Public Key Cryptography from Different Assumptions

Boaz Barak* Avi Wigderson†

August 2, 2008

Abstract

We construct a new public key encryption based on two assumptions:

1. One can obtain a pseudorandom generator with small locality by connecting the outputs to the inputs using any sufficiently good unbalanced expander.
2. It is hard to distinguish between a random graph that is such an expander and a random graph where a (planted) random logarithmic-sized subset S of the outputs is connected to fewer than $|S|$ inputs.

The validity and strength of the assumptions raise interesting new algorithmic and pseudorandomness questions, and we explore their relation to the current state-of-art.

1 Introduction

Public key encryption is a central notion in cryptography – indeed security of current electronic commerce is based on it. Fortifying the foundations of public-key cryptography is thus a central goal. However, despite 30 years of research, very few candidates for such encryptions are known, and these are based on a handful of computational problems of a very structured, algebraic nature, from the areas of number theory, lattices, and error-correcting codes (e.g., [DH76, RSA78, McE78, AD97]).[1]

In this work we give a construction of a public key encryption based on different assumptions. The proposed system has, at this point, two major drawbacks. First, the new assumptions are not as well-studied as, say, the hardness of factoring. Second, the scheme is far less efficient and secure than might be hoped: it can be broken in $n^{O(\log n)}$ time, where n is the key length / security parameter of the system.

These drawbacks certainly demand attention. Regarding the second drawback, it is quite likely that other, more efficient variants will allow subexponential (rather than superpolynomial) security, and we are pursuing such ideas. Regarding the first one, we initiate here a study of the algorithmic and pseudorandomness questions which arise, relate them to known results, and obtain some preliminary new ones.

The main advantage of the new scheme is the relatively general and unstructured nature of the new assumptions – a combination of only (specific, of course) *private*-key cryptography and average case hardness. These seem qualitatively different than previous constructions. In particular, unlike most previously known public key cryptosystems, we see no obvious way to reduce breaking our system to a problem in **SZK** (or even in **AM** ∩ **coAM**), nor does it seem one can break our system using known algorithmic techniques for quantum computing. For more on this see Section 2.2.

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

*Department of Computer Science, Princeton University, boaz@cs.princeton.edu. Supported by NSF grants CNS-0627526 and CCF-0426582, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

†Institute for Advanced Study, Princeton, NJ, avi@ias.edu.

[1] This is in contrast to *digital signature* schemes, that can be based on any one-way function (or equivalently, private-key encryption) [NY89, Rom90]. Obtaining an analogous result for public key encryption is a longstanding open problem, and cannot be achieved via a black-box reduction [IR89].

2 Assumptions and construction

We now describe the assumptions used by our public key encryption scheme.

Notation. An (m, n, d)-graph is a bipartite graph $G = (V_L, V_R, E)$ with m vertices on the left side, n vertices on the right side, and left-degree equal to d (i.e., $|V_L| = m$, $|V_R| = n$ and $|E| = dm$). We will typically think of $m \geq n$. We say that an (m, n, d)-graph is *almost right regular* if the right-degree of each vertex is at most $2(m/n)d$. For $k \in \mathbb{N}$ and $e > 0$, we say that an (m, n, d)-graph is a (k, e)-*expander* if for every subset $S \subseteq V_L$ with $|S| \leq k$, $|\Gamma_G(S)| \geq e|S|$, where $\Gamma_G(S)$ denotes the set of neighbors of S in G. For $y \in \{0,1\}^m$ and $S \subseteq [m]$, we denote by y_S the projection of y to the coordinates in S. U_n denotes the uniform distribution on $\{0,1\}^n$.

Our assumptions. We assume that there exist $d_0 \in \mathbb{N}$ and $\alpha_0 > 1/2$ such that:

- **Assumption UPP:** *(Ultra-Parallel Pseudorandomness: Poly-stretch pseudorandom generators in* $\mathbf{NC^0}$*, with quasirandom graph structure.)* There exists a function $f : \{0,1\}^{d_0} \to \{0,1\}$ such that for every almost right regular $(n^{1.1}, n, d_0)$-graph G that is an $(n^{0.1}, \alpha_0 \cdot d)$ expander, the function G_f is a (cryptographically secure) pseudorandom generator, where G_f is the function mapping n bits to $n^{1.1}$ with the i^{th} output bit computed by applying f to the input bits corresponding to the neighbors of the vertex $i \in V_L$.[2]

- **Assumption UE:** *(Hardness of planted Unbalanced Expansion problem, for small sets.)* There exist two distributions $\mathcal{G}_Y$ and $\mathcal{G}_N$ on almost right regular $(n^{1.1}, n, d_0)$-graphs such that:

  1. If G is chosen from $\mathcal{G}_N$, then G is an $(n^{0.1}, \alpha_0 \cdot d)$-expander with probability $1 - 1/n$.
  2. If $G = (V_L, V_R, E)$ is chosen from $\mathcal{G}_Y$, then with probability 1, G is *not* an $(n^{0.1}, 1)$-expander. In particular there exists a set $S \subseteq V_L$ of size $O(\log n)$ such that $|\Gamma_G(S)| < |S|$. Moreover, there is an algorithm that samples elements of $\mathcal{G}_Y$ together with this set S.
  3. For every polynomial-time algorithm A,

  $$\Pr_{b \in_R \{Y,N\}, G \in_R \mathcal{G}_b}[A(G) = b] \leq 0.6 \tag{1}$$

  Note that distinguishing between any two distributions $\mathcal{G}_N$ and $\mathcal{G}_Y$ above can of course be easily done in $O(n^{|S|}) = n^{O(\log n)}$ time. The assumption is that no significant shortcut is possible.

