Public-Key Cryptography from Different Assumptions

Benny Applebaum* Boaz Barak† Avi Wigderson‡

April 6, 2010

Abstract

This paper attempts to broaden the foundations of public-key cryptography. We construct new public-key encryption schemes based on new hardness-on-average assumptions for natural *combinatorial* **NP**-hard optimization problems. We consider the following assumptions:

1. It is infeasible to solve a random set of sparse linear equations mod 2, of which a small fraction is noisy.
2. It is infeasible to distinguish between a random unbalanced bipartite graph, and such a graph in which we "plant" at random in the large side a set S with only $|S|/3$ neighbors.
3. There is a pseudorandom generator in $\mathbf{NC^0}$ where every output depends on a random constant-size subset of the inputs.

We obtain semantically secure public-key encryption schemes based on several combinations of these assumptions with different parameters. In particular we obtain public-key encryption from Assumption 1 on its own, yielding the first noisy-equations type public-key scheme in which the noise rate is higher than one over the square root of the number of equations. We also obtain public-key encryption based on a combination of Assumptions 2 and 3. These are arguably of more "combinatorial"/"private-key" nature than any assumptions used before for public-key cryptography. Our proof involves novel "search to decision" and "search to prediction" reductions for *sparse* noisy linear equations.

The strength of our assumptions raise new algorithmic and pseudorandomness questions (and new parameters for old ones). We give some evidence for these assumptions by studying their resistance to certain classes of natural algorithms, including semi-definite programs, $\mathbf{AC^0}$ circuits, low-degree polynomials, and cycle counting. We also relate our assumptions to other problems such as planted clique and learning juntas.

*Department of Computer Science, Princeton University, 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. 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. Supported by NSF Grant CCF-0832797.

Contents

1 Introduction **3**

2 Our results and related work **4**
2.1 New cryptosystems 4
2.2 Evidence and applications of our assumptions 6
2.3 Prior works 7

3 Overview of the proofs of Theorems 2.2 and 2.3 **8**

I Cryptosystems **11**

4 Preliminaries **11**

5 PKE based on the Hardness of 3-LIN **12**
5.1 The Search3LIN Problem 12
5.2 A Public-Key Encryption Scheme 14

6 Proof of Theorem 5.5 **16**
6.1 From Approximate Search to Search 16
6.1.1 Proof of Lemma 6.3 17
6.2 From Prediction to Approximate Search 18
6.2.1 Proof of Lemma 6.7 19
6.3 Predicting over different distributions 20
6.4 Completing the proof of the main theorem 22

7 PKE based on d-LIN and Decisional Unbalanced Expansion **22**
7.1 The Assumption 22
7.2 A Public-Key Encryption Scheme 24
7.2.1 Security 24
7.3 Proof of Thm 7.4 25
7.4 Proof of Thm 7.3 28
7.4.1 Using a predictor to reduce AppSearchLIN to 2-LIN 28
7.4.2 Finding an approximate solution to 2-LIN 30
7.4.3 Complementing the proof of Thm. 7.11 31

8 PKE based on DUE and DSF **31**
8.1 The DSF assumption 31
8.2 Constructing PKE 32
8.3 Hardness of Learning Juntas 33

II Investigating assumptions **34**

9 Unconditional hardness of d-LIN **34**

10 On the hardness of DUE **36**
10.1 Counting cycles . . . 37
10.2 Reductions from other graph problems . . . 38

11 On the hardness of DSF **39**
11.1 k-wise independence . . . 40
11.2 Fooling linear tests . . . 40

12 Discussion and open problems **41**

References **43**

A Two Sampling Lemmas **47**

B Hardness of DUE: Technical details **49**
B.1 Counting cycles . . . 49
B.1.1 Expectation and variance of cycle count: proof of Theorem B.1 . . . 49
B.1.2 Poisson approximation of short cycle count: proof of Theorem B.2 . . . 51
B.2 Reductions from other graph problems . . . 52
B.2.1 Proof of Theorem B.12 . . . 54
B.2.2 Proof of Theorem B.13 . . . 55

1 Introduction

Public key encryption (PKE) is a central notion in cryptography, and many of the exciting cryptographic applications in theory and practice are based on it. But 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 or geometric nature, from the areas of number theory, lattices, and error-correcting codes (e.g., [DH76, RSA78, McE78, AD97]). This leaves open the troublesome possibility that a new mathematical breakthrough could render them insecure.

In this aspect public-key cryptography ("cryptomania" in the language of Impagliazzo [Imp95]) seems very different from *private key cryptography* ("minicrypt") where many different candidates exist, and can be based on seemingly much less structured combinatorial problems including natural average-case variants of **NP**-complete problems such as random 3-SAT [ACO08], planted clique [JP00], and learning parity with noise [GKL88, BFKL94].[1] Thus a major goal of cryptography is to base public-key encryption on assumptions that are weaker, or at least different, than those currently used.

A complete solution to this problem would be obtained by constructing public key encryption based solely on the existence of one-way functions. This is a longstanding open problem, and cannot be achieved via black-box reductions [IR89]. Short of that, we believe that major progress would be made by a construction of public key encryption based on a natural and well-studied average-case variant of an **NP**-complete problem. This paper is a step in this direction.

In this work we give constructions of a public key encryption based on different assumptions about the hardness of combinatorial problems (e.g., satisfying random local constraints and detecting graph expansion). The proposed systems are not as efficient as some known candidate constructions, and are 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 schemes is the relatively general and unstructured nature of the new assumptions. These include a variant of the *planted densest subgraph* problem, a pseudorandom generator based on the expander-based one-way function of Goldreich [Gol00] (a *private-key* primitive), and the 3LIN problem which can be seen as a sparse variant of the *learning parity with noise* problem with noise level much higher than those used before in public-key cryptography (in particular larger than $1/\sqrt{m}$, see Footnote 1). These seem qualitatively different than previous assumptions.

[1] Learning parity with noise (LPN), as well as the related mod p variant of "learning with errors" (LWE), have been used for public key cryptography as well [Ale03, Reg05, Pei09]. However, the known public key schemes require noise levels much lower than those needed for private-key cryptography. In particular all of these schemes inherently require noise of magnitude $\varepsilon < 1/\sqrt{m}$, where m is the number of equations. This seems to make a *qualitative* difference. Some evidence for this is the fact that for $\varepsilon < 1/\sqrt{m}$ LWE can be solved in **SZK** [GG98] and even (a suitable promise problem variant of) $\mathbf{NP} \cap \mathbf{coNP}$ [AR04], while in the worst-case these problems are **NP**-hard for sufficiently large noise [ABSS93, DKRS03].

2 Our results and related work

2.1 New cryptosystems

We say that a bipartite graph G is an (m, n, d)-*graph*, if it has m vertices on one side (which we call the "top" side), n vertices on the other side (called the "bottom"), and every top vertex has degree d. Similarly, an (m, n, d)-*matrix* is an $m \times n$ matrix over GF(2), in which every row has d entries of value 1. Roughly speaking, we consider the following assumptions (see Assumptions 5.2, 7.2 and 8.1 for precise statements):

Assumption $\mathsf{dLIN}(m, \varepsilon)$ It is infeasible to recover x from $(A, Ax + e)$, where A is a random (m, n, d) matrix, x is chosen randomly from $\mathrm{GF}(2)^n$, and $e \in \mathrm{GF}(2)^m$ is chosen such that $e_i = 1$ with probability ε independently for every i.

Assumption $\mathsf{DUE}(m, q, d)$ (Decisional Unbalanced Expansion) It is infeasible to distinguish between: **(a)** a random (m, n, d)-graph and **(b)** a random (m, n, d) graph in which the edges going out of a random q-sized subset S of the top vertices are modified to ensure S will have only $q/3$ neighbors.

Assumption $\mathsf{DSF}(m, d)$ (Decisional Sparse Function) With high probability, the following d-local, $\mathbf{NC^0}$ mapping G of n to m bits is a pseudorandom generator: every output bit of $G(x_1, \ldots, x_n)$ is $\mathrm{MAJ}(x', x'', x''')$ where each of x', x'', x''' is the parity of $d/3$ random coordinate of x.

In all of the above, "infeasibility" and "pseudorandomness" are defined with respect to probabilistic polynomial time (PPT) algorithms with some constant success probability (e.g., 0.99). The parameters m, d, q, ε can be functions of n. We construct three public-key encryption schemes each based on a different combination of the above assumptions:

Theorem 2.1. *For every constants $c > 0$ and function $m = m(n), d = d(n)$, if both Assumptions $\mathsf{DUE}(m, c \log n, d)$ and $\mathsf{DSF}(m, d)$ hold then there exists a semantically secure public key encryption.*

Both the DUE and DSF assumptions are arguably much more "combinatorial" and of a "private key" nature than any assumptions used before to construct public-key cryptography. DSF assumes that a variant of Goldreich's candidate one-way function is a pseudorandom generator— a strong assumption but still of a "private key" nature. DUE is closely related to the *densest subgraph problem*— a combinatorial optimization problem of independent interest [FPK01, Kho04, BCC+10, ABBG10].

Indeed, 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-th hyperedge of H contains the d neighbors of the i-th top-vertex of G. In this formulation, the DUE assumption is about the hardness of distinguishing hypergraphs that contain a somewhat *dense* sub-hypergraph — a set T of $q' = q/3$ vertices, such that the induced sub-hypergraph on T has at least q hyperedges— from graphs where the induced sub-hypergraph of every set of q' vertices (for q' up to roughly $n^{0.1}$ size or some other super-logarithmic bound) has only about q'/d edges. Thus DUE is equivalent to the problem of distinguishing between a random fairly sparse hypergraph ($m = O(n)$ hyperedges) and a random hypergraph with a planted somewhat *dense* (average degree larger than 1) small subgraph.[2]

[2]Note however that there is a more "algebraic" view to DUE, since a set of q vertices with $< q$ neighbors will result in a linear relation of length at most q in the rows of the adjacency matrix. So, one can think of DUE also as a

Note that we use DUE with a planted set of size $O(\log n)$. While, generally speaking, making the set smaller doesn't necessarily make the problem easier, there is always a brute force attack of time $\binom{n}{q}$, and hence the scheme obtained can be at best secure against $t^{O(\log t)}$-time adversaries, where t is the running time of the honest parties. While ideally one would want at least sub-exponential security, we note that the best construction of public key encryption using (even idealized) one-way functions has security at most $O(t^2)$ [Mer78, BGI08] and this is optimal for black-box methods [IR89, BMG09].

Theorem 2.2. *There is a constant c so that if Assumption $\mathsf{3LIN}(cn^{1.4}, n^{-0.2}/c)$ holds then there exists a semantically secure public key encryption.*

The 3LIN problem is a central and well studied constraint satisfaction problem. Furthermore, the above parameters seem to resist sub-exponential time algorithms (see Section 2.2.) It should be mentioned that other public key encryption schemes were based on solving (dense) random noisy equations mod 2 and mod p [Ale03, Reg05, Pei09]. Still our assumption seems different from those due to the sparsity and the use of larger noise rate (see Footnote 1 and Section 2.3). Moreover, our assumption is based on the hardness of a *search* problem (i.e. find an x satisfying most equations) with parameters for which *refutation* variant of 3LIN (i.e. certify that no such x satisfying most equations exists) is harder than refuting a random 3SAT formula with $O(n^{1.4})$ clauses. Note that finding an efficient algorithm to refute random 3SAT formulas with $o(n^{1.5})$ clauses is a longstanding open problem. Random 3SAT is perhaps the prototypical example of a "combinatorial" average-case computational problem. Of course, this connection is not formal, only suggestive, and does not directly shed light on the strength of our assumption, as no reductions are known between the *search* and *refutation* versions of random noisy 3XOR.

Theorem 2.3. *For every constants d, c and $q = q(n)$, there exists a constant c' such that if $d\mathsf{LIN}(c'n \log n, 1/(c'q)))$ and $\mathsf{DUE}(cn, q, 2d)$ hold then there exists a semantically secure public key encryption.*

Compared to Thm. 2.2, Thm. 2.3 allows us much more flexibility in the choice of parameters for 3LIN; specifically, we avoid the parameter range in which [FKO06]'s non deterministic algorithm for the refutation variant of this problem works. This comes at the expense of using the additional, combinatorial assumption DUE.[3] Again, it seems (see Section 2.2) that the resulting schemes achieves sub-exponential security.

We stress that we do *not* claim that our cryptosystems are "better" or "more secure" than previous candidates for public key encryption. Indeed, the integer factoring problem underlying schemes such as [Rab79] is probably the most well-studied average-case computational problem. Also, lattice based systems such as [AD97, Reg05, Pei09] have the important advantage of being based on *worst-case* problems such as gap shortest vector and shortest linearly independent vectors. Nevertheless we believe our constructions do suggest that problems with less algebraic or geometric structure may be useful for public key cryptography.

shortest codeword problem for the dual code of this matrix. However due to the imbalance, the dual code here has rate so close to 1 that it contains rather short codewords in any case (e.g., if $m = n^c$ then the dual code is a code over $\mathrm{GF}(2)^m$ with dimension $m - m^{1/c}$ and so regardless will have a codeword of length $m^{1/c}$).

[3] We note that known algorithms for DUE (i.e., counting small subgraphs) place some restrictions on the value of q for which $\mathsf{DUE}(cn, q, d)$ and we'll need $q \in [n^\varepsilon, \sqrt{n}]$ where ε is some constant depending on c. The actual range of parameters for which our result holds is somewhat broader.

2.2 Evidence and applications of our assumptions

To test the validity of our assumptions, we show unconditionally that they do resist various concrete algorithms, as well as provide some reductions between these and other computational problems. While our primary motivation is to broaden the foundations for public key cryptography, we believe that the computational problems we use are natural and interesting in their own right.[4] They broaden the class of hardness-on-average and pseudorandomness problems studied in the past in both the algorithmic and cryptographic communities, and focus attention on parameters of significance for public-key encryption.

The dLIN problem. We show that for the parameters we use, the noisy linear equation problem 3LIN *unconditionally* resists: (1) "Myopic" attacks that look at the entire matrix but only at some n^δ of the "noisy bits", or those that look at linear combinations of these "noisy bits". (2) Attacks that apply low-degree polynomials or $\mathbf{AC^0}$ circuits to the "noisy bits". (3) n^δ rounds of the Lasserre hierarchy [Las01] of semi-definite programs, for some constant $\delta > 0$. The first item follows similarly to the analysis of Mossel et al [MST03], the second item employs the results of Viola [Vio08] and Braverman [Bra09], and the third item is implied by Schoenebeck [Sch08]. (See Section 9 for formal statements and proofs of all these results.)

The last item is especially interesting as semidefinite programs seem to be the strongest algorithmic tool that is currently available to attack constraint satisfaction problems. Moreover, the Lasserre hierarchy is strictly stronger than other hierarchies for linear and semidefinite programs such as the Lovasz-Schrijver [LS91] hierarchies (LS, and LS+) and the Sherali-Adams [SA90] hierarchy [Lau03].

We also obtain a new (average-case) reduction from the dLIN problem into its decisional version (where one needs to distinguish $(1-\varepsilon)$-satisfiable random equations from completely random ones that will be of course only $1/2+o(1)$ satisfiable). 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 which turns to be significantly more challenging. As the sparse case is an important variant of this problem (see [Ale03, AIK04]), we believe that our reduction is of independent interest.

The DSF problem. We show that the non-linear pseudorandom generator G of the DSF assumption resists some of the above attacks as well. Specifically, its output is n^δ-wise independent and fools $\mathbf{AC^0}$ circuits and linear tests over GF(2). In fact, we prove a more general result about the security of the following construction of [Gol00]. For a sequence of m subsets of $[n]$, $S = S_1, \ldots, S_m$ of size $d = O(1)$ and a d-local predicate P, let $G_{S,P} : \{0,1\}^n \to \{0,1\}^m$ be the d-local mapping whose i-th output is obtained by applying the predicate P to the input string x restricted to the d indices of the set S_i. Goldreich [Gol00] conjectured that when the mapping is length preserving (i.e., $m = n$), the function $G_{S,P}$ is one-way for a random choice of the collection S and essentially any non-trivial predicate P. This assumption was supported by both theoretical and practical evidence [Gol00, Pan01, CEMT09]. Recently, [BQ09] showed that if the predicate P is biased towards a linear combination of two of its inputs, then the function becomes vulnerable when the output length m is sufficiently larger than the input length (i.e., $m > cn$ for a constant $c = c(d) > 1$).

[4] As an example, following this work, a variant of the DUE assumption was recently used by [ABBG10] (co-authored by the second author) to argue about the complexity of pricing financial derivatives. The DUE assumption also served as partial motivation for [BCC$^+$10]'s recent work on the densest subgraph problem.

We complement this by giving a combinatorial condition on S and P under which the function $G_{S,P}$ is pseudorandom with respect to the above family of non-trivial distinguishers (i.e., n^δ-wise independent tests, $\mathbf{AC^0}$ circuits and linear tests over GF(2)) even when m is polynomially larger than n.This suggests that the vulnerability discovered by [BQ09] only holds for a "bad" choice of the predicate P. (See Section 11 for details.)Our work also provides a new candidate for an $\mathbf{NC^0}$ pseudorandom generator with polynomial stretch (e.g., from n input bits to n^2 output bits). The existence of such a primitive is an important open question [CM01, MST03, AIK04, AIK04] which is also motivated by the ability to achieve highly efficient secure multiparty computation [IKOS08]. The only prior candidate (surviving a similar class of non-trivial attacks) was due to [MST03].

The DUE problem. We also show that the unbalanced expansion (DUE) problem resists "cycle counting" algorithms (a basic and surprisingly useful technique to identify dense subgraphs of a graph by counting the number of small cycles in the graph containing specific vertices [BCC$^+$10]). In addition we show that variants of the DUE assumption are implied by variants of other problems such as small-set vertex expansion in general (not necessarily bipartite) graphs, and the planted clique problem in $G_{n,p}$ for small $p = p(n)$. Finally, we prove that our third cryptosystem, which is based on a combination of DUE and DSF implies that a k-junta (i.e., a function $g : \{0,1\}^n \to \{0,1\}$ which depends in at most k of its variables) cannot be PAC-learned in less than $n^{\Omega(k)}$ time. The junta learning problem [Blu94, BL97] is one of the most central open problems in computational learning theory.[5]

2.3 Prior works

The notion of "structure" in complexity assumptions is necessarily informal but we can still offer below some comparisons of our schemes with previous ones. 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, Pei09]).

Our *non-linear* scheme (based on DSF and DUE) seems genuinely different from all previous constructions we are aware of. Our *linear* scheme (based on solely on 3LIN or dLIN and DUE) has some similarities to coding/lattice-based schemes but there are some important differences, which we now discuss.

Of the coding/lattice based schemes, the McEliece [McE78] system seems to use more algebraic structure, in the sense that the underlying assumption is that decoding a "shuffled" Goppa code is as hard as decoding a random linear code. A similar observation applies to the Hidden Field Equations (HFE) scheme of Patarin [Pat96] that implicitly assumes that a shuffled low degree univariate polynomial over GF(p^n) is indistinguishable from a random family of quadratic equations over GF(p)^n.

[5] In addition, the DSF assumption on its own can be formulated as a "dual" version of the junta learning problem in which the target function is not local, but instead the data points given to the learner are "local". More formally, in terms of learning theory, in DSF the learner should learn a function f_x, represented by an n-bit vector x, which maps a d-size set $S \subseteq [m]$ to the value $P(x_S)$ for some known (randomly chosen) predicate P.

