AGB 2.0: Refined Algebraic Attack against Regular Syndrome Decoding for PCG Applications

Hanlin Liu
Northwestern University
hanlin.liu@northwestern.edu

Xiao Wang
Northwestern University
wangxiao@northwestern.edu

Kang Yang
State Key Laboratory of Cryptology
yangk@sklc.org

Longhui Yin
Tsinghua University
ylh21@mails.tsinghua.edu.cn

Yu Yu
Shanghai Jiao Tong University
Shanghai Qi Zhi Institute
yuyu@yuyu.hk

Abstract

The Regular Syndrome Decoding (RSD) problem, introduced nearly two decades ago, is a regular version of the Syndrome Decoding (SD) problem, where the noise vector is divided into consecutive, equally sized blocks, each containing exactly one noisy coordinate. Recently, RSD has gained renewed attention for its applications in Pseudorandom Correlation Generator (PCG) and more. Very recently, several works presented the improved algebraic approach (AGB for short) and ISD approach (including regular-ISD and regular-RP) to utilize the regular structure of noise vectors and provide a more precise security evaluation for the RSD problem.

In this paper, we refine the AGB algorithm from a one-round process to a two-round process, and refer to the new algebraic algorithm as AGB 2.0. For each round, we guess a few noise-free positions, followed by a tailored partial XL algorithm. This interleaving strategy increases the probability of success by reducing the number of guessing noise-free positions, and effectively lowers the problem's dimension in each round. By fine-tuning position guesses in each round and optimizing the aggregate running time, our AGB 2.0 algorithm reduces the concrete security of the RSD problem by up to 6 bits for the parameter sets used in prior works, compared to the best-known attack. In particular, for a specific parameter set in Wolverine, the RSD security is 7 bits below the 128-bit target. We analyze the asymptotic complexity of algebraic attacks on the RSD problem over a finite field $\mathbb{F}$ with the field size $|\mathbb{F}| > 2$, when the noise rate ρ and code rate R satisfy $\rho + R < 1$. If $n\rho R^2 = O(1)$ where n is the noise length, the RSD problem over $\mathbb{F}$ can be solved in polynomial time, but it does not hold for the SD problem. We show that the ISD and its variants, including regular-ISD and regular-RP, are asymptotically less efficient than AGB for solving RSD problems with $R = o(1/\log(n))$ and $|\mathbb{F}| > e^{n\rho R}$.

1 Introduction

The Syndrome Decoding (SD) problem is one fundamental problem in coding theory, and also serves as the dual problem of Learning Parity with Noise (LPN). The SD problem asks to recover a sparse noise vector $\boldsymbol{e}$ such that $\mathbf{H} \cdot \boldsymbol{e} = \boldsymbol{y}$, where both matrix $\mathbf{H}$ and vector $\boldsymbol{y}$ are public. The hardness of SD problems has been used as the security foundation of cryptographic constructions, such as symmetric encryption and authentication [HB01], public key encryption [Ale03], commitment schemes [JKPT12], etc.

To achieve better efficiency, the recent works, including Pseudorandom Correlation Generator (PCG) [BCGI18], Secure Multi-Party Computation (MPC) [HOSS18] and digital signatures [CCJ23, CLY+24, BCC+24], consider a structured version of SD problem, known as Regular Syndrome Decoding (RSD) problem. RSD was initially introduced in the context of the FSB hash function [AFS05], and adopts a regular noise vector $\boldsymbol{e}$, i.e., it consists of consecutive equal-length blocks, each of weight exactly one. Particularly, the PCG paradigm uses the sparsity of a noise vector $\boldsymbol{e}$ to produce a large batch of correlated randomness, e.g., Oblivious Transfer (OT), Oblivious Linear Evaluation (OLE) and Vector OLE (VOLE), at the sublinear communication. The PCG paradigm has been used to construct a lot of concretely efficient OT and (V)OLE protocols [BCG+19b, BCG+19a, SGRR19, YWL+20, BCG+20, WYKW21, CRR21, BCG+22, AS22, BCCD23a, RRT23a, BBC+24]. These PCG-like protocols have been used to design various concretely efficient protocols, including efficient MPC protocols (e.g., [DILO22, RS22, CWYY23]), scalable zero-knowledge proofs (e.g., [WYKW21, BMRS21, DIO21, YSWW21]), privacy-preserving machine learning (e.g., [SGRR19, WYX+21, HjLHD22]), private set intersection (e.g., [RS21, RR22, BC23]), etc. The security of all PCG-like protocols and their applications rely on the hardness of the RSD problem, and they adopt RSD in the low-noise setting (i.e., RSD with low noise rate but very high dimension). Therefore, it is an important task to analyze the concrete security of low-noise RSD problems.

Several recent works exploited the regular structure of a noise vector to solve the RSD problem in the low-noise setting. Specifically, Briaud and Øygarden [BØ23] modeled the RSD problem as a quadratic system and solved it with an algebraic attack (AGB in short) approach. Esser and Santini [ES24] introduced regular-ISD algorithms by optimizing information set decoding (ISD) algorithms for the regular structure of noise vectors, and showed that advanced ISD techniques [Pra62, Ste88, Dum91, MMT11, BJMM12, MO15, BM18] for the SD problem over $\mathbb{F}_2$ can be adapted to the RSD problem over $\mathbb{F}_2$. Besides, Liu et al. [LWYY24] also show how to adapt the ISD algorithm [Pra62, Ste88, Dum91, MMT11, BJMM12] to utilize the regular structure of noise vectors over $\mathbb{F}_2$. In this paper, we revisit the AGB approach [BØ23], and refine it to estimate the concrete security of low-noise RSD problems over any field.

In terms of theoretical study, several recent works compared the hardness of RSD and SD problems through known algorithms and reductions. In particular, Carozza et al. [CCJ23] identified three parameter regimes where SD is easier, harder, or incomparable to RSD. Esser and Santini [ES24] described some cases where RSD is practically harder to solve than SD with current algorithms. They also found the parameter regimes where RSD can be solved in polynomial time due to the regular structure of noise vectors, while SD remains polynomially intractable. It is also important to find new parameter regimes, where RSD can be solved in polynomial time but SD is still hard, which will help us to better understand and compare the hardness of RSD and SD.

1.1 Overview of The AGB Algorithm

Briaud and Øygarden [BØ23] modeled an RSD instance $(\mathbf{H}, \boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{e})$ as a system of Multivariate Quadratic (MQ) equations. Here, $\boldsymbol{e}$ is a regular noise vector, i.e., a weight-t vector of length-βt, composed of t blocks, each of length-β and weight-1. Along with the linear equations $\boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{e}$, $\boldsymbol{e}$ is represented by quadratic equations. Specifically, the cross terms between distinct elements within each block are zero. The AGB algorithm then performs the following steps:

1. **Guessing noise-free positions.** Guess μ noise-free positions in each of the first f blocks of $\boldsymbol{e}$ [1], where f, μ are two parameters.

2. **Invoking the XL algorithm.** If the above noise-free positions guessed are correct, apply the XL algorithm [CKPS00, YCBC07, CCNY12] (as detailed in Section 2.3) to solve the resulting MQ system. This algorithm initially elevates the MQ system into a pre-determined degree $d \geq 2$ by multiplying suitable monomials to each equation. It then considers each monomial (of up to degree d) as an independent variable to linearize the system, and finally, it solves the linear system using Gaussian elimination.

The expected time of the AGB algorithm is given by:

$$T_{\mathsf{AGB}} = T_{\mathsf{XL}} / p(f, \mu) \ ,$$

where T_{XL} represents the cost of the XL algorithm, mainly due to Gaussian elimination, and $p(f, \mu) \stackrel{\text{def}}{=} (1 - \mu/\beta)^f$ is the success probability of the first step.

Among the attacks solving the RSD problem, the AGB algorithm [BØ23] stands out for two key reasons. First, it is one of the few "non-linear" attacks, which is outside of the linear testing framework that is used in the recent PCG works [BCG+22, BCCD23b] to analyze the security of the RSD problem with new linear codes, where this framework captures linear attacks, such as ISD [Pra62, Ste88, Dum91, MMT11, BJMM12, MO15, BM18] and Pooled Gauss [EKM17]. Therefore, studying "non-linear" attacks, such as AGB, is crucial to verify the security of these PCG schemes. Second, in contrast to other recent attacks [CCJ23, ES24, LWYY24, KL24], which mainly target the RSD problem over $\mathbb{F}_2$ or significantly large fields (e.g., $|\mathbb{F}| = 2^{128}$), the AGB algorithm can be used to attack the RSD problem over any-sized field. As a result, AGB and its improvements become increasingly useful for future RSD applications.

1.2 Our Result and Technique

Our AGB 2.0 approach improves the original AGB algorithm by incorporating the XL algorithm with a two-round position-guessing strategy:

1. **Applying a partial XL algorithm.** Apply the first two steps of the XL algorithm: elevating the MQ system to a degree d and linearizing it as described above.

2. **First round of guessing noise-free positions.** Guess μ_1 noise-free positions in each of the first f_1 blocks of $\boldsymbol{e}$.

[1] Note that $\boldsymbol{e}$ and $\mathbf{H}$ can be efficiently shuffled. For simplicity, we, w.l.o.g., assume throughout this paper that the guessed noise-free positions are in the first f blocks.

3. **First Gaussian elimination.** Reduce the number of unknowns without solving the entire system using Gaussian elimination.

4. **Second round of guessing noise-free positions.** Guess μ (resp., another $\mu - \mu_1$) noise-free positions in the i-th block with $f_1 < i \leq f$ (resp., each of the first f_1 blocks) of $\boldsymbol{e}$.

5. **Second Gaussian elimination.** If the noise-free position guesses are correct in both rounds, solve the remaining linear system with Gaussian elimination.

The expected time of our AGB 2.0 algorithm is

$$T_{\mathsf{AGB2.0}} = \frac{T_1}{p(f_1, \mu_1)} + \frac{T_2}{p(f, \mu)} \, .$$

Here, T_1 represents the cost of the first Gaussian elimination (Step 3), and T_2 represents the cost of the second Gaussian elimination (Step 5). Since $p(f_1, \mu_1) \geq p(f, \mu)$ and $T_2 \leq T_1 \approx T_{\mathsf{XL}}$, our AGB 2.0 algorithm becomes more efficient than AGB, especially as $p(f, \mu) \ll 1$. For example, let f and μ in AGB 2.0 be the same as the optimization parameters in AGB, with $f_1 = f/2$ and $\mu_1 = \mu$ [2]. This yields $p(f_1, \mu_1) = \sqrt{p(f, \mu)} \ll 1$. The first round of the algorithm reduces the number of unknowns by at least half, resulting in $T_2 \ll T_1$. Thus, the time complexity of AGB 2.0 can be expressed as:

$$T_{\mathsf{AGB2.0}} \leq \frac{T_1}{p(f_1, \mu_1)} + \frac{T_2}{p(f, \mu)} \approx \frac{T_{\mathsf{XL}}}{\sqrt{p(f, \mu)}} + \frac{T_2}{p(f, \mu)} \ll T_{\mathsf{AGB}} \, . \tag{1}$$

We analyze the asymptotic complexity of pure AGB (with $f = f_1 = \mu = \mu_1 = 0$ in AGB 2.0) using the Gröbner basis method [Fau99, Fau02, YCC04, ST21] to solve the RSD problem over a field $\mathbb{F}$ with the field size $|\mathbb{F}| > 2$, for $\rho + R < 1$, where ρ is the noise rate and R is the code rate. Our analysis yields two findings. First, we classify the parameter regime $n\rho R^2 = O(1)$, where n is the noise length, indicating that the problem can be solved in polynomial time. However, this does not apply to the SD setting. Second, we demonstrate that the ISD and its variants, such as regular-ISD [ES24] and regular-RP [KL24], have higher asymptotic costs than the pure AGB for the RSD problem when $R = o(1/\log(n))$ and $|\mathbb{F}| > e^{n\rho R}$.

