Public-Key Cryptography from Different Assumptions

[Extended Abstract]

Benny Applebaum[*]
Weizmann Institute of Science
Rehovot, Israel
abenny@cs.technion.ac.il

Boaz Barak[†]
Princeton University,
Princeton, NJ, USA
boaz@cs.princeton.edu

Avi Wigderson[‡]
Institute for Advanced Study
Princeton, NJ, USA
avi@ias.edu

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.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public key cryptosystems; F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on discrete structures

General Terms

Algorithms, Security, Theory

Keywords

3LIN, Densest Subgraph Problem, Expander Graphs, Learning Juntas, Learning Parity with Noise, NC0, Public Key Cryptography

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., [19, 52, 40, 3]). 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 [29]) 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 [1], planted clique [32], and learning parity with noise [26, 11].[1] Thus a major goal of cryptography is to base public-

[*]Work done while at Princeton University. Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797.

[†]Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

[‡]Supported by NSF Grant CCF-0832797.

STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA

[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 [4, 51, 47]. 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.

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 [30]. 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 $\mathbf{NP}$-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 [24] (a *private-key* primitive), and the $\mathsf{3LIN}$ 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.

Structure of the paper.

Section 2 contain somewhat informal statements of our results and some discussion on their implications and relation to prior works. Section 3 contains a description of our scheme and a high level overview of the proofs of our main results. Section 4 contains formal statements of the main results as well as some proof outlines. A preliminary full version of this paper is available on the authors' home page.

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 $\mathrm{GF}(2)$, in which every row has d entries of value 1. Roughly speaking, we consider the following assumptions (see Section 4 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 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 [22, 34, 9, 7].

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]

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 $O(t^2)$ [41, 10] and this is optimal for black-box methods [30, 8].

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 $\mathbf{SZK}$ [25] and even (a suitable promise problem variant of) $\mathbf{NP} \cap \mathbf{coNP}$ [2], while in the worst-case these problems are $\mathbf{NP}$-hard for sufficiently large noise [6, 20].

[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 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., of length $m^{1/c}$).

Theorem 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 [4, 51, 47]. 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 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, Thm. 3 allows us much more flexibility in the choice of parameters for 3LIN; specifically, we avoid the parameter range in which [21]'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 [49] is probably the most well-studied average-case computational problem. Also, lattice based systems such as [3, 51, 47] 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.

2.2 More about the assumptions

We now elaborate on the evidence that support our assumptions and on some of their additional applications. 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 d*LIN *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 [36] of semi-definite programs, for some constant $\delta > 0$. The first item follows similarly to the analysis of Mossel et al [43], the second item employs the results of Viola [55] and Braverman [15], and the third item is implied by Schoenebeck [53].

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 [39] hierarchies (LS, and LS+) and the Sherali-Adams [54] hierarchy [37].

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 [11] 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 [4, 5]), 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 [24]. 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 [24] 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 [24, 45, 17]. Recently, [14] 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

[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.

[4] As an example, following this work, a variant of the DUE assumption was recently used by [7] (co-authored by the second author) to argue about the complexity of pricing financial derivatives. The DUE assumption also served as partial motivation for [9]'s recent work on the densest subgraph problem.

$c = c(d) > 1$). 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.[5] This suggests that the vulnerability discovered by [14] only holds for a "bad" choice of the predicate P. 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 [18, 43, 5, 5] which is also motivated by the ability to achieve highly efficient secure multiparty computation [31]. The only prior candidate (surviving a similar class of non-trivial attacks) was due to [43].

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 [9]). 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 [12, 13] is one of the most central open problems in computational learning theory.[6]

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 [57] and the web site [38] 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. [49, 52]) and discrete log in various groups (e.g. [19, 42, 35]) and schemes based on knapsack/lattices/error correcting codes (e.g., [40, 3, 4, 51, 47]).

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 [40] 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 [46] 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$(p)^n$.

More similar to ours are the schemes of Alekhnovich [4] and Regev [51]. Regev's scheme (and others such as [48, 47]) 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 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 [25, 2]. Most similar to ours is Alekhnovich's scheme [4][7] 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 [25, 2] 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 TECHNIQUES

To highlight some of our techniques let us sketch the proofs of Thms. 2 and 3. We define SearchLIN(d, m, ε) 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 GF$(2)^n$ and $e \in$ GF$(2)^m$ is chosen at 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 Search3LIN(m, ε) = SearchLIN$(3, m, \varepsilon)$.