The scheme $\mathcal{E}(\mathcal{D}, \varepsilon)$:

Public encryption key: A d-sparse matrix $M \in \mathbb{F}_2^{m \times n}$ sampled from distribution $\mathcal{D}$.

Private decryption key: A q-size subset $S \subseteq [m]$ of the rows of M for which $m \in S$ and $\sum_{i \in S} M_i = \mathbf{0}^n \pmod 2$ where M_i denotes the i-th row of M.

Encryption: Choose a random vector $x \stackrel{R}{\leftarrow} \mathcal{U}_n$ and a random noise vector $e \stackrel{R}{\leftarrow} \mathrm{Ber}_\varepsilon^m$, let $b = Mx + e$. To encrypt 0, send the vector b. To encrypt 1, send the vector b with its last bit flipped.

Decryption: To decrypt $y \in \{0,1\}^m$, output the sum $\sum_{i \in S} y_i \pmod 2$. Note that decryption is successful with probability at least $1 - q\varepsilon$.

Figure 1: Our basic cryptosystem scheme, used in the proofs of Thms. 2.2 and 2.3. m, d, q, ε can depend on the security parameter n. The distribution $\mathcal{D}$ is over matrices with d 1's per row, in which the last row is a linear combination of $q-1$ other rows. We show that under certain assumptions the scheme can be instantiated to achieve constant privacy. This can be amplified to full-fledged security using [HR05].

More similar to ours are the schemes of Alekhnovich [Ale03] and Regev [Reg05]. Regev's scheme (and others such as [PVW08, Pei09]) is based on the *Learning With Error* problem that is a $\mod p$ analog of the learning parity with noise problem. Specifically, like our 3LIN problem it is the task of recovering x from $(A, Ax + e)$ except A is a random (dense) $m \times n$ matrix in $\mathrm{GF}(p)$ for $p > m$, and each coordinate of e is distributed as a discrete Gaussian with standard deviation εp. However, as mentioned in Footnote 1, to make decryption work in all those schemes one needs to take $\varepsilon \ll 1/\sqrt{m}$ which seems to make a qualitative difference in the nature of the problem [GG98, AR04]. Most similar to ours is Alekhnovich's scheme [Ale03][6] that uses the (decisional variant) of the standard (dense) parity with noise problem. However, he too needed to use noise level of less than $1/\sqrt{m}$. While no analogous results to [GG98, AR04] are known for the mod 2 case, it still seems as an important advantage that we are able to handle much higher noise levels (in some cases at the expense of using DUE as well).

3 Overview of the proofs of Theorems 2.2 and 2.3

To highlight some of our techniques let us sketch the proofs of Thms. 2.2 and 2.3. We define $\mathsf{SearchLIN}(d, m, \varepsilon)$ to be the problem of recovering x given a random d-sparse $m \times n$ matrix M and the vector $b = Mx + e$, where x is chosen at random in $\mathrm{GF}(2)^n$ and $e \in \mathrm{GF}(2)^m$ is chosen at

[6] Indeed, as observed by Ron Rivest and Madhu Sudan (personal communication), both our linear 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 ℓ_2) basis.

random so that $e_i = 1$ with probability ε (we call e an ε-noise vector). Note that with extremely high probability this x to be recovered is unique. We let $\mathsf{Search3LIN}(m, \varepsilon) = \mathsf{SearchLIN}(3, m, \varepsilon)$.

PKE from $\mathsf{3LIN}(\Omega(n^{1.4}), O(n^{-0.2}))$. The proof proceeds by a sequence of reductions, ending with showing that under our hardness assumption on the *search* problem, a related *prediction* problem is hard as well. This prediction problem gets essentially the same input, a matrix M and a vector $b = Mx + e$ *except its last bit*, and asks to predict that bit. In other words, given the value of $m-1$ noisy (3-sparse) equations, we are asked to predict the value of another independent equation. A natural way to predict is to solve the search problem, and use the recovered solution x to evaluate the new equation (which will predict it correctly with probability $1 - \epsilon$). Our reduction shows that essentially this is the only way to go about prediction. If search is hard, so is prediction, even if all we ask for is a constant advantage (say $1/10$) over guessing.

The twist is that the distribution of sparse matrices we use in this reduction is special. Formally, for any distribution $\mathcal{D}$ on $(m, n, 3)$-matrices, define the following *prediction* problem $\mathsf{Predict3LIN}(\mathcal{D}, \varepsilon)$: given M drawn from $\mathcal{D}$ and the first $m-1$ bits of $Mx + e$ where x, e as above, predict the m'th bit of $Mx + e$ with probability at least $3/5$. We will reduce the search problem $\mathsf{Search3LIN}(m, \varepsilon)$ (where matrices are drawn uniformly) to the prediction problem $\mathsf{Predict3LIN}(\mathcal{D}_0, \varepsilon)$, in which matrices are drawn from a special distribution $\mathcal{D}_0$.

Our cryptosystem. Before explaining the reduction, let us explain how the prediction problem $\mathsf{Predict3LIN}(\mathcal{D}_0, \varepsilon)$ can be turned into a public-key system. This system is also described in Figure 1. The distribution $\mathcal{D}_0$ has the property that if M is in the support of $\mathcal{D}_0$, then there is a linear relation involving M's last row and at most $q \ll 1/\varepsilon$ other rows. Moreover, it is possible to efficiently sample a random matrix M from $\mathcal{D}_0$ together with such a set S of rows involved in this linear relation. Since $q\varepsilon$ is small, if we add an ε-noise vector e to Mx, then with high probability no equation in S will be noisy, which means that given the value of $Mx + e$ on the coordinates in S, one can recover the value of Mx on the m^{th} coordinate. Thus, the linear relation can serve as a sort of "trapdoor" for the $\mathsf{Predict3LIN}(\mathcal{D}_0, \varepsilon)$ problem. One can turn this observation into a PKE by using relatively standard techniques such as hardness amplification [HR05].

Search to approximate search. To get from $\mathsf{Search3LIN}(m, \varepsilon)$ to $\mathsf{Predict3LIN}(\mathcal{D}_0, \varepsilon)$ we use a chain of three reductions through two intermediate problems. The first is an "approximate search" problem $\mathsf{AppSearch3LIN}(m, \varepsilon)$, which is the variant of $\mathsf{Search3LIN}$ in which the goal is relaxed to only recover a vector x' that is *close* to the true answer x in Hamming distance. We use error correcting properties of sparse equations to show that the two problems are equivalent up to constant loss in the parameters. In essence, we can use $O(n \lg n)$ more noisy equations to detect the "errors" in the approximate solution vector x' and correct them to recover x. (See Theorem 6.2.)

Search to prediction on the uniform distribution. The second intermediate problem is $\mathsf{Predict3LIN}(m, \varepsilon)$ which is the problem $\mathsf{Predict3LIN}(\mathcal{D}, \varepsilon)$ where $\mathcal{D}$ is the uniform distribution over $(m, n, 3)$-matrices - the *same* distribution on matrices used in $\mathsf{Search3LIN}(m, \varepsilon)$ and $\mathsf{AppSearch3LIN}(m, \varepsilon)$. We reduce $\mathsf{AppSearch3LIN}(m, \varepsilon)$ to $\mathsf{Predict3LIN}(m + O(n), \varepsilon)$. A key observation used in the proof is that by adding two 3-sparse random equations that share a common variable, we get a random 4-sparse equation of the form $x_i + x_j + x_k + x_\ell = b$, and so given such an equation one can turn a

predictor for $x_i + x_j + x_k$ into a predictor to x_ℓ. By carefully combining many pairs of equations it can be shown that at the end, we will get predictions for a large fraction of the variables, and that most of these predictions will be correct. Hence, together they form a good approximation for x. (See Theorem 6.5.)

Prediction on a different distribution. The last step (obtained in Theorem 6.9) is a reduction between the two prediction problems $\mathsf{Predict3LIN}(m, \varepsilon)$ to $\mathsf{Predict3LIN}(\mathcal{D}_0, \varepsilon)$ where $\mathcal{D}_0$ is the special distribution above. This step is composed of two stages. First we use the results of Feige, Kim, and Ofek [FKO06] to argue that small linear relations involving the last row of M will exist in our setting of parameters with some (small) constant probability for the uniform distribution. Therefore the statistical distance between $\mathcal{D}_0$ (in which such a relation is sampled first) and the uniform distribution is bounded away from 1. We complete the proof by showing how to turn a good predictor for $\mathsf{Predict3LIN}(\mathcal{D}, \varepsilon)$ into a good predictor A for $\mathsf{Predict3LIN}(\mathcal{D}', \varepsilon)$ for every two distributions $\mathcal{D}, \mathcal{D}'$ over matrices with related parameters whose statistical (or computational!) distance is bounded away from 1. This differs from most proofs of this type, since we want the difference in prediction probability of the two predictors to be much smaller than the statistical (or computationsl) distance of the two distributions! For example, even if A perfectly solves $\mathsf{Predict3LIN}(\mathcal{D}, \varepsilon)$ with no error, it might be a terrible predictor which errs with probability $1/2$ when instances are generated according to $\mathcal{D}'$. Still, we show how to turn it into a useful predictor with respect to $\mathcal{D}'$ as well. The idea is to identify (via sampling) the instances on which A is likely to succeed and use it *only* for these cases. Then, we amplify the success probability by breaking a single instance of $\mathsf{Predict3LIN}$ to many smaller instances of the problem. These instances are rerandomized by relying on the symmetry and linearity of the 3-LIN constraints.

Thm. 2.3: PKE from DUE and dLIN. The description above completes the reduction of Thm. 2.2. For Thm. 2.3, in which smaller values of m are used, such small linear relations between rows of M will *not* exist, and hence the distribution $\mathcal{D}_0$ as above will be statistically far from the uniform distribution on d-sparse matrices. Here our extra assumption DUE comes to the rescue, basically to prove that *computationally* its distance from uniform will be bounded away from 1. The next two paragraphs highlight the main ideas in that proof.

The use of DUE, as well as the extension to large sparsity $d > 3$ introduce some additional difficulties that we need to overcome. In particular, for our cryptosystem we need DUE to hold even if one of the members of the planted shrinking set is revealed. Hence, to prove security we show that solving this variant of DUE assumption (denoted by DUE_1) implies a solution to the original DUE problem.

In particular, given a random (m, n, d) graph with a planted shrinking set, an algorithm for DUE_1 can be used to distinguish with some constant advantage between nodes that participate in the shrinking set to other nodes. This distinguisher allows us to "shave" many nodes of the large side of the graph while preserving the existence of a (smaller) shrinking set. The resulting graph will have m' top nodes and n bottom nodes where $m' < n$. (Recall that we started with $m > n$ top nodes.) For this case, we can detect the existence of a shrinking set by using Hall's theorem via a matching based algorithm. This leads directly to a solution for DUE. Note that this argument shows only that, under the DUE assumption, the distribution $\mathcal{D}_0$ is not completely computationally-far from the uniform distribution. Here again, we need to rely on the strong version of the reduction from $\mathsf{PredictLIN}(\mathcal{D}, \varepsilon)$ to $\mathsf{PredictLIN}(\mathcal{D}', \varepsilon)$.

Another difficulty arises from the use of a large sparsity $d > 3$, as in this case the combination of two equations with overlap of one variable does not lead to an equation of sparsity $d + 1$ as in the $d = 3$ case. We overcome this problem by employing a different reduction from PredictLIN to AppSearchLIN. Specifically, given an instance of AppSearchLIN with locality d, we combine pairs of equations with no overlap to obtain a $2d$-LIN instance. Then, we generate $(2d-2)$-LIN equations by combining pairs of equations with a common variable. This information, together with a prediction algorithm for $2d$-LIN can be used to obtain a 2-LIN equation. By repeating this process we obtain a random 2-LIN (or MAX-CUT) instance. Now we can employ one of the known algorithms (e.g.,the SDP of [GW95]) to obtain a solution that satisfies a large fraction of the constraints. Finally, we argue that since the 2-LIN instance is random the resulting assignment is close to the original assignment and therefore it is a valid solution for AppSearchLIN.

The proof of Theorem 2.1 (a cryptosystem based on DUE and DSF) follows a roughly similar high level structure, and we omit details from this overview.

Part I
Cryptosystems

4 Preliminaries

d-sparse matrices. A matrix $M \in \mathbb{F}_2^{m \times n}$ is d-sparse if each of its rows contains exactly d ones. Such a matrix can also represent an (m, n, d) graph, which is a bipartite graph $G_M = ((V_{\mathsf{Top}}, V_{\mathsf{Bot}}), E)$ with m "top" nodes (each node correspond to a row) and n "bottom" nodes (each node correspond to a column) where each top node has degree d. We consider the uniform distribution $\mathcal{M}_{m,n,d}$ over d-sparse matrices in which each of the m rows of the matrix is chosen independently and uniformly at random from the set of all n-bit vectors of weight d. We will be especially interested in 3-sparse matrices, hence, we abbreviate $\mathcal{M}_{m,n,3}$ by $\mathcal{M}_{m,n}$.

d-LIN. A d-LIN instance with m-clauses and n variables is described by an m-bit vector b and a d-sparse matrix $M \in \mathbb{F}_2^{m \times n}$. If there exists an assignment x for which the weight of $Mx - b$ is at most εm then the problem is $(1 - \varepsilon)$-satisfiable. Let $\mathrm{Ber}_\varepsilon^m$ be the distribution of an m-bit vectors that each of its entries is 1 with probability ε independently of the others. A natural way to generate "almost satisfiable" d-LIN instances is to choose a matrix M from some distribution over d-sparse matrices (e.g., $\mathcal{M}_{m,n,d}$) and let $b = Mx + e$ where $x \xleftarrow{R} \mathcal{U}_n$ and $e \xleftarrow{R} \mathrm{Ber}_\varepsilon^m$; that is, b is a random vector of (expected) distance ε from $\mathrm{Im}(M)$.

Statistical distance and computational indistinguishability. 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 two sequences of distributions $\mathcal{X}_n, \mathcal{Y}_n$ (where n is an implicit or explicit security parameter) are ε-indistinguishable if for every probabilistic polynomial time algorithm C, we have $|\Pr[C(\mathcal{X}) = 1] - \Pr[C(\mathcal{Y}) = 1]| < \varepsilon$. A sequence of distributions $\mathcal{X}_n$ is ε-pseudorandom if $\mathcal{X}_n$ is ε-indistinguishable from $\mathcal{U}_n$, the uniform distribution over n bits.

Public-key encryption schemes. A public-key encryption scheme (PKE) allows two parties to communicate securely without sharing a secret key. We follow [HR05] and quantify the correctness and privacy of the scheme by two error parameters α and β.[7]. The definition becomes equivalent to the standard notion of semantic security [GM82] when both parameters are taken to be negligible, i.e., when α and β go down to zero faster than any inverse polynomial.

Definition 4.1. *A $(\alpha(n), \beta(n))$-secure public-key bit encryption scheme is a triple $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of probabilistic polynomial time algorithms such that*

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

If $\alpha(n)$ and $\beta(n)$ are constants that satisfy $\alpha < (1 - \sqrt{\beta})/2$, we say that the scheme is a *weak PKE*. It was shown in [HR05, Thm. 6] that a weak PKE can be converted into semantically secure PKE which supports arbitrary (polynomially) long messages.

5 PKE based on the Hardness of 3-LIN

5.1 The Search3LIN Problem

Definition 5.1. *The $\mathsf{Search3LIN}(m, \varepsilon)$ problem is defined as follows:*

- *Input: a random ε-satisfiable 3-LIN instance (M, b) sampled as follows: $M \stackrel{R}{\leftarrow} \mathcal{M}_{m,n}$ and $b = Mx + e$ where $x \stackrel{R}{\leftarrow} \mathcal{U}_n$ and $e \stackrel{R}{\leftarrow} \mathrm{Ber}_{\varepsilon}^{m}$.*
- *Output: the assignment x.*[8]

Let $m = m(n)$ and $\varepsilon = \varepsilon(n)$ be some functions of n. We say that $\mathsf{Search3LIN}(m, \varepsilon)$ is intractable *if for every probabilistic polynomial-time algorithm A, and every sufficiently large n, A solves $\mathsf{Search3LIN}(m(n), \varepsilon(n))$ with probability smaller than $2/3$.*

It is not hard to show that the problem becomes harder when m is decreased and ε is increased. Also, one can replace the constant $2/3$ by any arbitrary positive constant at the expense of increasing m by a multiplicative constant.

We will construct a public-key encryption scheme based on the following assumption:

Assumption 5.2. *The problem $\mathsf{Search3LIN}(C_0 n^{1.4}, C_1 n^{-0.2})$ is intractable for every constants $C_0 > 0$ and $C_1 > 0$.*[9]

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

[8] Our setting of the parameters m and ε will guarantee a unique solution with probability $1 - o(1)$.

[9] In fact, all our applications require a relaxed version of the assumption in which $\mathsf{Search3LIN}(C_0 n^{1.4}, C_1 n^{-0.2})$ is intractable for some fixed universal constants that can be explicitly calculated from our reductions.

While we define the notion of intractability with respect to polynomial time algorithms, one may hope the assumption holds even for subexponential algorithms with time complexity 2^{n^δ} for some constant $\delta > 0$. We now give some evidence for Assumption 5.2.

Unconditional resistance to limited attacks. Our assumption asserts that Search3LIN with $m = O(n^{1.4})$ equations and $\varepsilon = \Omega(n^{-0.2})$ noise rate is hard for all polynomial-time algorithms. This hardness can be proven (unconditionally) for several restricted computational models as follows. First, it is not hard to show that the distribution of the vector b looks random to an adversary that looks at no more than $k = n^\delta$ coordinates, for some fixed $0 < \delta < 1$. Hence, such a "myopic" adversary cannot solve the problem. Furthermore, a recent breakthrough of Braverman [Bra09] shows that k-wise independent distributions ε-fools the class of all $\mathbf{AC^0}$ circuits of sub-exponential size. Finally, the recent work of Schoenebeck [Sch08] shows that our assumption cannot be refuted by an expressive family of semidefinite programs which results from n^δ levels of the Lasserre hierarchy [Las01]. (The time complexity of these programs is exponential in the number of levels, and so in our case the lower bound corresponds to programs of subexponential time.) This result is especially interesting as semidefinite programs seem to be the strongest algorithmic tool that is currently available to attack constraint satisfaction problems. Moreover, the Lasserre hierarchy is strictly stronger than other hierarchies for linear and semidefinite programs such as the Lovasz-Schrijver [LS91] hierarchies (LS, and LS+) and the Sherali-Adams [SA90] hierarchy [Lau03]. (See Section 9 for formal statements of all these results.)