Remark 2.1. Some of the parameters are fixed to certain values for convenience only. In particular, the degree d_0 doesn't have to be constant. It just seems that the smaller d_0 is, the more reasonable is the UE assumption. Similarly, the number m of vertices on the left side doesn't have to be $n^{1.1}$. However, we do need $m = \omega(n)$ to ensure that one cannot break the UE assumption by looking at the second eigenvalue.[3] The set-size in the expansion parameter also does not have to be $n^{0.1}$. However, it has to be $\omega(\log n)$, since if there is a set of size $O(\log n)$ that does not expand then there is a polynomial-time circuit that will break the corresponding pseudorandom generator.

[2] We assume that the set of edges touching each vertex is labeled with numbers in $[d_0]$, and so to obtain the i^{th} output we'll apply f to $x_1, \ldots, x_{d_0}$ where x_j is the input bit corresponding to the j^{th} neighbor of i.

[3] In fact, if $m = O(n)$ then one can break UE even using simpler algorithms such as counting the number of copies of $K_{2,2}$ in the graph, or finding the shrinking set S in $m2^{O(|S|)}$ time by guessing one of m possibilities for a vertex v in S and then enumerating all $2|S|$-length paths from v.

A concrete suggestion. We have the following candidates for the distributions $\mathcal{G}_N$ and $\mathcal{G}_Y$ of the UE assumption:

- $\mathcal{G}_N$ is just a random graph - each left side vertex is connected to d_0 random neighbors.
- $\mathcal{G}_Y$ is sampled by choosing a a graph from $\mathcal{G}_N$, and modifying it as follows. Choose a random k-sized subset S of the left side, a random $k-1$-sized subset T of the right side, and reconnect any member of S to d_0 random neighbors in T. (Note: another possibility is to connect S and T via a random bi-regular graph with left-degree d_0.)

Note that these candidates can be distinguished with bias $\Omega(1/\log n)$, since the probability that the resulting graph will contain $K_{2,3}$ will be $O(m^{2.2+3-6}) = O(m^{-0.8})$ in $\mathcal{G}_N$, but $\Omega(|S|^{-1}) = \Omega(1/\log n)$ in $\mathcal{G}_Y$.

For the purposes of our application, one can strengthen the UE assumption to require that these particular candidates $\mathcal{G}_Y$ and $\mathcal{G}_N$ are hard to tell apart. This will allow to relax the UPP assumption to require that $\mathcal{G}_N$ yields a pseudorandom generator with high probability (rather than *every* sufficiently good lossless expander). However, we believe the current description of the assumptions is cleaner.

2.1 Public key encryption

These assumptions allow us to obtain a public key encryption scheme:

Theorem 2.2. *If Assumptions* UPP *and* UE *are true, then there exists a semantically secure public-key encryption scheme.*

Proof. We construct a public-key encryption scheme for one-bit messages. This is obtained by repeating the following *basic scheme* several times.

Basic Scheme

Public encryption key: A random $(n^{1.1}, n, d)$-graph $G = (V_L, V_R, E)$ sampled from distribution $\mathcal{G}_Y$ of Assumption UE.

Private decryption key: Corresponding subset $S \subseteq V_L$ such that $|\Gamma_G(S)| < |S|$ that is obtained from the sampler of $\mathcal{G}_Y$.

Encryption: To encrypt 0 send a random string in $\{0,1\}^{n^{1.1}}$. To encrypt 1, choose $r \in_{\mathrm{R}} \{0,1\}^n$ and send $G_f(r)$ where G_f is the pseudorandom generator corresponding to G as per Assumption UPP.

Decryption: To decrypt $y \in \{0,1\}^{n^{1.1}}$, we output 1 if and only if $y_S = (G_f(r'))_S$ for some $r' \in \{0,1\}^{\Gamma_G(S)}$ (taking $2^{|\Gamma_G(S)|}$ time). Note that these $|S|$ outputs do indeed depend only on the at most $|S| - 1$ bits in locations specified by $\Gamma_G(S)$.

Thus we always successfully decrypt an encryption of 1, and successfully decrypt an encryption of 0 with probability at least $1 - 2^{|S|-1}/2^{|S|} \geq 1/2$. We can repeat this basic scheme several times to ensure that encryption is successful with high probability. Below we will repeat the basic scheme $2 \log n$ times, so that the decryption error is only at most $1/n^2$.

Because $\mathcal{G}_Y$ and $\mathcal{G}_N$ may be distinguished with constant probability, we need to use standard techniques to amplify the security of this basic encryption scheme. Thus, our full public key encryption scheme is obtained by choosing n independent keys $G_1, \ldots, G_n$ for the basic scheme, and encrypting a bit b by choosing $x_1, \ldots, x_n$ at random subject to $x_1 \oplus \cdots \oplus x_n = b$, and encrypting x_i using the key G_i. Note that decryption is still successful with high probability $\geq 1 - 1/n$.

The proof of security uses standard techniques originating from Yao's work on hardness amplification [Yao82]. The reader familiar with these is invited to skip the formal proof and proceed to the discussion section.

