

# Public Key Cryptography from Different Assumptions

Benny Applebaum\*      Boaz Barak†      Avi Wigderson‡

December 31, 2008

## Abstract

This paper attempts to broaden the foundations of public-key cryptography. We construct a new public key encryption based on two “hardness on average” assumptions: **(1)** it is hard to “learn parity with noise” for random sparse equations, and **(2)** it is hard to approximate the vertex expansion of random unbalanced bipartite graphs.

More precisely, we show that a semantically secure public-key encryption scheme is implied by the following:

For some  $m > n > d$  and  $q, \mu$  satisfying  $\mu \ll 1/q$

1. It is hard to distinguish between **(a)** a random set of  $m$  equations on  $n$  variables in  $\text{GF}(2)$  such that each equation involves  $d$  variables, and **(b)** a random set of such equations that have a solution satisfying  $1 - \mu$  fraction of them.
2. It is hard to distinguish between **(a)** a random bipartite graph whose left and right sides have  $m$  and  $n$  vertices respectively and whose left degree is  $d$ , and **(b)** a random such graph that has a subset of  $q$  left vertices with at most  $q - 1$  neighbors.

We also construct a public-key encryption scheme based on a variant of Assumption 1 with *non-linear* equations (and no noise) as long as Assumption 2 holds for  $q = O(\log n)$ .

Most, if not all, previous constructions of public key encryption used hardness assumptions with significant algebraic structure. Our new assumptions, positing indistinguishability from uniform of certain natural distributions on instances of **NP**-complete problems, seem relatively unstructured and qualitatively different from previous ones.

We give some evidence for these assumptions by studying their resistance to certain natural algorithms, and relating them to variants of more widely studied assumptions such as the hardness of the “search version” of learning parity with noise and the planted clique problems.

**Keywords:** Public key encryption, expander graphs, cryptography in  $\text{NC}_0$ , lossless expanders, unbalanced expanders, planted problems

---

\*Department of Computer Science, Princeton University, [benny.applebaum@gmail.com](mailto:benny.applebaum@gmail.com). Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797.

†Department of Computer Science, Princeton University, [boaz@cs.princeton.edu](mailto:boaz@cs.princeton.edu). Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

‡Institute for Advanced Study, Princeton, NJ, [avi@ias.edu](mailto:avi@ias.edu).

# 1 Introduction

*Public key encryption* is a central notion in cryptography – indeed security of current electronic commerce is based on it. Fortifying the foundations of public-key cryptography (namely resting them on widely believed complexity assumptions) is thus a central goal. However, despite 30 years of research, very few candidates for such encryptions are known, and these are based on a handful of computational problems of a very structured, algebraic nature, from the areas of number theory, lattices, and error-correcting codes (e.g., [DH76, RSA78, McE78, AD97]). We note that obtaining public key encryption from one-way functions (for which many more candidates exist) is a longstanding open problem, and cannot be achieved via a black-box reductions [IR89].

In this work we give a construction of a public key encryption based on different assumptions (of which we have several variants). The proposed system is not as efficient as some known candidate constructions, and is based on assumptions that are not as well-studied as, say, the hardness of factoring. For this reason we initiate here a study of the algorithmic and pseudorandomness questions which arise, relate them to known results, and obtain some preliminary new ones. The main advantage of the new scheme is the relatively general and unstructured nature of the new assumptions: one related to the pseudorandomness of certain simple generators (based on nonlinear  $\mathbf{NC}^0$  maps or linear-with-noise maps), and the second based on the hardness of detecting small non-expanding sets in sparse random graphs. These seem qualitatively different than previous assumptions. In particular, while many known public key cryptosystems can be broken by efficient quantum adversaries or by algorithms in  $\mathbf{AM} \cap \mathbf{coAM}$ , our system seems (at least superficially) to lack the structure used by such attacks. For more on this see Section 1.4.

In short, we hope that this new proposal, even if broken, will open the door to many other proposals which break away from the existing algebraic mold.

## 1.1 Our assumptions

We introduce two new hardness assumptions that we call DSPN (for “Decisional Sparse Parity with Noise”) and DUE (for “Decisional Unbalanced Expansion”). Both assumptions involve a natural distribution over unbalanced bipartite graphs of small degree. Specifically, let  $\mathcal{G}_{m,n,d}$  be the uniform distribution over bipartite graphs  $G = (V_{\text{Out}}, V_{\text{In}}, E)$  with  $m$  output vertices  $V_{\text{Out}}$ ,  $n$  input vertices  $V_{\text{In}}$ , and output-degree equal to  $d$  (i.e.,  $|V_{\text{Out}}| = m$ ,  $|V_{\text{In}}| = n$  and  $|E| = dm$ ). Hereafter we refer to such graphs as  $(m, n, d)$ -graphs. For now think of the degree  $d$  as a constant, and on the imbalance ratio  $m/n$  as a larger constant.

### 1.1.1 The DSPN assumption

We can view an  $(m, n, d)$  graph  $G$ , as the underlying graph of a Boolean circuit. We will be interested in the case where each output gate computes the parity function  $\oplus$  and flips the result with probability  $\mu \ll 1/2$ . Formally, for an  $(m, n, d)$  graph  $G$ , we let  $G_{\oplus\mu}$  denote the probabilistic function from  $\{0, 1\}^n$  to  $\{0, 1\}^m$  obtained by mapping every  $x \in \{0, 1\}^n$  to  $Gx + e$  (identifying  $G$  with its adjacency matrix) where  $e = (e_1, \dots, e_m) \in \{0, 1\}^m$  is a *noise* vector in which each bit  $e_i$  is chosen independently with  $\Pr[e_i = 1] = \mu$ .

Our first assumption states that, when  $G$  is chosen from  $\mathcal{G}_{m,n,d}$  and  $x$  is a random  $n$ -bit string, the distribution of the pair  $(G, G_{\oplus\mu}(x))$  is pseudorandom. The assumption depends on several parameters, namely the values  $n, m, d, \mu$ , and also  $\epsilon > 0$  that is the indistinguishability bias. We

will elaborate later on the recommended choice of these parameters as a function of  $n$  (which serves as a single security parameter).

**Assumption DSPN**( $m, d, \mu, \epsilon$ ) (*Decisional Sparse Parity with Noise*). The distribution  $(G, G_{\oplus\mu}(x))$  is  $\epsilon$ -indistinguishable from the distribution  $(G, y)$  where  $G \in_{\mathbb{R}} \mathcal{G}_{m,n,d}$ ,  $x$  is randomly chosen from  $\{0, 1\}^n$ , and  $y$  is randomly chosen from  $\{0, 1\}^m$ .

Here and throughout this paper, we use the standard cryptographic notion of  $\epsilon$ -indistinguishability. That is, two sequences of distributions  $\{\mathcal{X}_n\}$  and  $\{\mathcal{Y}_n\}$  are  $\epsilon$ -indistinguishable if they cannot be distinguished with bias better than  $\epsilon$  by any polynomial-sized circuit family. (See Section 2 for more precise definitions.) However, one may hope the assumption holds even for subexponential circuits (below the trivial limit  $m^{\mu m}$  of exhaustive search).

### 1.1.2 The DUE assumption

Graphs chosen from  $\mathcal{G}_{m,n,d}$  will be, with high probability, very good expanders. That is, we expect that small sets  $S$  of output vertices will have almost  $d|S|$  neighbors. The distribution  $\mathcal{F}_{n,m,d}^q$  is a perturbed version of  $\mathcal{G}_{m,n,d}$  in which we plant a single  $q$ -size output subset  $S$  with a small (“shrinking”) neighborhood. Formally,  $\mathcal{F}_{n,m,d}^q$  is the result of the following random process: choose  $G$  from  $\mathcal{G}_{n,m,d}$ , choose at random subsets  $S \subseteq V_{\text{Out}}$  and  $T \subseteq V_{\text{In}}$  of sizes  $q$  and  $q-1$  respectively, and choose a random  $(q, q-1, d)$ -graph  $H$ . Then replace all edges in  $G$  that are adjacent to  $S$  with the edges from  $H$ . Our second assumption states that a random graph from  $\mathcal{G}_{n,m,d}$  is indistinguishable from a random graph in  $\mathcal{F}_{n,m,d}^q$ .

**Assumption DUE**( $m, d, q, \epsilon$ ) (*Decisional Unbalanced Expansion: Hardness of planted shrinking sets*). The distributions  $\mathcal{G}_{n,m,d}$  and  $\mathcal{F}_{n,m,d}^q$  are  $\epsilon$ -indistinguishable.

Again, one may hope the assumption holds even for subexponential circuits (below the trivial limit  $n^q$ ).

### 1.1.3 Choosing the parameters

Our assumptions impose conflicting constraints regarding the values of the parameters. Indeed, by using simple transformations, one can show that increasing  $m$  benefits DUE (i.e., makes it more plausible) but hurts DSPN. Also, at least intuitively, increasing  $d$  benefits DSPN but hurts DUE.<sup>1</sup> Moreover, our public-key scheme forces us to upper bound the product  $\mu q$  by a small constant, which again creates a conflict as DSPN benefits from a larger  $\mu$  while DUE benefits from a larger  $q$ .

The parameters, as well as their suggested values and restrictions are listed in Figure 1. For simplicity we did not state here the most general choices and tradeoffs possible among the different parameters.<sup>2</sup>

<sup>1</sup>On the extreme, when  $d = n$  the DSPN assumption asserts that the well studied problem of decoding random linear code is hard, while DUE becomes trivially false even when  $d > q$ .

<sup>2</sup>Most notably, we can relax the DSPN assumption and use a constant noise rate  $\mu$  at the price of making the system vulnerable to quasipolynomial-time attacks .

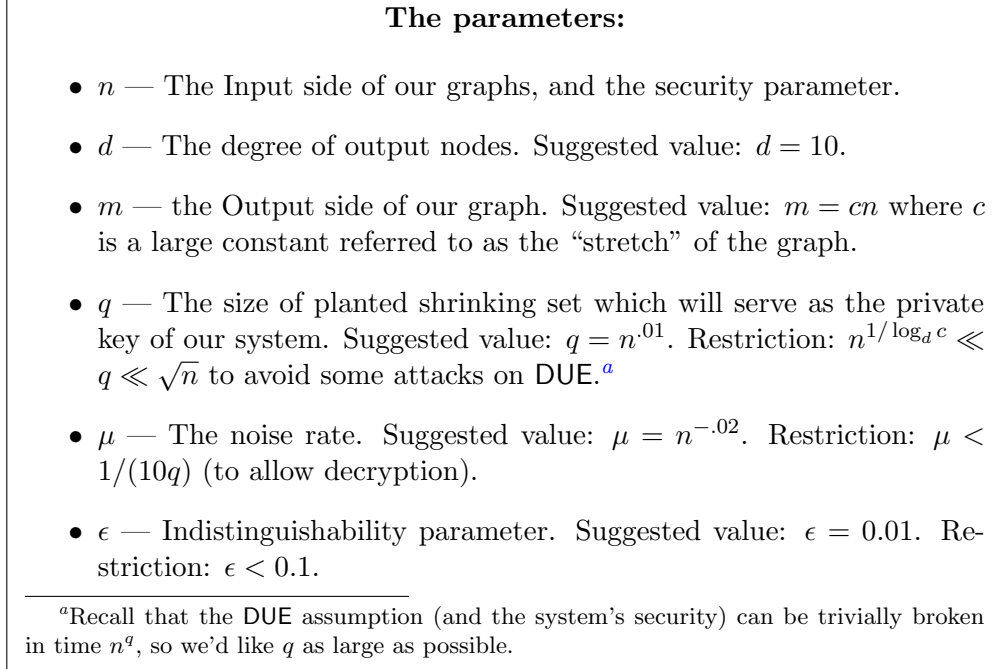


Figure 1: Summary of the parameters including restrictions and suggested values.

## 1.2 Our construction

The DSPN and DUE assumptions allow us to obtain a public key encryption scheme:

**Theorem 1.1.** *Suppose that Assumptions DSPN( $m, d, \mu, \epsilon$ ) and DUE( $m, d, q, \epsilon$ ) are true with parameters  $m, d, \mu, q, \epsilon$  where  $\epsilon \leq 1/20$  and  $q\mu < 1/20$ , then there exists a semantically secure public-key encryption scheme.*

**Proof idea.** We will construct a bit encryption scheme which can be converted to a full fledged scheme which supports encryption of polynomially long messages by using standard techniques (e.g., concatenation). Our basic scheme is defined in Figure 2. By the amplification theorem of [HR05] it suffices to show that the success probability of the decryption (i.e., *correctness*) is much larger than the probability that an adversary guesses the plaintext (i.e., *security*). Indeed, we decrypt the bit  $b$  from its encryption vector  $y$ , by checking if  $y_S$ , the projection on the secret subset  $S$ , can have a preimage. This is unlikely in the case where  $b = 0$  (as  $S$  is shrinking), but very likely when  $b = 1$ , since noise is small and almost surely will not occur on these coordinates. On the other hand, security follows from combining the two assumptions, thus DUE implies that encrypting using  $\hat{G} \in_{\mathbb{R}} \mathcal{F}_{n,m,d}^q$  is indistinguishable from encrypting using  $G \in_{\mathbb{R}} \mathcal{G}_{n,m,d}$ , which by DSPN is secure. The full proof of security is in Section 3.2.

**Alternative construction.** We also present a variant of our cryptosystem that works when DSPN is replaced by an assumption of pseudorandomness of a certain deterministic (i.e., noiseless) *non-linear* map from  $\{0, 1\}^n$  to  $\{0, 1\}^m$  (instead of a linear map plus noise). The decryption

### Basic Scheme:

**Public encryption key:** A random  $(m, n, d)$ - graph  $G = (V_{\text{Out}}, V_{\text{In}}, E)$  sampled from distribution  $\mathcal{F}_{n,m,d}^q$  of Assumption DUE.

**Private decryption key:** The planted  $q$ -sized subset  $S \subseteq V_{\text{Out}}$  whose neighborhood  $\Gamma(S)$  has cardinality smaller than  $q$ , and  $H$ , the subgraph of  $G$  induced by the vertices of  $S \cup \Gamma(S)$ .

**Encryption:** To encrypt 0 send a random string in  $\{0, 1\}^m$ . To encrypt 1, choose  $r \in_{\mathbb{R}} \{0, 1\}^n$  and send  $G_{\oplus \mu}(r)$ .

**Decryption:** To decrypt  $y \in \{0, 1\}^m$ , we output 1 if and only if the projection of  $y$  on  $S$  is in the image of the adjacency matrix of  $H$ .

Figure 2: The basic cryptosystem using  $\text{DSPN}(m, d, \mu, \epsilon)$  and  $\text{DUE}(m, d, q, \epsilon)$ . It achieves  $3/4 - q\mu/2$  correctness and  $3\epsilon$ -privacy as per Definition 3.1.

efficiency in this variant requires the DUE assumption to hold for sets of  $S$  of size  $O(\log n)$ , and this result can at best guarantee  $n^{\Omega(\log n)}$  security; see Section 3.3.2 for more details on this construction.

**Efficiency.** We remark that our encryption scheme, though polynomial-time, is very inefficient, and requires very long ciphertexts to achieve reasonable security. Improving the efficiency of the construction, and understanding the optimal setting of parameters, is an important research direction.

### 1.3 Validity of assumptions

Next we turn to studying whether these assumptions are true, and if so in what range of parameters. We show that at least some natural types of algorithms will fail to refute these assumptions, and we present some relations between them and variants of other widely studied computational problems. Our results are stronger for the case of the DSPN assumption.<sup>3</sup> First we show in Section 4.2 that in our setting, “one-wayness” implies pseudorandomness.

**Informal Theorem 1** (See Theorem 4.4). Assumption DSPN is implied by the assumption that it is hard to *find* a solution to a random satisfiable set of sparse linear equations perturbed by a small noise. In other words, we can replace the “decisional” type assumption DSPN with a similar assumption that is of a “search” type.

A similar reduction (from search to decision) was presented in [BFKL94] for the non-sparse case, however their techniques do not hold in the sparse case. We believe that the current reduction might turn useful in other contexts as well.

<sup>3</sup>DSPN was also more studied by other works. The non-sparse version of DSPN is equivalent to the hardness of decoding random-linear code (or learning parity with noise) which is widely believed to be hard on average (cf. [GKL93, KMR<sup>+</sup>94, BFKL94, GOL04, HB01, JW05, KS06]). In fact, close variants of the DSPN assumption have already appeared in [ALE03, Conjecture 1] and in [AIK06].