PKE from 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

[5] Roughly speaking, our condition requires S to be a good expander and P to satisfy a (stronger) variant of t-resiliency [16]. See full version for details.

[6] 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.

[7] 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 [3, 50], where the public key is roughly a generating set for a lattice whose dual lattice has a planted short (in ℓ_2) basis.

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

Public encryption key: A d-sparse matrix $M \in \mathrm{GF}(2)^{m \times n}$ sampled from distribution $\mathcal{D}$.

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

Encryption: Randomly choose $x \overset{R}{\leftarrow} \mathcal{U}_n$ and a noise vector $e \in \{0,1\}^m$ whose entries are iid Bernoulli variables with mean ε. To encrypt 0, send the vector $b = Mx + e$. To encrypt 1, send the vector b with its last bit flipped.

Decryption: To decrypt $y \in \{0,1\}^m$ output the sum $\sum_{i \in S} y_i$. (Decryption errs with probability at most $q\varepsilon$.)

Figure 1: Our basic cryptosystem scheme, used in the proofs of Thms. 2 and 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 [28].

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 [28].

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.

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.

Prediction on a different distribution.

The last step 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 [21] 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. 3: PKE from DUE *and* dLIN.

The description above completes the reduction of Thm. 2. For Thm. 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 $\mathsf{PredictLIN}$ to $\mathsf{AppSearchLIN}$. Specifically, given an instance of $\mathsf{AppSearchLIN}$ 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 [23]) 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 $\mathsf{AppSearchLIN}$.

4. FORMAL STATEMENTS

4.1 Preliminaries

The $\mathsf{SearchLIN}$ problem. A matrix M is *d-sparse* if every row has exactly d nonzero elements. The distribution $\mathcal{T}_{p,n,d}$ chooses a d-sparse matrix by selecting any of the possible $\binom{n}{d}$ rows with probability p. The distribution $\mathcal{M}_{m,n,d}$ picks an $m \times n$ d-sparse matrix by choosing each row independently at random (with replacement) from all possible $\binom{n}{d}$ choices. We define $\mathsf{SearchLIN}(d, m, \varepsilon)$ to be the problem of recovering x from $(M, Mx + e)$ where $M \overset{R}{\leftarrow} \mathcal{M}_{m,n,d}$, $x \overset{R}{\leftarrow} \mathbb{F}_2^n$, and e is chosen at random such that $e_i = 1$ with probability ε. We say that $\mathsf{SearchLIN}(d, m, \varepsilon)$ is *intractable* if for every PPT algorithm A, and every sufficiently large n, the probability that A solves $\mathsf{SearchLIN}(d, m(n), \varepsilon(n))$ is smaller than 2/3.

Indistinguishability. A pair of distribution ensembles $\mathcal{X}_n, \mathcal{Y}_n$ are *ε-indistinguishable* if for every PPT algorithm C, we have $|\Pr[C(\mathcal{X}) = 1] - \Pr[C(\mathcal{Y}) = 1]| < \varepsilon$. An ensemble $\mathcal{X}_n$ is *ε-pseudorandom* if $\mathcal{X}_n$ is ε-indistinguishable from $\mathcal{U}_n$, the uniform distribution over n-bit strings.

Public-key encryption scheme. A $(\alpha(n), \beta(n))$-*secure public-key bit encryption scheme* is a triple $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms such that the algorithm Gen, on input 1^n produces a pair of private and public keys $(\mathsf{pk}, \mathsf{sk})$ and the scheme satisfy the following properties:

- $(1 - \alpha)$-*correctness*: For a random bit $b \overset{R}{\leftarrow} \{0, 1\}$ and a pair of random keys $(\mathsf{pk}, \mathsf{sk}) \overset{R}{\leftarrow} \mathsf{Gen}(1^n)$ we have: $\Pr[\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(b)) = b] \geq 1 - \alpha$.
- β-*privacy*: The random variables $(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(0))$ and $(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(1))$ are $\beta(n)$-indistinguishable, where pk is chosen by $\mathsf{Gen}(1^n)$.

If α and β are constants that satisfy $\alpha < (1 - \sqrt{\beta})/2$, we say that the scheme is a *weak PKE*. It was shown in [28, Thm. 6] that a weak PKE can be converted into semantically secure PKE [27] which supports arbitrary (polynomially) long messages.