We compare our AGB 2.0 algorithm with the best-known attacks, including the AGB algorithm [BØ23], the ISD algorithms [MMT11, BJMM12] and their adaptations for regular noise [LWYY24, ES24], the CCJ algorithm [CCJ23], and the concurrent and independent regular-RP [KL24]. The comparison uses RSD parameter sets from Boyle et al. [BCGI18] and Liu et al. [LWYY24]. Selected results are presented in Table 1, while detailed results can be found in Table 2 of Section 4. Our AGB 2.0 algorithm reduces the concrete security of the RSD problem by up to 6 bits compared to the best-known attack, as shown in Table 1. We also compare AGB 2.0 with the best previous attacks for the RSD parameter sets used in practical PCG-like protocols Wolverine [WYKW21] and Ferret [YWL+20], in Table 4 and Table 5. Notably, the security of RSD for the Wolverine's parameter set $(n = 642048, k = 19870, t = 2508)$ is 7 bits lower than the target 128-bit security, as shown in Table 4. Note that the practical VOLE protocol Wolverine [WYKW21] has already

[2] In practice, our AGB 2.0 algorithm selects the parameters f_1, μ_1, f, μ that minimize $T_{\mathsf{AGB2.0}}$; this parameter choice is just one possible option as an example.

Table 1: A comparison of our AGB 2.0 algorithm with the previous best attacks, including ISD [MMT11, BJMM12], AGB [BØ23], CCJ [CCJ23], Regular-ISD [ES24] and the concurrent and independent Regular-RP [KL24], to evaluate the bit-security of **selected** RSD instances over a finite field $\mathbb{F}$ with noise length n, dimension k, and Hamming weight t of a regular noise vector.

source	Regular SD				previous best	AGB 2.0
	$\lvert\mathbb{F}\rvert$	n	k	t		
Liu et al. [LWYY24]	2^{128}	2^{18}	15336	1312	133	127
		2^{20}	32771	2467	131	127
Wolverine [WYKW21]	$2^{61}-1$	642048	19870	2508	126	121
Liu et al. [LWYY24]	2	2^{18}	15336	1312	122	117
		2^{20}	32771	2467	125	119
Ferret [YWL+20]		609728	36288	1269	138	134

been included in open-sourced libraries such as EMP-toolkit [WMK16], SecretFlow [Ant22] and ZK-RAM [YH23]. We provide a Python script [3] to estimate the bit-security of RSD instances against our AGB 2.0 attack.

1.3 Comparison with Related Techniques

Our multi-stage approach differs from the Crossbred algorithm [JV17], focusing on Boolean polynomial systems, as we focus on RSD problems over any field. Unlike Crossbred, which guesses some variables in one round similar to [BØ23], our method employs two rounds of guessing. Instead of enumerating all possible variable values as Crossbred, we leverage sparsely distributed variables and guess them to be zeros with higher probability. The Crossbred algorithm faces challenges with RSD over a large field, such as $|\mathbb{F}| = 2^{128}$, due to the need to enumerate all non-zero choices in $\mathbb{F}$.

Our high-level strategy resembles the ISD variant [CC98], which reuses computations across iterations to lower amortized costs, but the techniques differ. The ISD variant iteratively selects an information set by modifying one element at a time while keeping others unchanged. In contrast, our algorithm guesses noise-free positions in two rounds: the first round reduces the dimension of the RSD problem, and the second round solves it. By reusing results from the first round in later iterations, we achieve further decreases in amortized costs.

Concurrent work. In a concurrent work, Kim and Lee [KL24] introduced the regular Reduce and Prange (regular-RP) algorithm. Similar to AGB 2.0, the regular-RP consists of multiple rounds, and reuses the results from previous rounds. However, the regular-RP algorithm focuses on optimizing the Pooled Gauss approach [EKM17, LWYY24] to solve RSD over significantly larger fields, such as $|\mathbb{F}| = 2^{128}$. In contrast, our approach refines AGB, one of the few "non-linear" attacks, to address RSD over any field. In another concurrent work, Wang et al. [WWY+25] proposed a hybrid attack combining ISD with the algebraic approach. Compared to the best-known analysis, they further reduced the concrete security of LPN instances by up to 20 bits. Their analysis adopts an assumption different from the semi-regular assumption that our approach relies on. Our work is orthogonal

[3] Available at https://github.com/hansliu1024/AGB2.0.

from Wang et al.'s work [WWY+25], and has the potential of being applied in their hybrid attack framework to further reduce the cost of solving LPN problems.

2 Preliminaries

2.1 Notation and Definitions

We use $\log(\cdot)$ for the binary logarithm and $\ln(\cdot)$ for the natural logarithm. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \cdots, n\}$. The notation $\boldsymbol{x} \xleftarrow{\$} \mathcal{S}$ means sampling $\boldsymbol{x}$ uniformly from set $\mathcal{S}$, and $\boldsymbol{x} \leftarrow \mathcal{S}$ refers to sampling according to the distribution $\mathcal{S}$. We denote column vectors by bold lowercase letters (e.g., $\boldsymbol{a}$) and matrices by bold uppercase letters (e.g., $\mathbf{A}$). The identity matrix is written as $\mathbf{I}$. By slightly abusing the notation, for a finite field $\mathbb{F}$ and a vector $\boldsymbol{a}$, $|\mathbb{F}|$ represents the size of $\mathbb{F}$, and $|\boldsymbol{a}|$ refers to the Hamming weight of $\boldsymbol{a}$.

For a finite field $\mathbb{F}$, we denote the exact distribution of length n and Hamming weight t as $\mathsf{HW}_{t,n}(\mathbb{F})$. For $\boldsymbol{e} \leftarrow \mathsf{HW}_{t,n}(\mathbb{F})$, t random positions have uniform non-zero elements, and all other positions are zero. The regular distribution with length n and Hamming weight t is denoted as $\mathsf{RHW}_{t,n}(\mathbb{F})$. For $\boldsymbol{e} \leftarrow \mathsf{RHW}_{t,n}(\mathbb{F})$, the vector is divided into t consecutive blocks of size $\beta = \lfloor n/t \rfloor$, each with one noisy coordinate. If n/t is not an integer, the last $n - t \cdot \beta$ positions are set to zero.

Consider the multivariate polynomial ring $\mathcal{A} = \mathbb{F}[\boldsymbol{x}]$, where $\boldsymbol{x}$ represents n variables $\boldsymbol{x}_1, \ldots \boldsymbol{x}_n \in \mathbb{F}$ and coefficients are in the field $\mathbb{F}$. A polynomial $f \in \mathcal{A}$ is homogeneous if all terms have the same degree; otherwise, it is affine. For an affine polynomial f, we form its homogeneous part $f^{(h)}$ by including only the terms of degree $\deg(f)$ (i.e., excluding lower-degree terms). Throughout this paper, we do not distinguish between a polynomial $f \in \mathcal{A}$ and its corresponding equation $f = 0$.

For a subset $\mathcal{F} \subset \mathcal{A}$, the ideal generated by $\mathcal{F}$ is written as $\langle \mathcal{F} \rangle$. An ideal is homogeneous if its generators are homogeneous. To homogenize an affine ideal $\mathcal{I} \subset \mathcal{A}$ into $\mathcal{I}^{(h)}$, each generator f is replaced with its homogeneous part $f^{(h)}$, i.e., $\mathcal{I}^{(h)} \stackrel{\text{def}}{=} \langle \mathcal{F}^{(h)} \rangle$, where $\mathcal{F}^{(h)} \stackrel{\text{def}}{=} \{f^{(h)} : f \in \mathcal{F}\}$. If $\mathcal{I}$ is homogeneous, the set $\mathcal{I}_d \stackrel{\text{def}}{=} \{f \in \mathcal{I}, \deg(f) = d\} = \mathcal{I} \cap \mathcal{A}_d$ forms a subspace of $\mathcal{A}_d \stackrel{\text{def}}{=} \{f \in \mathcal{A} : f \text{ is homogeneous and } \deg(f) = d\}$, the space of homogeneous polynomials of degree d.

HILBERT FUNCTION AND HILBERT SERIES. We examine the Hilbert function for a homogeneous ideal $\mathcal{I} \subset \mathcal{A}$, which maps an integer d to the dimension of the quotient $\mathcal{A}_d / \mathcal{I}_d$, considered as a vector space over $\mathbb{F}$. The mapping is defined as:

$$\mathcal{HF}_{\mathcal{A}/\mathcal{I}}(d) \stackrel{\text{def}}{=} \dim_{\mathbb{F}} \left(\mathcal{A}_d / \mathcal{I}_d \right) .$$

The Hilbert series, a key analytical tool for homogeneous ideals, is defined for the quotient ring $\mathcal{A}/\mathcal{I}$ by the series:

$$\mathcal{H}_{\mathcal{A}/\mathcal{I}}(z) \stackrel{\text{def}}{=} \sum_{d=0}^{\infty} \mathcal{HF}_{\mathcal{A}/\mathcal{I}}(d) \cdot z^d .$$

In a finite field $\mathbb{F}$, all field equations of the form $\boldsymbol{x}_i^{|\mathbb{F}|} - \boldsymbol{x}_i = 0$ for each $i \in [n]$ are included. This analysis primarily concerns zero-dimensional ideals, where the quotient $\mathcal{A}/\mathcal{I}$ forms a finite-dimensional vector space. The degree of regularity d_{reg} is the smallest d for which $\mathcal{I}_d = \mathcal{A}_d$. In the series $\mathcal{H}$, $[z^j]\mathcal{H}(z)$ indicates the coefficient of z^j in $\mathcal{H}$.

MACAULAY MATRIX. The set Mon consists of all monomials of the form $\boldsymbol{x}_1^{\alpha_1} \cdots \boldsymbol{x}_n^{\alpha_n}$, where each α_i is a non-negative integer. Given a polynomial $f \in \mathcal{A}$ and a monomial $t \in \mathsf{Mon}$, $\mathsf{coeff}(f, t)$ denotes the coefficient of t in f. The set $\mathcal{A}_{\leq d}$ includes all polynomials with degrees up to d [4], and $\mathsf{Mon}_{\leq d}$ is the intersection of Mon and $\mathcal{A}_{\leq d}$, i.e., $\mathsf{Mon}_{\leq d} \stackrel{\text{def}}{=} \mathsf{Mon} \cap \mathcal{A}_{\leq d}$. For a finite set of polynomials $\mathcal{F} \subset \mathcal{A}$ and a degree $d \in \mathbb{N}$, we denote the shift of $\mathcal{F}$ to degree d as $\mathcal{I}_{\leq d}(\mathcal{F})$. This set, $\mathcal{I}_{\leq d}(\mathcal{F})$, consists of all products $t \cdot f$ with $\deg(t \cdot f) \leq d$, where $t \in \mathsf{Mon}$ and $f \in \mathcal{F}$. Specifically, $\mathcal{I}_{\leq d}(\mathcal{F}) = \{t \cdot f : t \in \mathsf{Mon}, f \in \mathcal{F}\} \cap \mathcal{A}_{\leq d}$. The Macaulay matrix $\mathsf{Mac}(\mathcal{F}, \mathcal{T})$ for $\mathcal{F} = \{f_1, \ldots, f_m\}$ relative to a monomial set $\mathcal{T} = \{t_1, t_2, \cdots, t_n\} \subseteq \mathsf{Mon}$ is an $(m \times n)$-matrix with (i, j)-entry $\mathsf{coeff}(f_i, t_j)$, arranged as follows:

$$\mathsf{Mac}(\mathcal{F}, \mathcal{T}) \stackrel{\text{def}}{=} \begin{pmatrix} \mathsf{coeff}(f_1, t_1) & \cdots & \mathsf{coeff}(f_1, t_n) \\ \vdots & & \vdots \\ \mathsf{coeff}(f_m, t_1) & \cdots & \mathsf{coeff}(f_m, t_n) \end{pmatrix}.$$

2.2 The Multivariate Quadratic Problem

The Multivariate Quadratic (MQ) problem is defined by a map $\mathcal{F}$, which consists of m quadratic polynomials $f_1, \ldots, f_m$ in n variables $\boldsymbol{x} \in \mathbb{F}^n$. The goal is to find $\boldsymbol{x}$ such that $f_i(\boldsymbol{x}) = 0$ for each polynomial. Notable algorithms for solving the MQ problem include F4 [Fau99], F5 [Fau02], and the XL algorithm [CKPS00].