We continue in Section 4.1 by showing that the distribution  $\text{Image}(G)+\text{noise}$ , passes some standard pseudorandomness tests. In particular:

**Informal Theorem 2** (See Theorem 4.2). Assumption DSPN cannot be refuted using distinguishers that are low degree polynomials, or by myopic distinguishers that apply *any* test to any  $o(n)$  coordinates. Moreover, under a conjecture of Linial and Nisan [LN90], DSPN cannot be refuted using distinguishers that are  $\mathbf{AC}^0$  circuits.

We then study the DUE assumption. It is a fairly natural average-case variant of the well-known vertex expansion problem. We start by showing in Section 5.1 that at least cycle-counting algorithms (that work well for testing expansion in random *balanced* graphs) cannot refute the DUE assumption:

**Informal Theorem 3** (See Theorem 5.1). Assumption DUE cannot be refuted by efficient algorithms that are based on counting cycles in the graph  $G$ .

We also show some additional “circumstantial evidence” for DUE by relating it to “small set expansion” and “planted clique” problems:

**Informal Theorem 4** (See Theorems 5.3 and 5.2). Variants of Assumption DUE are implied by variants of the planted clique problem in random graphs and approximating vertex expansion of small sets in general (not necessarily bipartite) graphs.

## 1.4 Discussion and prior work

Besides supporting the strength of our assumptions, we must explore to which extent they have less or different “structure” than previously used assumptions. Do they actually “expand” the foundations of public-key cryptography? We discuss this (necessarily informal) issue here. We do not review all previous assumptions used for candidates for public key encryption; see the survey [Zhu01] and the web site [Lip97] for more. It seems that currently those candidates that are considered secure can be classified as falling into two broad categories: schemes based on number theoretic or group theoretic problems such as factoring (e.g. [Rab79, RSA78]) and discrete log in various groups (e.g. [DH76, MIL85, Kob87]) and schemes based on knapsack/lattices/error correcting codes (e.g., [MCE78, AD97, ALE03, REG05]). Our scheme is more similar to schemes of the latter type but there are some important differences.

In the worst-case setting, “lack of structure” is captured nicely by  $\mathbf{NP}$ -completeness. In the average-case setting we don’t have a fully satisfactory analog, though it does seem that some  $\mathbf{NP}$ -complete problems (e.g., 3SAT) have natural distributions on which they are hard. Thus we feel that it would be a breakthrough to base a public key cryptosystem on, say, the hardness of finding assignments for a random 3CNF with number of clauses close to the satisfiability threshold. The DSPN assumption and its non-linear variant DSF (below) do seem at least close in spirit to this assumption, for different constraint-satisfaction problems. As for DUE, its nature seems more combinatorial than algebraic, being a basic question about expansion. Still, while we know of many works on related questions, this particular one deserves more scrutiny, being far less studied than “parity with noise”.

Average-case assumptions with “planted” structures such as DSPN and DUE immediately imply hardness of approximation results. Even though there is no reduction in the other direction, we believe that some insight into the structure or lack thereof of an average-case problem could be

gained from the hardness of corresponding approximation/gap problems. For DUE, while testing expansion is (co-)NP-hard, we do not have strong hardness of approximation results. For this reason we relate it to the better studied planted clique problems, and to (possibly more accessible) non-expansion of small sets in standard (not unbalanced bipartite) graphs. For DSPN however, a related well studied gap problem is  $d$ LIN. The input is a set of  $m$  equations, each depending on at most  $d$  of the  $n$  variables, and one needs to decide whether there is an assignment that satisfies least a  $1 - \mu$  fraction of them, or every assignment satisfies at most  $1/2 + \mu$  fraction (these two cases roughly correspond to decrypting 1 and 0 respectively). This is known to be NP-hard for  $d = 3$  and  $\mu = (\log n)^{-\Omega(1)}$  [Hås97] (using quasipolynomial reductions) and  $\mu = (\log \log n)^{-\Omega(1)}$  [MR08] (using polynomial reductions). It is possible the problem remains hard (possibly under slower reductions) for much smaller noise, perhaps down to  $\mu \sim n^{-\epsilon}$ . (For the related *nearest codeword* and *closest vector* problems, hardness for  $\mu = n^{-1/\log \log n}$  is known [ABSS93, DKRS03].)

A different way of defining structure in a cryptosystem is to ask what complexity consequences it has. Any secure public-key system implies  $\mathbf{NP} \not\subseteq \mathbf{BPP}$ . But many, if not most public-key systems in use, if secure, imply the (seemingly) much stronger conditions  $\mathbf{AM} \cap \mathbf{coAM} \not\subseteq \mathbf{BPP}$  or  $\mathbf{BQP} \not\subseteq \mathbf{BPP}$ . These consequences hold for all factoring and discrete log based systems, and the first holds also for all lattice-based systems [GK90, GG98, AR04, Sho97]. Our preliminary attempts to find such consequences for our system failed (though of course more effort is required). However, we note that other coding-based schemes such as [McE78, ALE03] also seem to resist such implications.

**Comparison with [McE78, ALE03].** Our scheme seems most related to coding-based cryptosystems such as the systems of McEliece [McE78] and Alekhnovich [ALE03]<sup>4</sup> and we now turn to compare our scheme with those. In short, we believe that our scheme is more combinatorial in nature than the McEliece cryptosystem, and is far more flexible (in noise levels) than Alekhnovich’s scheme (at the expense of introducing the DUE assumption). Additionally, our nonlinear variant seems to be qualitatively different than both. A more detailed comparison follows.

Both, the McEliece cryptosystem and Alekhnovich’s cryptosystem, are based on the (more standard) variant of the DSPN assumption where one does not require the equations to be sparse. However, McEliece’s system requires the additional assumption that, loosely speaking, one cannot distinguish a random matrix from a “shuffled” random generator matrix for a Goppa code (the shuffling randomly changes the input basis and permutes the output coordinates)<sup>5</sup>. This assumption seems to have more algebraic structure than our DUE assumption. In contrast, Alekhnovich has no additional assumption and uses solely the (non-sparse variant of the) DSPN assumption. However, he requires the assumption to hold even for very low noise levels, namely  $\mu < 1/\sqrt{m}$  (where  $m$  is the number of equations). While there are no known good algorithms even for this noise level, this is well beyond what is supported by corresponding worst-case hardness of approximation results. Moreover, there is a natural analog of the noisy linear equations problem, namely the *shortest vector problem (SVP)* in integer lattices, where  $\mu = 1/\sqrt{m}$  is a natural barrier for NP-hardness

---

<sup>4</sup>Indeed, as observed by Ron Rivest and Madhu Sudan (personal communication), both our scheme and Alekhnovich’s have a common generalization, where the public key is a matrix  $G$  whose dual subspace has a “planted” short vector, which serves as the private key. Similar structure occurs in many lattice-based cryptosystems such as [AD97, REG04], where the public key is roughly a generating set for a lattice whose dual lattice has a planted short (in  $\ell_2$ ) vector.

<sup>5</sup>This seems to be the cleanest formulation on the assumptions behind McEliece’s system; see [CFS01]. While the system can be instantiated with any efficiently decodable code, some choices for the code will make the system insecure.

results [GG98, AR04]. (Indeed, Alekhovich’s scheme was motivated by the Ajtai-Dwork lattice-based scheme that cannot be **NP**-hard to break for this reason.) We note that Regev [REG05] gave another public key cryptosystem based on the hardness of solving noisy linear equations over *large* alphabet. However he also needed a noise  $\mu$  of magnitude  $< 1/\sqrt{m}$ , and in fact the security of his scheme can be easily shown to imply  $\mathbf{AM} \cap \mathbf{coAM} \not\subseteq \mathbf{BPP}$ .

In contrast to this (necessarily) low noise in the Alekhovich and Regev schemes, our addition of the (arguably more combinatorial) DUE assumption allows us to tune the noise parameter  $\mu$  to much larger values such as  $\mu = m^{-\epsilon}$  for any  $\epsilon > 0$ , and in some variants, even up to any constant bounded away from  $1/2$  or use *nonlinear* equations (though those variants can guarantee at best  $n^{\Omega(\log n)}$  security).<sup>6</sup>

## 2 Preliminaries

**Graphs.** For a subset  $S$  of vertices in a graph  $G$ , we denote by  $\Gamma_G(S)$  the set of neighbors of  $S$  in  $G$  (we sometimes drop the subscript  $G$  when it is clear from the context). Throughout this paper, we identify bipartite graphs with their adjacency matrices. Thus for every bipartite graph  $G = (V_{\text{Out}}, V_{\text{In}}, E)$ , we denote by  $G$  also the  $|V_{\text{Out}}| \times |V_{\text{In}}|$  matrix over  $\text{GF}(2)$  such that for every  $u \in V_{\text{Out}}, v \in V_{\text{In}}, G_{u,v} = 1$  if and only if  $(u, v) \in E$ .

**Statistical distance and computational indistinguishability.** We use  $U_n$  to denote the uniform distribution over  $\{0, 1\}^n$ . The *statistical distance* between discrete probability distributions  $\mathcal{X}$  and  $\mathcal{Y}$ , denoted  $\Delta(\mathcal{X}, \mathcal{Y})$ , is defined as the maximum, over all functions  $C$ , of the *distinguishing advantage*  $|\Pr[C(\mathcal{X}) = 1] - \Pr[C(\mathcal{Y}) = 1]|$ . We say that  $\mathcal{X}$  and  $\mathcal{Y}$  are  $(\epsilon, T)$ -indistinguishable if for every Boolean circuit  $C$  of size at most  $T$ ,  $|\Pr[C(\mathcal{X}) = 1] - \Pr[C(\mathcal{Y}) = 1]| < \epsilon$ . We say that two sequences of distributions  $\mathcal{X}_n, \mathcal{Y}_n$  (where  $n$  is an implicit or explicit security parameter) are  $\epsilon$ -indistinguishable if for every constant  $c$  and any large enough  $n$ ,  $\mathcal{X}_n$  and  $\mathcal{Y}_n$  are  $(\epsilon, n^c)$ -indistinguishable. A sequence of distributions  $\mathcal{X}_n$  is  $\epsilon$ -pseudorandom if  $\mathcal{X}_n$  is  $\epsilon$ -indistinguishable from  $U_n$ , the uniform distribution over  $n$  bits.

## 3 Construction of public-key encryption scheme

In this section we prove our main theorem (Theorem 1.1) and discuss some extensions and variants of our construction.

### 3.1 Definition

A public-key encryption scheme allows two parties to communicate securely without sharing a secret key. Such a scheme should satisfy correctness – legitimate users should be able to decrypt ciphertexts correctly; and privacy – adversaries who see a ciphertext should learn nothing about the content of the message. We follow [HR05] and quantify the correctness and privacy by two

---

<sup>6</sup>The larger  $\mu$  is, the smaller the set size  $q$  in the DUE assumption, which can always be broken trivially in  $n^q$  time. However, if the stretch  $c = m/n$  is large enough, then this trivial algorithm is essentially the best algorithm we know for DUE. See also Section 5.1.

error parameters  $\alpha$  and  $\beta$ .<sup>7</sup> Our definition becomes equivalent to the standard notion of semantic security [GM82] when both parameters are taken to be negligible, i.e., when  $\alpha$  and  $\beta$  go down to zero faster than any inverse polynomial.

**Definition 3.1.** A  $(\alpha(n), \beta(n))$ -secure public-key bit encryption scheme is a triple  $(\text{Gen}, \text{Enc}, \text{Dec})$  of probabilistic polynomial time algorithms such that

- Algorithm  $\text{Gen}$ , on input  $1^n$  produces a pair  $(\text{pk}, \text{sk})$ .
- $((1 - \alpha)$ -correctness) For a random bit  $b \in_{\text{R}} \{0, 1\}$ ,  $\Pr[\text{Dec}_{\text{sk}}(\text{Enc}_{\text{pk}}(b)) = b] > 1 - \alpha(n)$ , where  $(\text{pk}, \text{sk}) \in_{\text{R}} \text{Gen}(1^n)$  and the probability is over the randomness of  $\text{Gen}$ ,  $\text{Enc}$ ,  $\text{Dec}$ , and the choice of  $b$ .
- $(\beta$ -privacy) The distributions  $(\text{pk}, \text{Enc}_{\text{pk}}(0))$  and  $(\text{pk}, \text{Enc}_{\text{pk}}(1))$  are  $\beta(n)$ -indistinguishable, where  $(\text{pk}, \text{sk}) \in_{\text{R}} \text{Gen}(1^n)$ .

If  $\alpha(n)$  and  $\beta(n)$  are both negligible, we say that the scheme is *semantically secure*.

As mentioned in the introduction, a bit encryption scheme can be converted to a full fledged scheme which supports encryption of polynomially long messages by using concatenation.

### 3.2 Main theorem

We can now prove our main theorem which is a slightly stronger version of Theorem 1.1.

**Theorem 3.2** (Thm. 1.1 restated). *Suppose that Assumptions  $\text{DSPN}(m, d, \mu, \epsilon)$  and  $\text{DUE}(m, d, q, \epsilon)$  are true with parameters  $m, d, \mu, q, \epsilon$  where  $\epsilon \leq \frac{(1-2\mu)^{2q}}{12}$ , then there exists a semantically secure public-key encryption scheme.*

Note that the condition  $\epsilon \leq \frac{(1-2\mu)^{2q}}{12}$  relaxes the condition that appears in the original statement of the theorem (Thm. 1.1).

*Proof.* We will use the following scheme (which is a close variant of the scheme presented in Section 1.2):

**Modified Scheme:**

**Public encryption key:** A random  $(m, n, d)$ - graph  $G = (V_{\text{Out}}, V_{\text{In}}, E)$  sampled from distribution  $\mathcal{F}_{n,m,d}^q$  of Assumption DUE.

**Private decryption key:** The planted  $q$ -sized subset  $S \subseteq V_{\text{Out}}$  whose neighborhood  $\Gamma(S)$  has cardinality smaller than  $q$ , and  $H$ , the subgraph of  $G$  induced by the vertices of  $S \cup \Gamma(S)$ .

**Encryption:** To encrypt 0 send a random string in  $\{0, 1\}^m$ . To encrypt 1, choose  $r \in_{\text{R}} \{0, 1\}^n$  and send  $G_{\oplus \mu}(r)$ .

**Decryption:** Let  $v \in \{0, 1\}^q$  be a non-zero vector orthogonal to the columns of the adjacency matrix of  $H$ . To decrypt  $y \in \{0, 1\}^m$ , we output 1 if and only if the projection of  $y$  on  $S$  is orthogonal to  $v$ .

<sup>7</sup>Our definitions of  $(1 - \alpha)$ -correctness and  $\beta$ -privacy are not exactly the same as in [HR05, Def. 8], but are equivalent up to a simple linear transformation.

It is clear that the scheme allows efficient encryption. Decryption is well defined (and is therefore also efficient) as the dimension of the matrix  $H$  is at most  $q - 1$  (being a linear function that maps  $q - 1$  bits to  $q$  bits). We prove that the scheme is  $(1 - \alpha)$ -correct and  $\beta$ -private for  $\alpha = \frac{2 - (1 - 2\mu)^q}{4}$  and  $\beta = 3\epsilon$ .

**Correctness.** We analyze the success probability of decryption. The decryption algorithm outputs 1 if some non-trivial linear equation (over  $\text{GF}(2)$ ) of the form  $\langle v, y_S \rangle = 0$  holds. When  $b = 1$  decryption errs whenever the sum of the noise bits indexed by  $v$  (i.e.,  $\langle v, e_S \rangle$ ) is 1, which happens with probability at most  $1/2 - 1/2(1 - 2 \cdot \mu)^q$ . When  $b = 0$  the decryption errs when the equation is satisfied which happens with probability  $1/2$ . Hence, when  $b$  is a random bit the error probability is  $\frac{2 - (1 - 2\mu)^q}{4}$  and the claim follows.

**Privacy.** Let  $G \in_{\mathcal{R}} \mathcal{F}$  be the real public-key and  $\hat{G} \in_{\mathcal{R}} \mathcal{G}$  be a “fake” public-key. By the DUE assumption the pair  $(G, \text{Enc}_G(0))$  is  $\epsilon$ -indistinguishable from the pair  $(\hat{G}, \text{Enc}_{\hat{G}}(0))$ . Moreover, by the DSPN assumption, the latter distribution is  $\epsilon$ -indistinguishable from the pair  $(\hat{G}, \text{Enc}_{\hat{G}}(1))$ . Finally, again by the DUE assumption,  $(\hat{G}, \text{Enc}_{\hat{G}}(1))$  is  $\epsilon$ -indistinguishable from  $(G, \text{Enc}_G(1))$ . It follows that  $(G, \text{Enc}_G(0))$  is  $3\epsilon$ -indistinguishable from  $(G, \text{Enc}_G(1))$ .