4.2 Properties of the scheme $\mathcal{E}(\mathcal{D}, \varepsilon)$

Two of our constructions are based on the general bit-encryption scheme which is described in Figure 1. Recall that 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. The following lemma shows that decryption succeeds the noise rate ε is sufficiently small.

LEMMA 1. *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.*

To prove that the scheme is secure, it should be shown that it is hard to predict the last entry of the ciphertext b. Our main technical theorem shows that as long as M is chosen from a distribution which is not too far from $\mathcal{M}_{m,n,d}$ or $\mathcal{T}_{m/\binom{n}{d},n}$ the scheme is "somewhat" secure assuming that $\mathsf{SearchLIN}$ is intractable for related parameters. Formally,

THEOREM 4. *Let $0 < \delta < 1$ be a constant and d be either 3 or an even number larger than 3, 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:*

1. *For $d = 3$: If $\mathsf{SearchLIN}(3, cm, \varepsilon)$ is intractable then the public-key encryption scheme $\mathcal{E}(\mathcal{D}, \varepsilon)$ is $(1 - \delta/2)$-private.*
2. *For $d > 3$: 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.*[8]

[8] As a special case, the theorem essentially shows that for d-LIN, prediction is as hard as search. By combining this with Yao's theorem [56], one can show that the corresponding decision problem (distinguishing a random $(1-\varepsilon)$-satisfiable instance from a random one) is as hard as search as well (with some loss in the parameters).

PROOF. We will sketch the proof only for the (simpler) case of $d = 3$. As described in Section 3, we gradually reduce $\mathsf{Search3LIN}$ to the task of breaking $\mathcal{E}(\mathcal{D}, \varepsilon)$.

Step 1: From $\mathsf{Search3LIN}$ to $\mathsf{AppSearch3LIN}$. Suppose that we have an algorithm A that solves $\mathsf{AppSearch3LIN}(m, \varepsilon)$; that is, given a random ε-satisfiable 3-LIN instance $(M, b = Mx + e)$ A outputs an assignment $\hat{x}$ which is 0.1-close to x. We show how to convert A into an algorithm that solves $\mathsf{Search3LIN}(m+t, \varepsilon)$ with almost the same success probability, where $t \geq \Omega(n \ln n)$. Given an input $(M, b) \in \mathbb{F}_2^{(m+t)\times n} \times \mathbb{F}_2^{m+t}$, we invoke the approximation algorithm A on the first m rows of the input, and get an approximation $\hat{x}$ for x. Then, we use the information given by the last t rows of the input to correct the errors in $\hat{x}$ as follows. We will recover the i-th bit of x by letting each equation of the form $x_i + x_k + x_\ell = v$ 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 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. To analyze the algorithm, we show that each index i is likely to participate in many equations and that w.h.p most of these equations will include only indices for which x and $\hat{x}$ agree, hence, we output x_i with probability at least $1 - o(1/n)$ and therefore (by a union bound) all the indices of x are recovered w.h.p.

Step 2: From $\mathsf{AppSearch3LIN}$ to $\mathsf{Predict3LIN}$. Suppose that we have an algorithm A that solves $\mathsf{Predict3LIN}(m, \varepsilon)$. That is, given a random input $(M, b = Mx + e)$ the algorithm A predicts the value b_m without reading it. It will be convenient to parse the input as a tuple (M', b', v) where (M', b') is the first $m - 1$ rows of (M, b) ("training set"), and $v = (i, j, k)$ is the support of the last row of M ("test triple"). We show how to use A to solve $\mathsf{AppSearch3LIN}(m + t, \varepsilon)$ for $t = \Omega(n)$. Our reduction employs a subroutine B which transforms a random 3-LIN instance with t equations and noise ε into a random 4-LIN instance with $t/5$ equations and noise $\varepsilon' = 2\varepsilon(1 - \varepsilon)$ and the *same* planted solution. Given B we proceeds as follows.

We break the 3-LIN instance into two parts: the first m equations (M, b) and the remaining t equations (T, z) which are converted into a random 4-LIN instance (R, y) with $t/5$ equations via the subroutine B. Then, we invoke the predictor A for $t/5$ iterations, where the 3-LIN instance (T, z) is being used as the training set in all the iterations, and the test triple of the r-th iteration $v = (i, j, k)$ is randomly chosen from the support of the r-th 4-LIN equation $x_i + x_j + x_k + x_\ell = b$. We guess the value of the variable x_ℓ (which does not appear in v) by XOR-ing A's prediction bit with the LHS b of the 4-LIN equation. Our guess will be correct if the equation is not noisy and the prediction succeeds (or if both events do not happen simultaneously). It can be shown that, if A is a good predictor (e.g., succeeds with probability 0.99) then, with high probability, we will get good guesses for most indices (e.g., 0.9 fraction) and thus obtain a good approximation for x.

