# Minimalist Cryptography: Excerpt from an NSF Proposal

Tal Malkin

February 13th, 2016

**Abstract**

This is part of a proposal submitted to NSF together with Allison Bishop Lewko in 2014 (and subsequently awarded). This part was written by Tal Malkin, influenced by discussions with various people (most notably Christina Brzuska, Siyao Guo, Igor Carboni Oliveira, and Alon Rosen). I reproduce it here for students in my advanced cryptography class, to give some possible ideas for a project combining cryptography and complexity (among other suggestions given to students). Note that some progress on the problems described here has been achieved, and I give some pointers in the last "addendum" section. However, most of the directions here remain open and interesting to explore. Finally, note that this, by nature of being a proposal, was written in a more informal and optimistic way than a research paper describing completed work would be written.

## 1 Complexity of Pseudorandom Primitives

Pseudorandom generators (PRG) and pseudorandom functions (PRF) are fundamental cryptographic primitives, well studied in the theoretical community, and widely used (often under the names "stream ciphers" and "block ciphers") in practice. Classic results in cryptography prove that PRG and PRF are equivalent to each other, and to many other fundamental cryptographic primitives such as one way functions (OWF), signatures, symmetric encryption, authentication, and more, where equivalence is defined by the existence of a polynomial time reduction. However, these primitives are not all created equal. For example, if $G$ is a PRG, it is immediately also a OWF, while constructing PRG from OWF is much more complex in various ways. As another example, given a PRF it is easy to construct a PRG of the same parallel-time complexity, while the GGM construction of PRF from PRG [13] incurs a multiplicative linear blow up in the circuit depth (as the PRG is applied sequentially for each bit of the input). There is also a wide gap between the efficiency of theoretical implementations of PRF and the corresponding designs used in practice (e.g., AES). The former have provable security based on well studied computational assumptions, but with a high price in performance. This motivates the following.

**Research Goals:** *minimize the complexity of pseudorandom primitives while keeping assumptions minimal, explore how large the complexity gap between PRG and PRF must be, and understand whether one can get some of the benefits of PRF at a lower price.*

Progress on these goals will help us gain a deeper theoretical understanding of these fundamental primitives, and may also serve as a step towards more practical provably secure pseudorandom primitives (for example, low depth circuits admit faster hardware and parallel implementations).

Towards these goals, we will also study weaker notions of PRF, focusing on *weak PRF (WPRF)*, which is required to be indistinguishable from a random boolean function[1] if given a polynomial number of *uniformly random* input output pairs. That is, $\{f_k : \{0,1\}^n \to \{0,1\}\}_{k\in\{0,1\}^\ell}$ is a WPRF if for any efficient adversary (distinguisher), when given polynomially many samples $(x_1, f(x_1)), \ldots (x_q, f(x_q))$ where $x_1, \ldots, x_q$ are independent uniform strings from $\{0,1\}^n$, the distinguisher cannot tell whether the function used is a random member $f_k$ of the family, or a truly random function. Note that WPRF is qualitatively stronger than PRG, in that its output must look random even when the input is known. But it is qualitatively weaker than (strong) PRF, since PRF allows the inputs to not only be seen but also chosen by the adversary.

WPRF have not been studied nearly as extensively as PRF, but this is beginning to change (cf., [22, 9, 21, 24, 10, 19, 1]. We believe that WPRF hold a lot of untapped potential for providing a good tradeoff between security and efficiency. Indeed, WPRF is sufficient – in a direct and efficient way – for many of the main applications that PRF has been used for (e.g., encryption and authentication). Moreover, as exemplified by several of the recent papers, WPRF seems to allow for significant gains in efficiency and make certain tasks easier (e.g., leakage resilience [24]). As a toy example, it is not hard to show that WPRF allow for a very simple domain extension technique, where given a WPRF $\{f_k\}$ with input size $n$, one can obtain a WPRF $\{f'_k\}$ with larger input size, by simply projecting the larger input to a fixed $n$-bit subset (say the prefix) and applying $f_k$ to it, resulting in only a small degradation in security. This is a simplification of Levin's domain extension technique for PRF, which required an almost universal hash function.