Holenstein and Renner show that, when  $(1 - 2\alpha)^2 > \beta$ , one can transform an  $(\alpha, \beta)$ -secure public-key cryptosystem to a semantically secure public-key cryptosystem [HR05, Thm. 6]. The proof is completed by noting that our parameters satisfy this condition.  $\square$

**Remark 3.3** (*Oblivious Transfer from DSPN and DUE*). Oblivious Transfer [RAB81] (OT) is a useful cryptographic primitive which allows a sender to send a message to a receiver with probability  $1/2$ , while the sender remains oblivious as to whether or not the receiver received the message. The existence of OT implies a secure protocol for any multi-party functionality [GMW87], but is not known to be implied by general public-key encryption scheme. However, [EGL82] shows how to construct OT from a public-key encryption scheme in which one can generate a “bad public key” that looks indistinguishable from the valid public key, but does not allow the generating party to distinguish between the encryption of 0 and the encryption of 1. Interestingly, our scheme satisfies this additional property and therefore it implies the existence of an OT-protocol.

### 3.3 Variants

In the following we present several variants of the basic construction. First, we examine variants of our assumptions that still suffice to prove the security of the scheme. Then, we suggest an alternative construction in which the noisy parity function is replaced with a non-linear predicate.

#### 3.3.1 Alternative assumptions: using expander graphs

We feel that the core property of random graphs key to the validity of both assumptions is strong expansion.<sup>8</sup> Say that a bipartite graph  $G = (V_{\text{Out}}, V_{\text{In}}, E)$  is an  $(k, \alpha)$  expander if for every  $S \subseteq V_{\text{Out}}$  with  $|S| \leq k$ ,  $|\Gamma_G(S)| \geq \alpha k$ . Then we can define the following assumptions:

<sup>8</sup>Similar conjectures were made with regards to a variant of DSPN, see [ALE03, Remark 1] and [AIK06].

- **DSPN'**: For every fixed  $(m, n, d)$ -graph  $G$  that is an  $(k, 0.9d)$ -expander, the distributions  $G_{\oplus\mu}(U_n)$  and  $U_m$  are  $\epsilon$ -indistinguishable, where  $k = \omega(\log n)$  is some additional parameter (e.g.,  $k = \sqrt{n}$ ).<sup>9</sup>
- **DUE'**: There exist two samplable distributions  $\mathcal{G}'$  and  $\mathcal{F}'$  on  $(m, n, d)$ -graphs such that: (1)  $\mathcal{G}'$  is a  $(k, 0.9d)$ -expander with probability  $1 - 1/n$ ; (2) If  $G = (V_{\text{Out}}, V_{\text{In}}, E)$  is chosen from  $\mathcal{F}'$ , then  $G$  always has a set  $S \subseteq V_L$  of size  $q$  such that  $|\Gamma_G(S)| < |S|$ . Moreover,  $G$  can be sampled efficiently together with the set  $S$ . (3)  $\mathcal{G}'$  and  $\mathcal{F}'$  are  $\epsilon$ -indistinguishable.

Clearly, **DSPN'** is stronger than **DSPN** (i.e.,  $\text{DSPN}' \Rightarrow \text{DSPN}$ ) while **DUE'** is weaker than **DUE** (i.e.,  $\text{DUE} \Rightarrow \text{DUE}'$ ). Our scheme can be based on either  $(\text{DSPN} \wedge \text{DUE})$  or on  $(\text{DSPN}' \wedge \text{DUE}')$ . We also mention that some of our evidence for **DSPN** actually holds for the stronger version **DSPN'** (see Section 4.1). Also, some of our evidence for **DUE** only holds for the weaker **DUE'** (see Section 5).

### 3.3.2 An alternative construction: using non-linear predicate

We now describe a variant of our scheme, in which the **DSPN** assumption is replaced by the existence of certain pseudorandom generators in  $\mathbf{NC}^0$ .<sup>10</sup> We replace the function  $G_{\oplus\mu}$  with the function  $G_f$  in which every output bit, instead of being the noisy XOR of the inputs corresponding to its neighbors in  $G$ , is obtained by applying some *deterministic* but *non-linear* function  $f : \{0, 1\}^d \rightarrow \{0, 1\}$  on these inputs. To prove the security of the new scheme we need to assume a non-linear version of **DSPN** namely,

**Assumption DSF**( $m, d, \epsilon$ ) (*Decisional Sparse Function*). There exists a function  $f : \{0, 1\}^d \rightarrow \{0, 1\}$  for which the distribution  $(G, G_f(U_n))$  is  $\epsilon$ -indistinguishable from the distribution  $(G, U_m)$  where  $G \in_{\mathcal{R}} \mathcal{G}_{m,n,d}$ .

In Appendix A we identify simple combinatorial properties that makes a function  $f$  a plausible candidate for the **DSF** assumption. Specifically, we suggest to use the majority of three parities on  $d/3$  bits each. Some of our evidence for the **DSPN** assumption hold for this (and other) instantiations of the **DSF** assumption as well. In particular, the following theorem is proved in Appendix A:

**Informal Theorem 5.** Assumption **DSF** instantiated with the “majority of three parities” function cannot be refuted using distinguishers that linear tests, or myopic distinguishers reading  $\sqrt{n}$  output bits. Moreover, under a conjecture of Linial and Nisan [LN90], **DSF** cannot be refuted using distinguishers that are  $\mathbf{AC}^0$  circuits.

**DSF vs. DSPN.** The main drawback of this non-linear variant is that we can no longer efficiently decrypt by checking that a projection of the output falls in some linear subspace. Rather, we check whether a string  $y$  was obtained by  $G_f(x)$  for  $G$  with a  $q$ -sized shrinking set  $S$  by enumerating over all  $2^{q-1}$  possibilities for the restriction of  $x$  to  $\Gamma(S)$  and checking if  $y$  is consistent with one of these. (Since there are only  $2^{q-1}$  possibilities, a random string will be inconsistent with probability at least  $1/2$ .) However, since we need decryption to be efficient, this requires setting  $q = O(\log n)$ , which in turns makes the system breakable in time  $n^q = n^{O(\log n)}$ . On the other hand, when using noisy parity we can decrypt efficiently for any value of  $q$ . This allows a much wider range of tradeoffs

<sup>9</sup>In fact, it seems reasonable that the assumption holds with negligible  $\epsilon$ .

<sup>10</sup>This variant was announced in the preprint [BW08], subsumed by the current work.

between  $q$  and  $\mu$ , and potentially results in subexponential security. Another important advantage is that for noisy parity we have a reduction between pseudorandomness and one-wayness. Still we believe that the non-linear alternative breaks further away from existing public-key systems, and it may be possible find a clever variant with efficient decoding and subexponential security.

**DSF and  $\mathbf{NC}^0$  cryptography.** The DSF assumption implies the existence of a pseudorandom generator of large (superlinear) stretch in  $\mathbf{NC}^0$ . The existence of such generator was studied recently in a sequence of works [CM01, MST03, AIK04, MST03]. Under widely believed assumptions, Applebaum et al [AIK04] show that there exists a pseudorandom generator mapping  $n$  bits to  $n + \sqrt{n}$  bits that can be computed in  $\mathbf{NC}^0$ . A construction that achieves linear stretch (e.g.,  $n \mapsto 2n$ ) based on a specific assumption (similar but weaker than DSPN) was given in [AIK06]. Finally, a candidate construction with polynomial stretch was given by [MST03], who also showed that a generator where each output bit depends on  $d$  input bits cannot have output of length longer than  $\tilde{O}(n^{d/2})$ . Our DSF assumption can be phrased as the assumption that a polynomial (say  $n \mapsto n^{1+\delta}$ ) stretch<sup>11</sup>  $\mathbf{NC}^0$  generator exists, and in fact it can be defined by mapping the inputs to the outputs via a random graph. This assumption is related to the [MST03] construction as well as to Goldreich’s [Gol00] candidate for a one-way function, which has the same nature but no output stretch.

## 4 On the validity of Assumption DSPN

In this section we collect some evidence for the DSPN assumption. Our main result here is a reduction from a search version of sparse parity with noise to the decision version of this problem (see Section 4.2). This allows us to base the DSPN assumption, which is an indistinguishability assumption, on a related “one-wayness” assumption. We also prove that the DSPN generator (or even the DSPN’ generator) unconditionally resists some simple statistical tests, such as sparse tests, linear tests and low degree polynomials, and is likely to pass others, e.g.,  $\mathbf{AC}^0$  circuits (see Section 4.1).

### 4.1 Unconditional resistance to simple statistical tests

We now study to what extent our generator  $G_{\oplus\mu}$  passes at least some very simple statistical tests (namely, those for which unconditionally secure pseudorandom generators are known to exist).

**Definition 4.1.** We say that a distribution  $D$  over  $\{0, 1\}^n$   $\epsilon$ -fools a class  $\mathcal{F}$  of boolean functions over  $\{0, 1\}^n$  if

$$\max_{f \in \mathcal{F}} |\Pr[f(D) = 1] - \Pr[f(U_n) = 1]| \leq \epsilon.$$

**Theorem 4.2.** *Let  $G$  be an  $(m, n, d)$ -graph that is a  $(k, 0.51d)$  expander. Then,  $G_{\oplus\mu}(U_n)$*

1. *0-fools  $k$ -wise tests. ( $G_{\oplus\mu}(U_n)$  is  $k$ -wise independent.)*
2.  *$\epsilon = 1/2 \cdot (1 - 2\mu)^k$ -fools linear tests. ( $G_{\oplus\mu}(U_n)$  is  $\epsilon$ -biased.)*
3.  *$8 \cdot (1 - 2\mu)^{k/2^{t-1}}$ -fools degree  $t$  polynomials over  $\text{GF}(2)$ .*

<sup>11</sup>The use of logarithmic-size shrinking set forces us to take  $m$  to be super linear in  $n$ .

Hence, if  $1/\mu = n^a$  and  $k = n^{a+b}$  is polynomially larger, we get subexponential hardness of  $\exp(-n^{b/2})$  against polynomials of degree smaller than  $0.49b \log(n)$ . Since the theorem holds for any (fixed) good expander it even supports the DSPN' assumption (which seems stronger than DSPN).

*Proof.* The first two items follow from the analysis of [MST03]. We sketch them here for completeness. We break the distribution  $G_{\oplus\mu}(U_n)$  into two independent parts:  $Y$  and  $E$  such that  $G_{\oplus\mu}(U_n) = Y + E$ . This is done by letting  $Y = G \cdot U_n$ , (here  $G$  stands for the adjacency matrix of the graph), and  $E$  be a random  $m$ -bit error vector whose entries take the value 1 with probability  $\mu$  independently of each other.

We prove (1) by showing that for every subset  $S \subseteq [m]$  with  $|S| \leq k$ ,

$$\Pr \left[ \sum_{i \in S} Y_i = 1 \right] = 1/2 \tag{1}$$

Indeed, by a simple counting argument, there exists  $i \in S$  with a unique neighbor  $j \in \Gamma_G(i) \setminus \Gamma_G(S \setminus \{i\})$ . Therefore, if we fix all inputs in  $\Gamma_G(S \setminus \{i\})$  (thus fixing  $Y_u$  for all  $u \in S$  with  $u \neq i$ ), then, the probability over the choice of the input  $j$  that  $Y_i = 1$  is equal to  $1/2$ , establishing (1).

For (2), it suffices to prove that for every subset  $S \subseteq [m]$  with  $|S| \geq k$ ,

$$\Pr \left[ \sum_{i \in S} E_i = 1 \right] = 1/2 - 1/2 \cdot (1 - 2\mu)^k.$$

This follows by the fact that the sum of  $t$  independent Bernoulli random variables with expectation  $\mu$  is 1 with probability  $1/2 - 1/2(1 - 2 \cdot \mu)^t$ .

We proceed with (3). The following claim shows that the distribution  $G_{\oplus\mu}(U_n)$  can be written as the sum of  $t$  independent copies of  $G_{\oplus\alpha}(U_n)$  for a related  $\alpha$ .

**Claim 4.3.** *Let  $\alpha = 1/2 - 1/2(1 - 2\mu)^{1/t}$ . Then,*

$$G_{\oplus\mu}(U_n) \equiv \sum_{i=1}^t G_{\oplus\alpha}(U_n^{(i)}),$$

where the  $U_n^{(i)}$ 's are independent copies of  $U_n$ .

*Proof of claim.* For every  $x^{(1)}, \dots, x^{(t)} \in \{0, 1\}^n$  we have

$$\sum_{i=1}^t G_{\oplus\alpha}(x^{(i)}) \equiv \sum_{i=1}^t Gx^{(i)} + E^{(i)} \equiv G \cdot \left( \sum_{i=1}^t x^{(i)} \right) + \sum_{i=1}^t E^{(i)} \equiv G_{\oplus\mu}(x),$$

where  $E^{(1)}, \dots, E^{(t)}$  are  $t$  independent error vectors of error-rate  $\alpha$  and  $x = \sum_{i=1}^t x^{(i)}$ . The last equality follows by noting that the entries of the vector  $\sum_i E^{(i)}$  are independently distributed with expectation  $1/2 - 1/2(1 - 2 \cdot \alpha)^t = \mu$ . The claim follows by choosing  $x^{(1)}, \dots, x^{(t)}$  uniformly and independently.  $\square$

Hence, by item 2,  $G_{\oplus\mu}(U_n)$  is the sum of  $t$  independent samples from  $\epsilon$ -biased distribution, where  $\epsilon = 1/2 \cdot (1 - 2\alpha)^k = 1/2 \cdot (1 - 2\mu)^{k/t}$ . Viola [Vio08] recently proved that in this case, the distribution  $G_{\oplus\mu}(U_n)$  also  $8 \cdot (2\epsilon)^{1/2^{t-1}}$ -fools degree  $t$  polynomials<sup>12</sup>, which completes the proof.  $\square$

**Fooling  $\mathbf{AC}^0$  circuits.** We were not able to show that our generator fools such constant depth (i.e.,  $\mathbf{AC}^0$ ) circuits. But the limited independence above is far higher than what would, at least conjecturally, imply such a pseudorandomness result. First recall that according to a conjecture of Linial and Nisan [LN90],  $\log^{\omega(1)} n$  independence implies that  $G_f(U_n)$  fools every  $\mathbf{AC}^0$  circuit. This conjecture has been open for a while, but was recently proven for the first nontrivial case of depth-2 circuits (i.e., DNF formulae) by Bazzi [Baz07]. It would be interesting to prove the weaker version of the general conjecture that will suffice for us, namely that  $n^\epsilon$ -independence fools  $\mathbf{AC}^0$  circuits. We note that the pseudorandom generator for  $\mathbf{AC}^0$  of [AW85] is a *very specific*  $n^\epsilon$ -independent distribution, and perhaps their argument can be generalized to prove this weaker conjecture.

## 4.2 From Search to Decision

Consider the SPN problem in which the goal of the adversary is to find the seed  $x$  given the pair  $(G, G_{\oplus\mu}(x))$ , where  $G \in_{\mathcal{R}} \mathcal{G}_{m,n,d}$  and  $x \in_{\mathcal{R}} U_n$ . This problem seems to be harder than its decisional variant, and therefore it seems plausible that it cannot be solved efficiently with overwhelming success probability. In other words, we assume that  $G_{\oplus\mu}$  is weakly one-way, formally,

**Assumption SPN( $m, d, \mu$ ).** There exists a constant  $c > 0$  such that for every polynomial size circuit family  $\{A_n\}$  and all sufficiently large  $n$ 's, the success probability of  $A_n$  in finding<sup>13</sup>  $x$  given  $(G, G_{\oplus\mu}(x))$  where  $G \in_{\mathcal{R}} \mathcal{G}_{m,n,d}$  and  $x \in_{\mathcal{R}} U_n$  is at most  $1 - n^{-c}$ .

We show that the DSPN assumption reduces to the SPN assumption for related parameters. Namely,

**Theorem 4.4** (Reducing search SPN to decisional SPN). *Let  $d(n) < \sqrt{n}/4$  be sparsity parameter, and  $0 < \mu(n) < 1/2$  be a noise parameter which is bounded away from  $1/2$ . Let  $n < m(n) < \text{poly}(n)$  be a length parameter, and  $\epsilon(n) > 1/\text{poly}(n)$  be an indistinguishability parameter. Then assumption DSPN( $m, d, \mu, \epsilon$ ) is implied by Assumption SPN( $m', d', \mu'$ ), where  $m' = m^3 \log^2 n / \epsilon^2$ ,  $d' = d/2$ , and  $\mu' = \frac{(1 - \sqrt{1 - 2\mu})}{2} \approx \frac{\mu}{2}$ .*

In particular, in order to get  $\epsilon$ -pseudorandom generator of output length  $n^{1.1}$  (for some small fixed constant  $\epsilon$ ) it suffices to assume one-wayness of  $(G, G_{\oplus\mu}(x))$  where  $G$  has output length of  $m = n^{3.4}$  and essentially the same degree  $d$  and noise rate  $\mu$ .