We now prove the security of the scheme in detail, and start with an high level overview. There are several ways to argue this, in line with several different proofs of the XOR lemma, and we'll utilize the one using Impagliazzo's hard-core set lemma [Imp95]. Specifically, will combine this lemma with Assumption UE to argue that there are sub-distributions $\mathcal{G}'_Y$ and $\mathcal{G}'_N$ with noticeable density in $\mathcal{G}_Y$ and $\mathcal{G}_N$ respectively, such that $\mathcal{G}'_Y$ and $\mathcal{G}'_N$ are computationally indistinguishable, and moreover, every graph in the support of $\mathcal{G}'_N$ is an $\alpha_0 \cdot d$ expander. We then note that if the key was chosen from $\mathcal{G}'_N$ instead of $\mathcal{G}_Y$ then by Assumption UPP, the encryption of 1 will be indistinguishable from the encryption of 0. Since with very high probability one of the n repetitions will fall in this sub-distribution, the security of the scheme follows.

Now we proceed formally. Suppose towards a contradiction that there exists a $T = \text{poly}(n)$-sized circuit A that distinguishes between an encryption of 1 and encryption of 0 with probability $\epsilon > 1/\text{poly}(n)$ (for infinitely many n's). That is,

$$\Pr[A(G_1, \ldots, G_n, Y_1, \ldots, Y_n) = b] \geq 1/2 + \epsilon \tag{2}$$

where the probability is over $G_1, \ldots, G_n$ chosen from $\mathcal{G}_Y$, b chosen in $\{0,1\}$, $x_1, \ldots, x_n$ chosen uniformly subject to $x_1 \oplus \cdots \oplus x_n = b$, and Y_i chosen as a random encryption of x_i using the key G_i according to the basic scheme.

By the UE Assumption, for every constant c, circuits of size $T' = cn^2T/\epsilon^2$ cannot distinguish between $\mathcal{G}_Y$ and $\mathcal{G}_N$ with probability better than 0.6. This implies that even if we let $\tilde{\mathcal{G}}_N$ equal $\mathcal{G}_N$ conditioned on the graph being an $\alpha_0 \cdot d$ expander, then still no T'-sized circuit can distinguish between $\mathcal{G}_Y$ and $\tilde{\mathcal{G}}_N$ with probability better than $0.6 + 1/n < 0.7$. Thus by the Hard-Core Lemma (e.g., see version in [Kal07]), there exists a distribution $\mathcal{D}$ such that (looking at distributions as vectors), $1/2\mathcal{G}_Y + 1/2\tilde{\mathcal{G}}_N = 0.3\mathcal{D} + 0.7\mathcal{D}'$, and for every T-sized circuit C,

$$\Pr_{G \in_{\mathrm{R}} \mathcal{D}}[C(G) = g(G)] \leq 1/2 + \epsilon/(100n), \tag{3}$$

where $g(G) = 1$ iff G is an $\alpha_0 \cdot d$ expander. In other words, if we let $\mathcal{D}_N = \mathcal{D}|g(G) = 1$ and $\mathcal{D}_Y = \mathcal{D}|g(G) = 0$, then the distributions $\mathcal{D}_N$ and $\mathcal{D}_Y$ are $\epsilon/(100n)$-indistinguishable for T-sized circuits. Note also that (3) implies that the event $g(G) = 1$ has probability roughly $1/2$ in $\mathcal{D}$. This means that we can express the distribution $\mathcal{G}_Y = p\mathcal{D}_Y + (1-p)\mathcal{D}''$ where $p \sim 0.3$, and so $\mathcal{G}_Y$ cannot be distinguished with probability $\epsilon/(100n)$ by T-sized circuits from the distribution $\mathcal{F} = p\mathcal{D}_N + (1-p)\mathcal{D}''$.

The above implies that Equation (2) above still holds (perhaps with ϵ replaced with 0.99ϵ) if the tuple of keys $G_1, \ldots, G_n$ was generated from the distribution $\mathcal{F}^n$ instead of $\mathcal{G}_Y^n$. The distribution $\mathcal{F}^n$ can be written as a convex combination of distributions that of the form $\mathcal{D}_1 \times \cdots \times \mathcal{D}_n$ where $\mathcal{D}_i$ is either $\mathcal{D}_N$ or $\mathcal{D}''$. Moreover, the coefficient of $\mathcal{D}''^n$ in this convex combination is exponentially small (i.e., $(1-p)^n$) and hence by an averaging argument there exists one distribution $\mathcal{E}$ of this form other than $\mathcal{D}''^n$ such that Equation (2) still holds (with ϵ replaced with, say, 0.98ϵ) even if the keys $G_1, \ldots, G_n$ are chosen from $\mathcal{E}$. There is one coordinate i in $\mathcal{E}$ where the key is chosen from $\mathcal{D}_N$. By another averaging argument there is a way to fix all keys G_j, values x_j, and encryptions Y_j for all coordinates j different from i so that Equation (2) still holds. Thus we get a T-sized circuit A' such that

$$\Pr_{G \in_{\mathrm{R}} \mathcal{D}_N, b \in_{\mathrm{R}} \{0,1\}}[A'(Y_{G,b}) = b] \geq 1/2 + \epsilon/2\,, \tag{4}$$

where $Y_{G,b}$ is the encryption of b using the key G in the basic scheme. But if G is chosen from $\mathcal{D}_N$ then it is a lossless expander, meaning that by the UPP Assumption the encryption of 0 ($2\log n$ independent random strings) is indistinguishable from the encryption of 1 ($2 \log n$ independent outputs of the pseudorandom generator). Thus we reached a contradiction. □

Note that assuming that the distribution $\mathcal{G}_N$ is sampleable as well (as is the case for our candidate distributions), the public key cryptosystem we construct has the property that one can generate a "bad public key" that looks indistinguishable from the valid public key, but does not allow the receiver to distinguish between the encryption of 0 and the encryption of 1. This means that this encryption system can be used to obtain an *oblivious transfer* protocol by following the framework of [EGL82]. (Thus implying also a secure protocol for any multi-party functionality [GMW87].)

