On the Common Bias of Majorities: Poly-Time Attacks on THR-XOR PRGs

Antonio Giulio D'Antona
Università di Pisa

Pierrick Méaux
University of Luxembourg
Pierrick.Meaux@uni.lu

Akin Ünal
Institute of Science and Technology Austria
Akin.Uenal@ist.ac.at

October 21, 2026

Abstract

Pseudorandom Generators (PRGs) based on Threshold-XOR predicates with large locality and high stretches have recently gained traction, since they lend themselves as shallow weak Pseudorandom Functions (PRFs) to fast multiparty computation protocols. In this work, we present novel fast attacks on such PRGs that achieve substantial advantages. Concretely, we break the security levels of most parameters collected by Boura, Couteau, Perrin and Rotella (ToSC'25), as well as those proposed by Fu, Li, Lyu and Liu (EC'26). On the asymptotic side, we prove that our attacks achieve an advantage of $n^{-n/a}$ where n denotes the seed length and a the locality of threshold predicates. As a consequence, when $a \in \Theta(n)$, we get poly-time attacks with noticeable advantage. These results close current gaps on the theoretical study of THR-XOR based Goldreich PRGs.

Additionally, we demonstrate how the attack advantage can be amplified for PRGs with large output lengths (which are common in the context of weak PRFs). Finally, we discuss alternative predicates for Goldreich PRGs that resist all currently known attacks.

1 Introduction

Pseudorandom Generators (PRGs) play a key role in the world of provable cryptography. They are equivalent to one-way functions [HILL99] and imply all of Minicrypt [Imp95]. A PRG is simply a deterministic function that stretches a uniformly random bit string to a longer pseudorandom bit string.

In praxis and in theory, trusted PRG candidates that are expected to achieve exponential security levels do exist; implied either by symmetric constructions or assumed hardness of mathematical problems, like LPN, LWE, DDH and many more. However, given the ubiquity of PRGs, it is of theoretical and practical interest to reduce the (circuit-)complexity of evaluating a PRG. Indeed, in praxis, lightweight PRG candidates are optimal for low-power devices, while in theory, low-complexity PRGs allow for derandomization of various complexity classes [NW88, Wil13].

In a *local PRG*, each output bit depends only on a constant number of input bits; equivalently, the evaluation complexity per output bit is constant. Local PRGs are of high interest for a multitude of reasons: In praxis, they can be evaluated in constant time if one allows for parallelization. In theory, they are a gateway to cryptography in NC_0 [AIK04]. In both worlds, local PRGs can be cheaply evaluated in private Multi-Party Computation (MPC) protocols and Fully Homomorphic Encryption (FHE) schemes, as they only need a constant number of multiplication gates for evaluation per output bit. On top of that, they can be evaluated by constant-degree polynomials over any field or the ring of integers. This is particularly important for their application in the construction of indistinguishability obfuscation [JLS21, JLS22]. (Though, it shall be mentioned that for the construction of iO, local *structured-seed* PRGs are sufficient [RVV24, ABCM25]. Structured-seed PRGs relax the security notion of PRGs by allowing the seed to come from any distribution that may depend on the description of the concrete PRG instance.)

The first local PRGs have been put forth by Goldreich [Gol11] who proposed PRGs based on Boolean functions $P : \{0,1\}^a \to \{0,1\}$ of constant arity $a \in O(1)$ (usually called *predicates*). To compute the output bits of such a PRG $F : \{0,1\}^n \to \{0,1\}^m$ one applies P to a substring of the seed $x \in \{0,1\}^n$

where the substring varies for each output bit. Usually, one samples uniformly at random a non-repeating list of a elements of $\{1,\ldots,n\}$ per output bit. Those lists constitute a description of a *Goldreich PRG*.

The security of Goldreich PRGs has been well researched in the theoretic cryptography literature [App13]. A popular choice for the predicate P is the combination

$$\mathsf{MAJ}_a(x_{i_1},\ldots,x_{i_a}) \oplus \mathsf{XOR}_b(x_{i_{a+1}},\ldots,x_{i_{a+b}})$$

of a majority predicate (that outputs the majority of its input bits) and a XOR predicate (that outputs the parity of its input bits). To achieve Goldreich PRGs based on $\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$ with polynomial stretches $\{0,1\}^n \to \{0,1\}^{n^{1+e}}$ one needs to set the arities a and b sufficiently high. A high value of b increases the resilience of the predicate and avoids correlation-based attacks [Sie84, App13, ABW10, MST03, BQ09]. A high value of a increases the algebraic immunity of the predicate and avoids algebraic attacks [CM03, AL16, Üna23b, Üna23a]. Indeed, one reason for the popularity of $\mathsf{MAJ} \oplus \mathsf{XOR}$-based PRGs is the fact that MAJ predicates enjoy optimal algebraic immunity, while threshold predicates (which output 1 if the number of ones in their input is at least a fixed threshold) with thresholds close to half of their size enjoy at least high algebraic immunities [DMS06, CM22].

However, note that all constant-degree PRGs (hence, also local PRGs) $f : \{0,1\}^n \to \{0,1\}^{n^{1+e}}$ have a common weakness: their maximum attainable stretch is a priori limited. Concretely, the exponent $1+e$ must be smaller than the (rational) degree of f if one wants to avoid poly-time attacks [AL16]. And even for lower values $0 < 1+e < \deg(f)$, subexponential time attacks are inevitable [Üna23b, Üna23a], as it is known that the quality of algebraic distinguishers degrade gracefully with the stretch exponent $1+e$. These bounds are inherent for all PRGs of constant degree, in particular for local PRGs based on $\mathsf{MAJ} \oplus \mathsf{XOR}$, and motivate the following question:

> *Can we stretch the output of* $\mathsf{MAJ} \oplus \mathsf{XOR}$ *PRGs to superpolynomial lengths by raising the locality of* MAJ *to non-constant arity while maintaining exponential levels of security?*

Non-Local PRGs. Before we answer the above question, let us explain why *non-local* PRGs based on $\mathsf{MAJ} \oplus \mathsf{XOR}$ with huge stretches are interesting in theory and praxis. As pointed out in [BCPR25], PRGs based on $\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$ with $a \in \omega(1)$ belong to the class of *shallow weak pseudorandom functions* (PRFs)[1]. Such weak PRFs have a very low evaluation complexity and enjoy applications in oblivious PRFs, pseudorandom correlation functions for 2-party protocols [BCM+24, CDD+24] and zero-knowledge proofs [ABBS25]. Additionally, like their local PRG counterparts, shallow weak PRFs are very MPC- and FHE-friendly [CGGI20]. In particular, shallow weak PRFs based on $\mathsf{THR} \oplus \mathsf{XOR}$ predicates bear a striking resemblance to the FLIP and FiLIP ciphers [MJSC16, MCJS19, GGM24], which were explicitly optimized for homomorphic encryption.

Contribution

In this work, we answer the above question negatively for almost all choices of non-constant a. Thus making huge theoretical and practical progress on the security of non-local PRGs and shallow weak PRFs based on $\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$.

Additionally, we introduce a novel metric of practical impact, the *common bias*, for predicates. As the advantage of the attacks given in this work (up to constant factors) equals the common bias, it is inevitable for a *good* predicate to have a low common bias.

Asymptotic Results. We prove that the common bias (and therefore the advantage of our poly-time attacks) of MAJ_a increases gracefully for asymptotically increasing $a(n)$.

Theorem 1. *Let* $F : \{0,1\}^n \to \{0,1\}^{\frac{1.1}{b}\cdot n\log(n)}$ *be a PRG based on* $\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$. *There is an inversion attack on* F *with time complexity* $O(\log(n)n^3)$ *and advantage* $n^{-O(n/a)}$.

[1] A weak PRF is a keyed function $f_k(z)$ that guarantees the pseudorandomness of evaluations $f_k(z_1),\ldots,f_k(z_{\mathsf{poly}(n)})$ even when given the uniformly random inputs $z_1,\ldots,z_{\mathsf{poly}(n)}$. A PRG based on some predicate P can be interpreted as a weak PRF by treating the seed $x \leftarrow \{0,1\}^n$ as the key of the PRF and a random input $z \in \{1,\ldots,n\}^a$ as a list of indices for the predicate $P : \{0,1\}^a \to \{0,1\}$ to be evaluated on. This means, we set $f_x(z) := P(x_{z_1},\ldots,x_{z_a})$.

For huge majority predicates of size $a \in \Theta(n)$, Theorem 1 implies a noticeable advantage of $n^{-O(1)}$, hence implying poly-time attack with $1/\operatorname{poly}(n)$ advantage, while only requiring a logarithmic multiplicative stretch.

While traditional results imply a rising security of local PRGs based on $\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$ for increasing a, our findings turn the story around and show that—to achieve exponential levels of security even for logarithmic stretches—it is mandatory to have small values for a. An interesting gap is left by the intermediate area of $a \in \omega(1) \cap O(\log(n))$. We leave it as an open question to clarify if one can hope for $\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$-based PRGs in this area with exponential levels of security.

Concrete Results. For practitioners, we consider PRGs based on the more general predicate $\mathsf{THR}_{a,t} \oplus \mathsf{XOR}_b$. I.e., we replace the MAJ_a predicate by the more general function $\mathsf{THR}_{a,t} : \{0,1\}^a \to \{0,1\}$ that accepts if its input has a Hamming weight of at least t.

Theorem 2. *Let $F : \{0,1\}^n \to \{0,1\}^{m'}$ be a PRG based on $\mathsf{THR}_{a,t} \oplus \mathsf{XOR}_b$. Denote by $\mu_{m',n,b}$ the probability[2] that a binary $m' \times n$ matrix with random rows of Hamming weight b has rank at least $n-1$.*

There is an inversion attack on F with time complexity $O(m'n^2)$ and advantage

$$p(m', n, a, t) \cdot \mu_{m',n,b} \cdot 0.5.$$

We demonstrate the performance of our attacks in Table 1 on the 4 parameter sets reported in [BCM+24, BCPR25]. Note that those parameters all aimed for 128 bits of security. Our attacks show that the first 3 parameter sets achieve significantly fewer than 128 bits of security.

n	m	m'	T	adv	$\log 2(T/\text{adv})$
256	$2^{40.00}$	256	$2^{24.00}$	$2^{-17.94}$	41.94
512	$2^{40.50}$	512	$2^{27.00}$	$2^{-35.17}$	62.17
894	$2^{33.33}$	894	$2^{29.41}$	$2^{-60.81}$	90.22
2048	$2^{33.00}$	2048	$2^{33.00}$	$2^{-138.21}$	171.21

Table 1: Performances of Algorithm 2 on the parameters of [BCM+24, CDD+24, BCPR25] for inverting a PRG $\mathcal{F} = \mathsf{THR}_{64,32} \oplus \mathsf{XOR}_{10} : \{0,1\}^n \to \{0,1\}^m$. T denotes the time complexity and adv the advantage of Algorithm 2. Note that Algorithm 2 only reads the first m' output bits of F and discards the rest. For details on T, adv, we refer to Remark 4.

Common Bias. To prove our results, we introduce the concept of the *common bias*, which generalizes the notion of bias and measures the correlation of multiple instances of a predicate. While, on a theoretical level, it measures how well an ensemble of predicates can be approximated by a constant function and can be expressed in terms of min-entropy, we cannot overstate its impact on the concrete security of $\mathsf{THR} \oplus \mathsf{XOR}$ PRGs. Having a low common bias is mandatory for predicates leading to PRGs, making the common bias a new metric for estimating the security of lightweight PRG candidates (confer Remark 1) and ranking it alongside *algebraic immunity* and *resiliency* as an additional important metric, with each capturing a different, complementary type of attack on predicate-based PRGs.

Alternative Predicates. In Section 5, we introduce novel predicates and parameters to achieve 128 bits of security for Goldreich PRGs with high stretch. We present two alternatives to MAJ predicates: The first interpolates between MAJ and XOR predicates, while the output of the second only depends on the residue class modulo 3 of the Hamming-weight of its input, effectively harnessing the difficulty of analyzing alternating-moduli weak PRFs [BIP+18, DGH+21, APRR24] in the PRG setting. Both predicates inhibit a simple structure by being symmetric, allowing for their usage in existing protocols, and avoid the attacks of this work by having a low common bias. As a completely novel result, we bound the algebraic immunity of such $\mathsf{MOD3}$ predicates by their arity over 3.

Technical Overview

To explain our results, we will begin with a toy attack given in Algorithm 1.

[2]The additional factor $\mu_{m',n,b}$ arises from the fact that a sparse random matrix need not have full rank with high probability. As we explain in Remark 2, for most concrete parameters considered in this work, one has $\mu_{m',n,b} \geq 0.5$ for $m' = n$.

Algorithm 1: Toy Inversion Algorithm for $\mathsf{MAJ} \oplus \mathsf{XOR}$-based PRGs.

Data: A PRG $\mathcal{F} : \{0,1\}^n \to \{0,1\}^n$ based on $\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$.
Some output $y = \mathcal{F}(x) \in \{0,1\}^n$.
Result: Input string $x \in \{0,1\}^n$.

1 Parse $\mathcal{M}(X) \oplus \mathcal{L}(X) := \mathcal{F}(X)$ where $\mathcal{M} = (\mathsf{MAJ}_{A_1}(X), \ldots, \mathsf{MAJ}_{A_n}(X))$ and $\mathcal{L} = (\mathsf{XOR}_{B_1}(X), \ldots, \mathsf{XOR}_{B_n}(X))$;
2 Compute $x' \in \{0,1\}^n$ such that $\mathcal{L}(x') = y \oplus (1, \ldots, 1)$;
3 Return x' ;

The attack assumes a PRG $\mathcal{F} : \{0,1\}^n \to \{0,1\}^n$ without requiring any stretch. For each output bit $j \in [n]$, it decomposes the PRG into the XOR-sum of two predicates MAJ_{A_j} and XOR_{B_j} computing this output bit. Remember that

$$\mathsf{MAJ}_{A_j}(x) := \begin{cases} 1, & \text{if } \#\{i \in A_j \mid x_i = 1\} \geq a/2, \\ 0, & \text{otherwise,} \end{cases}$$

and $\mathsf{XOR}_{B_j}(x) := \bigoplus_{i \in B_j} x_i$. Simply put, it decomposes the PRG into high-degree and linear terms. It replaces all high-degree terms $\mathsf{MAJ}_{A_1}(x), \ldots, \mathsf{MAJ}_{A_n}(x)$ by the constant 1, and solves the resulting linear equation system

$$1 \oplus \bigoplus_{i \in B_j} x_i = y_i, \qquad\qquad j \in [n].$$

For the ease of exposition, we make the benevolent assumption that this sparse linear equation system always has full rank and a unique solution and postpone a detailed discussion on this to Remark 2. With this assumption, the success probability of Algorithm 1 equals the probability of all majority predicates $\mathsf{MAJ}_{A_1}(x), \ldots, \mathsf{MAJ}_{A_n}(x)$ accepting a uniformly random bit string $x \leftarrow \{0,1\}^n$. Now, each $\mathsf{MAJ}_{A_i}(x)$ *individually* is unbiased, i.e.

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{MAJ}_{A_i}(x) = 1] = \frac{1}{2}$$

if a is odd. (For even a, we incur a small bias towards 1, however, we ignore this for now.) It is a typical heuristic at this point to assume that the probabilities of two different predicates MAJ_{A_i}, MAJ_{A_k} accepting x are independent, leading to a success probability of 2^{-n} for our attack. However, while this heuristic might be appropriate for small values of a, it becomes increasingly unlikely for growing sizes of majority predicates. To formalize this, let us define the *common bias*[3] of the predicates $\mathsf{MAJ}_{A_1}, \ldots, \mathsf{MAJ}_{A_n}$ by

$$\mathsf{c\text{-}bias}(\mathsf{MAJ}_{A_1}, \ldots, \mathsf{MAJ}_{A_n}) := \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{MAJ}_{A_1}(x) = \cdots = \mathsf{MAJ}_{A_n}(x) = 1].$$

To compute this quantity, we assume that all $A_1, \ldots, A_n \leftarrow \binom{[n]}{a}$ are uniformly and independently random size-a subsets of $\{1, \ldots, n\}$. Let us first consider the probability that one MAJ_{A_i} accepts some fixed bit string $x \in \{0,1\}^n$. Since A_i is random, this probability only depends on the Hamming-weight $\mathsf{w_H}(x)$ of x. Hence, we set

$$q(n,h,a) = \Pr_{A \leftarrow \binom{[n]}{a}}\left[\mathsf{MAJ}_A(1^h 0^{n-h}) = 1\right].$$

The random variable $\#(A \cap \{1, \ldots, h\})$ is distributed according to the *hypergeometric distribution*, whose probability mass function is given by

$$\Pr_A[\#(A \cap [h]) = t] = \frac{\binom{h}{t} \cdot \binom{n-h}{a-t}}{\binom{n}{a}}.$$

[3]For general fixed functions $f : \{0,1\}^n \to \{0,1\}^m$, we define the common bias to be $\mathsf{c\text{-}bias}(f) := \max_y \Pr_x[f(x) = y]$. More detailed definitions for distributions of functions are given in Definition 6.

n	$\mathsf{THR}_{2,2}$	MAJ_3	MAJ_5	MAJ_7	$\mathsf{THR}_{64,32}$	$\mathsf{MAJ}_{\mathrm{odd}(\sqrt{n})}$	$\mathsf{MAJ}_{\mathrm{odd}(n/4)}$
80	$2^{-67.10}$	$2^{-34.40}$	$2^{-27.47}$	$2^{-23.10}$	$2^{-2.66}$	$2^{-20.02}$	$2^{-11.20}$
128	$2^{-107.18}$	$2^{-55.05}$	$2^{-44.08}$	$2^{-37.20}$	$2^{-6.20}$	$2^{-25.95}$	$2^{-13.16}$
256	$2^{-214.05}$	$2^{-110.11}$	$2^{-88.38}$	$2^{-74.81}$	$2^{-14.94}$	$2^{-44.55}$	$2^{-16.10}$
512	$2^{-427.81}$	$2^{-220.22}$	$2^{-176.96}$	$2^{-150.00}$	$2^{-32.17}$	$2^{-74.09}$	$2^{-19.05}$
1024	$2^{-855.32}$	$2^{-440.45}$	$2^{-354.13}$	$2^{-300.39}$	$2^{-66.53}$	$2^{-117.17}$	$2^{-21.99}$
2048	$2^{-1710.33}$	$2^{-880.91}$	$2^{-708.47}$	$2^{-601.16}$	$2^{-135.21}$	$2^{-184.18}$	$2^{-24.91}$

Table 2: Each entry denotes the common bias of $\mathsf{THR}_{A_1,t},\ldots,\mathsf{THR}_{A_n,t}$ resp. $\mathsf{MAJ}_{A_1},\ldots,\mathsf{MAJ}_{A_n}$ for uniformly random $A_1,\ldots,A_n \leftarrow \binom{[n]}{a}$. $\mathrm{odd}(r)$ rounds a number r up to the next odd integer.