Definition 1 (The Multivariate Quadratic (MQ))**.** *The MQ problem aims to find a solution vector $\boldsymbol{x} \in \mathbb{F}^n$ such that for a given MQ map $\mathcal{F} = (f_1, \ldots, f_m) : \mathbb{F}^n \to \mathbb{F}^m$, the equation $f_i(\boldsymbol{x}) = 0$ holds for each $i \in [m]$. Each f_i is a quadratic polynomial with variables and coefficients from the finite field $\mathbb{F}$.*

2.3 The XL Algorithm

The XL algorithm, proposed by Courtois et al. [CKPS00], solves systems of multivariate polynomials over finite fields in two main stages. First, it elevates the degree of the MQ system to a predetermined degree $d \geq 2$ by multiplying all appropriate monomials with each polynomial in the MQ map $\mathcal{F}$. Second, it solves the system using Gaussian elimination, where each monomial up to degree d is considered an independent variable. Denote $\boldsymbol{x}_{\mathsf{Mon}_{\leq d}}$ as the variable vector, where each term corresponds to a monomial in $\mathsf{Mon}_{\leq d}$, including the constant term. The success of the XL algorithm is based on choosing a predetermined degree d as the witness degree d_{wit} (see Definition 2). This selection ensures that there are sufficient linearly independent equations for the monomials in the Macaulay matrix $\mathsf{Mac}\left(\mathcal{I}_{\leq d}(\mathcal{F}), \mathsf{Mon}_{\leq d}\right)$.

Definition 2 (The Witness Degree d_{wit}, detailed in [BFSS13, BØ23])**.** *Consider the polynomial system $\mathcal{F} = \{f_1, \ldots f_m\}$ over field $\mathbb{F}$, with its associated ideal $\mathcal{I} \stackrel{def}{=} \langle \mathcal{F} \rangle$. We define the $\mathbb{F}$-vector*

[4] We do not require that polynomials in $\mathcal{A}_{\leq d}$ be homogeneous.

spaces $\mathcal{I}_{\leq d}$ and $\mathcal{J}_{\leq d}$ as follows:

$$\mathcal{I}_{\leq d} \stackrel{def}{=} \{p \in I \mid \deg(p) \leq d\},$$
$$\mathcal{J}_{\leq d} \stackrel{def}{=} \left\{ \sum_{i=1}^{m} g_i f_i \;\middle|\; \sum_{i=1}^{m} g_i f_i \in I \text{ and each } \deg(g_i) \leq d - \deg(f_i) \right\}.$$

The witness degree d_{wit} for $\mathcal{F}$ is the smallest integer d such that $\mathcal{I}_{\leq d} = \mathcal{J}_{\leq d}$ and the leading monomial ideals of $\mathcal{I}_{\leq d}$ and $\mathcal{I}$ are equal, i.e., $\mathsf{LM}(\mathcal{I}_{\leq d}) = \mathsf{LM}(\mathcal{I})$. The leading monomial ideal $\mathsf{LM}(\mathcal{I})$ for any ideal $\mathcal{I}$ is generated by the leading monomials of all polynomials in $\mathcal{I}$ under any graded ordering.

For semi-regular systems [Bar04a], d_{reg} is the smallest degree such that $\mathcal{I}_{\leq d} = \mathcal{J}_{\leq d}$, and thus $d_{\mathsf{wit}} \geq d_{\mathsf{reg}}$. The witness degree d_{wit} and the regularity degree d_{reg} are used in different MQ algorithms.

Algorithm 1: The XL Algorithm [CKPS00]

Input: An polynomial system $\mathcal{F} = (f_1, \cdots, f_m) \subset \mathcal{A}$, and witness degree d_{wit}.
Output: A solution $\boldsymbol{x}$ such that $f_i(\boldsymbol{x}) = 0$ for all $i \in [m]$.

1 **Multiplication.** Compute the shift of $\mathcal{F}$ to degree $d := d_{\mathsf{wit}}$, i.e., $\mathcal{I}_{\leq d}(\mathcal{F}) = \{t \cdot f : t \in \mathsf{Mon}, f \in \mathcal{F}, \deg(t) + \deg(f) \leq d\}$;
2 **Macaulay matrix representation.** Construct the Macaulay matrix $\mathbf{A} = \mathsf{Mac}\left(\mathcal{I}_{\leq d}(\mathcal{F}), \mathsf{Mon}_{\leq d}\right)$ such that the terms $(1, \boldsymbol{x}_1^1, \ldots, \boldsymbol{x}_1^d)$ are eliminated last; the equation system is now $\mathbf{A}\boldsymbol{x}_{\mathsf{Mon}_{\leq d}} = \mathbf{0}$;
3 **Gaussian elimination.** Compute the reduced row echelon form $\mathbf{B}$ of $\mathbf{A}$ using Gaussian elimination and determine a univariate polynomial $g(\boldsymbol{x}_1) = 0$ in $\boldsymbol{x}_1$ with degree no greater than d;
4 **Solve.** Compute the roots $\boldsymbol{x}_1 \in \mathbb{F}$ with respect to $g(\boldsymbol{x}_1) = 0$ using factorization algorithms, such as square-free [Yun76], distinct-degree [KS95] or equal-degree factorization algorithms [vzGS92];
5 **Repeat.** Substitute a root and subsequently determine the values of the remaining variables one by one;

We present the XL algorithm in Algorithm 1. In Step 3-5, Gaussian elimination is applied to the Macaulay matrix to obtain and solve a single-variable equation, and then the solution is substituted to find the full solution. Gaussian elimination can be performed using LU decomposition, but for sparse Macaulay matrices, Wiedemann XL [YCBC07, CCNY12] uses a more efficient option: the Wiedemann algorithm [Wie86] or its variants [Cop94, Tho02].

The time complexity of the XL algorithm is mainly determined by Gaussian elimination. For the original XL, it is approximately the same as LU decomposition, estimated at $5.64 \cdot (\mathcal{M}_{\leq d})^{2.8}$, where $\mathcal{M}_{\leq d}$ is the number of columns in the Macaulay matrix $\mathbf{A} = \mathsf{Mac}\left(\mathcal{I}_{\leq d}(\mathcal{F}), \mathsf{Mon}_{\leq d}\right)$, with the constant 5.64 from the literature [BH74]. For Wiedemann XL, the complexity is $3 \cdot n_{\mathcal{F}} \cdot (\mathcal{M}_{\leq d})^2$, where $n_{\mathcal{F}}$ is the number of non-zero terms per row in the Macaulay matrix $\mathbf{A}$, the constant is 3 [DY09, Proposition 3] and [Beu21].

Another improvement for the XL algorithm is the hybrid approach [YCC04, BFP09, FK21]. This method involves guessing the values of k out of n variables (e.g., $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_k$) and solving

the system with the remaining (e.g., $\boldsymbol{x}_{k+1}, \cdots, \boldsymbol{x}_n$) using F4 [Fau99], F5 [Fau02], or the XL algorithm [CKPS00].

2.4 The Syndrome Decoding Problem

Recent works have leveraged the Syndrome Decoding (SD) problem and its regular variant (RSD) to design PCG-like protocols that generate large batches of correlated randomness, such as correlated oblivious transfer (COT) and oblivious linear evaluation (OLE), with sub-linear communication. These protocols rely on the choice of finite field $\mathbb{F}$, resulting in two main cases:

- Case 1: For $|\mathbb{F}| > 2$, these PCG-like protocols are used to construct vector OLE [BCGI18, BCG+19b, BCG+19a, SGRR19, WYKW21, CRR21, BCG+22, RRT23b] and OLE [BCG+19b, BCG+20, AS22, BCCD23b].
- Case 2: For $\mathbb{F} = \mathbb{F}_2$, these PCG-like protocols are used in the design of COT [BCG+19b, BCG+19a, YWL+20, CRR21, BCG+22, RRT23b], which can be transformed into standard OT protocols.

To achieve better efficiency, a series of works, e.g., [AFS05, HOSS18, BCGI18, BCG+19b, BCG+19a, YWL+20, WYKW21, BCG+22, BCCD23b, CCJ23], utilize the RSD problem. Following prior research [BØ23, ES24], we define the SD and RSD problems over a general field $\mathbb{F}$ as follows:

Definition 3 (Computational Syndrome Decoding (SD))**.** *Let $(n, k, t) \in \mathbb{N}^3$ with $k < n$ and $t < n$. Sample $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k)\times n}$ and $\boldsymbol{e} \leftarrow \mathsf{HW}_{t,n}(\mathbb{F})$. Given $(\mathbf{H}, \boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{e})$, the goal is to recover the noise vector $\boldsymbol{e}$.*

Definition 4 (Computational Regular Syndrome Decoding (RSD))**.** *Let $(n, k, t) \in \mathbb{N}^3$ with $k < n$ and $t < n$. Sample $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k)\times n}$ and $\boldsymbol{e} \leftarrow \mathsf{RHW}_{t,n}(\mathbb{F})$. Given $(\mathbf{H}, \boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{e})$, the goal is to recover the noise vector $\boldsymbol{e}$. The code rate is $R = k/n$, and the noise rate is $\rho = t/n$.*

In the RSD, we observe that $n \geq 2t$ and $\rho \leq 1/2$, based on the requirement that the length of each noise block $\beta = \lfloor n/t \rfloor \geq 2$. If $\beta = 1$, this indicates that the first t entries of the noise vector $\boldsymbol{e}$ are 1 by the definition of regular noise. In this case, the RSD problem is trivial.

2.5 Recent Algebraic Approaches for RSD Attacks

CONSTRUCTING ALGEBRAIC MODELS FOR RSD. Briaud and Øygarden [BØ23] proposed a polynomial system to solve the RSD problem, wherein the noise vector $\boldsymbol{e}$ is divided into t consecutive blocks, each containing one noisy coordinate and with a block length of $\beta = \lfloor n/t \rfloor$. The system is defined by $n - k$ parity-check equations $\mathbf{H} \cdot \boldsymbol{e} - \boldsymbol{y} = 0$ and additional constraints on the regularity of $\boldsymbol{e}$. In particular, Model 1 addresses the RSD problem over large fields, whereas Model 2 focuses on binary cases.