Finally, to implement the subroutine B we partition the 3-LIN equations into pairs that share a single variable x_i, and then combine each pair into a 4-LIN equation by simple addition. We make sure that the partition is oblivious to the other entries of the equations by first selecting a random representative from each equation, and then combining pairs that share a common representative. It is not hard to verify that such a strategy results in a random instance of 4-LIN system, and that, except with probability $\exp(-\Omega(t))$, we will get enough equations.

Step 3: Predicting over other distributions. Let $\mathcal{D}$ be a distribution ensemble over 3-sparse matrices which is $(1 - \delta)$-computationally close to $\mathcal{M}_{m,n}$ for some constant δ.[9] Suppose that $\mathcal{E}(\mathcal{D}, \varepsilon)$ is not $(1 - \delta/2)$-private. Hence, there exists a $(1 - \delta/4)$-predictor A for the problem $\mathsf{Predict3LIN}(\mathcal{D}, \varepsilon)$ in which the the matrix $\binom{M}{v}$ of the $\mathsf{Predict3LIN}$ instance $(M, b = Mx + e, v)$ is chosen from $\mathcal{D}$ and x and e are chosen as before. We show that A can be converted into an 0.99-predictor B for $\mathsf{Predict3LIN}(m', \varepsilon)$ (over the the uniform distribution) where $m' = c(m - 1) + 1$ and the constant $c = c(\delta)$ depends only on δ.

Let $\frac{1}{2} < \alpha < \beta < 1 - \delta/4$ be constants, we call a pair (M, v) *good* (resp. *bad*) if A succeeds on an instance $(M, b = Mx + e, v)$ with probability larger than β (resp., smaller than α) over a random choice of x and ε-noisy e. A proper choice of α and β guarantees that: (1) a random pair selected from $\mathcal{M}_{m,n}$ will be good with some constant probability $\gamma > 0$ (by Markov and the closeness of $\mathcal{D}$); and (2) one can efficiently distinguish bad pairs from good pairs by estimating A's success probability (say, via Chernoff bound).

Let us now describe the algorithm B. Given an input (M, b, v) for $\mathsf{Predict3LIN}(m', \varepsilon)$ partition the matrix M (resp. the vector b) to c sub-matrices $M_1, \ldots, M_c$ (resp. vectors $b_1, \ldots, b_c$) each with $m - 1$ rows. Rerandomize the i-th instance as follows: choose a random $x_i \xleftarrow{R} \mathcal{U}_n$ and a random permutation π_i over $[n]$; Generate the triple $(T_i = \pi_i(M_i), a_i = b_i + T_i \cdot \pi_i(x_i), v_i = \pi(v))$, where we write $\pi(R)$ to denote the matrix R with columns permuted according to π. Note that each of the c 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 for the original instance (M, v, x) by adding σ_i (over $\mathbb{F}_2$) to $\langle x_i, v \rangle$. Hence, we apply A to each instance, translate its answer into a prediction for (M, v, x) and output the majority over the predictions, ignoring the votes of the entries for which (T_i, v_i) are bad. Since $\frac{1}{2} < \alpha$ and $\gamma > 0$ are constants, a sufficiently large constant c decreases the error probability below 0.01. □

4.3 PKE based on 3LIN

We instantiate the scheme $\mathcal{E}(\mathcal{D}, \varepsilon)$ as follows. 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 either 0 or 2 ones. 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 $H \in \text{support}(H_{q,n}^{2,3})$ 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. First, we argue that the distribution $\mathcal{T}_{p,n,q}$ is not too far (in statistical distance) from $\mathcal{T}_{p,n}$.

LEMMA 2. *There exists a function $q = \Theta(n^{0.2})$ and a con-*

[9] The proof easily extends to the case where $\mathcal{D}$ is close to $\mathcal{T}_{p,n}$ for $p = m/\binom{n}{3}$.

stant $\delta < 1$ such that the distribution $\mathcal{T}_{p=n^{1.4}/\binom{n}{3},n}$ is $1-\delta$ statistically indistinguishable from $\mathcal{T}_{p,n,q}$.