2.2 Discussion

The biggest question raised by the above construction is of course to find out whether the UPP and UE assumptions are true. We present some preliminary findings in Sections 3 and 4. Another question is to what extent are our assumptions qualitatively different than previous constructions. We discuss this issue briefly here, although "qualitative difference" is of course a subjective term. 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]).

Comparing our assumptions to the ones used in prior works, it seems to us that the UPP Assumption is more of the "private key crypo" type, similar in nature to the kind of components or layers used in the design of block ciphers, hash functions, etc.. The UE Assumption also seems more combinatorial in nature than the kind of algebraic assumptions used before for the construction of public key encryption. But the question of connections between these assumptions and previous ones needs to be further investigated. One concrete open problem is whether or not our assumptions imply that there is a hard problem in the class **SZK** of *statistical zero knowledge proofs* (or more generally in the class $\mathbf{AM} \cap \mathbf{coAM}$). It seems that many previously known public key encryption schemes rely on the assumption that $\mathbf{SZK} \nsubseteq \mathbf{BPP}$. For example, this holds for discrete-log based schemes [GK90], as well as Lattice-based schemes relying on the hardness to approximate SVP/CVP within factors larger than $\sqrt{n}$ [GG98, AR04]. We do not know whether or not this was shown to hold for factoring-based schemes, though in any case these scheme do rely at least on the assumption that $\mathbf{NP} \cap \mathbf{coNP} \nsubseteq \mathbf{BPP}$.

Benny Applebaum (personal communication, July 2008) noted that since a function depending on d variables always has agreement at least 2^{-d} with some linear function, the UPP Assumption can be thought of as assuming hardness of learning parity with noise of magnitude roughly $1/2 - 2^{-d}$, though the noise added to different coordinates is not independent. In this sense the UPP assumption (and also the assumption of the existence of a large stretch pseudorandom generator in $\mathbf{NC^0}$) seems related to the assumption that learning noisy linear equations is hard. But, while the latter assumption was used to construct public key cryptosystems by Alekhnovich [Ale03] and Regev [Reg05], in both cases the magnitude of the noise is very small (less than $1/\sqrt{m}$ where m is the number of equations), as compared to a constant close to half in our case.[4] We believe (though this needs to be further investigated) that this quantitative difference translates to a *qualitative difference* between the assumptions. Some evidence for this belief is that in the worst-case, learning parity with constant noise is **NP**-hard [Hås97], while in contrast the security of [Reg05]'s cryptosystem relies on $\mathbf{SZK} \nsubseteq \mathbf{BPP}$ (though we don't know if that's the case for [Ale03]).

On a high level, our two assumptions can be seen as analogous to (often implicit) pairs of assumptions used in previous public key cryptosystems such as the (broken) Merkle-Hellman knapsack-based cryptosys-

[4] Regev's scheme used equations over a non-binary field— $\mathbb{F}_p$ for $p = O(m^2)$. The noise at each coordinate was a discrete Gaussian centered at 0 with standard deviation $o(p/\sqrt{m})$.

tem [MH78] and the (believed secure) McEliece coding-based cryptosystem [McE78]. We illustrate the analogy and difference between our assumptions and previous ones using the example of the McEliece cryptosystem. It seems (e.g., see [CFS01]) that the cleanest way to state the assumptions behind the latter system is that (1) it is hard to decode a codeword of random linear code that is corrupted in a small fraction of its coordinates and (2) there is a family $\mathcal{C}$ of efficiently decodable codes such it is hard to distinguish between the generating matrix of a random linear code and a generating matrix of a random code in $\mathcal{C}$ that is "scrambled" by changing the input to a random basis and permuting the coordinates. The best candidate for $\mathcal{C}$ seems to come from algebraic-geometric codes (also known as Goppa codes). Comparing the two constructions, our Assumption UPP is used in a way analogous to the way Assumption (1) is used in the McEliece system and our Assumption UE is analogous to Assumption (2). In fact, as we mentioned above, Assumption UPP can be viewed as a variant of Assumption (1). In contrast, it seems to us that Assumption (2) (which relies on the properties of the code family $\mathcal{C}$) is more "structured" than Assumption UE.[5] As mentioned above, [Ale03, Reg05] did construct public key cryptosystems that are based solely on variants on Assumption (1), but they needed the magnitude of noise to be very small (less than $1/\sqrt{m}$).

3 On the validity of Assumption UPP

Under widely believed assumptions, Applebaum et al [AIK04] show that there exists a pseudorandom generator mapping n bits to $n + n^{1/10}$ bits that can be computed in $\mathbf{NC^0}[4]$ (each output bit depending on 4 input bits). It has been conjectured that there exist pseudorandom generators in $\mathbf{NC^0}$ with output length $1.1n$ and even $n^{1.1}$ [AIK06, IKOS08]. Note that in the latter case we may assume without loss of generality that all the generator's outputs apply the same function to the input, since there is only a constant number of possible functions (and hence a generator with this property can be obtained by reducing the output by some constant factor). Currently the best negative result known is by Mossel et al [MST03], who showed that a generator where each output bit depends on d input bits cannot have output of length longer than $\tilde{O}(n^{d/2})$.

It seems reasonable that if such an $\mathbf{NC^0}$ generator exists, then it can be found by mapping the inputs to the outputs via a random graph. In fact, this assumption suffices for our construction. However we feel that the only quasirandomness property needed from the random graphs is lossless expansion, and so used this conjecture for Assumption UPP to allow a cleaner statement of the UE assumption. Note that the expansion constant α_0 in UPP has to be larger than $1/2$, since expansion smaller than $d/2$ is not (on its own) a sufficient condition for pseudorandomness. Indeed there are graphs with expansion $d/2$ that have two vertices with the same set of neighbors, which will correspond to having two output bits in the generator equalling one another with probability 1, thus violating pseudorandomness.

