The Complexity of Public-Key Cryptography

Boaz Barak

April 27, 2017

Abstract

We survey the computational foundations for public-key cryptography. We discuss the computational assumptions that have been used as bases for public-key encryption schemes, and the types of evidence we have for the veracity of these assumptions.

1 Introduction

Let us go back to 1977. The first (or fourth, depending on your count) "Star Wars" movie was released, ABBA recorded "Dancing Queen" and in August, Martin Gardner described in his *Scientific American* column the RSA cryptosystem [RSA78], whose security relies on the difficulty of integer factoring. This came on the heels of Diffie, Hellman, and Merkle's 1976 invention of public-key cryptography and the discrete-logarithm based Diffie–Hellman key exchange protocol [DH76b].

Now consider an alternative history. Suppose that, in December of that year, a mathematician named Dieter Chor discovered an efficient algorithm to compute discrete logarithms and factor integers. One could imagine that, in this case, scientific consensus would be that there is something *inherently impossible* about the notion of public-key cryptography, which anyway sounded "too good to be true". In the ensuing years, people would occasionally offer various alternative constructions for public-key encryption, but, having been burned before, the scientific and technological communities would be wary of adapting them, and treat such constructions as being *insecure* until proven otherwise.

This alternative history is of course very different from our own, where public-key cryptography is a widely studied and implemented notion. But are the underlying scientific facts so different? We currently have no strong evidence that the integer factoring and discrete logarithm problems are actually hard. Indeed, Peter Shor [Sho97] has presented an algorithm for this problem that runs in polynomial time on a so-called "quantum computer". While some researchers (including Oded Goldreich [Gole, Golg]) have expressed deep skepticism about the possibility of physically implementing this model, the NSA is sufficiently concerned about this possibility to warn that government and industry should transition away from these cryptosystems in the "not too far future" [NSA15]. In any case we have no real justification to assume the nonexistence of a *classical* (i.e., not quantum) algorithm for these problems, especially given their strong and not yet fully understood mathematical structure and the existence of highly non trivial subexponential algorithms [LLJMP90, COS86].

In this tutorial I want to explore the impact on the *theory* of cryptography of such a hypothetical (or perhaps not so hypothetical) scenario of a breakthrough on the discrete logarithm and

factoring problems, and use this as a launching pad for a broader exploration of the role of hardness assumptions in our field. I will discuss not just the *mathematical* but also the *social* and *philosophical* aspects of this question. Such considerations play an important role in any science, but especially so when we deal with the question of which unproven assumptions we should believe in. This is not a standard tutorial or a survey, in the sense that it is more about *questions* than *answers*, and many of my takes on these questions are rather subjective. Nevertheless, I do think it is appropriate that students or anyone else who is interested in research on the foundations of cryptography consider these types of questions, and form their own opinions on the right way to approach them.

Acknowledgements. This survey is written in honor of Oded Goldreich's 60th birthday. I was first exposed to the beauty of the foundations of cryptography through Oded, and while we may not always agree on specific issues, his teachings, writing, and our discussions have greatly influenced my own views on this topic. Oded wrote many essays worth reading on issues related to this survey, such as subjectivity and taste in science [Gola], computational assumptions in cryptography [Gold, Golb], as well as the distinction between pure and applied (or "intellectual" versus "instrumental") science [Golc, Golf]. I also thank Benny Applebaum, Nir Bitansky, and Shai Halevi for extremely insightful comments on earlier versions of this survey that greatly improved its presentation.

1.1 What Is Special About Public-Key Cryptography?

Perhaps the first instance of an unjustified subjective judgment in this survey is my singling out of the integer factoring and discrete logarithm problems, as well as other "public key type" assumptions, as particularly deserving of suspicion. After all, given that we haven't managed to prove $P \neq NP$, essentially *all* cryptographic primitives rest on unproven assumptions, whether it is the difficulty of factoring, discrete log, or breaking the AES cipher. Indeed, partially for this reason, much of the work on theoretical cryptography does not deal directly with particular hard problems but rather builds a *web of reductions* between different primitives. Reduction-based security has been a resounding success precisely because it allows to *reduce* the security of a great many cryptographic constructions to a relatively small number of simple-to-state and widely studied assumptions. It helped change cryptography from an alchemy-like activity which relied on "security by obscurity" to a science with well-defined security properties that are obtained under precisely stated conjectures, and is often considered the strongest component in secure applications.

Given the above, one can think of the canonical activity of a theoretical cryptographer as constructing a new (typically more sophisticated or satisfying stricter security notions) cryptographic primitive from an old primitive (that would typically be simpler, or easier to construct).[1] The "bottommost layer" of such primitives would have several *candidate constructions* based on various hardness assumptions, and new developments in cryptanalytic algorithms would simply mean that we have one fewer candidate.

The intuition above is more or less accurate for *private-key cryptography*. Over the last three decades, cryptographers have built a powerful web of reductions showing constructions of a great

[1] For example, by my rough count, out of the nearly 800 pages of Goldreich's two-volume canonical text [Gol01, Gol04], fewer than 30 deal with concrete assumptions.

many objects from the basic primitive of *one-way functions*.[2] And indeed, as discussed in Section 2 below, we do have a number of candidate constructions for one way functions, including not just constructions based on factoring and discrete logarithms, but also constructions based on simple combinatorial problems such as planted clique [JP00], random SAT [AC08], Goldreich's expander-based candidate [Gol11], as well as the many candidate block ciphers, stream ciphers, and hash functions such as [DR13, NIS02, Ber08, BDPVA11] that are widely used in practice and for many of which no significant attacks are known despite much cryptanalytic effort.

However, for *public-key* cryptography, the situation is quite different. There are essentially only two major strains of public-key systems.[3] The first family consists of the "algebraic" or "group-theoretic" constructions based on integer factoring and the discrete logarithm problems, including the Diffie–Hellman [DH76b] (and its elliptic curve variants [Mil85, Kob87]), RSA [RSA78], Rabin [Rab79], Goldwasser–Micali [GM82] schemes and more. The second family consists of the "geometric" or "coding/lattice"-based systems of the type first proposed by McEliece [McE78] (as well as the broken Merkle-Hellman knapsack scheme [MH78]). These were invigorated by Ajtai's paper on *lattices* [Ajt96], which was followed by the works of Ajtai-Dwork [Aw97], Goldreich-Goldwasser-Halevi [GGH97], and Hoffstein et al. [HPS98] giving *public-key* systems based on lattices, and by the later work of Regev [Reg09] who introduced the *Learning With Errors (LWE)* assumption and showed its equivalence to certain hardness assumptions related to lattices.[4]

The known classical and quantum algorithms call into question the security of schemes based on the algebraic/group-theoretic family. After all, as theoreticians, we are interested in schemes for which efficient attacks are not merely *unknown* but are *nonexistent*. There is very little evidence that this first family satisfies this condition. That still leaves us with the second family of lattice/coding-based systems. Luckily, given recent advances, there is almost no primitive achieved by the group-theoretic family that cannot be based on lattices, and in fact many of the more exciting recent primitives, such as fully homomorphic encryption [Gen09] and indistinguishability obfuscation [GGH$^+$13], are only known based on lattice/coding assumptions.

If, given these classical and quantum algorithms, we do not want to trust the security of these "algebraic"/"group theoretic" cryptosystems, we are left in the rather uncomfortable situation where all the edifices of public-key cryptography have only one foundation that is fairly well studied, namely the difficulty of lattice/coding problems. Moreover, one could wonder whether talking about a "web of abstractions" is somewhat misleading if, at the bottommost layer, every primitive has essentially only a single implementation. This makes it particularly important to find out whether pubic key cryptography can be based on radically different assumptions. More generally, we would like to investigate the "assumption landscape" of cryptography, both in terms of concrete assumptions and in terms of relations between different objects. Such questions have of course interested researchers since the birth of modern cryptography, and we will review in this tutorial some of the discoveries that were made, and the many open questions that still remain.

[2] These include some seemingly *public-key* notions such as digital signatures which were constructed from one-way functions using the wonderful and surprising notion of pseudorandom functions put forward by Goldreich, Goldwasser, and Micali [GGM86], as well as universal one-way hash functions [NY89, Rom90].

[3] I think this is a fair statement in terms of all systems that have actually been implemented and widely used (indeed by the latter metric, one might say there is only one major strain). However, as we will discuss in Section 5 below, there have been some alternative suggestions, including by this author.

[4] Admittedly, the distinction into "geometric" versus "algebraic" problems is somewhat subjective and arbitrary. In particular, lattices or linear codes are also Abelian groups. However, the type of problems on which the cryptographic primitives are based are more geometric or "noisy" in nature, as opposed to the algebraic questions that involve exact group structure.

Remark 1.1. One way to phrase the question we are asking is to understand what type of *structure* is needed for public-key cryptography. One-way functions can be thought of as a completely unstructured object, both in the sense that they can be implemented from any hard-on-the-average search or "planted" problem [IL89], as well as that they directly follow from functions that have pseudorandom properties. In contrast, at least at the moment, we do not know how to obtain public-key encryption without assuming the difficulty of *structured* problems, and (as discussed in Remark 3.1) we do not know how to base public-key encryption on private-key schemes. The extent to which this is inherent is the topic of this survey; see also my survey [Bar14] for more discussion on the role of structure in computational difficulty.

1.2 Organization

In the rest of this tutorial we discuss the assumption landscape for both private and public-key cryptography (see Sections 2 and 3, respectively). Our emphasis is not on the most efficient schemes, nor on the ones that provide the most sophisticated security properties. Rather we merely attempt to cover a sample of candidate constructions that represents a variety of computational hardness assumptions. Moreover, we do not aim to provide full mathematical descriptions of those schemes— there are many excellent surveys and texts on these topics— but rather focus on their *qualitative* features.

Many of the judgment calls made here, such as whether two hardness assumptions (that are not known to be equivalent) are "similar" to one another, are inherently *subjective*. Section 6 is perhaps the most subjective part of this survey , where we attempt to discuss what it is about a computational problem that makes it hard.

