The Leakage-Resilience Limit of a Computational Problem Is Equal to Its Unpredictability Entropy

Divesh Aggarwal and Ueli Maurer

Department of Computer Science
ETH Zurich
CH-8092 Zurich, Switzerland
{divesha,maurer}@inf.ethz.ch

Abstract. A cryptographic assumption is the (unproven) mathematical statement that a certain computational problem (e.g. factoring integers) is computationally hard. The leakage-resilience limit of a cryptographic assumption, and hence of a computational search problem, is the maximal number of bits of information that can be leaked (adaptively) about an instance, without making the problem easy to solve. This implies security of the underlying scheme against arbitrary side channel attacks by a computationally unbounded adversary as long as the number of leaked bits of information is less than the leakage resilience limit.

The hardness of a computational problem is typically characterized by the running time of the fastest (known) algorithm for solving it. We propose to consider, as another natural complexity-theoretic quantity, the success probability of the best polynomial-time algorithm (which can be exponentially small). We refer to its negative logarithm as the *unpredictability entropy* of the problem (which is defined up to an additive logarithmic term).

A main result of the paper is that the leakage-resilience limit and the unpredictability entropy are equal. This demonstrates, for the first time, the practical relevance of studying polynomial-time algorithms even for problems believed to be hard, and even if the success probability is too small to be of practical interest. With this view, we look at the best probabilistic polynomial time algorithms for the learning with errors and lattice problems that have in recent years gained relevance in cryptography.

We also introduce the concept of *witness compression* for computational problems, namely the reduction of a problem to another problem for which the witnesses are shorter. The length of the smallest achievable witness for a problem also corresponds to the *non-adaptive* leakage-resilience limit, and it is also shown to be equal to the unpredictability entropy of the problem. The witness compression concept is also of independent theoretical interest. An example of an implication of our result is that 3-SAT for n variables can be witness compressed from n bits (the variable assignments) to $0.41n$ bits.

D.H. Lee and X. Wang (Eds.): ASIACRYPT 2011, LNCS 7073, pp. 686–701, 2011.

1 Introduction and Motivation

1.1 Leakage Resilience of Cryptographic Assumptions

There have been many recent works (e.g., [2,3,10,13,16,14,21,33,37,38,40,42,43], and the references therein) aimed at designing cryptographic schemes that are secure against a large class of side-channel attacks. Some of these look at side channel attacks where the adversary can obtain some function of the secret key. We look at an even more general class of side-channel attacks where the adversary can obtain a *bounded amount of arbitrary information*. We model this kind of attack by allowing the adversary a bounded number of queries to an infinitely powerful oracle $\mathcal{O}$ that can be asked arbitrary binary (YES/NO) questions. This oracle was considered by Maurer [37] to study the hardness of factoring N given queries to this oracle.

Goldwasser et al [24] raised a more general question regarding leakage which is also the question that we are concerned with: Which of the cryptographic assumptions (rather than cryptographic schemes) are secure in the presence of leakage of some bits of information?

1.2 Complexity Notions

In this section, we introduce three notions, *unpredictability entropy*, *oracle complexity*, and *witness compressibility*, whose relationship we study in this paper.

A well-studied and realistic approach in the study of the computational complexity of a computational problem is to look at probabilistic polynomial time (PPT) algorithms that solve the problem. We define the *unpredictability entropy* [31] of a problem (essentially) as $-\log_2 p$, where p is the maximum possible success probability of a PPT algorithm for solving the problem. A common understanding is that the study of probabilistic algorithms makes sense only if the probability of success is non-negligible. While there have been a few results like [6,7,9,12,17,18,19,23,44,48] that look at the class of one-sided error probabilistic polynomial time (OPP) algorithms for decision problems with negligible success probability p, these are studied with the viewpoint of improving the bound on the exact worst-case complexity of the problem by repeating the algorithm $O(1/p)$ times and hence amplifying the success probability to a non-negligible quantity. However, we argue that PPT algorithms are interesting even if the success probability p is negligible and even if there exist other exact algorithms that run in time much less than $O(1/p)$.

Maurer [37] considered a class of PPT algorithms for search problems given the oracle $\mathcal{O}$. If the algorithm is allowed as many binary queries to the oracle as is the length of the solution/witness, then there is a trivial algorithm that solves the problem. Thus, this class of algorithms is looked at with the goal of minimizing the number of queries. The minimum number of queries required by a PPT algorithm for solving this problem with overwhelming probability is the *oracle complexity* (which is the same as the leakage-resilience limit) of the problem. A motivation for looking at such an oracle, as pointed out by the author, is

to determine whether the difficulty of a certain problem can be concentrated in a few difficult bits leading to a new complexity theoretic classification of problems. This question was answered in the affirmative for the integer factorization problem in [37] but it remains open for other computational problems. Consider, for instance, the problem of computing discrete logarithms modulo a prime q. One can see that the hardness of this problem and the integer factorization problem is closely related in the sense that almost all algorithms for solving the factoring problem have a variant that solves the discrete logarithm problem modulo a prime. A survey of this can be found in Chapter 3 of [28]. However, the hardness of the two problems seems to differ significantly in terms of the number of queries to $\mathcal{O}$ required in order to solve these problems in polynomial time. Factoring can be solved with a small number of queries but, to the best of our knowledge, there exists no algorithm that solves the discrete logarithm problem with a non-trivial number (i.e., substantially less than the solution size) of queries to $\mathcal{O}$. Thus, finding the oracle complexity seems to be an interesting research area in itself.

We introduce another related notion called the *witness compressibility* of a problem. This is the smallest size k such that there is a PPT reduction that reduces the witness size of a given instance to at most k with overwhelming probability. This quantity can be seen as the non-adaptive leakage-resilience limit of an assumption about the hardness of the problem. A problem is not resilient to k bits of non-adaptive leakage if and only if it is witness compressible up to k bits. [1]

Note that the three quantities, i.e., unpredictability entropy, oracle complexity, and witness compressibility can only be defined up to an additive logarithmic term (see Section 2.2).

1.3 Our Contributions

We show that for all search problems with an efficiently computable verification predicate, the following are equivalent.