**Comparison to the dense case.** A similar reduction (from search to decision) was presented in [BFKL94] (see also [REG05, KS06, AIK07]) for the non-sparse case, i.e., when the adjacency matrix of  $G$  is chosen uniformly from  $U_{m \times n}$ . However their reduction relies heavily on the fact that the columns of  $G$  are drawn from the uniform distribution, and so developing a reduction for the sparse

<sup>12</sup>The original bound is stated in terms of *character distance* and is translated here to *statistical distance* terms. The difference between these two notions, over  $\text{GF}(2)$ , is just a factor of two [BV07, Claim 33].

<sup>13</sup>If  $d$  is even, it is information-theoretic impossible to distinguish  $x$  from its complement, and so in this case we consider both answers as legitimate.

case requires new techniques.<sup>14</sup> We believe that the current reduction might turn useful in other contexts as well.

**Proof idea.** Our goal is to convert a distinguisher, which breaks the DSPN assumption to an inverting algorithm which breaks the SPN assumption.

The distinguisher of DSPN we have available can be converted (by standard techniques) into a predictor, which given the noisy value of some  $k$  (with  $k < m$ ) random equations in  $n$  unknowns with  $d$  variables per equations, predicts the value of another such random equation with non negligible advantage.

Inverting SPN calls for solving a system of  $m'$  noisy equations in  $n$  variables with a random set of  $d/2$  variables per equation. This will be done by using the above predictor to generate  $10n$  equations in only *two* random variables per equations and negligible ( $\ll 1/n$ ) noise. Clearly, this will suffice to (uniquely) recover the values all variables.

Let us explain how to generate one such equation. Take  $k$  random *disjoint* pairs of equations from the collection of  $m'$  equations, and another random pair having one element in common. Finally pick at random variables, say  $x_i, x_j$ . These will determine the input to the predictor as follows. Each of the  $k$  pairs generates one noisy equation in  $d$  variables, whose value we know (note that the noise  $\mu'$  was picked so that we get the appropriate distributions to the predictor). The last pair generates a random equation on  $d - 2$  variables (whose noisy value we know), but we only feed the predictor the  $d$ -set including these variables with  $x_i, x_j$ . From the answer of the predictor, our knowledge and linearity we get a noisy value for  $x_i + x_j$ .

Repeating this experiment sufficiently many times generates enough equations, and having logarithmically many values for each pair reduces the noise sufficiently. The full proof of Theorem 4.4 is deferred to Section 4.3.

### 4.3 Proof of Thm. 4.4

**Notation:** For  $x \in \{0, 1\}^n$ , we let  $\mathcal{P}_{d,\mu}(x)$  denote the distribution  $(S, y)$  where  $S$  is a random  $d$ -sized subset of  $[n]$  (i.e.,  $S$  is chosen by selecting  $d$  distinct elements from  $[n]$  uniformly and independently<sup>15</sup>), and  $y = \sum_{i \in S} x_i + e \pmod{2}$  where  $e$  is an “error” coin which is 1 w.p  $\mu$ . We define  $\mathcal{R}_{d,n}$  similarly to  $\mathcal{P}_{d,\mu}(x)$  except that  $y$  is uniformly chosen. We write  $\mathcal{P}_{d,\mu}^m(x)$  and  $\mathcal{R}_{d,n}^m$  to denote  $m$  independent samples from  $\mathcal{P}_{d,\mu}(x)$  and  $\mathcal{R}_{d,n}$  respectively. Note that the distribution  $\mathcal{P}_{d,\mu}^m(x)$  is just a representation of the distribution  $(G, G_{\oplus\mu}(x))$  where  $G \in_{\mathbb{R}} \mathcal{G}_{m,n,d}$ , and, similarly, the distribution  $\mathcal{R}_{d,n}^m$  represents the distribution  $(\mathcal{G}_{m,n,d}, U_m)$ .

#### 4.3.1 Sampling lemmas

The following lemmas are used to convert samples with sparsity  $d$  and noise rate  $\mu$  to samples with larger sparsity and larger noise rate without knowing the vector  $x$ . That is, we take samples from  $\mathcal{P}_{d,\mu}(x)$  and output samples from  $\mathcal{P}_{2d-2,\mu^*}(x)$  and from  $\mathcal{P}_{2d,\mu^*}(x)$ , where here and in the rest of this section we let

$$\mu^* = 2 \cdot \mu \cdot (1 - \mu). \tag{2}$$

<sup>14</sup>Indeed, the works of [ALE03, AIK06] relied on *decisional* assumptions similar to DSPN rather than on the intractability of a search problem.

<sup>15</sup>One may consider slightly different distributions, e.g., by letting  $S$  be a random  $d$ -sized *multiset*. This difference is not crucial as our results extend to this (and other) variants of the distribution.

**Lemma 4.5.** *There exists an efficient algorithm  $C$  such that for every  $x \in \{0, 1\}^n$ , and  $d < \sqrt{n}/4$ , the random variable  $C(\mathcal{P}_{d,\mu}^{4m+n}(x))$  is  $\exp(-m/4)$ -close (in statistical distance) to the random variable  $\mathcal{P}_{2d-2,\mu^*}^m(x)$ , where  $\mu^*$  is defined as in Eq. 2.*

*Proof.* Our algorithm will output a special failure symbol with probability at most  $\exp(-m/4)$ , and, conditioned on not failing, will perfectly emulate the distribution  $\mathcal{P}_{2d-2,\mu^*}^m(x)$ . Let  $(S_i, y_i)_{i=1}^{4m+n}$  denote  $C$ 's input and  $(S'_i, y'_i)_{i=1}^m$  denote  $C$ 's output conditioning on not failing. The idea is to find  $m$  disjoint pairs of input sets  $S_{i_1}$  and  $S_{i_2}$  such that  $S_{i_1}$  and  $S_{i_2}$  have a single common entry  $j$ , and combine them to a new output set  $S'_i$  that contains the entries  $(S_{i_1} \setminus \{j\}) \cup (S_{i_2} \setminus \{j\})$ . The corresponding output bit  $y'_i$  will be the sum (modulo 2) of  $y_{i_1}$  and  $y_{i_2}$ . It is not hard to see that the conditional distribution  $[y'_i | S'_i]$  is distributed correctly. Indeed,

$$y'_i = y_{i_1} + y_{i_2} = \left( \sum_{k \in (S_{i_1} \setminus \{j\})} x_k + e_{i_1} \right) + \left( \sum_{k \in (S_{i_2} \setminus \{j\})} x_k + e_{i_2} \right) = \left( \sum_{k \in S'_i} x_k \right) + (e_{i_1} + e_{i_2}),$$

where  $e_{i_1}$  and  $e_{i_2}$  are independent noise bits of rate  $\mu$  and therefore their sum is a noise bit of rate  $2\mu(1 - \mu)$ .

It is left to explain how to match the input sets. We partition the input sets  $S_i$  into  $n$  buckets  $B_j$  indexed by  $\{1, \dots, n\}$ . For each set  $S$ , we randomly choose a representative index  $j \in S$  (uniformly from all  $d$  entries of  $S$ ), and throw  $S$  to the corresponding bucket  $B_j$ . Then, we partition the sets in each bucket to pairs arbitrarily. Clearly, each pair shares a common entry. If a pair shares more than one common entry, we call it *bad* and throw it away. We let  $S'_i$  be the union of the  $i$ -th good pair. If we have less than  $m$  good pairs we output a failure symbol  $\perp$ . Since for every pair the matching does not depend on the other (non-representative) entries of the pair, the resulting combined sets  $(S'_i)_i$  are uniformly and independently distributed over all  $2d - 2$ -size sets. For the same reason, the probability that a pair is bad is at most  $d^2/(n - d) < 1/4$ . Finally, for each bucket at most a single set does not participate in the matching (in case the number of sets in the pile is odd), and so we try to merge at least  $(4m + n - n)/2 = 2m$  pairs. By a Chernoff bound, the probability that more than  $m$  of them will be bad is at most  $\exp(-4m \cdot (1/4)^2) \leq \exp(-m/4)$ , which completes the proof.  $\square$

**Lemma 4.6.** *There exists an efficient algorithm  $C$  such that for every  $x \in \{0, 1\}^n$ , and  $d < \sqrt{n}/4$ , the random variable  $C(\mathcal{P}_{d,\mu}^{4m}(x))$  is  $\exp(-m/4)$ -close (in statistical distance) to the random variable  $\mathcal{P}_{2d,\mu^*}^m(x)$ , where  $\mu^*$  is defined as in Eq. 2.*

*Proof.* Again,  $C$  will output a special failure symbol  $\perp$  with probability at most  $\exp(-m/4)$ , and, conditioned on not failing, will perfectly emulate the distribution  $\mathcal{P}_{2d,\mu^*}^m(x)$ . Let  $(S_i, y_i)_{i=1}^{4m}$  denote  $C$ 's input and  $T = (S'_i, y'_i)_{i=1}^m$  denote  $C$ 's output. For all odd  $i \in [4m - 1]$  check whether the sets  $S_i$  and  $S_{i+1}$  are disjoint. If so, call  $i$  good, and add the set  $S' = S_i \cup S_{i+1}$  together with the label  $y' = y_i + y_{i+1}$  to the output list  $T$ . If  $T$  has less than  $m$  entries output  $\perp$ , otherwise output the first  $m$  entries. It is not hard to verify that the entries of  $T$  are distributed independently according to  $\mathcal{P}_{2d,\mu^*}^m(x)$ . Also, the probability for  $i$  to be bad is at most  $d^2/(n - d) < 1/4$ , and hence, by a Chernoff bound, the failure probability is at most  $\exp(-4m \cdot (1/4)^2) \leq \exp(-m/4)$ .  $\square$

### 4.3.2 From prediction to pair-approximation

**Definition 4.7** ( $\epsilon$ -predictor). We say that an algorithm  $A$  predicts  $\mathcal{P}_{d,\mu^*}^{m+1}(U_n)$  with advantage  $\epsilon$  if

$$\Pr_{x \in_{\mathbb{R}} U_n} [A(\mathcal{P}_{d,\mu^*}^m(x), T) = \sum_{i \in T} x_i] \geq 1/2 + \epsilon,$$

where  $T \subset [n]$  is a random subset of size  $d$ .

We show that a predictor  $A$  for  $\mathcal{P}_{2d,\mu^*}^{m+1}(x)$  with advantage  $\epsilon$ , yields an algorithm  $B$  which given  $\mathcal{P}_{d,\mu}^{4m+2n}(x)$  and indices  $u, v$  guesses the value of  $x_u + x_v$  with essentially the same advantage. We refer to such an algorithm as pair-approximator.

**Lemma 4.8.** *Let  $d < \sqrt{n}/4$  be a positive integer, and let  $m = m(n) > n$ . Suppose that we have an efficient predictor  $A$  with advantage  $\epsilon$  for  $\mathcal{P}_{2d,\mu^*}^{m+1}(U_n)$ . Then, there exists an efficient pair-approximator  $B$  such that for every  $u, v \in [n]$*

$$\Pr[B(\mathcal{P}_{d,\mu}^{4m+2n}(x), u, v) = x_u + x_v] \geq 1/2 + \epsilon - 2\mu^*\epsilon - 3\exp(-n/4),$$

where  $x \in_{\mathbb{R}} U_n$  and  $\mu^*$  is defined as in the previous lemma.

*Proof.* Without loss of generality, assume that  $u \neq v$  as otherwise  $B$  outputs 0. We construct  $B$  as follows:

1. Input:  $(S_i, y_i)_{i=1}^{4m+2n}$  and  $u, v$ .
2. Use Lemma 4.6 to transform  $(S_i, y_i)_{i=1}^{4m}$  to  $m$  independent samples  $(S'_i, y'_i)_{i=1}^m$  from  $\mathcal{P}_{2d,\mu^*}(x)$  and, similarly, use Lemma 4.5 to transform  $(S_i, y_i)_{i=4m+1}^{4m+2n}$  to  $n/4$  independent samples  $(T_i, z_{m+n/4})_{i=1}^{n/4}$  from  $\mathcal{P}_{2d-2,\mu^*}(x)$ .
3. Search among all the  $\{T_i\}_i$  for the first set  $T_j$  for which  $T_j \cap \{u, v\}$  is empty. If such a set does not exist output a failure symbol  $\perp$  and terminate.
4. Choose a random permutation  $\pi : [n] \rightarrow [n]$  and invoke  $A$  on  $(\pi(S'_i), y'_i)_{i=1}^m$  and  $T = \pi(T_j \cup \{u, v\})$ , where  $\pi(S) = \{\pi(k) | k \in S\}$ . Output  $b - z_j$ , where  $b$  is  $A$ 's output.

**Analysis:** Suppose that the input to  $B$  is distributed properly, that is  $(S_i, y_i)_{i=1}^{4m+2n} \in_{\mathbb{R}} \mathcal{P}_{d,\mu}^{4m+2n}(x)$ . Then, the outputs of Step 2  $(S'_i, y'_i)_{i=1}^m$  and  $(T_i, z_{m+n/4})_{i=1}^{n/4}$  are distributed according to  $\mathcal{P}_{2d,\mu^*}^m(x)$  and  $\mathcal{P}_{2d-2,\mu^*}^{n/4}(x)$  up to exponentially-small error deviation of  $2\exp(-n/4)$ . Suppose that Step 3 also succeeds (which happens with probability at least  $1 - (2d/n - d)^{n/4} > 1 - \exp(-n/4)$ ). Then  $(\pi(S'_i), y'_i)_{i=1}^m$  is distributed according to  $\mathcal{P}_{2d,\mu^*}^m(\hat{x})$  where  $\hat{x}$  is obtained by permuting  $x$  according to  $\pi$ , i.e.,  $x_{\pi(i)} = \hat{x}_i$ . Also, the set  $T$  is a random  $2d$ -sized set which is statistically independent of  $(\pi(S'_i), y'_i)_{i=1}^m$ . Therefore the predictor  $A$  is guaranteed to output  $\sum_{i \in T} \hat{x}_i = (\sum_{i \in T_j} x_i) + x_u + x_v$  with probability at least  $1/2 + \epsilon$ . On the other hand,  $z_j$  equals to  $\sum_{i \in T_j} x_i$  with probability  $1 - \mu^*$ . Furthermore, the latter event is independent of the former, hence conditioning on full success in Steps 2 and 3, we get the correct value with probability  $1/2 + \epsilon - 2\mu^*\epsilon$ . A union bound over the failure probabilities and error deviations of Steps 2 and 3 yield the desired bound.  $\square$

When  $\mu$  is bounded away from  $1/2$  and  $\epsilon$  is noticeable, the success probability is  $1/2 + \Omega(\epsilon)$ .

### 4.3.3 Amplification

We would like to amplify the success probability by applying the above procedure to many independent copies of  $\mathcal{P}_{d,\mu}^{4m+2n}(x)$ . However, these samples are statistically dependent since the same  $x$  is used. Instead, we use the following claim to randomize the  $x$ 's.

**Claim 4.9.** *There exists an efficient algorithm  $R$  such that for every  $x \in \{0,1\}^n$  we have*

$$R(\mathcal{P}_{d,\mu}^m(x)) \equiv (\mathcal{P}_{d,\mu}^m(x'), r),$$

where  $x' \in_{\mathbb{R}} U_n$  and  $r$  is a “shift” vector which equals to  $x' - x$ .

*Proof.* On input  $(S_i, y_i)_{i=1}^m$ , the algorithm  $R$  chooses  $r \in_{\mathbb{R}} U_n$  and outputs  $((S_i, y'_i)_{i=1}^m, r)$  where  $y'_i = y_i + \sum_{j \in S_i} r_j$ .  $\square$

We can now amplify the success probability of a pair-approximator.

**Claim 4.10.** *Suppose that we have an efficient  $1/2 + \epsilon$  pair-approximator  $A$  for  $\mathcal{P}_{d,\mu}(U_n)$  which uses  $\ell$  samples to predict the sum  $x_u + x_v$  for any given pair of indices  $(u, v)$ . Then, there exists  $1 - \exp(-2t\epsilon^2)$  pair-approximator for  $\mathcal{P}_{d,\mu}(U_n)$  which uses  $t \cdot \ell$  samples to predict  $x_u + x_v$ .*

*Proof.* We partition the input  $\mathcal{P}_{d,\mu}^{t\ell}(x)$  to  $t$  independent samples blocks each of size  $\ell$ , and randomize them via Claim 4.9 by using independent random shift vector  $r^1, \dots, r^t$ . As a result we get  $t$  independent samples  $Q_1, \dots, Q_t$  from  $\mathcal{P}_{d,\mu}^{\ell}(U_n)$ . To approximate the sum of the  $u$ -th and  $v$ -th bit take the majority of  $(A((u, v), Q_1) + r_u^1 + r_v^1), \dots, A((u, v), Q_t) + r_u^t + r_v^t)$ . By a Chernoff bound the error probability is at most  $\exp(-2t\epsilon^2)$ .  $\square$

### 4.3.4 From pair-approximation to inversion

The following claim turns a good pair-approximator into an inverter.