Model 1 (The RSD over a large field [BØ23])**.** *Consider an RSD instance given by* $(\mathbf{H}, \boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{e})$ *over a large field* $\mathbb{F}$*. Define* $\mathcal{I} = \left\langle \mathcal{F}^{(h)} \right\rangle$*, where* $\mathcal{F}$ *is the set of polynomials described by* $\mathcal{F} \stackrel{def}{=} \mathcal{P} \cup \mathcal{B}$ [5]*, where*

- $\mathcal{P}$ *consists of* $n-k$ *linear equations derived from the parity-check matrix* $\mathbf{H}$*, specifically* $\mathbf{H} \cdot \boldsymbol{e} - \boldsymbol{y} = \mathbf{0}$*.*
- $\mathcal{B}$ *includes quadratic polynomials that model the regular noise* $\boldsymbol{e}$*, with* $\boldsymbol{e}_{i,j_1} \cdot \boldsymbol{e}_{i,j_2} = 0$ *for* $1 \leq i \leq t$ *and* $1 \leq j_1 < j_2 \leq \beta$*.*

Model 2 (The RSD over $\mathbb{F}_2$ [BØ23])**.** *Consider an RSD instance given by* $(\mathbf{H}, \boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{e})$ *over* $\mathbb{F}_2$*. Define* $\mathcal{I}_{\mathbb{F}_2} = \left\langle \mathcal{F}_{\mathbb{F}_2}^{(h)} \right\rangle$*, where* $\mathcal{F}_{\mathbb{F}_2}$ *is the set of polynomials described by* $\mathcal{F}_{\mathbb{F}_2} \stackrel{def}{=} \mathcal{P} \cup \mathcal{B} \cup \mathcal{Q} \cup \mathcal{L}$*, with* $\mathcal{P}$ *and* $\mathcal{B}$ *identical to those in Model 1. The additional sets are defined as follows:*

- $\mathcal{Q}$ *consists of field equations* $\boldsymbol{e}_{i,j}^2 - \boldsymbol{e}_{i,j} = 0$ *for* $1 \leq i \leq t$ *and* $1 \leq j \leq \beta$*.*
- $\mathcal{L}$ *includes* t *linear equations* $\sum_{j=1}^{\beta} \boldsymbol{e}_{i,j} - 1 = 0$ *for* $1 \leq i \leq t$*, ensuring that each block has a unique non-zero coordinate.*

Algorithm 2: The AGB for Model 1 or Model 2 [BØ23]

Input: A polynomial system $\mathcal{F}$ associated with Model 1 or Model 2 and parameters $(f, \mu, d_{\mathsf{wit}} = d(f, \mu))$.

Output: A solution to the corresponding polynomial system.

1 Guess μ noise-free positions for each of the first f blocks of $\boldsymbol{e}$;
2 Apply the Wiedemann XL algorithm with degree d_{wit} to solve the system;
3 Repeat the above steps until a solution is obtained;

DERIVING THE HILBERT SERIES. Briaud and Øygarden [BØ23] analyzed the Hilbert Series for both Model 1 and Model 2. Here, we present the analysis of Model 1, while the analysis for Model 2 follows a similar approach.

The Hilbert series for Model 1, denoted as $\mathcal{H}_{\mathcal{S}}(z)$, represents the Hilbert series of the quotient ring $\mathcal{S} = \mathcal{A} / \left\langle \mathcal{B}^{(h)} \right\rangle$. The coefficient of z^α in this series, represented by $[z^\alpha]\mathcal{H}_{\mathcal{S}}(z)$, counts monomials in the format $\boldsymbol{e}_{1,\ell_1}^{\alpha_1} \cdot \boldsymbol{e}_{2,\ell_2}^{\alpha_2} \cdots \boldsymbol{e}_{t,\ell_t}^{\alpha_t}$. Here, each $\alpha_i \geq 0$, each $\ell_i \in [\beta]$, and $\sum_{i=1}^{t} \alpha_i = \alpha$. Note that monomials containing $\boldsymbol{e}_{i,j_1} \cdot \boldsymbol{e}_{i,j_2}$, for $1 \leq i \leq t$ and $1 \leq j_1 < j_2 \leq \beta$, are excluded because each $\boldsymbol{e}_{i,j_1} \cdot \boldsymbol{e}_{i,j_2} = 0$, as specified in $\mathcal{B}$. Therefore, the coefficient of z^α in the series is given by

$$[z^\alpha]\mathcal{H}_{\mathcal{S}}(z) = \sum_{i=1}^{\alpha} \left(\beta^i \cdot \binom{t}{i} \cdot \binom{\alpha - 1}{i - 1} \right),$$

[5] Field equations $e_{i,j}^{|\mathbb{F}|} - e_{i,j} = 0$ are omitted as $|\mathbb{F}|$ is significantly larger than the witness degree d_{wit}, making them useless.

and the series is represented by

$$\mathcal{H}_{\mathcal{S}}(z) = \left(1 + \beta \cdot \frac{z}{1-z}\right)^{t} \text{ [6]},$$

which indicates the number of monomials relevant for analysis in AGB.

Let $\mathcal{H}_{\mathcal{A}/\mathcal{I}}(z)$ denote the Hilbert series for the quotient $\mathcal{A}/\mathcal{I}$, where $\mathcal{I} = \left\langle \mathcal{F}^{(h)} \right\rangle$ represents the homogeneous ideal. Note that we can express $\mathcal{H}_{\mathcal{A}/\mathcal{I}}(z)$ as $\mathcal{H}_{\mathcal{S}/\left\langle \mathcal{P}^{(h)} \right\rangle}(z)$, because $\mathcal{A}/\mathcal{I}$ is the same as $\mathcal{S}/\left\langle \mathcal{P}^{(h)} \right\rangle$. In this homogeneous regular system, for each polynomial $p_1 \in \mathcal{P}$ of degree 1, we include the equation $g \cdot p_1 = 0$, for every term $g \in \mathcal{A}_{\ell-1}$. This implies:

$$[z^{\ell}]\mathcal{H}_{\mathcal{S}/\left\langle p_1^{(h)} \right\rangle}(z) = [z^{\ell}]\mathcal{H}_{\mathcal{S}}(z) - [z^{\ell-1}]\mathcal{H}_{\mathcal{S}}(z)\,,$$

assuming p_1 is not a zero divisor in $\mathcal{S}$ (see Assumption 1). Therefore, the Hilbert polynomial is

$$\mathcal{H}_{\mathcal{S}/\left\langle p_1^{(h)} \right\rangle}(z) = (1-z) \cdot \mathcal{H}_{\mathcal{S}}(z)\,.$$

This procedure extends to $\mathcal{S}/\left\langle p_1^{(h)}, \cdots, p_{i-1}^{(h)} \right\rangle$ for each subsequent p_i, assuming that p_i is not a zero divisor in $\mathcal{S}/\left\langle p_1^{(h)}, \cdots, p_{i-1}^{(h)} \right\rangle$ (see Assumption 1). Multiplying by $(1-z)$ each time yields the final Hilbert series:

$$\mathcal{H}_{\mathcal{A}/\mathcal{I}}(z) = \left[(1-z)^{n-k} \cdot \left(1 + \beta \cdot \frac{z}{1-z}\right)^{t}\right]_{+} \text{ [7]}\,.$$

The witness degree d_{wit} is estimated by the index of the first non-positive coefficient in the series $\mathcal{H}_{\mathcal{A}/\mathcal{I}}(z)/(1-z)$.

Assumption 1 (see [BØ23])**.** *Consider an instance $\mathcal{F}$ of Model 1 with the degree of regularity d_{reg} for the ideal $\mathcal{I} = \left\langle \mathcal{F}^{(h)} \right\rangle$. Define the quotient ring $\mathcal{S} = \mathcal{A}/\left\langle \mathcal{B}^{(h)} \right\rangle$. Let $\mathcal{P} = \{p_1, \ldots, p_{n-k}\}$ represent the linear parity-check equations. Assume that for $1 \leq i \leq n-k$, if $g \cdot p_i = 0$ in the quotient space $\mathcal{S}/\left\langle p_1^{(h)}, \ldots, p_{i-1}^{(h)} \right\rangle$ and $\deg(g) + \deg(p_i) < d_{\mathsf{reg}}$, then $g = 0$ in this quotient space.*

For Model 2, Briaud and Øygarden [BØ23] need to address an additional issue: some polynomials in $\mathcal{L}^{(h)}$ become zero divisors in $\mathcal{A}/\left\langle \mathcal{B}^{(h)} \cup \mathcal{Q}^{(h)} \right\rangle$, which prevents Hilbert series analysis. To resolve this, they applied a graded ring homomorphism, denoted by $\mathcal{L}'$, to eliminate the t variables:

$$\begin{aligned} \mathcal{L}' : \mathbb{F}_2[\boldsymbol{e}] &\longrightarrow \mathbb{F}_2[\boldsymbol{x}] \\ \boldsymbol{e}_{i,j} &\longmapsto \boldsymbol{x}_{i,j}, \text{ for } 1 \leq i \leq t \text{ and } 1 \leq j < \beta \\ \boldsymbol{e}_{i,\beta} &\longmapsto \sum_{j=1}^{\beta-1} \boldsymbol{x}_{i,j}, \text{ for } 1 \leq i \leq t\,. \end{aligned} \tag{2}$$

[6] We can express $\frac{1}{1-z}$ as $1 + \sum_{i=1}^{\infty} z^i$.

[7] The notation $[\cdot]_+$ indicates the operation of dropping all terms after the first non-positive coefficient term.

This allowed analysis via the isomorphism:

$$\mathcal{A}/\mathcal{I}_{\mathbb{F}_2} \cong \mathcal{L}'(\mathcal{A})/\mathcal{L}'\left(\left\langle \mathcal{P}^{(h)} \cup \mathcal{B}^{(h)} \cup \mathcal{Q}^{(h)} \right\rangle\right) ,$$

where $\mathcal{I}_{\mathbb{F}_2} \stackrel{\text{def}}{=} \left\langle \mathcal{F}_{\mathbb{F}_2}^{(h)} \right\rangle$. They proposed Assumption 2, similar to Assumption 1, to derive the Hilbert series $\mathcal{H}_{\mathcal{A}/\mathcal{I}_{\mathbb{F}_2}}(z)$ for Model 2 as

$$\mathcal{H}_{\mathcal{A}/\mathcal{I}_{\mathbb{F}_2}}(z) = \left[\frac{(1+(\beta-1)\cdot z)^t}{(1+z)^{n-k}}\right]_+ .$$

Assumption 2 (see [BØ23])**.** *Consider an instance $\mathcal{F}_{\mathbb{F}_2}$ of Model 2 with degree of regularity d_{reg} for the ideal $\mathcal{I}_{\mathbb{F}_2}$. With $\mathcal{L}'$ defined in Equation (2), the quotient ring is given by $\mathcal{S}_{\mathbb{F}_2} = \mathcal{L}'(\mathcal{A})/\mathcal{L}'\left(\left\langle \mathcal{B}^{(h)} \cup \mathcal{Q}^{(h)} \right\rangle\right)$. For each parity-check equation $p_i \in \mathcal{P}$, let $p_i' = \mathcal{L}'\left(p_i^{(h)}\right)$. Assume that for $1 \le i \le n-k$, if $g \cdot p_i = 0$ in the quotient space $\mathcal{S}_{\mathbb{F}_2}/\left\langle p_1', \ldots, p_{i-1}' \right\rangle$ and $\deg(g)+\deg(p_i) < d_{\mathsf{reg}}$, then $g = 0$ in this quotient space.*

To clarify, while a simplified version is introduced above, Briaud and Øygarden [BØ23] adopt a hybrid approach (see Algorithm 2). This method starts with a noise-free guessing step before applying this simplified version. The guessing step leverages the sparsity of the noise vector to significantly reduce the witness degree d_{wit}, sometimes to as low as 2 in PCG parameters [BØ23, Table 2 and Table 3]. This reduction is crucial, as the size of the Macaulay matrix decreases exponentially with the degree d_{wit}. However, increasing the number of guessed noise-free positions reduces the probability of success, requiring more repetitions. Thus, AGB should carefully select parameters (f, μ) to minimize the running time. This guessing step assumes equations, such as $\boldsymbol{e}_{i,j} = 0$. The analysis and assumptions about the Hilbert series are similar to those in this simplified version.