2 Private-Key Cryptography

Before talking about public-key cryptography, let us discuss *private-key* cryptography, where we have a much cleaner theoretical and practical picture of the landscape of assumptions. The fundamental theoretical object of private-key cryptography is a *one-way function*:

Definition 1 (One-way function)**.** A function $F : \{0,1\}^* \to \{0,1\}^*$ is a *one-way function* if there is a polynomial-time algorithm mapping $r \in \{0,1\}^*$ to $F(r)$ and for every probabilistic polynomial-time algorithm A, constant c, and sufficiently large n,

$$\Pr_{w=F(r);r\leftarrow_R\{0,1\}^n}[F(A(w)) = w] < n^{-c} .$$

We denote by OWF the conjecture that one-way functions exist.

While a priori the definition of one-way functions does not involve any secret key, in a large body of works it was shown (mainly through the connection to psuedorandomness enabled by the Goldreich–Levin theorem [GL89]) that OWF is *equivalent* to the existence of many cryptographic primitives including:

- Pseudorandom generators [HILL99]
- Pseudorandom functions and message authentication codes [GGM86]

- Digital signatures [Rom90][5]
- Commitment schemes [Nao91].
- Zero knowledge proofs for every language in NP [GMW87].[6]

(See Goldreich's text [Gol01, Gol04] for many of these reductions as well as others.)

Thus, OWF can be thought of as the central conjecture of private-key cryptography. But what is the evidence for the *truth* of this conjecture?

2.1 Candidate Constructions for One-Way Functions

> *"From time immemorial, humanity has gotten frequent, often cruel, reminders that many things are easier to do than to reverse"*, Leonid Levin.

The main evidence for the OWF conjecture is that we have a great number of candidate constructions for one-way functions that are potentially secure. Indeed, it seems that "you can't throw a rock without hitting a one-way function" in the sense that, once you cobble together a large number of simple computational operations then, unless the operations satisfy some special property such as linearity, you will typically get a function that is hard to invert (indeed, people have proposed some formalizations of this intuition, see Sections 2.1.4 and 2.1.5). Here are some example candidate constructions for one-way functions:

2.1.1 Block Ciphers, Stream Ciphers and Hash Functions

Many practical constructions of symmetric key primitives such as block ciphers, stream ciphers, and hash functions are believed to satisfy the security definitions of pseudorandom permutations, pseudorandom generators, and collision-resistant hash functions, respectively. All these notions imply the existence of one-way functions, and hence these primitives all yield candidate one-way functions. These constructions (including DES, AES, SHA-x, etc.) are typically described in terms of a fixed finite input and key size, but they often can be naturally generalized (e.g., see [MV12]). Note that practitioners often require very strong security from these primitives, and any attack faster than the trivial 2^n (where n is the key size, block size, etc.) is considered a weakness. Indeed, for many constructions that are considered weak or "broken", the known attacks still require exponential time (albeit with an exponent much smaller than n).[7]

[5] While from the perspective of applied cryptography, digital signatures are part of *public-key* cryptography, from our point of view of computational assumptions, they belong in the private-key world. We note that the current constructions of digital signatures from symmetric primitives are rather inefficient, and there are some negative results showing this may be inherent [BM07].

[6] Actually, zero-knowledge proofs for languages outside of $\mathbf{P}$ imply a slightly weaker form of "non-uniform" one-way functions, see [OW93].

[7] The claim that is easy to get one-way functions might seem contradictory to the fact that there have been successful cryptanalytic attacks even against cryptographic primitives that were constructed and widely studied by experts. However, practical constructions aim to achieve the best possible efficiency versus security tradeoff, which does require significant expertise. If one is fine with losing, say, a factor 100 in the efficiency (e.g., build a 1000-round block cipher instead of a 10-round one), then the task of constructing such primitives becomes significantly easier.

2.1.2 Average Case Combinatorial Problems: Planted SAT, Planted Clique, Learning Parity with Noise

A *planted distribution* for an **NP** problem can be defined as follows:

Definition 2 (**NP** relations and planted problems)**.** A relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is an **NP** *relation* if there is a polynomial $p(\cdot)$ such that $|y| \leq p(|x|)$ for every $(x,y) \in R$ and there is a polynomial-time algorithm M that on input (x,y) outputs 1 iff $(x,y) \in R$.

A probabilistic polynomial-time algorithm G is a *sampler* for R if, for every n, $G(1^n)$ outputs with probability 1 a pair (x,y) such that $(x,y) \in R$.

We say that an algorithm A solves the *planted problem corresponding to* (G,R) if, for every n, with probability at least 0.9, $(x, A(x)) \in R$ where (x,y) is sampled from $G(1^n)$.

We say that the planted problem corresponding to (G,R) is *hard* if there is no probabilistic polynomial-time algorithm that solves it.

The following simple lemma shows that hard planted problems imply the OWF conjecture:

Lemma 2.1. *Suppose that there exists a hard planted problem (G,R). Then there exists a one-way function.*

Proof. We will show that a hard planted problem implies a *weak* one-way function, which we define here as a function F such that for every probabilistic polynomial-time A and sufficiently large m,

$$\Pr_{x=F(r);r\leftarrow_R\{0,1\}^m}[F(A(x)) = x] < 0.9 \; . \tag{1}$$

That is, we only require that an adversary fails to invert the function with probability larger than 90%, as opposed to nonnegligible probability as required in Definition 1. It is known that the existence of weak one-way functions implies the existence of strong ones (e.g., see [IL89],[Gol01, Sec 2.3]). Let G be a generator for a hard planted problem and let R be the corresponding relation. By padding, we can assume without loss of generality that the number of coins that G uses on input 1^n is n^c for some integer $c \geq 1$. For every $r \in \{0,1\}^*$, we define $F(r) = x$ where $(x,y) = G(1^n; r_1 \ldots r_{n^c})$ where $n = \lfloor |r|^{1/c} \rfloor$, and $G(1^n; r)$ denotes the output of G on input 1^n and coins r.

We now show that F is a weak one-way function. Indeed, suppose towards a contradiction that there exists a probabilistic polynomial-time algorithm A violating (1) for some sufficiently large m, and let $n = \lfloor m^{1/c} \rfloor$. This means that

$$\Pr_{(x,y)=G(1^n;r_{1,\ldots,n});r\leftarrow_R\{0,1\}^m}[G(1^n; A(x)) = x] \geq 0.9 \; ,$$

which in particular implies that, if we let $A'(x) = G(1^n; A(x))$, then with probability at least 0.9, $A'(x)$ will output a pair (x', y') with $x' = x$ and $(x', y') \in R$ (since the latter condition happens with probability 1 for outputs of G). Hence we get a polynomial-time algorithm to solve the planted problem with probability at least 0.9 on length n inputs. □

Using this connection, there are several natural planted problems that give rise to candidate one way functions:

The planted clique problem: It is well known that, in a random Erdős–Rényi graph $G_{n,1/2}$ (where every pair gets connected by an edge with probability 1/2), the maximum clique size will be $(2-o(1))\log n$ [GM75, BE76]. However, the greedy algorithm will find a clique of only $1 \cdot \log n$ size, and Karp asked in 1976 [Kar76] whether there is an efficient algorithm to find a clique of size $(1+\epsilon)\log n$. This remains open till this day. In the 1990s, Jerrum [Jer92] and Kucera [Kuc95] considered the easier variant of whether one can find a clique of size $k \gg \log n$ that has been *planted* in a random graph by selecting a random k-size set and connecting all the vertices in it. The larger k is, the easier the problem, and at the moment no polynomial-time algorithm is known for this question for any $k = o(\sqrt{n})$. By the above discussion, if this problem is hard for any $k > 2\log n$, then there exists a one-way function. Juels and Peinado [JP00] showed that, for $k = (1+\epsilon)\log n$, the planted distribution is *statistically close* to the uniform distribution. As a result there is a hard planted distribution (and hence a one-way function) as long as the answer to Karp's question is negative.

Planted constraint satisfaction problems: A (binary alphabet) *constraint satisfaction problem* is a collection of functions $C = \{C_1, \ldots, C_m\}$ mapping $\{0,1\}^n$ to $\{0,1\}$ such that every function C_i depends on at most a constant number of the input bits. The *value* of C w.r.t. an assignment $x \in \{0,1\}^n$ is defined as $\frac{1}{m}\sum_{i=1}^{m} C_i(x)$. The *value* of C is its maximum value over all assignments $x \in \{0,1\}^n$.

There are several models for random constraint satisfaction problems. One simple model is the following: for every predicate $P : \{0,1\}^k \to \{0,1\}$ and numbers n, m, we can select $C_1, \ldots, C_m$ by choosing every C_i randomly and independently to equal $P(y_1, \ldots, y_k)$ where $y_1, \ldots, y_k$ are random and independent *literals* (i.e., equal to either x_j or to $1 - x_j$ for some random j). Using standard measure concentration results, the following can be shown:

Lemma 2.2. *For predicate $P : \{0,1\}^k \to \{0,1\}$ and every $\epsilon > 0$ there exists some constant α (depending on k, ϵ) such that, if $m > \alpha n$ and $C = (C_1, \ldots, C_m)$ is selected at random from the above model, then with probability at least $1 - \epsilon$, the value of C is in $[\mu - \epsilon, \mu + \epsilon]$ where $\mu = \mathbb{E}_{x \leftarrow_R \{0,1\}^k}[P(x)]$.*

There are several *planted* models where, given $x \in \{0,1\}^n$, we sample at random an instance C such that the value of C w.r.t. x is significantly larger than μ. Here is one model suggested in [BKS13]:

Definition 3. Let P, n, m be as above, let $x \in \{0,1\}^n$, and D be some distribution over $\{0,1\}^k$. The (D, δ, x) planted model for generating a constraint satisfaction problem is obtained by repeating the following for $i = 1, \ldots, m$: with probability δ sample a random constraint C_i as above; otherwise sample a string d from D, and sample $y_1, \ldots, y_k$ to be random literals as above *conditioned* on the event that these literals applied to x yield d, and let C_i be the constraint $P(y_1, \ldots, y_k)$.