Assumption UPP is related to a candidate one-way function by Goldreich [Gol00], but is stronger than Goldreich's assumption in two ways: (1) we require that not only is it hard to invert the function G_f but in fact it is hard to distinguish its output from the uniform distribution, and (2) we require that output of G_f to be significantly larger than the input.

We now study to what extent the generator of Assumption UPP passes at least some very simple statistical tests (namely, those for which unconditionally secure pseudorandom generators are known to exist), and is likely pass others. These include sparse tests, linear tests, low degree polynomials and $\mathbf{AC^0}$ circuits. But first, we need to specify what function one should place at the vertices.

3.1 Resilient functions

Definition 3.1 (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 $a \in \{0,1\}^S$:

[5] If one uses a fixed good code C rather than a family then Assumption (2) is false if $\mathbf{SZK} \subseteq \mathbf{BPP}$ [PR97], and can also be broken in many practical cases by the heuristic of [Sen00].

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

For $\epsilon > 0$, 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 $[\epsilon, 1-\epsilon]$. 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 $1/2$. For Condition 2, note that for any fixing of at most 1/10 of the bits, when we choose at random all bits except for x_i (for i that is not fixed) then with probability $1/2$ we will have $y = z$, in which case the value of f will stay the same no matter whether x_i is equal to 0 or to 1. On the other hand, there's also a probability $1/2$ that we will have $y \neq z$, in which case changing the value of x_i will flip the value of f.

3.2 k-wise independence

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

Theorem 3.2. *Let G be an (m, n, d)-graph that is a $(k, (1-\epsilon)d)$ expander, and let f be a δ-resilient function for $\delta > 2\epsilon$. Then, the distribution $G_f(U_n)$ is k-wise independent.*

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

$$\Pr[\bigoplus_{i \in S} Y_i = 1] = 1/2 \tag{5}$$

Indeed, by a simple counting argument, there exists $i \in S$ such that $|\Gamma_G(i) \setminus \Gamma_G(S \setminus \{i\})| \geq (1-2\epsilon)d$. Therefore, if we fix all inputs in $\Gamma_G(S \setminus \{i\})$ (thus fixing Y_j for all $j \in S$ with $j \neq i$), then by the 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 $1/2$, establishing (5). □

Note that in this proof we only used Condition 1 of the definition of δ-resilient functions. In particular, Theorem 3.2 holds even if we use the *parity* function for f. (This was known before, see for example [MST03].)

3.3 Fooling $\mathbf{AC^0}$ circuits

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

3.4 Fooling linear tests

We now show that if G is 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).

Theorem 3.3. *Let G be an almost right regular $(n\ell, n, d)$-graph that is a $(k, (1-\epsilon)d)$-expander for $k > \omega(\ell^2)$. If f is δ resilient for $\delta > 2\epsilon$ then for every $S \subseteq [m]$,*

$$\Pr[\bigoplus_{i \in S} Y_i = 1] \in 1/2 \pm 2^{-\Omega(k/\ell^2)} , \tag{6}$$

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 (6) is implied by k-wise independence (i.e., Theorem 3.2). Let X_1 be an input vertex that is connected to S. Let S_1 be the set of at most $d\ell$ output vertices in S that are connected to X_1, let $V_1 = \Gamma_G(S_1)$ and let $S_1' = \Gamma_G(V_1)$ be the set of at most $2d\ell^2$ output vertices that share an input with a member of S_1. Remove S_1' from S and continue in this way to obtain $X_2, \ldots, X_t$ 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 $1/2$.

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

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 *every* 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 3.2. [MST03]'s second generator used a different construction and analysis than ours— they used a specific construction of locality two.

3.5 Fooling low degree $GF(2)$ polynomials

The last year has seen tremendous progress on constructing explicit distributions which fool not only polynomials of degree 1, but actually polynomials of degree up to $o(\log n)$. The best current result, Viola [Vio08] recently proved that the XOR of t independent samples from an ϵ-bias set fools degree t polynomials. Combining this with Theorem 3.3 (or the previous generator of [MST03]) we can obtain an $\mathbf{NC^0}$ pseudorandom generator fooling such polynomials by simply XOR'ing together several copies of our basic generator. Note however that the graph's structure resulting from this composition will not be any expander but rather have a special form (combining functions from disjoint pieces of the input). We note that this structure problem can be solved by considering non-Boolean inputs to the generator.

4 On the validity of Assumption UE

Assumption UE talks about the hardness of a combinatorial problem (graph expansion) that is **NP**-hard to solve exactly in the worst-case. But the variant of this problem that UE studies is more restricted (making UE much stronger) than the **NP**-complete variant in several important ways:

1. UE assumes hardness of an *average-case* problem, talking about planted distributions, rather than a worst-case problem.

2. UE talks about sets of *small* ($O(\log n)$) size, and so in particular the problem can be solved using $n^{O(\log n)}$ time.

3. UE is a *gap/approximation* problem, assuming the hardness of distinguishing between graphs that are almost ideal expanders (expansion close to d, in particular larger than $d/2$), and graphs that are rather bad expanders (expansion smaller than 1).