APPLYING THE AGB TO MODEL 1 AND MODEL 2. The AGB algorithm [BØ23] involves guessing noise-free positions and applying the XL algorithm, see Algorithm 2. For each parameter pair (f, μ) with $0 \le f \le t$ and $0 \le \mu \le \beta$, guessing μ noise-free positions in the first f blocks introduces new equations $\mathcal{G}_{f,\mu} \stackrel{\text{def}}{=} \{\boldsymbol{e}_{i,j} = 0 : \boldsymbol{e}_{i,j} \text{ is guessed to be zero}\}$ into the polynomial system. However, instead of constructing these equations, we directly remove the monomial variables containing $\boldsymbol{e}_{i,j}$ and their corresponding columns in the Macaulay matrix, thereby reducing the problem size.

Briaud and Øygarden [BØ23] estimated the expected time of the AGB approach on RSD over large fields and $\mathbb{F}_2$, see Theorem 1 and Theorem 2. They used Hilbert series analysis to estimate the witness degree d_{wit} and the size of the Macaulay matrix in AGB. In Model 1, they simplified the Hilbert series $\mathcal{A}/\left(\left\langle \mathcal{B}^{(h)} \right\rangle \cup \langle \mathcal{G}_{f,\mu} \rangle\right)$ and $\mathcal{A}/(\mathcal{I} \cup \langle \mathcal{G}_{f,\mu} \rangle)$ as $\mathcal{A}/\left\langle \mathcal{B}^{(h)} \right\rangle$ and $\mathcal{A}/\mathcal{I}$, respectively. The witness degree $d_{\mathsf{wit}} = d(f,\mu)$ is computed as the index of the first non-positive coefficient in the series $\mathcal{H}_{\mathcal{A}/\mathcal{I},f,\mu}(z)/(1-z)$, since $[z^d]\mathcal{H}_{\mathcal{A}/\mathcal{I},f,\mu}(z)/(1-z)$ represents the difference between the columns and rank of the Macaulay matrix when shifted to degree d. $\mathcal{M}_{1,\le d(f,\mu)}^{f,\mu}$ is the count of monomials of degree $\le d(f,\mu)$ in this specialized system. Following AGB [BØ23], we determine the maximal number of non-zero terms in each row of the corresponding Macaulay matrix as $n_{f,\mu} \stackrel{\text{def}}{=} k+1-f\cdot\mu$ [8].

[8] Since $\mathbf{H}$ is uniformly random, it has full rank with high probability. By rearranging $\mathbf{H}$'s columns so that the first $n-k$ are linearly independent, we can apply Gaussian elimination to obtain its reduced row echelon form. This configuration reduces the non-zero terms in each row to $n_{f,\mu} \stackrel{\text{def}}{=} k+1-f\cdot\mu$.

Theorem 1 (The AGB over a Large Field [BØ23])**.** *The expected time of the* AGB *algorithm on Model 1 in terms of the number of* $\mathbb{F}$*-operations is estimated as*

$$\min_{\substack{0\le f\le t\\ 0\le \mu\le \beta}} \left(p(f,\mu)^{-1}\cdot 3\cdot n_{f,\mu}\cdot \left(\mathcal{M}^{f,\mu}_{1,\le d(f,\mu)}\right)^2\right) ,$$

where $p(f,\mu)\overset{def}{=}(1-\mu/\beta)^f$, $n_{f,\mu}\overset{def}{=}k+1-f\cdot\mu$,

$$\mathcal{M}^{f,\mu}_{1,\le d}\overset{def}{=}\sum_{i=0}^{d}[z^i]\mathcal{H}_{\mathcal{A}/\left\langle \mathcal{B}^{(h)}\right\rangle ,f,\mu}(z) ,$$

with the series

$$\mathcal{H}_{\mathcal{A}/\left\langle \mathcal{B}^{(h)}\right\rangle ,f,\mu}(z)=\left(1+(\beta-\mu)\cdot\frac{z}{1-z}\right)^f\cdot\left(1+\beta\cdot\frac{z}{1-z}\right)^{t-f} .$$

The index $d(f,\mu)$ *is the first non-positive coefficient in the series* $\mathcal{H}_{\mathcal{A}/\mathcal{I},f,\mu}(z)/(1-z)$*, where*

$$\mathcal{H}_{\mathcal{A}/\mathcal{I},f,\mu}(z)=\left[(1-z)^{n-k}\cdot\left(1+(\beta-\mu)\cdot\frac{z}{1-z}\right)^f\cdot\left(1+\beta\cdot\frac{z}{1-z}\right)^{t-f}\right]_+ .$$

The quotient ring is defined as $\mathcal{S}\overset{\text{def}}{=}\mathcal{L}'(\mathcal{A})\,/\mathcal{L}'\left(\left\langle \mathcal{B}^{(h)}\cup\mathcal{Q}^{(h)}\right\rangle\right)$. The guessing equations also become zero divisors in $\mathcal{S}$, but this can be resolved using a graded ring homomorphism that maps guessed variables to 0 while leaving others unchanged. Using Hilbert series analysis, the expected runtime of the AGB algorithm in Model 2 is given by Theorem 2.

Theorem 2 (The AGB over $\mathbb{F}_2$ [BØ23])**.** *The expected time of the* AGB *algorithm on Model 2, in terms of the number of* $\mathbb{F}_2$*-operations, is estimated as*

$$\min_{\substack{0\le f\le t\\ 0\le \mu\le \beta}} \left(p(f,\mu)^{-1}\cdot 3\cdot n_{f,\mu}\cdot \left(\mathcal{M}^{f,\mu}_{2,\le d(f,\mu)}\right)^2\right)$$

where $p(f,\mu)\overset{def}{=}(1-\mu/\beta)^f$, $n_{f,\mu}\overset{def}{=}k+1-f\cdot\mu$,

$$\mathcal{M}^{f,\mu}_{2,\le d}\overset{def}{=}\sum_{j=0}^{d}[z^j]\mathcal{H}_{\mathcal{S},f,\mu}(z) ,$$

the series $\mathcal{H}_{\mathcal{S},f,\mu}(z)\overset{def}{=}\mathcal{H}_{\mathcal{S}/\left\langle \mathcal{L}'(\mathcal{G}_{f,\mu})\right\rangle}$ *is shown to satisfy*

$$\mathcal{H}_{\mathcal{S},f,\mu}(z)=(1+(\beta-1-\mu)\cdot z)^f\cdot(1+(\beta-1)\cdot z)^{t-f} .$$

The index $d(f,\mu)$ *is the first non-positive coefficient in the series* $\mathcal{H}_{\mathcal{A}/\mathcal{I}_{\mathbb{F}_2},f,\mu}(z)/(1-z)$*, where* $\mathcal{H}_{\mathcal{A}/\mathcal{I}_{\mathbb{F}_2},f,\mu}(z)\overset{def}{=}\mathcal{H}_{\mathcal{A}/(\mathcal{I}_{\mathbb{F}_2}\cup\left\langle \mathcal{G}_{f,\mu}\right\rangle)}$ *and it is given by*

$$\mathcal{H}_{\mathcal{A}/\mathcal{I}_{\mathbb{F}_2},f,\mu}(z)=\left[\frac{(1+(\beta-1-\mu)\cdot z)^f\cdot(1+(\beta-1)\cdot z)^{t-f}}{(1+z)^{n-k}}\right]_+ .$$

3 AGB 2.0: a Refined Algebraic Attack

The AGB approach [BØ23] (refer to Section 2.5) involves two main steps: guessing the noise-free positions and applying the XL algorithm. Our AGB 2.0 refines this method by integrating the XL algorithm with a two-round position-guessing strategy. By iteratively guessing a small number of noise-free positions with higher success probability, the problem's dimension can be significantly reduced through Gaussian elimination.

To utilize previous results on the estimated witness degree d_{wit} and monomial count in [BØ23], we design the two-round guessing process such that after the i-th round, the cumulative result is equivalent to having guessed μ_i positions in each of the first f_i blocks of noise $\boldsymbol{e}$ for $i \in \{1, 2\}$. Given the RSD instance $(\mathbf{H}, \boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{e})$, our AGB 2.0 algorithm first computes the parameters $(f, \mu, f_1, \mu_1, d_{\mathsf{wit}})$ that optimize the expected running time, see Equations (3) or (4), similar to [BØ23]. The ranges are $0 \leq f_1 \leq f \leq t$ and $0 \leq \mu_1 \leq \mu \leq \beta$, with the estimated witness degree $d_{\mathsf{wit}} = d(f, \mu)$.

The remaining steps are detailed in Algorithm 3. It begins with the partial XL algorithm that involves shifting $\mathcal{F}$ to degree d_{wit} and constructing the Macaulay matrix $\mathbf{A}$ (line 1). Next, it guesses the noise-free positions using (f_1, μ_1) (line 3) with a success probability $p(f_1, \mu_1)$. Gaussian elimination is then used to reduce the problem's dimension. For noise-free guessing equations $\mathcal{G}_{f_1,\mu_1}$ or others except $\mathcal{P}$, we follow Section 2.5, as detailed in [BØ23], to remove the corresponding monomials and equations, resulting in the Macaulay matrix $\mathbf{A}$ of scale $\mathcal{M}^{f_1,\mu_1}_{i,\leq d(f,\mu)}$, with $i \in \{1, 2\}$, instead of the size of $\mathsf{Mon}_{\leq d(f,\mu)}$. In the second round, we guess the remaining noise-free positions using parameters (f, μ) (see lines 7) with a success probability $p(f, \mu)/p(f_1, \mu_1)$, where $p(f, \mu) \stackrel{\text{def}}{=} (1 - \mu/\beta)^f$. We then remove the corresponding monomials and complete the XL algorithm with Gaussian elimination.

In our AGB 2.0, repeated trials in the second round share the same guessed positions from the first round, resulting in correlation. However, this correlation is irrelevant since we only consider the expected running time. Our AGB 2.0 solves the RSD problem when both rounds of noise-free guesses are successful, allowing us to determine its expected time. The expected time of our AGB 2.0 is

$$T_{\mathsf{AGB2.0}} = \frac{T_1}{p(f_1, \mu_1)} + \frac{T_2}{p(f, \mu)} .$$

Here, T_1 and T_2 are the costs of the first and second Gaussian eliminations, respectively.