Analogously to Lemma 2.2, if C is sampled from the (D, δ, x) model, then with high probability the value of C w.r.t. x will be at least $(1-\delta)\mu_D - \epsilon$ where $\mu_D = \mathbb{E}_{x \leftarrow_R D}[P(x)]$. If $\mu_D > \mu$, then we can define the planted problem as trying to find an assignment with value at least, say, $\mu_D/2 + \mu/2$. [BKS13] conjectured that this planted problem is hard as long as D is a *pairwise independent* distribution. This conjecture immediately gives rise to many candidate one-way functions based on predicates such as k-XOR, k-SAT, and more.

It was shown by Friedgut [Fri99] that every random constraint satisfaction problem satisfies a *threshold* condition in the sense that, for every ϵ, as n grows, there is a value $m(n)$ such that the

probability that a random instance of $(1-\epsilon)m(n)$ constraints has value 1 is close to 1, while the probability that a random instance of $(1+\epsilon)m(n)$ has value 1 is close to 0. It is widely believed that the value $m(n)$ has the value $\alpha^* n$ for some constant α^* depending on the problem (and are concrete guesses for this constant for many predicates) but this has not yet been proven in full generality and in particular the case of $3SAT$ is still open. It is also believed that, for k sufficiently large (possibly even $k = 3$ is enough), it is hard to find a satisfying (i.e., value 1) assignment for a random k-SAT constraint satisfaction problem where $m(n)$ is very close (but below) the threshold. Using a similar reasoning to [JP00] (but much more sophisticated techniques), Achlioptas and Coja-Oghlan [AC08] showed that this conjecture implies the hardness of a certain planted variant and hence yields another candidate for one-way functions.

2.1.3 Unsupervised learning and Distributional One-Way Functions

Unsupervised learning applications yield another candidate for one-way functions. Here one can describe a *model* M as a probabilistic algorithm that, given some parameters $\theta \in \{0,1\}^n$, samples from some distribution $M(\theta)$. The models studied in machine learning are all typically efficiently computable in the forward direction. The challenge is to solve the *inference* problem of recovering θ (or some approximation to it) given s independent samples $x_1, \ldots, x_s$ from $M(\theta)$.[8]

Consider s that is large enough so that the parameters are *statistically identifiable*.[9] For simplicity, let us define this as the condition that, for every θ, with high probability over the choice of $x = (x_1, \ldots, x_s)$ from $M(\theta)$, it holds that

$$P_{\theta'}(x) \ll 2^{-n} P_\theta(x) \tag{2}$$

for every $\theta' \neq \theta$, where for every set of parameters ϑ and $x = (x_1, \ldots, x_s)$,

$$P_\vartheta(x) = \prod_{i=1}^{s} \Pr[M(\vartheta) = x_i] \; .$$

Now, suppose that we had an algorithm A that, given $x = (x_1, \ldots, x_s)$, could sample uniformly from the distribution on uniform parameters θ' and random coins $r_1, \ldots, r_s$ conditioned on $M(\theta'; r_i) = x_i$ for all $i \in \{1, \ldots, s\}$. Then (2) implies that, if the elements in x itself were sampled from $M(\theta)$ then with probability $1 - o(1)$ the first output θ' of A will equal θ. Thus, if there is a number of samples s where the unsupervised learning problem for M is statistically identifiable but computationally hard, then the process $\theta, r_1, \ldots, r_s \mapsto M(\theta; r_1), \ldots, M(\theta; r_s)$ is hard to invert in this distributional sense. But Impagliazzo and Luby [IL89] showed that the existence of not just weak one-way functions but even *distributional one-way functions* implies the existence of standard one-way functions, and hence any computationally hard unsupervised learning problem yields such a candidate.

The *Learning Parity with Noise* (LPN) problem is one example of a conjectured hard unsupervised learning problem that has been suggested as a basis for cryptography [GKL93, BFKL93]. Here the parameters of the model are a string $x \in \{0,1\}^n$ and a sample consists of a random

[8] This is a very general problem that has been considered in other fields as well, often under the name "parameter estimation problem" or "inverse problem", e.g., see [Tar05].

[9] In many applications of machine learning, the parameters come from a continuous space, in which case they are typically only identifiable up to a small error. For simplicity, we ignore this issue here, as it is not very significant in our applications.

$a \in \{0,1\}^n$ and a bit $b = \langle a, x \rangle + \eta \pmod 2$, where η is chosen to equal 0 with probability $1 - \delta$ and 1 with probability δ for some constant $\delta > 0$. Using concentration of measure one can show that this model is statistically identifiable as long as the number of samples s is at least some constant times n, but the best known "efficient" algorithm requires $\exp(\Theta(n/\log n))$ samples and running time [BKW03] ([Lyu05] improved the number of samples at the expense of some loss in error and running time). Thus, if this algorithm cannot be improved to work in an optimal number of samples and polynomial time, then one-way functions exist.[10]

2.1.4 Goldreich's One-Way Function Candidate

Goldreich has proposed a very elegant concrete candidate for a one-way function [Gol11] which has caught several researchers' interest. Define an (n, m, d) graph to be a bipartite graph with n left vertices, m right vertices, and right degree d. Goldreich's function $Gol_{H,P} : \{0,1\}^n \to \{0,1\}^m$ is parameterized by an (n, m, d) graph H and a predicate $P : \{0,1\}^d \to \{0,1\}$. For every $x \in \{0,1\}^m$ and $j \in [m]$, the j^{th} output bit of Goldreich's function is defined as $Gol_{H,P}(x)_j = P(x_{\overleftarrow{\Gamma}_H(j)})$, where we denote by $\overleftarrow{\Gamma}_H(j)$ the set of left-neighbors of the vertex j in H, and x_S denotes the restriction of x to the coordinates in S.

Goldreich conjectured that this function is one way as long as P is sufficiently "structureless" and H is a sufficiently good expander. Several follow-up works showed evidence for this conjecture by showing that it is not refuted by certain natural families of algorithms [CEMT09, Its10]. Other works showed that one needs to take care in the choice of the predicate P and ensure that it is *balanced*, as well as not having other properties that might make the problem easier [BQ12]. Later works also suggested that Goldreich's function might even be a *pseudorandom generator* [ABW10, App12, OW14]. See Applebaum's survey [App15] for more about the known constructions, attacks, and (several surprising) applications of Goldreich's function and its variants.

2.1.5 Random Circuits

Perhaps the most direct formalization of the intuition that if you cobble together enough operations, then you get a one-way function comes from a conjecture of Gowers [Gow96] (see also [HMMR05]). He conjectured that for every n, there is some polynomial $m = m(n)$ such that, if we choose a sequence $\overline{\sigma} = (\sigma_1, \ldots, \sigma_m)$ of m *random local permutations* over $\{0,1\}^n$, then the function $\sigma_1 \circ \cdots \sigma_m$ would be a *pseudorandom function*. We say that $\sigma : \{0,1\}^n \to \{0,1\}^n$ is a *local permutation* if it is obtained by applying a permutation on $\{0,1\}^3$ on three of the input bits. That is, there exist $i, j, k \in [n]$ and a permutation $\pi : \{0,1\}^3 \to \{0,1\}^3$ such that $\sigma(x)_\ell = x_\ell$ if $\ell \notin \{i, j, k\}$ and $\sigma(x)_{i,j,k} = \pi(x_i, x_j, x_k)$. The choice of the sequence $\overline{\sigma}$ consists of the *seed* for the pseudorandom function. Since pseudorandom functions imply one-way functions, this yields another candidate.

[10] Clearly, the lower the noise parameter δ, the easier this problem, but the best known algorithm requires δ to at most a logarithmic factor away from the trivial bound of $\delta = 1/n$ (where with good probability one could get n non-noisy linear equations). As δ becomes smaller, and in particular smaller than $1/\sqrt{n}$, the problem seems to acquire some *structure* and becomes more similar to the *learning with errors problem* discussed in Section 4.2 below. Indeed, as we mention there, in this regime Alekhnovich [Ale11] showed that the learning parity with noise problem can yield a *public-key* encryption scheme.

2.1.6 Private-Key Cryptography from Public-Key Assumptions

While it is an obvious fact, it is worth mentioning that all the assumptions implying *public-key* cryptography also imply *private-key* cryptography as well. Thus one can obtain one-way functions based on the difficulty of integer factoring, discrete logarithm, learning with errors, and all of the other assumptions that have been suggested as bases for public-key encryption and digital signatures.

3 Public-Key Cryptography: an Overview

We have seen that there is a wide variety of candidate private-key encryption schemes. From a superficial search of the literature, it might seem that there are a great many *public-key* systems as well. However, the currently well-studied candidates fall into only two families: schemes based on the difficulty of algebraic problems on certain concrete *Abelian groups*, and schemes based on the difficulty of geometric problems on linear *codes* or integer *lattices*; see Figure 1.

Family	**Sample cryptosystems**	**Structural properties**
"Algebraic" family: Abelian groups	Diffie–Hellman (ElGamal, elliptic curve cryptography), RSA	Polynomial-time quantum algorithm, subexponential classical algorithms (for all but elliptic curves), can break in $\mathbf{NP} \cap \mathbf{coNP}$
"Geometric" family: coding / lattices	Knapsack (Merkle–Hellman), McEliece, Goldreich–Goldwasser–Halevi, Ajtai–Dwork, NTRU, Regev	Can break in $\mathbf{NP} \cap \mathbf{coNP}$ or **SZK**. Non trivial classical and quantum algorithms for special cases (knapsack, principal ideal lattices)

Figure 1: The two "mainstream" families of public-key cryptosystems

Do these two families contain all the secure public schemes that exist? Or perhaps (if you think large-scale quantum computing could become a reality, or that the existing classical algorithms for the group-based family could be significantly improved) are lattices/codes the *only* source for secure public-key cryptography? The short answer is that we simply do not know, but in this survey I want to explore the long answer.