Relation to other variants of the problem. We can define other natural variants of the Search3LIN problem. First, consider the distinguishing version Decide3LIN(m, ε) in which the goal is to distinguish between a Yes-instance that comes from the distribution of Search3LIN(m, ε) from a No-instance in which $M \stackrel{R}{\leftarrow} \mathcal{M}_{m,n}$ and the vector b is uniformly chosen. An algorithm that solves the search problem Search3LIN(m, ε) with probability δ can be directly used to solve the distinguishing problem Decide3LIN(m, ε) with similar advantage. Hence, the intractability of Decide3LIN$(O(n^{1.4}), \Omega(n^{-0.2}))$ implies Assumption 5.2. In fact, all the lower bounds mentioned above hold against the decision problem as well. In addition, several previous works [Ale03, MST03, AIK06] assumed that the Decide3LIN(m, ε) problem, or close variants of it, is intractable for every $m = O(n)$ and constant $0 < \varepsilon < 1/2$. It is also natural to consider the refutation version of 3-LIN Refute3LIN(m, ε) in which one has to distinguish between Yes and No instances with zero error over Yes instances. (This can be seen as certifying an instance for not being ε-satisfiable.) The work of Feige [Fei02] establishes a connection between the refutation problem for 3-LIN and the refutation of 3-SAT formulas. In particular, it can be shown that an efficient solution to Refute3LIN$(O(n^{1.4}), \Omega(n^{-0.2}))$ allows to refute 3-SAT with $m = O(n^{1.4})$ clauses. Such a refutation algorithm for 3-SAT would resolve an important open question as the best algorithm for the problem works only for $m = n^{1.5}$, or more generally, for $m = n^{d/2}$ in the case of d-SAT [FGK05]. (We mention that a non-deterministic algorithm is known for $m = n^{1.4}$ [FKO06].) While we do not know of any formal connection between the hardness of Refute3LIN and the hardness of Search3LIN, it seems that, at least in an intuitive level, the intractability of one supports the intractability of the other.

5.2 A Public-Key Encryption Scheme

We will rely on the general bit-encryption scheme which is described in Figure 1. The public key is matrix M, and the private-key is a short non-trivial linear dependency S among the rows of M which includes the last row. To encrypt the bit σ, one generates an m-bit vector b by perturbing a random vector in the image of M, and then XOR-s the last entry of b with the plaintext σ. The knowledge of the short linear-dependency S allows to decrypt the ciphertext b' by summing-up the bits that are indexed by the set S. This algorithm works well as long as the noise rate ε is sufficiently smaller than $1/|S|$. Of course, one should instantiate the scheme with a key-generation algorithm that describes exactly how to sample the matrix M together with the short linear dependency S. To prove that the scheme is secure, it should be shown that it is hard to predict the last entry of the ciphertext b.

Before we instantiate the scheme we can establish its correctness.

Lemma 5.3. *For every pair (M, S) of public/private keys, and every plaintext $\sigma \in \{0,1\}$, the decryption errs with probability at most $\alpha = \frac{1}{2} - \frac{1}{2}(1-2\cdot\varepsilon)^q < \varepsilon q$, where the probability is taken over the randomness of the encryption algorithm.*

Proof. Let $x \xleftarrow{R} \mathcal{U}_n$ and $e \xleftarrow{R} \mathrm{Ber}_\varepsilon^m$ be the random vectors used by the encryption algorithm. The decryption algorithm outputs the value

$$\sum_{i\in S}(Mx)_i + e_i + \sigma = \sum_{i\in S} e_i + \sigma,$$

where the equality follows from the degeneracy of M_S. Hence, the decryption errs whenever the sum of the noise bits indexed by S is 1, which happens with probability at most $\frac{1}{2} - \frac{1}{2}(1-2\cdot\varepsilon)^q < \varepsilon q$. □

Instantiating the key generation algorithm. Let $H_{q,n}^{2,3}$ be the uniform distribution over matrices with n columns and q rows, where each row contains exactly 3 ones and each column contains exactly 2 ones. Let $\mathcal{T}_{p,n}$ be denote a distribution over 3-sparse matrices with n columns in which each possible row is selected with probability p. Consider the distribution $\mathcal{T}_{p,n,q}$ which is simply $\mathcal{T}_{p,n}$ conditioned on the event that the matrix T contains a submatrix of rows $H \in \mathrm{support}(H_{q,n}^{2,3})$ that includes the last row of T. Since $H_{q,n}^{2,3}$ is efficiently samplable (via standard techniques), it is possible to sample a triple (M, H, S) where $M \xleftarrow{R} \mathcal{T}_{p,n,q}$, $H \xleftarrow{R} H_{q,n}^{2,3}$ and $M_S = H$ (i.e., S is a q-size subset of the rows of M which points to the submatrix H). We will use this distribution for our key-generation algorithm, with the parameters $p = n^{-1.6}$, and $q = \Theta(n^{0.2})$ and set the noise rate $\varepsilon \ll 1/q$ to $\Theta(n^{0.2})$. We show that the resulting scheme provides a non-trivial level of security and correctness, and therefore can be later converted into a semantically secure cryptosystem.

First, we argue that the distribution $\mathcal{T}_{p,n,q}$ is not too far (in statistical distance) from $\mathcal{T}_{p,n}$.

Lemma 5.4. *There exists a function $q(n) = \Theta(n^{0.2})$, such that the statistical distance between $\mathcal{T}_{n^{-1.6},n}$ and $\mathcal{T}_{n^{-1.6},n,q}$ is at most $1-\delta$, for some absolute constant $0 < \delta < 1$ which does* not *depend on n.*

Proof. Let E denote the event in which a matrix $T \xleftarrow{R} \mathcal{T}_{n^{-1.6},n}$ contains a submatrix $H \in \mathrm{support}(H_{q,n}^{2,3})$ that includes the last row of T. By the definition of $\mathcal{T}_{n^{-1.6},n,q}$, it suffices to show that E happens with probability δ for some constant δ. Consider the event $F_{\alpha,\beta,\gamma,\theta}$ in which T contains at least $\alpha n^{1.4}$

copies of submatrices from support($H^{2,3}_{\beta n^{0.2},n}$) such that each row of T participates in at most $\gamma n^{0.2}$ distinct copies. Furthermore T has at most $\theta n^{1.4}$ rows. Feige, Kim and Ofek [FKO06, App. B] showed that for some constants $\alpha, \beta, \gamma, \theta > 0$ the event $F = F_{\alpha,\beta,\gamma,\theta}$ happens with probability $1 - o(1)$. Hence, it suffices to show that conditioned on the event F, the event E happens with constant probability $\delta' = \delta'(\alpha, \beta, \gamma, \theta)$. Indeed, construct a bipartite graph in which the rows of T are on the left side and the submatrices of support($H^{2,3}_{\beta n^{0.2},n}$) are on the other side, and connect a submatrix node to all its rows. Now, by counting edges, it follows that the average degree of the right side (row side) is at least $\alpha\beta n^{1.6}$. However, the maximal degree of a row node is $\gamma n^{0.2}$ and there are at most $\theta n^{1.4}$ such nodes. It follows, by a Markov inequality, that at least a fraction of $\delta' = \alpha\beta/(\gamma\theta)$ of the rows in T have positive degree, i.e., they participate in a submatrix from support($H^{2,3}_{\beta n^{0.2},n}$). By symmetry, it follows that the last row participates in such a submatrix with probability δ'. □

Let $\mathcal{E}(\mathcal{D}, \varepsilon)$ be the scheme of of Figure 1 instantiated with noise $\varepsilon = \varepsilon(n)$ and public-key taken from some distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}$. Our main theorem shows that if $\mathcal{D}$ is not too far from $\mathcal{T}_{m/\binom{n}{3},n}$, and Search3LIN$(Cm, \varepsilon)$ is intractable for sufficiently large C, then $\mathcal{E}(\mathcal{D}, \varepsilon)$ is weakly secure.

Theorem 5.5 (main theorem)**.** *For every constant $0 < \delta < 1$, there exists a constant $C = C(\delta)$, such that for every function $m(n) = \Omega(n \lg n)$, every $\varepsilon = \varepsilon(n) \leq 0.01$, and every distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}$ which is $(1-\delta)$-computationally indistinguishable from $\mathcal{T}_{m/\binom{n}{3},n}$ the following holds. If* Search3LIN(Cm, ε) *is intractable then the public-key encryption scheme $\mathcal{E}(\mathcal{D}, \varepsilon)$ is $(1-\delta/2)$-private.*

The theorem transforms the intractability of Search3LIN which is a search problem over the uniform distribution, into indistinguishability (with respect to any related distribution) by going through several intermediate notions (e.g., approximate search and unpredictability with respect to the uniform distribution). Section 6 is devoted to the proof of Theorem 5.5.

By choosing the parameters q and ε properly, we can now obtain a weak public-key bit encryption scheme and thus derive the following corollary.

Corollary 5.6. *Under Assumption 5.2, there exists a semantically-secure public-key encryption scheme.*

Proof. Let $p = n^{-1.6}$. Let $q(n), \delta$ and $C = C(\delta)$ be the parameters obtained from Lemma 5.4 and Theorem 5.5. By combining Lemma 5.4, Theorem 5.5 and Assumption 5.2, we get that the scheme the scheme $\mathcal{E}(\mathcal{T}_{p=n^{-1.6},n,q}, \varepsilon)$ is $\beta = (1 - \delta/2)$-private scheme for any choice of $\varepsilon = \Omega(n^{-0.2})$. By letting $\varepsilon = C_1 n^{-0.2}$ for sufficiently small constant $C_1 > 0$, we can make the decryption error α of Lemma 5.3 arbitrarily close to 0. In particular, we can make sure that α is smaller than $(1 - \sqrt{\beta})/2$ as $\beta = 1 - \delta/2$ is bounded away from 1. Hence, we obtain a weak PKE which can be converted to a semantically secure PKE via the transformation of [HR05]. □

Remark 5.7 (*Oblivious Transfer from Assumption 5.2*)**.** *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, [EGL85] 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, as well all other schemes presented in this paper, satisfy this additional property and therefore it implies the existence of an OT-protocol.*

6 Proof of Theorem 5.5

Roadmap. Theorem 5.5 will be proved in three steps. In Section 6.1 we will show that the exact search problem Search3LIN reduces to an approximate search version of the problem AppSearch3LIN in which the goal is to find an approximate solution $\hat{x}$ which is close to the exact solution x. Then, in Section 6.2, we show that AppSearch3LIN reduces to a prediction version of the problem Predict3LIN, in which the goal is to predict the last bit of a random 3-LIN instance. Finally, in Section 6.3 we relate the hardness of Predict3LIN with respect to the uniform distribution to the hardness of Predict3LIN with respect to other distributions.

6.1 From Approximate Search to Search

We relate the search problem to the following approximate-search version.

Definition 6.1. *The* AppSearch3LIN(m, ε) *problem is defined as follows:*

- *Input: a random* ε*-satisfiable 3-Lin instance* (M, b) *sampled as follows:* $M \stackrel{R}{\leftarrow} \mathcal{M}_{m,n}$ *and* $b = Mx + e$ *where* $x \stackrel{R}{\leftarrow} \mathcal{U}_n$ *and* $e \stackrel{R}{\leftarrow} \mathrm{Ber}_{\varepsilon}^{m}$*.*
- *Output: an assignment* $\hat{x}$ *which is 0.1-close to* x*.*

Let $m = m(n)$ *and* $\varepsilon = \varepsilon(n)$ *be some functions of* n*. We say that* AppSearch3LIN(m, ε) *is* intractable *if for every probabilistic polynomial-time algorithm* A*, and every sufficiently large* n*,* A *solves* AppSearch3LIN$(m(n), \varepsilon(n))$ *with probability smaller than* 0.7*.*

Theorem 6.2. *Suppose that* Search3LIN$(m + t, \varepsilon)$ *is intractable where* $\varepsilon \leq 1/4$ *and* $t \geq Kn \ln n$ *for some universal constant* K*. Then,* AppSearch3LIN(m, ε) *is also intractable.*

Remark. The theorem holds for general constant sparsity d and general constant $\varepsilon < 1/2$ at the expense of taking t to be $O(m + n \ln n)$ where the constant in the O notation depends in both ε and d. Note that the overhead t is "swallowed" in a constant multiplicative overhead as long as $m = \Omega(n \lg n)$. This is essentially optimal and we cannot hope for a linear overhead $t = O(n)$ when $m = O(n)$ since Search3LIN$(O(n), \varepsilon)$ is information-theoretically hard. (There is not enough information to recover x as about ε^d fraction of the bits of x are expected to be involved only in noisy equations.)

To prove the theorem we show how to convert an algorithm A that solves AppSearch3LIN(m, ε) with probability 0.7 for infinitely many n's, into an algorithm C that solves Search3LIN$(m + t, \varepsilon)$ for the same input lengths. Given an input $(M, b) \in \mathbb{F}_2^{(m+t)\times n} \times \mathbb{F}_2^{m+t}$, the algorithm C does the following. At the first step C invokes the approximation algorithm A on the first m rows of the input, and gets an approximation $\hat{x}$ for x. Then, at the second step, C uses the information given

by the last t rows of the input to correct the errors in $\hat{x}$. Specifically, given the additional list of equations $(T, v) \in \mathbb{F}_2^{t \times n} \times \mathbb{F}_2^t$, we will recover the i-th bit of x by letting each equation of the form $x_i + x_k + x_\ell = v_s$ to vote for the correct value of x_i. This vote is simply $\hat{x}_k + \hat{x}_\ell + v_s$, i.e., we compute the value of x_i assuming that v_s is not noisy and that the approximation for $\hat{x}_k$ and $\hat{x}_\ell$ is correct. Finally, we take the majority of all votes.

The following lemma, analyzes the probability that C guesses the right value of x_i.

Lemma 6.3. *Fix $i \in [n]$ and let K be a sufficiently large constant. Conditioned on $\hat{x}$ being an 0.1-close to x, the algorithm C outputs x_i with probability at least $1 - o(1/n)$, where the probability is taken over the randomness of the last $t = Kn \ln n$ rows of the input, i.e., over the choice of $T \stackrel{R}{\leftarrow} \mathcal{M}_{t,n}$ and over the noise vector $e' \stackrel{R}{\leftarrow} \mathrm{Ber}_\varepsilon^t$ which consists of the last t entries of the original $m + t$-long noise vector.*

The above lemma, together with a union bound, implies that, conditioned on $\hat{x}$ being a good approximation, the algorithm C recovers x with probability larger than $1 - o(1)$. Hence, the overall probability success is at least $0.7 - o(1) > 2/3$ for infinitely many n's, and Theorem 6.2 follows.

6.1.1 Proof of Lemma 6.3

Fix x and $\hat{x}$ and let S be the set of indices on which x and $\hat{x}$ do not agree. By our assumption the size of S is bounded by $0.1n$. We will show that the index i is likely to participate in many equations and that w.h.p most of these equations will vote for the correct value x_i.

Let Q be the set of rows in T in which i participates. Let $B \subseteq Q$ be the set of rows in which i participates and, in addition, at least one of the other entries of the row is in S. We claim that,

$$\Pr_{T \stackrel{R}{\leftarrow} \mathcal{M}_{t,n}} [(|Q| > 0.5 \cdot t/n) \bigwedge (|B| < 0.3|Q|)] > 1 - 1/n^{\Omega(K)}. \tag{1}$$

Indeed, think of T as chosen by the following random process: for each row we first determine whether it consists of the index i by tossing a coin whose success probability is $\alpha = \binom{n}{2}/\binom{n}{3} > 1/n$; then for those rows in which i appears we randomly choose two additional distinct random indices from $[n] \setminus \{i\}$; finally, for those rows in which i does not appear, choose three distinct indices uniformly at random from the set $[n] \setminus \{i\}$. Hence, the random variable $|Q|$ can be written as the sum of t independent Bernoulli trials each with success probability α. Similarly, the random variable $|B|$ can be written as the sum of $|Q|$ independent Bernoulli trials each with success probability $\beta < \frac{2|S|}{n} < 0.2$. Hence, Eq. 1 follows from a multiplicative Chernoff bound.

Fix a matrix T for which the event $(|Q| > 0.5K \ln n) \wedge (|B| < 0.3|Q|)$ holds. We claim that in this case the algorithm C recovers x_i with probability at least $1 - 1/n^{\Omega(K)}$, where the probability is now taken over the choice of the error vector e'. Let χ_j be an indicator variable which equals to one if the j-th equation outputs a good vote for x_i. If j is in $\overline{B} = Q \setminus B$, then χ_j is one whenever the corresponding error bit e'_j is zero which happens with probability $1 - \varepsilon$. On the other hand, if $j \in B$ then χ_j equals to one with probability at least ε. Also, the χ_j's are independently distributed. It follows that the probability of getting a majority of correct votes increases when $|B|$ decreases and when ε decreases. Hence, it suffices to consider the case where $|B| = 0.3|Q|$ and $\varepsilon = 1/4$. Let $\chi_B = \sum_{i \in B} \chi_i$ and $\chi_{\overline{B}} = \sum_{i \in \overline{B}} \chi_i$. Note that the expected value of χ_B is $\varepsilon|B| = 0.075|Q|$ and the expected value of $\chi_{\overline{B}}$ is $(1 - \varepsilon)|\overline{B}| = 0.525|Q|$. Therefore, the overall sum of the χ_j's is larger than $|Q|/2$ as long as both χ_B and $\chi_{\overline{B}}$ do not deviate too much from their expectation, which by a

Chernoff bound, happens with probability $1 - 1/n^{\Omega(K)}$. Formally, we can lower bound the success probability $\Pr[\sum_{i \in Q} \chi > 0.5|Q|]$ by

$$\Pr[(\chi_B > 0.9\varepsilon|B|) \wedge (\chi_{\overline{B}} > 0.9(1-\varepsilon)|\overline{B}|)] \quad \geq \quad 1 - 1/n^{\Omega(K)} \tag{2}$$

where the inequality follows by applying a union bound and a Chernoff bound. The Lemma now follows by combining Eq. 1 and 2 via a union bound, and by choosing a sufficiently large constant K. □

6.2 From Prediction to Approximate Search

We relate the Approximate Search problem to the following prediction version. We let $\mathcal{D} = \{\mathcal{D}_n\}$ be a sequence of distributions where $\mathcal{D}_n$ is distributed over 3-sparse matrices with n columns and a polynomial number of rows.

Definition 6.4. *The* Predict3LIN$(\mathcal{D}, \varepsilon)$ *problem is defined as follows:*

- *Input: a 3-weight vector v together with an ε-satisfiable 3-LIN instance (M, b) sampled as follows: $\binom{M}{v} \stackrel{R}{\leftarrow} \mathcal{D}_n$ and $b = Mx + e$ where $x \stackrel{R}{\leftarrow} \mathcal{U}_n$ and $e \stackrel{R}{\leftarrow} \mathrm{Ber}_\varepsilon^m$.*
- *Output: the inner product $\langle x, v \rangle = \sum_i x_i \cdot v_i \pmod 2$.*

We say that Predict3LIN$(\mathcal{D}, \varepsilon)$ *is* δ-intractable *if for every probabilistic polynomial-time algorithm A, and every sufficiently large n, A solves* AppSearch3LIN$(\mathcal{D}_n, \varepsilon(n))$ *with probability smaller than $\delta(n)$. By default we take $\delta = 0.99$. In the special case where $\mathcal{D}_n$ is taken to be the uniform distribution $\mathcal{M}_{m,n}$ for $m = m(n) > n$ we use the abbreviation* Predict3LIN(m, ε).

Theorem 6.5. *Let $\varepsilon \leq 0.01$. If* AppSearch3LIN$(m + 25n, \varepsilon)$ *is intractable then* Predict3LIN(m, ε) *is* 0.99*-intractable.*

Before we introduce the reduction we will need the following lemma which transforms a random 3-LIN instance to a random 4-LIN instance.