(i) There exists a PPT algorithm that solves a problem S with success probability $\Theta(2^{-k})$.
(ii) There exists a PPT algorithm that makes at most k queries to $\mathcal{O}$ and solves the problem S with a constant success probability.
(iii) There exists a PPT reduction that reduces the witness size of a given instance of S to at most k with constant probability.

This implies that the three quantities, i.e., unpredictability entropy, oracle complexity, and witness compressibility are essentially equal.

From this result, we get an exact characterization of the leakage-resilience of a cryptographic assumption about the hardness of some computational problem S in terms of the best possible PPT algorithm for S. A cryptographic assumption is robust up to k bits of leakage if and only if there is an algorithm that solves

[1] Witness compression should not be confused with instance compression that has been studied in [29,20].

the corresponding problem with probability $\Theta(2^{-k})$. This provides motivation for improving the success probability of PPT algorithms for various computational problems. With this goal in mind, we present in this paper the best PPT algorithms for some problems relevant in cryptography - in particular for the learning with errors and lattice problems that have recently gained substantial importance in cryptography.

The results of this paper also raise some interesting questions in complexity theory. One question this paper draws attention to is the following: Which problems have optimal witness size, or stated differently, which problems can or cannot be efficiently reduced to problems with a smaller witness size? Combining the results of [44] with our result gives evidence that the witness size of Circuit-SAT cannot be compressed under reasonable complexity theoretic assumptions. However, for instance if we look at the 3-SAT problem, which is also an NP-complete problem, combining our results with Schöning's PPT algorithm [48] that solves 3-SAT with probability $(4/3)^{-n}$, we conclude that the witness of 3-SAT can be compressed to a $\log_2 4/3$-fraction, i.e., about 41.5% of its original size.

1.4 Organization of This Paper

In Section 2, we introduce the definitions of problems and complexity notions mentioned in the introduction. In Section 3, we prove the witness compression lemma and establish the equivalence of **(i)**, **(ii)** and **(iii)** mentioned in Section 1.2. In Section 4 we give/mention the best known PPT algorithms for some problems relevant in cryptography. In Section 5, we conclude and give a list of open problems that emerge from the results of this paper.

2 Definitions

2.1 Computational Search Problems

A computational search problem S is characterized by an instance space $\mathcal{X}$, a solution (or witness) space $\mathcal{W}$, and a (verification) predicate $V : \mathcal{X} \times \mathcal{W} \to \{0,1\}$. Each element of $\mathcal{X}$ and $\mathcal{W}$ is assumed to be represented as a bitstring. In this paper, unless otherwise stated, we consider problems for which there is a polynomial time algorithm that computes the predicate V. We call this set of problems $\mathcal{PC}$.[2]

The instance space $\mathcal{X}$ can be partitioned into two sets: the set $\mathcal{X}_1$ and $\mathcal{X}_0$ of instances for which there exists a witness and for which there exists no witness, respectively, i.e.,

$$\mathcal{X}_1 := \{x \in \mathcal{X} \mid \exists\, w \in \mathcal{W},\ V(x,w) = 1\}\ , \text{ and}$$

$$\mathcal{X}_0 := \{x \in \mathcal{X} \mid \forall\, w \in \mathcal{W},\ V(x,w) = 0\}\ .$$

[2] The name of this class, $\mathcal{PC}$, is taken from [26].

The sets $\mathcal{X}_1$ and $\mathcal{X}_0$ are sometimes referred to as the set of YES instances and that of NO instances, respectively.

We define $\gamma^V : \mathcal{X}_1 \mapsto \mathbb{N}$ as the size of the smallest witness for a given x, i.e.,

$$\gamma^V(x) := \min_{w \in \mathcal{W}, V(x,w)=1} |w| \ .^3$$

A *search problem* is the problem of finding, for a given element $x \in \mathcal{X}_1$, a witness $w \in \mathcal{W}$ such that $V(x, w) = 1$. By $\mathcal{O}$, we denote the infinitely powerful oracle that can answer arbitrary binary questions. The oracle, and hence the language in which questions are asked can be defined freely, and hence need not be specified (it can be thought of as being universally quantified).

Let $p : \mathbb{N} \times \mathbb{N} \mapsto [0, 1]$ and $q : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N} \cup \{0\}$ be functions.

Definition 1. Let $\mathsf{S} = (\mathcal{X}, \mathcal{W}, V)$ be a search problem. An algorithm $\mathcal{F}$ is called a (p, q)-*solver* for S if for all $m, n \in \mathbb{N}$ and for all $x \in \mathcal{X}_1$ such that $|x| \leq m$ and $\gamma(x) \leq n$, $\mathcal{F}$ makes at most $q(m, n)$ queries to $\mathcal{O}$, and with probability at least $p(m, n)$, computes a $w \in \mathcal{W}$ such that $V(x, w) = 1$.

In the above definition, $\mathcal{F}$ is called *efficient* if it runs in time polynomial in the size of input.

2.2 Complexity Notions

Now, we introduce the notion of witness compressibility. A problem is k-witness compressible if there exists another predicate V' such that for any given instance of the problem, there exists a witness of length at most k with respect to V', and given this witness one can efficiently compute a witness with respect to V. More formally,

Definition 2. A search problem S defined by $\mathsf{S} = (\mathcal{X}, \mathcal{W}, V)$ is *(deterministic) k-witness compressible* if there exists a witness set $\mathcal{W}'$, a predicate $V' : \mathcal{X} \times \mathcal{W}' \mapsto \{0, 1\}$, and a polynomial time algorithm $T : \mathcal{X} \times \mathcal{W}' \mapsto \mathcal{W}$ such that for all $x \in \mathcal{X}_1$,

- $\gamma^{V'}(x) \leq k(|x|, \gamma^V(x))$.
- For all $w \in \mathcal{W}'$, $V'(x, w) = 1$ if and only if $V(x, T(x, w)) = 1$.

As has been often seen in complexity theory, the best known PPT algorithm/reduction is significantly faster than the best known deterministic polynomial time algorithm/reduction, e.g. primality testing. In fact sometimes the former exists but the latter eludes discovery. Thus it is reasonable to look at the following randomized version of the above definition.