We will discuss some of the alternative public-key systems that have been proposed in the literature (see Section 5 and Figure 3) and ask what is the evidence for their security, and also to what extent are they *truly different* from the first two families. We will also ask whether this game of coming up with candidates and trying to break them is the best we can do or is there a more *principled* way to argue about the security of cryptographic schemes.

As mentioned, our discussion will be inherently *subjective*. I do not know of an objective way to argue that two cryptographic schemes belong to the "same family" or are "dissimilar". Some readers might dispute the assertion that there is any crisis or potential crisis in the foundations of

public-key cryptography, and some might even argue that there is no true difference between our evidence for the security of private and public-key cryptography. Nevertheless, I hope that even these readers will find some "food for thought" in this survey which is meant to provoke discussion more than to propose any final conclusions.

Remark 3.1. One could ask if there really is an inherent difference between public-key and private-key cryptography or maybe this is simply a reflection of our ignorance. That is, is it possible to build a public-key cryptosystem out of an arbitrary one-way function and hence base it on the same assumptions as *private-key* encryption? The answer is that we do not know, but in a seminal work, Impagliazzo and Rudich [IR89] showed that this cannot be done via the standard form of black-box security reductions. Specifically, they showed that, even given a random oracle, which is an idealized one-way function, one cannot construct a key-exchange protocol with a black-box proof that is secure against all adversaries running in polynomial time (or even $\omega(n^6)$ time, where n is the time expended by the honest parties). Barak and Mahmoody [BM09] improved this to $\omega(n^2)$ time, thus matching Merkle's 1974 protocol discussed in Section 5.1 below.

4 The Two "Mainstream" Public-Key Constructions

I now discuss the two main families of public-key constructions- ones that have their roots in the very first systems proposed by Diffie and Hellman [DH76b], Rivest Shamir and Adleman [RSA78], Rabin [Rab79], Merkle and Hellman [MH78], and McEliece [McE78] in the late 1970s.

4.1 The "Algebraic" Family: Abelian-Group Based Constructions

Some of the first proposals for public-key encryption were based on the *discrete logarithm* and the *factoring* problems, and these remain the most widely deployed and well-studied constructions. These were suggested in the open literature by Diffie and Hellman [DH76b], Rivest, Shamir, and Adelman [RSA78] and Rabin [Rab79], and in retrospect we learned that these scheme were discovered a few years before in the intelligence community by Ellis, Cocks, and Williamson [Ell99]. Later works by Miller [Mil85] and Koblitz [Kob87] obtained analogous schemes based on the discrete logarithm in *elliptic curve* groups.

These schemes have a rich algebraic structure that is essential to their use in the public-key setting, but also enable some nontrivial algorithmic results. These include the following:

- The factoring and discrete logarithm problems both fall in the class **TFNP**, which are **NP** search problems where *every* input is guaranteed to have a solution. Problems in this class cannot be **NP**-hard via a Cook reduction unless **NP** = **coNP** [MP91].[11] There are also some other complexity containments known for these problems [GK93, BO06].
- The integer factoring problem and discrete logarithm problem over $\mathbb{Z}_p^*$ have subexponential algorithms running in time roughly $\exp(\tilde{O}(n^{1/3}))$, where n is the bit complexity [LLJMP90].
- Very recently, *quasipolynomial*-time algorithms were shown for the discrete logarithm over finite fields of small characteristic [JP16].

[11] The proof is very simple and follows from the fact that, if SAT could be reduced via some reduction R to a problem in **TFNP**, then we could certify that a formula is *not* in SAT by giving a transcript of the reduction.

- There is no general sub-exponential discrete logarithm algorithm for elliptic curves, though sub-exponential algorithms are known for some families of curves such as those with large genus [ADH99]
- Shor's algorithm [Sho97] yields a polynomial time *quantum* algorithm for both the factoring and discrete logarithm problem.

4.2 The "Geometric Family": Lattice/Coding/Knapsack-Based Cryptosystems

The second type of public-key encryption candidates also have a fairly extended history.[12] Merkle and Hellman proposed in 1978 their knapsack scheme [MH78] (which, together with several of its variants, was later broken by lattice reduction techniques [Sha83]). In the same year, McEliece proposed a scheme based on the Goppa code [McE78]. In a seminal 1996 work, Ajtai [Ajt96] showed how to use integer lattices to to obtain a one-way function based on *worst-case* assumptions. Motivated by this work, Goldreich, Goldwasser, and Halevi [GGH97], as well as Ajtai and Dwork [Aw97] gave lattice-based public-key encryption schemes (the latter based also on *worst-case* assumptions). Around the same time, Hoffstein, Pipher, and Silverman constructed the NTRU public-key system [HPS98], which in retrospect can be thought of as a [GGH97]-type scheme based on lattices of a particularly structured form. In 2003, Regev [Reg03] gave improved versions of the Ajtai–Dwork cryptosystem. In 2003 Alekhnovich [Ale11] gave a variant of the Ajtai–Dwork system based on the problem of learning parity with (very small) noise, albeit at the expense of using *average-case* as opposed to worst-case hardness assumptions. See the survey [Pei15] for a more comprehensive overview of lattice-based cryptography.

Remark 4.1 (discreteness + noise = hardness?)**.** One way to think about all these schemes is that they rely on the *brittleness* of the Gaussian elimination algorithm over integers or finite fields. This is in contrast to the robust *least-squares minimization algorithm* that can solve even *noisy* linear equations over the real numbers. However, when working in the discrete setting (e.g., when x is constrained to be integers or when all equations are modulo some q), no analog of least-squares minimization is known. The presumed difficulty of this problem and its variants underlies the security of the above cryptosystems.

The Learning with Errors Problem (LWE). The cleanest and most useful formalization of the above intuition was given by Regev [Reg09], who made the following assumption:

Definition 4. For functions $\delta = \delta(n)$ and $q = q(n)$, the *learning with error (LWE)* problem with parameters q, δ is the task of recovering a fixed random $s \in \mathbb{Z}_q^n$, from $poly(n)$ examples (a, b) of the form

$$b = \langle s, a \rangle + \lfloor \eta \rfloor \pmod{q} \tag{3}$$

where a is chosen at random in $\mathbb{Z}_q^n$ and η is chosen from the normal distribution with standard deviation δq.

[12] The terminology of "group based" versus "lattice/code based" is perhaps not the most descriptive, as after all, lattices and codes are commutative groups as well. One difference seems to be the inherent role played by *noise* in the lattice/coding based constructions, which gives them a more geometric nature. However, it might be possible to trade non-commutativity for noise, and it has been shown that solving some lattice-based problems reduces to non-Abelian hidden subgroup problems [Reg04].

PRIVATE KEY: $s \leftarrow_R \mathbb{Z}_q^n$

PUBLIC-KEY: $(a_1, b_1), \ldots, (a_m, b_m)$ where each pair (a_i, b_i) is sampled independently according to (3).

ENCRYPT $m \in \{0,1\}$: Pick $\sigma_1, \ldots, \sigma_m \in \{\pm 1\}$, output (a', b') where $a' = \sum_{i=1}^m \sigma_i a_i \pmod q$ and $b' = \sum_{i=1}^m \sigma_i b_i + \lfloor \frac{q}{2} \rfloor \pmod q$.

DECRYPT (a', b'): Output 0 iff $|\langle s, a' \rangle - b' - \lfloor \frac{q}{2} \rfloor \pmod q| < q/100$.

Figure 2: Regev's simple public-key cryptosystem based on the LWE problem [Reg09]. The scheme will be secure as long as LWE holds for these parameters and $m \gg n \log q$. Decryption will succeed as long as the noise parameter δ is $o(1/\sqrt{m})$.

The *LWE assumption* is the assumption that this problem is hard for some $\delta(n)$ of the form n^{-C} (where C is some sufficiently large constant). Regev [Reg09] and Peikert [Pei09] showed that it is also equivalent (up to some loss in parameters) to its *decision* version where one needs to distinguish between samples of the form (a, b) as above and samples where b is simply an independent random element of $\mathbb{Z}_q$. Using this reduction, LWE can be easily shown to imply the existence of public-key cryptosystems, see Figure 2.

Regev [Reg09] showed that if the LWE problem with parameter $\delta(n)$ is easy, then there is a $\tilde{O}(n/\delta(n))$-factor (worst-case) approximation *quantum* algorithm for the *gap shortest vector problem* on lattices. Note that even if one doesn't consider quantum computing to be a physically realizable model, such a reduction can still be meaningful, and recent papers gave classical reductions as well [Pei09, BLP+13].

The LWE assumption is fast becoming the centerpiece of public-key cryptography, in the sense that a great many schemes for "plain" public-key encryption, as well as encryption schemes with stronger properties such as fully homomorphic [Gen09, BV11], identity based, or more, rely on this assumption, and there have also been several works which managed to "port" constructions and intuitions from the group-theoretic world into LWE-based primitives (e.g., see [PW11, CHKP12]).

Ideal/ring LWE. The *ideal* or *ring* variants of lattice problems correspond to the case when the matrix A has structure that allows to describe it using n numbers as opposed to n^2, and also often enables faster operations using a fast-Fourier-transform like algorithm. Such optimizations can be crucial for practical applications. See the manuscript [Pei16] for more on this assumption and its uses.

Approximate GCD. While in lattice-based cryptography we typically think of lattices of high dimension, when the numbers involved are large enough one can think of very small dimensions and even *one-dimensional* lattices. The computational question used for such lattices is often the *approximate greatest common denominator* (GCD) problem [How01] where one is given samples of numbers obtained by taking an integer multiple of a secret number s plus some small noise, and

the goal is to recover s (or at least distinguish between this distribution and the uniform one). Approximate GCD has been used for obtaining analogs of various lattice-based schemes (e.g., [vDGHV10]).