There are other reasons to focus on WPRF, including their connection to learnablity (roughly speaking, learnability of a class precludes it from having a WPRF, and a limited version of the converse can also be proved). One main motivation for us is the potential for constructing WPRF in complexity classes where PRF provably cannot exist. This direction was put forward very recently by Akavia et al. [1], who provide the first such (conjectured) WPRF candidate. We next elaborate on this and our proposed research questions.

### 1.1 Circuit Complexity of WPRF

We start with a brief reminder of some relevant complexity classes. All the classes we discuss are of polynomial size circuits, and thus we omit explicit mention of this. Recall that $\mathsf{NC}^i$ is the class of depth $O(\log^i(n))$ Boolean circuits with bounded fan-in. $\mathsf{AC}^0$ is the class of constant-depth, unbounded fan-in circuits, $\mathsf{AC}^0[m]$ allows $\mathsf{MOD_m}$ gates in addition to AND, OR, NOT (thus, $\mathsf{AC}^0[2]$ allows parity gates), and $\mathsf{AC}^0 \circ \mathsf{MOD}_2$ is a subclass of $\mathsf{AC}^0[2]$ where parity gates are allowed only on the bottom layer. $\mathsf{TC}^0$ is the class of constant depth unbounded fan-in circuits with threshold gates in addition to AND,OR,NOT. Known relationships among these classes are summarized as follows (several of these containments are believed or conjectured to be strict):

$$\mathsf{NC}^0 \subseteq \mathsf{AC}^0 \subset \mathsf{AC}^0 \circ \mathsf{MOD}_2 \subseteq \mathsf{AC}^0[2] \subset \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{NC}$$

**PRG:** It is believed that PRG exist in $\mathsf{NC}^0$. Indeed, Applebaum, Kushilevitz and Ishai [2] prove that if PRG exist in, say, $\mathsf{NC}^1$ (which is the case under standard cryptoraphic assumptions such as hardness of factoring), then it can be compiled into a PRG in $\mathsf{NC}^0$ where each bit of the output depends on only 4 bits of the input.

[1] We focus here on boolean functions. Functions with $m$ bit output can be expressed as $m$ boolean functions.

**PRF:** In contrast, PRF cannot exist even in higher classes. First, Linial, Mansour and Nisan [17] proved that $\mathsf{AC}^0$ can be learned in quasi-polynomial time, and thus there are no (exponentially secure) PRFs in $\mathsf{AC}^0$. A stronger result proving that PRF cannot exist even in $\mathsf{AC}^0[2]$, was proved by Razborov and Rudich [25], via the natural proofs barrier. On the positive side, there are constructions of PRF in $\mathsf{TC}^0$ based on concrete assumptions of DDH and LWR [22, 3]. As for PRF from generic assumptions, Naor and Reingold [22] also show constructions of PRF from WPRF (via synthesizers) of logarithmic depth (which further motivates the search for complexity of WPRF). However, no parallel ($\mathsf{NC}$) constructions of PRF from PRG are known; in fact, the linear depth of the classic GGM construction of PRF from PRG has not been improved.

**WPRF:** What are the lower and upper bounds for the complexity of WPRF? First, we note that the proof of [25] about impossibility of PRF in $\mathsf{AC}^0[2]$ does *not* carry through for WPRF. Indeed, their proof involves querying the function on specific inputs (e.g., all those with a suffix of zeros), which are very unlikely to be sampled through a random selection of polynomially many inputs. However, the learning result of [17] has the same consequences even for WPRF, implying that WPRF cannot exist in $\mathsf{AC}^0$. This discrepancy between the lower bounds for PRF and WPRF begs the following question:

*Does $\mathsf{AC}^0[2]$ (or better yet, $\mathsf{AC}^0 \circ \mathsf{MOD}_2$) contain weak pseudorandom functions (WPRF)?*

This question was posed very recently by Akavia et al. [1], who started investigating it. They conjecture that the answer is yes, and provide a concrete function in $\mathsf{AC}^0 \circ \mathsf{MOD}_2$ that they conjecture is a WPRF. While they do not have a proof, they provide some evidence to its pseudorandomness by showing that it avoids specific attacks on PRFs in this class. They also study the class $\mathsf{AC}^0 \circ \mathsf{MOD}_2$ more generally, providing some conjectured properties of the class which they prove for some cases.