Definition 3. A search problem S defined by $\mathsf{S} = (\mathcal{X}, \mathcal{W}, V)$ is *k-witness compressible* within ϵ if there exists a witness set $\mathcal{W}'$, an efficiently samplable random variable S that takes values from a set $\mathcal{S}$, a set of predicates $V'_S : \mathcal{X} \times \mathcal{W}' \mapsto \{0, 1\}$, and polynomial time algorithms $T_S : \mathcal{X} \times \mathcal{W}' \mapsto \mathcal{W}$ parametrized by S such that for all $x \in \mathcal{X}_1$,

[3] We omit the predicate V if it is clear from the context.

– $\Pr\left(\gamma^{V'_S}(x) \leq k\left(|x|, \gamma^V(x)\right)\right) \geq 1 - \epsilon(|x|, \gamma^V(x))$.[4]
– For all $w \in \mathcal{W}', s \in \mathcal{S}$, $V'_s(x, w) = 1$ if and only if $V(x, T_s(x, w)) = 1$.

With these definitions in place we now define the three quantities that we show, in this paper, are (essentially) equal. Let $k = k(m, n)$ be some integer valued function.

Definition 4. A search problem S has *unpredictability entropy at most* k if there exists an efficient $(2^{-k}, 0)$-solver for S.

Definition 5. A search problem S has *oracle-complexity at most* k if there exists an efficient $(1 - \epsilon, k)$-solver for S for some negligible function $\epsilon(m, n)$.[5]

Definition 6. A search problem S *is* k*-witness compressible* if S is k-witness compressible within ϵ for some negligible function $\epsilon(m, n)$.

Note that in these definitions, $k(m, n)$ is unique only up to an additive term of $O(\log_2 m)$. Also note that we can have an alternative version of these definitions where, for instance, the unpredictability entropy is *equal* to $k(m, n)$ (again, up to an additive term of $O(\log_2 m)$) by saying that there exists an efficient $(2^{-k(m,n)}, 0)$-solver but no efficient $(2^{-k(m,n)+\omega(\log_2 m)}, 0)$-solver for S. However, it would be cumbersome to make these alternative definitions precise and so we avoid them.

3 Relations between Complexity Notions for Search Problems

3.1 Two Simple Results

In this section, we give two simple relations between complexity notions for search problems.

Lemma 1. *For any search problem* S *and any functions* $p = p(m, n)$, $q = q(m, n)$, *and* $k = k(m, n) \leq q(m, n)$, *if there exists an efficient* (p, q)*-solver for* S, *then there exists an efficient* $(p \cdot 2^{-k}, q - k)$*-solver for* S.

Proof. Let S be a search problem and let $\mathcal{F}$ be an efficient (p, q)-solver for S. Let $\mathcal{F}'$ be an algorithm that simulates $\mathcal{F}$ except that it guesses the answer to the last k oracle queries uniformly at random. Thus $\mathcal{F}'$ makes $q - k$ queries and guesses the answer to the k queries correctly with probability 2^{-k} and hence succeeds in solving S with probability at least $p \cdot 2^{-k}$.

It is folklore as observed by a number of papers, e.g., [40,3,4] that non-adaptive leakage-resilience is the same as adaptive leakage-resilience. This can be seen in our terminology by the following lemma.

[4] The witness length is at most k with probability at least $1 - \epsilon$, where k and ϵ are both functions of $|x|$ and $\gamma^V(x)$.
[5] The term negligible, like the term efficient, is in terms of the input size m. So, for any m large enough, and any n, and any polynomial $P(.)$, $\epsilon(m, n) < 1/P(m)$.

Lemma 2. *For any functions $k = k(m,n)$ and $\epsilon = \epsilon(m,n)$, every search problem is k-witness compressible within ϵ if and only if it has an efficient $(1-\epsilon, k)$-solver.*

Proof. ($\Rightarrow$) The idea is that, with probability $1-\epsilon$, the witness size of an instance is reduced to size k, and hence we can use k queries to $\mathcal{O}$ to obtain a witness for the resulting instance.

Let $\mathsf{S} = (\mathcal{X}, \mathcal{W}, V)$ be the search problem. There exists some $\mathcal{W}'$, $V'_S : \mathcal{X} \times \mathcal{W}' \mapsto \{0,1\}$ and $T_S : \mathcal{X} \times \mathcal{W}' \mapsto \mathcal{W}$ as in Definition 3. We give a polynomial time algorithm $\mathcal{F}$ that is a $(1-\epsilon, k)$-solver for S. On input $x \in \mathcal{X}$, $\mathcal{F}$ generates $S = s$ and then uses k queries to $\mathcal{O}$ to ask for w', the string formed from the last k bits of a smallest length witness $w \in \mathcal{W}'$ (if it exists) such that $V'_s(x,w) = 1$. Then the algorithm outputs $T_s(x,w')$.

Let $m = |x|$ and $n = \gamma^V(x)$. With probability at least $1-\epsilon$, $S = s$ such that the conditions of Definition 3 hold. Thus, $w' = w$ since $\gamma^{V'_s}(x) \leq k$. Hence $V'_s(x,w') = 1$, which implies $V(x, T_s(x,w')) = 1$.

($\Leftarrow$) Let $\mathcal{F}$ be a $(1-\epsilon, k)$-solver for S. Define $\mathcal{W}'$ as the set of all bitstrings and let S denote the random choices made by $\mathcal{F}$. Define $T_s(x,w)$ to be the output of $\mathcal{F}$ on input x, $S = s$ and the result of the oracle queries equal to w. Further, define $V'_s(x,w)$ as $V(x, T_s(x,w))$. This gives the desired result.

3.2 The Witness Compression Lemma

We state a few lemmas that we need in order to prove the main lemma of this section.