For simplicity, we assume that a is odd. I.e., MAJ_A only accepts 1^h0^{n-h} if A contains at least $\frac{a+1}{2}$ indices from $[h]$. Hence, we have

$$q(n,h,a) = \sum_{t=(a+1)/2}^{a} \Pr_A\left[\#(A \cap [h]) = t\right] = \sum_{t=(a+1)/2}^{a} \frac{\binom{h}{t}\cdot\binom{n-h}{a-t}}{\binom{n}{a}}.$$

The probabilities of two different majority predicates accepting a random bit string $x \leftarrow \{0,1\}^n$ are not[4] independent. However, their probabilities only depend on the Hamming weight of x. If we fix x to be some bit string with Hamming weight h, their probabilities become independent again. Hence, the probability of all majority predicates $\mathsf{MAJ}_{A_1},\ldots,\mathsf{MAJ}_{A_n}$ accepting $x = 1^h0^{n-j}$ is exactly $q(n,h,a)^n$. If we partition our event space according to $\mathsf{w_H}(x)$, we get

$$\begin{aligned}\textsf{c-bias}(\mathsf{MAJ}_{A_1},\ldots,\mathsf{MAJ}_{A_n}) &= \sum_{h=0}^{n} \Pr_x\left[\mathsf{w_H}(x)=h\right]\cdot q(n,h,a)^n \\ &= \sum_{h=0}^{n} 2^{-n}\cdot\binom{n}{h}\cdot q(n,h,a)^n.\end{aligned}$$

By substituting the sum formula for $q(n,h,a)$, we obtain an exact formula for the common bias and the advantage of Algorithm 1. This formula may be featureless on first sight, however, we can use it to estimate the bit security induced by different threshold predicates in $\mathsf{THR}\oplus\mathsf{XOR}$ PRGs (cf. Table 2).

Remark 1 (Concrete Impact of Common Bias). For concrete parameters, we compute the runtime of the attacks of this work by n^3 and their advantage by $\textsf{c-bias}_{\mathrm{ac}}(\mathsf{THR}_{a,t})/8$, leading to a bit security of $3\log(n)+3-\log(\textsf{c-bias}_{\mathrm{ac}}(\mathsf{THR}_{a,t}))$ in Tables 1, 3 and 4. One should, however, not be misled by the term $3\log(n)$, as this stems from Gaussian elimination and can be improved to $2\log(n)+\log(b)$ by Wiedemann's algorithm [Wie86] or to $\log(n)$ by preprocessing linear tests for fixed XOR predicates.

We therefore advise to always choose parameters such that $\textsf{c-bias}_{\mathrm{ac}}(\mathsf{THR}_{a,t})$ is less than 2^{-128}, as this is the one bottleneck of the attacks presented here.

On Asymptotic Bounds. From a theoretic point of view, the formula for the common bias of MAJ_a is unsatisfying as it does not directly tell us its asymptotic growth. In the theoretic part of this work, we prove that

$$\textsf{c-bias}(\mathsf{MAJ}_{A_1},\ldots,\mathsf{MAJ}_{A_n}) = \sum_{h=0}^{n} \Pr_x\left[\mathsf{w_H}(x)=h\right]\cdot q(n,h,a)^n$$

is lower bounded by $n^{-O(n/a)}$. This means, that for $a \in \Theta(n)$, we get a poly-time attack with noticeable advantage $n^{-O(1)}$. Hence, we also completely break PRGs with large MAJ-predicates in theory.

To prove our asymptotic bound, we identify a critical Hamming weight threshold h for seeds, that is slightly over $n/2$. Specifically, h is so large that $q(n,h,a)$ is very high, but it is also low enough such that the probability of a random bit string having Hamming weight $\geq h$ is noticeable.

[4] As counterexample, consider the case $n=a$. If $\mathsf{MAJ}_{A_1}(x)=1$, it is impossible to have $\mathsf{MAJ}_{A_2}(x)=0$.

Remark 2 (On the Rank of Random Sparse Matrices). Note that Algorithm 1 may fail if the linear equation system $\mathcal{L}(X) = y \oplus (1, \ldots, 1)$ does not have full rank over $\mathbb{Z}_2$. Now, if $\mathcal{L}$ is a uniformly random $\mathbb{Z}_2$-linear map it is known that the probability of it being regular is given by $\prod_{k=1}^{n}(1 - 2^{-k})$ which converges towards $0.289\ldots$ [Slo26]. However, the linear map $\mathcal{L} = (\mathsf{XOR}_{B_1}, \ldots, \mathsf{XOR}_{B_n})$, corresponds to a random sparse matrix where every row has Hamming weight b. Hence, the probability of $\mathcal{L}$ being invertible depends on b besides n. It is known [CFP19] that the probability of $\mathcal{L}$ having rank $n-1$ or more converges towards zero if b is constant. To guarantee a high probability of $\mathcal{L}$ having rank $\geq n-1$, one needs to take at least $\Omega(n \log(n))$ output predicates into account.

To deal with this rank deficiency problem, we need to adapt Algorithm 1 slightly to Algorithm 2. For our asymptotic results, we require a logarithmic stretch of $c \cdot n \log(n)$ with $c > 1/b$. This allows to mathematically rigorously prove their correctness.

For our concrete results, however, we bound the probability $\mu_{n,n,b}$ of $\mathcal{L}$ having rank at least $n-1$ for uniformly random XOR_b predicates. While there exists no closed formula for $\mu_{n,n,b}$, we could verify experimentally that $\mu_{n,n,b} \geq 50\%$ for almost all parameters listed in this work.[5] Therefore, for our concrete results, we make the very cautious assumption that $\mathcal{L}$ has rank $n-1$ with probability at least 25%. Since an equation system of rank $n-1$ has at most 2 solutions, Algorithm 2 guesses randomly one of them. Therefore, the advantage of our actual attack Algorithm 2 is estimated by $\mathsf{c\text{-}bias}_{\mathrm{ac}}(\mathsf{THR})/8$. In other words, the security bits reported in Tables 1, 3, 4 and 10 are by a tolerance margin of 3 bits larger than the logarithm of runtime over avantage.

Remark 3 (Single-Instance vs. Multi-Instance). All our attacks have a low (polynomial) time complexity and a low (noticeable up to subexponential) advantage. One might ask if it is possible to trade-off runtime for success probability and increase both by a fixed factor to get an attack with high advantage and polynomial up to subexponential runtime.

In theory, where we consider only isolated single-instance games, this is not possible, as the advantage of our attacks depends on the seed $x \leftarrow \{0,1\}^n$ having a high (or low) Hamming-weight. As the adversary cannot ask the PRG-challenger to re-roll the seed, it is not possible to repeat the attacks such that their success probability increases.

However, in praxis, it is more realistic to assume that an adversary has access to multiple instances of the PRG-security game. Indeed, if a $\mathsf{MAJ} \oplus \mathsf{XOR}$-based PRG is used in a commercially successful 2PC protocol, we can expect an attacker to see millions or billions new instances of this protocol initiated every day. In such an environment, an attacker can hop from user to user until its attack succeeds, effectively trading off runtime for success probability.

On Seed Hardening. The attentive reader may have noticed that the advantage of our attacks and the common bias of threshold predicates is dominated by the probability of seeds $x \leftarrow \{0,1\}^n$ having a relatively high or low Hamming weight. It is therefore natural to ask, if hardening seeds somehow (for example by requiring them all to have Hamming weight exactly $n/2$) avoids our attacks completely. While for a lot of applications, seed hardening is not an option as the seed is output by another algorithm, in the context of weak PRFs it might be indeed thinkable to restrict keys to those of balanced Hamming weight. We discuss this countermeasure in Section D and demonstrate that for PRGs with huge output stretches it does, in fact, *not* prevent our attack. Concretely, we will explain how one can strategically select threshold predicates such that their common bias is again $n^{-O(n/a)}$, even if $x \leftarrow \{0,1\}^n$ is conditioned on $\mathsf{w_H}(x) = n/2$.

On Selecting Predicates. In Section E, we give an elementary algorithm for heuristically selecting threshold predicates with maximum correlation, therefore amplifying their common bias and leading to better attacks. While we cannot rigorously bound the improvement in advantage by selecting specific predicates, we will report attack numbers under mild statistical assumptions.

On Voting. We note that the common bias of majorities (and, more generally, the hypergeometric distribution) capture more general phenomena and have meaning beyond cryptanalysis. We illustrate an application in the context of social voting in Section F.

[5] With the only exception $n = 263$, $b = 5$, where we approximated $\mu_{263,263,5} \approx 29.3\%$.

n	m	a	b	c-bias($\mathsf{MAJ}_{A_1}, \ldots, \mathsf{MAJ}_{A_n}$)	FLLL26 [FLLL26]	Algorithm 2
42	42	37	5	$2^{-2.06}$	$2^{51.18}$	$\mathbf{2^{21.24}}$
80	80	37	5	$2^{-6.66}$	$2^{93.97}$	$\mathbf{2^{28.63}}$
128	128	37	5	$2^{-11.92}$	$2^{143.00}$	$\mathbf{2^{35.92}}$
54	146	47	7	$2^{-2.23}$	$2^{52.26}$	$\mathbf{2^{22.49}}$
80	239	47	7	$2^{-5.00}$	$2^{84.97}$	$\mathbf{2^{26.97}}$
128	430	47	7	$2^{-9.54}$	$2^{135.00}$	$\mathbf{2^{33.54}}$
65	524	57	8	$2^{-2.25}$	$2^{61.07}$	$\mathbf{2^{23.31}}$
80	715	57	8	$2^{-3.73}$	$2^{65.97}$	$\mathbf{2^{25.70}}$
128	1448	57	8	$2^{-7.79}$	$2^{110.00}$	$\mathbf{2^{31.79}}$
85	7225	75	10	$2^{-2.29}$	$2^{61.23}$	$\mathbf{2^{24.52}}$
128	16384	75	10	$2^{-5.55}$	$2^{82.00}$	$\mathbf{2^{29.55}}$
256	65536	75	10	$2^{-14.09}$	$2^{181.00}$	$\mathbf{2^{41.09}}$

Table 3: Performance of Algorithm 2 at inverting a PRG $\mathcal{F} = \mathsf{MAJ}_a \oplus \mathsf{XOR}_b : \{0,1\}^n \to \{0,1\}^m$. For the concrete parameters here, Algorithm 2 only considers the first $m' = n$ output bits of $\mathcal{F}$ and discards the rest. We compare here the performance of Algorithm 2 with the best reported performances of inversion algorithms of [FLLL26].

Related Work

The attacks in this work can be summarized as linear approximation attacks that substitute high-degree terms by constants and solve the resulting linear equation system. This attack paradigm is not new. It is, for example, already employed in *Guess-and-Solve* attacks (aka *Guess-and-Determine* or *Guess-and-Decode*) [CDM+18, YGJL22], which mainly focused on the 5-local predicate $\mathsf{THR}_{2,2} \oplus \mathsf{XOR}_3 = X_1 \cdot X_2 + X_3 + X_4 + X_5$. Other examples are given by the linear distinguishers of Applebaum and Lovett [AL16] on local PRGs of low algebraic immunity.

Fu, Li, Lyu and Liu [FLLL26] have been the first to attack PRGs based on $\mathsf{THR} \oplus \mathsf{XOR}$-predicates with large localities. Their work improved significantly the status quo for concrete parameters. Their main strategy is to group the threshold predicates of the PRG according to small common intersections, therefore increasing the correlation in each group. For each group, it is assumed that all bits in the common intersection are identical. By doing so, the THR-predicates become significantly biased and are replaced by independent noise values. This allows for treating $\mathsf{THR} \oplus \mathsf{XOR}$ equalities per group as noisy linear equation systems and using a Gauss-based solving algorithm for Syndrome Decoding to extract a seed. Since [FLLL26] and this work attack the same PRGs, we will explicitly point out differences of both attack strategies in the following.

The advantages of the attacks of [FLLL26] and this work behave significantly different: [FLLL26] reports for their attacks always a high, but heuristic, advantage, while the advantage of Algorithm 2 degrades with decreasing locality a. Also, for estimating their advantage, [FLLL26] assumes the worst possible Hamming weight $n/2$ of the seed $x \in \{0,1\}^n$, while Algorithm 2 makes explicit usage of seeds with non-average Hamming weights.

Now, Algorithm 2 offers several improvements over the attacks of [FLLL26]: it is conceptually more elegant and allows for a mathematically rigorous analysis. Also, it always runs in polynomial time and requires no stretch of the PRG for concrete parameters and only a small logarithmic stretch for asymptotic ones. In comparison, [FLLL26] reports an exponential time complexity of $O(n^3 \cdot 2^{n - n \cdot \log(1 + \frac{2s}{\sqrt{2\pi(a-s)}})})$ and requires a polynomial stretch of $m = n^s$. Even concretely, Algorithm 2 significantly outperforms [FLLL26] by 30–140 bits, as demonstrated in Table 3. Additionally, Algorithm 2 takes explicitly the rank deficiency of sparse random matrices into account, while [FLLL26] implicitly assumes all occurring sparse equation systems to be of full rank.

The differences in time complexities are inherent, as the algorithms of [FLLL26] are based on grouping and solving noisy equation systems. As Syndrome Decoding is exponentially hard, this necessitates an exponential number of Gaussian eliminations. In comparison, Algorithm 2 cuts down the number of

n	m	a	b	c-bias($\mathsf{MAJ}_{A_1}, \ldots, \mathsf{MAJ}_{A_n}$)	Algorithm 2	targeted bit-sec
85	85	37	5	$2^{-10.22}$	$2^{29.44}$	80
133	133	37	5	$2^{-15.46}$	$2^{36.63}$	128
263	263	37	5	$2^{-29.48}$	$2^{53.60}$	256
91	281	47	7	$2^{-9.06}$	$2^{28.59}$	80
141	485	47	7	$2^{-13.74}$	$2^{35.16}$	128
285	1170	47	7	$2^{-26.92}$	$2^{51.38}$	256
105	1075	57	8	$2^{-8.89}$	$2^{29.03}$	80
155	1929	57	8	$2^{-12.99}$	$2^{34.82}$	128
303	5274	57	8	$2^{-24.83}$	$2^{49.55}$	256
146	21316	75	10	$2^{-9.79}$	$2^{31.36}$	80
205	42025	75	10	$2^{-13.73}$	$2^{36.77}$	128
365	133225	75	10	$2^{-24.22}$	$2^{49.76}$	256

Table 4: Performance of Algorithm 2 on challenge parameters proposed by [FLLL26] at inverting a PRG $\mathcal{F} = \mathsf{MAJ}_a \oplus \mathsf{XOR}_b : \{0,1\}^n \to \{0,1\}^m$. For the concrete parameters here, Algorithm 2 only considers the first $m' = n$ output bits of $\mathcal{F}$ and discards the rest. Note that the targeted security levels of [FLLL26] are clearly undercut by Algorithm 2.

Gaussian eliminations to one by exploiting the common bias of threshold predicates. This trick is so impactful that Algorithm 2 can in fact break all challenge parameters proposed by [FLLL26] in Table 4, for which the algorithms of [FLLL26] reported a runtime of at least 2^{128}.

Attacks on Local PRGs. We only list here some attacks on local PRGs. Exhaustive overviews can be found in [App13, Üna23a]. Bogdanov and Qiao [BQ09] gave approximation-based attacks on local PRGs, while Applebaum, Ishai and Kushilevitz [AIK08] gave shrinking-set attacks. Both attacks are poly-time if the PRG has a large enough polynomial stretch, however, for lower stretches their runtimes degrade gracefully.

An important class of attacks is given by attacks based on low *resilience*. The resilience-bound for local predicates has first been observed by Siegenthaler [Sie84]. It allows for poly-time attacks if the stretch is large enough [App13] by solving noisy linear equation systems. However, this attack does not degrade gracefully if the stretch is not large enough [OW14].

Algebraic Attacks. Algebraic attacks constitute a large class of attacks on PRGs of low degree (hence, also on local PRGs). Several works [CM03, AL16] could show that the stretch a local PRG may attain is bounded by the *rational degree*, that is also known as the *algebraic immunity* for Boolean functions [MPC04]. [CDM+18] extended those attacks to PRGs based on predicates with low *bit-fixing* rational degree (which is bounded by the algebraic immunity). It is not clear, if those attacks degrade gracefully for lower stretches, however, this seems highly plausible.

The general case of PRGs of low degree has been handled by [Üna23b, Üna23a]. These attacks compute a test polynomial that maps images of the PRG to zero while random bit strings are mapped to non-zero values with high probability. The degree of these test polynomials decreases for increasing stretches of the PRG, leading to graceful descends of runtime.

2 Preliminaries

We denote by $\log(\cdot) = \log_2(\cdot)$ the logarithm to base 2. By $e = 2.718\ldots$, we denote Euler's number, and by $\ln(\cdot) = \log_e(\cdot)$ the natural logarithm, that is, the logarithm to base e. Set additionally $[n] := \{1, 2, \ldots, n\}$, $[n, m] := [m] \setminus [n-1]$, $\mathbb{N} := \{1, 2, \ldots\}$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. By $\mathsf{w_H}(x)$, we denote the Hamming weight of a bit string $x \in \{0,1\}^n$. For a set S and $a \in \mathbb{N}_0$, we denote by

$$\binom{S}{a} := \{A \subseteq S \mid \#A = a\}$$

the set of all subsets of S of size a. Additionally, we set

$$\mathsf{poly}(n) := \bigcup_{c>0} O(n^c) \qquad \text{and} \qquad \mathsf{negl}(n) := \bigcap_{c>0} o(n^{-c}).$$

A function f is *noticeable*, if $f \notin \mathsf{negl}(n)$.

2.1 Statistical Preliminaries

The Chernoff bound implies that the probability of a uniformly random bit string $x \leftarrow \{0,1\}^n$ to have Hamming weight $\geq n/2 + \sqrt{n} \cdot \ell(n)$ is at most $\exp(-\frac{2}{3} \cdot \ell^2)$. This is indeed tight up to constants in the exponent:

Lemma 1. *Let $\ell \in o(\sqrt{n})$ such that $n/2 + \sqrt{n} \cdot \ell(n)$ always lies in $\{\lceil n/2 \rceil, \ldots, n\}$. There is a constant $c > 0$ and some positive function $p \in \mathsf{poly}(n)$ such that*

$$\Pr_{x \leftarrow \{0,1\}^n} \left[\mathsf{w_H}(x) \geq n/2 + \sqrt{n} \cdot \ell(n)\right] \geq \frac{\exp(-c \cdot \ell(n)^2)}{p(n)}$$

for almost all $n \in \mathbb{N}$.

We think that Lemma 1 is folklore, but could not find a reference with a proof of it. Hence, we give our own proof in Section A.

Definition 1 (HyperGeometric Distribution)**.** Let $0 \leq t \leq a \leq n$ and $0 \leq h \leq n$. The *hypergeometric* probability mass function is given by

$$\mathrm{hyg}(n, h, a, t) := \frac{\binom{h}{t} \cdot \binom{n-h}{a-t}}{\binom{n}{a}}. \tag{1}$$

A random variable X is distributed hypergeometrically with respect to n, h, a, if we have $\Pr[X = t] = \mathrm{hyg}(n, h, a, t)$ for $t \in \{0, \ldots, a\}$. We denote this *hypergeometric distribution* by $\mathrm{HyG}(n, h, a)$ and write $X \leftarrow \mathrm{HyG}(n, h, a)$.

Let $x \in \{0,1\}^n$ be any bit string of Hamming weight h. The hypergeometric distribution measures the probability that a random non-consecutive length-a substring of x has Hamming weight exactly t. Hoeffding [Hoe94] showed that the tail bound for the hypergeometric distribution is upper bounded by the tail bound for the Bernoulli distribution with parameter h/n.