Lemma 6.6. *There exists an efficient algorithm B such that given an input pair (T, b) outputs a pair (R, v) such that for every $x \in \{0,1\}^n$, if the input distribution is*

$$(T, b = Tx + e), \text{ where } T \stackrel{R}{\leftarrow} \mathcal{M}_{4t+n,n}, e \stackrel{R}{\leftarrow} \mathrm{Ber}_\varepsilon^{4t+n},$$

then the output distribution is $\exp(-t/4)$-close (in statistical distance) to

$$(R, Rx + e'), \text{ where } R \stackrel{R}{\leftarrow} \mathcal{M}_{t,n,4}, e' \stackrel{R}{\leftarrow} \mathrm{Ber}_{2\varepsilon(1-\varepsilon)}^t. \tag{3}$$

Note that B does *not* know x. To prove Lemma 6.6 we view (T, b) as a system of 3-LIN equations in x and partition them into pairs of equations that share a single variable x_i, each pair can be combined into a 4-LIN equation by simple addition. Furthermore, if the partition is done carefully, then the resulting 4-LIN system is uniformly distributed. See Section A for a full proof of the lemma.

We will prove Theorem 6.5 by showing how to convert an algorithm A that solves Predict3LIN(m, ε) with probability 0.99 for infinitely many n's, into an algorithm C that solves AppSearch3LIN$(m + 25n, \varepsilon)$ for the same input lengths.

Algorithm $C(M, b)$.

1. Partition M (resp. b) into two parts S and T (resp. w and z) where S (resp. w) consists of the first m rows and T (resp. z) the remaining $25n$ rows.

2. Use the algorithm B of Lemma 6.6 to transform (T, z) to a a pair (R, y) where each row of $R \in \mathbb{F}_2^{t \times n}$ has weight 4 and $y \in \{0,1\}^t$, where $t = 6n$.

3. Let $\hat{x} = 0^n$. For $j = 1, \ldots, t$ do the following:

   (a) Let r_j be the j-th row of R. Choose a random index i_j from the support of r_j and define a 3-weight vector $u_j = r_j \setminus \{i_j\}$. If i_j was already chosen in one of the previous iterations than ignore the current iteration. Invoke A on (S, w, u_j) and record the result in σ_j. Set $\hat{x}_{i_j}$ to be $y_j + \sigma_j$.

4. Output $\hat{x}$.

Theorem 6.5 follows from the following lemma.

Lemma 6.7. *For those input lengths on which A solves* $\mathsf{Predict3LIN}(m, \varepsilon)$ *with probability* 0.99*, the probability that $C(M, Mx + e)$ outputs an assignment $\hat{x}$ which is 0.1-close to x is larger than* $0.8 - o(1)$*, where* $M \xleftarrow{R} \mathcal{M}_{m,n}, x \xleftarrow{R} \mathcal{U}_n, e \xleftarrow{R} \mathrm{Ber}_\varepsilon^m$.

6.2.1 Proof of Lemma 6.7

Fix an input length on which A succeeds. We say that the matrix S, the assignment x and the error vector e' (obtained by taking the first m coordinates of the $m + 25n$ bit error vector e) are good if

$$\Pr[A(S, w = Sx + e', v) = \langle x, v \rangle] > 0.95, \tag{4}$$

where the probability is taken over the choices of x, e' and v.

Claim 6.8. *Suppose that S, x and e' are good and that the output of B is distributed according to Eq. 3 with no statistical deviation. Then, with probability* $1 - 2^{-\Omega(n)}$*, the output $\hat{x}$ is* 0.1*-close to x.*

Proof. First observe that in each tuple (i_j, u_j, v_j), the index i_j is a random index in $[n]$, the vector u_j is a random vector of weight 3 conditioned on being zero at the i_j-th coordinate, and $v_j = \langle x, u_j \rangle + \chi_j$ where χ_j is an error bit which equals to 1 with probability $\varepsilon' = 2(1 - \varepsilon)\varepsilon < 0.02$. Furthermore, all i_j, u_j, χ_j are independently distributed.

Call an iteration j *effective* if i_j was observed for the first time, and call it *successful* if $\hat{x}_{i_j} = x_{i_j}$. We wish to lower-bound the number of iterations which are both effective and successful.

First, observe that, except with exponentially small probability (taken over the choice of i_j's), there are at least $0.99n$ effective iterations. Indeed, the indices $(i_1, \ldots, i_t)$ contain less than $0.99n$ distinct indices with probability at most

$$\binom{n}{0.99n} 0.99^{6n} < 2^{n(H_2(0.99) + 6 \lg(0.99))} = 2^{-\Omega(n)}.$$

Fix a sequence $(i_1, \ldots, i_t)$ such that there are at least $0.99n$ effective iterations. We will show that in this case, except with exponentially probability, at least $0.91n$ of the effective iterations are also successful.

Let a_j be an indicator random variable which is set to 1 if A fails to predict $\langle x, \hat{u}_j \rangle$ in the j-th iteration. The probability that the j-th iteration is successful is therefore $\Pr[\chi_j + a_j = 0]$. Fix S, x and e', and let Q be the set of all weight-3 vectors v for which Eq. 4 holds. By the goodness of S, x and e', the set Q consists at least 0.95 fraction of all the possible $\binom{n}{3}$ triples. Recall that u_j is uniformly distributed over all $\binom{n}{3} - \binom{n-1}{2}$ triples in which i_j does not participate. Hence, for every fixed i_j, the probability that u_j lands in Q is at least $0.95 - \Theta(n^2/n^3) > 0.94$. It follows that the j-th iteration is successful with probability at least $0.98 \cdot 0.94 + 0.02 \cdot 0.06 > 0.92$. Since each iteration is successful independently of the others (as all the ξ_j's and a_j's are independent) we get, by a Chernoff bound, that except with exponentially small probability, at least 0.91 fraction of the affective iterations are all successful. It follows that, whp, we have more than $0.9n$ iterations which are both successful and affective, and the claim follows. □

To finish the proof of Lemma 6.7 we note that by Markov's inequality, the input (S, x, e') is good with probability at least 0.8 and that, by Lemma 6.6, the deviation of algorithm B results only in error of $2^{-\Omega(n)}$. Hence, except with probability $0.2 + 2^{-\Omega(n)}$, the string $\hat{x}$ is 0.1-close to x, and the lemma follows.

6.3 Predicting over different distributions

In the following, we will consider variants of the Predict3LIN problem, in which the distribution of the matrix $\binom{M}{v}$ is changed.

Theorem 6.9. *Let $\mathcal{D} = \{\mathcal{D}_n\}$ be a distribution ensemble which is $(1-\delta)$-computationally indistinguishable from $\mathcal{M}_{m,n}$ or $\mathcal{T}_{p,n}$ for $p = m/\binom{n}{3}$. Then, there exists a constant $C = C(\delta)$ for which the 0.99-intractability of* Predict3LIN(Cm, ε) *implies that* Predict3LIN$(\mathcal{D}, \varepsilon)$ *is $1 - \delta/4$ intractable.*

We will prove the theorem in several steps. First, we show how to take a good predictor A that solves Predict3LIN$(\mathcal{D}, \varepsilon)$ with probability θ and transform it into a good predictor for Predict3LIN$(\mathcal{C}, \varepsilon)$ where the distributions $\mathcal{C}$ and $\mathcal{D}$ are not too far. If the distributions are close (say, of statistical distance δ), then it is clear that A works well on Predict3LIN$(\mathcal{C}, \varepsilon)$ (i.e., solves it with probability $\theta - \delta$). However, we are interested in the case where $\mathcal{C}$ and $\mathcal{D}$ are not very close, i.e., the distance is at most $1 - \delta$ for some small but fixed δ. In this case, even if A solves Predict3LIN$(\mathcal{D}, \varepsilon)$ with very high probability (e.g., $\theta = 1$), we cannot hope to turn A to a good predictor on $\mathcal{C}$ as A might perform very badly on inputs which are common in $\mathcal{C}$ but exceptional under $\mathcal{D}$. The important observation is that we can efficiently detects these cases, and turn A into an algorithm B for Predict3LIN$(\mathcal{C}, \varepsilon)$ which sometimes declares "I do not know", but conditioned on not saying so, outputs the correct prediction with good probability. Formally, we call such an algorithm a *weak predictor* if, for infinitely many n's, it outputs "I do not know" with no more than constant probability $\alpha < 1$, and, conditioned on not outputting "I do not know", it outputs a correct prediction with probability $\beta > 1/2$ which is bounded away from $1/2$.

Lemma 6.10 (weak prediction over close distributions)**.** *Let $\mathcal{D} = \{\mathcal{D}_n\}$ be a distribution ensemble which is $(1-\delta)$-statistically close to the ensemble $\mathcal{C} = \{\mathcal{C}_n\}$, for some constant $0 < \delta < 1$. Suppose that there exists a predictor algorithm A that solves* Predict3LIN$(\mathcal{D}, \varepsilon)$ *with probability $1-\delta/4$. Then, there exists a* weak predictor *B for* Predict3LIN$(\mathcal{C}, \varepsilon)$.

Proof. Let $\alpha = (1-3\delta/4)/(1-\delta/2)$ and let $\beta = (\alpha - 0.5)/4$ by our assumption on $0 < \delta < 1$ being a constant, both α and β are positive constants. Fix a good input length n for which A predicts well. Let (M, b, v) be our input for $\mathsf{Predict3LIN}(\mathcal{C}, \varepsilon)$. First, we estimate the probability

$$\gamma(M, v) \stackrel{\text{def}}{=} \Pr_{x \stackrel{R}{\leftarrow} \mathcal{U}_n, e \stackrel{R}{\leftarrow} \mathrm{Ber}_\varepsilon^\ell} [A(T, Tx + e, u) = \langle x, u \rangle],$$

up to an additive error of β with confidence $1 - 2^{-\Omega(n)}$ by using Chernoff bound. (This is done by choosing n random x's and e's and checking how many of them satisfy $A(M, Mx + e, v) = \langle x, v \rangle$.) If the estimation is smaller than $\alpha - \beta$ we output "I do not know". Otherwise, we invoke A on the triple (M, b, v) and output its prediction.

Analysis: Call a pair (M, v) good if $\gamma(T, u) \geq \tau$. By Markov's inequality, when (M, v) is chosen from $\mathcal{D}$, it is good with probability at least $1 - \delta/2$. Hence, a sample from $\mathcal{C}$ is good with probability at least $1 - \delta/2 - (1 - \delta) = \delta/2$ which is a constant. It follows that we output "I do not know" with probability at most $1 - \delta/2 + 2^{-\Omega(n)}$ which is strictly smaller than 1. Moreover, if we do output a prediction then it is correct with probability $\alpha - 2\beta - 2^{-\Omega(n)} > 1/2$ which is bounded away from $1/2$. Hence, we get a weak predictor. □

Remark 6.11. *It is not hard to verify that the above lemma generalizes to computational distance, i.e., to the case where no* efficient *adversary can distinguish the ensemble $\mathcal{D}_n$ from the ensemble $\mathcal{C}_n$ with probability greater than $1 - \delta$.*

Our next step is to show that the intractability of $\mathsf{Predict3LIN}$ over $\mathcal{M}_{m,n}$ implies that $\mathsf{Predict3LIN}$ is also intractable with respect to $\mathcal{T}_{p,n}$ where $m \approx p\binom{n}{3}$. This is follows from the fact that $\mathcal{T}_{p,n}$ can be emulated given a sample from $\mathcal{M}_{2m,n}$ up to a statistical distance of $2^{-\Omega(m)}$. (We always assume that $p = p(n)$ is efficiently computable.)

Lemma 6.12. *Let $m = p\binom{n}{3}$. Suppose that there exists a weak-predictor A for $\mathsf{Predict3LIN}(\mathcal{T}_{p,n}, \varepsilon)$. Then, there exists a weak-predictor for $\mathsf{Predict3LIN}(2m, \varepsilon)$.*

Proof. The lemma follows from the fact that $\mathcal{T}_{p,n}$ can be easily emulated given a sample from $\mathcal{M}_{2m,n}$ up to a statistical distance of $2^{-\Omega(m)}$. Indeed, $\mathcal{T}_{p,n}$ can be written as a convex combination of $\beta_t \cdot \mathcal{M}_{t,n}$. Hence, given an input for $\mathsf{Predict3LIN}(2m, \varepsilon)$ that consists of $M \stackrel{R}{\leftarrow} \mathcal{M}_{2m,n}, v \stackrel{R}{\leftarrow} \mathcal{M}_{1,n}$ and $b = Mx + e$ where $x \stackrel{R}{\leftarrow} \mathcal{U}_n$, and $e \stackrel{R}{\leftarrow} \mathrm{Ber}_\varepsilon^m$, we can sample an integer t according to $\beta = (\beta_1, \ldots \beta_{\binom{n}{3}})$ if $t > 2m$, which happens with exponentially small probability, we fail. Otherwise, we output (M', b', v) where M' and b' are the first $t - 1$ rows of M and b. Clearly, a good prediction for (M', b', v) is also good for (M, b, v). □

Finally, we show that a weak predictor for $\mathsf{Predict3LIN}(m, \varepsilon)$ can be amplified into a predictor that solves $\mathsf{Predict3LIN}(tm, \varepsilon)$ for sufficiently large constant t. The idea is to partition an instance of $\mathsf{Predict3LIN}(tm, \varepsilon)$ into t instances for $\mathsf{Predict3LIN}(m, \varepsilon)$ and invoke the weak predictor on all of them. The symmetric structure of the 3-LIN problem allows us to rerandomize each of the t instances and therefore amplify the success probability.

Lemma 6.13 (Amplifying unpredictability via re-randomization)**.** *Suppose that we have a weak predictor for $\mathsf{Predict3LIN}(m, \varepsilon)$ then there exists a constant t (which depends only in the parameters of the weak predictor), for which $\mathsf{Predict3LIN}(tm, \varepsilon)$ is not 0.99-intractable.*

Proof. Suppose that we have weak predictor A which outputs "I do not know" with probability $\alpha < 1$, and, conditioned on not outputting "I do not know", it outputs a good prediction with probability $\beta > 1/2$. Let $t = t(\alpha, \beta)$ be a constant that will be determined later. Given an input (M, b, v) for $\mathsf{Predict3LIN}(Cm, \varepsilon)$ partition the matrix M (resp. the vector b) to t sub-matrices $M_1, \ldots, M_t$ (resp. vectors $b_1, \ldots, b_t$) each with m rows. Rerandomize the i-th instance as follows: choose a random $x_i \stackrel{R}{\leftarrow} \mathcal{U}_n$ and a random permutation π_i over $[n]$; Generate the triple $(T_i = \pi_i(M_i), b_i + T_i \cdot \pi_i(x_i), \pi(v))$, where we abuse notation and write $\pi(A)$ to denote the matrix A with columns permuted according to π.

Note that each of the t instances we created is a random instance of $\mathsf{Predict3LIN}(m, \varepsilon)$ and, in addition, all the instances are independently distributed. Also, observe that given a good prediction σ_i for the i-th instance we can compute a good prediction the original instance (M, v, x) by adding σ_i (over $\mathbb{F}_2$) to $\langle x_i, v \rangle$. Hence, we can apply A to each instance, translate its answer into a prediction for (M, v, x) (or to an "I do not know" symbol) and output the majority over the actual predictions. Since α and β are constants, we can use a Chernoff bound, to decrease the error probability below 0.01 by taking t to be a sufficiently large constant. □

The proof of Theorem 6.9 now follows as a corollary from Lemmas 6.10, 6.12, 6.13 and Remark 6.11.

6.4 Completing the proof of the main theorem

By combining Theorems 6.2, 6.5, and 6.9, we derive the following (stronger) version of Theorem 5.5.

Theorem 6.14 (main theorem - restated)**.** *For every constant $0 < \delta < 1$, there exists a constant $C = C(\delta)$, such that for every function $m = m(n)$, every $\varepsilon = \varepsilon(n) \leq 0.01$, and every distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}$ which is $(1-\delta)$-computationally indistinguishable from $\mathcal{M}_{m,n}$ or $\mathcal{T}_{m/\binom{n}{3},n}$ the following holds. If $\mathsf{Search3LIN}(C(m + n \lg n), \varepsilon)$ is intractable then the public-key encryption scheme $\mathcal{E}(\mathcal{D}, \varepsilon)$ is $(1 - \delta/2)$-private.*

7 PKE based on d-LIN and Decisional Unbalanced Expansion

In Section 5 we constructed a PKE based on the intractability of $\mathsf{Search3LIN}(n^{1.4}, n^{-0.2})$, our goal in this section is to relax this assumption and replace it with the intractability of solving d-LIN problem with a smaller number of equations ($m = n \log n$), and larger noise rate (e.g., $\varepsilon = n^{-0.1}$). We do this at the expense of introducing an additional assumption regarding the hardness of detecting the vertex expansion of random unbalanced bipartite graphs with planted shrinking set.

7.1 The Assumption

The DUE problem. In the following we view a d-sparse matrix $M \in \mathbb{F}_2^{m \times n}$ as a bipartite graph $G = ((V_{\mathsf{Top}}, V_{\mathsf{Bot}}), E)$ with m "top" nodes (each node correspond to a row) and n "bottom" nodes (each node correspond to a column) where each top node has degree d. Graphs chosen from $\mathcal{M}_{m,n,d}$ will be, with high probability, very good expanders. That is, we expect that small sets S of top vertices will have almost $d|S|$ neighbors. The distribution $\mathcal{F}^q_{n,m,d}$ is a perturbed version of $\mathcal{M}_{m,n,d}$ in which we plant a single q-size top subset S with a small ("shrinking") neighborhood. Formally, $\mathcal{F}^q_{m,n,d}$ is the result of the following random process: choose G from $\mathcal{M}_{n,m,d}$, choose at

random subsets $S \subseteq V_{\mathsf{Top}}$ and $T \subseteq V_{\mathsf{Bot}}$ of sizes q and $q/3$ respectively, and choose a random graph $H \stackrel{R}{\leftarrow} \mathcal{M}_{q/3,q,d}$. Then replace all the $d|S|$ edges in G that are incident to S with the edges from H. In the DUE problem the goal is to distinguish between a random graph sampled from $\mathcal{M}_{n,m,d}$ to a graph sampled from $\mathcal{F}^q_{n,m,d}$.

Definition 7.1. *Let $m = m(n), d = d(n)$ and $q = q(n)$ be some functions of n. We say that* $\mathsf{DUE}(d, m, q)$ *is* δ-intractable *if the distribution ensembles $\mathcal{M}_{n,m(n),d(n)}$ and $\mathcal{F}^q_{n,m(n),d(n)}$ are δ computationally indistinguishable.*

The problem becomes harder when m is increased, and, at least intuitively, it is becomes easier when d is increased as in this case it is harder to detect the shrinking subgraph. (This is opposed to d-LIN where the problem becomes easier when m is increased, and harder when d is decreased.) Again, one may hope that the problem is hard even for subexponential circuits (below the trivial limit n^q).

Dense hyper-subgraph formulation. The DUE problem 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. It can also be formulated as a conjecture on the hardness of a *planted dense subgraph problem in hypergraphs*. 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-th hyperedge of H contains the d neighbors of the i-th top-vertex of G. In this formulation, the DUE assumption is about the hardness of distinguishing hypergraphs that contain a somewhat *dense* sub-hypergraph — a set T of $q' = q/3$ vertices, such that the induced sub-hypergraph on T has at least q hyperedges— from graphs where the induced sub-hypergraph of every set of q' vertices (for q' up to roughly $n^{0.1}$ size or some other super-logarithmic bound) has only about q'/d edges. Thus DUE is equivalent to the problem of distinguishing between a random fairly sparse hypergraph ($m = O(n)$ hyperedges) and a random hypergraph with a planted somewhat *dense* (average degree larger than 1) small subgraph. Indeed, the analog of this problem for standard *graphs* (i.e., 2-uniform hypergraphs) has been studied by several works (e.g., [FPK01, Kho04, BCC$^+$10, ABBG10]). 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].