Our AGB 2.0 balances repetition count and dimension reduction compared to AGB. As f_1 and μ_1 increase, the dimension of the problem decreases noticeably. However, higher values of (f_1, μ_1) reduce the probability of success, which requires more repetitions. When $(f_1, \mu_1) = (f, \mu)$, our AGB 2.0 becomes AGB [BØ23], demonstrating that our method is a more flexible generalization. Specifically, we compare the two algorithms by setting the parameters f and μ in our AGB to be the same as the optimized parameters in AGB (refer to Footnote 2). Recall that $T_{\mathsf{AGB}} = T_{\mathsf{XL}}/p(f, \mu)$. In the first round, $T_1 \approx T_{\mathsf{XL}}$ because the Gaussian eliminations are of a similar scale, but $p(f_1, \mu_1) > p(f, \mu)$ since AGB 2.0 splits the guessing process. In the second round, $T_2 = o(T_{\mathsf{XL}})$ due to the dimension reduction from the first round. The intuition behind $T_{\mathsf{AGB2.0}} \leq T_{\mathsf{AGB}}$ is supported by the PCG parameter choices (see Table 2). Specifically, when $p(f, \mu) = o(1)$, we have $T_{\mathsf{AGB2.0}} = o\left(T_{\mathsf{AGB}}\right)$, see analysis in Equation (1).

[9] At this stage, $\mathbf{I}'$ may not be an identity matrix, as it is formed by removing some columns (corresponding to the

Algorithm 3: The AGB 2.0

Input: A polynomial system $\mathcal{F}$ associated with Model 1 or Model 2 and parameters $(f, \mu, f_1, \mu_1, d_{\mathsf{wit}})$.

Output: A solution to the corresponding polynomial system.

Calculate the shift of $\mathcal{F}$ to degree d_{wit} as $\mathcal{I}_{\leq d_{\mathsf{wit}}}$, and the corresponding Macaulay matrix $\mathbf{A} = \mathsf{Mac}\left(\mathcal{I}_{\leq d_{\mathsf{wit}}}, \mathsf{Mon}_{\leq d_{\mathsf{wit}}}\right)$; the equation system is now $\mathbf{A} \cdot \boldsymbol{e}_{\mathsf{Mon}_{\leq d_{\mathsf{wit}}}} = \mathbf{0}$;

repeat $1/p(f_1, \mu_1)$ **times**

  Guess μ_1 noise-free positions in each of the first f_1 blocks of $\boldsymbol{e}$;

  Remove the monomials in $\boldsymbol{e}_{\mathsf{Mon}_{\leq d_{\mathsf{wit}}}}$ and their columns in $\mathbf{A}$ (related to the guessed noise-free positions) to obtain a simplified equation $\mathbf{A}' \cdot \boldsymbol{e}_{r1} = \mathbf{0}$;

  Apply Gaussian elimination to reduce the equation to the form $\begin{bmatrix}\mathbf{I} & \mathbf{C}_{r1}\end{bmatrix} \cdot \boldsymbol{e}'_{r1} = \mathbf{0}$, where $\boldsymbol{e}'_{r1}$ is a permutation of $\boldsymbol{e}_{r1}$;

  repeat $p(f_1, \mu_1)/p(f, \mu)$ **times**

    Guess another $\mu - \mu_1$ noise-free positions in each of the first f_1 blocks of $\boldsymbol{e}$; and guess μ positions in the i-th blocks of $\boldsymbol{e}$, for each $f_1 < i \leq f$;

    Remove the monomials in $\boldsymbol{e}'_{r1}$ and their columns in $\begin{bmatrix}\mathbf{I} & \mathbf{C}_{r1}\end{bmatrix}$ (related to the guessed noise-free positions) to obtain $\begin{bmatrix}\mathbf{I}' & \mathbf{C}'_{r1}\end{bmatrix} \cdot \boldsymbol{e}_{r2} = \mathbf{0}$; and by switching the rows, rewrite it as $\begin{bmatrix}\mathbf{I} & \mathbf{C}^1_{r2} \\ \mathbf{0} & \mathbf{C}^2_{r2}\end{bmatrix} \cdot \begin{bmatrix}\boldsymbol{e}^1_{r2} \\ \boldsymbol{e}^2_{r2}\end{bmatrix} = \mathbf{0}$, namely $\boldsymbol{e}^1_{r2} + \mathbf{C}^1_{r2} \cdot \boldsymbol{e}^2_{r2} = \mathbf{0}$ and $\mathbf{C}^2_{r2} \cdot \boldsymbol{e}^2_{r2} = \mathbf{0}$ [9];

    Apply Gaussian elimination to solve the equation system $\mathbf{C}^2_{r2} \cdot \boldsymbol{e}^2_{r2} = \mathbf{0}$;

    If a solution is found, compute $\boldsymbol{e}^1_{r2} = -\mathbf{C}^1_{r2} \cdot \boldsymbol{e}^2_{r2}$, quit all loops and return;

Theorem 3 and Theorem 4 describe the runtime of our AGB 2.0 for solving the RSD problem over a larger field and the binary field $\mathbb{F}_2$, respectively. Theorem 4 follows a similar approach to the proof of Theorem 3.

Theorem 3 (The AGB 2.0 over a Large Field)**.** *The expected time of the* AGB 2.0 *on Model 1, in terms of* $\mathbb{F}$*-operations, is*

$$\min_{\substack{0 \leq f_1 \leq f \leq t \\ 0 \leq \mu_1 \leq \mu \leq \beta}} \left(\frac{T_1(f_1, \mu_1, d(f, \mu))}{p(f_1, \mu_1)} + \frac{T_2(f_1, \mu_1, d(f, \mu))}{p(f, \mu)} \right) , \tag{3}$$

where the success probability $p(f, \mu)$*, the maximal number of non-zero terms* n_{f_1, μ_1}*, the witness degree* $d(f, \mu)$*, the column count of the Macaulay matrix* $\mathcal{M}^{f_1, \mu_1}_{1, \leq d(f, \mu)}$ *and the series* $\mathcal{H}_{\mathcal{A}/\mathcal{I}, f, \mu}(z)$ *are defined in Theorem 1. The column count of the Macaulay matrix is given by*

$$\mathcal{N}^{f_1, \mu_1}_{1, \leq d(f, \mu)} = \sum_{j=0}^{d(f, \mu)} [z^j] \mathcal{H}_{\mathcal{A}/\mathcal{I}, f_1, \mu_1}(z) \, .$$

guessed noise-free positions) from the identity matrix. The vectors $\boldsymbol{e}^1_{r2}$ and $\boldsymbol{e}^2_{r2}$ together form a permutation of the vector $\boldsymbol{e}_{r2}$.

The running time of the Gaussian elimination is

$$T_1(f_1,\mu_1,d(f,\mu)) = 5.64 \cdot \left(\mathcal{M}^{f_1,\mu_1}_{1,\leq d(f,\mu)}\right)^{2.8}$$
$$T_2(f_1,\mu_1,d(f,\mu)) = 5.64 \cdot \left(\mathcal{N}^{f_1,\mu_1}_{1,\leq d(f,\mu)}\right)^{2.8} .$$

Proof. Similarly to the proof of AGB [BØ23], $\mathcal{M}^{f_1,\mu_1}_{1,\leq d(f,\mu)}$ represents the count of monomials related to the columns of $\mathbf{A}'$, while $\mathcal{N}^{f_1,\mu_1}_{1,\leq d(f,\mu)}$ represents the count of equations related to the columns of $\mathbf{C}_{r1}$, which is also the difference between the number of columns of $\mathbf{A}'$ and its rank. The size of $\mathbf{C}^2_{r2}$ is bounded by the number of columns of $\mathbf{C}_{r1}$.

To analyze the expected runtime, we consider the number of iterations and the time for each key step. The first guess (line 3) and the combined guesses (lines 3 and 7) succeed with probabilities $p(f_1,\mu_1)$ and $p(f,\mu)$ respectively. Thus, the conditional probability of the second guess given the first is $p(f,\mu)/p(f_1,\mu_1)$. We take the reciprocal of these probabilities to determine the number of repetitions needed for a constant success probability.

The matrix $\mathbf{A}'$ has dimensions

$$\left(\mathcal{M}^{f_1,\mu_1}_{1,\leq d(f,\mu)} - \mathcal{N}^{f_1,\mu_1}_{1,\leq d(f,\mu)}\right) \times \mathcal{M}^{f_1,\mu_1}_{1,\leq d(f,\mu)} ,$$

and each polynomial has at most n_{f_1,μ_1} terms. The expected time for the first Gaussian elimination is:

$$T_1(f_1,\mu_1,d(f,\mu)) = 5.64 \cdot \left(\mathcal{M}^{f_1,\mu_1}_{1,\leq d(f,\mu)}\right)^{2.8} .$$

Following [BH74], the expected runtime for the second Gaussian elimination is:

$$T_2(f_1,\mu_1,d(f,\mu)) = 5.64 \cdot \left(\mathcal{N}^{f_1,\mu_1}_{1,\leq d(f,\mu)}\right)^{2.8} .$$

Here, the dimension of $\mathbf{C}^2_{r2}$ is bounded by

$$\left(\mathcal{M}^{f_1,\mu_1}_{1,\leq d(f,\mu)} - \mathcal{N}^{f_1,\mu_1}_{1,\leq d(f,\mu)}\right) \times \mathcal{N}^{f_1,\mu_1}_{1,\leq d(f,\mu)} .$$

□

Theorem 4 (The AGB 2.0 over $\mathbb{F}_2$)**.** *The expected time of the* AGB 2.0 *on Model 2, in terms of* $\mathbb{F}_2$*-operations, is*

$$\min_{\substack{0\leq f_1\leq f\leq t\\ 0\leq \mu_1\leq \mu\leq \beta}} \left(\frac{T_1(f_1,\mu_1,d(f,\mu))}{p(f_1,\mu_1)} + \frac{T_2(f_1,\mu_1,d(f,\mu))}{p(f,\mu)}\right) , \tag{4}$$

where the success probability $p(f,\mu)$, the maximal number of non-zero terms n_{f_1,μ_1}, the witness degree $d(f,\mu)$, the column count of the Macaulay matrix $\mathcal{M}^{f_1,\mu_1}_{2,\leq d(f,\mu)}$ and the series $\mathcal{H}_{\mathcal{A}/\mathcal{I}_{\mathbb{F}_2},f,\mu}(z)$ are defined in Theorem 2. The column count of the Macaulay matrix is given by

$$\mathcal{N}^{f_1,\mu_1}_{2,\leq d(f,\mu)} = \sum_{j=0}^{d(f,\mu)} [z^j]\mathcal{H}_{\mathcal{A}/\mathcal{I}_{\mathbb{F}_2},f_1,\mu_1}(z) .$$

The running time of the Gaussian elimination is

$$T_1(f_1, \mu_1, d(f, \mu)) = 5.64 \cdot \left(\mathcal{M}^{f_1,\mu_1}_{1,\leq d(f,\mu)}\right)^{2.8}$$
$$T_2(f_1, \mu_1, d(f, \mu)) = 5.64 \cdot \left(\mathcal{N}^{f_1,\mu_1}_{2,\leq d(f,\mu)}\right)^{2.8}.$$

We do not extend to three or more rounds because an additional round only slightly reduces bit security for commonly used PCG parameters. After the second round of Algorithm 3 (lines 7 to 9), adding another round would involve guessing the noise-free position using parameters (f_2, μ_2) and applying Gaussian elimination to solve the system, where $f_1 \leq f_2 \leq f$ and $\mu_1 \leq \mu_2 \leq \mu$. However, Table 3 shows that the optimal parameters (f, μ) and (f_1, μ_1) are nearly identical, leaving few options for (f_2, μ_2). Therefore, the three-round optimization provides only a slight improvement in bit security for commonly used PCG parameters compared to the two-round method, as our calculations confirm.