We propose to study this question, trying to prove a positive answer based on any standard assumption, or alternatively trying to prove lower bounds. One direction to start with is proving that the proposed candidate, or another one in $\mathsf{AC}^0 \circ \mathsf{MOD}_2$, is a WPRF under the LPN (learning with noise) assumption.[1] proved LPN may be necessary (under the conjecture mentioned below); proving that it is sufficient would be very interesting.

Another direction we will study, is trying to come up with a WPRF family in $\mathsf{AC}^0[2]$, trying to utilize the possibly stronger power of this class. In particular, $\mathsf{IP}_n \in \mathsf{AC}^0[2]$ is conjectured not to be in $\mathsf{AC}^0 \circ \mathsf{MOD}_2$ [26]. Moreover, [1] conjecture that all $\mathsf{AC}^0 \circ \mathsf{MOD}_2$ functions have a large Fourier coefficient (which is false for $\mathsf{IP}_n$). Thus, perhaps we can find a way to use $\mathsf{IP}_n$ in a construction of WPRF (perhaps by exploiting the fact that $\mathsf{IP}_n$ is self-reducible?).

A different approach is to start with an arbitrary WPRF in a relatively high complexity class (say $\mathsf{NC}^1$), and see if we can compile it into WPRF in $\mathsf{AC}^0[2]$. This is reminiscent of the *randomized encodings* approach, that allowed Applebaum, Ishai and Kushilevitz [2] to compile various cryptographic primitives (PRG, OWF, CRHF, PKE, Signatures, Commitment, ZKP) from a higher compelxity class to $\mathsf{NC}^0$. Of course, their notion of randomized encodings could not work for WPRF, because there cannot be WPRF in $\mathsf{NC}^0$. [2] discuss what fails trying to apply their approach to PRF, and while one of their arguments does not hold for WPRF, the main argument does: a randomized encoding needs *secret* and *fresh* randomness, while WPRF have either secret randomness that is reused (the key), or fresh randomness that is given to the adversary (the input). Nonetheless, we feel there is some hope in trying to generalize the notion, since we do not need the resulting function to be in $\mathsf{NC}^0$, but rather in $\mathsf{AC}^0[2]$. We will thus investigate the following question:

*Can we develop a generalized version of "randomized encodings" that could be applied to WPRF?*

The most general result would be a definition and construction of randomized encodings that can work (from/to) higher complexity classes. A more specific direction towards our goal, is finding a randomized encoding with the following properties. Given a WPRF $F(k,x) = f_k(x)$, define a randomized encoding $\hat{F}(k,x,r) \in \mathsf{NC}^0$ and then construct a WPRF of the form $h_{k,r}(x,s) = Ext(\hat{F}(k,x,r),s)$ where Ext is a strong computational extractor (with first argument being the source and the second argument being the seed). The idea here is that $\hat{F} \in \mathsf{NC}^0$ need not (and cannot) be a WPRF itself (so we relax the security requirement of randomized encodings), but should have enough computational entropy that can be extracted by a computational extractor. In order for this approach to possibly work, we need to carefully choose both the encoding and the extractor. For the encoding, $\hat{F}$ must preserve enough of computational entropy from $F$ (this is not the case in general for any encoding). For the extractor, its complexity cannot be too low, so that the resulting function does not fall in $\mathsf{AC}^0$; it also cannot be too high, as our goal is to achieve WPRF in as low complexity as possible.

A final approach we are planning to check, is trying to base a WPRF directly on Goldreich's [12] candidate OWF in $\mathsf{NC}^0$.