Lemma 3. *Let $Y_1, \ldots, Y_t$ be pairwise independent binary random variables where $\Pr(Y_i = 1) = p$ for $1 \leq i \leq t$. Then*

$$\Pr(\exists i \in \{1,\ldots,t\} \ : \ Y_i = 1) \geq \max(tp - \frac{t^2p^2}{2}, 1 - \frac{1}{tp})$$

Proof. We give two ways to bound the term on the left. Using Bonferroni inequalities [15],

$$\begin{aligned}
\Pr(\exists i \in \{1,\ldots,t\} \ : \ Y_i = 1) &= \Pr(Y_1 = 1 \vee Y_2 = 1 \vee \cdots \vee Y_t = 1) \\
&\geq \sum_{1 \leq i \leq t} \Pr(Y_i = 1) - \sum_{1 \leq i_1 < i_2 \leq t} \Pr(Y_{i_1} = 1 \wedge Y_{i_2} = 1) \\
&= tp - \frac{t(t-1)}{2} p^2 \\
&\geq tp - \frac{t^2p^2}{2} \ .
\end{aligned}$$

Now, let $Y = Y_1 + \cdots + Y_t$. The expected value of Y is $E(Y) = tp$ and the variance of Y is $Var(Y) = tp(1-p)$. Thus,

$$\begin{aligned}\Pr\left(\exists i \in \{1,\ldots,t\} \ : \ Y_i = 1\right) &= 1 - \Pr(Y = 0) \\ &\geq 1 - \Pr\left(|Y - E(Y)| \geq E(Y)\right) \\ &\geq 1 - \frac{Var(Y)}{E(Y)^2} \\ &= 1 - \frac{1-p}{tp} \geq 1 - \frac{1}{tp}\ ,\end{aligned}$$

where the second last inequality follows from the Chebyshev's inequality.

Lemma 4. *Let $\mathbb{F}$ be a finite field of cardinality 2^ℓ, let ϕ be a bijection from $\mathbb{F}$ to $\{0,1\}^\ell$, and let $T \subset \{0,1\}^\ell$. Further, let $y_1,\ldots,y_t$ be some fixed distinct elements of $\mathbb{F}$. Then, for randomly chosen $A, B \in_R \mathbb{F}$, the probability that $\phi\left(Ay_i + B\right) \in T$ for some $1 \leq i \leq t$ is at least* $\max\left(\frac{t|T|}{2^\ell} - \frac{t^2|T|^2}{2(2^\ell)^2}, 1 - \frac{2^\ell}{t|T|}\right)$.[6]

Proof. Define binary random variables $Y_1,\ldots,Y_t$ such that $Y_i = 1$ if $\phi\left(Ay_i + B\right) \in T$. Thus,

$$\Pr(Y_i = 1) = \frac{|T|}{2^\ell}\ ,$$

and it can be easily seen that the Y_i's are pairwise independent random variables. Therefore, by Lemma 3, the probability that $\phi\left(Ay_i + B\right) \in T$ for some $1 \leq i \leq t$ is at least

$$\max\left(\frac{t|T|}{2^\ell} - \frac{t^2|T|^2}{2(2^\ell)^2}, 1 - \frac{2^\ell}{t|T|}\right).$$

Now, we state the main lemma of this section.

Lemma 5. ***[Witness Compression Lemma]*** *Let $k = k(m,n)$ and $k' = k'(m,n) \geq k(m,n)$ be any functions. Every search problem with an efficient $(2^{-k}, 0)$-solver is k'-witness compressible within $\frac{1}{2^{k'-k}}$.*

Proof. Let $\mathsf{S} = (\mathcal{X}, \mathcal{W}, V)$ be a search problem and let $\mathcal{F}$ be an efficient $(2^{-k}, 0)$-solver for S. For a given input instance $x \in \mathcal{X}_1$, let $R \in \{0,1\}^\ell$ denote the random choices made by $\mathcal{F}$. Then,

$$\Pr\left(V(x, \mathcal{F}\left(x, R\right)) = 1\right) \geq 2^{-k}\ . \tag{1}$$

We define the set $\mathcal{R}\left(x\right)$ as the set of r such that $\mathcal{F}$ is successful in finding a witness for x for this choice of r, i.e.,

$$\mathcal{R}\left(x\right) = \left\{r \in \{0,1\}^\ell \mid V(x, \mathcal{F}\left(x, r\right)) = 1\right\}\ .$$

From (1), it follows that $|\mathcal{R}\left(x\right)| \geq 2^{\ell-k}$ for all $x \in \mathcal{X}_1$.

Now, let $\mathbb{F}, \phi, A, B$ and $y_1,\ldots,y_t$ be as in Lemma 4. Thus, by using the second bound from Lemma 4, with $t = 2^{k'}$, and $T = \mathcal{R}(x)$, we get

$$\Pr\left(\exists 1 \leq i \leq 2^{k'} : \phi\left(Ay_i + B\right) \in \mathcal{R}(x)\right) \geq 1 - \frac{1}{2^{k'-k}}\ .$$

[6] Note that the result of this lemma will hold for any pairwise independent random function from $\mathbb{F}$ to itself, instead of $Ay + B$.

Then, let $S = (A, B)$ be uniformly distributed over $\mathbb{F} \times \mathbb{F}$. Furthermore, define $\mathcal{W}' = \{0,1\}^*$,

$$T_S(x, w) = \mathcal{F}(x, \phi(Ay_w + B)), \text{ and } V'_S(x, w) = V(x, T_S(x, w)) \ .$$

In the above argument, we can also use the first bound from Lemma 4 to show that the problem is k-witness compressible within $\frac{1}{2}$.

3.3 The Main Result

Combining the results of Lemma 1, 2, and 5, we get the following result:

Theorem 1. *For any search problem* S*, and for any functions* $k = k(m, n)$ *and* $c = c(m, n) = \omega(\log_2 m)$*:*

- *If* S *is* k*-witness compressibile, then* S *has oracle complexity at most* k*.*
- *If* S *has oracle complexity at most* k*, then* S *has unpredictability entropy at most* k*.*
- *If* S *has unpredictability entropy at most* k*, then* S *is* $k + c$*-witness compressibile.*

Note that the results of this section are useful only if $k(m, n) = \omega(\log_2 m)$ because otherwise the corresponding search problem is solvable in expected polynomial time. Thus, without loss of generality, we can assume $k(m, n) = \omega(\log_2 m)$ and then choosing $c(m, n)$ as any function asymptotically smaller than k but larger than $\log_2 m$ (e.g. $c = \sqrt{k}$), we get that the three quantities in Theorem 1 are essentially equivalent for functions in $k + o(k)$.