The (omitted) proof relies on the results of Feige, Kim and Ofek [21]. We can now establish Thm. 2.

Proof of Thm 2. Let p, $q(n)$ and δ be the parameters obtained from Lemma 2 and let $c = c(\delta)$ be the parameter obtained from Thm. 4. Suppose that $\mathsf{Search3LIN}(cn^{1.4}, \varepsilon)$ is intractable for $\varepsilon = kn^{-0.2}$ where k is a sufficiently small constant that will be determined later. By Lemma 2 and Thm. 4, we get that the scheme $\mathcal{E}$ instantiated with the distribution $\mathcal{T}_{p=n^{-1.6},n,q}$ and noise rate ε is $\beta = (1-\delta/2)$-private. Since β is bounded away from 1, and since the error α tends to zero with k (Lemma 1), a sufficiently small constant k will make the error α smaller than $(1-\sqrt{\beta})/2$. Hence, we obtain a weak PKE which can be converted to a semantically secure PKE via the transformation of [28]. □

4.4 PKE based on dLIN and DUE

In the previous section 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 the $\mathsf{SearchLIN}$ 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 adding the DUE as an additional 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. We call such a graph an (m,n,d) graph. 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 $H \overset{R}{\leftarrow} \mathcal{M}_{q/3,q,d}$. 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 $\mathcal{F}^q_{n,m,d}$. Let $m = m(n), d = d(n)$ and $q = q(n)$ be some functions of n. We say that $\mathsf{DUE}(m,q,d)$ 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 key generation algorithm. Again, we rely on the general bit-encryption of Figure 1. Our key generation algorithm will be based on the DUE planted distribution. 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., $\sum_{j\in S'} M'_j \pmod 2 = \mathbf{0}$.

Security. Let $\mathcal{K} = \mathcal{K}^q_{n,m,d}$ denote 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 Thm. 4. 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 5. *Suppose that $\mathsf{DUE}(cm, q, d)$ 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.*

Proof sketch. Let $\mathcal{K}_n = \mathcal{K}^q_{n,m,d}$, $\mathcal{F}_n = \mathcal{F}^q_{n,m,d}$ and $\mathcal{M}_n = \mathcal{M}_{n,m,d}$. Assume, towards a contradiction, that we have a δ-distinguisher A for $\mathcal{K}_n$ and $\mathcal{M}_n$ where $\delta = 1 - 1/500c^2$. We use A to break DUE as follows. Given a graph G, create the graph G_i in which the i-th row is permuted with the last row, and invoke A on all the G_i's for $i \in [m]$. Define a graph G' by keeping only the rows i of G for which $A(G_i)$ outputs "planted". If the number of remaining rows is larger than n/kc we output "random" where k is some fixed universal constant (e.g., 18). Otherwise, think of G' as a bipartite graph with (at most) n/kc 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".

We argue that when $G \overset{R}{\leftarrow} \mathcal{M}_n$ the above algorithm outputs "random" with probability $1 - o(1)$. Indeed, suppose that we outputted "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/kc which shrinks. By standard calculations, a random graph $G \overset{R}{\leftarrow} \mathcal{M}_n$, will have such a set only with probability $o(1)$.

It is left to show that when $G \overset{R}{\leftarrow} \mathcal{F}_n$ the output is "planted" with some fixed constant probability (e.g., 1/20). To prove this we claim that for a typical graph $G \overset{R}{\leftarrow} \mathcal{K}_n$, we have that $A(G_i)$ outputs (1) "planted" for most rows i in the shrinking set, and (2) "random" for most rows i outside of the shrinking set. Assuming that the claim is true (with the appropriate parameters), we complete the argument by noting that if G satisfies (2) then G' has at most n/kc rows, and, in addition by (1), at least $q/2$ rows from the shrinking set appear in G'. Since the neighborhood of these rows is at most $q/3$, the graph G' has no perfect matching (by Hall's theorem), and the algorithm outputs "planted".