Theorem 3 (Hypergeometric Tail Bound [Hoe94])**.** *We have for $\beta \in (0, h/n)$*

$$\Pr_{X \leftarrow \mathrm{HyG}(n,h,a)}[X \leq \beta \cdot a] \leq \exp(-2(h/n - \beta)^2 a).$$

Denote by $\mathcal{M}_b^{m \times n} = \left\{L = (l_i)_{i \in [m]} \in \mathbb{Z}_2^{m \times n} \;\middle|\; \mathsf{w_H}(l_i) = b\right\}$ the set of *b-sparse* binary $m \times n$ matrices, i.e., $L \in \mathcal{M}_b^{m \times n}$ if all rows of L have Hamming weight b. If b is even, $\mathcal{M}_b^{m \times n}$ will contain no full-rank matrix, as $(1, \ldots, 1)$ lies in the kernel of all b-sparse matrices. Hence, we consider the target rank of $n - 1$.

Definition 2. Set $\mu_{m,n,b} := \Pr_{L \leftarrow \mathcal{M}_b^{m \times n}}[\mathrm{rank}(M) \geq n - 1]$.

We could verify experimentally that $\mu_{n,n,10} \geq 50\%$ for $n \in \{256, 512, 894, 2048\}$. Cooper, Frieze and Pedgen [CFP19] showed, however, that asymptotically one needs $m \in \Omega(\log(n) \cdot n)$ to guarantee that $\mu_{m,n,b}$ does not vanish for evem $b \in O(1)$.

Theorem 4 (Rank of Random Sparse Binary Matrix[CFP19])**.** *Let $b \in O(1)$. If b is odd, set $n^* = n$, otherwise, $n^* = n - 1$. Let $(c_n)_n$ be a convergent sequence of real numbers such that we always have $m = n(\log(n) + c_n)/b \in \mathbb{N}$. ($(c_n)_n$ may converge towards $+\infty$ or $-\infty$.) We have*

$$\lim_{n \to \infty} \Pr_{L \leftarrow \mathcal{M}_b^{m \times n}}[\mathrm{rank}(L) = n^*] = \begin{cases} 0, & \text{if } c_n \to -\infty, \\ \exp(-\exp(-c)), & \text{if } c_n \to c, \\ 1, & \text{if } c_n \to +\infty. \end{cases}$$

2.2 PRGs

Definition 3 (Predicates)**.** Let $A \subset [n]$ be of size $a = \#A$ and $t \leq a$ We define the *Threshold predicate* for A by

$$\mathsf{THR}_{A,t} : \{0,1\}^n \longrightarrow \{0,1\}$$
$$x \longmapsto \begin{cases} 1, \text{ if } \#\{i \in A \mid x_i = 1\} \geq t, \\ 0, \text{ otherwise.} \end{cases}$$

If a is odd, the *Majority predicate* is defined by $\mathsf{MAJ}_A = \mathsf{THR}_{A,(a+1)/2}$. If a is even, it is defined by $\mathsf{MAJ}_A = \mathsf{THR}_{A,a/2}$. The *XOR predicate* for A is given by

$$\mathsf{XOR}_A : \{0,1\}^n \longrightarrow \{0,1\}$$
$$x \longmapsto \begin{cases} 1, \text{ if } \#\{i \in A \mid x_i = 1\} \text{ is } \textit{odd}, \\ 0, \text{ otherwise.} \end{cases}$$

Definition 4 (Pseudorandom Generator)**.** A function $F : \{0,1\}^n \to \{0,1\}^m$ with $m \geq n$ is called a *pseudorandom generator* (PRG) if we have for each poly-time distinguisher $\mathcal{A}$

$$\mathrm{adv}_F(\mathcal{A}) := \left| \Pr_{x \leftarrow \{0,1\}^n} [\mathcal{A}(F, F(x)) = 1] - \Pr_{y \leftarrow \{0,1\}^m} [\mathcal{A}(F, y) = 1] \right| \in \mathsf{negl}(n).$$

Definition 5 (Predicate-Based Functions)**.** Let $P : \{0,1\}^a \to \{0,1\}$ be a boolean function. A *P-based function* $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$ is determined by index-lists $(c_i^{(1)})_i, \ldots, (c_i^{(m)})_i \in [n]^a$ (where we require that each $(c_i^{(j)})_i$ contains every element of $[n]$ at most once). For $x \in \{0,1\}^n$, the j-th output y_j of $y = \mathcal{F}(x)$ is evaluated as

$$y_j = P(x_{c_1^{(j)}}, \ldots, x_{c_a^{(j)}}).$$

If P is symmetric, we can replace the sequences $(c_i^{(j)})_i$ by subsets $A_j = \{c_1^{(j)}, \ldots, c_a^{(j)}\}$. If P is the xor-sum of two symmetric predicates of arities a and b, then we can replace the sequences $(c_i^{(j)})_i$ by pairs (A_j, B_j) of subsets $A_j = \{c_1^{(j)}, \ldots, c_a^{(j)}\}$ and $B_j = \{c_{a+1}^{(j)}, \ldots, c_{a+b}^{(j)}\}$.

3 Bounding the Common Bias

To reason about the advantage of our algorithms, we introduce the notion of *common bias* for (distributions of) predicates and functions.

Definition 6 (Common Bias for Functions)**.** Let $\mathcal{D}$ be a distribution of functions $\{0,1\}^n \to \{0,1\}^m$. The *average-case common bias* of $\mathcal{D}$ is given by

$$\textsf{c-bias}_{\mathrm{ac}}(\mathcal{D}) := \max_{y : \mathsf{supp}(\mathcal{D}) \to \{0,1\}^m} \left(\Pr_{\mathcal{F} \leftarrow \mathcal{D}, x \leftarrow \{0,1\}^n} [\mathcal{F}(x) = y(\mathcal{F})] \right).$$

The *worst-case common bias* of $\mathcal{D}$ is given by

$$\textsf{c-bias}_{\mathrm{wc}}(\mathcal{D}) := \min_{\mathcal{F} \in \mathsf{supp}(\mathcal{D})} \left(\max_{y \in \{0,1\}^m} \left(\Pr_{x \leftarrow \{0,1\}^n} [\mathcal{F}(x) = y] \right) \right).$$

For a predicate $P : \{0,1\}^a \to \{0,1\}$, we define its common biases by

$$\textsf{c-bias}^m_{\mathrm{ac}}(P) := \textsf{c-bias}_{\mathrm{ac}}(\mathcal{D}_P), \qquad \textsf{c-bias}^m_{\mathrm{wc}}(P) := \textsf{c-bias}_{\mathrm{wc}}(\mathcal{D}_P)$$

where $\mathcal{D}_P$ is the uniform distribution of P-based functions $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$.

We first devise a concrete formula for $\mathsf{c\text{-}bias}^m_{\mathrm{ac}}(\mathsf{THR}_{a,t})$ and a concrete bound for $\mathsf{c\text{-}bias}^m_{\mathrm{wc}}(\mathsf{THR}_{a,t})$ in this section. We then show that both are asymptotically lower-bounded by $(2m)^{-\Theta(n/a)}/\mathsf{poly}(n)$. This implies that both quantities are noticeable for $a \in \Theta(n)$ linear.

We start by defining two probabilities for threshold predicates: The first probability measures the event that a uniformly random threshold predicate of fixed size a with threshold t accepts a fixed bit string x of length n and Hamming weight h.

$$q(n,h,a,t) := \Pr_{A \leftarrow \binom{[n]}{a}}\left[\mathsf{THR}_{A,t}(x) = 1\right].$$

(Note that the concrete string x is irrelevant. We could take for example $x = 1^h0^{n-h}$.) The second probability measures the event that m uniformly random threshold predicates, with size a and threshold t, accept a uniformly random bit string $x \leftarrow \{0,1\}^n$.

$$p(m,n,a,t) := \Pr_{A_1,\ldots,A_m \leftarrow \binom{[n]}{a}, x \leftarrow \{0,1\}^n}\left[\mathsf{THR}_{A_1,t}(x) = \cdots = \mathsf{THR}_{A_m,t}(x) = 1\right].$$

Note that $p(m,n,a,t)$ is exactly the average-case common-bias of $\mathsf{THR}_{a,t}$ if $t \leq (a+1)/2$. In general, we have

$$\mathsf{c\text{-}bias}^m_{\mathrm{ac}}(\mathsf{THR}_{a,t}) \geq \max(p(m,n,a,t), p(m,n,a,a+1-t)).$$

Proposition 1. *Let $m,n,h,a,t \in \mathbb{N}$ such that $n > 0$, $h \leq n$ and $t \leq a \leq n$. We have*

$$q(n,h,a,t) = \Pr_{X \leftarrow \mathrm{HyG}(n,h,a)}[X \geq t]$$

$$p(m,n,a,t) = 2^{-n} \cdot \sum_{h=0}^{n} \binom{n}{h} \cdot q(n,h,a,t)^m.$$

Proof. First, we use counting arguments to determine $q(n,h,a,t)$. We count the number of size a subsets of the seed x of Hamming weight h that give inputs accepted by the threshold predicate, that is, all the subsets of size a having at least t ones. Each size-a subset of x of weight i is obtained by taking i ones from the h ones of the seed and $a-i$ from the $n-h$ zeroes of the seed. Accordingly, there are $\binom{h}{i} \cdot \binom{n-h}{a-i}$ subsets of Hamming weight i. Summing from $i = t$ to a we count all the subsets where the threshold t-out-of-a accepts. Since $q(n,h,a,t)$ measures the probability for a random subset of size a, we divide the previous quantity by the total number of size-a subsets:

$$q(n,h,a,t) = \frac{1}{\binom{n}{a}} \cdot \sum_{i=t}^{a} \binom{h}{i} \cdot \binom{n-h}{a-i} = \sum_{i=t}^{a} \mathrm{hyg}(n,h,a,i) = \Pr_{X \leftarrow \mathrm{HyG}(n,h,a)}[X \geq t].$$

To determine $p(m,n,a,t)$ we rewrite the probability, conditioned on the Hamming weight of the seed

$$\begin{aligned}
&p(m,n,a,t)\\
&= \Pr_{A_1,\ldots,A_m \leftarrow \binom{[n]}{a}, x \leftarrow \{0,1\}^n}\left[\mathsf{THR}_{A_1,t}(x) = \cdots = \mathsf{THR}_{A_m,t}(x) = 1\right]\\
&= \sum_{h=0}^{n} \Pr_{A_1,\ldots,A_m \leftarrow \binom{[n]}{a}}\left[\mathsf{THR}_{A_1,t}(x) = \cdots = \mathsf{THR}_{A_m,t}(x) = 1 \mid \mathsf{w_H}(x) = h\right]\\
&\cdot \Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w_H}(x) = h\right]\\
&= \sum_{h=0}^{n} \Pr_{A_1,\ldots,A_m \leftarrow \binom{[n]}{a}}\left[\mathsf{THR}_{A_1,t}(1^h0^{n-h}) = \cdots = \mathsf{THR}_{A_m,t}(1^h0^{n-h}) = 1\right]\\
&\cdot \Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w_H}(x) = h\right]\\
&= \sum_{h=0}^{n} \left(\Pr_{A \leftarrow \binom{[n]}{a}}\left[\mathsf{THR}_{A,t}(1^h0^{n-h}) = 1\right]\right)^m \cdot \Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w_H}(x) = h\right]\\
&= \sum_{h=0}^{n} q(n,h,a,t)^m \cdot \frac{\binom{n}{h}}{2^n} = 2^{-n} \sum_{h=0}^{n} \binom{n}{h} \cdot q(n,h,a,t)^m.
\end{aligned}$$

□

Proposition 1 gives a concrete formula for the average case. Now, we turn to the worst-case common bias.

Proposition 2. *We have for each set* $A \in \binom{[n]}{a}$

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{THR}_{A,t}(x) = 1 \mid \mathsf{w_H}(x) = h] = q(n,h,a,t).$$

Proof. Denote by P the group of permutations over $[n]$. For a bit string x and a permutation τ, we set $\tau(x) = x_{\tau(1)} \cdots x_{\tau(n)}$. Further, we set $\tau(A) = \{\tau(i) \mid i \in A\}$. We have

$$\begin{aligned}
&\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{THR}_{A,t}(x) = 1 \mid \mathsf{w_H}(x) = h] = \Pr_{\tau \leftarrow P}\left[\mathsf{THR}_{A,t}(\tau(1^h 0^{n-h})) = 1\right] \\
&= \Pr_{\tau \leftarrow P}\left[\mathsf{THR}_{\tau(A),t}(1^h 0^{n-h}) = 1\right] \\
&= \Pr_{A' \leftarrow \binom{[n]}{a}}\left[\mathsf{THR}_{A',t}(1^h 0^{n-h}) = 1\right] = q(n,h,a,t).
\end{aligned}$$

□

Now, set

$$p'(m,n,a,t) = 2^{-n} \cdot \sum_{h=0}^{n} \binom{n}{h} \cdot (m \cdot q(n,h,a,t) - (m-1)).$$

We claim that p' lower bounds $\mathsf{c\text{-}bias}^m_{\mathrm{wc}}(\mathsf{THR}_{a,t})$ if $t \leq (a+1)/2$.

Proposition 3. *For* $t \leq (a+1)/2$, *we have* $\mathsf{c\text{-}bias}^m_{\mathrm{wc}}(\mathsf{THR}_{a,t}) \geq p'(m,n,a,t)$.

Proof. Let $A_1, \ldots, A_m \subset [n]$ all be of size a. We have

$$\begin{aligned}
&\mathsf{c\text{-}bias}^m_{\mathrm{wc}}(\mathsf{THR}_{a,t}) \geq \Pr_{x \leftarrow \{0,1\}^n}[\forall i : \ \mathsf{THR}_{A_i,t}(x) = 1] \\
&= \sum_{h=0}^{n} \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{w_H}(x) = h] \cdot \Pr_{x \leftarrow \{0,1\}^n}[\forall i : \ \mathsf{THR}_{A_i,t}(x) = 1 \mid \mathsf{w_H}(x) = h].
\end{aligned}$$

By a union bound[6], the probability of the event of all $\mathsf{THR}_{A_1,t}, \ldots, \mathsf{THR}_{A_m,t}$ accepting x of Hamming weight h can be bounded as

$$\begin{aligned}
&\Pr_{x \leftarrow \{0,1\}^n}[\forall i : \ \mathsf{THR}_{A_i,t}(x) = 1 \mid \mathsf{w_H}(x) = h] \\
&= 1 - \Pr_{x \leftarrow \{0,1\}^n}[\exists i : \ (\mathsf{THR}_{A_i,t}(x) = 0) \mid \mathsf{w_H}(x) = h] \\
&\geq 1 - \sum_{i=1}^{m} \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{THR}_{A_i,t}(x) = 0 \mid \mathsf{w_H}(x) = h] \\
&= 1 - \sum_{i=1}^{m} (1 - \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{THR}_{A_i,t}(x) = 1 \mid \mathsf{w_H}(x) = h]) \\
&= \sum_{i=1}^{m} \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{THR}_{A_i,t}(x) = 1 \mid \mathsf{w_H}(x) = h] - (m-1).
\end{aligned}$$

Because of Proposition 2, we have for each fixed set A_i

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{THR}_{A_i,t}(x) = 1 \mid \mathsf{w_H}(x) = h] = q(n,h,a,t).$$

Hence, the claim follows by $\sum_{i=1}^{m} \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{THR}_{A_i,t}(x) = 1 \mid \mathsf{w_H}(x) = h] = m \cdot q(n,h,a,t)$. □

To derive asymptotic bounds for p', we first show that q (respectively, $m \cdot q - (m-1)$) is larger than $1/2$ if the Hamming weight h of x is large. Indeed, this allows for lower bounds, since $p'(m,n,a,t) \geq 2^{-n}\binom{n}{h} \cdot (m \cdot q(n,h,a,t) - (m+1))$.

[6] The union bound stipulates $\Pr[\exists i : \ E_i] \leq \sum_i \Pr[E_i]$ for arbitrary events E_i.

Lemma 2. *For $h \geq n \cdot \frac{t}{a} + n \cdot \sqrt{\frac{\ln(2m)}{2a}}$, we have*

$$m \cdot q(n,h,a,t) - (m-1) \geq \frac{1}{2}.$$

Proof. Because of Proposition 1 and Theorem 3, we have (with $\beta = t/a$)

$$q(n,h,a,t) = \Pr_{X \leftarrow \mathrm{HyG}(n,h,a)}[X \geq t] \geq 1 - \exp\left(-2\left(\frac{h}{n} - \frac{t}{a}\right)^2 \cdot a\right),$$

which implies

$$m \cdot q(n,h,a,t) - (m-1) \geq 1 - m \cdot \exp\left(-2\left(\frac{h}{n} - \frac{t}{a}\right)^2 \cdot a\right).$$

For $h \geq n \cdot \frac{t}{a} + n \cdot \sqrt{\frac{\ln(2m)}{2a}}$, we have $\frac{h}{n} - \frac{t}{a} \geq \sqrt{\frac{\ln(2m)}{2a}}$, hence

$$\begin{aligned} 1 - m \cdot \exp\left(-2\left(\frac{h}{n} - \frac{t}{a}\right)^2 \cdot a\right) &\geq 1 - m \cdot \exp\left(-2 \cdot \frac{\ln(2m)}{2a} \cdot a\right) \\ &\geq 1 - \frac{m}{2m} = \frac{1}{2}. \end{aligned}$$

□

Theorem 5. *Let $t \leq (a+1)/2$. There is a positive polynomial $r \in \mathsf{poly}(n)$ such that*

$$p'(m,n,a,t) \geq (r(n) \cdot (2m)^{\Theta(n/a)})^{-1}.$$

Proof. Set $h = \left\lceil n \cdot \frac{t}{a} + n \cdot \sqrt{\frac{\ln(2m)}{2a}} \right\rceil$. Because of Lemma 2, we have

$$\begin{aligned} p'(m,n,a,t) &= 2^{-n} \cdot \sum_{k=0}^{n} \binom{n}{k} \cdot (m \cdot q(n,k,a,t) - (m-1)) \\ &\geq 2^{-n} \cdot \binom{n}{h} \cdot (m \cdot q(n,h,a,t) - (m-1)) \\ &\geq 2^{-n-1} \cdot \binom{n}{h} = 0.5 \cdot \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{w_H}(x) \geq h]. \end{aligned}$$

We want to invoke Lemma 1. For this end, set $\ell = \frac{h - n/2}{\sqrt{n}}$. We have

$$\begin{aligned} \ell &= \frac{\left\lceil n \cdot \frac{t}{a} + n \cdot \sqrt{\frac{\ln(2m)}{2a}} \right\rceil - n/2}{\sqrt{n}} \overset{t \leq (a+1)/2}{\leq} \frac{\left\lceil n \cdot \frac{a+1}{2a} + n \cdot \sqrt{\frac{\ln(2m)}{2a}} \right\rceil - n/2}{\sqrt{n}} \\ &= \Theta\left(\frac{\sqrt{n}}{2a} + \sqrt{\frac{n \ln(2m)}{2a}}\right). \end{aligned}$$

We can absorb $\frac{\sqrt{n}}{2a}$ into $\Theta\left(\sqrt{\frac{n\ln(2m)}{2a}}\right)$. It follows $\ell \in O\left(\sqrt{\frac{n\ln(2m)}{2a}}\right)$. Lemma 1 implies now the existence of a polynomial $r \in \mathsf{poly}(n)$ such that

$$\begin{aligned} \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{w_H}(x) \geq h] &= \Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w_H}(x) \geq n/2 + \sqrt{n} \cdot \ell\right] \\ &\geq \frac{\exp(-\Theta(\ell^2))}{r(n)} = \frac{\exp\left(-\Theta\left(\frac{n \cdot \ln(2m)}{2a}\right)\right)}{r(n)} = \frac{(2m)^{-\Theta(n/(2a))}}{r(n)}. \end{aligned}$$

□

Since $\mathsf{c\text{-}bias}^m_{\mathrm{ac}}(\mathsf{THR}_{a,t}), \mathsf{c\text{-}bias}^m_{\mathrm{wc}}(\mathsf{THR}_{a,t})$ are both lower bounded by the maximum of $p'(m,n,a,t)$ and $p'(m,n,a,a+1-t)$, Theorem 5 directly implies the following:

Corollary 1. *For $m \geq n$, we have*

$$\mathsf{c\text{-}bias}^m_{\mathrm{ac}}(\mathsf{THR}_{a,t}) \geq \mathsf{c\text{-}bias}^m_{\mathrm{wc}}(\mathsf{THR}_{a,t}) \geq (2m)^{-\Theta(n/a)}.$$

4 Inversion Attacks

We present in Algorithm 2 our inverter for $\mathsf{THR} \oplus \mathsf{XOR}$-based PRGs $\mathcal{F}$. Asymptotically, because of

Algorithm 2: Inversion Algorithm for $\mathsf{THR} \oplus \mathsf{XOR}$-based PRGs.