Structural properties of lattice-based schemes. The following structural properties are known about these schemes:

- All the known lattice-based public-key encryption schemes can be broken using oracle access to an $O(\sqrt{n})$ approximation algorithm for the lattice closest vector problem. Goldreich and Goldwasser showed that such an efficient algorithm exists if the class **SZK** (which is a subset of $\mathbf{AM} \cap \mathbf{coAM}$) is in **P** (or **BPP**, for that matter). Aharonov and Regev showed this also holds if $\mathbf{NP} \cap \mathbf{coNP} \subseteq \mathbf{P}$ [AR05]. Note that, while most experts believe that $\mathbf{NP} \cap \mathbf{coNP}$ is *not* contained in **P**, this result can still be viewed as showing that these lattice-based schemes have some computational *structure* that is not shared with many one-way function candidates.

- Unlike the lattice-based schemes, we do not know whether Alekhnovich's scheme [Ale11] is insecure if $\mathbf{AM} \cap \mathbf{coAM} \subseteq \mathbf{P}$ although it does use a variant of the learning parity with very low noise, which seems analogous to the closest vector problem with an approximation factor larger than $\sqrt{n}$. A recent result of Ben-Sasson et al. [BBD+16] suggests that using such a small amount of noise might be an *inherent* limitation of schemes of this general type.[13]

- The order-finding problem at the heart of Shor's algorithm can be thought of as an instance of a more general *hidden subgroup problem*. Regev showed a reduction from lattice problem into this problem for diehedral groups [Reg04]. Kuperberg gave a subexponential (i.e., $\exp(O(\sqrt{n}))$ time) quantum algorithm for this problem [Kup05], though it does not yield a subexponential quantum algorithm for the lattice problems since Regev's reduction has a quadratic blowup.

- A sequence of recent results showed that these problems are significantly easier (both quantumly and classically) in the case of *principal ideal* lattices which have a short basis that is obtained by taking shifts of a single vector (see [CDPR15] and the references therein).

The bottom line is that these schemes still currently represent our best hope for secure public-key systems if the group-theoretic schemes fail for a quantum or classical reason. However, the most practical variants of these schemes are also the ones that are more structured, and even relatively mild algorithmic advances (such as subexponential classical or quantum algorithms) could result in the need to square the size of the public-key or worse. Despite the fact that this would only be a polynomial factor, this can have significant real-world implications. One cannot hope to simply "plug in" a key of 10^6 or 10^9 bits into a protocol designed to work for keys of 10^3 bits and expect it to work as is, and so such results could bring about significant changes to the way we do security over the Internet. For example, it could lead to a centralization of power, where key exchange will be so expensive that users would share public-keys with only a few large corporations and governments, and smaller companies would have to route their communication through these larger corporations.

[13] [BBD+16] define a general family of public-key encryption schemes which includes Alekhnovich's scheme as well as Regev's and some other lattice-based schemes. They show that under a certain conjecture from additive combinatorics, all such schemes will need to use noise patterns that satisfy a generalized notion of being $\sqrt{n}$-sparse.

Remark 4.2 (Impagliazzo's worlds)**.** In a lovely survey, Impagliazzo [Imp95] defined a main task of computational complexity as determining which of several qualitatively distinct "worlds" is the one we live in, see Figure 4. That is, he looked at some of the various possibilities that, as far as we know, the open questions of computational complexity could resolve in, and saw how they would affect algorithms and cryptography.

As argued in Section 2 above, there is very strong evidence that one-way functions exist, which would rule out the three worlds Impagliazzo named as *"Algorithmica"*, *"Heuristica"*, and *"Pessiland"*. This survey can be thought of as trying to understand the evidence for ruling out the potential world *"Minicrypt"* where private-key cryptography (i.e., one-way functions) exist but not public-key cryptography. Impagliazzo used the name *"Cryptomania"* for the world in which public-key crypto, secure multiparty computation, and other similar primitives exist; these days people also refer to *"Obfustopia"* as the world where even more exotic primitives such as indistinguishability obfuscation [GGH+13] exist.

Scheme	**Computational assumption**	**Notes**
Merkle puzzles [Mer78]	Strong one way functions	Only quadratic security
Alekhnovich [Ale11]	Solving linear mod 2 equations with $\approx 1/\sqrt{n}$ noise	Mod 2 analog of Regev/Ajtai–Dwork, though not known to be solvable in **NP** $\cap$ **coNP/SZK**
ABW Scheme 1 [ABW10]	Planted 3LIN with $n^{1.4}$ clauses and noise $n^{-0.1}$	Similar to refuting random 3SAT with $n^{1.4}$ clauses, has nondeterministic refutation; some similarities to Alekhnovich
ABW Scheme 2 [ABW10]	Planted 3LIN with m clauses and noise δ + unbalanced expansion with parameters $(m, n, \delta m)$	Some similarities to Alekhnovich
ABW Scheme 3 [ABW10]	Nonlinear constant locality PRG with expansion $m(n)$ + unbalanced expansion with parameters $(m, n, \log n)$	At best $n^{\Omega(\log n)}$ security
Couveignes, Rostovtsev, Stolbunov [Cou06, RS06]	Isogeny star problem	Algebraic structure, similarities to elliptic curve cryptography, subexponential quantum algorithm
Patarin HFE systems [Pat96]	Planted quadratic equations	Several classical attacks
Sahai–Waters IO based system [SW14]	Indistinguishality obfuscation or witness encryption	All currently known IO/WE candidates require much stronger assumptions than Lattice schemes

Figure 3: A nonexhaustive list of some "non-mainstream" public-key candidates. See also Section 5

World	**Condition**	**Algorithmic implications**	**Cryptographic implications**
Algorithmica	**P** = **NP**	Algorithmic paradise, all **NP** and polynomial-hierarchy problems can be solved	Essentially no crypto
Heuristica	No average-case hard **NP** problem	Almost algorithmic paradise (though harder to solve problems in polynomial hierarchy)	Essentially no crypto
Pessiland	No hard *planted* **NP** problem (i.e., one-way functions)	Have hard on average algorithmic problem though can do all unsupervised learning	Essentially no crypto
Minicrypt	No public-key crypto	Algorithmic benefits minimal (can factor large integers, do discrete log, solve linear equations with very small noise)	CPA and CCA secure private-key encryption, pseudorandom functions, digital signatures, zero-knowledge proofs, etc.
Cryptomania	LWE conjecture holds but not IO	No algorithmic benefits known for lack of IO	All of the above plus CPA and CCA secure public-key encryption, secure multiparty computation, fully homomorphic encryption, private information retrieval, etc.
Obfustopia	LWE and IO		All of the above plus a growing number of applications including functional encryption, witness encryption, deniable encryption, replacing random oracles in certain instances, multiparty key exchange, and many more.

Figure 4: A variant of Impagliazzo's worlds from [Imp95]. We have redefined Cryptomania to be the world where LWE holds and denote by "Obfustopia" the world where indistinguishability obfuscators (IO) exist (see also [GPSZ16]).

5 Alternative public-key constructions

The group-theoretic and lattice-based families described above represent the main theoretical and practical basis for public-key encryption, as well as the more advanced applications, including secure multiparty computation [Yao82, GMW87], fully homomorphic encryption [Gen09, BV11], and many other primitives. However, there have been other proposals in the literature. We do not attempt a comprehensive survey here but do give some pointers (for another perspective, see also the NIST report [CJL+16]; these days, such alternative constructions are often grouped under the category of "post-quantum cryptography").

5.1 Merkle puzzles

The first public-key encryption proposed by an academic researcher was Ralph Merkle's "puzzle-based" scheme which he submitted to the *Communications of the ACM* in 1975 [Mer78] (as well as described in a project proposal for his undergraduate security class in the University of Berkeley), see Figure 5.[14]

Merkle's scheme can yield up to a *quadratic* gap between the work required to run the scheme and work required to break it, in an idealized (and not fully specified) model. Biham, Goren and Ishai [BGI08] showed that this model can be instantiated using exponentially strong one way functions.

Merkle conjectured that it should be possible to obtain a public-key scheme with an *exponential* gap between the work of the honest parties and the adversary but was unable to come up with a concrete candidate. (The first to do so would be Diffie and Hellman, who, based on a suggestion of John Gill to look at modular exponentiation, came up with what is known today as the *Diffie–Hellman key exchange*.) As mentioned in Remark 3.1, [BM09] (building on [IR89]) showed that Merkle's original protocol is *optimal* in the setting where we model the one-way function as a random oracle and measure running time in terms of the number of queries to this function.

We should note that, although n^2 security is extremely far from what we could hope for, it is not completely unacceptable. As pointed out by Biham et al. [BGI08], any superlinear security guarantee only becomes better with technological advances, since, as the honest parties can afford more computation, the ratio between their work and the adversary's grows.

5.2 Other Algebraic Constructions

There were several other proposals made for public-key encryption schemes. Some of these use *stronger* assumptions than those described above, for the sake of achieving better efficiency or some other attractive property. We briefly mention here schemes that attempt to use qualitatively different computational assumptions.

[14] Merkle's scheme, as well as the Diffie–Hellman scheme it inspired, are often known in the literature as *key-exchange protocols*, as opposed to a *public-key encryption schemes*. However, a key-exchange protocol that takes only two messages (as is the case for both Merkle's and Diffie–Hellman's schemes) is essentially the same as a (randomized) public-key encryption scheme, and indeed Diffie and Hellman were well aware that the receiver can use the first message as a public key that can be placed in a "public file" [DH76b]. I believe that this confusion in notation arose from the fact that the importance of randomization for encryption was not fully understood until the work of Goldwasser and Micali [GM82]. Thus, Diffie and Hellman reserved the name "public-key encryption" for a deterministic map we now call a *trapdoor permutation* that they envisioned as yielding an encryption by computing it in the forward direction and a signature by computing its inverse.