Remark 1: Theorem 1 implies that an assumption of the hardness of a search problem S is secure up to k bits of leakage of arbitrary information if and only if there is no PPT algorithm that succeeds in solving S with probability $\Theta(2^{-k})$. However, the hardness assumptions we consider are worst case assumptions and not average case assumptions, which are more relevant in practice. Note that this is not a disadvantage, since our result implies a corresponding result for average case assumptions, just by restricting the set of instances of the problem to those where the problem is successful with significant (though possibly exponentially small) probability.

Remark 2: A similar result as Theorem 1 can also be proved for decision problems (using essentially the same proofs) but for that we need to be more careful in defining the oracle complexity of a problem and also the success probability of a PPT algorithm and we do not do so in this version of the paper.

4 PPT Algorithms for Problems Relevant in Cryptography

In this section, we give the best PPT algorithms known for various search problems relevant in cryptography. We look in more detail at the learning with errors and lattice problems that have been of interest in cryptography in recent years.

4.1 Factoring and Discrete Logarithms

There is a sequence of results [47,11,30] that show that partial information about p and q is enough to factor the RSA modulus pq. The best result in this direction is the result in [37] that, under a conjecture, shows that there is a polynomial time algorithm that factors N given $\epsilon \log_2 N$ questions to $\mathcal{O}$ where ϵ is some arbitrary constant. Equivalently, there exists a PPT algorithm that factors N with probability $2^{-\epsilon \log_2 N}$.

Even though the problem of computing discrete logarithms modulo a prime is closely related to the problem of factoring integers, to the best of our knowledge, there exists no non-trivial PPT algorithm for solving discrete logarithms in $\mathbb{Z}_p$. The same holds for the Computational Diffie Hellman problem.

It would be interesting to come up with an algorithm for solving discrete logarithm modulo a prime p that runs in time polynomial in $\log_2 p$ and succeeds with probability better than the trivial $\frac{\text{poly}(\log_2 p)}{p}$.

4.2 Lattices

Preliminaries An n-dimensional lattice is a discrete additive subgroup of $\mathbb{R}^n$. A set of linearly independent vectors that generates a lattice is called a basis and is denoted by $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$. The lattice Λ generated by the basis $\mathbf{B}$ is

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{Bz} = \sum_{i=1}^{n} z_i \mathbf{b}_i \ : \ z \in \mathbb{Z}^n \right\} .$$

For any point $\mathbf{t} \in \mathbb{R}^n$, the distance of $\mathbf{t}$ to the closest point in the lattice is written as $\text{dist}(\mathbf{t}, \Lambda)$.

The Gram Schmidt orthogonalization of $\mathbf{B}$, denoted as $\{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n\}$, is defined as

$$\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \tilde{\mathbf{b}}_j, \text{ where } \mu_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} .$$

By $\lambda_1(\Lambda)$, we denote the length of the shortest non-zero vector of the lattice Λ. For this paper, the lengths are always assumed to be in the ℓ_2 norm. If the lattice is clear from the context, then we write it simply as λ_1. It is well known and can be shown easily that

$$\lambda_1 \geq \min_i \|\tilde{\mathbf{b}}_i\| .$$

Definition 7. A basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is a *δ-LLL Reduced Basis* [35] if the following holds:

- $\forall\, 1 \leq j < i \leq n$, $\mu_{i,j} \leq \frac{1}{2}$,
- $\forall\, 1 \leq i < n$, $\delta \|\tilde{\mathbf{b}}_i\|^2 \leq \|\mu_{i+1,i} \tilde{\mathbf{b}}_i + \tilde{\mathbf{b}}_{i+1}\|^2$.

We choose $\delta = \frac{3}{4}$ and then it can be easily seen (e.g., refer to [25]) from the above definition that for a δ-LLL reduced basis, $\forall\, 1 \leq i < n$, $\|\mathbf{b}_i\| \leq \sqrt{2}\|\mathbf{b}_{i+1}\|$. Since there is an efficient algorithm [35] to compute an LLL-reduced basis, we assume, unless otherwise stated, that the given basis is always LLL-reduced and hence satisfies the above mentioned properties.

Now, we define some problems over lattices that we are interested in for this paper.

Definition 8. The *shortest vector problem* is defined as follows: Given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, it is required to find a vector $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v}\| = \lambda_1$.

A decision variant, whose hardness many cryptographic schemes are based on, is the gap shortest vector problem defined as follows.

Definition 9. The *gap shortest vector problem* GapSVP_γ for some $\gamma = \gamma(n)$ is defined as follows: Given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$ and $d > 0$ such that $d \notin [\lambda_1/\gamma, \lambda_1)$, decide whether $d \geq \lambda_1$ or $d < \lambda_1/\gamma$.

Next we define the closest vector problem (CVP) and bounded distance decoding (BDD) which is a special case of the CVP.

Definition 10. The *closest vector problem* CVP is defined as follows: Given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, and $\mathbf{t} \in \mathbb{R}^n$, find $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v} - \mathbf{t}\| = \mathrm{dist}(\mathbf{t}, \Lambda)$.

Definition 11. The *α-bounded distance decoding problem* BDD_α for some $0 < \alpha = \alpha(n) < 1/2$ is defined as follows: Given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, and $\mathbf{t} \in \mathbb{R}^n$ such that $\mathrm{dist}(\mathbf{t}, \Lambda) \leq \alpha\lambda_1$, find $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v} - \mathbf{t}\| = \mathrm{dist}(\mathbf{t}, \Lambda)$.

Shortest Vector Problem In this section, we give a polynomial time algorithm that computes the shortest vector of a lattice with probability $\frac{1}{2^{(n+1)(n+2)/4}}$. This algorithm, of course, also solves the GapSVP problem.