**Claim 4.11.** *Suppose that we have an efficient  $1 - \epsilon$  pair-approximator  $A$  for  $\mathcal{P}_{d,\mu}(U_n)$  which uses  $\ell$  samples to predict the sum  $x_u + x_v$  for any given pair of indices  $(u, v)$ . Then, there exists an inverter  $B$  for  $\mathcal{P}_{d,\mu}^{\ell}(U_n)$  with success probability  $1 - n \cdot \epsilon - \exp(-2\ell(1/2 - \mu)^2)$ .*

In particular, when  $\epsilon$  is negligible,  $\ell$  is super-logarithmic and  $\mu$  is bounded away from  $1/2$  the failure probability is negligible.

*Proof.* Let  $(S_i, y_i)_{i=1}^{\ell}$  be a sample from  $\mathcal{P}_{d,\mu}^{\ell}(U_n)$ . Invoke the approximator  $A$  on  $(Q, (1, i))$  for all  $2 \leq i \leq n$ . This gives a set of  $n - 1$  linear equations of the form  $x_1 + x_i = b_i$  which, by union bound, are all correct with probability  $1 - n\epsilon$ . We solve this linear system and get two solutions  $x$  and its complement. If  $d$  is even, we are done (see Footnote13). Otherwise, output the solution which agrees with more than half of the equations  $\sum_{j \in S_i} x_j = y_i$  where  $i = 1, \dots, \ell$ . Since  $d$  is odd, each equation is satisfied by either  $x$  or its complement. Hence, we err only when at least half of our samples are noisy, which happens with probability  $\exp(-2\ell(1/2 - \mu)^2)$ . A union bound gives the desired bound.  $\square$

### 4.3.5 Concluding the proof

We can now easily base unpredictability on the SPN assumption.

**Theorem 4.12.** *Let  $d < \sqrt{n}/4$  be an integer, and  $0 < \mu < 1/2$  be a noise parameter which is bounded away from  $1/2$ . Let  $n < m < \text{poly}(n)$  be a length parameter and  $\epsilon > 1/\text{poly}(n)$  be an indistinguishability parameter.*

- **Assumption:** (Search is slightly hard)  $\text{SPN}(m \log^2 n/\epsilon^2, d, \mu)$  holds.
- **Implication:** (Prediction is hard) For every efficient predictor  $A$  and all sufficiently large  $n$ 's

$$\Pr_{x \in {}_R U_n} [A(\mathcal{P}_{2d, \mu^*}^{m+1}(x), T) = \sum_{i \in T} x_i] < 1/2 + \epsilon,$$

where  $T \subset [n]$  is a random subset of size  $d$ , and  $\mu^*$  is defined as in Eq. 2.

*Proof.* Assume, towards a contradiction, an efficient predictor for  $\mathcal{P}_{2d, \mu^*}^{m+1}(U_n)$  with advantage  $\epsilon$  for infinitely many  $n$ 's. Then, by Lemma 4.8, we get a pair-approximator for  $\mathcal{P}_{d, \mu}^{6m}(U_n)$  with success probability  $1/2 + \Omega(\epsilon)$ , which can be amplified to  $1 - \text{neg}(n)$  for  $\mathcal{P}_{d, \mu}^{m \log^2 n/\epsilon^2}(U_n)$  (by Claim 4.10). Finally, by Claim 4.11, we get an inverter for  $\mathcal{P}_{d, \mu}^{m \log^2 n/\epsilon^2}(U_n)$  whose error probability is still negligible (for infinitely many  $n$ 's), in contradiction to the  $\text{SPN}(m \log^2 n/\epsilon^2, d, \mu)$  assumption.  $\square$

Yao's Theorem [Yao82] allows us to transform unpredictability to indistinguishability.

**Corollary 4.13** (Restatement of Theorem 4.4). *Let  $d, \mu, m$  and  $\epsilon$  be defined as in Theorem 4.12.*

- **Assumption:** (Search is slightly hard)  $\text{SPN}(m^3 \log^2 n/\epsilon^2, d, \mu)$  holds.
- **Implication:** (Decision is hard) assumption  $\text{DSPN}(m, 2d, \mu^*, \epsilon)$  holds, where  $\mu^*$  is defined as in Eq. 2.

*Proof.* Assume, towards a contradiction, an efficient distinguisher for  $\mathcal{P}_{2d, \mu^*}^m(U_n)$  with advantage  $\epsilon$  for infinitely many  $n$ 's. Then, by Yao's reduction [Yao82] there exists an efficient predictor for  $\mathcal{P}_{2d, \mu^*}^{m-1}(U_n)$  with advantage  $\epsilon/m$ . The corollary now follows from Theorem 4.12.  $\square$

**Remark 4.14.** While Theorem 4.12 is meaningful for  $m = n^{1+\delta}$ , Corollary 4.13 gives nothing from the assumption that  $\text{SPN}(n^3, d, \mu)$  holds. The reason is that Yao's theorem shows that an  $1/2 + \epsilon$  unpredictable sequence of length  $m$  is only  $1/2 + m\epsilon$  pseudorandom, which is meaningless in our case as  $\epsilon > 1/m$ . Instead, we can use Yao's XOR-Lemma to show that the xor of  $k$  independent copies of this distribution is  $\epsilon^k$  unpredictable and then conclude that it is also  $m\epsilon^k$  pseudorandom. The drawback is that the resulting distribution is not an  $\mathcal{P}_{\ell, \rho}^m$  distribution.

## 5 On the validity of Assumption DUE

In this section we provide some evidence for the validity of Assumption DUE. Assumption DUE can be seen as an average-case variant of a combinatorial problem (graph expansion) that is NP-hard to solve exactly in the worst-case. This assumption also implies a fairly strong *hardness of approximation* result for graph expansion (hardness to distinguish between expansion ratio  $(1 -$

$o(1)d$  vs. 1) that is beyond what is known to be implied by  $\mathbf{P} \neq \mathbf{NP}$ . We start by considering how various natural algorithms fare against this problem. We then relate variants of the DUE assumption (specifically close variants of the DUE' assumption mentioned in Section 3.3.1) to the hardness of variants of other natural combinatorial problems such as the planted clique problem and small-set expansion in general (not necessarily bipartite) graphs. The technical details and the proofs of this section are deferred to Section B.

## 5.1 Counting cycles

Key to the validity of the DUE assumption is careful choices of the parameters: the stretch  $c = m/n$ , the degree  $d$  and the size  $q$  of the planted nonexpanding set  $S$ . It is instructive to see how (and which) simple algorithms can break this assumption for the wrong parameters. All attacks use in different ways the fact that the subgraph induced on  $S \cup \Gamma(S)$  is much denser than the rest of the graph.

- Assume  $c = 1$ , namely no stretch. Then it is well known that approximating expansion can be done via the second eigenvalue [AM84, AL08], and hence this value will vary considerably between the distributions  $\mathcal{G}$  and  $\mathcal{F}$ .
- In fact, as long as  $c \ll d$ , we can distinguish between the two cases by just looking at the degree distribution, since  $d$ , the amount added to the degrees in  $\Gamma(S)$  in  $\mathcal{F}$  is larger than the standard deviation of the input degree, which on average is  $cd$ . Using similar considerations one can show that as long as  $c \ll d^2$  we can distinguish between the two distribution by looking at the number of 4 cycles.
- Assume  $d \ll c$  but  $c$  is still small enough to allow  $c^{\log_d q} \ll n$ . Even in this case the density of the planted set can be used, but now with a more sophisticated algorithm, which follows a suggestion of Moses Charikar.<sup>16</sup> Pick  $k$  such that  $10q = d^{2k}$ , and for each vertex in the graph check if it is contained in at least two  $2k$ -cycles. The calculations we do later show that, in expectation, the density of the planted subgraph guarantees that this property will hold for almost every vertex in the planted subgraph, but no vertex outside it! We note that using the “color-coding” algorithm of [AYZ95] this algorithm can be implemented in polynomial time despite the fact that  $k$  is logarithmic in  $n$ .

As demonstrated, subjecting the DUE assumption to standard algorithmic attacks serves as both a “sanity check”, and helps understand the range of parameters in which the assumption might hold. All the algorithms above essentially rely on counting the numbers of small subgraphs in the given graph, with the hope that they “pick up” the planted, denser part. Here we focus on counting short (actually, up to  $n^\epsilon$  so as to examine the possibility of subexponential attacks) cycles in the given graph. We feel that similar results hold for other subgraphs, and that such “local attacks” are not too useful in our chosen parameters.

We let  $\hat{\mathcal{G}}$  and  $\hat{\mathcal{F}}$  denote the variants of  $\mathcal{G}$  and  $\mathcal{F}$  where each edge is chosen with probability  $d/n$  independently (rather than insisting on  $d$ -regularity). Our analysis for cycle counts is done with respect to these distributions. We believe that it can be extended for the distributions  $\mathcal{G}$  and  $\mathcal{F}$  above. Moreover, by dropping vertices with too small a degree,  $\hat{\mathcal{G}}$  and  $\hat{\mathcal{F}}$  can be used for the DSPN assumption and our cryptosystem as well.

---

<sup>16</sup>His original algorithm used a certain quasipolynomially large linear program

**Theorem 5.1.** *For cycles of length  $\ll \log_d n$ , the distributions of the cycle count in  $\hat{\mathcal{G}}$  and  $\hat{\mathcal{F}}$  are  $o(1)$ -close. For cycles of length  $\ll q^{1/4}$ , the two distributions cannot be distinguished by any threshold test (i.e., a test that checks if the count is above or below some threshold between the two expectations).*

See Section B.1 for a more precise statement of the theorem, as well as the proof. To prove Theorem 5.1 we first compute fairly tight bounds on the first few moments of both these random variables. In the case of very short cycles (length  $\ll \log_d n$ ), we are then able to show that both are very close to Poisson random variables with very close expectations. In the case of larger cycles this may not hold, but we are still able to use the moment bounds to rule out threshold tests. We conjecture that threshold tests are actually optimal and thus the result can be extended to show  $o(1)$  statistical distance even in this case.

We remark that by the well known trace formula connecting eigenvalues and cycle counts, the results above suggest that the two distributions will produce extremely close 2nd eigenvalues of  $GG^T$ . However, to make this into a proof one would need to extend the results on distribution and concentration of the second eigenvalue known for random regular graphs to matrices of the form  $GG^T$  where  $G$  is a random regular *unbalanced* graphs.

## 5.2 Reductions from other graph problems

The best evidence for DUE would be to show that it is implied by a much more standard hardness assumption, by reducing some widely-studied computational problem to the task of distinguishing between the distribution  $\mathcal{G}$  and  $\mathcal{F}$  of DUE. This is of course much preferred over just ruling out certain types of algorithms, as is done in Section 5.1. Unfortunately we have no such results, and indeed there seems to be an inherent difficulty in reducing between average-case problems with natural distributions over the inputs, as the image of a reduction typically induces a rather restricted distribution on inputs. However, we are able to show some evidence for the DUE' assumption mentioned in Section 3.3.1 (which also suffices, in conjunction with DSPN', for our cryptosystem). But even for this case the evidence is not as strong as we'd like, and we believe further research is needed. We state the results below informally. More general and precise statements and proofs can be found in Section B.2.

**Theorem 5.2** (See also Theorem B.10). *If it is hard to distinguish given a (not necessarily bipartite)  $d$  regular  $n$  vertex graph  $G$ , between the case that set  $S \subseteq V(G)$  of size  $q$  has  $|\Gamma_G(S)| \leq 2|S|$  and the case that  $G$  is a  $(q', 0.99d)$  (i.e., lossless) expander for  $q' > q$ , then DUE' is true with the same parameters up to constant factors.*

The reduction (presented in Theorem B.10) is very simple. We remark that the hard instances for this problem would be graphs that in both cases are *not* very good expanders for large sets, so that the lack of expansion in the first case would not be detectable using eigenvalues.

**Theorem 5.3** (See Theorems B.12, B.13). *If the planted  $k$ -clique problem is hard in  $G_{n, 2^{-\log^{0.99} n}}$  then it is hard to:*

1. (Shrinking vs. moderate expansion) *Given a bipartite graph  $G = (V_{\text{In}}, V_{\text{Out}}, E)$  distinguish between the case that there is a  $q = \text{poly}(k)$ -sized set  $S \subseteq V_{\text{Out}}$  with  $|\Gamma_G(S)| < |S|$ , and the case where for every set  $S \subseteq V_{\text{Out}}$  with  $|S| < 2^{\log^{0.9} n}$ ,  $|\Gamma_G(S)| > d^{0.9}|S|$  (where  $d$  is the degree).*

2. (Shrinking vs. unique neighbor expansion) *Given a bipartite graph  $G = (V_{\text{In}}, V_{\text{Out}}, E)$  distinguish between the case that there is a  $q = \text{poly}(k)$ -sized set  $S \subseteq V_{\text{Out}}$  with  $|\Gamma_G(S)| < |S|$ , and the case where for every set  $S \subseteq V_{\text{Out}}$  with  $|S| < 2^{\log^{0.9} n}$ ,  $S$  has a unique neighbor: a vertex  $v \in \Gamma(S)$  that has only one neighbor in  $S$ .*

The first part is obtained by a very simple reduction. We start by mapping a graph  $G = (V, E)$  into an  $(|V|, |E|, 2)$ -bipartite graph by having  $V_{\text{In}} = V$  and  $V_{\text{Out}} = E$ , and connecting every vertex in  $V_{\text{Out}}$  to the two vertices that the corresponding edge touches. We then duplicate vertices to translate the expansion parameters to the desired range. The second part starts with the same reduction, but then modifies it by composing it with a lossless disperser in a way motivated by the zig-zag construction. We remark that unique neighbor expansion seems very closely related to lossless expansion, and hence the conclusion of the second part can be viewed as a close variant of DUE'.

**Dense hyper-subgraph problem.** The relation to the planted clique is not so surprising, as we can reformulate DUE as a conjecture on the hardness of a *planted dense hyper-subgraph problem*. We can look at an  $(m, n, d)$ -graph  $G$  as a  $d$ -uniform *hypergraph*  $H$  of  $n$  vertices and  $m$  hyperedges, where the  $i^{\text{th}}$  hyperedge of  $H$  contains the  $d$  neighbors of the  $i^{\text{th}}$  left-vertex of  $G$ . In this formulation, the DUE assumptions is about the hardness of distinguishing hypergraphs that contain a somewhat *dense* sub-hypergraph — a set  $T$  of  $k - 1$  vertices, such that the induced sub-hypergraph on  $T$  has at least  $k$  hyperedges— from graphs where the induced sub-hypergraph of every set of  $k$  vertices (for  $k$  up to roughly  $n^{0.1}$  size or some other super-logarithmic bound) has only about  $k/d$  edges. Thus DUE is essentially equivalent to the problem of distinguishing between a random fairly sparse hypergraph ( $cn$  hyperedges) and a random hypergraph with a planted somewhat *dense* (average degree larger than 1) small subgraph.<sup>17</sup> Indeed, the analog of this program for standard *graphs* (i.e., 2-uniform hypergraphs) has been studied by several works (e.g., [FPK01, KHO04]). This is known as the *densest  $k$ -subgraph* problem— finding a subgraph of  $k$  vertices with highest average degree. The variant of this problem where we ask for a subgraph of high *minimum* degree is fixed-parameter intractable [ASS08].

**Remark 5.4.** Other problems that seem closely related to the DUE problem are (1) *certifying expansion*— show an efficient algorithm that outputs 1 with high probability on a random graph, but never outputs 1 if there exists a  $q$ -sized set  $S$  with  $< |S|$  neighbors and (2) *search unique-neighbor variant* show an algorithm that given every graph with a  $q$ -sized set  $S$  with  $< |S|$  neighbors finds a subset  $S'$  of size  $q'$  (for  $q'$  perhaps somewhat larger than  $q$ ) such that  $S'$  has no unique neighbors.

## 6 Conclusions and challenges

Investigating the validity of these assumptions, and the general question of possibility of less structured public key cryptosystems, give rise to a variety of interesting open question. We list below some concrete related challenges in various areas:

- *Cryptography:* Find a public key encryption scheme based *solely* on (variants of) the DSPN assumption for noise rate that is not too small (i.e.,  $\mu = \Omega(1)$  or even  $\mu = n^{-0.49}$ ).