Data: A PRG $\mathcal{F} = \mathcal{T} \oplus \mathcal{L} : \{0,1\}^n \to \{0,1\}^{m'}$ where $\mathcal{T} = (\mathsf{THR}_{A_1,t}, \ldots, \mathsf{THR}_{A_{m'},t})$ and $\mathcal{L} = (\mathsf{XOR}_{B_1}, \ldots, \mathsf{XOR}_{B_{m'}})$. An output $y \in \{0,1\}^{m'}$.
Result: A preimage $x \in \{0,1\}^n$ or $\perp$.

```
Set a := #A_1;
if t ≤ (a+1)/2 then
    Set z := (1, ..., 1) ∈ {0,1}^{m'};
else
    Set z := (0, ..., 0) ∈ {0,1}^{m'};
end
for i ∈ [m'] do
    for j ∈ [n] do
        if j ∈ B_i then
            Set l_{i,j} := 1;
        else
            Set l_{i,j} := 0;          /* L = (l_{i,j}) ∈ Z_2^{m'×n} is a matrix representation of L */
        end
    end
end
Compute V := {x ∈ {0,1}^n | L · x = y ⊕ z};        /* We use Gaussian elimination here */
if V = ∅ then
    Return ⊥;
else
    return x ← V;
end
```

sparsity, this algorithm requires the function $\mathcal{F} : \{0,1\}^n \to \{0,1\}^{m'}$ to have a small logarithmic stretch m'/n. However, for concrete parameters, we could verify experimentally that $m' = n$ suffices to guarantee that $\mathrm{rank}(L) \geq n - 1$ with at least 45% probability.

In Definition 2, we denoted by $\mu_{m',n,b} = \Pr_{M \leftarrow \mathcal{M}_b^{m \times n}}[\mathrm{rank}(M) \geq n-1]$ the probability of a random binary b-sparse matrix having rank at least $n-1$. We can bound the advantage and running time of Algorithm 2 as follows:

Lemma 3. *Algorithm 2 has a time complexity of $O(m'n^2)$. In the average case, its advantage is lower-bounded by $\frac{1}{2} \cdot \mu_{m',n,b} \cdot \mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'}(\mathsf{THR}_{a,t})$.*

Proof. The time complexity of Algorithm 2 is dominated by applying Gaussian elimination on $L \in \mathbb{Z}_2^{m' \times n}$ to compute a basis of $V = \{x \in \mathbb{Z}_2^n \mid L \cdot x = y - z\}$. This time complexity can be upper bounded by $O(m'n^2)$.[7]

For the advantage, note that our algorithm succeeds if $z = \mathcal{T}(x)$ and if it guesses $x \leftarrow V$ correctly. The probability of $z = \mathcal{T}(x)$ is $\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'}(\mathsf{THR}_{a,t})$. If $z = \mathcal{T}(x)$, then V will contain x among $2^{n-\mathrm{rank}(L)} - 1$ other elements. With probability $\mu_{m',n,b}$, V is of size 2. In this case, Algorithm 2 will sample the correct x from V with probability at least $1/2$. Hence, the bound on *Algorithm* 2's advantage follows. □

We see that the advantage of Algorithm 2 is directly linked to the common bias of THR-predicates. While for concrete parameters, it suffices to take $m' = n$, asymptotically, we need by Theorem 4 that $b \cdot \frac{m'}{n} - \log(n)$ is strictly monotonically increasing.

[7] By using special linear solvers for sparse matrices, e.g. Wiedemann's algorithm [Wie86], one can further reduce the time complexity to $O(bm'n)$.

Corollary 2. *Let $m' \geq c \cdot n\log(n)$ for some constant $c > 1/b$. Then, Algorithm 2's advantage in the average case at inverting $\mathcal{F} : \{0,1\}^n \to \{0,1\}^{m'}$ is at least $(2m')^{-\Theta(n/a)}$. For $a \in \Theta(n)$, Algorithm 2 gives a poly-time attack with noticeable advantage.*

Remark 4 (On Concrete Runtime Claims). For the numbers reported in Tables 1, 3, 4 and 10, we ignore constant factors (as is usual in cryptanalysis) and calculate the combined runtime of Algorithm 2 as $T = m' \cdot n^2$. To bound the advantage of Algorithm 2, we need to bound $\mu_{m',n,b}$. To the best of our knowledge, $\mu_{m',n,b}$ has no closed formula. Therefore, for the parameters considered in Tables 1, 3, 4 and 10, we estimate $\mu_{m',n,b}$ empirically using 1000 independent experiments for each tuple (m', n, b). For $m' = n$, the resulting estimates are reported below.

b	$n = m'$					
5	42	80	85	128	133	263
$\mu_{n,n,5}$	77.9%	71.3%	66.9%	59.1%	57.7%	29.3%
7	54	80	91	128	141	285
$\mu_{n,n,7}$	84.6%	83.9%	83.8%	83.3%	83.8%	78.5%
8	65	80	105	128	155	303
$\mu_{n,n,8}$	55.7%	54.9%	58.6%	54.8%	54.0%	52.9%

b	$n = m'$								
10	85	128	146	205	256	365	512	894	2048
$\mu_{n,n,10}$	57.7%	59.3%	57.1%	54.2%	56.7%	56.8%	56.2%	54.9%	53.6%

Since all our approximations of $\mu_{n,n,b}$ are larger than $1/4$, we take a conservative approach and lower bound it by $1/4$. This leads to an advantage of $\text{adv} = \mathsf{c\text{-}bias}_{\text{ac}}^{m'}(\mathsf{THR}_{a,t}) \cdot 2^{-3}$. Following the multi-instance interpretation of Remark 3, we report the performance of Algorithm 2 as T/adv.

Note that we always assume that our attack only reads the first m' output bits of a PRG $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$ and discards the rest. This way, we can have runtimes that are significantly lower than m.

5 Alternative Predicates

One way to circumvent our attacks is to replace the MAJ-XOR predicate with alternative predicates that still meet the constraints of its use, namely being efficiently computable and expressible as the XOR of two symmetric functions. To investigate such alternatives, we first recall several notions from the theory of Boolean functions used in cryptography, which provide the tools needed to analyze the parameters relevant to attacks on Goldreich's PRG. We then introduce a family of predicates that interpolate between the XOR and the majority functions. We show that predicates in this family that are close to the majority function exhibit similar cryptographic properties, while offering improved resistance against the attacks based on common bias. Then, we study the Boolean function that outputs 1 if and only if the input has Hamming weight congruent to 0 modulo 3, and show how it can serve as a building block for constructing a suitable predicate. Finally, we exhibit the relevant parameters for concrete instances of these families of predicates and propose instances for 128-bit security.

5.1 Predicates and Boolean Functions used in Cryptography

Predicates are commonly viewed as mappings from $\{0,1\}^n$ to $\{0,1\}$, whereas Boolean functions are defined as mappings from $\mathbb{F}_2^n$ to $\mathbb{F}_2$, with both inputs and outputs naturally taken in the field $\mathbb{F}_2$. We refer to the book by Carlet [Car21] for a comprehensive treatment of Boolean functions in cryptography. In this subsection, we briefly recall the notions of algebraic normal form, balancedness, nonlinearity, and algebraic immunity. The first notion fixes the representation used to later discuss algebraic degree, while the remaining three are essential for analyzing attacks on Goldreich's PRG.

Definition 7 (Algebraic Normal Form (ANF) and degree)**.** The *Algebraic Normal Form* of a Boolean function f is given by its n-variate polynomial representation over $\mathbb{F}_2$ (i.e. belonging to

$\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$):

$$f(x) = \sum_{I \subseteq [n]} a_I \left(\prod_{i \in I} x_i \right) = \sum_{I \subseteq [n]} a_I x^I,$$

where $a_I \in \mathbb{F}_2$. The algebraic degree of f is: $\deg(f) = \max_{\{I \mid a_I = 1\}} \#I$ (with the convention that $\deg(0) = 0$). Any term $\prod_{i \in I} x_i$ is called a monomial and its degree equals $\#I$.

Definition 8 (Balancedness)**.** A Boolean function $f \in \mathcal{B}_n = \{0,1\}^{\{0,1\}^n}$ is said to be *balanced* if and only if $\#\mathsf{supp}(f) = \#\mathsf{supp}(f+1) = 2^{n-1}$, where the support of f denotes the set $\{x \in \mathbb{F}_2^n \mid f(x) = 1\}$.

Note that if a predicate is not balanced, the output of Goldreich's PRG instantiated with this predicate is also unbalanced, and its bias toward 0 or toward 1 can be exploited to distinguish the pseudorandom output from uniform. For polynomial stretch, one therefore considers t-resilient predicates (see, e.g., [AL16]), namely predicates that remain balanced even after fixing up to t input variables.

Definition 9 (Nonlinearity)**.** The *nonlinearity* $\mathsf{NL}(f)$ of a Boolean function $f \in \mathcal{B}_n$, where n is a positive integer, is the minimum Hamming distance between f and all the affine functions in $\mathcal{B}_n$:

$$\mathsf{NL}(f) = \min_{g,\, \deg(g) \leq 1} \{d_H(f, g)\},$$

with $d_H(f, g) = \#\{x \in \mathbb{F}_2^n \mid f(x) \neq g(x)\}$ the Hamming distance between f and g, and $g(x) = a \cdot x + \varepsilon$; $a \in \mathbb{F}_2^n, \varepsilon \in \mathbb{F}_2$ (where $\cdot$ is an inner product in $\mathbb{F}_2^n$).

Nonlinearity allows measuring the distance of a Boolean function from its closest affine approximation. For any affine function we have:

$$\Pr_{x \leftarrow \{0,1\}^n} [f(x) = \ell(x)] \leq 1 - \frac{\mathsf{NL}(f)}{2^n}.$$

Since the constant functions 0 and 1 are affine, the above equation also applies for $\Pr[f(x) = c]$ as considered in the equations of the bias. There is also a connection to the common bias, which measures the correlation between multiple equations and constant functions whereas the nonlinearity measures the correlation between only one predicate and affine functions.

Definition 10 (Algebraic Immunity, [MPC04])**.** The *algebraic immunity* of a Boolean function $f \in \mathcal{B}_n$, denoted as $\mathsf{AI}(f)$, is defined as:

$$\mathsf{AI}(f) = \min_{g \neq 0} \{\deg(g) \mid fg = 0 \text{ or } (f+1)g = 0\},$$

The function g is called an *annihilator* of f resp. $f + 1$.

In [AL16], the algebraic immunity is also referred as rational degree, and they show that Goldreich's PRG for a polynomial stretch s is insecure if used with a predicate f such that $\mathsf{AI}(f) < s$.

5.2 From XOR to MAJ

In this part, we introduce a class of functions that interpolates between the XOR and majority functions in n variables. We consider functions that coincide with the majority function on inputs of low and high Hamming weight, and with XOR_n (or its complement) for the remaining inputs.

Definition 11. Let $n \in \mathbb{N}$ and $d \in \mathbb{N}_0$ such that $d \leq n/2$ and $\varepsilon \in \{0,1\}$. We denote by $f_{d,\epsilon,n}$ the function defined by:

$$f_{d,\epsilon,n} : \{0,1\}^n \longrightarrow \{0,1\}$$

$$(x_1, \ldots, x_n) \longmapsto \begin{cases} 0, & \text{if } \mathsf{w_H}(x) < d, \\ 0, & \text{if } \mathsf{w_H}(x) \equiv \varepsilon \mod 2 \text{ and } \mathsf{w_H}(x) \in [d, n-d], \\ 1, & \text{if } \mathsf{w_H}(x) \not\equiv \varepsilon \mod 2 \text{ and } \mathsf{w_H}(x) \in [d, n-d], \\ 1, & \text{if } \mathsf{w_H}(x) > n - d. \end{cases}$$

Parameter	3	4	5	6	7	8	9	10	11	12	13	14	15
Degree	2	4	5	5	7	8	8	10	11	11	13	14	14
Algebraic immunity	1	2	2	2	3	3	3	4	4	4	5	5	5
Nonlinearity	2	5	7	14	37	85	170	341	683	1366	2731	5461	10922

Table 5: Cryptographic parameters of the function $\mathsf{MOD3}_n$ functions for $n \in [3, 15]$.

Parameter	3	4	5	6	7	8	9	10	11	12	13	14	15
Degree	3	4	4	6	7	7	9	10	10	12	13	13	15
Algebraic immunity	1	2	2	2	3	3	3	4	4	4	5	5	5
Nonlinearity	1	5	10	21	37	74	171	341	682	1365	2731	5462	10923

Table 6: Cryptographic parameters of the function $1\mathsf{MOD3}_n$ functions for $n \in [3, 15]$.

Remark 5. The function $f_{d,\epsilon,n}$ behaves like $\mathsf{XOR}_n + \varepsilon$ on the slices of intermediate Hamming weight (namely, between d and $n - d$), and like the majority function on inputs of extremal Hamming weight. The two corner cases are $f_{0,\varepsilon,n}$, which is the affine function $\mathsf{XOR}_n + \varepsilon$, and $f_{\lfloor (n/2) \rfloor, \lceil (n/2) \rceil + 1 \mod 2, n}$, which is the majority function.

First, let us point out that all these functions are balanced when n is odd.

Proposition 4. *Let $n \in \mathbb{N}$ be odd and $d \in \mathbb{N}_0$ such that $d \leq n/2$ and $\varepsilon \in \{0, 1\}$. The function $f_{d,\epsilon,n}$ is balanced.*

The proof is given in Section B. Then, we show the algebraic immunity of these functions is at least d. To do so we recall a property on threshold functions.

Further, we note that the algebraic immunity of these functions is at least d. To do so we recall a property on threshold functions.

Property 1 (Algebraic Immunity of Threshold Functions, *i.e.* [CM22]). Let $n \in \mathbb{N}$ and $d \in \mathbb{N}_0$ such that $1 \leq d \leq n$. The threshold function $\mathsf{THR}_{n,d}$ satisfies:

$$\mathsf{AI}(\mathsf{THR}_{n,d}) = \min(d, n - d + 1).$$

Proposition 5. *Let $n \in \mathbb{N}$ and $d \in \mathbb{N}_0$ such that $d \leq n/2$ and $\varepsilon \in \{0, 1\}$. The function $f_{d,\epsilon,n}$ satisfies:*

$$\mathsf{AI}(f_{d,\epsilon,n}) \geq d.$$

The proof is given in Section B.

5.3 Properties of Hamming Weight Modulus 3

In this part, we introduce functions defined according to the congruence class modulo three of the Hamming weight of their input.

Definition 12. Let $n \in \mathbb{N}$ and $t \in \{0, 1, 2\}$. We denote by $t\mathsf{MOD3}_n$ the function defined by

$$\begin{aligned} t\mathsf{MOD3}_n : \{0,1\}^n &\longrightarrow \{0,1\} \\ (x_1, \ldots, x_n) &\longmapsto \begin{cases} 1, & \text{if } \mathsf{w_H}(x) \equiv t \pmod 3, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

We will also use the notation $\mathsf{MOD3}_n$ to denote $t\mathsf{MOD3}_n$ when $t = 0$. In Table 5 (resp. Table 6) we denote some properties of $\mathsf{MOD3}_n$ (resp. $1\mathsf{MOD3}_n$)for small values of m obtained experimentally (using Sage).

First, we establish a lower bound on the algebraic degree of functions whose definition depends solely on the Hamming weight of the input modulo three. To this end, we begin by recalling a result on the algebraic degree of symmetric Boolean functions.

Property 2 (Theorem 1 [CV05]). Let f be a symmetric Boolean function with simplified value vector $\mathbf{v}_f = (v_0, \ldots, v_n)$ where v_i denotes the value f takes on an input of Hamming weight i. Then, $\mathbf{v}_f$ is periodic with period 2^t, $2^t < n$ if and only if $\deg(f) \leq 2^t - 1$.

Proposition 6. *Let $n \in \mathbb{N}$ with $n \geq 6$, and let f be a symmetric Boolean function whose value depends only on the congruence class modulo three of the Hamming weight of its input, that is,*

$$f(x_1, \ldots, x_n) = \begin{cases} a, & \text{if } \mathsf{w_H}(x) \equiv 0 \pmod 3, \\ b, & \text{if } \mathsf{w_H}(x) \equiv 1 \pmod 3, \\ c, & \text{if } \mathsf{w_H}(x) \equiv 2 \pmod 3, \end{cases}$$

where $a, b, c \in \{0, 1\}$. Then, either f is constant or $\deg(f) \geq \frac{n}{3}$.

The proof of the proposition is given in Section B.

We next establish a bound on the algebraic immunity of the function $\mathsf{MOD3}_n$. To this end, we first prove a property of symmetric Boolean functions whose value depends only on the congruence class modulo three of the Hamming weight of their input, the proof is given in Section B.

Proposition 7. *Let $n \in \mathbb{N}$ with $n \geq 6$, and let f be an n-variable Boolean function. If, for every triplet of distinct indices $(i, j, k) \in [n]$, and for every pair $(x, y) \in \{0, 1\}^n \times \{0, 1\}^n$ such that:*

- *$x_i = x_j = x_k = 0$ and $y_i = y_j = y_k = 1$,*
- *for all $\ell \in [n] \setminus \{i, j, k\}$, $x_\ell = y_\ell$,*

the function f satisfies $f(x) = f(y)$, then the value of f depends only on the congruence class modulo three of the Hamming weight of its input.

We can now give a lower bound on the algebraic immunity of $\mathsf{MOD3}_n$ (the proof is given in Section B).

Proposition 8. *Let $n \in \mathbb{N}$ with $n \geq 6$. The function $\mathsf{MOD3}_n$ satisfies:*

$$\mathsf{AI}(\mathsf{MOD3}_n) \geq \frac{n}{3}.$$

5.4 Concrete Parameters

In this part, we present concrete parameter choices for the two classes of functions considered in this section, in order to compare them with the $\mathsf{THR} \oplus \mathsf{XOR}$ predicate and to propose new challenge parameters.