It is left to prove the claim. The first part of the claim follows (via Markov) from the fact that A, being a δ distinguisher, outputs "planted" with high probability over $G \overset{R}{\leftarrow}$

$\mathcal{K}_n$. The second part of the claim follows from DUE; as if (2) is violated $A(G_i)$ outputs "planted" with high probability over $G \overset{R}{\leftarrow} \mathcal{F}_n$ and a random $i \overset{R}{\leftarrow} [m]$, and since $A(H_i)$, where $H \overset{R}{\leftarrow} \mathcal{M}_n$ and $i \overset{R}{\leftarrow} [m]$, outputs "random" with high probability (by our assumption on A), we break DUE. □

Corollary 1 (Thm. 3 restated). *For every constants d, c and function $q = o(n)$, there exist constants a, b for which: if $\mathsf{DUE}(cn, q, 2d)$ is $1/2000c^2$ intractable as well as $\mathsf{SearchLIN}(d, an \log n, 1/(bq)))$ is intractable then there exists a semantically secure PKE.*

Proof. Suppose that the assumptions hold with parameters d, q, c and sufficiently large constant $a, b > 1$. By Thms. 4 and 5, the resulting scheme is β-private for some constant $0 < \beta < 1$ where β does not depend on the constant b. Hence, by taking b to be sufficiently large, we can reduce the decryption error $\alpha = \frac{1}{2} - \frac{1}{2}(1 - 2 \cdot \varepsilon)^q$ (see Lemma 1) below $(1 - \sqrt{\beta})/2$ and get a PKE. □

4.5 PKE based on DSF and DUE

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. We prove that there exists a PKE under the assumption that **(1)** (*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 $(G, \mathcal{U}_m)$ where $G \overset{R}{\leftarrow} \mathcal{M}_{m,n,d}$; and **(2)** $\mathsf{DUE}(m, q, d)$ is ε intractable for some $q \in \Theta(\log n)$.

The cryptosystem. The cryptosystem is slightly different than the previous ones, and is inspired by Naor's commitment [44]. To generate a key choose a graph $G \overset{R}{\leftarrow} \mathcal{F}^q_{n,m,d}$ together with a q-size shrinking set S, as well as a random string $r \overset{R}{\leftarrow} \mathcal{U}_m$. The pair (G, r) is the public-key, while 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. To encrypt we choose a random $x \overset{R}{\leftarrow} \mathcal{U}_n$. To encrypt the bit 0 output $y = G_f(x)$; To encrypt the bit 1, output $y = G_f(x) + r \pmod 2$. To decrypt a ciphertext z, we 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 $2^{q/3} = \mathrm{poly}(n)$ preimages.) It is not hard to prove the following Lemma which establishes Theorem 1.

Lemma 3. *(1) The scheme is errorless with respect to a fraction of $1 - 2^{-q/3}$ of the public-keys. (2) Under the assumptions above, the scheme is 4ε private.*

Proof sketch. (1) Fix G, S and H and let $v = r_S$. We show that for all but $2^{-q/3}$ fraction of the v's the scheme is errorless. Indeed, a decryption error can happen only if there are two different preimages $w_0, w_1 \in \{0,1\}^{q/3}$ for which $H_f(w_0) = H_f(w_1) + v$. Hence, the probability of choosing a "bad" v is at most $2^{q/3} \cdot 2^{q/3}/2^q$. (2) Follows by a standard hybrid argument. □

Since our decryption algorithm looks at only $O(\log m)$ of the bits of an m-bit ciphertext, and since for most of the keys the scheme is errorless, we can show, via standard techniques [33], that there is no efficient algorithm to PAC learn $O(\log m)$-juntas under the same assumptions.

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 [9].

5. REFERENCES

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

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

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

[4] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307, 2003.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

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

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

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

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

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

[16] Chor, Goldreich, Hastad, Freidmann, Rudich, and Smolensky. The bit extraction problem or t-resilient functions. In *FOCS*, 1985.

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

[18] M. Cryan and P. B. Miltersen. On pseudorandom generators in NC^0. In *Proc. 26th MFCS*, 2001.

[19] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(5):644–654, 1976.

[20] 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.

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

[22] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

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

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

[25] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.

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

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

[28] 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.

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

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

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

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

[33] M. Kearns and L. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994.

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

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

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

[37] 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.

[38] 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`.

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

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

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

[42] V. S. Miller. Use of elliptic curves in cryptography. In *CRYPTO*, volume 218, pages 417–426, 1985.

[43] E. Mossel, A. Shpilka, and L. Trevisan. On epsilon-biased generators in NC^0. *Random Struct. Algorithms*, 29(1):56–81, 2006.

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

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

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

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

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

[49] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, 1979.

[50] O. Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004.

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

[52] 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.

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

[54] 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.

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

[56] A. C. C. Yao. Theory and applications of trapdoor functions. In *FOCS*, pages 80–91, 1982.

[57] 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.