<sup>17</sup>We say “essentially” because in the planted hypergraph problem the natural distribution would be not to fix the number of edges but rather to include each of the possible  $\binom{n}{d}$  edges with probability  $p = cn/\binom{n}{d}$ . This corresponds to the case where the size of the large side of the bipartite graph is not fixed but rather only concentrated around  $cn$ .

- *Algorithms*: Find an algorithm that breaks our DUE assumption or significantly outperforms subgraph counting based algorithms. Such algorithm could turn out to have interesting application to other currently open computational problems on graphs.
- *Hardness of approximation*: Show that the DUE' problem is at least hard in the worst-case in the sense that it's hard to distinguish between a graph that is a lossless expander and a graph that has a shrinking subset. This could be related to various other problems on vertex expansion whose approximation status is pretty much wide open, even assuming conjectures such as the Unique Games Conjecture [Kho02]. Another, perhaps simpler task, is to establish (variants of) the DUE' conjecture under the assumption that, say, the  $n^{0.1}$ -planted clique problem in  $G_{n,1/2}$  cannot be solved in time  $n^{o(\log n)}$ .
- *Complexity*: Show that breaking our scheme is in  $\mathbf{AM} \cap \mathbf{coAM}$  or give good evidence why this is not the case. Do the same for Alekhnovich's scheme, or other candidate public key cryptosystems proposed in the literature. Note that all schemes that are truly widely-studied are known to be broken in  $\mathbf{AM} \cap \mathbf{coAM}$ , and so it is still a plausible possibility that if  $\mathbf{AM} \cap \mathbf{coAM} = \mathbf{P}$  then there is no secure public key encryption scheme.
- *Concrete models*: Prove unconditionally that the parity with noise generator fools  $\mathbf{AC}^0$  circuits. Find additional natural families of algorithms to study the DUE assumption.

**Acknowledgements.** We thank Noga Alon, Moses Charikar, Thomas Holenstein, Ron Rivest, and Madhu Sudan for useful discussions.

## References

- [ABSS93] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997.
- [AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293. ACM, 1997.
- [AIK04] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in  $\mathbf{NC}^0$ . *SIAM J. Comput.*, 36(4):845–888, 2006. Prelim version FOCS' 04.
- [AIK06] B. Applebaum, Y. Ishai, and E. Kushilevitz. On pseudorandom generators with linear stretch in  $\mathbf{NC}^0$ . In *Proc. of RANDOM*, volume 4110 of *Lecture Notes in Computer Science*, pages 260–271. Springer, 2006.
- [AIK07] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography with constant input locality. In *Advances in Cryptology: Proc. of CRYPTO '07*, pages 92–110, 2007.
- [AKS98] N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. *Random Struct. Algorithms*, 13(3-4):457–466, 1998. Prelim version SODA' 98.
- [ALE03] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307. IEEE Computer Society, 2003.

- [ALo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.
- [AM84] N. Alon and V. D. Milman. Eigenvalues, expanders and superconcentrators (extended abstract). In *FOCS*, pages 320–322. IEEE, 24–26 Oct. 1984.
- [AR04] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. *J. ACM*, 52:749–765, 2005. Prelim version FOCS '04.
- [ASS08] O. Amini, I. Sau, and S. Saurabh. Parameterized complexity of the smallest degree-constrained subgraph problem. In *IWPEC*, volume 5018 of *Lecture Notes in Computer Science*, pages 13–29. Springer, 2008.
- [AW85] M. Ajtai and A. Wigderson. Deterministic simulation of probabilistic constant-depth circuits. *Advances in Computing Research - Randomness and Computation*, 5:199–223, 1989. Prelim version FOCS '85.
- [AYZ95] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995.
- [BAR82] A. D. Barbour. Poisson convergence and random graphs. *Math. Proc. Cambridge Philos. Soc.*, 92(2):349–359, 1982.
- [BAZ07] L. Bazzi. Polylogarithmic independence can fool DNF formulas. In *FOCS*, pages 63–73. IEEE Computer Society, 2007.
- [BFKL94] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology: Proc. of CRYPTO '93*, volume 773 of *LNCS*, pages 278–291, 1994.
- [Bol01] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001.
- [BV07] A. Bogdanov and E. Viola. Pseudorandom bits for polynomials. In *FOCS '07: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 41–51, Washington, DC, USA, 2007. IEEE Computer Society.
- [BW08] B. Barak and A. Wigderson. Public key cryptography from different assumptions. Cryptology ePrint Archive, Report 2008/335, 2008. Contains a preliminary announcement of some of the results in this paper.
- [CFS01] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a mceliece-based digital signature scheme. *Proceedings of AsiaCrypt*, pages 157–174, 2001.
- [CM01] M. Cryan and P. B. Miltersen. On pseudorandom generators in  $NC^0$ . In *Proc. 26th MFCS*, 2001.
- [DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(5):644–654, Nov. 1976.
- [DKRS03] I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating cvp to within almost-polynomial factors is np-hard. *Combinatorica*, 23(2):205–243, 2003.

- [EGL82] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985. Prelim version CRYPTO '82.
- [FPK01] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.
- [GG98] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000. Prelim year STOC '98.
- [GK90] O. Goldreich and E. Kushilevitz. A perfect zero knowledge proof for a problem equivalent to discrete logarithm. In *CRYPTO*, pages 57–70. Springer, Aug. 1990.
- [GKL93] O. Goldreich, H. Krawczyk, and M. Luby. On the existence of pseudorandom generators. *SIAM J. Comput.*, 22(6):1163–1175, 1993. Preliminary version in Proc. 29th FOCS, 1988.
- [GM82] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, Apr. 1984. Prelim version STOC' 82.
- [GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game — A completeness theorem for protocols with honest majority. In *STOC*, pages 218–229. ACM, 1987.
- [Gol00] O. Goldreich. Candidate one-way functions based on expander graphs. Technical Report TR00-090, Electronic Colloquium on Computational Complexity (ECCC), 2000.
- [Gol04] O. Goldreich. *Foundations of Cryptography, Volumes 1 and 2*. Cambridge University Press, 2001,2004.
- [Hås97] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. Prelim version STOC '97.
- [HB01] N. J. Hopper and M. Blum. Secure human identification protocols. In *Advances in Cryptology: Proc. of ASIACRYPT '01*, volume 2248 of *LNCS*, pages 52–66, 2001.
- [HR05] T. Holenstein and R. Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In V. Shoup, editor, *Advances in Cryptology — CRYPTO '05*, Lecture Notes in Computer Science, pages 478–493. Springer-Verlag, Aug. 2005.
- [IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pages 44–61, New York, 1989. ACM.
- [JW05] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology: Proc. of CRYPTO '05*, volume 3621 of *LNCS*, pages 293–308, 2005.
- [Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of 34th STOC*, pages 767–775, New York, 2002. ACM Press.
- [Kho04] S. Khot. Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. In *FOCS*, pages 136–145, 2004.

- [KMR<sup>+</sup>94] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proc. 26th STOC*, pages 273–282, 1994.
- [Kob87] N. Kobitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.
- [KS06] J. Katz and J.-S. Shin. Parallel and concurrent security of the hb and hb+ protocols. In *Advances in Cryptology: Proc. of Eurocrypt 06'*, volume 4004 of *LNCS*, pages 73–87, 2006.
- [Lip97] H. Lipmaa. Cryptology pointers: Public key cryptography: Concrete systems, 1997. Web site, url: <http://www.adastral.ucl.ac.uk/~helger/crypto/link/public/concrete.php>.
- [LN90] N. Linial and N. Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10:349–365, 1990. Prelim version STOC 90.
- [McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *The Deep Space Network Progress Report, DSN PR 42-44, January and February 1978,*, 1978.
- [Mil85] V. S. Miller. Use of elliptic curves in cryptography. In *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 18–22 Aug. 1985.
- [MR08] D. Moshkovitz and R. Raz. Two query PCP with sub-constant error. In *Proc. 49th FOCS*, 2008.
- [MST03] E. Mossel, A. Shpilka, and L. Trevisan. On epsilon-biased generators in  $NC^0$ . *Random Struct. Algorithms*, 29(1):56–81, 2006. Prelim version FOCS '03.
- [RAB79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology, Jan. 1979.
- [RAB81] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.
- [REG04] O. Regev. New lattice-based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, 2004.
- [REG05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93. ACM, 2005.
- [RSA78] R. L. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [RVW00] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *FOCS*, pages 3–13. IEEE, 2000.
- [Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, Oct. 1997.
- [Vio08] E. Viola. The sum of  $d$  small-bias generators fools polynomials of degree  $d$ . In *IEEE Conference on Computational Complexity*, pages 124–127. IEEE Computer Society, 2008.

- [Yao82] A. C. C. Yao. Theory and applications of trapdoor functions. In *FOCS*, pages 80–91. IEEE, 3–5 Nov. 1982.
- [Zhu01] H. Zhu. Survey of computational assumptions used in cryptography broken or not by Shor’s algorithm. Master’s thesis, School of Computer Science McGill University, 2001.

## A Non-linear variant

In this section we discuss the non-linear variant of our encryption scheme. It is obtained by the following theorem:

**Theorem A.1.** *Suppose that Assumptions DSF( $m, d, \epsilon$ ) and DUE( $m, d, q, \epsilon$ ) are true with parameters  $m, d, q, \epsilon$  where  $\epsilon \leq 1/20$  and  $q = O(\log n)$ , then there exists a semantically secure public-key encryption scheme.*

*Proof Sketch:* We use essentially the same cryptosystem as in the proof of Theorem 1.1. The encrypting algorithm can be the same regardless of whether the function is linear or non-linear. For decryption we use exhaustive search on all possible  $2^{q-1}$  preimages of the projection of the ciphertext to the planted coordinate set  $S$ .  $\square$

We now discuss what candidate nonlinear functions can be used to instantiate Assumption DSF

**Definition A.2** ( $\delta$ -resilient functions). Let  $\delta > 0$ . We say that a function  $f : \{0, 1\}^d \rightarrow \{0, 1\}$  is  $\delta$ -resilient if for every subset  $S \subseteq [d]$  with  $|S| < \delta d$  and  $a \in \{0, 1\}^S$ :

1.  $\Pr_{w \in_{\mathbb{R}} W_{S,a}}[f(w) = 1] = 1/2$ , where  $W_{S,a}$  is the distribution over  $w \in \{0, 1\}^d$  chosen such that  $w_S = a$  and for  $i \notin S$ ,  $w_i$  is a random bit.
2. For every  $i \notin S$ ,  $\Pr_{w \in_{\mathbb{R}} W_{S,a}}[f(w) = f(w \oplus e^i)] \in (0, 1)$ , where  $e^i$  is the vector that has 1 in the  $i^{\text{th}}$  coordinate and 0 everywhere else.

For  $\epsilon > 0$ , we say that the function  $f$  is  $(\delta, \epsilon)$ -resilient if in Condition 2 the probability is not just in the interval  $(0, 1)$  but in the interval  $[\epsilon, 1 - \epsilon]$ . Note that this probability is over a sample space of size at most  $2^d$ , and hence every  $\delta$ -resilient function is  $(\delta, 2^{-d})$ -resilient. (Recall that in our application we think of  $d$  as small or even a constant.)

Condition 1 is equivalent to requiring that the function is a *perfect bit-fixing extractor* for bit-fixing sources of entropy more than  $(1 - \delta)d$  (this is also known as a  $\delta d$  *perfect exposure resilient function*).

The parity function satisfies Condition 1, even with  $\delta = 1$ , but does not satisfy Condition 2 no matter how small  $\delta$  is. An example for a  $1/10$ -resilient function is the “majority on three parities” function. This is the function  $f : \{0, 1\}^{3k} \rightarrow \{0, 1\}$  such that on input  $w = x_1, \dots, x_k, y_1, \dots, y_k, z_1, \dots, z_k \in \{0, 1\}^{3k}$ ,  $f$  outputs the majority of the three bits  $x, y, z$  where  $x = x_1 \oplus \dots \oplus x_k$ ,  $y = y_1 \oplus \dots \oplus y_k$ , and  $z = z_1 \oplus \dots \oplus z_k$ . Indeed, as long as less than a third of the bits are fixed, all the values  $x, y, z$  will be uniform and independent, and hence  $MAJ(x, y, z)$  will equal 1 with probability  $1/2$ . For Condition 2, note that for any fixing of at most  $1/10$  of the bits, when we choose at random all bits except for  $x_i$  (for  $i$  that is not fixed) then with probability  $1/2$  we will have  $y = z$ , in which case the value of  $f$  will stay the same no matter whether  $x_i$  is equal to 0 or to 1. On the other hand, there’s also a probability  $1/2$  that we will have  $y \neq z$ , in which case changing the value of  $x_i$  will flip the value of  $f$ .

## A.1 $k$ -wise independence

We start by showing that our generator is  $k$ -wise independent for  $k = n^{0.1}$ :

**Theorem A.3.** *Let  $G$  be an  $(m, n, d)$ -graph that is a  $(k, (1-\epsilon)d)$  expander, and let  $f$  be a  $\delta$ -resilient function for  $\delta > 2\epsilon$ . Then, the distribution  $G_f(U_n)$  is  $k$ -wise independent.*

*Proof.* The proof follows the proof of Part 1 of Theorem 4.2. Let  $Y = G(U_n)$ . We will prove the theorem by showing that for every subset  $S \subseteq [m]$  with  $|S| \leq k$ ,

$$\Pr\left[\bigoplus_{i \in S} Y_i = 1\right] = 1/2 \tag{3}$$

Indeed, by a simple counting argument, there exists  $i \in S$  such that  $|\Gamma_G(i) \setminus \Gamma_G(S \setminus \{i\})| \geq (1-2\epsilon)d$ . Therefore, if we fix all inputs in  $\Gamma_G(S \setminus \{i\})$  (thus fixing  $Y_j$  for all  $j \in S$  with  $j \neq i$ ), then by the  $2\epsilon$ -resiliency of  $f$ , the probability over the choice of inputs in  $\Gamma_G(i) \setminus \Gamma_G(S \setminus \{i\})$  that  $Y_i = 1$  is equal to  $1/2$ , establishing (3).  $\square$

Note that in this proof we only used Condition 1 of the definition of  $\delta$ -resilient functions. In particular, Theorem A.3 holds even if we use the *parity* function for  $f$ . (This was known before, see for example [MST03].) Note that, as mentioned above, Theorem A.3 implies that if Linial and Nisan's [LN90] conjecture is true, then  $G_f$  fools every  $\mathbf{AC}^0$  circuit.

## A.2 Fooling linear tests

We say that an  $(m, n, d)$ -graph is *almost right regular* if the right-degree of each vertex is at most  $2(m/n)d$ . We now show that if  $G$  is almost right regular and a good expander and  $f$  is a resilient function, then the distribution  $G_f(U_n)$  fools all linear tests (i.e., is an  $\epsilon$ -bias sample space). Note that random graphs satisfy these properties with high probability.

**Theorem A.4.** *Let  $G$  be an almost right regular  $(n\ell, n, d)$ -graph that is a  $(k, (1-\epsilon)d)$ -expander for  $k > \omega(\ell^2)$ . If  $f$  is  $\delta$  resilient for  $\delta > 2\epsilon$  then for every  $S \subseteq [m]$ ,*

$$\Pr\left[\bigoplus_{i \in S} Y_i = 1\right] \in 1/2 \pm 2^{-\Omega(k/\ell^2)}, \tag{4}$$

where the constant in the  $\Omega$  notation depends on  $d$  but not on  $\ell, n$ .

*Proof.* We may assume that  $|S| \geq k$ , since otherwise (4) is implied by  $k$ -wise independence (i.e., Theorem A.3). Let  $X_1$  be an input vertex that is connected to  $S$ . Let  $S_1$  be the set of at most  $d\ell$  output vertices in  $S$  that are connected to  $X_1$ , let  $V_1 = \Gamma_G(S_1)$  and let  $S'_1 = \Gamma_G(V_1)$  be the set of at most  $2d\ell^2$  output vertices that share an input with a member of  $S_1$ . Remove  $S'_1$  from  $S$  and continue in this way to obtain  $X_2, \dots, X_t$  for  $t \geq |S|/(2d\ell^2) = \Omega(k/\ell^2)$ . Note that by construction, the sets  $V_1, \dots, V_t$  are disjoint.

CLAIM: If we fix at random an assignment for the variables in  $V_i \setminus \{X_i\}$ , then with probability at least  $2^{-d}$ , the function mapping the bit  $X_i$  to  $\sum_{j \in S_i} Y_j$  is equal to  $X_i$  or to  $1 \oplus X_i$ .

The claim concludes the proof since then with probability  $1 - (1 - 2^{-d})^t = 1 - 2^{-\Omega(t)}$ , for any fixing of  $[n] \setminus \{X_1, \dots, X_t\}$ , the resulting function is a non-constant affine function of  $X_1, \dots, X_t$  and hence equals 1 with probability  $1/2$ .