Let $\mathsf{SearchLIN}(d, m, \varepsilon)$ be the natural generalization of Search3LIN to sparsity d; that is, the goal is to find a solution to a d-LIN instance chosen from the uniform distribution $\mathcal{M}_{m,n,d}$ with noise rate ε. We will construct a public-key encryption scheme based on the following assumption which combines the intractability of SearchLIN and DUE:

Assumption 7.2. *There exists a function $q = q(n) = o(n)$, and an even constant d, for which:*

1. $\mathsf{SearchLIN}(d/2, m_S, \varepsilon)$ *is intractable for every $m_S \in O(n \lg n)$ and every $\varepsilon \in \Omega(1/q)$.*
2. $\mathsf{DUE}(d, m_D = Cn, q)$ *is $1/2000C^2$ intractable for sufficiently large constant C.*

Remarks on Assumption 7.2.

- (Parameters) It seems likely that Assumption 7.2 holds for the following setting of the parameters: Take d to be a small constant, q to be n^δ for some small constant $0 < \delta \ll 1/2$ (e.g., $1/10$), and set C to be much larger than $d^{1/\delta}$ (say, $100d^{1/\delta}$ or even $d^{2/\delta}$). See Section 10 for a detailed discussion on the last constraint.

- (Evidence for SearchLIN) In addition to all the evidence listed in Section 5.1 (i.e., resistance against k-wise adversaries, $\mathbf{AC^0}$ circuits, and Lasserre SDPs), we can rely on Viola [Vio08] and prove that the current parameters supply resistance against adversaries that can be represented as polynomials over $\mathbb{F}_2$ of degree $\alpha \lg n$ for some constant α. (See Section 9.)
- (Evidence for DUE) A natural way to try and break DUE is by counting the numbers of small subgraphs in the given graph, with the hope that they "pick up" the planted, denser part. Indeed, this approach seems to yield the best known algorithms for the densest sub-graph problem [BCC$^+$10]. In Section 10 we show that DUE cannot be broken by counting short (up to n^ϵ) 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 also relate DUE (and its variants) 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.

7.2 A Public-Key Encryption Scheme

Again, we rely on the general bit-encryption of Figure 1, but this time our key generation algorithm will be based on the DUE planted distribution.

The key generation algorithm. We will sample a pair of private/public-keys as follows. Let M be an d-sparse matrix chosen from $\mathcal{F}^q_{n,m(n),d}$ and let S be a shrinking set of size q. We say that a row i in S is degenerate if it is spanned by the other rows in S. Go over the rows of S in a random order, until a degenerate row i is found. (Such a row must exist as S is shrinking and therefore the column rank of the rows indexed by S is smaller than q.) Then, permute the i-th row of M with the last row, and set the public-key to be the permuted matrix M', and the private key to be the set S' which contains the last row of M' and the rows that span it, i.e., $\sum_{j \in S'} M'_j \pmod 2 = \mathbf{0}$.

7.2.1 Security

We will rely on a generalization of Theorem 5.5 in order to prove that the scheme is secure. Recall that we use $\mathcal{E}(\mathcal{D}, \varepsilon)$ to denote the scheme of of Figure 1 instantiated with noise $\varepsilon = \varepsilon(n)$ and public-key taken from some distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}$.

Theorem 7.3 (generalization of Thm 5.5)**.** *Let $0 < \delta < 1$ be a constant, $d \in \mathbb{N}$ be an even number, and $m(n) = \Omega(n \lg n), \varepsilon = \varepsilon(n) \leq 0.01$ be functions. Let $\mathcal{D} = \{\mathcal{D}_n\}$ be a distribution ensemble which is $(1-\delta)$-computationally indistinguishable from $\mathcal{M}_{m,n,d}$ or $\mathcal{T}_{m/\binom{n}{d},n}$. Then there exists a constant C which depends only in δ and d such that if* $\mathsf{SearchLIN}(d/2, Cm, \frac{1-\sqrt{1-2\varepsilon}}{2})$ *is intractable then the public-key encryption scheme $\mathcal{E}(\mathcal{D}, \varepsilon)$ is $(1-\delta/2)$-private.*

The main new ingredient in the proof of the theorem (compared to the proof of Thm 5.5) is a new reduction from the prediction variant of d-LIN to the approximate-search variant of $2d$-LIN which uses MAX-CUT as an intermediate problem. See Section 7.4 for full details.

Let $\mathcal{K} = \mathcal{K}^q_{n,m,d}$ be the distribution of the public key. Our goal now is to show that $\mathcal{K}$ is $(1-\delta)$ computationally close to the ensemble $\mathcal{M} = \mathcal{M}_{n,m,d}$ for some constant δ, and then apply Theorem 7.3. Recall that the intractability of DUE asserts that the above is true for the distribution

$\mathcal{F} = \mathcal{F}^q_{n,m,d}$, and note that $\mathcal{F}$ can be written as a convex combination $\alpha\mathcal{K} + (1-\alpha)\overline{\mathcal{K}}$, where $\overline{\mathcal{K}}$ is essentially $\mathcal{F}$ conditioned on the last row being out of the planted shrinking set. In general, this does not necessarily means that a good distinguisher for $\mathcal{K}$, yields a good distinguisher for $\mathcal{F}$. Consider, for example, an algorithm that always output 1 on inputs from $\mathcal{K}$, but will output 0 on inputs from $\overline{\mathcal{K}}$. Such an algorithm can be a very good distinguisher for $\mathcal{K}$, and still be useless for $\mathcal{F}$ (say, if it outputs 1 on the uniform distribution with probability α). The crux of the lemma, is to show that in this case, we can distinguish $\mathcal{K}$ from $\overline{\mathcal{K}}$, and therefore can recover some information regarding the planted shrinking set. This information allows us to certify the "shrinkage" of a noticeable fraction of the graphs in $\mathcal{F}$, and so it leads to a distinguisher for $\mathcal{F}$.

Theorem 7.4. *Suppose that* $\mathsf{DUE}(d, Cm, q)$ *is* $1/2000C^2$ *intractable. Then* $\mathcal{K}^q_{n,m,d}$ *is* $(1-\delta)$ *computationally close to the ensemble* $\mathcal{M}_{n,m,d}$ *where* $\delta = \delta(C)$ *is a constant which depends only in the constant* C*.*

We can prove the following corollary.

Corollary 7.5. *Under Assumption 7.2, there exists a semantically-secure public-key encryption scheme.*

Proof. We will use the parameters d and q which satisfy Assumption 7.2, and set the length parameter m to the function $m_D(n) \in \Theta(n)$ promised in Assumption 7.2. By Theorems 7.3 and 7.4, Assumption 7.2 implies that the resulting scheme is β-private for some constant $0 < \beta < 1$ as long as the noise rate ε is taken to be $1/Cq$ for an arbitrary constant C. Furthermore, β does not depend on the constant C. Hence, by taking C to be sufficiently large, we can reduce the decryption error $\alpha = \frac{1}{2} - \frac{1}{2}(1-2\cdot\varepsilon)^q$ (see Lemma 5.3) below $(1-\sqrt{\beta})/2$. Now, we have a weak PKE and the corollary follows by [HR05, Thm. 6]. □

We mention that, again, our weak encryption scheme can be converted to an oblivious transfer protocol.

7.3 Proof of Thm 7.4

We prove the following Theorem:

Theorem 7.6 (Thm 7.4 restated)**.** *Let* $q(n) = o(n)$ *and* $d \geq 3$*. Suppose that* $\mathsf{DUE}(d, Cm, q)$ *is* $1/2000C^2$ *intractable. Then* $\mathcal{K}^q_{n,m,d}$ *is* α *computationally close to the ensemble* $\mathcal{M}_{n,m,d}$ *where* $\alpha = \max(1 - 1/500C^2, 0.99)$*.*

Let $\mathcal{K}_n = \mathcal{K}^q_{n,m,d}$, $\mathcal{F}_n = \mathcal{F}^q_{n,m,d}$ and $\mathcal{M}_n = \mathcal{M}_{n,m,d}$. For a matrix $G \in \text{support}(\mathcal{F}_n)$ we let $S(G)$ be the planted shrinking set of rows of G, let $\overline{S}(G) = [m] \setminus S(G)$ be the set of rows out of the planted shrinking set, and let $D(G) \subseteq S(G)$ be the subset of degenerate rows (the ones that are spanned by other rows in $S(G)$), and let $\overline{D}(G) = S(G) \setminus D(G)$.

Consider the following ensembles over pairs:

1. (G, i) where $G \xleftarrow{R} \mathcal{F}_n$ and $i \xleftarrow{R} D(G)$.
2. (G, i) where $G \xleftarrow{R} \mathcal{M}_n$ and $i \xleftarrow{R} [m]$.

It is not hard to see that a δ-distinguisher for this ensembles exists if and only if there exists a δ-distinguisher for $\mathcal{K}_n$ and $\mathcal{M}_n$. (Recall that $\mathcal{K}_n$ is just $\mathcal{F}_n$ conditioned on having the last row in $D(\mathcal{F})$.) Hence, we may prove the theorem in these terms.

Assume, towards a contradiction, that there exists a probabilistic polynomial time distinguisher A such that for infinitely many n's we have

$$\Pr_{G \stackrel{R}{\leftarrow} \mathcal{F}_n, i \stackrel{R}{\leftarrow} D(G)}[A(G,i)=1] - \Pr_{G \stackrel{R}{\leftarrow} \mathcal{M}_n, i \stackrel{R}{\leftarrow} [m(n)]}[A(G,i)=1] > \max(1 - 1/500C^2, 0.99). \tag{5}$$

We will present an algorithm B that given a graph G outputs "planted" with probability $1/20$ when the graph G is chosen from $\mathcal{F}_n$, and outputs "random" whp $(1-o(1)$ over a random graph $G \stackrel{R}{\leftarrow} \mathcal{M}_n$. Such an algorithm clearly breaks our assumption regarding DUE.

The algorithm B. Given a matrix G, we remove all the rows i for which $A(G,i)$ outputs 0. Let G' be the resulting matrix. If G' has more than $n/18C$ rows output "random". Otherwise, think of G' as a bipartite graph with (at most) $n/18C$ left vertices with degree d and n right vertices, and check whether there exists a perfect matching that consists of all left nodes. If so output "random", otherwise, output "planted".

Claim 7.7. *For all sufficiently large n's, if $G \stackrel{R}{\leftarrow} \mathcal{M}_n$ then B will output "random" with probability at least $1-o(1)$.*

Proof. Suppose that $B(G)$ outputs "planted". Then, G' has no matching that consists of all left vertices, and therefore, by Hall's theorem, G' has a shrinking left set. Since G' is a subgraph of G, it follows that G has a left set of vertices of size at most $n/18C$ which shrinks. By standard calculations, a random graph $G \stackrel{R}{\leftarrow} \mathcal{M}_n$, will have such a set only with probability $o(1)$. Specifically, this probability is bounded by

$$\begin{aligned}
\sum_{s=d}^{n/18C} \binom{Cn}{s} \cdot \binom{n}{s} \cdot \left(\frac{s}{n}\right)^{ds} &\leq \sum_{s=d}^{n/18C} \left(\frac{Cn^2 e^2 s^d}{s^2 n^d}\right)^s \\
&\leq \sum_{s=3}^{n/18C} \left(\frac{Ce^2 s}{n}\right)^s \\
&\leq \sum_{s=3}^{\lg^2 n} \left(\frac{Ce^2 \lg^2 n}{n}\right)^3 + \sum_{s=\lg^2 n}^{n/18C} \left(\frac{Ce^2 n}{18Cn}\right)^{\lg^2 n} \leq o(1),
\end{aligned}$$

where the second inequality holds as $d \geq 3$. This completes the proof of the claim. □

It is left to prove the following lemma.

Lemma 7.8. *For infinitely many n's, if $G \stackrel{R}{\leftarrow} \mathcal{F}_n$ then B will output "planted" with probability at least $1/20$.*

Before we prove the lemma, let us collect several observations. By the assumption on the indistinguishability of DUE, for all sufficiently large n's we have

$$\Pr_{G \stackrel{R}{\leftarrow} \mathcal{F}_n, i \stackrel{R}{\leftarrow} [m]}[A(G,i)=1] - \Pr_{G \stackrel{R}{\leftarrow} \mathcal{M}_n, i \stackrel{R}{\leftarrow} [m(n)]}[A(G,i)=1] < 1/2000C^2, \tag{6}$$

as otherwise, we can tell whether a graph G came from $\mathcal{F}_n$ or from $\mathcal{M}_n$ by appending a random index i and applying A to (G, i).

Fix an n for which both Eq. 5 and 6 hold, and let $m = m(n), q = q(n)$, $\mathcal{K} = \mathcal{K}_n$, $\mathcal{F} = \mathcal{F}_n$ and $\mathcal{M} = \mathcal{M}_n$. We say that a matrix G chosen from $\mathcal{F}$ is good if the following holds:

$$\Pr_{i \overset{R}{\leftarrow} S(G)}[A'(G, i) = 1] \geq 0.6 \tag{7}$$

$$\Pr_{i \overset{R}{\leftarrow} \overline{S}(G)}[A'(G, i) = 1] \leq 1/20C^2 \tag{8}$$

Claim 7.9. *A random $G \overset{R}{\leftarrow} \mathcal{F}$ satisfies the first property of goodness (Eq. 7) with probability at least $1/10$.*

Proof. By Eq. 5 we may assume that

$$\Pr_{G \overset{R}{\leftarrow} \mathcal{F}, i \overset{R}{\leftarrow} D(G)}[A(G, i) = 1] > 0.99.$$

Hence, by Markov's inequality, with probability at least $1/10$, a random $G \overset{R}{\leftarrow} \mathcal{F}$ satisfies

$$\Pr_{i \overset{R}{\leftarrow} D(G)}[A(G, i) = 1] > 0.9.$$

To finish the proof it suffices to show that for every $G \in \text{support}(\mathcal{F})$ we have $|D(G)| \geq 2|S(G)|/3$, as in this case

$$\Pr_{i \overset{R}{\leftarrow} S(G)}[A(G, i) = 1] \geq 2/3 \Pr_{i \overset{R}{\leftarrow} D(G)}[A(G, i) = 1] > 0.6.$$

Indeed, for a matrix G let H be the submatrix which consists of the rows of the shrinking set $S(G)$. Then, the number of non-degenerate rows $|\overline{D}(G)|$ is at most $\text{rank}(H)$ which is bounded by $q/3$ as H has only $q/3$ non-zero columns. □

Claim 7.10. *A random $G \overset{R}{\leftarrow} \mathcal{F}$ which satisfies the first property of goodness (Eq. 7), will also satisfy the second property (Eq. 8) with probability at least $1/2$.*

Proof. Let $\mathcal{E}$ be the distribution of $G \overset{R}{\leftarrow} \mathcal{F}$ conditioned on the event that G satisfies Eq. 7 but violates Eq. 8. If the claim does not hold we can write

$$\begin{aligned}
\Pr_{G \overset{R}{\leftarrow} \mathcal{F}, i \overset{R}{\leftarrow} [m]}[A(G, i) = 1] &> \frac{1}{2} \cdot \frac{1}{10} \Pr_{G \overset{R}{\leftarrow} \mathcal{E}, i \overset{R}{\leftarrow} [m]}[A(G, i) = 1] \\
&= \frac{1}{20} \left(\Pr_{G \overset{R}{\leftarrow} \mathcal{E}, i \overset{R}{\leftarrow} S(G)}[A(G, i) = 1] \frac{q}{m} + \Pr_{G \overset{R}{\leftarrow} \mathcal{E}, i \overset{R}{\leftarrow} \overline{S}(G)}[A(G, i) = 1](1 - \frac{q}{m}) \right) \\
&> \frac{1}{20} \left(0.6 \frac{q}{m} + (1/20C^2)(1 - \frac{q}{m}) \right) > 1/400C^2,
\end{aligned}$$

where the last inequality follows as $C \geq 1$. On the other hand, by 5, we know that

$$\Pr_{G \overset{R}{\leftarrow} \mathcal{M}_n, i \overset{R}{\leftarrow} [m(n)]}[A(G, i) = 1] < 1/50C.$$

It follows that

$$\Pr_{G\stackrel{R}{\leftarrow}\mathcal{F}_n, i\stackrel{R}{\leftarrow}[m]}[A(G,i)=1] - \Pr_{G\stackrel{R}{\leftarrow}\mathcal{M}_n, i\stackrel{R}{\leftarrow}[m(n)]}[A(G,i)=1] > 1/400C^2 - 1/500C^2 = 1/2000C^2,$$

which contradicts Eq. 6. □

By combining the two claims it follows that a random $G \stackrel{R}{\leftarrow} \mathcal{F}$ is good with probability at least 1/20. We can now prove Lemma 7.8 and finish the proof of the theorem.

Proof of Lemma 7.8. It suffices to show that B outputs "planted" whenever G is good. Indeed, if G is good then G' has at most $q+(m-q)/20C^2 < q+n/20C < n/18C$ rows (recall that $q = o(n)$). In addition, by the first property of goodness, at least $0.6q$ rows of the shrinking set $S(G)$ appear at G'. Furthermore, the neighborhood of these rows is at most $q/3$ and therefore, by Hall's theorem, G' has no perfect matching, and the claim follows. □

7.4 Proof of Thm 7.3

The proofs of Theorems 6.2 and 6.9 directly generalize to the case of d-LIN for arbitrary constant d (with a constant multiplicative loss of the parameters in the case of Theorem 6.2.) However, the approach taken in the proof of Theorem 6.5 (which transforms a predictor to approximate-inverter) is specifically tailored to the case of $d = 3$. Below, we show how to prove a generalization of this theorem by relying on a different approach.

Theorem 7.11. *There exists a constant $C > 0$ and a constant $0 < \mu < 1/2$ for which the following hold. For every $\varepsilon \leq \mu$, $m > n$ and a constant $d \geq 3$, the intractability of* $\mathsf{AppSearchLIN}(d, Cm, \varepsilon)$ *implies that* $\mathsf{PredictLIN}(2d, m, 2\varepsilon(1-\varepsilon))$ *is $(1-\mu)$-intractable.*

The idea is to use the predictor to obtain a random instance of 2-LIN problem, then to use an approximation algorithm to obtain a string $\hat{x}$ which satisfies many of the constraints, and finally argue that since the instance is random $\hat{x}$ is close in Hamming distance to x.

7.4.1 Using a predictor to reduce AppSearchLIN to 2-LIN

We show how to convert an algorithm A that solves $\mathsf{PredictLIN}(2d, m, 2\varepsilon(1-\varepsilon))$ with probability $1-\mu$ for infinitely many n's, into an algorithm B that takes a random instance of $\mathsf{AppSearchLIN}(d, \Theta(m+t), \varepsilon)$ and generates a random instance of $\mathsf{AppSearchLIN}(2, t, \varepsilon')$ with the same planted assignment where $\varepsilon' = \varepsilon'(\mu, \varepsilon)$ decreases with μ and ε. Given an input $(M, b) \in \mathbb{F}_2^{(5t+4m-4)\times n} \times \mathbb{F}_2^{5t+4m-4}$, the algorithm B does the following.

Algorithm $B(M, b)$.

1. Partition M (resp. b) into two parts M_1 and M_2 (resp. b_1 and b_2) where M_1 (resp. b_1) consists of the first $5t$ rows and M_2 (resp. b_2) the remaining $4m - 4$ rows.