If the above directions do not seem successful, we will also try to work on proving impossibility of WPRF in $\mathsf{AC}^0 \circ \mathsf{MOD}_2$. One place to start is trying to extend the ideas behind the "natural proof" attacks of [25]. As mentioned above, their current proof applies to PRFs, and it is not hard to adapt it to apply also to *non-adaptive PRF (NAPRF)*, where the adversary must submit all queries non-adaptively. Using a result of [16], we have extended their proof to rule out NAPRF (and thus also PRF) not only in $\mathsf{AC}^0[p^k]$ for any prime $p$, but also to restricted $\mathsf{TC}^0$ with at most $n^{1+\alpha}$ wires.

Extending the results to WPRF is more challenging. We will explore whether we can infer additional properties of most common circuit lower bounds that allow us to rule out weak PRFs in low complexity classes such as ACC. What about lower bound proofs that establish stronger results, such as average-case hardness? This direction is challenging, as it is likely that new attacks on WPRF will give new learning results. Nonetheless, it is conceivable that this would not be the case, if the new attacks rely on a distinguisher with a large number of samples.

**Complexity of (W)PRF from PRG.** We next turn out attention to the question of constructing WPRF (or PRF), from generic assumptions, and in particular from PRG. We ask: *What is the minimal parallel complexity of WPRF constructions based on any PRG? Can we improve over GGM? Is there an* $\mathsf{NC}$ *reduction of WPRF to PRG?*

While we have constructions of PRF in $\mathsf{NC}$ (even in $\mathsf{TC}^0$) from concrete assumptions, the above questions (stated already in [22]) remain open, even for WPRF. Note that if we show an $\mathsf{NC}^i$ reduction of WPRF to PRG, we will obtain an $\mathsf{NC}^{i+1}$ reduction of PRF to PRG, and will prove that PRG in $\mathsf{NC}^1$ implies WPRF in $\mathsf{NC}^i$ and PRF in in $\mathsf{NC}^{i+1}$.

One natural approach towards constructing such a reduction, is to "parallelize" the GGM construction: the input $x$ (possibly after some processing) is divided to blocks of size up to $\log n$ bits, then the GGM construction is applied on each of the blocks, and the results are combined using some combining function $C$. The combining function $C$ should be chosen carefully, so as to avoid learning attacks. We have started exploring this approach as well as two others towards answering the above question.

We will also study a related question which may be easier: can we construct WPRF from trapdoor permutations (TDP)? In particular, we will explore constructions of TDP with parallel complexity for their *sampling* and *inverting* algorithms. Any such construction in $\mathsf{NC}$ will yield

a PRF in $\mathsf{NC}$, using a result shown by [22]. This will provide the first parallel construction of PRF from generic assumptions. We will study a (generalized) randomized encoding approach to achieving this.

A different aspect of the reduction complexity of PRF to PRG is one that deals with the *number* of applications, rather than the depth. We ask: *How many applications of PRG are necessary to construct WPRF? How many applications of WPRF are necessary to construct PRF?*

The GGM construction uses linearly many calls to the PRG, as well as a linear depth. While so far we discussed minimizing the latter, we now propose to minimize the former. We start with a basic form of this question: can we rule out WPRF from PRG with a *single application* of the PRG (and arbitrary non-cryptographic polynomial computations)? We will then study the more general question posed above, trying to find upper and lower bounds (through black-box separations) for the reduction complexity of WPRF to PRG, as well as of PRF to WPRF.

### 1.2 The Power of Negation for Pseudorandom Primitives

Let $f : \{0,1\}^n \to \{0,1\}$ be an arbitrary function. We define the *negation complexity* of $f$, denoted by $\mathsf{Neg}(f)$, to be the minimum number of negation gates in any fan-in two circuit with AND,OR,NOT gates computing $f$. In particular, $f$ is *monotone* if and only if $\mathsf{Neg}(f) = 0$. For a monotone function $f$, one can ask whether $f$ has a polynomial size *monotone circuit* (namely a circuit with no negation gates).