Our AGB 2.0 uses the same assumptions as the AGB [BØ23], enabling us to rely on existing AGB experiments to support our claims without requiring additional experiments. Specifically, our AGB 2.0 algorithm splits the AGB process into two distinct rounds, each with its own parameter set: (f, μ) and (f_1, μ_1). Thus, our AGB 2.0 requires two separate AGB experiments for each parameter set. Similar assumptions form the basis of many algebraic attacks. For instance, the well-known Arora-Ge system [AG11] is employed to solve LWE and is generally assumed to be semi-regular [ACF+12, STA20]. Experiments conducted by Albrecht et al. [ACFP14] have confirmed this assumption.

3.1 Asymptotic Analysis

We demonstrate a theoretical analysis of plain AGB (that is, without noise-free guessing, with $f = f_1 = \mu = \mu_1 = 0$ in our AGB 2.0). Briaud and Øygarden [BØ23] proposed an asymptomatic complexity for plain AGB using the Gröbner basis method [Fau99, Fau02, YCC04, ST21] to solve the binary RSD problem, as stated in Theorem 7. We use a similar approach to evaluate the complexity of the plain AGB to solve the RSD problem over larger fields when $\rho + R < 1$, where ρ is the noise rate and R is the code rate. This analysis, detailed in Theorem 5, yields two important findings. First, if $n\rho R^2 = O(1)$, where n is the noise length, the RSD problem over a larger field can be solved in polynomial time, as stated in Corollary 1; however, this does not hold for the SD setting. Second, under the conditions $R = o(1/\log(n))$ and $|\mathbb{F}| > e^{n\rho R}$, the plain AGB is asymptotically more efficient than the ISD and its variants for addressing the RSD problem, as demonstrated in Theorem 6, where $|\mathbb{F}|$ is the field size. The analysis of plain AGB extends directly to the case of noise-free position guessing with $f = t$, as the problem still follows model 1 or model 2 after guessing, as shown in Corollary 2 and Corollary 3.

To analyze the asymptotic performance of the plain AGB, it is important to consider the asymptotic value of d_{reg}. Techniques such as the Gröbner basis method can yield improved asymptotic results [Gal14]. Following the approach outlined in [BØ23], we apply the saddle-point method to analyze the asymptotic behavior of an integral for a specific d. Additional details on this method in the context of Hilbert series can be found in [Bar04a, Chapter 4]. We compute each

coefficient in the series using a Cauchy integral:

$$\left[z^d\right] \mathcal{H}_{\mathcal{A}/\mathcal{I}}(z) = \frac{1}{2i\pi} \cdot \oint \frac{1}{z^{d+1}} \cdot \mathcal{H}_{\mathcal{A}/\mathcal{I}}(z)\,\mathrm{d}z \ .$$

Using the asymptotic analysis of d_{reg}, we derive the time complexity of plain AGB via the Gröbner basis method. As shown by Semaev and Tenti [ST21], the complexity of T_{AGB} is given by:

$$T_{\mathsf{AGB}} = O\left(\mathcal{M}_{d_{\mathsf{reg}}}^2 \cdot \mathcal{M}_{d_{\mathsf{reg}}-1}^2 \cdot \mathcal{M}_{2d_{\mathsf{reg}}-2}\right) \ ,$$

where $\mathcal{M}_d$, the number of monomials up to degree d, is

$$\mathcal{M}_d = O\left(d \cdot \left(\frac{n}{t}\right)^d \cdot \binom{t}{d} \cdot 2^d\right) = O\left((2n)^d\right) \ .$$

In Theorem 5 and Theorem 7, when $R = o(1)$ and $\rho = o(1)$, the estimation of d_{reg}, $O(n\rho R^2) - 1$, may become negative. In such cases, we demonstrate that the plain AGB algorithm remains polynomial-time because we can increase the estimation of d_{reg} to a positive constant by raising the code rate $R = k/n$ through discarding a sufficient number of parity check equations, while keeping the noise $\boldsymbol{e}$ unchanged.

Theorem 5 (The asymptotic value of d_{reg} for Model 1)**.** *Let $R + \rho < 1$. When $n \to +\infty$, the degree of regularity d_{reg} for Model 1 asymptotically follows:*

1. *For constant code rate R, we obtain*

$$d_{\mathsf{reg}} \sim n\rho \cdot \left(\sqrt{1-\rho} - \sqrt{1-R-\rho}\right)^2 \ .$$

Especially, if noise rate $\rho = o(1)$,

$$d_{\mathsf{reg}} \sim n\rho \cdot \left(2 - R - 2\sqrt{1-R}\right) \ .$$

2. *For code rate $R = o(1)$, we obtain*

$$d_{\mathsf{reg}} + 1 \sim \frac{n\rho R^2}{4(1-\rho)} \ .$$

Proof. We first calculate the witness degree d_{reg} using the Cauchy integral

$$\mathcal{I}_d(n) = \frac{1}{2i\pi} \cdot \int \underbrace{\frac{1}{z^{d+1}} \cdot (1-z)^{n-k} \cdot \left(1 + \beta \cdot \frac{z}{1-z}\right)^t}_{=e^{n \cdot f(z)}} \mathrm{d}z \ ,$$

where $\beta = n/t$ and $f(z) = -\frac{d+1}{n} \cdot \ln(z) + (1-R) \cdot \ln(1-z) + \rho \cdot \ln\left(1 + \rho^{-1} \cdot z/(1-z)\right)$. Then, we identify the saddle points as solutions to the equation:

$$zf'(z) = -\frac{d+1}{n} - (1-R) \cdot \frac{z}{1-z} + \frac{\rho z}{(\rho - \rho z + z) \cdot (1-z)} = 0 \ . \tag{5}$$

By multiplying $(\rho - \rho z + z) \cdot (1 - z) \cdot n$ on both sides, Equation (5) can be rewritten as a quadratic equation $P(z) = p_2 \cdot z^2 + p_1 \cdot z + p_0 = 0$, where

$$\begin{aligned} p_2 &= -(1-\rho) \cdot (n - nR - d - 1) \\ p_1 &= \rho R n + (d+1)(2\rho - 1) \\ p_0 &= -\rho \cdot (d+1). \end{aligned}$$

A double root of $P(z) = 0$ indicates coalescing saddle points; otherwise, the integral grows exponentially, as described in [Bar04b, p.94] and [ACF+15]. This leads to the condition $p_1^2 = 4p_0p_2$, which gives a quadratic equation $q_2 \cdot d^2 + q_1 \cdot d + q_0 = 0$, where

$$\begin{aligned} q_2 &= 1 \\ q_1 &= -2n\rho \cdot (2 - 2\rho - R) + 2 \\ q_0 &= R^2\rho^2n^2 + 2n\rho \cdot (2\rho + R - 2) + 1 \, . \end{aligned}$$

Solving this equation, we obtain that

$$d = n\rho \cdot (2 - 2\rho - R) - 1 \pm 2n\rho\sqrt{(1-\rho)(1-R-\rho)} \, .$$

Note that d_{reg} is approximated by the smallest root. The proof concludes by examining the equation above in different scenarios.

1. For constant code rate R, we have

$$\begin{aligned} d_{\mathsf{reg}} &\sim n\rho \cdot (2 - 2\rho - R) - 2n\rho\sqrt{(1-\rho)(1-R-\rho)} \\ &\sim n\rho \cdot \left(\sqrt{1-\rho} - \sqrt{1-R-\rho}\right)^2 \, . \end{aligned}$$

   Especially, if noise rate $\rho = o(1)$,

$$d_{\mathsf{reg}} \sim n\rho \cdot \left(2 - R - 2\sqrt{1-R}\right) \, .$$

2. For $R = o(1)$, we have

$$\begin{aligned} d_{\mathsf{reg}} + 1 &\sim n\rho \cdot (2 - 2\rho - R) - 2n\rho\sqrt{(1-\rho)(1-R-\rho)} \\ &\sim n\rho \cdot (2 - 2\rho - R) - 2n\rho(1-\rho)\sqrt{1 - \frac{R}{1-\rho}} \sim \frac{n\rho R^2}{4(1-\rho)} \, , \end{aligned}$$

   since

$$\sqrt{1 - \frac{R}{1-\rho}} = \left(1 - \frac{R}{2(1-\rho)} - \frac{R^2}{8(1-\rho)^2} + o(R^2)\right) \, .$$

□

By applying Theorem 5 and the time complexity $T = n^{O(d_{\mathsf{reg}})}$ from the Gröbner basis method by Semaev and Tenti [ST21], we derive Corollary 1.

Corollary 1. *The time complexity of the plain* AGB *for solving the RSD problem is* $\mathsf{poly}(n)$ *using the Gröbner basis method, given that* $n\rho R^2 = O(1)$ *and* $R + \rho < 1$ *is a positive constant, where* n *is the noise length,* R *is the code rate, and* ρ *is the noise rate.*

Theorem 6. *The ISD method and its variants, including regular-ISD [ES24] and regular-RP [KL24], are asymptotically higher costs than the plain* AGB *method, which employs the Gröbner basis method, to solve the RSD problem with noise length* n*, noise rate* ρ*, code rate* $R = o(1/\log(n))$*, and field size* $|\mathbb{F}| > e^{n\rho R}$*.*

Proof. We start by evaluating the complexity of the ISD framework when applied to the RSD problem, where the field size $|\mathbb{F}| > e^{n\rho R}$. In such cases, the ISD method transitions to the Pooled Gauss approach due to the prohibitive time required to search for a nonzero element in $|\mathbb{F}|$, leading to a complexity of:

$$T_{\mathsf{ISD}} \geq \left(1 - \frac{t}{n}\right)^{-k} = (1-\rho)^{-nR} \approx e^{kt/n} = e^{n\rho R}\,,$$

as noted in the paper [LWYY24].

Next, we assess the plain AGB's running time. The time complexity $T_{\mathsf{AGB}} = n^{O(d_{\mathsf{reg}})}$ follows from the Gröbner basis method by Semaev and Tenti [ST21]. Based on Theorem 5, as $R = o(1)$ and $R + \rho < 1$, we have $d_{\mathsf{reg}} = O(n\rho R^2)$. Therefore, the time complexity is $T_{\mathsf{AGB}} = n^{O(d_{\mathsf{reg}})} = 2^{O\left(n\rho R^2 \log(n)\right)}$.

Finally, we show that $n\rho R^2 \log(n) = o(n\rho R)$, since $R = o(1/\log(n))$, thus completing the proof. □

After guessing the μ noise-free positions in each block, we obtain a new model 2 with noise length $n' = n - t\mu$, dimension $k' = k - t\mu$, code rate $R' = \frac{R-\rho\mu}{1-\rho\mu}$, and noise rate $\rho' = \frac{\rho}{1-\rho\mu}$. Applying Theorem 5 with n', R', and ρ', we derive Corollary 2.