On its own, Property 1 (average-case hardness) is not that problematic. If we could show for example that this problem is **NP**-hard under Karp reductions, then if we assume that any **NP**-problem is hard on the average-case on some distribution X, this implies that our problem is hard on the distribution $f(X)$, where f is the reduction function. Moreover, if the reduction is of the standard type that maps witnesses to witnesses, then under the assumption that a one-way function exists (and hence we can sample hard-on-the-average **NP** problems together with the witnesses) we would have established UE fully, including the sampling condition. (Of course we don't really expect this problem to be **NP**-complete with a polynomial-time reduction, since it has an $n^{O(\log n)}$ time algorithm.)

We also have some evidence that on its own, Property 2 (small set size) is not that problematic as well. That is, using known results from the area of *parameterized complexity*, we can show that at least some worst-case *exact computation* variant of this problem is hard even for sets of small size. In fact, we show that under plausible assumptions, the problem cannot be solved in polynomial time for sets of non-constant size.

Property 3 (gap/approximation hardness) seems harder to argue, especially in conjunction with Property 2. While there are some hardness of approximation results for graph expansion, the gap they establish is not in the right region for us. As a partial result, we show in Theorem 4.3 that if a variant of the planted clique problem is hard, it is hard to distinguish between the case that a bipartite graph of (non-constant) degree d has expansion at least $d^{0.9}$ and the case that some set of size $k = O(\log n)$ has at most k/d^9 neighbors (hence having expansion $\ll 1$). We also give in Theorem 4.5 another reduction which relates the planted clique problem to distinguishing unique-neighbor expansion from non expansion.

These connections and others we suggest in the subsections ahead give some "circumstantial evidence" that the UE Assumption holds, as variants of this assumption (with different settings of parameters) are implied by the hardness of variants of some well studied computational problems.

4.1 Hypergraph formulation

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} left-vertex of G. In this formulation, the UE assumptions is about the hardness of distinguishing hypergraphs that contain a somewhat *dense* sub-hypergraph — a set T of $k-1$ vertices, such that the induced sub-hypergraph on T has at least k hyperedges— from graphs where the induced sub-hypergraph of every set of k vertices (for k up to roughly $n^{0.1}$ size or some other super-logarithmic bound) has only about k/d edges. Indeed, the task of distinguishing our candidates $\mathcal{G}_Y$ and $\mathcal{G}_N$ is essentially equivalent to the problem of distinguishing between a random fairly sparse hypergraph ($n^{1.1}$ hyperedges) and a random hypergraph with a planted somewhat *dense* (average degree larger than 1) small subgraph.[6] Indeed, the analog of this program for standard *graphs* (i.e., 2-uniform hypergraphs) has been studied by several works (e.g., [FPK01, Kho04]). This is known as the *densest k-subgraph* problem—finding a subgraph of k vertices with highest average degree. The variant of this problem where we ask for a subgraph of high *minimum* degree is fixed-parameter intractable [ASS08].

The following immediate observation relates the *clique* problem to expansion:

Lemma 4.1. *For every graph $G = (V, E)$, let $\hat{G}$ be the edge-vertex incidence graph of G.*[7] *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 $|\Gamma_{\hat{G}}(S)| \leq k$.*

Under plausible assumptions (that some problem in the class **SNP** that includes 3SAT and CLIQUE cannot be solved in $2^{o(n)}$ time), the k-clique problem for n-vertex graphs cannot be solved in $n^{o(k)}$ steps [FK97, CCF+04]. Thus under these assumptions a worst-case *exact computation* version of the expansion problem, where one needs to decide given (G, s, ℓ) whether there exists a set S of left-side vertices such that $|S| \leq s$ and $|\Gamma_G(S)| \leq \ell$, cannot be solved in $n^{o(\sqrt{s})}$ time.

4.2 Planted clique problem and expansion

A natural route to show hardness of the *approximation* or *gap* problem, is to use the **PCP** Theorem. Unfortunately, it seems that current **PCP** techniques are not well suited to showing hardness of problems with short witnesses. Moreover, we do not know how to establish the gap we need even for large sets using **PCP**. Following Feige [Fei02], we use instead assumptions on average-case hardness of certain problems. This has the added advantage of directly showing average-case hardness of our problem as well.

Definition 4.2 (Planted Clique problem)**.** For $n, k \in \mathbb{N}$ and $p \in (0, 1)$, the *planted k-clique problem in $G_{n,p}$* is to distinguish between the following two distributions:

1. $\mathcal{G}_N$ - the random graph $G_{n,p}$, where each edge is included with probability p.
2. $\mathcal{G}_Y$ - the random graph $G_{n,p}$ where in addition we select a k-subset S of $[n]$ at random and add all edges between pairs in S.

Note that for every $k > 2 \log n$ and $p \leq 1/2$, the planted k-clique problem in $G_{n,p}$ can be solved in time $n^{O(\log n)}$ since with high probability the maximum clique of a random graph will be at most $2 \log n$. The well-studied case is the planted k-clique problem in $G_{n,1/2}$, where there is an efficient algorithm for $k \sim \sqrt{n}$, but nothing non-trivial is known for, say $k = n^{0.4}$. We were not able to reduce the planted k-clique problem in $G_{n,1/2}$ to the UE problem, but rather only reduce the planted clique problem in $G_{n,p}$ for some $p \ll 1/2$ into a variant of the UE Problem.