Table 7 reports the parameters of several functions in 11 variables. In this example, we observe that the majority function achieves comparable or better values for the standard cryptographic criteria, namely balancedness, algebraic immunity, and nonlinearity, but exhibits a significantly worse common bias.

Table 8 reports the values of the common bias for alternative predicates evaluated on the challenge parameters of [FLLL26], using the parameters of our attack given in Table 4. We observe that the predicates introduced in Section 5.2, as well as $\mathsf{MOD3}_n$, achieve uniformly better common bias values across all the parameters considered, with several instances exhibiting a significant improvement.

We propose new challenge parameters targeting 128-bit security in Table 9[8], following the blueprint of Applebaum and Lovett [AL16] and subsequent parameter recommendations [FLLL26, BCPR25]. In particular, we require an algebraic immunity of at least $\lceil 8s + 1 \rceil$ and a resilience of at least $\lceil 5s \rceil$. Our main novelty is to additionally enforce a common bias $\mathsf{c\text{-}bias}^n$ smaller than 2^{-128}. Some parameters based on the interpolation predicates satisfy the common-bias criterion, but are vulnerable to a degree-one generalization of common bias, which we discuss in Appendix 5. We therefore exclude the affected parameters from the proposed challenges.

[8] In a previous version of this paper, we listed in Table 9 parameters of $f_{d,\varepsilon,a} \oplus \mathsf{XOR}_b$ for $(n, s, d, \varepsilon, a, b) \in \{(256, 2, 17, 0, 101, 10), (412, 4.5, 37, 0, 127, 23), (894, 3.4, 29, 0, 71, 17)\}$. We removed them after an AsiaCrypt reviewer pointed out to us that they were broken. Independently, Wang, Wang, Liu and Meier [WWLM26] also broke the 128 bit security of those candidate parameters.

Function	deg	AI	NL	balanced	c-bias$^{16}_{\text{ac}}$	c-bias$^{22}_{\text{ac}}$	c-bias$^{33}_{\text{ac}}$	c-bias$^{55}_{\text{ac}}$	c-bias$^{110}_{\text{ac}}$
$f_{5,1,11} = \mathsf{MAJ}_{11}$	8	6	772	yes	2.69	4.20	6.82	11.93	24.63
$f_{5,0,11}$	8	5	352	yes	4.32	6.45	10.07	17.17	34.82
$f_{4,1,11}$	8	4	112	yes	6.40	9.55	14.76	24.96	50.28
$f_{3,0,11}$	10	3	22	yes	8.87	13.36	20.63	34.73	69.74
$f_{2,1,11}$	10	2	2	yes	11.83	17.38	27.34	46.30	93.00
$f_{1,0,11} = \mathsf{XOR}_{11}$	1	1	0	yes	14.96	21.00	32.00	54.00	109.00
$\mathsf{MOD3}_{11}$	11	4	683	no	7.90	12.52	19.23	32.17	64.37
$1\mathsf{MOD3}_{11}$	10	4	682	no	8.04	12.55	19.21	32.11	64.26
$2\mathsf{MOD3}_{11}$	11	4	683	no	7.90	12.52	19.23	32.17	64.37

Table 7: Cryptographic parameters of alternative functions in 11 variables. The column c-bias$^{n}_{\text{ac}}$ refers to minus the logarithm in basis two of the average case common bias over n equations, for a seed of size n. The larger this value is, the more resistant the function is to our attack.

n	λ	MAJ_{47}	$f_{23,0,47}$	$f_{21,0,47}$	$f_{19,0,47}$	$f_{17,0,47}$	$f_{14,1,47}$	$f_{7,0,47}$	$\mathsf{MOD3}_{47}$
91	12	6.21	7.30	10.63	14.75	19.68	28.67	58.23	60.25
141	16	11.00	12.61	17.95	24.59	32.46	46.70	93.15	91.84
285	30	24.51	27.64	38.62	52.38	68.58	97.76	192.23	184.26
n	λ	MAJ_{57}	$f_{28,1,57}$	$f_{24,1,57}$	$f_{17,0,57}$	$f_{11,0,57}$	$f_{7,0,57}$	$f_{4,1,57}$	$\mathsf{MOD3}_{57}$
105	12	6.01	6.94	13.28	30.65	52.81	71.98	87.44	68.44
155	16	10.20	11.51	21.28	47.64	80.85	109.53	134.16	100.03
303	28	22.28	24.73	44.34	96.90	162.39	218.72	267.34	193.62
n	λ	MAJ_{75}	$f_{37,0,75}$	$f_{31,0,75}$	$f_{25,0,75}$	$f_{17,0,75}$	$f_{11,0,75}$	$f_{5,0,75}$	$\mathsf{MOD3}_{75}$
146	13	6.89	7.71	17.27	31.58	58.92	86.81	121.68	93.01
205	17	10.91	12.01	25.93	46.54	85.49	124.97	174.69	129.86
365	28	21.58	23.46	48.94	86.42	156.54	227.19	315.98	229.89

Table 8: Average common bias c-bias$^{n+\lambda}_{\text{ac}}$ (minus logarithm in base two) for some predicate over a bits and different seed sizes n with λ extra equations. The larger this value is, the more resistant the function is to our attack.

We observe that for small values of n, no instance of $\mathsf{MAJ} \oplus \mathsf{XOR}$ can simultaneously achieve the algebraic immunity required for the considered stretch and a sufficiently low common bias. In this parameter regime, the $\mathsf{MAJ} \oplus \mathsf{XOR}$ predicate is therefore inadequate, and viable constructions may instead rely on instances from the two new predicate families introduced in this section, combined with the XOR predicate. For larger values of n, instances of $\mathsf{MAJ} \oplus \mathsf{XOR}$ satisfying all three constraints do exist. However, even in this regime, replacing $\mathsf{MAJ} \oplus \mathsf{XOR}$ by a predicate interpolating between MAJ and XOR allows one to reduce the common bias at essentially no additional cost: only a very small number of extra variables is required (two in the examples reported in the tables), while algebraic immunity and resilience remain unchanged. For the same parameters, the MOD3 alternative requires approximately 3/2 times more variables, but in return guarantees a substantially lower common bias.

Approximating MOD3 by a constant reduces $\mathsf{MOD3} \oplus \mathsf{XOR}$ to a noisy linear system, with a noise rate close to 1/3. This is reminiscent of LPN, for which no efficient general algorithm is known at such noise rates. Similar noisy linear systems already arise from threshold predicates after fixing some of their variables, as the resulting restrictions are generally unbalanced.

We conclude this section by pointing out the similarity between Goldreich PRGs instantiated with $\mathsf{MOD3} \oplus \mathsf{XOR}$ predicates and weak PRFs based on alternating moduli [BIP+18, DGH+21, APRR24]. Indeed, combining an XOR predicate with a MOD3 predicate intertwines arithmetic equations modulo 2 and modulo 3, effectively leading to equation systems modulo 6 closely related to those arising

n	s	$\mathsf{MAJ}_a \oplus \mathsf{XOR}_b$	$f_{d,\varepsilon,a} \oplus \mathsf{XOR}_b$	$\mathsf{MOD3}_a \oplus \mathsf{XOR}_b$
256	2	$\emptyset$	$\emptyset$	(51,11)
512	4.5	$\emptyset$	$\emptyset$	(111,24)
894	3.4	$\emptyset$	$\emptyset$	(87,18)
1024	2	$\emptyset$	(17,0,35,10)	(51,11)
1536	3	(49,15)	(25,0,51,15)	(75,16)
2000	4	(64,21)	(33,0,67,21)	(99,22)
2048	3	(49,15)	(25,0,51,15)	(75,16)

Table 9: New challenges for 128-bit of a Goldreich PRG $\mathcal{F} : \{0,1\}^n \to \{0,1\}^{n^s}$.

in alternating-moduli constructions, which resist currently known algebraic solving techniques. The substantial body of cryptanalysis on alternating-moduli weak PRFs [CCKK21, JMN23, AR25, SW25] therefore provides both evidence for the potential hardness of these constructions and a natural starting point for further analysis of these PRG candidates.

Acknowledgments:

We used LLMs to improve the quality of writing (grammar, phrasing, ...) of this manuscript. We are inclined to Geoffroy Couteau for encouraging us to research $\mathsf{MAJ} \oplus \mathsf{XOR}$-based PRGs in the large locality setting. We thank Qingju Wang and Antoine Sidem for helpful discussions on the topic.

Pierrick Méaux was funded by the European Research Council (ERC) under the Advanced Grant program (reference number: 787390).

This research was funded in whole or in part by the Austrian Science Fund (FWF) 10.55776/ESP1735525. For Open Access purposes, the author has applied a CC BY public copyright license to any author accepted manuscript version arising from this submission.

References

[ABBS25] Amit Agarwal, Carsten Baum, Lennart Braun, and Peter Scholl. Low-bandwidth mixed arithmetic in VOLE-based ZK from low-degree PRGs. pages 396–426, 2025.

[ABCM25] Benny Applebaum, Dung Bui, Geoffroy Couteau, and Nikolas Melissaris. Structured-Seed Local Pseudorandom Generators and Their Applications. In Alina Ene and Eshan Chattopadhyay, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025)*, volume 353 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:26, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. pages 171–180, 2010.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. pages 166–175, 2004.

[AIK08] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in nc0. *Comput. Complex.*, 17(1):38–69, apr 2008.

[AL16] Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. pages 1087–1100, 2016.

[App13] Benny Applebaum. Cryptographic hardness of random local functions-survey. page 599, 2013.

[APRR24] Navid Alamati, Guru-Vamsi Policharla, Srinivasan Raghuraman, and Peter Rindal. Improved alternating-moduli PRFs and post-quantum signatures. pages 274–308, 2024.

[AR25] Irati Manterola Ayala and Håvard Raddum. Zeroed out: Cryptanalysis of weak PRFs in alternating moduli. 2025(2):1–15, 2025.

[BCM+24] Dung Bui, Geoffroy Couteau, Pierre Meyer, Alain Passelègue, and Mahshid Riahinia. Fast public-key silent OT and more from constrained Naor-Reingold. pages 88–118, 2024.

[BCPR25] Christina Boura, Geoffroy Couteau, Léo Perrin, and Yann Rotella. SoK: On shallow weak PRFs A common symmetric building block for MPC protocols. 2025(3):289–336, 2025.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: New simple PRF candidates and their applications. pages 699–729, 2018.

[BQ09] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. In Irit Dinur, Klaus Jansen, Joseph Naor, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 392–405, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Car21] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, 2021.

[CCKK21] Jung Hee Cheon, Wonhee Cho, Jeong Han Kim, and Jiseung Kim. Adventures in crypto dark matter: Attacks and fixes for weak pseudorandom functions. pages 739–760, 2021.

[CDD+24] Geoffroy Couteau, Lalita Devadas, Srinivas Devadas, Alexander Koch, and Sacha Servan-Schreiber. QuietOT: Lightweight oblivious transfer with a public-key setup. pages 197–231, 2024.

[CDM+18] Geoffroy Couteau, Aurélien Dupin, Pierrick Méaux, Mélissa Rossi, and Yann Rotella. On the concrete security of Goldreich's pseudorandom generator. pages 96–124, 2018.

[CFP19] Colin Cooper, Alan Frieze, and Wesley Pegden. On the rank of a random binary matrix. *The Electronic Journal of Combinatorics*, 26, 2019.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption over the torus. 33(1):34–91, January 2020.

[CM03] Nicolas Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. pages 345–359, 2003.

[CM22] Claude Carlet and Pierrick Méaux. A complete study of two classes of boolean functions: Direct sums of monomials and threshold functions. *IEEE Trans. Inf. Theory*, 68(5):3404–3425, 2022.

[CV05] Anne Canteaut and Marion Videau. Symmetric boolean functions. *IEEE Trans. Inf. Theory*, 51(8):2791–2811, 2005.

[DGH+21] Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. MPC-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. pages 517–547, 2021.

[DMS06] Deepak Kumar Dalai, Subhamoy Maitra, and Sumanta Sarkar. Basic theory in construction of boolean functions with maximum possible annihilator immunity. 40(1):41–58, 2006.

[FLLL26] Ximing Fu, Mo Li, Shihan Lyu, and Chuanyi Liu. Attacks on goldreich's pseudorandom generators by grouping and solving. pages 3–32, 2026.

[GGM24] François Gérard, Agnese Gini, and Pierrick Méaux. TooLIP: How to find new instances of FiLIP cipher with smaller key size and new filters. pages 21–45, 2024.

[Gol11] Oded Goldreich. *Candidate one-way functions based on expander graphs*, page 76–87. Springer-Verlag, Berlin, Heidelberg, 2011.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. 28(4):1364–1396, 1999.

[Hoe94] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In N. I. Fisher and P. K. Sen, editors, *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer New York, New York, NY, 1994.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147, 1995.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. pages 60–73, 2021.

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. pages 670–699, 2022.

[JMN23] Thomas Johansson, Willi Meier, and Vu Nguyen. Differential cryptanalysis of mod-2/mod-3 constructions of binary weak prfs. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 477–482, 2023.

[MCJS19] Pierrick Méaux, Claude Carlet, Anthony Journault, and François-Xavier Standaert. Improved filter permutators for efficient FHE: Better instances and implementations. pages 68–91, 2019.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. pages 311–343, 2016.

[MPC04] Willi Meier, Enes Pasalic, and Claude Carlet. Algebraic attacks and decomposition of Boolean functions. pages 474–491, 2004.

[MST03] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On e-biased generators in NC0. pages 136–145, 2003.

[NW88] Noam Nisan and Avi Wigderson. Hardness vs. randomness (extended abstract). pages 2–11, 1988.

[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. IEEE Computer Society, 2014.

[RVV24] Seyoon Ragavan, Neekon Vafa, and Vinod Vaikuntanathan. Indistinguishability obfuscation from bilinear maps and LPN variants. pages 3–36, 2024.

[Sie84] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications (corresp.). *IEEE Transactions on Information Theory*, 30(5):776–780, 1984.

[Slo26] N. J. A: Sloane. Decimal expansion of product (1 - 1/2 k), 2026.

[SW25] Antoine Sidem and Qingju Wang. General key recovery attack on pointwise-keyed functions - application to alternating moduli weak PRFs. pages 131–154, 2025.

[Ünal23a] Akin Ünal. New baselines for local pseudorandom number generators by field extensions. Cryptology ePrint Archive, Report 2023/550, 2023.

[Ünal23b] Akin Ünal. Worst-case subexponential attacks on PRGs of constant degree or constant locality. pages 25–54, 2023.

[Wie86] D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, 1986.

[Wil13] Ryan Williams. Natural proofs versus derandomization. pages 21–30, 2013.

[WWLM26] Tianning Wang, Haoyang Wang, Fukang Liu, and Willi Meier. Cryptanalysis of goldreich's PRGs with MAJ–XOR predicates. Cryptology ePrint Archive, Paper 2026/2156, 2026.

[YGJL22] Jing Yang, Qian Guo, Thomas Johansson, and Michael Lentmaier. Revisiting the concrete security of goldreich's pseudorandom generator. *IEEE Transactions on Information Theory*, 68(2):1329–1354, 2022.

Appendix

A Left Out Proofs

We will give here an elementary proof of Lemma 1. For the ease of notation, we introduce the following notation:

Definition 13. For two functions $f, g : \mathbb{N} \to \mathbb{R}$, we write $f \sim g$ if there is a $c \in \mathbb{N}$ such that

$$f(n) \leq n^c \cdot g(n) \leq n^{2c} \cdot f(n)$$

for almost all $n \in \mathbb{N}$.

Note that from $f_1 \sim f_2$ and $g_1 \sim g_2$, we cannot deduce $f_1 + g_1 \sim f_2 + g_2$.[9] However, we always[10] have for $f_1 \sim f_2$, $g_1 \sim g_2$

$$f_1 \cdot g_1 \sim f_2 \cdot g_2.$$

Additionally, we have for $f(n) \sim g(n)$ and $t(n) \sim n$

$$f(t(n)) \sim g(t(n)).$$

We prove both in the subsequent lemmas:

Lemma 4. *Let $f_1, f_2, g_1, g_2 : \mathbb{N} \to \mathbb{R}$. If $f_1 \sim f_2$ and $g_1 \sim g_2$, then $f_1 \cdot g_1 \sim f_2 \cdot g_2$.*

Proof. Let $c > 0$ and $n_0 \in \mathbb{N}$, such that we have for all $n \geq n_0$

$$f_1(n) \leq n^c \cdot f_2(n) \leq n^{2c} \cdot f_1(n),$$
$$g_1(n) \leq n^c \cdot g_2(n) \leq n^{2c} \cdot g_1(n).$$

It follows directly

$$f_1(n) \cdot g_1(n) \leq n^{2c} \cdot f_2(n) \cdot g_2(n) \leq n^{4c} \cdot f_1(n) \cdot g_1(n)$$

for all $n \geq n_0$. Hence, $f_1 \cdot g_1 \sim f_2 \cdot g_2$. □

Lemma 5. *Let $f, g : \mathbb{N} \to \mathbb{R}$ and $t : \mathbb{N} \to \mathbb{N}$. If $f \sim g$ and $t \sim 1$, we have $f \circ t \sim g \circ t$.*

Proof. It suffices to show that $f(t(n)) \leq \mathsf{poly}(n) \cdot g(t(n))$. Mutatis mutandis, one can show that $g(t(n)) \leq \mathsf{poly}(n) \cdot f(t(n))$. From both bounds, the claim follows.

Note additionally, that $f(n)$ and $g(n)$ need to have the same sign for all $n \in \mathbb{N}$. Therefore, we can assume that both functions are non-negative.

Let $c > 0$ and $n_0 \in \mathbb{N}$, such that we have for all $n \geq n_0$

$$f(n) \leq n^c \cdot g(n),$$

[9] Take as counterexample, $f_1 = 1$, $f_2 = n$, $g_1 = -1$ and $g_2 = -n^2$.

[10] Pay attention that the multiplicative rule for $\sim$ can only be used a constant number of times. For example, $1 \sim 2$ does not imply $1^n \sim 2^n$.

$$t(n) \leq n^c.$$

Set additionally

$$\begin{aligned} A_0 &= \{n \in \mathbb{N} \mid n \geq n_0, g(t(n)) = 0\}, \\ A_1 &= \{n \in \mathbb{N} \mid n \geq n_0, g(t(n)) \neq 0, t(n) < n_0\}, \\ A_2 &= \{n \in \mathbb{N} \mid n \geq n_0, g(t(n)) \neq 0, t(n) \geq n_0\}. \end{aligned}$$

For all $n \in A_0$, we have $f(t(n)) = 0$.

For $n \in A_1$, we have

$$\frac{f(t(n))}{g(t(n))} \leq \max\left\{ \frac{f(z)}{g(z)} \;\middle|\; z \in \{1, \ldots, n_0 - 1\}, g(z) \neq 0 \right\} \in O(1).$$

Set $Z := \max\left\{ \frac{f(z)}{g(z)} \;\middle|\; z \in \{1, \ldots, n_0 - 1\}, g(z) \neq 0 \right\}$.