ASSUMPTIONS: $f : S \to \{0,1\}^*$ is an "ideal" 1-to-1 one-way function, that requires almost $|S|$ times as much time to invert as it does to compute. Let $n = |S|$.

PRIVATE KEY: $x_1, \ldots, x_{\sqrt{n}}$ that are chosen independently at random in S.

PUBLIC-KEY: $f(x_1), \ldots, f(x_{\sqrt{n}})$

ENCRYPT $m \in \{0,1\}$: Pick x at random in S, and if $f(x)$ appears in the public-key then output $f(x), h(x) \oplus m$ where $h(\cdot)$ is a "hardcore bit function" that can be obtained, e.g., by the method of Goldreich–Levin [GL89]. If $f(x)$ is not in the public-key then try again.
DECRYPT (y, b): Output $h(x_i) \oplus b$ where i is such that $f(x_i) = y$.

Figure 5: In Merkle's puzzle-based public-key encryption, the honest parties make $\approx \sqrt{n}$ invocation to an ideal one-way function, while an adversary making $\ll n$ invocations would not be able to break it

Hidden field equations. Patarin [Pat96] (following a work of Matsumoto and Imai [MI88]) proposed the *Hidden Field Equations* (HFE) cryptosystem. It is based on the difficulty of a "planted" variant of the quadratic equation problem over a finite field. The original HFE system was broken by Kipnis and Shamir [KS99], and some variants have been attacked as well. It seems that currently fewer attacks are known for HFE-based signatures, though our interest here is of course only in public-key encryption; see [CDF03] for more information about known attacks.

Isogeny star. Rostovtsev and Stolbunov [RS06] (see also [Cou06]) proposed a cryptographic scheme based on the task of finding an *isogeny* (an algebraic homomorphism) between two elliptic curves. Although this scheme is inspired by elliptic-curve cryptography, its security does not reduce to the security of standard elliptic-curve based schemes. In particular, there are no known quantum algorithms to attack it, though there have been some related results [CJS14, BJS14]. Another group-theoretic construction that was suggested is to base cryptography on the conjugacy problem for *braid groups* though some attacks have been shown on these proposals (e.g., see [MU07] and references therein).

5.3 Combinatorial(?) Constructions

Applebaum, Barak and Wigderson [ABW10] tried to explore the question of whether public-key encryption can be based on the conjectured average-case difficulty of *combinatorial* problems. Admittedly, this term is not well defined, though their focus was mostly on *constraint satisfaction problems*, which are arguably the quintessential combinatorial problems.

[ABW10] gave a construction of a public-key encryption scheme (see Figure 6) based on the following conjectures:

ASSUMPTIONS:
(i) There is a constant d and some $f : \{0,1\}^d \to \{0,1\}$ such that, if we choose a random (n, m, d) graph H, then the map $G : \{0,1\}^n \to \{0,1\}^m$ where $G_H(x)_j = f(x_{\overleftarrow{\Gamma}_H(j)})$ is a pseudorandom generator, where $\overleftarrow{\Gamma}_H(j)$ denotes the left-neighbors of j in H.

(ii) It is hard to distinguish between a random (n, m, d) graph H and a random (n, m, d) graph where we *plant* a set S of right vertices of size $k = O(\log n)$ such that $|\overleftarrow{\Gamma}_H(S)| = k - 1$ where $\overleftarrow{\Gamma}_H(S)$ denotes the set of left-neighbors of S in H.

KEY GENERATION: Choose a random (n, m, d) graph H with a planted nonexpanding set S. The public-key is H, and the private key is S.

ENCRYPT $m \in \{0,1\}$: If $m = 0$ then output a random $y \in \{0,1\}^m$. If $m = 1$ pick random $x \in \{0,1\}^n$ and output $y = G_H(x)$.

DECRYPT y: Output 1 iff $y_S \in \{G_H(x)_{|S} : x \in \{0,1\}^n\}$. By our condition this set has at most 2^{k-1} elements.

Figure 6: The ABW Goldreich-generator-based encryption scheme (a simplified variant)

- A *local pseudorandom generator*: this is a strengthening of the assumption that Golreich's one-way function discussed in Section 2.1.4 is secure. Namely, we assume that we can obtain a pseudorandom generator mapping n bits to m bits where every output bit applies some predicate f to a constant number d of input bits. Furthermore, we assume that we can do so by choosing which input bits map into which output bits using a random (n, m, d) bipartite graph as defined in Section 2.1.4.[15]
- The *unbalanced expansion problem*: this is the problem of distinguishing between a random (n, m, d) bipartite graph as above, and such a graph where we *plant* a set S of size k of left vertices such that S has at most $k - 1$ neighbors on the right-hand side (as opposed to the $(d - 1 - o(1))k$ neighbors you would expect in a random graph).[16] Expansion problems in graphs have been widely studied (e.g., see [HLW06]), and at the moment no algorithm is known for this range of parameters.

The larger m is compared with n, the stronger the first assumption and the weaker the second

[15] [ABW10] also gave a version of their cryptosystem which only assumed that the function is one way, and more general reductions between these two conditions were given in [App12].

[16] One only needs to conjecture that it has to distinguish between these graphs with some constant bias, as there are standard techniques for hardness amplification in this context.

assumption. Increasing the parameter k makes the second problem harder (and in fact, depending on m/n, at some point the assumption becomes *unconditionally* true since there would exist such a nonexpanding set with high probability even in a random graph). Moreover, there is always a way to solve the expansion problem in $\binom{n}{k}$ time, and so smaller values of k make the problem quantitatively easier. [ABW10] showed that, if both assumptions hold for a set of parameters (n, m, d, k) where $k = O(\log n)$, then there exists a public-key cryptosystem.

By construction, the above cryptosystem cannot achieve better than $n^{\Omega(\log n)}$ security which is much better than the n^2 obtained by Merkle puzzles but still far from ideal. It also relies on the somewhat subtle distinction between $n^{O(k)}$ and $poly(n)2^{O(k)}$ complexity. [ABW10] showed how to get different tradeoffs if, instead of using a *non-linear* function f for the pseudo-random generator, we use a linear function with some probabilistic additional noise. The noise level δ should satisfy $\delta k = O(1/\log n)$ for efficient decryption, and so the lower the noise level we consider (and hence the stronger we make our assumption on the pseudo-random generator), the larger value of k we can afford. In particular, if we assume a sufficiently low level of noise, then we can get k to be so large as to avoid the second assumption (on difficulty of detecting nonexpanding sets) altogether. However, there is evidence that at this point the first assumption becomes more "structured" since it admits a non-constructive short certificate [FKO06].

Using such a linear function f raises the question of to which extent these schemes are different from coding-based schemes such as Alekhnovich's. Indeed, there are similarities between these schemes and the main difference is the use of the unbalanced expansion assumption. An important question is to find the extent to which this problem is *combinatorial* versus *algebraic*. We do not yet fully understand this question, nor even the right way to formally define it, but it does seem key to figuring out whether the [ABW10] scheme is truly different from the coding/lattices-based constructions. On one hand, the unbalanced expansion questions "feels" combinatorial. On the other hand, the fact that we require the set S to have fewer than S neighbors implies that, if we define for each right-vertex j in H a linear equation corresponding to the sum of variables in $\overleftarrow{\Gamma}_H(S)$, then the equations corresponding to S are *linearly dependent*. So this problem can be thought of as the task of looking for a short linear dependency.

Thinking about the *noise level* might be a better way of considering this question than the combinatorial versus algebraic distinction. That is, one could argue that the main issue with the coding/lattice-based constructions is not the algebraic nature of the linear equations (after all, both the knapsack and approximating kXOR problems are **NP** hard). Rather, it is the fact that they use a noise level smaller than $1/\sqrt{n}$ (or, equivalently, a larger than $\sqrt{n}$ approximation factor) that gives them some *structure* that could potentially be a source of weakness. In particular, using such small noise is quite analogous to using an approximation factor larger than $\sqrt{n}$ for lattice problems, which is the reason why lattice-based schemes can be broken in $\mathbf{NP} \cap \mathbf{coNP}$. However, at the moment no such result is known for either the [Ale11] or [ABW10] schemes.

This viewpoint raises the following open questions:

- Can we base a public-key encryption scheme on the difficulty of solving $O(n)$ random kXOR equations on n variables with a planted solution satisfying $1 - \epsilon$ of them for some constant $\epsilon > 0$?
- Does the reliance on the unbalanced expansion problem introduce new *structure* in the problem? For example, is there *a nondeterministic* procedure to certify the *nonexistence* of a short non-expanding subset in a graph?

One way to get evidence for a negative answer for the second question would be to get a *worst-case* **NP** hardness of approximation result for the unbalanced expansion problem with parameters matching those used by [ABW10]. We do not at the moment know whether such a result is likely or not to hold.

5.4 Public-key Cryptography from Indistinguishability Obfuscators

From the early writing of Diffie and Hellman [DH76a], it seems that one of the reasons why they believed that public-key cryptography is at least not inherently impossible is the following: Given a block cipher/pseudorandom permutation collection $\{p_k\}$, one could imagine fixing a random key k and letting P_k be a program that on input x outputs $p_k(x)$. Now, if P_k was compiled via some "optimizing compiler" to a low-level representation such as assembly language, one could imagine that it would be hard to "extract" k from this representation. Thus, one can hope to obtain a public-key encryption scheme (or, more accurately, a trapdoor permutation family) by letting the *public encryption key* be this representation of P_k, which enables computing the map $x \mapsto p_k(x)$, and letting the *private decryption key* (or *trapdoor*) be the secret key k that enables computing the map $y \mapsto p_k^{-1}(y)$. It seems that James Ellis, who independently invented public-key encryption at the British intelligence agency GCHQ, had similar thoughts [Ell99].

Diffie and Hellman never managed to find a good enough instantiation of this idea, but over the years people have kept trying to look for such an *obfuscating compiler* that would convert a program P to a functionally equivalent but "inscrutable" form. Many practical attempts at obfuscation have been broken, and the paper [BGI+12] showed that a natural definition for security of obfuscation is in fact *impossible* to achieve. However, [BGI+12] did give a weaker definition of security, known as *indistinguishability obfuscation* (IO), and noted that their impossibility result did not rule it out. (See the survey [Bar16].)