Theorem 2. *There exists a polynomial algorithm that, given a basis $\mathbf{B}$ of a lattice $\Lambda = \mathcal{L}(\mathbf{B})$, finds the shortest vector of Λ with probability $\frac{1}{2^{(n+1)(n+2)/4}}$.*

Proof. Since an LLL-reduced basis can be computed efficiently, we assume without loss of generality that $\mathbf{B}$ is an LLL-reduced basis. Let the shortest vector $\mathbf{u}$ of the lattice be $\mathbf{u} = a_1\tilde{\mathbf{b}}_1 + a_2\tilde{\mathbf{b}}_2 + \cdots + a_n\tilde{\mathbf{b}}_n$. Since $\tilde{\mathbf{b}}_1 = \mathbf{b}_1$ is a lattice vector, therefore $\|\mathbf{u}\| \leq \|\tilde{\mathbf{b}}_1\|$. By the property of the LLL basis, $\|\tilde{\mathbf{b}}_1\| \leq 2^{(i-1)/2}\|\tilde{\mathbf{b}}_i\|$, which implies $\|\mathbf{u}\| \leq 2^{(i-1)/2}\|\tilde{\mathbf{b}}_i\|$. Thus, $|a_i| \leq 2^{(i-1)/2}$. The component a_i is determined by the coefficients of $\mathbf{b}_i, \ldots, \mathbf{b}_n$ in $\mathbf{u}$. Thus, given the coefficients of $\mathbf{b}_n, \ldots, \mathbf{b}_{i+1}$, the coefficient of $\mathbf{b}_i$ can be chosen correctly with probability $1/(2 \cdot 2^{(i-1)/2}) = 2^{-(i+1)/2}$. This gives a polynomial time algorithm that succeeds in finding the shortest vector with probability

$$\prod_{i=1}^{n} 2^{-(i+1)/2} = 2^{-(n+1)(n+2)/4}\,.$$

Closest Vector Problem In this section, we give a polynomial time algorithm that solves the closest vector problem with probability $\frac{1}{2^{n(n+1)/4}}$.

Theorem 3. *There exists a polynomial time algorithm that, given a basis* $\mathbf{B}$ *of an* n*-dimensional lattice* $\Lambda = \mathcal{L}(\mathbf{B})$*, and* $\mathbf{t} \in \mathbb{R}^n$*, finds* $\mathbf{v} \in \Lambda$ *such that* $\|\mathbf{v} - \mathbf{t}\| = dist(\mathbf{t}, \Lambda)$ *with probability* $\frac{1}{2^{n(n+1)/4}}$.

Proof. Let $\mathbf{t} = \rho_1 \tilde{\mathbf{b}}_1 + \ldots \rho_n \tilde{\mathbf{b}}_n$ and let the closest vector to $\mathbf{t}$ in the lattice be $\mathbf{u} = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_n\mathbf{b}_n$. Babai's algorithm [5] returns a vector $\mathbf{x}$ such that $\|\mathbf{x} - \mathbf{t}\| \leq \frac{1}{2}2^{n/2}\|\tilde{\mathbf{b}}_n\|$. Thus $\|\mathbf{u} - \mathbf{t}\| \leq \|\mathbf{x} - \mathbf{t}\| \leq \frac{1}{2}2^{n/2}\|\tilde{\mathbf{b}}_n\|$, which implies $|a_n - \rho_n| \leq \frac{1}{2}2^{n/2}$. Thus the algorithm proceeds as follows: Choose $\hat{a}_n$ uniformly at random from $(\rho_n - \frac{1}{2}2^{n/2}, \rho_n + \frac{1}{2}2^{n/2})$ and recursively compute the closest vector to $\mathbf{t} - \hat{a}_n\mathbf{b}_n$ in the lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$. The probability that $(\hat{a}_1, \ldots, \hat{a}_n) = (a_1, \ldots, a_n)$ is

$$\prod_{i=1}^{n} 2^{-i/2} = 2^{-n(n+1)/4} .$$

Bounded Distance Decoding (BDD) Problem The algorithm given in the previous section, of course, also solves the BDD problem since BDD is a special case of the closest vector problem. However, there exists an algorithm for BDD_α with a larger success probability $\frac{1}{\alpha^n 2^{(n+1)(n+2)/4}}$ as given below.

Theorem 4. *There exists a polynomial time algorithm that, given a basis* $\mathbf{B}$ *of an* n*-dimensional lattice* $\Lambda = \mathcal{L}(\mathbf{B})$*, and* $\mathbf{t} \in \mathbb{R}^n$ *such that* $dist(\mathbf{t}, \Lambda) \leq \alpha\lambda_1$ *for some* $0 < \alpha(n) < 1/2$*, finds* $\mathbf{v} \in \Lambda$ *such that* $\|\mathbf{v} - \mathbf{t}\| = dist(\mathbf{t}, \Lambda)$ *with probability* $\frac{1}{\alpha^n 2^{(n+1)(n+2)/4}}$.

Proof. Since an LLL-reduced basis can be computed efficiently, we assume without loss of generality that $\mathbf{B}$ is an LLL-reduced basis. Let $\mathbf{t} = t_1\tilde{\mathbf{b}}_1 + t_2\tilde{\mathbf{b}}_2 + \cdots + t_n\tilde{\mathbf{b}}_n$ and the closest vector $\mathbf{u}$ of the lattice be $\mathbf{u} = u_1\tilde{\mathbf{b}}_1 + u_2\tilde{\mathbf{b}}_2 + \cdots + u_n\tilde{\mathbf{b}}_n$. Since $\tilde{\mathbf{b}}_1 = \mathbf{b}_1$ is a lattice vector, therefore $\|\mathbf{u} - \mathbf{t}\| \leq \alpha\|\tilde{\mathbf{b}}_1\|$. By the property of the LLL basis, $\|\tilde{\mathbf{b}}_1\| \leq 2^{(i-1)/2}\|\tilde{\mathbf{b}}_i\|$, which implies $\|\mathbf{u} - \mathbf{t}\| \leq 2^{(i-1)/2}\alpha\|\tilde{\mathbf{b}}_i\|$. Thus, $|u_i - t_i| \leq \alpha 2^{(i-1)/2}$. The component u_i is determined by the coefficients of $\mathbf{b}_i, \ldots, \mathbf{b}_n$ in $\mathbf{u}$. Thus, given the coefficients of $\mathbf{b}_n, \ldots, \mathbf{b}_{i+1}$, the coefficient of $\mathbf{b}_i$ can be chosen correctly with probability $1/(2\alpha \cdot 2^{(i-1)/2}) = 2^{-(i+1)/2}\alpha^{-1}$. This gives a polynomial time algorithm that succeeds in finding the shortest vector with probability

$$\prod_{i=1}^{n} \frac{1}{2^{(i+1)/2}\alpha} = \frac{1}{\alpha^n 2^{(n+1)(n+2)/4}} .$$

4.3 Learning with Errors and Its Relation to Lattice Problems

In this section, we mention the best PPT algorithm for the learning with errors (LWE) problem and its relation to the lattice problems with respect to leakage. The proofs and other details are omitted.