Theorem 4.3. *Let $\epsilon > 0$ and suppose that the planted $\sqrt{\log n}$-clique problem in $G_{n,2^{-\log^{1-\epsilon} n}}$ is hard. Then, there exist two sampleable distributions $\mathcal{G}_Y$ and $\mathcal{G}_N$ over $(m, n, d = 2\log^{10\epsilon} n)$ for $m \sim n^2 2^{-\log^{1-\epsilon} n}$ that are computationally indistinguishable such that:*

1. *With high probability, a graph from $\mathcal{G}_N$ is a $(2^{\log^{0.9} n}, d^{0.8})$-expander.*
2. *Every graph from $\mathcal{G}_Y$ has a set of $\log n/3$ left vertices with at most $\log^{1/2+11\epsilon} n$ neighbors.*

[6] We say "essentially" because in the planted hypergraph problem the natural distribution would be not to fix the number of edges but rather to include each of the possible $\binom{n}{d}$ edges with probability $p = n^{1.1}/\binom{n}{d}$. This corresponds to the case where the size of the large side of the bipartite graph is not fixed but rather only concentrated around $n^{1.1}$.

[7] 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.

Proof. We reduce the planted clique problem to distinguishing between the above two cases. Let G be an instance of the planted $\sqrt{\log n}$-clique problem in $G_{n,2^{-\log^{1-\epsilon} n}}$. We let $\hat{G}$ be the $(m, n, 2)$-graph that is the edge-vertex incidence graph of G. If G had a clique of size $k = \sqrt{\log n}$ then $\hat{G}$ has a set of $\binom{k}{2} > \log n/3$ vertices that have only k neighbors. On the other hand, if G is a random element in $G_{n,2^{-\log^{1-\epsilon} n}}$, then the following lemma implies that with high probability $\hat{G}$ is a $(2^{\log^{0.9} n}, 1/(10\log^{\epsilon} n))$-expander:

Lemma 4.4. *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,

$$p_{k,t} \leq \binom{n}{k}\binom{k^2}{t}2^{-\ell t} \leq n^k k^{2t} p^t .$$

Taking logs we see that as long as

$$k \log n + 2t \log k \ll \ell t,$$

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 $t > (10 \log n/\ell)k$. □

Now make $d' = \log^{10\epsilon} n$ copies of every righthand side vertex u of $\hat{G}$ and connect these copies also to the same neighbors as u. The resulting graph G' has left-degree $d = 2d'$ and for every subset S of the left vertices of G', $|\Gamma_{G'}(S)| = d'|\Gamma_G(S)|$. Hence if G had a $\sqrt{\log n}$ clique then G' will have a set of size $\log n/3$ left-vertices with $d'\sqrt{\log n}$ neighbors, while if G was chosen from $G_{n,2^{-\log^{1-\epsilon} n}}$ then with high probability G' will be an $(2^{\log^{0.9} n}, d'/(10\log^{\epsilon} n))$-expander. □

The two drawbacks of Theorem 4.3 is that (1) the theorem only gives evidence of hardness of a variant of the UE problem where even in the NO case, the graph is not a lossless expander, and (2) even for this variant, the evidence is rather weak, since to our knowledge the planted clique problem in $G_{n,p}$ has been only extensively studied for constant p. It would be very interesting if there is a reduction from the planted clique problem in $G_{n,1/2}$ to (variants of) the UE problem, even if the reduction takes slightly superpolynomial time.

One reason why it may be hard to reduce the planted clique problem to our problem is that in the planted clique problem we do not know of any efficient distinguisher with non-negligible bias, while by looking at the existence of constant-sized subgraphs, we can distinguish with $\Omega(1/\log n)$ bias at least the specific candidates $\mathcal{G}_Y$ and $\mathcal{G}_N$ we put forward in Section 2. (Of course if UE is true then Impagliazzo's Hard-core Lemma tells us that there are other distributions $\mathcal{G}'_Y$ and $\mathcal{G}'_N$ that are cannot be distinguished with non-negligible advantage, but it does not guarantee that they are sampleable.)

4.3 Planted clique and unique neighbor expansion

As mentioned above, one drawback of Theorem 4.3 is that even in the NO case, the graphs obtained were not lossless expanders. We now show a different reduction, where in the NO case the graphs will have a property that is weaker than, but closely related to, lossless expansion. Alas, the size of the sets in the YES case will not be logarithmic but slightly super-logarithmic.

A central property of lossless expanders (with expansion $> d/2$ for sets of size $\leq k$) is the *unique neighbor* property, which is that for every subset S of the left-side of size at most k, there is a vertex $v \in \Gamma(S)$ with only one neighbor in S. Note that in particular this implies that $|\Gamma(S)| \geq |S|$. We say that G is a *unique neighbor expander for k-sets* if it has the unique neighbor property for sets of size $\leq k$. A generalization of the UE problem is to distinguish between unique neighbor expanders and graphs where some $O(\log n)$-sized set shrinks. We are able to relate this problem to the planted clique problem, but with a caveat— the set that shrinks will not be of $O(\log n)$ size but rather polylog(n) (in addition the degree will not be constant).

Theorem 4.5. *There is a polynomial-time reduction from the planted k-clique problem in $G_{n,2^{-\ell}}$ to distinguishing between two distributions $\mathcal{G}_Y$ and $\mathcal{G}_N$ over $(m, n', O(\log n))$ graphs, with $m = \Theta(n^2 2^{-\ell})$ and $n' = n\,\mathrm{polylog}(n)$ such that:*

1. *With high probability a graph G from $\mathcal{G}_Y$ is a unique neighbor expander for sets of size at most $2^{\ell/10}$.*
2. *With probability 1, a graph G from $\mathcal{G}_N$ will have a set of size k^2 with at most $O(\frac{k \log^2 n}{\ell})$ neighbors.*

Note that in particular this means that if, say, the planted $\log^{1+2\epsilon} n$-clique problem is hard in $G_{n,2^{-\log^{\epsilon} n}}$ then we can't distinguish between unique neighbor expanders for $2^{\log^{1-\epsilon} n/10}$-sets and graphs where there is a set of size $\log^{2+4\epsilon} n$ with at most $O(\log^{1+3\epsilon} n)$ neighbors.