For $n \in A_2$, we have

$$\frac{f(t(n))}{g(t(n))} \leq t(n)^c \leq n^{c^2}.$$

Hence, for all $n \geq n_0$, we have

$$f(t(n)) \leq Z \cdot n^{c^2} \cdot g(t(n)).$$

Since $Z \cdot n^{c^2} \in \mathsf{poly}(n)$, the claim follows. □

Lemma 6. *Let there be two functions $a, b : \mathbb{N} \to \mathbb{R}_{>0}$. If a is monotonically increasing, we have*

$$\left(1 + \frac{1}{a(n)}\right)^{b(n)} \sim \exp\left(\Theta\left(\frac{b(n)}{a(n)}\right)\right)$$

and

$$\left(1 - \frac{1}{a(n)}\right)^{b(n)} \sim \exp\left(-\Theta\left(\frac{b(n)}{a(n)}\right)\right)$$

Proof. It is known that $(1 + 1/x)^x$ is monotonically growing and lies in the interval $(1, e)$ for all $x > 0$. Now, consider

$$\left(1 + \frac{1}{a(n)}\right)^{b(n)} = \left(\left(1 + \frac{1}{a(n)}\right)^{a(n)}\right)^{b(n)/a(n)}.$$

Since $a(n) > 0$ is monotonically increasing, $(1+1/a)^a$ is bounded from below by $(1+1/a(0))^{a(0)} \in (1, e)$. Hence, we can deduce

$$\left((1 + 1/a(0))^{a(0)}\right)^{b(n)/a(n)} \leq \left(1 + \frac{1}{a(n)}\right)^{b(n)} \leq e^{b(n)/a(n)}.$$

By taking logarithms, one can see that the first claim follows. The second claim follows analogously. □

Lemma 7. *We have*

$$\binom{n}{h} \sim \frac{n^n}{h^h \cdot (n-h)^{n-h}}.$$

Proof. Note that we have

$$\binom{n}{h} = \frac{n!}{h! \cdot (n-h)!}.$$

Now, according to Stirling's approximation, it holds

$$n \cdot \log(n) - n \cdot \log(e) \leq \log(n!) \leq n \cdot \log(n) - n \cdot \log(e) + d \cdot \log(n)$$

for some constant $d > 0$ where we take the logarithm log in base two. Hence, we can deduce $n! \sim \left(\frac{n}{e}\right)^n$. For the binomial, we get

$$\binom{n}{h} = \frac{n!}{h! \cdot (n-h)!} \sim \left(\frac{n}{e}\right)^n \cdot \left(\frac{e}{h}\right)^h \cdot \left(\frac{e}{n-h}\right)^{n-h} = \frac{n^n}{h^h \cdot (n-h)^{n-h}}.$$ □

We can now give a proof of Lemma 1:

Lemma 1. It suffices to show that

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w}_\mathsf{H}(x) = n/2 + \sqrt{n} \cdot \ell(n)\right] = \binom{n}{n/2 + \sqrt{n} \cdot \ell(n)} \cdot 2^{-n} \geq \frac{\exp(-c \cdot \ell(n)^2)}{p(n)}$$

for some constant $c > 0$ and positive polynomial $p \in \mathsf{poly}(n)$. Set $f(n) = \sqrt{n} \cdot \ell(n)$. According to Lemma 7, we have

$$\begin{aligned}
&\binom{n}{n/2 + \sqrt{n} \cdot \ell(n)} \cdot 2^{-n} \sim \frac{n^n}{(n/2+f)^{n/2+f} \cdot (n/2-f)^{n/2-f}} \cdot 2^{-n} \\
=&\frac{(n/2)^n}{(n/2+f)^{n/2+f} \cdot (n/2-f)^{n/2-f}} = \left(\frac{n/2}{n/2+f}\right)^{n/2+f} \cdot \left(\frac{n/2}{n/2-f}\right)^{n/2-f} \\
=&\left(\frac{(n/2)^2}{(n/2+f) \cdot (n/2-f)}\right)^{n/2} \cdot \left(\frac{n/2-f}{n/2+f}\right)^{f} \\
=&\left(\frac{(n/2)^2}{(n/2)^2 - f^2}\right)^{n/2} \cdot \left(1 - \frac{2f}{n/2+f}\right)^{f} \geq \left(1 - \frac{2f}{n/2+f}\right)^{f}.
\end{aligned}$$

Resubstituting $f(n) = \ell(n) \cdot \sqrt{n}$, we get

$$\begin{aligned}
&\left(1 - \frac{2f}{n/2+f}\right)^{f} = \left(1 - \frac{2\ell\sqrt{n}}{n/2+\ell\sqrt{n}}\right)^{\ell\sqrt{n}} \\
=&\left(1 - \frac{2\ell}{\sqrt{n}/2+\ell}\right)^{\ell\sqrt{n}} = \left(1 - \frac{2\ell^2}{\ell\sqrt{n}/2+\ell^2}\right)^{\ell\sqrt{n}}
\end{aligned}$$

Note that we can bound $\left(1 - \frac{2\ell^2}{\ell\sqrt{n}/2+\ell^2}\right)$ between $\left(1 - \frac{2\ell^2}{\ell\sqrt{n}}\right)$ and $\left(1 - \frac{2\ell^2}{\ell\sqrt{n}/2}\right)$ for n large enough, since $\ell \in o(\sqrt{n})$. According to Lemma 6, we hence get

$$\left(1 - \frac{2f}{n/2+f}\right)^{f} = \left(1 - \frac{2\ell^2}{\ell\sqrt{n}/2+\ell^2}\right)^{\ell\sqrt{n}} \sim \exp(-\Theta(\ell^2)).$$

The claim of the lemma follows now by

$$\begin{aligned}
&\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w}_\mathsf{H}(x) = n/2 + \sqrt{n} \cdot \ell(n)\right] = \binom{n}{n/2 + \sqrt{n} \cdot \ell(n)} \cdot 2^{-n} \\
\sim&\left(\frac{(n/2)^2}{(n/2)^2 - \ell^2 n}\right)^{n/2} \cdot \left(1 - \frac{2\ell\sqrt{(n)}}{n/2+\ell\sqrt{n}}\right)^{\ell\sqrt{n}} \geq \left(1 - \frac{2\ell^2}{\ell\sqrt{n}/2+\ell^2}\right)^{\ell\sqrt{n}} \\
\sim&\exp(-\Theta(\ell^2)).
\end{aligned}$$ □

B Proofs for Section 5

B.1 Proof of balancedness of $f_{d,\epsilon,n}$

Proof. We show that $f_{d,\epsilon,n}$ takes the values 0 and 1 equally often. Pair each $x \in \{0,1\}^n$ with $\bar{x} = (1-x_1, \ldots, 1-x_n)$, choosing x with $\mathsf{w_H}(x) < n/2$. Since n is odd, these pairs form a partition of $\{0,1\}^n$.

If $\mathsf{w_H}(x) < d$, then $f_{d,\epsilon,n}(x) = 0$ and $f_{d,\epsilon,n}(\bar{x}) = 1$, since $\mathsf{w_H}(\bar{x}) = n - \mathsf{w_H}(x) > n - d$. If $\mathsf{w_H}(x) \in [d, n/2[$, then $\mathsf{w_H}(x)$ and $n - \mathsf{w_H}(x)$ have opposite parity, and hence $f_{d,\epsilon,n}(\bar{x}) = 1 - f_{d,\epsilon,n}(x)$. Thus each pair contributes one input on which $f_{d,\epsilon,n}$ equals 0 and one input on which it equals 1. Therefore $f_{d,\epsilon,n}$ is balanced. □

B.2 Proof of the algebraic immunity bound for $f_{d,\epsilon,n}$

Proof. First, we prove by contradiction that $f_{d,\epsilon,n}$ does not admit any non-null annihilator of degree strictly less than d. By definition, $f_{d,\epsilon,n}$ takes the value 1 on all inputs whose Hamming weight lies between $n-d+1$ and n, inclusive. Assume that there exists a non-null function g of degree strictly less than d that annihilates $f_{d,\epsilon,n}$. This implies that $g \cdot \mathsf{THR}_{n,n-d+1} = 0$, which is impossible since $\mathsf{AI}(\mathsf{THR}_{n,n-d+1}) = \min(n-d+1, d) = d$ by Property 1.

Then, we show by contradiction that $1 + f_{d,\epsilon,n}$ does not admit any non-null annihilator of degree strictly less than d. We define the function h by

$$h(x_1, \ldots, x_n) = 1 + f_{d,\epsilon,n}(x_1 + 1, \ldots, x_n + 1).$$

Since this transformation is an affine change of variables, the functions h and $f_{d,\epsilon,n}$ share the same affine-invariant parameters, in particular the algebraic immunity. As $f_{d,\epsilon,n}$ takes the value 0 on all inputs whose Hamming weight lies between 0 and $d-1$, inclusive, the function h takes the value 1 on all inputs whose Hamming weight lies between $n-d+1$ and n, inclusive. Using the same argument as in the first part, the existence of a non-null annihilator of h of degree strictly less than d would imply the existence of a non-null annihilator of degree strictly less than d for the threshold function $\mathsf{THR}_{n,n-d+1}$, which is impossible. Consequently, h does not admit any non-null annihilator of degree strictly less than d, and therefore neither does $1 + f_{d,\epsilon,n}$.

Since neither $f_{d,\epsilon,n}$ nor $1 + f_{d,\epsilon,n}$ admits a non-null annihilator of degree strictly less than d, we conclude that $\mathsf{AI}(f_{d,\epsilon,n}) \geq d$.

□

B.3 Proof of Proposition 6

Proof. First, we observe that the triplet $(a,b,c) = (0,0,0)$ corresponds to the null function, and that $(a,b,c) = (1,1,1)$ corresponds to the constant function 1, which are the only constant Boolean functions.

Since $\mathbf{v}_f$ is periodic with period three, we show that it cannot be periodic with period 2^t for any $2^t < n$. We distinguish cases according to the value of $2^t \bmod 3$.

- If $2^t \equiv 1 \pmod 3$, then $a = v_0 = v_{2^t} = b$ and $b = v_1 = v_{2^t+1} = c$, which implies $a = b = c$.
- If $2^t \equiv 2 \pmod 3$, then $a = v_0 = v_{2^t} = c$ and $b = v_1 = v_{2^t+1} = a$, and hence again $a = b = c$.

Since we assume that f is not constant, $\mathbf{v}_f$ cannot be periodic with period $2^t < n$. Applying Property 2, this implies that $\deg(f) \geq 2^t$. Taking $t = \lfloor \log_2(n-1) \rfloor$, we have $2^t < n-1$, and therefore

$$\deg(f) > 2^{\log_2(n-1)-1} = \frac{n-1}{2}.$$

Since $n \geq 6$, we conclude that $\deg(f) \geq \frac{n}{3}$.

□

B.4 Proof of Proposition 7

Proof. We determine the sets of inputs on which f must take identical values under the stated property.

We begin with the all-zero vector 0_n, which is the unique vector of Hamming weight 0. Since the property holds for any choice of a triplet of indices, starting from 0_n we can reach any vector of Hamming weight 3, and by immediate induction, any vector whose Hamming weight is congruent to 0 modulo three. Accordingly, f takes the same value on all such inputs.

Next, we consider an input of Hamming weight one, namely a canonical vector e_ℓ. The function f takes the same value on e_ℓ and on any vector of Hamming weight four whose ℓ-th coordinate is equal to 1, for instance a vector of the form $e_\ell + e_i + e_j + e_k$, where the four indices are distinct (which is possible since $n \geq 6$). Applying the defining property of f, we conclude that f also takes the same value on e_i. Since e_i was chosen arbitrarily subject only to being different from e_ℓ, it follows that f takes the same value on all vectors of Hamming weight 1. By immediate induction, the same holds for all inputs whose Hamming weight is congruent to 1 modulo three.

Finally, we consider an input of Hamming weight two, say $e_\ell + e_m$. The function f takes the same value on all vectors of the form $e_\ell + e_m + e_i + e_j + e_k$, where the five indices are distinct (again possible since $n \geq 6$). Moreover, f also takes the same value on vectors of the form $e_\ell + e_i$ or $e_i + e_j$, and therefore on all vectors of Hamming weight two. Applying the defining property of f once more, we deduce that f takes the same value on all inputs whose Hamming weight is congruent to 2 modulo three. This concludes the proof.

□

B.5 Proof of the algebraic immunity bound for $\mathsf{MOD3}_n$

Proof. We start by showing that $\mathsf{MOD3}_n$ does not admit any non-null annihilator of degree strictly less than $n/3$. We prove this claim by induction.

First, based on experimental results (see Table 5), we have $\mathsf{AI}(\mathsf{MOD3}_n) \geq n/3$ for all n in the range $[3, 6]$, hence $\mathsf{MOD3}_n$ does not admit non-null annihilator of degree lower than $n/3$ for these values. Assume now, as the induction hypothesis, that $\mathsf{MOD3}_n$ does not admit a non-null annihilator of degree lower than $n/3$. We want to show that this implies that $\mathsf{MOD3}_{n+3}$ does not admit a non-null annihilator of degree lower than $(n+3)/3$.

Let g_{n+3} be an arbitrary annihilator of $\mathsf{MOD3}_{n+3}$. By isolating its dependence on the first three variables, we rewrite $g_{n+3}(x_1, \ldots, x_{n+3})$:

$$\begin{aligned}
&g_{n+3}(x_1, \ldots, x_{n+3}) \\
&= g_{n+3}(x_1, x_2, x_3, y) \\
&= g_0(y) + x_1 g_1(y) + x_2 g_2(y) + x_3 g_3(y) + x_1 x_2 g_{1,2}(y) + x_1 x_3 g_{1,3}(y) \\
&\quad + x_2 x_3 g_{2,3}(y) + x_1 x_2 x_3 g_{1,2,3}(y),
\end{aligned}$$

where the eight functions $g_0, \ldots, g_{1,2,3}$ are n-variable Boolean functions in the variables $y = (x_4, \ldots, x_{n+3})$.

Since g_{n+3} annihilates $\mathsf{MOD3}_{n+3}$, it vanishes on all inputs where $\mathsf{MOD3}_{n+3}$ takes the value 1. First, consider the subspace defined by $x_1 = x_2 = x_3 = 0$. In this case,

$$g_{n+3}(0, 0, 0, y) = g_0(y),$$

and g_0 is an annihilator of $\mathsf{MOD3}_n$. Similarly, on the subspace $x_1 = x_2 = x_3 = 1$, we obtain

$$g_{n+3}(1, 1, 1, y) = g_0(y) + g_1(y) + g_2(y) + g_3(y) + g_{1,2}(y) + g_{1,3}(y) + g_{2,3}(y) + g_{1,2,3}(y),$$

which is also an annihilator of $\mathsf{MOD3}_n$.

Define the Boolean function h_n by

$$h_n(y) = g_{n+3}(0, 0, 0, y) + g_{n+3}(1, 1, 1, y).$$

Since h_n is a linear combination of annihilators of $\mathsf{MOD3}_n$, it is itself an annihilator of $\mathsf{MOD3}_n$. We now distinguish the following cases.

- Assume first that $g_{n+3}(0,0,0,y) \neq g_{n+3}(1,1,1,y)$. Then h_n is non-null, and by the induction hypothesis it has degree at least $n/3$. Since

$$h_n(y) = g_1(y) + g_2(y) + g_3(y) + g_{1,2}(y) + g_{1,3}(y) + g_{2,3}(y) + g_{1,2,3}(y),$$

  at least one of these seven functions must have degree at least $n/3$. Consequently,

$$\deg(g_{n+3}) \geq \deg(h_n) + 1 \geq n/3 + 1.$$

- Assume now that $g_{n+3}(0,0,0,y) = g_{n+3}(1,1,1,y)$, but that there exists another choice of indices (i,j,k) different from $(1,2,3)$ such that the restrictions of g_{n+3} to the subspaces $x_i = x_j = x_k = 0$ and $x_i = x_j = x_k = 1$ are different. In this case, applying the same argument after permuting the variables yields again $\deg(g_{n+3}) \geq n/3 + 1$.

- Finally, assume that for all triplets $(i,j,k) \in [n+3]$, the restrictions of g_{n+3} to the subspaces $x_i = x_j = x_k = 0$ and $x_i = x_j = x_k = 1$ coincide. By Proposition 7, this implies that g_{n+3} is a symmetric Boolean function whose value depends only on the congruence class modulo three of the Hamming weight of its input. By Proposition 6, either g_{n+3} has degree at least $(n+3)/3 = n/3+1$, or it is constant. Since g_{n+3} cannot be the constant function equal to one, it can only be the null function.

In all cases, g_{n+3} is either the trivial annihilator or has degree at least $n/3+1$, which completes the induction step. Therefore, $\mathsf{MOD3}_n$ does not admit any non-null annihilator of degree less than $n/3$.

To conclude the analysis of the algebraic immunity of $\mathsf{MOD3}_n$, it remains to consider annihilators of $1 + \mathsf{MOD3}_n$. We observe that

$$1 + \mathsf{MOD3}_n = 1\mathsf{MOD3}_n + 2\mathsf{MOD3}_n,$$

where $1\mathsf{MOD3}_n$ and $2\mathsf{MOD3}_n$ take the value 1 on disjoint subsets of $\{0,1\}^n$. Hence, any annihilator of $1 + \mathsf{MOD3}_n$ must simultaneously annihilate $1\mathsf{MOD3}_n$ and $2\mathsf{MOD3}_n$.

Using the same inductive argument as above, replacing $\mathsf{MOD3}_n$ by $1\mathsf{MOD3}_n$, we show that $1\mathsf{MOD3}_n$ does not admit any non-null annihilator of degree strictly less than $n/3$. The base cases are supported by experimental results (see Table 6 for $n \in [3,15]$), and the induction step again relies on Propositions 7 and 6. It follows that $1 + \mathsf{MOD3}_n$ also does not admit any non-null annihilator of degree strictly less than $n/3$. Consequently, $\mathsf{AI}(\mathsf{MOD3}_n) \geq n/3$.

□

B.6 Alternative View of Common Bias and Generalization

The common bias introduced in Section 3 can also be interpreted as the probability that several Boolean functions simultaneously agree with constant functions. More generally, one can consider their simultaneous agreement with functions of bounded algebraic degree. We briefly introduce this generalization here, with particular attention to degree one, which still leads to a system of linear equations.

The average-case common bias of P can equivalently be written as

$$\mathsf{c\text{-}bias}^m_{\mathrm{ac}}(P) = \max_{\substack{(g_1,\ldots,g_m):\mathsf{supp}(\mathcal{D}_P)\to(\mathcal{B}_a)^m \\ \deg(g_i)=0,\ \forall i}} \Pr_{\mathcal{F}\leftarrow\mathcal{D}_P,\ x\leftarrow\{0,1\}^n} \left[P_i(x) = g_i(\mathcal{F})(x),\ \forall i \in [m]\right].$$

This formulation views common bias as the simultaneous approximation of the m applications of P by constant Boolean functions, it suggests the following generalization.

Definition 14 (Average-Case Degree-d Common Bias)**.** Let $P : \{0,1\}^a \to \{0,1\}$, and let $\mathcal{D}_P$ be the uniform distribution of P-based functions $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$. For $\mathcal{F} \leftarrow \mathcal{D}_P$, let $P_1, \ldots, P_m$ denote the m applications of P defining $\mathcal{F}$. The average-case degree-d common bias of P is

$$\mathsf{c\text{-}bias}^{m,d}_{\mathrm{ac}}(P) := \max_{\substack{(g_1,\ldots,g_m):\mathsf{supp}(\mathcal{D}_P)\to(\mathcal{B}_a)^m \\ \deg(g_i)\leq d,\ \forall i}} \Pr_{\mathcal{F}\leftarrow\mathcal{D}_P,\ x\leftarrow\{0,1\}^n} \left[P_i(x) = g_i(\mathcal{F})(x),\ \forall i \in [m]\right].$$

The original average-case common bias is precisely the average-case degree-0 common bias. The degree-1 case is of particular interest for the attacks considered in this work. Replacing the applications of the predicate P by affine functions still results in a system of linear equations, and therefore leads to an attack with the same asymptotic complexity as in the degree-0 case. Moreover, for a Boolean function the quality of its best affine approximation is captured by its nonlinearity. When a best affine approximation of P is a constant function, considering degree-1 common bias does not improve the attack obtained from degree-0 common bias. This is the case for the majority predicates considered in the main part of this work. Hence, for these predicates, the degree-1 generalization gives the same attack as the common-bias attack analyzed in the main part.