Theorem 5. *For some function $\beta = \beta(n)$ such that $\beta q = \omega(\log_2 n)$, $\beta(n) = o(1/\log_2 n)$ and $m \geq n$, there is a polynomial time algorithm that solves search-$\mathsf{LWE}_{n,q,m,\overline{\psi}_\beta}$ with probability $(\beta q \log_2 n)^{-n}$ for a constant fraction α of the inputs.*

Note that it is straightforward to interpret the above mentioned algorithms for LWE and lattice problems as PPT algorithms that succeed with constant probability given $-\log_2 p$ queries to $\mathcal{O}$ (or equivalently $-\log_2 p$ bits of leakage), where p is the success probability of the algorithm. We do not need the witness compression lemma to make this conclusion. The witness compression lemma however implies that if there is any PPT algorithm for any of these problems that succeeds with probability $p' < p$, then there is a PPT algorithm that makes $-\log_2 p'$ queries to $\mathcal{O}$ and succeeds with constant probability.

It is common practice to base the LWE-based schemes on the hardness of lattice based schemes. In the same spirit, by a careful inspection of the reduction of BDD to LWE from [46], we get the following result:

Theorem 6. *If there exists a PPT algorithm that solves search-$\mathsf{LWE}_{n,q,m,\psi_\beta}$ with probability p then there exists a PPT algorithm that solves $\mathsf{BDD}_{\frac{\beta}{n}}$ with probability $cp^{\lceil n/\log_2 q \rceil}$) for some constant c.*

By Theorem 6, we can base the LWE assumption with leakage on the exponential hardness of the BDD assumption as follows.

Corollary 1. *If there exists no polynomial time algorithm that solves $\mathsf{BDD}_{\frac{\beta}{n}}$ with probability $2^{-\delta n^2}$, then the search-$\mathsf{LWE}_{n,q,m,\psi_\beta}$ assumption is robust to $\delta n \log_2 q - o(\log_2 q)$ bits of leakage.*

5 Conclusions and Open Problems

We show that the unpredictability entropy of a problem is equal to its leakage-resilience limit. This provides motivation to look at PPT algorithms for problems relevant in cryptography with maximum possible success probability. A question that is wide open is to what extent can the success probability of PPT algorithms be improved for various problems like the discrete logarithm problem, search LWE problem or various lattice problems. Note that if we repeatedly run algorithms for lattice problems given in Section 5 to amplify the success probability to a non-negligible quantity, we get algorithms with running time $2^{O(n^2)}$, which is much worse than the best known algorithms that run in time $2^{O(n)}$ [1,39]. Due to this large gap, one might expect that it should be possible to improve the success probability of a PPT algorithm and this has eluded discovery because of lack of attention to this question.

The witness compression lemma implies that the best known PPT algorithms, for instance [6,7,9,12,17,18,19,23], immediately give a lower bound on the maximum witness compressibility of the corresponding problems.

The results of [44] give evidence that perhaps Circuit-SAT is not witness compressible to any non-trivial witness size. In fact the result of [44], which

shows that there exists no non-trivial PPT algorithm for Circuit-SAT (and hence for all **NP** problems) under reasonable complexity assumptions, can be proved by proving a decision version of the witness compression lemma. If there exist non-trivial PPT algorithms for all **NP** problems, we can repeatedly apply the witness compression lemma until the witness size is reduced to a constant, thus resulting in a sub-exponential time algorithm for any **NP** problem, which is not believed to be possible. It is interesting to look at the question of which are the other problems that, like Circuit-SAT are not witness compressible. The discrete logarithm problem modulo a prime seems to be a candidate.

Another interesting research direction is to look at PPT-reductions, i.e., PPT algorithms for solving one "hard" problem given a PPT algorithm for solving another problem (with possibly negligible success probability). Consider, for instance, the reduction of [36] from GapSVP to BDD. This reduction was derived from the main idea of [41] in obtaining the first public key cryptosystem whose hardness was based on the GapSVP. This reduction does not seem to translate easily to the case of PPT algorithms, since given a BDD oracle that solves the problem with an exponentially small probability, it is not clear how to use it to solve the GapSVP problem. If such a reduction was possible, we could base the leakage-resilience of the search LWE assumption on the exponential hardness of the GapSVP problem.

References

1. Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: STOC 2001, pp. 601–610 (2001)
2. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous Hardcore Bits and Cryptography Against Memory Attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
3. Alwen, J., Dodis, Y., Wichs, D.: Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009)
4. Alwen, J., Dodis, Y., Wichs, D.: Survey: Leakage Resilience and the Bounded Retrieval Model. In: Kurosawa, K. (ed.) Information Theoretic Security. LNCS, vol. 5973, pp. 1–18. Springer, Heidelberg (2010)
5. Babai, L.: On Lovász' Lattice Reduction and the Nearest Lattice Point Problem. Combinatorica 6(1), 1–13 (1986)
6. Beigel, R., Eppstein, D.: 3-coloring in Time $o(1.3446^n)$: A No-mis Algorithm. In: FOCS 1995, pp. 444–452 (1995)
7. Beigel, R.: Finding Maximum Independent Sets in Sparse and General Graphs. In: SODA 1999, pp. 856–857 (1999)
8. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant Learning, the Parity Problem, and the Statistical Query Model. Journal of the ACM 50(4), 506–519 (2003)
9. Byskov, J.: Algorithms for k-colouring and Finding Maximal Independent Sets. In: SODA 2003, pp. 456–457 (2003)
10. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-Resilient Functions and All-or-Nothing Transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000)