In a recent breakthrough, a candidate construction for an IO compiler was given by [GGH+13]. They also showed (see Figure 7) that an IO compiler is sufficient to achieve Diffie and Hellman's dream of constructing a public-key encryption scheme based only on one-way functions.[17] Now from a first look, this might seem to make as much sense as a bottle opener made out of diamonds: after all, we can already build public-key encryption from the learning with error assumption, while building IO from LWE would be a major breakthrough with a great many applications. Indeed, many of the current candidate constructions for IO would be easily broken if LWE was easy. (And in fact might be broken regardless [MSZ16].)

However, a priori, it is not at all clear that achieving IO requires an *algebraic* approach. While at the moment it seems far removed from any techniques we have, one could hope that a more combinatorial/program transformation approach can yield an IO obfuscator without relying on LWE. One glimmer of hope is given by the observation that despite the great many applications of IO, so far we have not been able to obtain primitives such as fully homomorphic encryption that imply that $\mathbf{AM} \cap \mathbf{coAM} \not\subseteq \mathbf{BPP}$ (see also [AS15]). In contrast, such primitives do follow from LWE.

[17] Another construction (which enjoyed extra interesting properties) of a public key encryption scheme from IO was given by [SW14].

ASSUMPTIONS: $G : \{0,1\}^n \to \{0,1\}^{2n}$ is a pseudorandom generator. $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$ is an indistinguishabilty obfuscation (IO) compiler. Let $n = |S|$.

PRIVATE KEY: $x_0 \in_R \{0,1\}^n$.

PUBLIC-KEY: $y_0 = G(x_0)$

ENCRYPT $m \in \{0,1\}$: Let $F_m : \{0,1\}^n \to \{0,1\}$ be defined as follows:

$$F_m(x) = \begin{cases} m & G(x) = y_0 \\ 0 & \text{otherwise} \end{cases}$$

Output $O(C_m)$ where C_m is a canonical circuit for F_m.

DECRYPT C: Output $C(x_0)$. Indeed $C(x_0) = F_m(x_0) = m$.

ARGUMENT FOR SECURITY: We need to shoe that $(y_0, E_{y_0}(0))$ is indistinguishable from $(y_0, E_{y_0}(1))$. By pseudorandomness of G, this is indistinguishable from the case that y_0 is chosen at random in $\{0,1\}^{2n}$. But then with high probability $y_0 \notin G(\{0,1\}^n)$ and hence F_0 and F_1 both equal the identically zero function, and hence $O(C_0)$ and $O(C_1)$ are indistinguishable by the I.O. property.

Figure 7: Public Key Encryption from Indistinguishabilty Obfuscation and One-Way Functions.

6 Is Computational Hardness the Rule or the Exception?

As long as the **P** versus **NP** question remains open, cryptography will require unproven assumptions. Does it really make sense to distinguish between an assumption such as the hardness of LWE and assuming hardness of the problems that yield private-key encryption? This is a fair question. After all, many would argue that the only real evidence we have that $\mathbf{P} \neq \mathbf{NP}$ is the fact that a lot of people have tried to get algorithms for **NP**-hard problems and failed. That same evidence exists for the LWE assumption as well.

However, I do feel there is a *qualitative* difference between these assumptions. The reason is that assuming $\mathbf{P} \neq \mathbf{NP}$ yields a *coherent and beautiful* theory of computational difficulty that agrees with all current observations. Thus we accept this theory not only because we do not know how to refute it, but also because, following Occam's razor principle, one should accept the cleanest/most parsimonious theory that explains the world as we know it. The existence of one-way functions, with the rich web of reductions that have been shown between it and other problems, also yields such a theory. Indeed, these reductions have shown that one-way functions are a *minimal* assumption for almost all of cryptography.

In contrast, while LWE has many implications, it has not been shown to be *minimal* for "Cryptomania" in the sense that it is not known to be implied by any primitives such as public-key

encryption or even stronger notions such as fully homomorphic encryption. We also do not have a clean theory of average-case hardness that would imply the difficulty of LWE (or the existence of public-key encryptions).

In fact, I believe it is fair to say that we don't have a clean theory of average-case hardness at all.[18] The main reason is that *reductions*—which underpin the whole theory of worst-case hardness, as well as the web of reductions between cryptographic primitives—seem to have very limited applicability in this setting. As a rule, a reduction from a problem A to a problem B typically takes a *general* instance of A and transforms it to a *structured* instance of B. For example, the canonical reduction from 3SAT to 3COL takes a general formula φ and transforms it into a graph G that has a particular form with certain *gadgets* that correspond to every clause of φ. While this is enough to show that, if A is hard in the worst-case then so is B, it does not show that, if A is hard on, say, uniformly random instances, then this holds for B as well. Thus reductions have turned out to be extremely useful for relating the worst-case complexity of different problems, or using the conjectured average-case hardness of a particular problem to show the hardness of other problems on *tailored* instances (as we do when we construct cryptographic primitives based on average-case hardness). However, by and large, we have not been able to use reductions to relate the hardness of natural average-case problems, and so we have a collection of incomparable tasks including integer factoring, discrete logarithms, the RSA problem, finding planted cliques, finding planted assignments in 3SAT formulas, LWE, etc. without any reductions between them.[19]

Even the successful theory of worst-case complexity is arguably more *descriptive* or *predictive* than *explanatory*. That is, it tells us which problems are hard, but it does not truly explain to us *why* they are hard. While this might seem as not a well-defined question, akin to asking "why is 17 a prime?", let me try to cast a bit more meaning into it, and illustrate how an *explanatory* theory of computational difficulty might be useful in situations such as average-case complexity, where reductions do not seem to help.

What makes a problem easy or hard? To get some hints on answers, we might want to look at what algorithmicists do when they want to efficiently solve a problem, and what cryptographers do when they want to create a hard problem. There are obviously a plethora of algorithmic techniques for solving problems, and in particular many clever data structures and optimizations that can make improvements that might be moderate in theory (e.g., reducing an exponent) but make all the difference in the world in practice. However, if we restrict ourselves to techniques that help show a problem can be solved in better than brute force, then there are some themes that repeat themselves time and again. One such theme is *local search*. Starting with a guess for a solution and making local improvements is a workhorse behind a great many algorithms. Such algorithms crucially rely on a structure of the problem where *local* optima (or at least all ones you are likely to encounter) correspond to *global* optima. In other words, they rely on some form of *convexity*.

Another theme is the use of *algebraic cancellations*. The simplest such structure is *linearity*, where we can continually deduce new constraints from old ones without a blowup in their complexity. In particular, a classical example of cancellations in action is the algorithm to efficiently compute the *determinant* of a matrix, which works even though at least one canonical definition

[18] Levin [Lev86] has proposed a notion of completeness for average-case problems, though this theory has not been successful in giving evidence for the hardness of natural problems on natural input distributions.

[19] One notable exception is the set of reductions between different variants of lattice problems, which is enabled by the existence of a *worst-case to average-case* reduction for these problems [Ajt96]. However, even there we do not know how to relate these problems to tasks that seem superficially similar such as the *learning parity with noise* [GKL93, BFKL93] problem.

of it involves computing a sum on an exponential number of terms.

On the cryptography side, when applied cryptographers try to construct a hard function such as a hash function or a block cipher, there are themes that recur as well. To make a function that is hard to invert, designers try to introduce *nonlinearity* (the function should not be linear or close to linear over any field and in fact have large algebraic degree so it is hard to "linearize") and *nonlocality* (we want the dependency structure of output and input bits to be "expanding" or "spread out"). Indeed, these themes occur not just in applied constructions but also in theoretical candidates such as Goldreich's [Gol11] and Gowers' [Gow96] (where each takes one type of parameters to a different extreme).

Taken together, these observations might lead to a view of the world in which computational problems are presumed hard unless they have a structural reason to be easy. A theory based on such structure could help to *predict*, and more than that to *explain*, the difficulty of a great many computational problems that currently we cannot reach with reductions. However, I do not know at the moment of any such clean theory that will not end up "predicting" some problems are hard where they are in fact solvable by a clever algorithm or change of representation. In the survey [BS14], Steurer and I tried to present a potential approach to such a theory via the conjecture that the *sum of squares* convex program is *optimal* in some domains. While it might seem that making such conjectures is a step backwards from cryptography as a science towards "alchemy", we do hope that it is possible to extract some of the "alchemist intuitions" practitioners have, without sacrificing the predictive power and the mathematical crispness of cryptographic theory. However, this research is still very much in its infancy, and we still do not even know the right way to formalize our *conjectures*, let alone try to prove them or study their implications. I do hope that eventually an explanatory theory of hardness will emerge, whether via convex optimization or other means, and that it will not only help us design cryptographic schemes with stronger foundations for their security, but also shed more light on the mysterious phenomena of efficient computation.

References

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 171–180, 2010.

[AC08] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 793–802, 2008.

[ADH99] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh A. Huang. A subexponential algorithm for discrete logarithms over hyperelliptic curves of large genus over gf(q). *Theor. Comput. Sci.*, 226(1-2):7–18, 1999.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108, 1996.

[Ale11] Michael Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, 20(4):755–786, 2011. Published posthumously. Preliminary version in FOCS '03.

[App12] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 805–816, 2012.

[App15] Benny Applebaum. The cryptographic hardness of random local functions - survey. *IACR Cryptology ePrint Archive*, 2015:165, 2015.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP cap conp. *J. ACM*, 52(5):749–765, 2005. Preliminary version in FOCS '04.

[AS15] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 191–209. IEEE, 2015.

[Aw97] Miklós Ajtai and Cynthia D work. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 284–293, 1997.