Proof sketch for Theorem 4.5. 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 $G' : [m] \times ([2] \times [d']) \to [n] \times [100s] \times [d]$ 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 4.3. 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. □

Remark 4.6. We note that by itself the unique neighbor expansion property does not seem sufficient for pseudorandomness. It seems that there is a set S of many outputs with $d - 1$ of their neighbors in a small set T and only one unique neighbor outside T, then by guessing the inputs in T we will be able to predict many more of these outside neighbors. Nevertheless, it is possible that variants of this property could turn out useful for variants of the construction. Moreover, the unique-neighbor property seems so closely related to lossless expansion, that studying it may shed light on the truth of the UE assumption.

Remark 4.7. Other problems that seem closely related to the UE 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 an $O(\log n)$-sized set S with $< |S|$ neighbors and (2) *search variant* show an algorithm that given every graph with an $O(\log n)$-sized set S with $< |S|$ neighbors finds a subset S' of size $O(\log^2 n)$ such that S' has no unique neighbors.

5 Conclusions

The two main challenges are to investigate more the validity of our assumptions, and improve the efficiency and security of the encryption. Toward the latter end, in a follow-up work joint with Benny Applebaum, we consider a variant of our encryption where the non-linear function is replaced with parity with noise. Under plausible assumptions, this variant can achieve sub-exponential security. A related advantage of this variant is that it enables making different tradeoffs between the parameters of the UPP and UE assumptions.

Acknowledgements. We thank Benny Applebaum for useful discussions.

References

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

[AIK04] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput*, 36(4):845–888, 2006. Prelim version FOCS' 04.

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

[Ale03] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307. IEEE Computer Society, 2003.

[AR04] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. *J. ACM*, 52:749–765, 2005. Prelim version FOCS '04.

[ASS08] O. Amini, I. Sau, and S. Saurabh. Parameterized complexity of the smallest degree-constrained subgraph problem. In *IWPEC*, volume 5018 of *Lecture Notes in Computer Science*, pages 13–29. Springer, 2008.

[AW85] M. Ajtai and A. Wigderson. Deterministic simulation of probabilistic constand-depth circuits. *Advances in Computing Research - Randomness and Computation*, 5:199–223, 1989. Prelim version FOCS '85.

[Baz07] L. Bazzi. Polylogarithmic independence can fool DNF formulas. In *FOCS*, pages 63–73. IEEE Computer Society, 2007.

[CCF$^+$04] J. Chen, B. Chor, M. Fellows, X. Huang, D. W. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput*, 201(2):216–231, 2005. Prelim version CCC '04.

[CFS01] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a mceliece-based digital signature scheme. *Proceedings of AsiaCrypt*, pages 157–174, 2001.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(5):644–654, Nov. 1976.

[EGL82] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985. Prelim version CRYPTO '82.

[Fei02] U. Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543, New York, May 19–21 2002. ACM Press.

[FK97] U. Feige and J. Kilian. On limited versus polynomial nondeterminism. *Chicago Journal of Theoretical Computer Science*, Mar. 1997.

[FPK01] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[GG98] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000. Prelim year STOC '98.

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

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game — A completeness theorem for protocols with honest majority. In *STOC*, pages 218–229. ACM, 1987.

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

[Hås97] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. Prelim version STOC '97.

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

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *FOCS*, pages 538–544. IEEE, 1995.

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

[Kal07] S. Kale. Boosting and hard-core set constructions: a simplified approach. Technical Report Report TR07-131, ECCC, 2007.

[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.

[Lip97] H. Lipmaa. Cryptology pointers: Public key cryptography: Concrete systems, 1997. Web site, url: `http://www.adastral.ucl.ac.uk/~helger/crypto/link/public/concrete.php`.

[LN90] N. Linial and N. Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10:349–365, 1990. Prelim version STOC 90.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *The Deep Space Network Progress Report, DSN PR 42-44, January and February 1978,*, 1978.

[MH78] R. C. Merkle and M. E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, IT-24:525–530, 1978.

[Mil85] V. S. Miller. Use of elliptic curves in cryptography. In *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 18–22 Aug. 1985.

[MST03] E. Mossel, A. Shpilka, and L. Trevisan. On epsilon-biased generators in NC^0. *Random Struct. Algorithms*, 29(1):56–81, 2006. Prelim version FOCS '03.

[NY89] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43, Seattle, 1989. ACM.

[PR97] E. Petrank and R. M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43(5):1602–1604, 1997.

[Rab79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology, Jan. 1979.

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

[Rom90] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, New York, 1990. ACM.

[RSA78] R. L. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RVW00] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *FOCS*, pages 3–13. IEEE, 2000.

[Sen00] N. Sendrier. Finding the permutation between equivalent linear codes: thesupport splitting algorithm. *Information Theory, IEEE Transactions on*, 46(4):1193–1203, 2000.

[Vio08] E. Viola. The sum of d small-bias generators fools polynomials of degree d. In *IEEE Conference on Computational Complexity*, pages 124–127. IEEE Computer Society, 2008.

[Yao82] A. C. C. Yao. Theory and applications of trapdoor functions. In *FOCS*, pages 80–91. IEEE, 3–5 Nov. 1982.

[Zhu01] H. Zhu. Survey of computational assumptions used in cryptography broken or not by Shor's algorithm. Master's thesis, School of Computer Science McGill University, 2001.