PROOF OF CLAIM: Note that since  $X_i$  has right degree  $2\ell d < k$ ,  $|S_i| < k$ , and hence  $S_i$  is an expanding set, implying that there exists an output  $j \in S_i$  with  $|\Gamma_G(j) \setminus \Gamma_G(S_i \setminus \{j\})| \geq (1 - 2\epsilon)d$ . Now fix all inputs except for  $X_i$  in  $\Gamma_G(S_i \setminus \{j\})$ , this means that for every  $k \in S_i \setminus \{j\}$ ,  $Y_k$  is now a function of  $X_i$ , which is either a constant function or  $X_i \oplus b$  for some  $b \in \{0, 1\}$ , and in particular the same holds for  $\bigoplus_{k \in S_i \setminus \{j\}} Y_k$ . But now by the fact that  $f$  is  $\delta$ -resilient for  $\delta > 2\epsilon$ , if we choose at random the inputs in  $\Gamma_G(j) \setminus \Gamma_G(S_i \setminus \{j\})$  then we have positive (and at least  $2^{-d}$ ) probability for both the event that  $Y_j$  is a constant function of  $X_i$ , and the event that  $Y_j$  is equal to  $X_i \oplus b$  for some constant  $b$ . Thus, no matter that was the function  $\bigoplus_{k \in S_i \setminus \{j\}} Y_k$ , with probability at least  $2^{-d}$  the function  $Y_j \oplus \bigoplus_{k \in S_i \setminus \{j\}} Y_k = \bigoplus_{k \in S_i} Y_k$  will be a non-constant affine function of  $X_i$ .  $\square$

We note that a generator of small locality (number of inputs connected to each output) fooling linear tests was constructed before by Mossel et al [MST03]. The difference is that they were interested in a single construction with as small locality as possible while we want to show that a random graph (and even a sufficiently good expander) gives rise to such a generator. Their construction was obtained by XOR'ing together two generators on independent seeds. The first generator handled sparse tests using  $k$ -wise independence as in Theorem A.3. [MST03]'s second generator used a different construction and analysis than ours—they used a specific construction of locality two.

## B Validity of DUE: Omitted details from Section 5

### B.1 Counting cycles

**Notation:** For convenience of analysis, we use here slightly different distributions  $\hat{\mathcal{F}}$  and  $\hat{\mathcal{G}}$  over random and planted graphs than the candidate distributions  $\mathcal{G}$  and  $\mathcal{F}$  suggested in Section 1.1. We believe that the results below do extend to the distributions of Section 1.1, and in any case our cryptosystem can be modified to work with the distributions used here. We let  $\hat{\mathcal{G}} = \hat{\mathcal{G}}_{m,n,d}$  be a random bipartite graph with  $m$  left-side vertices,  $n$  right-side vertices, and where each edge is chosen independently with probability  $d/n$ .<sup>18</sup> The distribution  $\hat{\mathcal{F}}$  is chosen by taking a random graph from  $\hat{\mathcal{G}}_{m-q,n-q+1,d}$  and then adding  $q$  vertices to the left side, connecting them to a randomly chosen subset of size  $q - 1$  of the right side. We let  $c$  denote the value  $m/n$ , and assume  $c > d > 2$ . We let  $2k$  denote the length of the cycles we are considering (since this is a bipartite graph, the cycles must all be even). We always assume  $k < q^{1/4}$  (recall that the running time of an algorithm using such cycle counts is roughly  $n^k$ ). We let  $X = X_{m,n,d,k}$  denote the number of  $2k$ -cycles in  $\hat{\mathcal{G}}$  and  $X' = X'_{m,n,d,q,k}$  denote the number of  $2k$ -cycles in  $\hat{\mathcal{F}}$ . It can be shown that  $\mathbb{E}[X'] > \mathbb{E}[X]$ .

We present two results showing the limitations of cycle counts to distinguish between the two distributions when the stretch is large:

**Theorem B.1.** *In the notation above it holds that:*

$$|\mathbb{E}[X] - \mathbb{E}[X']| < \frac{10qk}{\sqrt{m}} \sigma(X),$$

where  $\sigma(X) = \sqrt{\text{Var}[X]}$  denotes the standard deviation of  $X$ . Moreover, for  $k > 10 \log n / \log c$  and every threshold  $\tau \in [\mathbb{E}[X], \mathbb{E}[X']]$ , if  $qk = o(\sqrt{n})$  then

$$|\Pr[X < \tau] - \Pr[X' < \tau]| < o(1)$$

<sup>18</sup>To use this graph for the DSPN assumption we could discard all vertices with too small a degree.

**Theorem B.2.** *In the notation above, for every  $\epsilon > 0$ , if  $d^2/c < \epsilon$ ,  $d^{10k}/q < \epsilon$ , and  $qd^{10k}/n < \epsilon$  then the statistical distance of  $X$  and  $C'$  is at most  $\epsilon$ .*

Theorem B.2 is only meaningful for  $k \ll \log q / \log d$ , but rules out the existence of any algorithm that uses only graph cycle count to distinguish between the two cases of DUE. By setting the stretch  $c$  to be a large enough power of  $d$ , we can ensure that Theorem B.1 is meaningful for every  $k$  that is not covered by Theorem B.2 and is not very large (e.g.,  $k < m^{1/4}$ ). However Theorem B.1 does not rule out all possible algorithms using the cycle count but only certain natural ones that test whether the count is above or below some threshold in  $[\mathbb{E}[X], \mathbb{E}[X']]$ . (For example, it does not rule out an algorithm that bases its decision on whether the count is even or odd.) We conjecture that (for  $k$  that is not too large) these natural algorithms are in fact *optimal* for these two distribution, and hence the statistical distance between  $X$  and  $X'$  is  $o(1)$  as long as  $q$  is not too large (say  $k < q^{1/4}$ ).

### B.1.1 Expectation and variance of cycle count: proof of Theorem B.1

We now prove Theorem B.1. We start by computing the expectation and variance of  $X$ . We let  $n^{(k)} = n!/(n-k)!$ . Note that  $n^k \geq n^{(k)} \geq n^k(1 - k^2/n)$  and hence we'll frequently use the approximation  $n^{(k)} \sim n^k$ . The expectation of the number of cycles in an  $(n, m, d)$  graph from  $\mathcal{G}$  is easily shown to be:

$$\mathbb{E}[X] = m^{(k)} n^{(k)} (d/n)^{2k} / k \sim c^k d^{2k} / k \quad (5)$$

To compute the variance, we write  $X = \sum_{\alpha} X_{\alpha}$ , where  $\alpha$  ranges over all the  $m^{(k)} n^{(k)} / k$  potential  $2k$ -cycles in a graph with  $n$  input and  $m$  output vertices, and  $X_{\alpha}$  is the indicator random variable that is equal to 1 if the cycle  $\alpha$  exists in the graph. Note that  $\mathbb{E}[X_{\alpha}] = (d/n)^{2k}$ . Now

$$\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \sum_{\alpha, \beta} \mathbb{E}[X_{\alpha} X_{\beta}] - \sum_{\alpha, \beta} (d/n)^{4k} \quad (6)$$

Clearly for every  $\alpha, \beta$ ,  $\mathbb{E}[X_{\alpha} X_{\beta}] \geq \mathbb{E}[X_{\alpha}] \mathbb{E}[X_{\beta}] = (d/n)^{4k}$ . Thus for every set  $H$  of pairs  $(\alpha, \beta)$ , the RHS of (6) is lower bounded by

$$\sum_{(\alpha, \beta) \in H} (\mathbb{E}[X_{\alpha} X_{\beta}] - (d/n)^{4k}).$$

Let  $H$  be the set of pairs  $\alpha, \beta$  that share exactly one edge (and hence two vertices). We can verify that  $|H| = m^{(2k-1)} n^{(2k-1)}$  and (using the approximation  $n^{(k)} \sim n^k$ ) this implies the following claim:

**Claim B.3.**  $\text{Var}[X] \geq |H|(d/n)^{4k} \geq c^{2k} d^{4k} / (2m) = \mathbb{E}[X]^2 (k/m)$

The first part of Theorem B.1 now follows from the following lemma:

**Lemma B.4.**  $\mathbb{E}[X'] = (1 \pm 10kq/m) \mathbb{E}[X]$

*Proof.* We write  $X' = \sum_{0 \leq a, b \leq k} X^{a,b}$  where  $X^{a,b}$  denotes the number of  $2k$  cycles that have  $a$  of their  $k$  output-side vertices in the planted shrinking set and  $b$  of their  $k$  input-side vertices in the neighborhood set of this planted set. Thus,  $X_{0,0} = X_{n, m-q, d}$  and hence has expectation  $(m-q)^{(k)} n^{(k)} (d/n)^{2k} / k$ , which as can be seen by writing  $m-q = m(1-q/m)$ , contributes a factor at most  $kq/m$  to the difference between  $\mathbb{E}[X]$  and  $\mathbb{E}[X']$ . On the other hand,  $X_{k,k} = X_{q, q-1, d}$  which has

expectation  $\sim d^{2k}$  which is negligible compared to  $\mathbb{E}[X]$ . We claim that for  $(a, b) \notin \{(0, 0), (k, k)\}$ ,  $\mathbb{E}[X^{a,b}] \leq \frac{c^k d^{2k} q}{k n} = (q/n) \mathbb{E}[X]$ . Indeed, if  $a > b$  then  $X^{a,b} = 0$  with probability one, since in any cycle in a bipartite graph, a set of  $\ell$  left-side vertices has at least  $\ell$  neighbors. Thus, if the cycle contains  $a$  left-side vertices that are in the planted shrinking set, then there must be at least  $a$  right-side vertices in the neighborhood of this set. Note also that the only case  $a = b$  is if the cycle is fully contained in either the planted set and its neighborhood, or has no vertices in either of them (i.e., if  $a = b = k$  or  $a = b = 0$ ). Thus we may assume  $a < b$ . Now,

$$\begin{aligned} \mathbb{E}[X^{a,b}] &= \frac{1}{k} m^{(k-a)} q^{(a)} n^{(k-b)} (q-1)^{(b)} \left(\frac{d}{n}\right)^{2(k-a)} \left(\frac{d}{q}\right)^{2a} \leq \\ &\frac{1}{k} c^{k-a} n^{2k-a-b} q^{a+b} d^{2k} q^{-2a} n^{-2k+2a} \leq \frac{c^k d^{2k}}{k} \left(\frac{q}{n}\right)^{b-a} \leq \frac{c^k d^{2k} q}{k n} \end{aligned}$$

□

**Proof sketch of “moreover” part.** By following the above calculation, we can see that the variance can also be *upper bounded* by the sum, for  $i = 1 \dots 2k$  of  $t_i = |H_i|(d/n)^{4k-i}$ , where  $H_i$  denotes the set of pairs of cycles that share  $i$  edges. Since for  $i < 2k$ , these must share at least  $i+1$  vertices, we can bound the term  $t_i$  for  $i < 2k$  by  $m^{2k} n^{2k-i-1} (d/n)^{4k-i} \leq c^{2k} d^{4k} / n = \mathbb{E}[X]^2 / n$ , while the term  $t_k$  is equal to  $\mathbb{E}[X]$ . Thus we have that

$$\mathbb{E}[X^2] \leq \mathbb{E}[X]^2 + \frac{2k}{n} \mathbb{E}[X]^2 + \mathbb{E}[X].$$

(Note that for  $\mathbb{E}[X] \gg n$ , the rightmost term is negligible.) We can use the same idea to also bound higher moments of  $X$ , and show that for every fixed  $\ell$

$$\mathbb{E}[X^\ell] \leq \mathbb{E}[X]^\ell + O\left(\frac{k}{n} \mathbb{E}[X]^\ell + \mathbb{E}[X]\right).$$

This upper bound on moments can be used to show the following:

**Claim B.5.** *Let  $Z = (X - \mathbb{E}[X])^2$ . Then  $\mathbb{E}[Z^2] \leq (1 + (100qk)/m) \mathbb{E}[Z]^2$ .*

We then use the following consequence of Cauchy-Schwarz that is sometimes known as the Paley-Zygmund inequality:

**Lemma B.6.** *For a nonnegative random variable  $Z$ , if  $\mathbb{E}[Z^2] \leq (1 + \epsilon) \mathbb{E}[Z]^2$  then*

$$\Pr[Z < \tau \mathbb{E}[Z]] < \tau^2 + \epsilon$$

This implies the following “anti-concentration” bound. Let  $\delta > 0$  be arbitrarily small and fix  $T$  to be  $\sqrt{\frac{\delta}{m}} \mathbb{E}[X]$ . Then

$$\Pr[|X - \mathbb{E}[X]| < T] = \Pr[(X - \mathbb{E}[X])^2 < T^2] < 100qk/m + \delta.$$

That is  $X$  is unlikely to be “too close” to its expectation. Roughly speaking, we then use the characterization of the proof of Lemma B.4 to present  $X'$  as equal to  $Y' + Y''$  where  $Y'$  is distributed identically to  $X$  and  $\mathbb{E}[|Y''|] = O\left(\frac{qk}{n} \mathbb{E}[X]\right) \ll T$ , meaning that  $|Y''| < T/10$  with  $1 - o(1)$  probability. Since  $\mathbb{E}[X] - \mathbb{E}[X'] < T/10$ , this means that for every  $\tau \in [\mathbb{E}[X], \mathbb{E}[X']]$ , we can bound  $|\Pr[\mathbb{E}[X] < \tau] - \Pr[\mathbb{E}[X'] < \tau]|$  by the probability that  $X$  is in  $[\mathbb{E}[X] - T/2, \mathbb{E}[X] + T/2]$  up to some  $o(1)$  additive term.

### B.1.2 Poisson approximation of short cycle count: proof of Theorem B.2

Theorem B.2 will follow from the following lemma:

**Lemma B.7.** *Let the numbers  $n, m = cn, d, q, k$  and random variables  $X, X'$  be as above, then*

$$\Delta(X, P_\lambda) < \epsilon \quad (7)$$

$$\Delta(X', P_{\lambda'} + P_{\lambda''}) < \epsilon \quad (8)$$

where  $\epsilon = 10kd^{4k}(c^{2k}q/n + 1/q)$ ,  $\lambda = c^k d^{2k}/k$ ,  $\lambda'' = (1 - q/m)^k \lambda$ ,  $\lambda' = d^{2k}/k$ ,  $P_\lambda$  denotes the Poisson distribution with expectation  $\lambda$ , and  $\Delta(\cdot)$  denotes statistical distance.

Using basic properties of the Poisson distribution we get the following corollary:

**Corollary B.8.** *For any  $\epsilon > 0$ , if  $d^2 < \epsilon c$ ,  $d^{10k} < \epsilon q$ , and  $q < \epsilon n/d^{10k}$  then  $\Delta(X, X') < \epsilon$*

*Proof.* We use the facts that  $P_\lambda + P_{\lambda'} \equiv P_{\lambda + \lambda'}$  and that  $\Delta(P_\lambda, P_{\lambda(1+\epsilon)}) \leq \epsilon\sqrt{\lambda}$ . In our case, the condition  $d^2 < \epsilon c$  implies that  $\lambda'^2/\lambda = \frac{d^{4k}}{kc^k d^{2k}} < \epsilon^k/k \leq \epsilon^2/2$  (we can assume  $k \geq 2$ ). For the purposes of bounding statistical distance increasing  $c$  only helps,<sup>19</sup> and hence we may assume  $c = d^2/\epsilon$ . Plugging this into the bounds of Lemma B.7 gives the corollary.  $\square$

We now turn to proving Theorem B.7:

*Proof of Theorem B.7.* We start by showing (7). As in the proof of Theorem B.1, we write  $X = \sum_\alpha X_\alpha$  where  $\alpha$  ranges over all the  $m^{(k)}n^{(k)}/k$  potential  $2k$  cycles and  $X_\alpha$  is the indicator variable that is equal to 1 if the cycle  $\alpha$  exists in the graph. We note that if  $\alpha, \beta$  do not share an edge then  $X_\alpha$  and  $X_\beta$  are independent.