11. Coppersmith, D.: Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 178–189. Springer, Heidelberg (1996)
12. Dantsin, E., Goerdt, A., Hirsch, E., Kannan, R., Kleinberg, J., Papadimitriou, C., Raghavan, P., Schöning, U.: A Deterministic $(2-2/(k+1))^n$ Algorithm for k-SAT Based on Local Search. Theoretical Computer Science 289(1), 69–83 (2002)
13. Dodis, Y., Goldwasser, S., Kalai, Y., Peikert, C., Vaikuntanathan, V.: Public-Key Encryption Schemes with Auxiliary Inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)
14. Dodis, Y., Kalai, Y., Lovett, S.: On Cryptography with Auxiliary Input. In: STOC 2009, pp. 621–630 (2009)
15. Dohmen, K.: Improved Bonferroni Inequalities with Applications: Inequalities and Identities of Inclusion-Exclusion Type. Springer, Berlin (2003)
16. Dziembowski, S., Pietrzak, K.: Leakage-resilient Cryptography. In: FOCS 2008, pp. 293–302 (2008)
17. Eppstein, D.: Improved Algorithms for 3-coloring, 3-edge-coloring, and Constraint Satisfaction. In: SODA 2001, pp. 329–337 (2001)
18. Eppstein, D.: Small Maximal Independent Sets and Faster Exact Graph Coloring. Journal of Graph Algorithms and Applications 7, 131–140 (2003)
19. Fomin, F., Grandoni, F., Kratsch, D.: Measure and Conquer: A Simple $o(2^{0.288n})$ Independent Set Algorithm. In: SODA 2006, pp. 18–25 (2006)
20. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. Journal of Computer and System Sciences 77(1), 91–106 (2011)
21. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.: Leakage-Resilient Signatures. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 343–360. Springer, Heidelberg (2010)
22. Goldreich, O., Goldwasser, S.: On the Limits of Nonapproximability of Lattice Problems. Journal of Computation and Systems Sciences 60(3), 540–563 (2000)
23. Gramm, J., Hirsch, E., Niedermeier, R., Rossmanith, P.: Worst Case Upper Bounds for Max-2-sat with an Application to Max-cut. Discrete Applied Mathematics 130(2), 139–155 (2003)
24. Goldwasser, S., Kalai, Y., Peikert, C., Vaikuntanathan, V.: Robustness of the Learning With Errors Assumption. In: ICS 2010. Tsinghua University Press, Beijing (2010)
25. Goldwasser, S., Micciancio, D.: Complexity of Lattice Problems: a Cryptographic Perspective. The Kluwer International Series in Engineering and Computer Science, vol. 671. Kluwer Academic Publishers, Boston
26. Goldreich, O.: Computational Complexity: A Conceptual Perspective. Cambridge University Press, NY
27. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197–206 (2008)
28. Gregg, J.: On Factoring Integers and Evaluating Discrete Logarithms. Bachelor's Thesis. Harvard College, Cambridge, Massachusetts
29. Harnik, D., Naor, M.: On the compressibility of NP instances and cryptographic applications. In: FOCS 2006, pp. 719–728 (2006)
30. Heninger, N., Shacham, H.: Reconstructing RSA Private Keys from Random Key Bits. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 1–17. Springer, Heidelberg (2009)
31. Hsiao, C., Lu, C., Reyzin, L.: Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 169–186. Springer, Heidelberg (2007)

32. Klein, P.: Finding the Closest Vector When it is Unusually Close. In: SODA 2000, pp. 937–941 (2000)
33. Katz, J., Vaikuntanathan, V.: Signature Schemes with Bounded Leakage Resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)
34. Lenstra, H.: Factoring Integers with Elliptic Curves. Annals of Mathematics 126, 649–673 (1987)
35. Lenstra, A., Lenstra, H., Lovász, L.: Factoring Polynomials wth Rational Coefficients. Mathematische Annalen 261(4), 515–534 (1982)
36. Lyubashevsky, V., Micciancio, D.: On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 577–594. Springer, Heidelberg (2009)
37. Maurer, U.: On the Oracle Complexity of Factoring Integers. Computational Complexity 5(4), 237–247 (1996)
38. Micali, S., Reyzin, L.: Physically Observable Cryptography. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
39. Micciancio, D., Voulgaris, P.: A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In: STOC 2010, pp. 351–358 (2010)
40. Naor, M., Segev, G.: Public-key Cryptosystems Resilient to Key Leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
41. Peikert, C.: Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. In: STOC 2009 (2009)
42. Pietrzak, K.: A Leakage-Resilient Mode of Operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009)
43. Petit, C., Standaert, F., Pereira, O., Malkin, T., Yung, M.: A Block Cipher Based Pseudo Random Number Generator Secure Against Side-channel Key Recovery. In: ASIACCS 2008, pp. 56–65 (2008)
44. Paturi, R., Pudlák, P.: On the Complexity of Circuit Satisfiability. In: STOC 2010 (2010)
45. Paturi, R., Pudlák, P., Zane, F.: Satisfiability Coding Lemma. In: FOCS 1997, pp. 566–574 (1997)
46. Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In: STOC 2005 (2005)
47. Rivest, R., Shamir, A.: Efficient Factoring Based on Partial Information. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 31–34. Springer, Heidelberg (1986)
48. Schöning, U.: A Probabilistic Algorithm for k-SAT and Constraint Satisfaction Problems. In: FOCS 1999 (1999)
49. Sipser, M.: A Complexity Theoretic Approach to Randomness. In: STOC 1983, pp. 330–335 (1983)
50. Stockmeyer, L.: The Complexity of Approximate Counting. In: STOC 1983, pp. 118–126 (1983)
51. Valiant, L., Vazirani, V.: NP is as Easy as Detecting Unique Solutions. Theoretical Computer Science 47, 85–93 (1986)