2. Use the algorithm A_1 of Lemma A.1 to transform (M_1, b_1) to a pair (R, y) where each row of $R \in \mathbb{F}_2^{t \times n}$ has weight $2d - 2$ and $y \in \{0,1\}^t$.

3. Use the algorithm A_2 of Lemma A.2 to transform (M_2, b_2) to a pair (R', y') where each row of $R' \in \mathbb{F}_2^{(m-1) \times n}$ has weight $2d$ and $y' \in \{0,1\}^{m-1}$.

4. For $j = 1, \ldots, t$ do the following:

   (a) Generate an instance of $\mathsf{PredictLIN}(2d, m, 2\varepsilon(1-\varepsilon))$ as follows. Let r_j be the j-th row of R. Choose two distinct random indices i_j and k_j that do not participate in the support of r_j and define a $2d$-weight vector u_j by taking r_j and turning the i_j and k_j components to 1's. Invoke the predictor A on (R', y', u_j) and record the result in σ'_j.

5. Output the pair (M', b') where $M' \in \mathbb{F}_2^{t \times n}$ is the 2-sparse matrix whose j-th row has ones in the locations i_j and k_j, and the j-th entry of the vector $b' \in \{0,1\}^t$ is $\sigma'_j + y_j \pmod 2$.

Lemma 7.12. *If the input to B is a random instance of $\mathsf{AppSearchLIN}(d, 5t + 4m - 4, \varepsilon)$ with planted assignment x, then the output of B is at most $(0.1 + o(1))$-far (in statistical distance) from an instance of $\mathsf{AppSearchLIN}(2, t, \varepsilon')$ with the same planted assignment x. Furthermore, ε' decreases with ε and μ.*

Proof. First note that by definition the output matrix $M' \in \mathbb{F}_2^{t \times n}$ is a random 2-sparse matrix. Let us assume for simplicity that Lemmas A.1 and A.2 perfectly generate uniform $\varepsilon' = 2\varepsilon(1-\varepsilon)$-noisy instances of $2d$-LIN (resp. $(2d-2)$-LIN) with no statistical deviation. (This assumption will have only exponentially small affect on the final probability quantities.)

For a triple of $(m-1, n, 2d)$-sparse matrix T, an n bit vector x, and an $m-1$ noise vector e, define

$$\alpha(T, x, e) = \Pr[A(T, Tx + e, z) = \langle x, z,] \rangle,$$

where the probability is taken over the coin tosses of A and the uniform choice of a $2d$-sparse vector z. Let us condition on the event that for the original planted assignment x and the pair (R', y') generated in the 3-rd step of B, we have that $\alpha = \alpha(R', x, y' - R'x)$ is not smaller than $\beta = 1 - 10\mu$. We let E denote this event. By Markov's inequality, E happens with probability at least 0.9 over the choice of x, M_2, b_2 and the internal coin tosses of A_2. (Recall that over a random instance, our predictor succeeds with probability $1 - \mu$.)

Claim 7.13. *Conditioned on E, the output vector b' can be written as $M'x + e'$ where $e' \overset{R}{\leftarrow} \mathrm{Ber}^t_{\varepsilon'}$ and $\varepsilon' = 2\varepsilon(1-\varepsilon)\alpha + (1-\alpha)(1 - 2\varepsilon(1-\varepsilon)) \leq 2\varepsilon(1-\varepsilon)(1-10\mu) + (10\mu)(1 - 2\varepsilon(1-\varepsilon))$.*

Proof. Each of the entries of the vector σ' is a good prediction with probability exactly α independently of the other entries (where the probability is taken over the choice of the indices i_j and k_j and the matrix R which in turn is induced by the choice of M_1). In addition, By Lemma A.1, each entry of y is noisy with probability $2\varepsilon(1-\varepsilon)$ independently of the other entries and independently of R, and so independently of whether the σ'_j's are correct. Hence, each of the entries of the output vector b' is noisy with probability $2\varepsilon(1-\varepsilon)\alpha + (1-\alpha)(1-2\varepsilon(1-\varepsilon))$, and the claim follows. □

Hence, the output of the algorithm is at most $0.1+o(1)$-far from an instance of $\mathsf{AppSearchLIN}(2, t, \varepsilon')$, and the lemma follows. (The $o(1)$ terms is due to the deviation in Lemmas A.1 and A.2.) □

7.4.2 Finding an approximate solution to 2-LIN

We should now solve a random instance of $\mathsf{AppSearchLIN}(2, t = \Theta(n), \varepsilon')$. Recall that there are known algorithms which given a $(1-\varepsilon)$-satisfiable 2-LIN instance find an assignment x' which satisfy $(1-\varepsilon')$ of the constraint of the problem. (This is the standard notion of approximating 2-LIN.) In particular, the seminal work of [GW95] provides such an algorithm with $\varepsilon' = O(\sqrt{\varepsilon})$. However, we are interested in slightly different approximation task: we need to find an assignment $\hat{x}$ which is very close to the planted assignment x. Fortunately, when the 2-LIN instance is random (more precisely, when the constraint graph is a good expander) we can show that the two tasks are essentially equivalent.

Lemma 7.14. *For every constant $0 < \delta < 1$ there exists a constant $C = C(\delta) > 1$ for which the following hold. Let (M, b) be a random 2-LIN instance where $M \stackrel{R}{\leftarrow} \mathcal{M}_{t=Cn,n,2}$ and $b = Mx + e$ with $x \stackrel{R}{\leftarrow} \mathcal{U}_n$ and $e \stackrel{R}{\leftarrow} \mathrm{Ber}^t_\varepsilon$. Then, except with exponentially small probability, any assignment x' which violates less than a $\delta/5 - 2\varepsilon$ fraction of the constraints will be at least δ-close (in Hamming weight) to x.*

Proof. Let $C = C(\delta)$ be a constant whose value will be determined later. Think of M as a random $(t, n, 2)$ graph $((V_{\mathsf{Top}}, V_{\mathsf{Bot}}), E)$. For a set S of bottom vertices we define $\Gamma_1(S)$ to be the set of neighbors of S which have only a single neighbor in S. We prove the following claim:

Claim 7.15. *With all but exponentially small probability over the choice of M, for every set S of size at most δn, we have $|\Gamma_1(S)| > \delta C n/5$.*

Proof. Consider the event F where for every set S of size at most δn, we have (F1) $|E(S)| < 2.5\delta C n$ and (F2) $|\Gamma(S)| > 1.5\delta C n$, where $E(S)$ is the set of edges incident to S, and $\Gamma(S)$ is the set of neighbors of S. By counting edges we observe that whenever F holds we also have $|\Gamma_1(S)| > 0.2\delta C n$. Hence, it suffices to bound the probability of F. For a fixed set S the event (F1) is violated with probability at most $\exp(-\frac{1}{16} \cdot \frac{2Cn\delta}{3})$. This follows by defining a random variable to each of the $2Cn$ edges which indicates whether The i-th edge touches S. These are independent Bernoulli variables with expectation $p \leq \delta$, hence the bound can be derived from a Chernoff bound.

Similarly, we can bound the probability that (F2) is violated for a fixed set S by $\exp(-\frac{Cn\delta}{16})$. Again, this can be proved by defining Cn random variable which indicate whether the i-th top node is connected to a node in S, and then applying a Chernoff bound.

By applying a union bound over all S's we get that F is violated with probability at most

$$\binom{n}{\delta n} \cdot \exp(-Cn\delta/24),$$

which is smaller than $\exp(-\Omega(n))$ for sufficiently large constant $C = C(\delta)$. □

Let us now condition on the event where M satisfies the claim and the planted assignment x satisfies $(1-2\varepsilon)$-fraction of the constraints of (M, b) (i.e., $y = Mx$ is 2ε-close to b). By a Chernoff bound and by the claim above, this happens with all but exponentially small probability. Assume that x' is δ-far from x. Then, the vectors $y = Mx$ and $y' = Mx'$ disagree in at least $(\delta/5)Cn$ indices and so y' differ from b in at least $(\delta/5 - 2\varepsilon)Cn$ locations. It follows that x' violates at least a $\delta/5 - 2\varepsilon$ fraction of the constraints. □

7.4.3 Complementing the proof of Thm. 7.11

Let θ be a sufficiently small constant and let $\mu, \varepsilon < \theta$ and let C be a sufficiently large constant. Suppose that there exists an algorithm A that solves $\mathsf{PredictLIN}(2d, m, 2\varepsilon(1-\varepsilon))$ with probability $1-\mu$ for infinitely many n's. Given an instance of $\mathsf{AppSearchLIN}(d, 5Cn + 4m - 4, \varepsilon)$ we will find, with probability 0.8, an approximate assignment which is 0.1-close to the planted assignment, and derive a contradiction.

Our approximate-inversion algorithm works as follows. First we transform the d-LIN instance to an instance which is $(0.1 + o(1))$-close to an instance of $\mathsf{AppSearchLIN}(2, Cn, \varepsilon'(\theta))$ via Lemma 7.12. Then, we apply the algorithm of [GW95] to obtain a solution $\hat{x}$ which satisfies a fraction of at least $1 - \varepsilon''(\theta)$ of the constraints with probability at least $0.9 - o(1)$ (where $\varepsilon'' = O(\sqrt{\varepsilon'})$). Now use Lemma 7.14 to argue that with probability $0.9 - o(1)$, the vector $\hat{x}$ is at least δ-close (in Hamming weight) to x for $\delta(C, \theta)$ which is smaller than 0.1 for a proper choice of C and θ. □

8 PKE based on DUE and DSF

We now describe a variant of our schemes, in which the d-LIN assumption is replaced by the hardness of non-linear constraint satisfaction problem which relies on the existence of certain pseudorandom generators in $\mathbf{NC^0}$. (This variant was announced in the preprint [BW08], subsumed by the current work.) We also show that the security of the resulting scheme implies that $O(\log n)$ Juntas cannot be learned efficiently.

8.1 The DSF assumption

For an (m, n, d) graph $G = ((V_{\mathsf{Top}}, V_{\mathsf{Bot}}), E)$, and a predicate $f : \{0,1\}^d \to \{0,1\}$, we define the function $G_f : \{0,1\}^n \to \{0,1\}^m$ obtained by mapping every $x \in \{0,1\}^n$ to $(f(x_{\Gamma(1)}, \ldots, f(x_{\Gamma(m)})$, where $\Gamma(i)$ denotes the neighbors of the i-th "top" node. Goldreich [Gol00] considered the case where $m = n$ and conjectured that whenever the graph G is a good expander (e.g., when G is random) and the predicate f is non-trivial, the function G_f is one-way. This conjecture is supported by several practical and theoretical evidences, including resistance against "myopic" backtracking algorithms [Gol00, Pan01, CEMT09]. We rely on a similar (yet stronger assumption) and conjecture that even when m is super-linear, the function G_f is a good pseudorandom generator. Our third PKE will be based on a combination of this assumption and DUE. Formally,

Assumption 8.1. *There exist parameters $m = m(n) = \omega(n)$, $\varepsilon = \varepsilon(n) \leq 1/10$ and a constant d, for which:*

1. $\mathsf{DUE}(d, m, q)$ *is ε intractable for some $q \in \Theta(\log n)$.*

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

Note that $q = O(\log n)$ and therefore, in order to avoid some attacks on DUE, we should let m be superlinear in n (e.g., $m = n^{1.1}$). We think of ε as a small constant (e.g., 0.01).

Evidence for DSF. In Section 11 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 SearchLIN assumption hold for this (and other) instantiations of the DSF assumption as well. In particular, the following theorem is proved in Section 11:

Theorem 8.2. *Assumption* DSF *instantiated with the "majority of three parities" function cannot be refuted using distinguishers that compute* $\mathbf{AC^0}$ *circuits, linear tests, or myopic distinguishers reading all the entries of* G *and* $\sqrt{n}$ *of the remaining output bits.*

Recently, [BQ09] showed that Goldreich's function becomes vulnerable when the output length m is sufficiently large and the predicate f is biased towards a pair of the inputs. Theorem 8.2 (and its generalization for δ-resilient functions see Section 11) complements this result and shows that when the predicate is not sensitive to small sets of inputs, the resulting function seems to be secure even for large values of m's.

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 (closely related to SearchLIN) 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[10] $\mathbf{NC^0}$ 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.

8.2 Constructing PKE

Let $m = m(n), q = q(n), \varepsilon = \varepsilon(n), d \in \mathbb{N}$, and $f : \{0,1\}^d \to \{0,1\}$ be parameters which satisfy Assumption 8.1. We rely on the following construction which is inspired by Naor's commitment [Nao91]:

- Key generation: Given security parameter 1^n, we will choose a graph $G \stackrel{R}{\leftarrow} \mathcal{F}^q_{n,m,d}$ together with a q-size shrinking set S, as well as a random string $r \stackrel{R}{\leftarrow} \mathcal{U}_m$. We publish the pair (G, r) as the public-key. We let the private key consists of the shrinking set S, and the graph H which is the subgraph of G induced by the set S and its neighbors.

[10] The use of logarithmic-size shrinking set forces us to take m to be super linear in n.

- Encryption: Choose a random $x \xleftarrow{R} \mathcal{U}_n$. To encrypt the bit 0 output $y = G_f(x)$; To encrypt the bit 1, output $y = G_f(x) + r \pmod 2$.
- Decryption: given a ciphertext z, output 0 if and only if z_S the restriction of z to the set S is in the image of H_f. (This verification can be implemented efficiently by trying all possible $2^{q/3} = \text{poly}(n)$ preimages.)

Lemma 8.3. *All but a $2^{-q/3}$ fraction of the keys are errorless. In fact, for every G sampled from $\mathcal{F}^q_{n,m,d}$ for all but a $2^{-n/3}$ fraction of $r \in \{0,1\}^m$, we have perfect correctness $\mathsf{Dec}(\mathsf{Enc}(\sigma; x)) = \sigma$ for every randomness x and plaintext $\sigma \in \{0,1\}$.*

Proof. Fix G, S and H. Let $r' = r_S$. Call r' bad if there are two different preimages $w_0, w_1 \in \{0,1\}^{q/3}$ for which $H_f(w_0) = H_f(w_1) + r'$. Clearly, a decryption error can happen only if r' is bad. Since any bad r' corresponds to (at least) one pair of w_0, w_1, we can bound the number of bad r's by $2^{q/3} \cdot 2^{q/3} \leq 2^{2q/3}$. However, $r' \xleftarrow{R} \mathcal{U}_q$ and therefore r' is bad with probability at most $2^{-q/3}$. □

Lemma 8.4. *Under Assumption 8.1, the above scheme is 4ε private.*

Proof. Let $G \xleftarrow{R} \mathcal{F}^q_{n,m,d}, r \xleftarrow{R} \mathcal{U}_m)$ be the real public-key and $\hat{G} \xleftarrow{R} \mathcal{M}_{n,m,d}$ be a "fake" public-key. Define the following hybrids:

$$D_1 = (G, r, \mathsf{Enc}_G(0)), \qquad D_2 = (\hat{G}, r, \mathsf{Enc}_{\hat{G}}(0)), \qquad D_3 = (\hat{G}, r, \mathcal{U}_m)$$
$$D_4 = (\hat{G}, r, \mathcal{U}_m + r), \qquad D_5 = (\hat{G}, r, \mathsf{Enc}_{\hat{G}}(1)), \qquad D_6 = (G, r, \mathsf{Enc}_G(1)).$$

By the DUE assumption, the pair D_1 and D_2 (resp., D_5 and D_6) are ε-indistinguishable. Moreover, by the DSF assumption, the pair D_2 and D_3 (resp., D_4 and D_5) are also ε-indistinguishable. Finally, it is not hard to see that D_4 and D_3 are just equivalent. Hence, D_1 and D_6 are 4ε indistinguishable. □

Since, $\varepsilon < 1/10$ and $q \in \Theta(\log n)$ we get a weak PKE scheme with $1 - 1/\text{poly}(n)$-correctness and $2/5$-privacy. Hence, we derive the following corollary

Corollary 8.5. *Under Assumption 8.1, there exits a semantically secure PKE.*

8.3 Hardness of Learning Juntas

A function $g : \{0,1\}^m \to \{0,1\}$ is k-junta if it depends in at most k of its variables. The problem of learning k-juntas in less than $m^{\Omega(k)}$ time is a well-known open problem in computational learning theory [Blu94, BL97]. We can use Assumption 8.1 to argue that $O(\log m)$-juntas cannot be PAC-learned in polynomial time. The idea is to use the fact that given an m-bit ciphertext our decryption algorithm looks at only $O(\log n) = O(\log m)$ of the bits of the ciphertext, and hence it computes an $O(\log n) = O(\log m)$-junta. The security of the PKE implies that this function is hard to learn. Formally,

Lemma 8.6. *Under Assumption 8.1, no efficient algorithm PAC-learns $O(\log m)$ juntas.*

Proof. By Lemma 8.4, for every efficient algorithm A, and all sufficiently large n's we have

$$\Pr_{G,s,\sigma,x}[A(G,r,\mathsf{Enc}_{G,r}(\sigma;x))=\sigma] \leq \frac{1}{2}+\frac{4\varepsilon}{2} < 0.7. \tag{9}$$

Suppose that we have a learner L with accuracy 0.9 and confidence 0.9. Then, we can use it to attack the scheme. Given an input $(G,r,w=\mathsf{Enc}_{G,r}(\sigma;x))$, we will use L to learn the decryption function $g=\mathsf{Dec}_{H,S}$ with instances coming from the distribution $z \stackrel{R}{\leftarrow} \mathsf{Enc}_{G,r}(\mathcal{U}_1)$. We can sample labeled examples from this distribution by choosing a random label $b \stackrel{R}{\leftarrow} \mathcal{U}_1$ and letting $z=\mathsf{Enc}_{G,r}(b;\mathcal{U}_n)$. When L outputs an hypothesis h, we apply it to w and output the resulting label.

Analysis. Suppose that the input comes from the "right" distribution, i.e., $G \stackrel{R}{\leftarrow} \mathcal{F}^q_{n,m,d}, r \stackrel{R}{\leftarrow} \mathcal{U}_m, \sigma \stackrel{R}{\leftarrow} \mathcal{U}_1$, and $x \stackrel{R}{\leftarrow} \mathcal{U}_n$. If the key is errorless, which by Lemma 8.3 happens with probability $1-o(1)$, the emulation of the distribution $(z \stackrel{R}{\leftarrow} \mathsf{Enc}_{G,r}(\mathcal{U}_1), g(z))$ is perfect, and therefore we output the right result with probability $1-0.1-0.1=0.8$. Overall we break the scheme with probability $0.8-o(1)$ which contradicts Eq.9. □

Part II
Investigating assumptions

9 Unconditional hardness of d-LIN

Assumptions 5.2 and 8.1 asserts that $\mathsf{SearchLIN}$ is hard (for different choices of parameters). We show that this hardness can be proven unconditionally for several restricted computational models. In fact, we prove a stronger statement: for almost all d-sparse matrices $M \in \mathbb{F}_2^{m\times n}$ the distribution $\mathcal{D}_{M,\varepsilon}$ of the m-bit vector $b=Mx+e$, where $x \stackrel{R}{\leftarrow} \mathcal{U}_n$ and $e \stackrel{R}{\leftarrow} \mathrm{Ber}^m_\varepsilon$, looks pseudorandom to a large family of algorithms. Roughly speaking, we show that this is the case as long as M forms a good expander. More formally, we view a M as a bipartite graph $(V_{\mathsf{Top}}, V_{\mathsf{Bot}}, E)$ and say that it is a (k,α) expander if for every $S \subseteq V_{\mathsf{Top}}$ with $|S| \leq k$, the neighborhood $\Gamma_M(S)$ of S has cardinality at least $\alpha|S|$. We say that a distribution D over $\{0,1\}^m$ ε-fools a class $\mathcal{F}$ of boolean functions over $\{0,1\}^m$ if for every $f \in \mathcal{F}$ we have $|\Pr[f(D)=1]-\Pr[f(U_m)=1]| \leq \varepsilon$. We can now prove the following theorem.