Barbour [BAR82] (see exposition in [BOL01, § 4.3, Pf of Thm 4.16]) proved the following lemma:

**Lemma B.9.** *Let  $X = \sum_\alpha X_\alpha$  where for every  $\alpha$ ,  $X_\alpha$  is an indicator variable. Suppose moreover that  $\mathbb{E}[X_\alpha X_\beta] \geq \mathbb{E}[X_\alpha] \mathbb{E}[X_\beta]$  for every  $\alpha, \beta$  and there is a symmetric reflexive relation  $\sim$  such that  $\alpha \not\sim \beta$  implies that  $X_\alpha, X_\beta$  are independent. Then,*

$$\Delta(X, P_{\mathbb{E}[X]}) \leq 4 \left( \sum_{\substack{\alpha \sim \beta \\ \alpha \neq \beta}} \mathbb{E}[X_\alpha X_\beta] + \sum_\alpha \mathbb{E}[X_\alpha]^2 \right) \quad (9)$$

In our case the relation  $\sim$  is sharing at least one edge, and for each  $\ell \in \{1..2k-1\}$  we count the contribution to the RHS of (9) of the pairs of cycles that share  $\ell$  edges. Since they must share at least  $\ell + 1$  vertices, this contribution can be bounded by

$$m^{(2k - \lfloor (\ell+1)/2 \rfloor)} n^{(2k - \lceil (\ell+1)/2 \rceil)} (d/n)^{4k-\ell} \leq d^{4k} c^{2k}/n,$$

thus establishing (7).<sup>20</sup>

To show (8), we follow the proof of Lemma B.4, and write  $X'$  as  $X' = \sum_{0 \leq a, b \leq k} X^{a,b}$ . The same calculation as above says that  $X_{k,k}$  is within  $4kd^{4k}/q$  distance to the Poisson distribution

<sup>19</sup>We can always transform an input graph with  $m$  left vertices into a graph with  $m' > m$  vertices by adding  $m' - m$  vertices each with random neighbors.

<sup>20</sup>Note that  $\sum_\alpha \mathbb{E}[X_\alpha^2] \leq c^k d^{4k}/n^{4k} \leq (cd/n)^k$ .

$P_{\lambda}$ . Thus  $X_{0,0} + X_{k,k}$  is  $4kd^{4k} (c^{2k}/n + 1/q)$ -close to  $P_{\lambda} + P_{\lambda'}$ . Thus all that is left is to bound the probability that  $X^{a,b}$  is non zero for  $(a,b) \notin \{(0,0), (k,k)\}$ . But in the proof of Lemma B.4, we show that  $\mathbb{E}[X^{a,b}] \leq \frac{c^k d^{2k}}{k} \frac{q}{n}$ . Hence by Markov we get that the probability that  $X^{a,b} > 0$  (and hence greater or equal to 1) is this expectation.  $\square$

## B.2 Reductions from other graph problems

To state our results we introduce the following notation. Fixing  $n, m, d$ , for every  $q, e, q', e'$  with  $q' > q, e' > e$  we let  $\text{DUE}'_{(q,e)\text{vs}(q',e')}$  denote the gap problem of distinguishing, given a bipartite graph  $G = (V_{\text{In}}, V_{\text{Out}}, E)$ , between the following two case:

**YES case:** There exists a subset  $S \subseteq V_{\text{In}}$  of size at most  $q$  such that  $|\Gamma(S)| < e|S|$

**NO case:** For every subset  $S \subseteq V_{\text{In}}$  of size at most  $q'$ ,  $|\Gamma(S)| > e'|S|$ .

By abuse of notation, we also denote by  $\text{DUE}'_{(q,e)\text{vs}(q',e')}$  the assumption that the above problem is hard on the average, in the sense that there exist two sampleable distributions  $\mathcal{G}, \mathcal{F}$  over  $(m, n, d)$ -graphs such that  $\mathcal{F}$  is in the YES case with  $1 - o(1)$  probability, and  $\mathcal{G}$  is in the NO case with  $1 - o(1)$  probability, but no polynomial-time algorithm can distinguish between the two with advantage better than, say,  $1/100$ .<sup>21</sup> The  $\text{DUE}'$  assumption corresponds to the  $\text{DUE}'_{(q,e)\text{vs}(q',e')}$  assumption for  $e = 1, e' = 0.9d$ , and  $q'$  equalling the parameter  $k$  (see Section 3.3.1).

**Small set expansion.** Fix  $n$  to be some graph size parameter. For every  $q, q', e, e'$  such that  $q' > q, e' > e$  we define the promise problem  $\text{SSE}_{(q,e)\text{vs}(q',e')}$  whose input is an  $n$  vertex graph  $G = (V, E)$  (not necessarily bipartite) as follows:

**YES case:** There is a set  $S \subseteq V$  of size at most  $q$  such that  $|\Gamma_G(S)| < e|S|$ .

**NO case:** For every  $S \subseteq V$  of size at most  $q'$ ,  $|\Gamma_G(S)| > e'|S|$ .

We show the following theorem:

**Theorem B.10.** *For every  $\epsilon > 0$  and integer  $e$  that divides  $d$ ,  $\text{SSE}_{(q,e)\text{vs}(q',(1-\epsilon)d)}$  on  $n$ -sized graphs of degree  $d$  reduces to  $\text{DUE}'_{(eq,1)\text{vs}(q',(1-2\epsilon)d)}$  on  $(en, n, d/e)$  graphs.*

(The condition that  $e$  is an integer that divides  $d$  can be easily dropped at the cost of a slightly more cumbersome statement.)

*Proof.* Given a graph  $G = (V, E)$  input to  $\text{SSE}$ , we construct the graph  $G' = (V_{\text{Out}}, V_{\text{In}}, E)$  as follows. Each vertex  $u \in V$  has one corresponding vertex  $u' \in V_{\text{In}}$  and  $e$  corresponding vertices  $u'_1, \dots, u'_e$  in  $V_{\text{Out}}$ . We split the  $d$  neighbors of  $u$  arbitrarily to  $e$  groups  $S_1, \dots, S_e$  of size  $d/e$  each. For every  $v \in S_i$ , we connect  $u'_i$  to  $v$ . Thus each vertex  $u'_i$  will have  $d/e$  neighbors corresponding to  $d/e$  of the neighbors of  $u$  in  $G$ .

<sup>21</sup>For simplicity of analysis, we will allow both distributions  $\mathcal{F}$  and  $\mathcal{G}$  to be over graphs  $G = (V_{\text{In}}, V_{\text{Out}}, E)$  such that it does not necessarily hold that  $|V_{\text{In}}| = n$  and  $|V_{\text{Out}}| = m$ , but rather  $|V_{\text{In}}|$  and  $|V_{\text{Out}}|$  are random variables concentrated around  $n$  and  $m$  respectively. Note that this does not matter for our cryptosystem applications. We believe our analysis can extend to the case that  $\Pr[|V_{\text{In}}| = n, |V_{\text{Out}}| = m] = 1$ .

Clearly, for every set  $S$  in  $G$ , the corresponding set  $S'$  in  $G'$  has size  $e|S|$  and  $|\Gamma_G(S)|$  neighbors. In particular, if  $|\Gamma_G(S)| < e|S|$  then  $|\Gamma_{G'}(S')| < |S'|$ . On the other hand we claim that if every set  $S$  of size at most  $q'$  in  $G$  has  $(1 - \epsilon)d|S|$  neighbors, then every set  $S'$  of size  $q'$  in  $G'$  has at least  $(1 - 2\epsilon e)(d/e)q'$  neighbors. Suppose otherwise, then  $S'$  has more than  $2\epsilon dq'$  non-unique edges, where we say that an edge  $(u, v)$  out of  $S'$  is *non unique* if there is some other edge  $(w, v)$  in  $G'$  with  $w \in S'$ . Now let  $S$  be the set (of size at most  $q'$ ) of all vertices in  $G$  corresponding to the vertices in  $S'$ . Then  $S$  will have also more than  $2\epsilon dq' \geq 2\epsilon d|S|$  non-unique edges, implying that it has less than  $(1 - \epsilon)d|S|$  neighbors.  $\square$

**Remark B.11.** The resulting graph of the reduction does not seem highly imbalanced, in the sense that if, say  $e = 2$ , then it will be only an  $(2n, n, d)$ . However, imbalance can always be increased by either adding more vertices to  $V_{\text{Out}}$  and connecting each one to  $d$  random neighbors, or “hashing down”  $V_{\text{In}}$  by composing it with, say, a random unbalanced expander.

**Planted clique problem.** We define the *decisional planted  $k$ -clique problem in  $G_{n',p}$*  ( $k\text{DPC}_{n',p}$  for short) as the problem of distinguishing between a random graph from  $G_{n',p}$  and a random graph in  $G_{n',p}$  in which we add edges to make a random  $k$ -sized subset of vertices a clique. The search variant of this problem (where one is looking to find the planted set) has been fairly widely studied for  $p = 1/2$  and currently the best-known polynomial-time algorithms only work when  $k = \Omega(\sqrt{n})$  [AKS98]. To our knowledge, for substantially smaller  $k$  (e.g.,  $k = n^{0.1}$  or even  $k = 2^{\log^{0.9} n}$ ) there are no non-trivial algorithms for either the search or decision problems, and even for smaller values of  $p$ . (Note that there is a trivial  $n^{O(\log n)}$ -time distinguishing algorithm, since a random graph has maximum clique of size at most  $2 \log n$  with high probability.) Our first result is the following:

**Theorem B.12.** *The  $k\text{DPC}_{n',2^{-\ell}}$  problem reduces to  $\text{DUE}'_{(k^2/3,1)_{\text{vs}}(2^{\ell/10}, dk/(30 \log n'))}$  in  $(m, n, d)$  graphs for  $m = \Theta(n^2 2^{-\ell})$ ,  $n = n'$  polylog( $n'$ ) and  $d = k/3$ .*

A seemingly reasonable setting of parameters would be  $\ell = 100 \log^{0.99} n'$  and  $q = 2^{\log^{0.95} n}$ , in which case the conclusion will be the hardness of  $\text{DUE}'_{(q,1)_{\text{vs}}(2^{\log^{0.9} n}, d^{0.9})}$ . However, this conclusion is not fully satisfying since the expansion is only  $d^{0.9}$  as opposed to, say,  $0.9d$ . Since our goal is to use this with the  $\text{DSPN}'$  assumption, we need expansion parameters that make the latter assumption reasonable. In particular, we need parameters that ensure that the adjacency matrix of the graph has no short (i.e., less than  $1/\mu$ ) linear dependency, since such a dependency could be used to break  $\text{DSPN}'$  in the same way that our decryption algorithm works. To have expansion imply any non-trivial condition on such linear dependencies, we need the expansion to be *lossless*— namely larger than  $d/2$ . While we can't get quite that, we do get a somewhat close condition— *unique neighbor expansion*.<sup>22</sup> A graph  $G = (V_{\text{In}}, V_{\text{Out}}, E)$  is a *unique neighbor expander for  $q$  sets* if for  $S \subseteq V_{\text{Out}}$  with  $|S| \leq q$ , there exists  $u \in \Gamma(S)$  that has only one neighbor in  $S$ . It's easy to see that such a graph  $G$  has expansion factor greater than 1 for sets of size  $\leq q$ , and that there are no  $q$  rows in the adjacency matrix of  $G$  that are linearly dependent. We have the following result on unique neighbor expansion:

<sup>22</sup>We do not think that on its own, unique neighbor expansion of  $G$  will imply pseudorandomness of  $G_\mu(U_n)$ . However, perhaps variants of this property could be sufficient (e.g. perhaps requiring that a constant fraction of the vertices in  $S$  have  $\Omega(d)$  unique neighbors).

**Theorem B.13.** *The  $k\text{DPC}_{n',2^{-\ell}}$  problem reduces to  $\text{DUE}'_{(k^2/3, O(\log^2 n/k\ell))_{\text{vs}} (2^{\ell/10, \text{u.n.}})}$ , where by this we denote the variant of  $\text{DUE}'$  where the NO condition is replaced with being a unique neighbor expander for sets of size at most  $2^{\ell/10}$ .*

**Remark B.14.** Note that our reduction for small set expansion is a worst-case gap preserving reduction, which in particular means that  $\text{DUE}'$  if there is some distribution that makes, say,  $\text{SSE}_{(q,2)_{\text{vs}} (q',0.99d)}$  hard. In contrast, the reduction from planted clique is an average-case to average-case reduction that uses the particular distribution over the inputs in the planted clique problem.

### B.2.1 Proof of Theorem B.12

Our reduction is very simple, and uses the notion of an edge-vertex incidence graph. This allows to relate the clique question to expansion, as is encapsulated by the following immediate observation:

**Claim B.15.** *For every graph  $G = (V, E)$ , let  $\hat{G}$  be the edge-vertex incidence graph of  $G$ .<sup>23</sup> Then,  $G$  has a  $k$ -clique if and only if there is a subset  $S$  of  $\binom{k}{2}$  left vertices of  $\hat{G}$  such that  $|\Gamma_{\hat{G}}(S)| \leq k$ .*

The following simple lemma is the heart of the proof. It implies that the edge-vertex incidence graph of a random graph  $G$  from  $G_{n,p}$  will be a decent expander:

**Lemma B.16.** *With high probability over  $G$  chosen from  $G_{n',2^{-\ell}}$ , for every  $t < 2^{\ell/10}$ , every subset of  $t$  edges of  $G$  touches at least  $t \cdot \frac{\ell}{10 \log n'}$  vertices.*

*Proof.* Let's bound the probability  $p_{k,t}$  that there exists a set of  $k$  vertices whose induced graph has at least  $t$  edges. By using the simplest bounds,

$$p_{k,t} \leq \binom{n'}{k} \binom{k^2}{t} 2^{-\ell t} \leq n'^k k^{2t} 2^{-\ell t}.$$

Taking logs we see that as long as

$$k \ll t\ell/10 \log n'$$

this probability will be very close to 0. In our setting  $\log k \ll \ell$ , and hence we only need to show  $k \log n' \ll \ell t$ , which holds if  $t > (10 \log n'/\ell)k$ .  $\square$

**Proof of Theorem B.12 from Lemma B.16.** Let  $\hat{G}$  be the edge-vertex incidence graph of  $G$ . Note that in the planted case we'll have a set  $S$  of size at least  $k^2/3 + 1$  (actually  $\binom{k}{2}$ ) output vertices with only  $k$  neighbors. Now make  $k/3$  copies of every input vertex  $u$  of  $\hat{G}$  and connect these copies to the same neighbors as  $u$ . The resulting graph has degree  $2k/6$  and the expansion has increased by a factor  $k/3$ , meaning that still  $|\Gamma(S)| < |S|$ . On the other hand in the random case, the lemma implies that for every set  $S$  of size at most  $2^{\ell/10}$ , its expansion in the new graph is at least  $\ell k/(30 \log n)$ .  $\square$

---

<sup>23</sup>That is,  $\hat{G}$  is the  $(|E|, |V|, 2)$  bipartite graph such that the  $e^{th}$  left vertex of  $\hat{G}$  is connected to the two vertices of the  $e^{th}$  edge in  $G$ .

### B.2.2 Proof of Theorem B.13

We now sketch the proof for Theorem B.13. The proof is inspired by the Zig-Zag product [RVW00]. Say that a function  $D : [m] \times [d'] \rightarrow [s']$  is an  $s$ -lossless disperser if for every  $s$ -sized subset  $S$  of  $[m]$ , there exists  $i \in [d']$  such that the mapping  $|D(S, \{i\})| > 0.9|S|$ . For  $d' = 100 \log m$ , a random function  $D : [m] \times [d'] \rightarrow [100s]$  will be such a disperser with high probability.

We can look at an  $(m, n, d)$  graph  $G$  as a function from  $[m] \times [d]$  to  $[n]$ , which we also denote by  $G$ . Let  $s = 100 \log n / \ell$  and define the function  $G' : [m] \times ([d] \times [d']) \rightarrow [n] \times [100s] \times [d]$  as follows:

$$G'(u, i, j) = \langle G(u, i), D(u, j), j \rangle.$$

For every set  $S \subseteq [m]$  of vertices of  $G$ , if  $|\Gamma_G(S)| \geq |S|/s$  then there exists  $u \in \Gamma_G(S)$  with at most  $s$  preimages in  $G$ . Let  $S_u$  be the set of these preimages. For some  $i \in [d']$ , this set  $S_u$  will be mapped by  $D$  to at  $0.9|S_u|$  outputs, and hence there will be some  $x \in S_u$  with the unique neighbor  $\langle u, D(x, i), i \rangle$ . On the other hand, clearly for every  $S \subseteq [m]$ ,

$$|\Gamma_{G'}(S)| \leq |\Gamma_G(S)| \cdot O(sd').$$

Now let  $G$  be the  $(m, n, 2)$  graph obtained from the proof of Theorem B.12. In the NO case every not too large (less than  $2^{\ell/10}$  vertices) subset  $S$  of  $G$  has at least  $|S|\ell/(10 \log n)$  vertices. Thus, setting  $s = 10 \log n / \ell$ , the graph  $G'$  will be a unique neighbor expanders. However, in the YES case there will be a set of  $k^2$  vertices with  $k$  neighbors, and hence in  $G'$  this set will have at most  $O(ksd) = O(k \log^2 n / \ell)$  neighbors.