The study of monotone classes of functions and negation complexity has been prevalent in the complexity community as well as in computational learning theory, but little attention has been given to it in the cryptographic context (we will discuss two relevant exceptions [8, 14]). Recently, Goldreich and Izsak [14] have initiated a study of whether basic cryptographic primitives may be monotone. They study OWF and PRG, and show an inherent gap between the two in this respect, by proving: (1) if any OWF exist, then there exist OWF with polynomial-size monotone circuits, but (2) no monotone function can be a PRG. To quote from their paper: *these two results indicate that in the "monotone world" there is a fundamental gap between one-way functions and pseudorandom generators; thus, the "hardness-vs-randomness" paradigm fails in the monotone setting.*

We propose to expand their study in two directions: studying **stronger primitives** (WPRF and PRF), and studying the **negation complexity** (rather than just monotonicity) of these primitives. We believe that these directions are natural and interesting. For example, this study may enlighten us regarding the "randomness-vs-unpredictability" paradigm in the monotone world.

We start by recalling an important classic result regarding negation complexity of arbitrary boolean functions. Markov [20] proved (in 1958) that for any boolean function $f$ on $n$ variables, $\mathsf{Neg}(f) \leq \log(n+1)$.[2] Fischer [11] extends Markov's theorem to prove that this transformation from any circuit computing $f$ to one with only $\log(n+1)$ negations can be made efficient. Moreover, it can be shown [18] that these negations can be computed directly on the input $x$, independently of the function $f$. That is, there is a *fixed* $M_n : \{0,1\}^n \to \{0,1\}^{\log(n+1)}$ with $\log(n+1)$ negations, such that for any $f : \{0,1\}^n \to \{0,1\}$ that can be computed by a circuit of size $s$, there is a monotone circuit $A$ of size $\mathsf{poly}(s)$, satisfying $A(x, M_n(x)) = f(x)$ for all $x$. Next, we summarize specific questions we plan to pursue and some preliminary progress.

**Negation complexity of PRG.** Based on the above, we know that the negation complexity of

[2] In fact, Markov proved something more general, essentially proving that the negation complexity of a function is equal to its "alternating complexity", which we will not define here.

any PRG is more than 0, and at most logarithmic. We ask whether there exists a PRG $G$ such that $0 < \mathsf{Neg}(G) << \log n$? In particular, does there exist a PRG that can be computed by a circuit with a single negation? We will study these questions, possibly trying to leverage recent techniques from [5].

**Negation complexity of PRF.** It is easy to see that no monotone function can be a PRF. Indeed, the distinguisher may simply ask two queries $x \prec x'$ and check whether the outputs satisfy $f(x) \preceq f(x')$. For a monotone function the answer must always be yes, but for random functions there is 25% probability that the answer will be no. What can be done with more negations? In preliminary work, we[3] have succeeded to settle the negation complexity for any PRF, showing it is essentially tight with Markov's logarithmic upper bound. Let $\mathcal{F}_n$ denote the set of all Boolean functions on $n$ variables. We say that a function $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is $(\varepsilon, s)$-secure if for every circuit $C$ of size at most $s$,

$$\left| \Pr_{s\in\{0,1\}^n} \left[C^{F(s,.)} = 1\right] - \Pr_{f\in\mathcal{F}_n} \left[C^{f(.)} = 1\right] \right| \leq \varepsilon.$$

**Theorem** If $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is $(1/4, n^2)$-secure then any circuit computing $F(.,.)$ contains at least $\log n - O(1)$ negation gates.

This lower bound applies for both the adaptive case (PRF) and the non-adaptive one (NAPRF). The proof constructs a distinguisher (of size $O(n^2)$) that uses an alternating walk on the $n$-dimensional hypercube from $0^n$ to the middle layer (and applying Markov's theorem).

**Negation complexity of WPRF.** We start by noting that the distinguisher for PRF mentioned above (both the simple one for monotone functions and the one we constructed for logarithmically many negations) do not apply for WPRF. Thus, the negation complexity of WPRF remains open. The first natural question is whether a monotone function can be a WPRF. It is tempting to think that the PRG lower bound would immediately transfer over to the WPRF setting, since PRG can be constructed very simply from WPRF However, note that this transformation does not preserve monotonicity, even if we start with a family where for all $k$, $f_k$ is monotone, since the key $k$ is one of the inputs to $G$. Nonetheless, it seems that we can prove that WPRF cannot be monotone, as a corollary of results about learning classes of monotone functions, and the connection between WPRF and hardness of learning. In particular, Blum, Burch and Langford [6] show a weak learner for all monotone boolean functions, which can be shown to imply that for any family $\{f_k\}$ of WPRF, $f_k$ cannot be monotone except for a negligible fraction of the keys.