[Bar14] Boaz Barak. Structure vs. combinatorics in computational complexity. *Bulletin of the European Association for Theoretical Computer Science*, 112, 2014. Survey, also posted on Windows on Theory blog.

[Bar16] Boaz Barak. Hopes, fears, and software obfuscation. *Commun. ACM*, 59(3):88–96, 2016.

[BBD+16] Eli Ben-Sasson, Iddo Ben-Tov, Ivan Damgård, Yuval Ishai, and Noga Ron-Zewi. On public key encryption from noisy codewords. In *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part II*, pages 417–446, 2016.

[BDPVA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak SHA-3 submission. *Submission to NIST (Round 3)*, 6(7):16, 2011.

[BE76] Béla Bollobás and Paul Erdös. Cliques in random graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 80, pages 419–427. Cambridge Univ Press, 1976.

[Ber08] Daniel J Bernstein. The salsa20 family of stream ciphers. In *New stream cipher designs*, pages 84–97. Springer, 2008.

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 278–291, 1993.

[BGI08] Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions. In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, pages 55–72, 2008.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[BJS14] Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *Progress in Cryptology–INDOCRYPT 2014*, pages 428–442. Springer, 2014.

[BKS13] Boaz Barak, Guy Kindler, and David Steurer. On the optimality of semidefinite relaxations for average-case and generalized constraint satisfaction. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 197–214, 2013.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. Preliminary version in STOC '00.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.

[BM07] Boaz Barak and Mohammad Mahmoody-Ghidary. Lower bounds on signatures from symmetric primitives. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 680–688, 2007.

[BM09] Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 374–390. Springer, 2009.

[BO06] Joshua Buresh-Oppenheim. On the tfnp complexity of factoring, 2006.

[BQ12] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. *Computational Complexity*, 21(1):83–127, 2012.

[BS14] Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms, 2014.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106, 2011.

[CDF03] Nicolas T Courtois, Magnus Daum, and Patrick Felke. On the security of hfe, hfev-and quartz. In *Public Key Cryptography—PKC 2003*, pages 337–350. Springer, 2003.

[CDPR15] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. *IACR Cryptology ePrint Archive*, 2015:313, 2015.

[CEMT09] James Cook, Omid Etesami, Rachel Miller, and Luca Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In *Theory of Cryptography*, pages 521–538. Springer, 2009.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[CJL+16] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. *National Institute of Standards and Technology Internal Report*, 8105, 2016. Available on `http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf`.

[CJS14] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

[COS86] Don Coppersmith, Andrew M Odlzyko, and Richard Schroeppel. Discrete logarithms ingf (p). *Algorithmica*, 1(1-4):1–15, 1986.

[Cou06] Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006:291, 2006.

[DH76a] Whitfield Diffie and Martin E Hellman. Multiuser cryptographic techniques. In *Proceedings of the June 7-10, 1976, national computer conference and exposition*, pages 109–112. ACM, 1976.

[DH76b] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DR13] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[Ell99] James H Ellis. The history of non-secret encryption. *Cryptologia*, 23(3):267–273, 1999.

[FKO06] Uriel Feige, Jeong Han Kim, and Eran Ofek. Witnesses for non-satisfiability of dense random 3cnf formulas. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 497–508, 2006.

[Fri99] Ehud Friedgut. Sharp thresholds of graph properties, and the -sat problem. *Journal of the American mathematical Society*, 12(4):1017–1054, 1999. With an appendix by Jean Bourgain.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 112–131, 1997.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49, 2013.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GK93] Oded Goldreich and Eyal Kushilevitz. A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm. *J. Cryptology*, 6(2):97–116, 1993.

[GKL93] Oded Goldreich, Hugo Krawczyk, and Michael Luby. On the existence of pseudorandom generators. *SIAM J. Comput.*, 22(6):1163–1175, 1993. Preliminary versions in CRYPTO '88 and FOCS '88.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32, 1989.

[GM75] Geoffrey R Grimmett and Colin JH McDiarmid. On colouring random graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 77, pages 313–324. Cambridge Univ Press, 1975.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 365–377, 1982.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229, 1987.

[Gola] Oded Goldreich. Lessons from kant: On knowledge, morality, and beauty. Essay available on `http://www.wisdom.weizmann.ac.il/~oded/on-kant.html`,.

[Golb] Oded Goldreich. On cryptographic assumptions. Short note available on `http://www.wisdom.weizmann.ac.il/~oded/on-assumptions.html`,.

[Golc] Oded Goldreich. On intellectual and instrumental values in science. Essay available on `http://www.wisdom.weizmann.ac.il/~oded/on-values.html`,.

[Gold] Oded Goldreich. On post-modern cryptography. Short note available on `http://www.wisdom.weizmann.ac.il/~oded/on-pmc.html`, revised on 2012.

[Gole] Oded Goldreich. On quantum computing. Essay available on `http://www.wisdom.weizmann.ac.il/~oded/on-qc.html`,.

[Golf] Oded Goldreich. On scientific evaluation and its relation to understanding, imagination, and taste. Essay available on `http://www.wisdom.weizmann.ac.il/~oded/on-taste.html`,.

[Golg] Oded Goldreich. On the philosophical basis of computational theories. Essay available on `http://www.wisdom.weizmann.ac.il/~oded/on-qc3.html`,.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[Gol11] Oded Goldreich. Candidate one-way functions based on expander graphs. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 76–87. 2011. Original version published as ECCC TR00-090 in 2000.

[Gow96] WT Gowers. An almost m-wise independent random permutation of the cube. *Combinatorics, Probability and Computing*, 5(02):119–130, 1996.

[GPSZ16] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. *IACR Cryptology ePrint Archive*, 2016:102, 2016.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. Preliminary versions in STOC '89 and STOC '90.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[HMMR05] Shlomo Hoory, Avner Magen, Steven Myers, and Charles Rackoff. Simple permutations mix well. *Theor. Comput. Sci.*, 348(2-3):251–261, 2005.

[How01] Nick Howgrave-Graham. Approximate integer common divisors. In *Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers*, pages 51–66, 2001.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *Algorithmic number theory*, pages 267–288. Springer, 1998.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 230–235, 1989.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147, 1995.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 44–61, 1989.

[Its10] Dmitry Itsykson. Lower bound on average-case complexity of inversion of goldreich's function by drunken backtracking algorithms. In *Computer Science–Theory and Applications*, pages 204–215. Springer, 2010.

[Jer92] Mark Jerrum. Large cliques elude the metropolis process. *Random Structures & Algorithms*, 3(4):347–359, 1992.

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

[JP16] Antoine Joux and Cécile Pierrot. Technical history of discrete logarithms in small characteristic finite fields - the road from subexponential to quasi-polynomial complexity. *Des. Codes Cryptography*, 78(1):73–85, 2016.

[Kar76] Richard M Karp. The probabilistic analysis of some combinatorial search algorithms. *Algorithms and complexity: New directions and recent results*, 1:19, 1976.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 19–30, 1999.

[Kuc95] Ludek Kucera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212, 1995.

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005.

[Lev86] Leonid A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.

[LLJMP90] Arjen K Lenstra, Hendrik W Lenstra Jr, Mark S Manasse, and John M Pollard. The number field sieve. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 564–572. ACM, 1990.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX*

2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings, pages 378–389, 2005.

[McE78] RJ McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, 1978.

[Mer78] Ralph C. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, 1978. Originally submitted in August 1975.

[MH78] Ralph C Merkle and Martin E Hellman. Hiding information and signatures in trapdoor knapsacks. *Information Theory, IEEE Transactions on*, 24(5):525–530, 1978.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynominal-tuples for efficient signature-verification and message-encryption. In *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, pages 419–453, 1988.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO'85 Proceedings*, pages 417–426. Springer, 1985.

[MP91] Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13. Cryptology ePrint Archive, Report 2016/147, 2016. `http://eprint.iacr.org/`.

[MU07] Alex D Myasnikov and Alexander Ushakov. Length based attack and braid groups: cryptanalysis of anshel-anshel-goldfeld key exchange protocol. In *Public Key Cryptography–PKC 2007*, pages 76–88. Springer, 2007.

[MV12] Eric Miles and Emanuele Viola. Substitution-permutation networks, pseudorandom functions, and natural proofs. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 68–85, 2012.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991. Preliminary version in CRYPTO '89.

[NIS02] NIST. Secure hash standard, 2002. Federal Information Processing Standard Publication 180-2. US Department of Commerce, National Institute of Standards and Technology (NIST).

[NSA15] Cryptography today: Memorandum on suite b cryptography, 2015. Retrieved on 2/29/16 from `https://www.nsa.gov/ia/programs/suiteb_cryptography/`,.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 33–43, 1989.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *ISTCS*, pages 3–17, 1993.

[OW14] Ryan O'Donnell and David Witmer. Goldreich's PRG: evidence for near-optimal polynomial stretch. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 1–12, 2014.

[Pat96] Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *Advances in Cryptology—EUROCRYPT'96*, pages 33–48. Springer, 1996.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342. ACM, 2009.

[Pei15] Chris Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939, 2015. `http://eprint.iacr.org/`.

[Pei16] Chris Peikert. How (not) to instantiate ring-LWE, 2016. Unpublished manuscript; available at `web.eecs.umich.edu/~cpeikert/pubs/instantiate-rlwe.pdf`.

[PW11] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM Journal on Computing*, 40(6):1803–1844, 2011.

[Rab79] Michael O Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, MIT technical report, 1979.

[Reg03] Oded Regev. New lattice based cryptographic constructions. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 407–416, 2003.

[Reg04] Oded Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009. Preliminary version in STOC 2005.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 387–394, 1990.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptology ePrint Archive*, 2006:145, 2006.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[Sha83] Adi Shamir. A polynomial time algorithm for breaking the basic merkle-hellman cryptosystem. In *Advances in Cryptology*, pages 279–288. Springer, 1983.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Preliminary version in FOCS '94.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 475–484, 2014.

[Tar05] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation.* siam, 2005.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 24–43, 2010.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164, 1982.