Theorem 9.1. *Let $M \in \mathbb{F}_2^{m\times n}$ be a d-sparse matrix that is a $(k, 0.51d)$ expander. Then, $\mathcal{D}_{M,\varepsilon}$*

1. *0-fools k-wise tests. ($\mathcal{D}_{M,\varepsilon}$ is k-wise independent.)*
2. *$\delta=\frac{1}{2}\cdot(1-2\varepsilon)^k$-fools linear tests. ($\mathcal{D}_{M,\varepsilon}$ is δ-biased.)*
3. *$8\cdot(1-2\varepsilon)^{k/2^{t-1}}$-fools degree t polynomials over $\mathbb{F}_2$.*

Proof. The first two items follow from the analysis of [MST03]. We sketch them here for completeness. We break the distribution $\mathcal{D}_{M,\varepsilon}$ into two independent parts: Y and E such that $\mathcal{D}_{M,\varepsilon}=Y+E$. This is done by letting $Y=M\cdot U_n$, and E be a random m-bit error vector whose entries take the value 1 with probability ε independently of each other.

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

$$\Pr\left[\sum_{i \in S} Y_i = 1\right] = \frac{1}{2} \tag{10}$$

Indeed, by a simple counting argument, there exists $i \in S$ with a unique neighbor $j \in \Gamma_M(i) \setminus \Gamma_M(S \setminus \{i\})$. Therefore, if we fix all inputs in $\Gamma_M(S \setminus \{i\})$ (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 $\frac{1}{2}$, establishing (10).

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

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

This follows by the fact that the sum of t independent Bernoulli random variables with expectation ε is 1 with probability $\frac{1}{2} - \frac{1}{2}(1 - 2 \cdot \varepsilon)^t$.

We proceed with (3). The following claim shows that the distribution $\mathcal{D}_{M,\varepsilon}$ can be written as the sum of t independent copies of $\mathcal{D}_{M,\alpha}$ for a related α.

Claim 9.2. *Let $\alpha = \frac{1}{2} - \frac{1}{2}(1 - 2\varepsilon)^{1/t}$. Then,*

$$\mathcal{D}_{M,\varepsilon} \equiv \sum_{i=1}^{t} \mathcal{D}^{(i)}_{M,\alpha},$$

where the $\mathcal{D}^{(i)}_{M,\alpha}$'s are independent copies of $\mathcal{D}_{M,\alpha}$.

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

$$\sum_{i=1}^{t} M(x^{(i)} + E^{(i)}) \equiv M \cdot \left(\sum_{i=1}^{t} x^{(i)}\right) + \sum_{i=1}^{t} E^{(i)} \equiv Mx + E,$$

where $E^{(1)}, \ldots, E^{(t)}$ are t independent error vectors of error-rate α, E is an error vectors of error-rate ε, 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 $\frac{1}{2} - \frac{1}{2}(1 - 2 \cdot \alpha)^t = \varepsilon$. The claim follows by choosing $x^{(1)}, \ldots, x^{(t)}$ uniformly and independently. □

Hence, by item 2, $\mathcal{D}_{M,\varepsilon}$ is the sum of t independent samples from δ-biased distribution, where $\delta = \frac{1}{2} \cdot (1 - 2\alpha)^k = \frac{1}{2} \cdot (1 - 2\varepsilon)^{k/t}$. Viola [Vio08] recently proved that in this case, the distribution $\mathcal{D}_{M,\varepsilon}$ also $8 \cdot (2\delta)^{1/2^{t-1}}$-fools degree t polynomials[11], which completes the proof. □

We can use Theorem 9.1 to validate the average hardness of our constructions.

Corollary 9.3. *The cryptosystem constructed in Section 5.2 which is based on* Search3LIN$(m = O(n^{1.4}), \varepsilon = \Omega(n^{-0.2}))$ *cannot be broken with probability greater than $o(1)$ by the following classes of algorithms:*

[11] 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 GF(2), is just a factor of two [BV07, Claim 33].

1. *Semidefinite programs which results from n^δ levels of the Lasserre hierarchy [Las01] where $\delta > 0$ is some constant.*

2. *Myopic algorithms that given a 3-LIN instance (M, b) apply an arbitrary function to (M, b') where b' is the restriction of b to at most n^δ indices which can be chosen adaptively, where $\delta > 0$ is some constant.*

3. *For any constant depth d, boolean circuits of that have NOT gates, and unbounded fan-in AND and OR gates ($\mathbf{AC^0}$ circuits) of size $\exp(-n^{\delta_d})$, where $\delta_d > 0$ is a constant that depends on d.*

Proof. The first item is proved by Schoenebeck [Sch08]. The second item follows from the first item of Theorem 9.1 together with the fact that a sample from $\mathcal{M}_{m,n}$ is guaranteed to be a $(n^\delta, 0.51 \cdot 3)$ expander with probability $1 - o(1)$ for some constant $\delta > 0$. Now we know that for almost all M's the vector $b \stackrel{R}{\leftarrow} \mathcal{D}_{M,\varepsilon}$ is $k = n^\delta$-wise independent, and therefore we can apply the recent breakthrough of Braverman [Bra09] to show that it also resists $\mathbf{AC^0}$ circuits. Specifically, [Bra09] shows that k-wise independent distributions α-fools the class of all $\mathbf{AC^0}$ circuits of depth d and size ℓ as long as $k = \log(\ell/\alpha)^{O(d^2)}$. Hence, when k is polynomial in n (i.e., $k = n^\delta$), as in our case, we get subexponential hardness of $\exp(-n^{\delta_d})$ for circuits of depth d for any constant d. □

For the parameters used in Assumption 7.2 (used for our second cryptosystem), we can obtain a stronger corollary.

Corollary 9.4. *Let $a > 0$ be a positive constant. The problem* $\mathsf{SearchLIN}(d \geq 3, m = O(n \lg n), \varepsilon = 1/n^a)$ *cannot be solved with probability greater than $o(1)$ by the following classes of algorithms:*

1. *Semidefinite programs which results from n^δ levels of the Lasserre hierarchy [Las01] where $\delta > 0$ is some constant.*

2. *Myopic algorithms that given a 3-LIN instance (M, b) apply an arbitrary function to (M, b') where b' is the restriction of b to at most n^δ indices which can be chosen adaptively, where $\delta > 0$ is some constant.*

3. *For any constant $d > 0$, $\mathbf{AC^0}$ circuits of depth d and size $\exp(-n^{\delta_d})$.*

4. *Polynomials of degree $t = \Omega(\log n)$ over $\mathbb{F}_2$.*

Proof. It is not hard to show that a sample from $\mathcal{M}_{m,n,d}$ is guaranteed to be a $(k = n^\delta, 0.51 \cdot d)$ expander with probability $1 - o(1)$ for every constant $0 < \delta < 1$. Now the first three items are proved similarly to the proof of the previous corollary. For the last item use Theorem 9.1 and note that the noise rate satisfies $1/\varepsilon = n^a$ and the expansion holds for polynomially larger sets of size $k = n^{a+b}$. Hence, we get subexponential hardness of $\exp(-n^{b/2})$ against polynomials of degree smaller than $0.49b \log(n)$. □

10 On the hardness of 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 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.

10.1 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, Alo86], and hence this value will vary considerably between the distributions $\mathcal{M}$ 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.[12]. 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 is 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^ϵ 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 $\hat{G}$ and $\hat{F}$ denote the variants of $\mathcal{M}$ and $\mathcal{F}$ where each edge is chosen with probability d/n independently (rather than insisting on d-regularity). We also assume that the planted set has only mild shrinkage of q to $q-1$ (rather than q vs. $q/3$). Our analysis for cycle counts is done with respect to these distributions. We believe that it can be extended for the distributions $\mathcal{M}$ and $\mathcal{F}$

[12] His original algorithm used a certain quasipolynomially large linear program

above. Moreover, by dropping vertices with too small a degree, $\hat{\mathcal{M}}$ and $\hat{\mathcal{F}}$ can be used for (variants of) our cryptosystems as well.[13]

Theorem 10.1. *For cycles of length $\ll \log_d n$, the distributions of the cycle count in $\hat{\mathcal{M}}$ 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 10.1 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 Poission 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.

10.2 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 $\mathcal{M}$ and $\mathcal{F}$ of DUE. This is of course much preferred over just ruling out certain types of algorithms, as is done in Section 10.1. 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 a variant of the DUE assumption (which we denote by DUE$'$) in which the distribution $\mathcal{M}$ is an arbitrary distribution over expander graphs and $\mathcal{F}$ is an arbitrary distribution with planted shrinking set (which also suffices, in conjunction with a variant of DUE or PredictLIN, for our cryptosystems). 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 10.2 (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 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.

[13] For the second cryptosystem this would require to assume the intractability of DecideLIN rather than SearchLIN, this is a stronger, yet plausible, assumption for which all our evidences hold as well.

Theorem 10.3 (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 $G = (V_{\mathsf{Bot}}, V_{\mathsf{Top}}, E)$ distinguish between the case that there is a $q = \mathrm{poly}(k)$-sized set $S \subseteq V_{\mathsf{Top}}$ with $|\Gamma_G(S)| < |S|$, and the case where for every set $S \subseteq V_{\mathsf{Top}}$ 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 $G = (V_{\mathsf{Bot}}, V_{\mathsf{Top}}, E)$ distinguish between the case that there is a $q = \mathrm{poly}(k)$-sized set $S \subseteq V_{\mathsf{Top}}$ with $|\Gamma_G(S)| < |S|$, and the case where for every set $S \subseteq V_{\mathsf{Top}}$ 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 $V_{\mathsf{Bot}} = V$ and $V_{\mathsf{Top}} = E$, and connecting every vertex in V_{Top} 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'.

Remark 10.4. *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.*

11 On the hardness of DSF

In this section we discuss what candidate nonlinear predicates can be used to instantiate Assumption DSF.

Definition 11.1 (δ-resilient functions)**.** *Let $\delta > 0$. We say that a function $f : \{0,1\}^d \to \{0,1\}$ is* δ-resilient *if for every subset $S \subseteq [d]$ with $|S| < \delta d$ and $a \in \{0,1\}^S$:*

1. $\Pr_{w \xleftarrow{R} W_{S,a}}[f(w) = 1] = \frac{1}{2}$*, 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 $i \notin S$, $\Pr_{w \xleftarrow{R} W_{S,a}}[f(w) = f(w \oplus e^i)] \in (0,1)$, where e^i is the vector that has 1 in the i^{th} coordinate and 0 everywhere else.*

For $\varepsilon > 0$, we say that the function f is (δ, ε)-*resilient* if in Condition 2 the probability is not just in the interval $(0,1)$ but in the interval $[\varepsilon, 1-\varepsilon]$. Note that this probability is over a sample space of size at most 2^d, and hence every δ-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 δd *perfect exposure resilient function*).

The parity function satisfies Condition 1, even with $\delta = 1$, but does not satisfy Condition 2 no matter how small δ is. An example for a 1/10-resilient function is the "majority on three parities" function. This is the function $f : \{0,1\}^{3k} \to \{0,1\}$ such that on input $w = x_1,..,x_k,y_1,..y_k,z_1,..,z_k \in \{0,1\}^{3k}$, f outputs the majority of the three bits x, y, z where $x = x_1 \oplus \cdots \oplus x_k$, $y = y_1 \oplus \cdots \oplus y_k$, and $z = z_1 \oplus \cdots \oplus z_k$. 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 $\frac{1}{2}$. 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 $\frac{1}{2}$ 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 $\frac{1}{2}$ that we will have $y \neq z$, in which case changing the value of x_i will flip the value of f.

11.1 k-wise independence

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

Theorem 11.2. *Let G be an (m, n, d)-graph that is a $(k, (1-\varepsilon)d)$ expander, and let f be a δ-resilient function for $\delta > 2\varepsilon$. Then, the distribution $G_f(U_n)$ is k-wise independent.*

Proof. The proof follows the proof of Part 1 of Theorem 9.1. Let $Y = G(U_n)$. We will prove the theorem by showing that for every subset $S \subseteq [m]$ with $|S| \leq k$,

$$\Pr[\bigoplus_{i \in S} Y_i = 1] = \frac{1}{2} \tag{11}$$

Indeed, by a simple counting argument, there exists $i \in S$ such that $|\Gamma_G(i) \setminus \Gamma_G(S \setminus \{i\})| \geq (1-2\varepsilon)d$. 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 2ε-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 $\frac{1}{2}$, establishing (11). □

Note that in this proof we only used Condition 1 of the definition of δ-resilient functions. In particular, Theorem 11.2 holds even if we use the *parity* function for f. (This was known before, see for example [MST03].) Note that, as mentioned above, Theorem 11.2 implies that G_f fools every $\mathbf{AC^0}$ circuit (of sub-exponential size) via the result of Braverman [Bra09].

11.2 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 ε-bias sample space). Note that random graphs satisfy these properties with high probability.

Theorem 11.3. *Let G be an almost right regular $(n\ell, n, d)$-graph that is a $(k, (1-\varepsilon)d)$-expander for $k > \omega(\ell^2)$. If f is δ resilient for $\delta > 2\varepsilon$ then for every $S \subseteq [m]$,*

$$\Pr[\bigoplus_{i \in S} Y_i = 1] \in \frac{1}{2} \pm 2^{-\Omega(k/\ell^2)}, \tag{12}$$

where the constant in the Ω notation depends on d but not on ℓ, n.

Proof. We may assume that $|S| \geq k$, since otherwise (12) is implied by k-wise independence (i.e., Theorem 11.2). Let X_1 be an bottom vertex that is connected to S. Let S_1 be the set of at most $d\ell$ top 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 $2d\ell^2$ top vertices that share an input with a member of S_1. Remove S'_1 from S and continue in this way to obtain $X_2, \ldots, X_t$ for $t \geq |S|/(2d\ell^2) = \Omega(k/\ell^2)$. Note that by construction, the sets $V_1, \ldots, V_t$ 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 $[n] \setminus \{X_1, \ldots, X_t\}$, the resulting function is a non-constant affine function of $X_1, \ldots, X_t$ and hence equals 1 with probability $\frac{1}{2}$.

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 $|\Gamma_G(j) \setminus \Gamma_G(S_i \setminus \{j\})| \geq (1-2\varepsilon)d$. 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 δ-resilient for $\delta > 2\varepsilon$, 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. □

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 11.2. [MST03]'s second generator used a different construction and analysis than ours— they used a specific construction of locality two.

12 Discussion and open problems

Structure in computational problems. In the worst-case setting, "lack of structure" is captured nicely by **NP**-completeness. In the average-case setting we don't have a fully satisfactory analog, though it does seem that some **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 3LIN assumption and its non-linear variant DSF 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". Nevertheless, we believe that our results, showing relations between DUE and problems such as planted clique and unbalanced expansion, and ruling out certain natural algorithms for it, do provide some very preliminary evidence for the "lack of structure" for DUE. More generally, we believe that breaking either the 3LIN or DUE assumption will be of interest beyond cryptography, to areas such as coding and learning theory (for 3LIN) and combinatorial optimization (for DUE).

Insights from approximation algorithms. Average-case assumptions with "planted" structures such as 3LIN and DUE immediately imply hardness of approximation results. There is of course no reduction in the other direction, since an **NP**-hardness result does not yield average-case hardness for the "natural" distribution on inputs, even if one assumes that there is an **NP**-language that is hard on the average. Still 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, there is still a significant gap between the best known algorithms and the best hardness of approximation results. For this reason we relate it to the better studied planted clique problems, and to the (possibly more accessible) problem of non-expansion of small sets in standard (not unbalanced bipartite) graphs. (Note that this is vertex expansion, and not edge expansion.) For 3LIN however, a related well studied gap problem is dLIN. 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 $\frac{1}{2}+\mu$ 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].) 1

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) stronger conditions $\mathbf{AM}\cap\mathbf{coAM}\not\subseteq\mathbf{BPP}$ or $\mathbf{BQP}\not\subseteq\mathbf{BPP}$.[14] These consequences hold for all factoring and discrete log based systems, and the first holds also for all lattice or "learning with errors" based systems [GK90, GG98, AR04, Sho97]. Our preliminary attempts to find such consequences for our systems failed, though of course more effort is required. (We note however that the system of Theorem 2.2 does satisfy a condition of a somewhat similar flavor in the sense that it's based on a search problem for which the corresponding refutation problem has a non-deterministic algorithm.) We also note that other coding-based

[14]Some of these consequences only hold for *promise* problem version of these problems. Defining these promise version analogs is subtle as there are trivial **NP** hard problems in $\mathbf{promise}-\mathbf{NP}\cap\mathbf{promise}-\mathbf{coNP}$. Nevertheless, it is possible to define more restricted classes $\mathbf{promise}(\mathbf{NP}\cap\mathbf{coNP})$ and $\mathbf{promise}(\mathbf{AM}\cap\mathbf{coAM})$ that do not contain an **NP**-hard problem unless the polynomial hierarchy collapses, and the task of breaking the above cryptosystem falls in these restricted classes [Gol05, Vad05].

schemes such as [McE78, Ale03] also seem to resist such implications.

Acknowledgements. We thank Noga Alon, Moses Charikar, Shafi Goldwasser, Thomas Holenstein, Ron Rivest, Madhu Sudan and Salil Vadhan for useful discussions. We also thank Aditya Bhaskara for sharing with us a copy of [BCC$^+$10].

References

[ABBG10] S. Arora, B. Barak, M. Brunnermeier, and R. Ge. Computational complexity and information asymmetry in financial products. In *ICS*, 2010.

[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.

[ACO08] D. Achlioptas and A. Coja-Oghlan. Algorithmic barriers from phase transitions. In *FOCS*, pages 793–802, 2008.

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997.

[AIK04] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput*, 36(4):845–888, 2006.

[AIK06] B. Applebaum, Y. Ishai, and E. Kushilevitz. On pseudorandom generators with linear stretch in NC^0. In *Proc. of RANDOM*, volume 4110, pages 260–271, 2006.

[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.

[Ale03] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307, 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, 1984.

[AR04] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. *J. ACM*, 52:749–765, 2005.

[ASS08] O. Amini, I. Sau, and S. Saurabh. Parameterized complexity of the smallest degree-constrained subgraph problem. In *IWPEC*, pages 13–29, 2008.

[AYZ95] N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.

[Bar82] A. D. Barbour. Poisson convergence and random graphs. *Math. Proc. Cambridge Philos. Soc.*, 92(2):349–359, 1982.

[BCC+10] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-density — an $O(n^{1/4})$-approximation for densest k-subgraph. In *STOC*, 2010.

[BFKL94] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO*, pages 278–291, 1994.

[BGI08] E. Biham, Y. J. Goren, and Y. Ishai. Basing weak public-key cryptography on strong one-way functions. In *TCC*, volume 4948, pages 55–72, 2008.

[BL97] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[Blu94] A. L. Blum. Relevant examples and relevant features: Thoughts from computational learning theory. AAAI Fall Symposium on Relevance, 1994.

