114–116, 1978.

J. Naor & M. Naor. Small-Bias Probability Spaces: Efficient Constructions and Applications. *SIAM Journal of Computing*, 22(4):838–856, 1993.

P. Pudlak & Z. Vavřin. Computation of rigidity of order n/r for one simple matrix. *Comm. Math. Univ. Carol.*, 32(2):213–218, 1991.

I. Pak & V. Vu. On mixing of certain random walks, cutoff phenomenon and sharp threshold of random matroid processes. *Discrete Applied Math.*, 110:251–272, 2001.

A. Razborov. On rigid matrices. Manuscript (in Russian), 1989.

A. Razborov & S. Rudich. Natural Proofs. Journal of Computer and System Sciences, 55(1):24–35, 1997.

M. Shokrollahi, D. Spielman & V. Stemann. A Remark on Matrix Rigidity. *Information Processing Letters*, 64(6):283–285, 1997.

Г. С. Цейтин. О сложности вывода в исчислении высказываний. In А. О. Слисенко, editor, Исследования по конструктивной математике и математической логике, II; Записки научных семинаров ЛОМИ, т. 8, pages 234–259. Наука, Ленинград, 1968. Engl. translation: G. C. Tseitin, On the complexity of derivations in propositional calculus, in: *Studies in mathematics and mathematical logic, Part II*, ed. A. O. Slissenko, pp. 115–125.

L. Valiant. Graph-Theoretic Arguments in Low-Level Complexity. In *Proc. 6th Symposium on Mathematical Foundations of Computer Science*, 162–176, 1977.

Manuscript received 24 June 2009

Michael Alekhnovich
Laboratory for Computer Science,
MIT, Cambridge, MA, USA