For the interpolation predicates $f_{d,\varepsilon,a}$ introduced in Section 5, the situation can be different. By construction, these functions coincide with XOR_a or $1 + \mathsf{XOR}_a$ on a large fraction of their inputs. Consequently, XOR_a or $1 + \mathsf{XOR}_a$ can provide a substantially better degree-1 approximation than either constant function. Some parameters can therefore satisfy the common-bias criterion considered in the main part of the paper while being vulnerable to its degree-1 generalization.

For example, consider the parameter $n = 894$ based on $f_{29,0,71} \oplus \mathsf{XOR}_{17}$. This parameter satisfies the degree-0 common-bias criterion for a 128-bit security target. However, $f_{29,0,71}$ disagrees with XOR_{71} on only a fraction

$$\frac{2\sum_{i=0}^{28}\binom{71}{i}}{2^{71}} \approx 0.03612$$

of its inputs. Thus, replacing $f_{29,0,71}$ by XOR_{71} gives an affine approximation which is correct with probability approximately 0.96388 for each equation. In particular, the probability that this approximation is correct simultaneously for $n = 894$ equations is at least

$$(1 - 0.03612)^{894} \approx 2^{-47.45},$$

which is well above the 2^{-128} target. We therefore do not include this parameter, nor the other parameters vulnerable to this degree-1 generalization, among the challenges proposed in Table 9.

C Distinguishing Attacks

While our inversion Algorithm 2 performs well for concrete parameters, it needs a logarithmic multiplicative stretch to guarantee that the underlying linear equation system has a rank deficit of at most one.

Therefore, we present in Algorithm 3 a distinguisher for $\mathsf{THR} \oplus \mathsf{XOR}$-based PRGs $\mathcal{F} : \{0,1\}^n \to \{0,1\}^{n+\lambda}$, which only needs a small additive stretch of $\lambda \in o(n)$.

The following lemma bounds the advantage and running time of Algorithm 3:

Lemma 8. *Algorithm 3 has a time complexity of $O((n+\lambda)n^2)$. In the average case, its advantage is lower-bounded by $\mathsf{c\text{-}bias}_{\mathrm{ac}}(\mathsf{THR}_{a,t}) - 2^{-\lambda}$, in the worst case, by $\mathsf{c\text{-}bias}_{\mathrm{wc}}(\mathsf{THR}_{a,t}) - 2^{-\lambda}$.*

Proof. The time complexity of Algorithm 3 is dominated by using Gaussian elimination to compute a permutation matrix $\Pi \in \mathbb{Z}_2^{(n+\lambda)\times(n+\lambda)}$ and an invertible lower triangular matrix $T \in \mathbb{Z}_2^{(n+\lambda)\times(n+\lambda)}$ such that $T \cdot \Pi \cdot L$ is in upper triangular form.[11] Note that we only need to compute the first n columns of T and can assume that it is of shape

$$T = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} \overline{T} & 0 \\ \underline{T} & I_\lambda \end{pmatrix}$$

where $\overline{T} \in \mathbb{Z}_2^{n\times n}$ is in lower triangular shape, and $I_\lambda \in \mathbb{Z}_2^{\lambda\times\lambda}$ is the identity matrix. Hence, we only need to store $(n+\lambda)n$ entries for T and can bound the asymptotic cost of computing it by $O((n+\lambda)n^2)$.

[11] The reason why such T, Π exist is that any matrix can be decomposed into the product of a permutation matrix, a lower triangular matrix and an upper triangular matrix, in this order. In fact, this is what Gaussian elimination implicitly computes by swapping rows and applying elementary row operations.

Algorithm 3: Distinguishing algorithm for $\mathsf{THR} \oplus \mathsf{XOR}$-based PRGs.

Data: A PRG $\mathcal{F} = \mathcal{T} \oplus \mathcal{L} : \{0,1\}^n \to \{0,1\}^{n+\lambda}$ where $\mathcal{T} = (\mathsf{THR}_{A_1,t}, \ldots, \mathsf{THR}_{A_{n+\lambda},t})$ and $\mathcal{L} = (\mathsf{XOR}_{B_1}, \ldots, \mathsf{XOR}_{B_{n+\lambda}})$. A potential output $y \in \{0,1\}^{n+\lambda}$.

Result: Decision bit $\beta \in \{0,1\}$.

Set $a := \#A_1$;
if $t \leq (a+1)/2$ **then**
  Set $z := (1, \ldots, 1) \in \{0,1\}^{n+\lambda}$;
else
  Set $z := (0, \ldots, 0) \in \{0,1\}^{n+\lambda}$;
end
for $i \in [n+\lambda]$ **do**
  for $j \in [n]$ **do**
    if $j \in B_i$ **then**
      Set $l_{i,j} := 1$;
    else
      Set $l_{i,j} := 0$; `/* L = (l_{i,j}) ∈ Z_2^{(n+λ)×n} is a matrix representation of L */`
    end
  end
end
Compute a permutation matrix $\Pi \in \mathbb{Z}_2^{(n+\lambda)\times(n+\lambda)}$ and a lower triangular matrix $T \in \mathbb{Z}_2^{(n+\lambda)\times(n+\lambda)}$ such that $T \cdot \Pi \cdot L$ is an upper triangular matrix; `// We use Gaussian elimination here.`
Decompose $\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} := T$ with $T_2 \in \mathbb{Z}_2^{\lambda\times(\lambda+n)}$;
Set $e := T_2 \cdot \Pi \cdot (y - z) \in \mathbb{Z}_2^{\lambda}$;
if $e = (0, \ldots, 0)$ **then**
  Return 0;
else
  Return 1;
end

For the advantage, we distinguish two cases: If $y = \mathcal{F}(x)$ for some $x \in \mathbb{Z}_2^n$, the algorithm will output 0 if

$$\mathsf{THR}_{A_1,t}(x) = \cdots = \mathsf{THR}_{A_{n+\lambda},t}(x) = 1$$

for $t \leq (a+1)/2$ (all THR-predicates are biased towards 1), or if

$$\mathsf{THR}_{A_1,t}(x) = \cdots = \mathsf{THR}_{A_{n+\lambda},t}(x) = 0$$

for $t > (a+1)/2$ (all THR-predicates are biased towards 0). Indeed, let us assume that we have $\mathcal{T}(x) = (\mathsf{THR}_{A_1,t}(x), \ldots, \mathsf{THR}_{A_{n+\lambda},t}(x)) = z$. Then, we can rewrite the equation system $\mathcal{F}(x) = y$ as

$$L \cdot x = \mathcal{L}(x) = y - \mathcal{T}(x) = y - z.$$

In particular, we get

$$\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} \cdot \Pi \cdot (y - z) = T \cdot \Pi \cdot L \cdot x = (T\Pi L)x.$$

Since $T\Pi L$ is an upper triangular matrix of rectangular shape $(n+\lambda) \times n$, the last λ entries of $(T\Pi L)x$, which are exactly $T_2\Pi(y-z)$, must be zero. Hence, the equality check $T_2\Pi(y-z) = 0$ passes if $y = \mathcal{F}(x)$ and all THR-predicates are 1 resp. 0 (depending on t and a). The probability of all THR-predicates

being 1 resp. 0 is lower bounded by the average-case common bias $\mathsf{c\text{-}bias}_{\mathrm{ac}}^{n+\lambda}(\mathsf{THR}_{a,t})$, if $A_1, \ldots, A_{n+\lambda}$ are distributed uniformly at random, and by the worst-case common bias $\mathsf{c\text{-}bias}_{\mathrm{wc}}^{n+\lambda}(\mathsf{THR}_{a,t})$, in general. Hence, the algorithm returns 0 if $\mathcal{F}(x) = y$ with probability at least $\mathsf{c\text{-}bias}_{\mathrm{ac}}^{n+\lambda}(\mathsf{THR}_{a,t})$ resp. $\mathsf{c\text{-}bias}_{\mathrm{wc}}^{n+\lambda}(\mathsf{THR}_{a,t})$.

In case, $y \leftarrow \mathbb{Z}_2^{n+\lambda}$ is distributed uniformly at random, the check $T_2\Pi(y-z) = 0$ will only hold with probability at most $2^{-\lambda}$. Indeed, T_2 contains the identity matrix of size λ, hence, $T_2\Pi \in \mathbb{Z}_2^{\lambda\times(n+\lambda)}$ has rank λ. Since $y - z$ is uniformly random, the vector $e = T_2\Pi(y-z)$ will be a random bit string of length λ. It follows that Algorithm 3 outputs 1 with probability $1 - 2^{-\lambda}$ if $y \leftarrow \mathbb{Z}_2^{n+\lambda}$.

In conclusion, we get an advantage of at least

$$\left|(1 - \mathsf{c\text{-}bias}_{\mathrm{ac}}^{n+\lambda}(\mathsf{THR}_{a,t})) - (1-2^{-\lambda})\right| \geq \mathsf{c\text{-}bias}_{\mathrm{ac}}^{n+\lambda}(\mathsf{THR}_{a,t}) - 2^{-\lambda}$$

resp. $\mathsf{c\text{-}bias}_{\mathrm{wc}}^{n+\lambda}(\mathsf{THR}_{a,t}) - 2^{-\lambda}$. □

As for the inversion Algorithm 2, the advantage of Algorithm 3 is directly linked to the common bias of THR-predicates. By Corollary 1, we therefore get the following:

Corollary 3. *The advantage of Algorithm 3 is lower bounded by*

$$(2n+2\lambda)^{-\Theta(n/a)} - 2^{-\lambda}.$$

D On PRGs with Balanced Seeds

We note that fixing the Hamming weight of the seed is not a common practice with Goldreich's PRG, but it is with concrete stream ciphers following a similar structure. For the stream cipher FLIP [MJSC16], where the same predicate of arity n is applied on (pseudorandom) permutations of keys for each output bit, the Hamming weight of keys is fixed to $n/2$ to avoid weak keys. The same restriction applies to the keys of the stream cipher FiLIP [MCJS19], where each output bit is computed by applying the same predicate of arity a on pseudorandom subsets of (size-n) keys xored with pseudorandom binary strings of size a.

Given our results, one might think that our attacks only depend on the challenger in the PRG-security game rolling a *weak* seed, and that simply hardening the seeds somehow might avert our attacks. This might indeed increase the bit security, however, if the PRG has a huge stretch as in the candidates of [BCM+24, BCPR25], our attack is still applicable, even if all seeds are conditioned on having Hamming weight exactly $n/2$.

In this section, let us sketch the case where all seeds $x \leftarrow \{0,1\}^n$ for some $\mathsf{THR}_{a,t} \oplus \mathsf{XOR}_b$-based PRG $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$ are uniformly random conditioned on their Hamming weight being perfectly balanced, i.e., $\mathsf{w_H}(x) = n/2$ (we assume that n is even for now). One can achieve this by generating a random permutation $\pi : [n] \to [n]$ and applying it to $0^{n/2}1^{n/2} \in \{0,1\}^n$.

To prepare Algorithm 2 for PRGs with such seed distributions, we compute a parameter n' such that

$$\frac{\binom{n'}{a}}{\binom{n}{a}} \cdot m \approx n.$$

Note that $\frac{\binom{n'}{a}}{\binom{n}{a}} \cdot m$ is the expected number of threshold predicates $\mathsf{THR}_{A_j,t}$ such that A_j lies in $[n']$. Hence, we can assume with a decent probability that there is a sufficient number of output bits whose threshold predicates just depend on the first n' input bits.[12] The next lemma shows us that the Hamming-weight of the first n' bits of x is distributed hypergeometrically:

Lemma 9. *For n even, $0 < n' < n$ and $n' - n/2 \leq h \leq \min\{n/2, n'\}$, we have*

$$\Pr_{x\leftarrow\{0,1\}^n}[\mathsf{w_H}(x_1,\ldots,x_{n'}) = h \mid \mathsf{w_H}(x) = n/2] = \frac{\binom{n/2}{h}\cdot\binom{n/2}{n'-h}}{\binom{n}{n'}} = \mathrm{hyg}(n, n/2, n', h).$$

[12] For most parameters of Table 1, we will have approximately $n' \approx n \cdot 3/4$.

Proof. Let x be uniformly random conditioned on $\mathsf{w_H}(x) = n/2$. As mentioned earlier, we can imagine that x is a randomly permuted version of $0^{n/2}1^{n/2}$. When we apply a random permutation to $0^{n/2}1^{n/2}$, the corresponding indices of $0^{n/2}1^{n/2}$ that fall into $[n']$ become a random subset A of $[n]$ of size n'. From this random subsets A, we want to count the ones where $n'-h$ elements fall into $[n/2]$ and h entries fall into $\{n/2+1,\ldots,n\}$, since those are the ones that yield a Hamming weight of exactly h for the prefix $(x_1,\ldots,x_{n'})$. Hence, the probability that $(x_1,\ldots,x_{n'})$ has Hamming weight h if x is uniformly random conditioned on $\mathsf{w_H}(x) = n/2$ is given by $\frac{\binom{n/2}{h}\cdot\binom{n/2}{n'-h}}{\binom{n}{n'}}$. □

For the common bias of predicates $\mathsf{THR}_{A_{i_1},t},\ldots,\mathsf{THR}_{A_{i_{m'}},t}$ that all just depend on $(x_1,\ldots,x_{n'})$ (which are expected to exist), we now get:

Corollary 4. *Let $t \le a \le n' \le n$. Define the* restricted common bias *of $\mathsf{THR}_{a,t}$ by*

$$\begin{aligned}&\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'|n'}(\mathsf{THR}_{a,t})\\ :=&\Pr_{A_1,\ldots,A_{m'}\leftarrow\binom{[n']}{a},x\leftarrow\{0,1\}^n}\left[\forall i\in[m']:\ \mathsf{THR}_{A_i,t}(x)=1 \mid \mathsf{w_H}(x)=n/2\right].\end{aligned}$$

We have

$$\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'|n'}(\mathsf{THR}_{a,t}) = \sum_{h=0}^{n/2}\mathrm{hyg}(n,n/2,n',h)\cdot q(n',h,a,t)^{m'}$$

where

$$q(n',h,a,t) = \Pr_{X\leftarrow\mathrm{HyG}(n',h,a)}[X\ge t] = \sum_{i=t}^{a}\frac{\binom{h}{i}\cdot\binom{n-h}{a-i}}{\binom{n'}{a}}.$$

Proof. Using Lemma 9, we have

$$\begin{aligned}&\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'|n'}(\mathsf{THR}_{a,t})\\ =&\Pr_{A_1,\ldots,A_{m'}\leftarrow\binom{[n']}{a},x\leftarrow\{0,1\}^n}\left[\forall i\in[m']:\ \mathsf{THR}_{A_i,t}(x)=1 \mid \mathsf{w_H}(x)=n/2\right]\\ =&\sum_{h=0}^{n/2}\Pr_{x\leftarrow\{0,1\}^n}\left[\mathsf{w_H}(x_1,\ldots,x_{n'})=h \mid \mathsf{w_H}(x)=n/2\right]\\ &\cdot\Pr_{A_1,\ldots,A_{m'}\leftarrow\binom{[n']}{a}}\left[\forall i\in[m']:\ \mathsf{THR}_{A_i,t}(1^h0^{n'-h})=1\right]\\ =&\sum_{h=0}^{n/2}\frac{\binom{n/2}{h}\cdot\binom{n/2}{n'-h}}{\binom{n}{n'}}\cdot q(n',h,a,t)^{m'}.\end{aligned}$$

□

If $\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'|n'}(\mathsf{THR}_{a,t})$ is large enough, we can again apply Algorithm 2 to the output bits with threshold predicates $\mathsf{THR}_{A_{i_1},t},\ldots,\mathsf{THR}_{A_{i_{m'}},t}$ of the PRG $\mathcal{F}$. In Table 10, we give the corresponding advantages and bit-securities for the parameters of [BCM+24, BCPR25]. In comparison to Table 1, the bit-securities implied by our attack significantly increased. However, we can still break the 128-bits security of the first 2 parameter sets. A main performance loss for our attack stems from the additional runtime $O(ma)$, as we now need to go through all m threshold predicates and select the ones whose bits lie in $[n']$.

D.1 Asymptotic Bounds

From an asymptotic point of view, as long as m is large enough to guarantee m' threshold predicates over $x_1,\ldots,x_{n'}$, the asymptotic bound for the common bias does not change.

n	m	n'	m'	T	$\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'\mid n'}$	$\log_2(T/\mathrm{adv})$
256	$2^{40.00}$	191	256	$2^{46.00}$	$2^{-28.86}$	77.86
512	$2^{40.50}$	374	512	$2^{46.50}$	$2^{-63.09}$	112.59
894	$2^{33.33}$	701	894	$2^{39.34}$	$2^{-144.99}$	187.33
2048	$2^{33.00}$	1621	2048	$2^{39.02}$	$2^{-356.14}$	398.16

Table 10: Performances of Algorithm 2 on the parameters of [BCM+24, CDD+24, BCPR25] for distinguishing a structured-seed PRG $\mathcal{F} = \mathsf{THR}_{64,32} \oplus \mathsf{XOR}_{10} : \{0,1\}^n \to \{0,1\}^m$ where all seeds are guaranteed to have a Hamming-weight of $n/2$.

Lemma 10. *Let $n' \leq n$ such that both $n', n-n' \in \Theta(n)$ are linear in n. We have*

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w_H}(x_1,\ldots,x_{n'}) = h \mid \mathsf{w_H}(x) = n/2\right] \geq n^{-\Theta(n/a)}$$

for $h' = \left\lceil n'/2 + n' \cdot \sqrt{\frac{\ln(2m')}{2a}} \right\rceil$ and $m' \in \mathsf{poly}(n')$, $m' \geq n'$.