Towards proving a stronger lower bound on the negation complexity of WPRF, we will explore whether the weak learning algorithm of [6] can be extended to a weak learning algorithm for the class of functions computed with $t$ negations. If so, this would imply in particular that any WPRF will require more than $t$ negations for most values of the key.

In the other direction, we can hope to use hardness of learning results for monotone classes of functions, in order to prove upper bounds on the negation complexity of WPRF. The only work (to the best of our knowledge) that addresses this, is the result of Dachman-Soled et al. [8] (joint with Malkin). In that paper, they show under different cryptographic assumptions that there exists classes of monotone functions that are hard to learn with accuracy better than $1/2 + 1/\mathsf{poly}(n)$, essentially matching many weak learning upper bounds. These results can yield monotone functions with *some* hardness, but the parameters are not strong enough to obtain WPRF with cryptographic security. This is expected, as indeed there are no monotone WPRF.

[3] PI Malkin, together with her student Igor Carboni Oliveira, and with Ilan Orlov and Alon Rosen

We will try to extend their techniques for classes of functions with some $t << \log n$ negations. One component in their proof uses the result of Berkovitz [4] regarding the monotonicity of the slice function (on the middle layer of the hypercube). They apply the slice function on an arbitrary PRF to get a monotone function, and then amplify its hardness using a noise-sensitivity based approach to hardness amplification, following O'Donnell [23]. A possible direction is to extend Berkovitz result to apply to several layers close to the middle layer, when some negations are allowed. It seems that using this approach we will able to construct a WPRF with about $(\log n)/2$ negations. We then may also be able to use a better (non-monotone) combination function for the hardness amplification.

**Tradeoffs of Negations with Security or Efficiency.** So far, we focused on achieving pseudorandom primitives with minimal negation complexity. It is also interesting to study the tradeoffs between pseudorandmness and monotonicity. Specifically, if we restrict ourselves to circuits with $t$ negations (even when $t$ is less than the minimum required to achieve exponential security, e.g. $t = 0$), how much security (pseudorandmness) can we achieve? Can we prove a tight bound close to $1/2^t$ distinguishability?

On the other end of the spectrum, can we obtain more efficient constructions by using more negations than the minimum required? For example, can we achieve better parallel complexity if we use more than $\log(n + 1)$ negations for PRF? We will study this question, trying to obtain stronger lower bounds than those of Markov on the number of negations in bounded-depth circuits for PRFs. Such a result may explain the fact that practical candidates for PRF seem to use a large number of negations (or XORs); is this due to the fact that negations can provide speedup?

## 2 Addendum: New Results

Since the writing of the above almost two years ago, several relevant new results have been published, and I point out the most relevant ones.

Regarding **Circuit Complexity of PRF**, the recent work of Carmosino et al [7] provides a learning algorithm with membership queries for $\mathsf{AC}^0[p]$ (for any prime $p \geq 2$, in particular for $\mathsf{AC}^0[2]$). This gives a simpler and direct proof that there is no PRF in this class (although still leaves open the problems outlined in Section 1.1). I have also heard that there may be a new result coming up, constructing WPRF from Goldreich's candidate OWF [12] (one of the directions we mentioned in Section 1.1), but I have no further details (and in particular do not know what complexity class the suggested WPRF is in).

Regarding the **The Power of Negations in Cryptography**, a paper by that name [15] has studied the problems outlined in Section 1.2 above, providing lower bounds for various primitives (OWP, PRF, ECC, and others), and leaving several interesting open problems.

## References


[1] A. Akavia, A. Bogdanov, S. Guo, A. Kamath, and A. Rosen. "Candidate weak pseudorandom functions in ac0 ∘ mod2." In *ITCS* (2014).

[2] B. Applebaum, Y. Ishai, and E. Kushilevitz. "Cryptography in $\mathrm{NC}^0$." *SIAM J. Comput.* **36** (2006), 845–888.