[BMG09] B. Barak and M. Mahmoody-Ghidary. Merkle puzzles are optimal — an $O(n^2)$ attack on key exchange from a random oracle. In *Proceedings of CRYPTO '09*, 2009.

[Bol01] B. Bollobás. *Random Graphs*. 2001.

[BQ09] A. Bogdanov and Y. Qiao. On the security of goldreich's one-way function. In *APPROX-RANDOM*, pages 392–405, 2009.

[Bra09] M. Braverman. Poly-logarithmic independence fools AC^0 circuits. In *CCC*, pages 3–8, 2009.

[BV07] A. Bogdanov and E. Viola. Pseudorandom bits for polynomials. In *FOCS*, pages 41–51, 2007.

[BW08] B. Barak and A. Wigderson. Public key cryptography from different assumptions. Cryptology ePrint Archive, Report 2008/335, 2008. Contains a preliminary annoucement of some of the results in this paper.

[CEMT09] J. Cook, O. Etesami, R. Miller, and L. Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In *TCC*, 2009.

[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, 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.

[EGL85] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[Fei02] U. Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543, 2002.

[FGK05] Friedman, Goerdt, and Krivelevich. Recognizing more unsatisfiable random k-SAT instances efficiently. *SIAM J. Comput*, 35, 2005.

[FKO06] U. Feige, J. H. Kim, and E. Ofek. Witnesses for non-satisfiability of dense random 3CNF formulas. In *FOCS*, pages 497–508, 2006.

[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.

[GK90] O. Goldreich and E. Kushilevitz. A perfect zero knowledge proof for a problem equivalent to discrete logarithm. In *CRYPTO*, pages 57–70, 1990.

[GKL88] O. Goldreich, H. Krawczyk, and M. Luby. On the existence of pseudorandom generators. *SIAM J. Comput*, 22:1163, 1993.

[GM82] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[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, 1987.

[Gol00] O. Goldreich. Candidate one-way functions based on expander graphs. Technical Report TR00-090, Electronic Colloquium on Computational Complexity (ECCC), 2000.

[Gol05] O. Goldreich. On promise problems. Available on the author's home page at `http://www.wisdom.weizmann.ac.il/~oded/prpr.html`, 2005.

[GW95] Goemans and Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42, 1995.

[Hås97] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[HR05] T. Holenstein and R. Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In *CRYPTO*, pages 478–493, 2005.

[IKOS08] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In *STOC*, pages 433–442, 2008.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147, 1995.

[IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pages 44–61, 1989.

[JP00] A. Juels and M. Peinado. Hiding cliques for cryptographic security. *Designs, Codes and Cryptography*, 20(3):269–280, 2000.

[Kho04] S. Khot. Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. In *FOCS*, pages 136–145, 2004.

[Kob87] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[Las01] J. B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *IPCO: 8th Integer Programming and Combinatorial Optimization Conference*, 2001.

[Lau03] M. Laurent. A comparison of the Sherali-Adams, Lovasz-Schrijver, and Lasserre relaxations for 0-1 programming. *MOR: Mathematics of Operations Research*, 28:470–496, 2003.

[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`.

[LS91] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report*, pages 42–44, 1978.

[Mer78] R. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, 1978.

[Mil85] V. S. Miller. Use of elliptic curves in cryptography. In *CRYPTO*, volume 218, pages 417–426, 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.

[Nao91] M. Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[Pan01] S. K. Panjwani. An experimental evaluation of goldreich's one-way function. Technical report, IIT, Bombay, 2001.

[Pat96] J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *EUROCRYPT*, pages 33–48, 1996.

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342, 2009.

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008.

[Rab79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, 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. *J. ACM*, 51(6):899–942, 2004.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[RSA78] R. L. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. 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, 2000.

[SA90] H. D. Sherali and W. P. Adams. A hierarchy of relaxation between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Disc. Math.*, 3:411–430, 1990.

[Sch08] G. Schoenebeck. Linear level Lasserre lower bounds for certain k-csps. In *FOCS*, pages 593–602, 2008.

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput*, 26(5):1484–1509, 1997.

[Vad05] S. Vadhan. Personal communication, November 2009., 2005.

[Vio08] E. Viola. The sum of d small-bias generators fools polynomials of degree d. In *CCC*, pages 124–127, 2008.

[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 Two Sampling Lemmas

We will show how to transform a random instance of d-LIN with m equations and noise rate of ε into a random instance of $2d$-LIN or $(2d-2)$-LIN with $m' < m$ equations and noise rate of $\varepsilon^\star > \varepsilon$ while preserving the planted solution x. Here and in the rest of this section we let

$$\varepsilon^\star = 2 \cdot \varepsilon \cdot (1 - \varepsilon). \tag{13}$$

Notation: It will be convenient to think of d-weight vectors as sets of size d. For $x \in \{0,1\}^n$, we let $\mathcal{P}_{d,\varepsilon}(x)$ 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, and $y = \sum_{i \in S} x_i + \xi \pmod 2$ where ξ is an "error" coin which is 1 with probability ε. We write $\mathcal{P}^m_{d,\varepsilon}(x)$ to denote m independent samples from $\mathcal{P}_{d,\varepsilon}(x)$. Note that the distribution $\mathcal{P}^m_{d,\varepsilon}(x)$ is just a representation of the input distribution $(M, Mx + e)$ of $\mathsf{PredictLIN}(d, m, \varepsilon)$.

Lemma A.1. *[Generalization of Lemma 6.6] There exists an efficient algorithm C such that for every $x \in \{0,1\}^n$, and $d < \sqrt{n}/4$, the random variable $C(\mathcal{P}^{4m+n}_{d,\varepsilon}(x))$ is $\exp(-m/4)$-close (in statistical distance) to the random variable $\mathcal{P}^m_{2d-2,\varepsilon^\star}(x)$, where $\varepsilon^\star$ is defined as in Eq. 13.*

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 $\mathcal{P}^m_{2d-2,\varepsilon^\star}(x)$. 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 ε and therefore their sum is a noise bit of rate $2\varepsilon(1-\varepsilon)$.

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 $\{1,\ldots,n\}$. 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 $\exp(-4m\cdot(1/4)^2) \le \exp(-m/4)$, which completes the proof. □

To prove the security of our second encryption scheme we will also need the following variant of the sampling lemma.

Lemma A.2. *There exists an efficient algorithm C such that for every $x \in \{0,1\}^n$, and $d < \sqrt{n}/4$, the random variable $C(\mathcal{P}^{4m}_{d,\varepsilon}(x))$ is $\exp(-m/4)$-close (in statistical distance) to the random variable $\mathcal{P}^m_{2d,\varepsilon^\star}(x)$, where $\varepsilon^\star$ is defined as in Eq. 13.*

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 $\mathcal{P}^m_{2d,\varepsilon^\star}(x)$. 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 $\mathcal{P}_{2d,\varepsilon^\star}(x)$. 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 $\exp(-4m\cdot(1/4)^2) \le \exp(-m/4)$. □

B Hardness of DUE: Technical details

We complement missing technical details from Section 10.

B.1 Counting cycles

Notation: For convenience of analysis, we use here slightly different distributions $\hat{\mathcal{F}}$ and $\hat{\mathcal{M}}$ over random and planted graphs than the candidate distributions $\mathcal{M}$ and $\mathcal{F}$ suggested in Section 7.1. We believe that the results below do extend to the distributions of Section 7.1, and in any case our cryptosystem can be modified to work with the distributions used here. We let $\hat{\mathcal{M}} = \hat{\mathcal{M}}_{m,n,d}$ 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.The distribution $\hat{\mathcal{F}}$ is chosen by taking a random graph from $\hat{\mathcal{M}}_{m-q,n-q+1,d}$ 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 bipratite 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 $\hat{\mathcal{M}}$ and $X' = X'_{m,n,d,q,k}$ denote the number of $2k$-cycles in $\hat{\mathcal{F}}$. It can be shown that $\mathsf{E}[X'] > \mathsf{E}[X]$.

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:*

$$\left|\mathsf{E}[X] - \mathsf{E}[X']\right| < \tfrac{10qk}{\sqrt{m}}\sigma(X)\,,$$

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

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

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

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 $[\mathsf{E}[X], \mathsf{E}[X']]$. (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 $\mathcal{M}$ is easily shown to be:

$$\mathsf{E}[X] = m^{(k)} n^{(k)} (d/n)^{2k}/k \sim c^k d^{2k}/k \tag{14}$$

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

$$\mathsf{Var}[X] = \mathsf{E}[X^2] - \mathsf{E}[X]^2 = \sum_{\alpha,\beta} \mathsf{E}[X_\alpha X_\beta] - \sum_{\alpha,\beta} (d/n)^{4k} \tag{15}$$

Clearly for every α, β, $\mathsf{E}[X_\alpha X_\beta] \geq \mathsf{E}[X_\alpha]\,\mathsf{E}[X_\beta] = (d/n)^{4k}$. Thus for every set H of pairs (α, β), the RHS of (15) is lower bounded by

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

Let H be the set of pairs α, β 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. $\mathsf{Var}[X] \geq |H|(d/n)^{4k} \geq c^{2k}d^{4k}/(2m) = \mathsf{E}[X]^2(k/m)$

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

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

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 $\mathsf{E}[X]$ and $\mathsf{E}[X']$. On the other hand, $X_{k,k} = X_{q,q-1,d}$ which has expectation $\sim d^{2k}$ which is negligible compared to $\mathsf{E}[X]$. We claim that for $(a, b) \notin \{(0, 0), (k, k)\}$, $\mathsf{E}[X^{a,b}] \leq \frac{c^k d^{2k}}{k}\frac{q}{n} = (q/n)\,\mathsf{E}[X]$. Indeed, if $a > b$ then $X^{a,b} = 0$ with probability one, since in any cycle in a bipartite graph, a set of ℓ left-side vertices has at least ℓ 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,

$$\mathsf{E}[X^{a,b}] = \tfrac{1}{k} m^{(k-a)} q^{(a)} n^{(k-b)} (q-1)^{(b)} \left(\tfrac{d}{n}\right)^{2(k-a)} \left(\tfrac{d}{q}\right)^{2a} \leq$$
$$\tfrac{1}{k} c^{k-a} n^{2k-a-b} q^{a+b} d^{2k} q^{-2a} n^{-2k+2a} \leq \tfrac{c^k d^{2k}}{k} \left(\tfrac{q}{n}\right)^{b-a} \leq \tfrac{c^k d^{2k}}{k}\tfrac{q}{n}$$

□

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 $i = 1...2k$ 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 $m^{2k}n^{2k-i-1}(d/n)^{4k-i} \leq c^{2k}d^{4k}/n = \mathsf{E}[X]^2/n$, while the term t_k is equal to $\mathsf{E}[X]$. Thus we have that

$$\mathsf{E}[X^2] \leq \mathsf{E}[X]^2 + \tfrac{2k}{n}\,\mathsf{E}[X]^2 + \mathsf{E}[X]\,.$$

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

$$\mathsf{E}[X^\ell] \leq \mathsf{E}[X]^\ell + O(\tfrac{k}{n}\,\mathsf{E}[X^\ell] + \mathsf{E}[X])\,.$$

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

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

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 $\mathsf{E}[Z^2] \leq (1+\varepsilon)\,\mathsf{E}[Z]^2$ then*

$$\Pr[Z < \tau\,\mathsf{E}[Z]] < \tau^2 + \varepsilon$$

This implies the following "anti-concentration" bound. Let $\delta > 0$ be arbitrarily smal and fix T to be $\sqrt{\frac{\delta}{m}}\,\mathsf{E}[X]$. Then

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

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 $\mathsf{E}[|Y''|] = O(\frac{qk}{n}\,\mathsf{E}[X]) \ll T$, meaning that $|Y''| < T/10$ with $1 - o(1)$ probability. Since $\mathsf{E}[X] - \mathsf{E}[X'] < T/10$, this means that for every $\tau \in [\mathsf{E}[X], \mathsf{E}[X']]$, we can bound $|\Pr[\mathsf{E}[X] < \tau] - \Pr[\mathsf{E}[X'] < \tau]|$ by the probability that X is in $[\mathsf{E}[X] - T/2, \mathsf{E}[X] + T/2]$ 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*

$$\Delta\,(X, P_\lambda) < \varepsilon \tag{16}$$

$$\Delta\left(X', P_{\lambda'} + P_{\lambda''}\right) < \varepsilon \tag{17}$$

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

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

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

Proof. We use the facts that $P_\lambda + P_{\lambda'} \equiv P_{\lambda+\lambda'}$ and that $\Delta(P_\lambda, P_{\lambda(1+\varepsilon)}) \leq \varepsilon\sqrt{\lambda}$. In our case, the condition $d^2 < \varepsilon c$ implies that $\lambda'^2/\lambda = \frac{d^{4k}}{kc^k d^{2k}} < \varepsilon^k/k \leq \varepsilon^2/2$ (we can assume $k \geq 2$). For the purposes of bounding statistical distance increasing c only helps,[15] and hence we may assume $c = d^2/\varepsilon$. Plugging this into the bounds of Lemma B.7 gives the corollary. □

We now turn to proving Theorem B.7:

Proof of Theorem B.7. We start by showing (16). As in the proof of Theorem B.1, we write $X = \sum_\alpha X_\alpha$ where α ranges over all the $m^{(k)}n^{(k)}/k)$ potential $2k$ cycles and X_α is the indicator variable that is equal to 1 if the cycle α exists in the graph. We note that if α, β do not share an edge then X_α and X_β 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 α, X_α is an indicator variable. Suppose moreover that $\mathsf{E}[X_\alpha X_\beta] \geq \mathsf{E}[X_\alpha]\,\mathsf{E}[X_\beta]$ for every α, β and there is a symmetric reflexive relation $\sim$ such that $\alpha \not\sim \beta$ implies that X_α, X_β are independent. Then,*

$$\Delta\left(X, P_{\mathsf{E}[X]}\right) \leq 4\left(\sum_{\substack{\alpha\sim\beta \\ \alpha\neq\beta}} \mathsf{E}[X_\alpha X_\beta] + \sum_\alpha \mathsf{E}[X_\alpha]^2\right) \tag{18}$$

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 (18) of the pairs of cycles that share ℓ 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 (16).[16]

To show (17), 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 $P_{\lambda'}$. Thus $X_{0,0} + X_{k,k}$ is $4kd^{4k}\left(c^{2k}/n + 1/q\right)$-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 $\mathsf{E}[X^{a,b}] \leq \frac{c^k d^{2k}}{k}\frac{q}{n}$. Hence by Markov we get that the probability that $X^{a,b} > 0$ (and hence greater or equal to 1) is this expectation.

□

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 $\mathsf{DUE}'_{(q,e)\text{vs }(q',e')}$ denote the gap problem of distinguishing, given a bipartite graph $G = (V_{\mathsf{Bot}}, V_{\mathsf{Top}}, E)$, between the following two case:

[15] 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.

[16] Note that $\sum_\alpha \mathsf{E}[X_\alpha^2] \leq c^k d^{4k}/n^{4k} \leq (cd/n)^k$.

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

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

By abuse of notation, we also denote by $\mathsf{DUE}'_{(q,e)_{\mathrm{vs}\ (q',e')}}$ the assumption that the above problem is hard on the average, in the sense that there exist two sampleable distributions $\mathcal{M}, \mathcal{F}$ over (m,n,d)-graphs such that $\mathcal{F}$ is in the YES case with $1-o(1)$ probability, and $\mathcal{M}$ 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, $1/100$.[17] The DUE' assumption corresponds to the $\mathsf{DUE}'_{(q,e)_{\mathrm{vs}\ (q',e')}}$ assumption for $e = 1$, $e' = 0.9d$, and q' equalling the parameter k.

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 $\mathsf{SSE}_{(q,e)_{\mathrm{vs}\ (q',e')}}$ 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 $\varepsilon > 0$ and integer e that divides d, $\mathsf{SSE}_{(q,e)_{vs}\ (q',(1-\varepsilon)d)}$ on n-sized graphs of degree d reduces to $\mathsf{DUE}'_{(eq,1)_{vs}\ (q',(1-2e\varepsilon)d)}$ 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 SSE, we construct the graph $G' = (V_{\mathsf{Top}}, V_{\mathsf{Bot}}, E)$ as follows. Each vertex $u \in V$ has one corresponding vertex $u' \in V_{\mathsf{Bot}}$ and e corresponding vertices $u'_1, \ldots, u'_e$ in V_{Top}. We split the d neighbors of u arbitrarily to e groups $S_1, \ldots, S_e$ 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.

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 $(1-\varepsilon)d|S|$ neighbors, then every set S' of size q' in G' has at least $(1-2\varepsilon e)(d/e)q'$ neighbors. Suppose otherwise, then S' has more than $2\varepsilon dq'$ 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 $2\varepsilon dq' \geq 2\varepsilon d|S|$ non-unique edges, implying that it has less than $(1-\varepsilon)d|S|$ neighbors. □

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_{Top} and connecting each one to d random neighbors, or "hashing down" V_{Bot} by composing it with, say, a random unabalanced expander.*

[17] For simplicity of analysis, we will allow both distributions $\mathcal{F}$ and $\mathcal{M}$ to be over graphs $G = (V_{\mathsf{Bot}}, V_{\mathsf{Top}}, E)$ such that it does not necessarily hold that $|V_{\mathsf{Bot}}| = n$ and $|V_{\mathsf{Top}}| = m$, but rather $|V_{\mathsf{Bot}}|$ and $|V_{\mathsf{Top}}$ 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 $\Pr[|V_{\mathsf{Bot}}| = n, |V_{\mathsf{Top}}| = m] = 1$.

Planted clique problem. We define the *decisional planted k-clique problem in $G_{n',p}$* ($k\mathsf{DPC}_{n',p}$ 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 $p = \frac{1}{2}$ 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 $k\mathsf{DPC}_{n',2^{-\ell}}$ problem reduces to $\mathsf{DUE}'_{(k^2/3,1)_{vs}\,(2^{\ell/10},dk/(30\log n'))}$ in (m,n,d) graphs for $m = \Theta(n^2 2^{-\ell})$, $n = n'\mathrm{polylog}(n')$ 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 $\mathsf{DUE}'_{(q,1)_{\mathrm{vs}}\,(2^{\log^{0.9} n},d^{0.9})}$. 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 in variants of our cryptosystem, we need better expansion parameters that ensure that the adjacency matrix of the graph has no short (i.e., less than $1/\mu$) linear dependency as otherwise the system becomes insecure. 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*. A graph $G = (V_{\mathsf{Bot}}, V_{\mathsf{Top}}, E)$ is a *unique neighbor expander for q sets* if for $S \subseteq V_{\mathsf{Top}}$ 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 dependant. We have the following result on unique neighbor expansion:

Theorem B.13. *The $k\mathsf{DPC}_{n',2^{-\ell}}$ problem reduces to $\mathsf{DUE}'_{(k^2/3,O(\log^2 n/k\ell))_{vs}\,(2^{\ell/10},\mathrm{u.n.}))}$, where by this we denote the variant of 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 DUE' if there is some distribution that makes, say, $\mathsf{SSE}_{(q,2)_{vs}\,(q',0.99d)}$ 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.*[18] *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 $|\Gamma_{\hat{G}}(S)| \leq k$.*

[18] 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.

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 $t > (10 \log n'/\ell)k$. □

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 $\ell k/(30 \log n)$. □

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 $D : [m] \times [d'] \to [s']$ 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 $D : [m] \times [d'] \to [100s]$ 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 $G' : [m] \times ([d] \times [d']) \to [n] \times [100s] \times [d]$ 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.