Proof. We use here the notation and tools of Section A. Set $n'' = n - n'$ and $f = h - n'/2$. We have

$$\begin{aligned}
&\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{w_H}(x_1,\ldots,x_{n'}) = h \mid \mathsf{w_H}(x) = n/2\right] = \frac{\binom{n/2}{h}\cdot\binom{n/2}{n'-h}}{\binom{n}{n'}} \\
\sim&\frac{(n/2)^{n/2}}{h^h\cdot(n/2-h)^{n/2-h}}\cdot\frac{(n/2)^{n/2}}{(n'-h)^{n'-h}\cdot(n/2-n'+h)^{n/2-n'+h}}\cdot\frac{n'^{n'}\cdot(n-n')^{n-n'}}{n^n} \\
=&\frac{1}{h^h\cdot(n/2-h)^{n/2-h}}\cdot\frac{1}{(n'-h)^{n'-h}\cdot(n/2-n'+h)^{n/2-n'+h}}\cdot\frac{n'^{n'}\cdot(n-n')^{n-n'}}{2^n}.
\end{aligned}$$

As a next step, we substitute

$$\begin{aligned}
h &= n'/2 + f, \\
n/2 - h &= n''/2 - f, \\
n' - h &= n'/2 - f, \\
n/2 - n' + h &= n''/2 + f, \\
n - n' &= n'', \\
n &= n' + n'',
\end{aligned}$$

which yields

$$\begin{aligned}
&\frac{1}{h^h\cdot(n/2-h)^{n/2-h}}\cdot\frac{1}{(n'-h)^{n'-h}\cdot(n/2-n'+h)^{n/2-n'+h}}\cdot\frac{n'^{n'}\cdot(n-n')^{n-n'}}{2^n} \\
=&\frac{1}{(n'/2+f)^{n'/2+f}\cdot(n''/2-f)^{n''/2-f}}\cdot\frac{1}{(n'/2-f)^{n'/2-f}\cdot(n''/2+f)^{n''/2+f}} \\
&\cdot\frac{n'^{n'}\cdot n''^{n''}}{2^{n'}\cdot 2^{n''}} \\
=&\left(\frac{(n'/2)^2}{(n'/2+f)\cdot(n'/2-f)}\right)^{n'/2}\cdot\left(\frac{n'/2-f}{n'/2+f}\right)^f \\
&\cdot\left(\frac{(n''/2)^2}{(n''/2+f)\cdot(n''/2-f)}\right)^{n''/2}\cdot\left(\frac{n''/2-f}{n''/2+f}\right)^f \\
=&\left(\frac{(n'/2)^2}{(n'/2)^2-f^2}\right)^{n'/2}\cdot\left(\frac{n'/2-f}{n'/2+f}\right)^f\cdot\left(\frac{(n''/2)^2}{(n''/2)^2-f^2}\right)^{n''/2}\cdot\left(\frac{n''/2-f}{n''/2+f}\right)^f.
\end{aligned}$$

The products $\left(\frac{(n'/2)^2}{(n'/2)^2-f^2}\right)^{n'/2}$ and $\left(\frac{(n''/2)^2}{(n''/2)^2-f^2}\right)^{n''/2}$ are larger than 1, hence, we can omit them. It suffices to show that $\left(\frac{n'/2-f}{n'/2+f}\right)^f$ and $\left(\frac{n''/2-f}{n''/2+f}\right)^f$ are lower-bounded by $\exp(-\Theta(\ln(n')\cdot n'/a))$. Indeed, we have

$$\begin{aligned}\left(\frac{n'/2-f}{n'/2+f}\right)^f &= \left(1-2\frac{f}{n'/2+f}\right)^f \\ &= \exp\left(-\Theta\left(\frac{f^2}{n'/2+f}\right)\right) \\ &= \exp\left(-\Theta\left(\frac{f^2}{n'}\right)\right) \\ &= \exp\left(-\Theta\left(\frac{\left(n'\cdot\sqrt{\frac{\ln(2m')}{2a}}\right)^2}{n'}\right)\right) \\ &= \exp\left(-\Theta\left(\frac{n'\cdot\ln(n')}{a}\right)\right) = n'^{-\Theta(n'/a)},\end{aligned}$$

since $n' \in \Theta(n)$, $m' \in \mathsf{poly}(n')$, $m' \geq n'$ and $f = \Theta(n'\cdot\sqrt{\ln(m')/(2a)})$. Mutatis mutandis, we get $\left(\frac{n''/2-f}{n''/2+f}\right)^f = \exp(-\Theta(\ln(n')\cdot\frac{n'}{a}))$. Hence, the claim follows. □

Corollary 5. *Let $a \leq n' \leq n$ such that $n', n-n' \in \Theta(n)$. Let $m' \in \mathsf{poly}(n')$ such that $m' \geq n'$. Assume $t \leq (a+1)/2$. We have*

$$\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'|n'}(\mathsf{THR}_{a,t}) \geq n^{-\Theta(n/a)}.$$

Proof. We will only sketch the calculations here. So, for simplicity, let us make the mild approximation of $\frac{t}{a} \approx \frac{1}{2}$. Set

$$h' = \left\lceil n'/2 + n'\cdot\sqrt{\frac{\ln(2m')}{2a}}\right\rceil.$$

Because of Lemma 10, we have

$$\Pr_{x\leftarrow\{0,1\}^n}\left[\mathsf{w_H}(x_1,\ldots,x_{n'}) = h' \mid \mathsf{w_H}(x) = n/2\right] \geq n^{-\Theta(n/a)}.$$

Lemma 2 implies on the other hand

$$q(n',h',a,t)^{m'} \geq m'\cdot q(n',h',a,t) - (m'-1) \geq \frac{1}{2}.$$

Combined, we get

$$\begin{aligned}&\mathsf{c\text{-}bias}_{\mathrm{ac}}^{m'|n'}(\mathsf{THR}_{a,t}) \\ =&\sum_{h=0}^{n/2}\Pr_{x\leftarrow\{0,1\}^n}\left[\mathsf{w_H}(x_1,\ldots,x_{n'}) = h \mid \mathsf{w_H}(x) = n/2\right]\cdot q(n',h,a,t)^{m'} \\ \geq&\Pr_{x\leftarrow\{0,1\}^n}\left[\mathsf{w_H}(x_1,\ldots,x_{n'}) = h' \mid \mathsf{w_H}(x) = n/2\right]\cdot q(n',h',a,t)^{m'} \\ \geq& n^{-\Theta(n/a)}\cdot\frac{1}{2}.\end{aligned}$$

□

To guarantee the existence of m' threshold predicates that only depend on the first n' input bits we need a huge stretch m. In fact, since $n/2 \ll n' \ll n$ the stretch

$$m = \frac{\binom{n}{a}}{\binom{n'}{a}}\cdot m' \approx (n/n')^a\cdot m'$$

Algorithm 4: Selection algorithm

Data: Integers $m > n > a$ and sets $A_1, \dots, A_m \subset [n]$ of size a. An integer L.
Result: Set of indices $I \subset [m]$ such that $\#I \geq L$ and $\#\bigcap_{i \in I} A_i \geq S_L(m, n, a)$.

```
Set I = {1, ..., m};                          // set of the indices we are considering
Set J = ∅;          // set of the common intersection of the sets we are considering
while L ≤ ⌈ma/n⌉ do
    Set count = (0, ..., 0) ∈ Z^n;
    for i ∈ I do
        for j ∈ A_i \ J do
            count[j] = count[j] + 1;
        end
    end
    Find j ∈ [n] \ J such that count[j] is maximum;
    Set I = {i ∈ I | j ∈ A_i};
    Set J = J ∪ {j};
    Set m = #I;
    Set n = n − 1;
    Set a = a − 1;
end
Return I;
```

is exponential in a. Hence, for linear $a \in \Theta(n)$, where Algorithm 2 works best, we need an exponential number of output bits. For smaller arities, we can relax this. For example, if $a \in \sqrt{n}$, then we only need $m = \exp(\Theta(\sqrt{n}))$ output bits and have an advantage of at least $n^{-\Theta(\sqrt{n})}$.

E Select&Guess&Solve Attacks

Given a PRG $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m$, we can prepend a selection phase to Algorithm 2 and Algorithm 3, in which we use Algorithm 4 to select a number of output bits to increase the common bias. Heuristically[13], this yields a higher advantage, at the cost of only a slight increase in the algorithm's time complexity.

Lemma 11. *Let $m > n > a$ be fixed integers. Let $A_1, \dots, A_m \subset [n]$ be sets of size a. There exists $I \subset [m]$ of size at least $\lceil ma/n \rceil$ such that $\bigcap_{i \in I} A_i \neq \emptyset$.*

Proof. Let $S = \{(i, j) \in [m] \times [n] \mid j \in A_i\}$. For every $i \in [m]$, let

$$S_i = \{j \in [n] \mid (i, j) \in S\}.$$

For every $j \in [n]$, let $S^j = \{i \in [m] \mid (i, j) \in S\}$. We have $\#S = \sum_{i \in [m]} \#S_i = ma$.

Let $j^* = \arg\max_{j \in [n]} \#S^j$, and let $L = \#S^{j^*}$. We have $ma = \#S = \sum_{j \in [n]} \#S^j \leq Ln$. Thus, $L \geq ma/n$ and S^{j^*} is the set we are looking for. □

Definition 15. Given the integers L, m, n, a, recursively define $S_L(m, n, a)$ as

$$S_L(m, n, a) := \begin{cases} 1 + S_L(\lceil ma/n \rceil, n - 1, a - 1), & \text{if } L < \lceil ma/n \rceil, \\ 0, & \text{otherwise.} \end{cases}$$

Lemma 12. *Algorithm 4 returns a set of indices $I \subset [m]$ such that $\#I \geq L$ and $\#\bigcap_{i \in I} A_i \geq S_L(m, n, a)$. Its time complexity is $O(ma)$.*

[13] When selecting threshold predicates with maximum correlation, the selected threshold predicates are not sampled uniformly random, anymore. Therefore, our bounds on the common bias are not applicable. For reporting concrete numbers, we will make the heuristic that the resulting common bias is close to the average common bias on appropriately sampled threshold predicates.

Proof. Call the input parameters L, m^*, n^*, a^* and call m, n, a the respective variables at any point of execution of the algorithm. The following invariants hold after the execution of line 12 in Algorithm 4:

$$\#I \geq \left\lceil \frac{ma}{n} \right\rceil \geq L, \tag{2}$$

$$\bigcap_{i \in I} A_i \supseteq J.$$

Invariant 2 holds because of Lemma 11, as at each iteration the value of a represents the size of the sets we're considering minus the common intersection we already found. After each execution of the body of the while loop in lines 3-16, the size of J increases by 1. The body of the while loop is executed at least $S_L(m^*, n^*, a^*)$ times because of the way we recursively defined S_L. Thus, after the last execution of the body of the while loop we will have $\#\bigcap_{i \in I} A_i \geq \#J \geq S_L(m^*, n^*, a^*)$.

For the time complexity, the cost of one iteration of the body of the while loop is dominated by $O(ma)$. The cost is thus upper bounded by

$$m^*a^* + m^*a^*\frac{a^*}{n^*} + \ldots + m^*a^*\left(\frac{a^*}{n^*}\right)^{s-1} \leq m^*a^* \sum_{k=0}^{\infty} \left(\frac{a^*}{n^*}\right)^k = O(m^*a^*). \qquad \square$$

E.1 Heuristic Advantage Bound

Let $I \subseteq [m]$ be the output of Algorithm 4 such that $J = \bigcap_{i \in I} A_i$ has at least s elements and $\#I = L$. We can compute the common bias of THR_{A_i}, $i \in I$ by

$$\begin{aligned}
&\Pr_x[\mathsf{THR}_{A_i,t}(x) = 1 \forall i \in I] \\
=&\sum_{h=0}^{s} \Pr\left[\mathsf{w_H}(x_J) = h\right] \cdot \Pr_x[\mathsf{THR}_{A_i,t}(x) = 1 \ \forall i \in I \mid \mathsf{w_H}(x_J) = h] \\
=&\sum_{h=0}^{s} \Pr\left[\mathsf{w_H}(x_J) = h\right] \cdot \Pr_x[\mathsf{THR}_{A_i \setminus J, t-h}(x) = 1 \ \forall i \in I].
\end{aligned}$$

The distribution $A_i \setminus J$ is uniformly random in $\binom{[n]\setminus J}{a-s}$, however, the collection of subsets $(A_i \setminus J)_{i \in I}$ is not statistically independent.Therefore, we cannot decompose $\Pr_x[\mathsf{THR}_{A_i \setminus J, t-h}(x) = 1 \ \forall i \in I]$ into a product, and we cannot replace this term by average-case common bias.

Heuristic: To approximate the common bias of THR_{A_i}, $i \in I$, we pretend that the sets $A_i \setminus J$, $i \in I$, were distributed independently in $\binom{[n]\setminus J}{a-s}$.[14] By doing so, we can approximate their common bias by

$$\begin{aligned}
&\Pr_x[\mathsf{THR}_{A_i,t}(x) = 1 \forall i \in I] \\
=&\sum_{h=0}^{s} \Pr\left[\mathsf{w_H}(x_J) = h\right] \cdot \Pr_x[\mathsf{THR}_{A_i \setminus J, t-h}(x) = 1 \ \forall i \in I] \\
\approx&\sum_{h=0}^{s} \Pr\left[\mathsf{w_H}(x_J) = h\right] \cdot \mathsf{c\text{-}bias}^L_{\mathrm{ac}}(\mathsf{THR}_{a-s,t-h}) \\
=&2^{-s} \cdot \sum_{h=0}^{s} \binom{s}{h} \cdot \mathsf{c\text{-}bias}^L_{\mathrm{ac}}(\mathsf{THR}_{a-s,t-h}).
\end{aligned}$$

In Table 11, we report the resulting advantages and bit securities.

[14] Note that this is really an approximation. If we sample 2 subsets of size 2 from [3] and remove one common element i, we get 2 singletons from $[3] \setminus \{i\}$. The probability of those 2 singletons to be identical is 1/3. However, if we had sampled to elements from $[3] \setminus \{i\}$ uniformly at random, they would be equal with probability 1/2. Hence, the resulting subsets are not distributed uniformly random in $[3] \setminus \{i\}$.

n	m	L	s	T	adv	$\log 2(T/\text{adv})$
256	$2^{40.00}$	256	15	$2^{46.00}$	$2^{-13.17}$	59.17
512	$2^{40.50}$	512	10	$2^{46.50}$	$2^{-22.56}$	69.06
894	$2^{33.33}$	894	6	$2^{39.34}$	$2^{-42.03}$	81.36
2048	$2^{33.00}$	2048	4	$2^{39.02}$	$2^{-102.92}$	141.94

Table 11: Advantage and runtimes of Algorithm 4 together with Algorithm 2. Note that the advantages and bit securities are approximated under an independence-heuristic.

Remark 6 (On Concrete Runtime Claims). For the numbers reported in Table 11, we calculate the runtime of Algorithms 2 and 4 as $T_2 = n^3 + ma$. For the advantage, we take $s = S_n(m, n, a)$ and bound it as $\text{adv} = 2^{-3} \cdot 2^{-s} \cdot \sum_{h=0}^{s} \binom{s}{h} \cdot p(n + \lambda, n - s, a - s, t - h)$, where the factor 2^{-3} again lower bounds the probability of a random sparse square binary matrix to have corank at most one. The performance of Algorithms 2 and 4 is given as T/adv.

F On the Common Bias of First-Past-the-Post Voting

We illustrate here that the concept of the common bias has more general applications beyond cryptography and cryptanalysis. As example, we consider the *First-Past-the-Post* (FPP) voting system[15]. In FPP, a vote is conducted among n voters by splitting all voters into m disjoint districts $A_1, \ldots, A_m \subset \{1, \ldots, n\}$. After all voters cast their vote, let's say $x_1, \ldots, x_n$, a vote for each district A_j is derived by taking the *majority* of the corresponding votes $(x_i)_{i \in A_j}$. If there are only two options, 0 and 1, for each vote x_i, which could e.g. represent a two-party system, we can model FPP as a map $\mathcal{F} : \{0,1\}^n \to \{0,1\}^m, x \mapsto (\mathsf{MAJ}_{A_1}(x), \ldots, \mathsf{MAJ}_{A_m}(x))$ of majority predicates. Of course, at this point, we cannot apply any of our mathematical results, as the voting behavior $x \in \{0,1\}^n$ is usually not a uniformly random bit string.

However, we can consider the following thought experiment: Imagine you are responsible for conducting a vote for electing a house of m representatives among n voters. Each voter can only vote for one of two parties, 0 and 1. In most countries, m is relatively small (usually 300 to 1000) and n is comparably large (tens or hundreds of millions). You want to use the FPP voting system for conducting the vote, however, you are aware of the phenomenon of Gerrymandering where districts $A_1, \ldots, A_m$ are maliciously chosen such that the party that has less than $n/2$ total votes attains more than $m/2$ of the total seats. You want to avoid Gerrymandering at all costs, and your idea is to simply choose the districts $A_1, \ldots, A_m$ uniformly at random such that each district has size n/m, all districts are disjoint, and their union contains all voters $1, \ldots, n$. In this regime, we can apply our methods and prove two results: The first result is positive and states that you will avoid the effect of Gerrymandering with overwhelming probability. The second result is negative and states that if one party has a tiny lead in total votes it will receive all m seats with overwhelming probability.

To see both results, remember that the probability $q(n, h, n/m, n/(2m)) = \Pr_{A_j}\left[\mathsf{MAJ}_{A_j}(x) = 1\right]$ is independent of the distribution of x, as long as x has Hamming weight h. By Proposition 1 and Theorem 3, we have

$$q\left(n, h, \frac{n}{m}, \frac{n}{2m}\right) \geq 1 - \exp\left(-2\left(\frac{h}{n} - \frac{1}{2}\right)^2 \cdot \frac{n}{m}\right).$$

Let $\alpha \in (0,1)$ be dependent on n. For $h \geq n/2 + \sqrt{\ln(\alpha^{-1})nm}$, we have

$$q\left(n, h, \frac{n}{m}, \frac{n}{2m}\right) \geq 1 - \exp\left(-2\left(\sqrt{\frac{m}{n}}/\ln(\alpha^{-1})\right)^2 \cdot \frac{n}{m}\right) = 1 - \exp\left(2\ln(\alpha)\right) = 1 - \alpha^2.$$

This means, the probability q to win one seat is at least $1 - \alpha^2$ if h is at least $n/2 + \sqrt{\ln(\alpha^{-1}) \cdot nm}$. Now, the distributions of $A_1, \ldots, A_m$ are not independent, since $A_1, \ldots, A_m$ need to be disjoint. However, we

[15] Also known as *Winner-take-all* system.

can bound the common bias of the seats in this case for fixed $h = \mathsf{w}_{\mathsf{H}}(x)$ by

$$\begin{aligned}
&\Pr_{A_1,\ldots,A_m \leftarrow \binom{[n]}{n/m}, \bigcup_i A_i = \{1,\ldots,n\}} [\forall i : \ \mathsf{MAJ}_{A_i}(x) = 1 \mid \mathsf{w}_{\mathsf{H}}(x) = h] \\
\geq & 1 - \sum_{i=1}^{m} (1 - \Pr_{A_i \leftarrow \binom{[n]}{n/m}} [\mathsf{MAJ}_{A_i}(x) = 1 \mid \mathsf{w}_{\mathsf{H}}(x) = h]) \\
= & 1 - m \cdot (1 - q(n, h, m/n)) = 1 - m\alpha^2.
\end{aligned}$$

With $m = 500$ and $\alpha = \frac{1}{\sqrt{50000}}$, the common bias is at least 99%, while the necessary number of votes is $h = n/2 + \sqrt{\ln(\alpha^{-1})nm} = n/2 + \sqrt{\frac{1}{2} \cdot \ln(50000) \cdot 500 \cdot n} \approx n/2 + 52 \cdot \sqrt{n}$. It follows, for an increasing number of voters, one of the party needs less and less percental lead in total votes to secure all seats if all voting districts are distributed randomly. Of course, the weaknesses of FPP voting are already known.

Voters n	Necessary Percental Lead $\frac{h}{n} - \frac{1}{2}$
1M	5,2%
10M	1,6%
100M	0,5%
1B	0,16%

Table 12: Lower Bounds on the number of total votes a party needs to secure all of $m = 500$ seats with at least 99% probability. Percental numbers are rounded.

We see our results and thought experiment merely as a quantification of some of its weaknesses. Our interpretation is that some level of bias is intrinsic to FPP voting and cannot be eliminated.