[3] A. Banerjee, C. Peikert, and A. Rosen. "Pseudorandom functions and lattices." In *EUROCRYPT* (2012), 719–737.

[4] S. J. Berkowitz. "On some relationships between monotone and non-monotone circuit complexity." Technical Report (1982).

[5] E. Blais, C. Canonne, I. C. Oliveira, R. Servedio, and L. Tan. "Learning circuits with few negations." manuscript (2014).

[6] A. Blum, C. Burch, and J. Langford. "On learning monotone boolean functions." In *FOCS* (1998), 408–415.

[7] M. Carmosino, R. Impagliazzo, V. Kabanetz, and A. Kolokova. "Algorithms from natural lower bounds." *Electronic Colloquium on Computational Complexity (ECCC)* (2016). Available at `http://eccc.hpi-web.de/report/2016/008/`.

[8] D. Dachman-Soled, H. K. Lee, T. Malkin, R. A. Servedio, A. Wan, and H. Wee. "Optimal cryptographic hardness of learning monotone functions." *Theory of Computing* **5** (2009), 257–282.

[9] I. Damgård and J. B. Nielsen. "Expanding pseudorandom functions; or: From known-plaintext security to chosen-plaintext security." In *CRYPTO* (2002), 449–464.

[10] Y. Dodis, E. Kiltz, K. Pietrzak, and D. Wichs. "Message authentication, revisited." In *EUROCRYPT* (2012), 355–374.

[11] M. Fischer. "The complexity of negation-limited networks a brief survey." In *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 2023, 1975*, ed. H. Brakhage, Lecture Notes in Computer Science **33** (1975). 71–82.

[12] O. Goldreich. "Candidate one-way functions based on expander graphs." In *Studies in Complexity and Cryptography* (2011). 76–87.

[13] O. Goldreich, S. Goldwasser, and S. Micali. "How to construct random functions." *Journal of the Association for Computing Machinery* **33** (1986), 792–807.

[14] O. Goldreich and R. Izsak. "Monotone circuits: One-way functions versus pseudorandom generators." *Theory of Computing* **8** (2012), 231–238.

[15] S. Guo, T. Malkin, I. C. Oliveira, and A. Rosen. "The power of negations in cryptography." In *TCC* (2015).

[16] R. Impagliazzo, R. Paturi, and M. E. Saks. "Size-depth tradeoffs for threshold circuits." *SIAM J. Comput.* **26** (1997), 693–707.

[17] N. Linial, Y. Mansour, and N. Nisan. "Constant depth circuits, fourier transform, and learnability." *J. ACM* **40** (1993), 607–620.

[18] R. Lipton. "Re-gifting an old theorem." (2013). Available at `http://rjlipton.wordpress.com/2013/12/26/re-gifting-an-old-theorem/`.

[19] V. Lyubashevsky and D. Masny. "Man-in-the-middle secure authentication schemes from lpn and weak prfs." In *CRYPTO (2)* (2013), 308–325.

[20] A. A. Markov. "On the inversion complexity of a system of functions." *J. ACM* **5** (1958), 331–334.

[21] U. M. Maurer and J. Sjödin. "A fast and key-efficient reduction of chosen-ciphertext to known-plaintext security." In *EUROCRYPT* (2007), 498–516.

[22] M. Naor and O. Reingold. "Synthesizers and their application to the parallel construction of pseudo-random functions." *J. Comput. Syst. Sci.* **58** (1999), 336–375.

[23] R. O'Donnell. "Hardness amplification within $\mathrm{np}$." *J. Comput. Syst. Sci.* **69** (2004), 68–94.

[24] K. Pietrzak. "A leakage-resilient mode of operation." In *EUROCRYPT* (2009), 462–482.

[25] A. A. Razborov and S. Rudich. "Natural proofs." *J. Comput. Syst. Sci.* **55** (1997), 24–35.

[26] R. A. Servedio and E. Viola. "On a special case of rigidity." *Electronic Colloquium on Computational Complexity (ECCC)* **19** (2012), 144.