Corollary 2. *For a constant code rate* R *and a constant error rate* ρ*, where* $R + \rho < 1$*, when* $n \to +\infty$*, an equivalent of the degree of regularity* $d_{\mathsf{reg}}(u)$ *of a specialized system obtained from Model 1 by guessing* $0 \leq u < R/\rho - 1$ *error-free positions in each block is*

$$d_{\mathsf{reg}}(u) \sim \frac{n\rho}{(1-\rho\mu)} \cdot \left(\sqrt{1-\rho(\mu+1)} - \sqrt{1-R-\rho}\right)^2\,.$$

Theorem 7 (The asymptotic value of d_{reg} for Model 2 [BØ23])**.** *Let* $\rho < R$*. When* $n \to +\infty$*, the degree of regularity* d_{reg} *for Model 2 asymptotically follows:*

1. *For constant code rate* R*, we have*

$$d_{\mathsf{reg}} \sim \frac{n\rho \cdot \left(\sqrt{1-\rho} - \sqrt{1-R}\right)^2}{1-2\rho}$$

Especially, if noise rate $\rho = o(1)$*,*

$$d_{\mathsf{reg}} \sim n\rho\left(2 - R - 2\sqrt{1-R}\right)\,.$$

2. *For code rate $R = o(1)$, we have*

$$d_{\mathsf{reg}} + 1 \sim \frac{(1-\rho/R)^2 n\rho R^2}{4} .$$

Especially, if noise rate $\rho = o(R)$,

$$d_{\mathsf{reg}} + 1 \sim \frac{n\rho R^2}{4} .$$

Corollary 3 (see [BØ23])**.** *For a constant code rate R and a constant error rate ρ, when $n \to +\infty$, an equivalent of the degree of regularity $d_{\mathsf{reg}}(\mu)$ of a specialized system obtained from Model 2 by guessing $0 \leq \mu \leq R/\rho - 2$ error-free positions in each block is*

$$d_{\mathsf{reg}}(u) \sim \frac{n\rho}{1-\rho(\mu+2)} \cdot \left(\sqrt{1-\rho(\mu+1)} - \sqrt{1-R}\right)^2 .$$

4 Application to Concrete Parameter Sets

To assess the computational cost of the AGB 2.0 for specific RSD instances, we developed a script to automatically find optimal parameters (f, μ, f_1, μ_1) that minimize time costs (refer to Footnote 3). Similar to the AGB [BØ23], the advantage increases with more parity-check equations (i.e., a lower code rate). Since we are particularly interested in the RSD problem for PCG applications, we compare our AGB 2.0 with previous RSD attacks [LWYY24, BØ23, CCJ23, ES24] using parameters from PCG-related works [BCGI18, LWYY24, YWL+20, WYKW21] in Table 2, Table 4 and Table 5.

Boyle et al. [BCGI18] initially chose SD parameters for 80-bit security in PCG applications, which subsequent studies have re-evaluated. Liu et al. [LWYY24] then investigated the RSD problem and proposed suggested parameters for 128-bit security. Briaud and Øygarden [BØ23] compared their AGB approach with the analysis by Liu et al. [LWYY24], focusing on parameter selections from various studies [BCGI18, YWL+20, WYKW21, LWYY24]. Esser and Santini [ES24] introduced regular-ISD and compared it with previous research [LWYY24, BØ23, CCJ23, EB22] under some parameter sets, including Tinykey [HOSS18], PCG-related works [BCGI18, YWL+20, WYKW21, LWYY24] and the signature scheme [CCJ23]. In concurrent and independent work [KL24], Kim and Lee introduced the regular Reduce and Prange (regular-RP) algorithm, effective for solving regular SD over larger fields, such as $|\mathbb{F}| = 2^{128}$.

As shown in Table 2, the AGB 2.0 algorithm has advantages over previous methods, particularly for RSD instances with field size $\log |\mathbb{F}| = 128$, as considered in [BCGI18, LWYY24]. The parameter choices in [LWYY24] are similar to those in [BCGI18], but with larger noise weights for higher security. The optimal parameters (f, f_1, μ, μ_1) are listed in Table 3. In particular, our AGB 2.0 reduces the estimated time complexity by up to 6 bits for these instances. In Table 4, our AGB 2.0 shows advantages over previous attacks for some RSD parameter sets used by Wolverine [WYKW21]. Wolverine's 128-bit secure parameter choice $(n = 642048, k = 19870, t = 2508)$ is estimated to offer only 121 bit-security.

Table 2: A comparison of our AGB 2.0 algorithm with prior approaches, including ISD [MMT11, BJMM12], AGB [BØ23], CCJ [CCJ23], Regular-ISD (RISD) [ES24] and the concurrent and independent Regular-RP (RRP) [KL24], for assessing the bit-security of RSD instances over $\mathbb{F}$ with noise length n, dimension k, and Hamming weight t of a regular noise.

Regular SD			$(\log \lvert\mathbb{F}\rvert = 128)$				$\mathbb{F} = \mathbb{F}_2$					
n	k	t	ISD	AGB	RRP	AGB 2.0	ISD	AGB	CCJ	RISD	RRP	AGB 2.0
2^{10}	652	106	194	179	150	176	159	145	129	113	117	139
2^{12}	1589	172	155	150	136	143	121	135	160	109	120	128
2^{14}	3482	338	150	150	137	142	122	138	204	118	125	130
2^{16}	7391	667	151	150	141	140	125	139	249	126	130	129
2^{18}	15336	1312	153	133	147	127	129	122	274	126	137	117
2^{20}	32771	2467	155	131	148	127	135	125	335	138	139	119
2^{22}	64770	4788	156	110	151	110	135	103	360	140	143	103
2^{10}	652	57	111	111	94	108	90	101	90	76	84	98
2^{12}	1589	98	100	107	89	102	80	103	115	78	84	98
2^{14}	3482	198	101	110	93	104	83	106	143	84	88	100
2^{16}	7391	389	103	111	97	104	87	108	171	90	92	101
2^{18}	15336	760	105	107	100	101	92	104	192	93	97	101
2^{20}	32771	1419	107	102	103	102	97	98	216	98	100	98
2^{22}	64770	2735	108	104	105	103	99	103	233	103	102	103

SPEEDING UP PARAMETERS f, f_1, μ, μ_1 SEARCHING. Based on Theorem 3 and Theorem 4, our AGB 2.0 requires optimized parameters f, f_1, μ, and μ_1 to minimize runtime, where $0 \leq f_1 \leq f \leq t$ and $0 \leq \mu_1 \leq \mu \leq \beta$. This process is time-consuming, so we propose a simpler approximate method.

We first calculate the witness degree $d(0,0)$ for AGB 2.0 without guessing the noise-free position, where $f = f_1 = \mu = \mu_1 = 0$. Next, we introduce noise-free position guessing, where $d(f,\mu)$ must be smaller than $d(0,0)$ for optimality, as guessing the noise-free position leads to a smaller success probability. Our method then identifies optimized parameter candidates (f,μ) followed by (f_1,μ_1), generating candidate pairs $(f^i,\mu^i,d^i_{\mathsf{wit}})$. For each candidate $(f^i,\mu^i,d^i_{\mathsf{wit}})$, d^i_{wit} represents the witness degree obtained using the parameters f^i and μ^i. For distinct candidates $(f^i,\mu^i,d^i_{\mathsf{wit}})$ and $(f^j,\mu^j,d^j_{\mathsf{wit}})$, if $d^i_{\mathsf{wit}} = d^j_{\mathsf{wit}}$, it is impossible for both $f^i \leq f^j$ and $\mu^i \leq \mu^j$ to hold. This is because (f^i,μ^i) performs better with a higher success rate and similar Macaulay matrix dimensions. Thus, our method works as follows: for each f^i and d^i_{wit}, we use binary search to select the smallest μ^i such that the witness degree matches d^i_{wit} after guessing with parameters f^i and μ^i. After finding the optimized parameter candidates for (f,μ), we apply the same method to determine the optimal (f_1,μ_1) to minimize runtime.

Table 3: The optimal parameter (f, f_1, μ, μ_1) of our AGB 2.0 assessing the bit-security of RSD problems over $\mathbb{F}$ with noise length n, dimension k, and Hamming weight t for a regular noise distribution.

Regular SD			$(\log \lvert\mathbb{F}\rvert = 128)$		$(\mathbb{F} = \mathbb{F}_2)$	
n	k	t	(f, μ)	(f_1, μ_1)	(f, μ)	(f_1, μ_1)
2^{10}	652	106	$(105, 5)$	$(89, 5)$	$(88, 5)$	$(75, 5)$
2^{12}	1589	172	$(156, 8)$	$(148, 7)$	$(154, 7)$	$(143, 6)$
2^{14}	3482	338	$(324, 8)$	$(324, 6)$	$(322, 7)$	$(321, 5)$
2^{16}	7391	667	$(612, 8)$	$(539, 6)$	$(604, 7)$	$(513, 5)$
2^{18}	15336	1312	$(1216, 6)$	$(1174, 3)$	$(1197, 5)$	$(1170, 2)$
2^{20}	32771	2467	$(2365, 5)$	$(2350, 2)$	$(1181, 8)$	$(1166, 2)$
2^{22}	64770	4788	$(3688, 2)$	$(3672, 2)$	$(0, 0)$	$(0, 0)$
2^{10}	652	57	$(55, 10)$	$(53, 9)$	$(55, 9)$	$(52, 8)$
2^{12}	1589	98	$(95, 13)$	$(91, 10)$	$(95, 12)$	$(89, 9)$
2^{14}	3482	198	$(197, 12)$	$(197, 7)$	$(197, 11)$	$(197, 6)$
2^{16}	7391	389	$(375, 11)$	$(328, 5)$	$(375, 10)$	$(320, 4)$
2^{18}	15336	760	$(180, 30)$	$(55, 6)$	$(150, 31)$	$(45, 6)$
2^{20}	32771	1419	$(1256, 3)$	$(1236, 3)$	$(1159, 2)$	$(1144, 2)$
2^{22}	64770	2735	$(0, 0)$	$(0, 0)$	$(0, 0)$	$(0, 0)$

Table 4: A comparison of our AGB 2.0 algorithm with prior algorithms for estimated bit-security of RSD instances over a finite field $\mathbb{F}$.

Regular SD			$\mathbb{F} = \mathbb{F}_{2^{61}-1}$		
n	k	t	ISD [LWYY24]	AGB [BØ23]	AGB 2.0
642048	19870	2508	154	126	<u>121</u>
10805248	589760	1319	<u>160</u>	164	184

Table 5: A comparison of our AGB 2.0 algorithm with prior algorithms for assessing the bit-security of RSD instances over $\mathbb{F}_2$.

Regular SD			$\mathbb{F} = \mathbb{F}_2$				
n	k	t	ISD [LWYY24]	AGB [BØ23]	CCJ [CCJ23]	RISD [ES24]	AGB 2.0
609728	36288	1269	138	142	343	140	<u>134</u>
10805248	589760	1319	<u>151</u>	164	492	157	183

References

[ACF+12] Martin Albrecht, Carlos Cid, Jean-Charles Faugere, Robert Fitzpatrick, and Ludovic Perret. On the complexity of the arora-ge algorithm against lwe. In *SCC 2012–Third international conference on Symbolic Computation and Cryptography*, pages 93–99,

2012.

[ACF+15] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Algebraic algorithms for LWE problems. *ACM Commun. Comput. Algebra*, 49(2):62, 2015.

[ACFP14] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, and Ludovic Perret. Algebraic algorithms for LWE. Cryptology ePrint Archive, Report 2014/1018, 2014.

[AFS05] Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A family of fast syndrome based cryptographic hash functions. In Ed Dawson and Serge Vaudenay, editors, *Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 64–83. Springer, 2005.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP 2011, Part I*, volume 6755 of *LNCS*, pages 403–415. Springer, Berlin, Heidelberg, July 2011.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307. IEEE Computer Society Press, October 2003.

[Ant22] Ant Group. SecretFlow: A Unified Trusted Privacy-Preserving Computing Framework. https://github.com/secretflow, 2022.

[AS22] Damiano Abram and Peter Scholl. Low-communication multiparty triple generation for SPDZ from ring-LPN. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 221–251. Springer, Cham, March 2022.

[Bar04a] Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2004.

[Bar04b] Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Pierre and Marie Curie University, Paris, France, 2004.

[BBC+24] Maxime Bombar, Dung Bui, Geoffroy Couteau, Alain Couvreur, Clément Ducros, and Sacha Servan-Schreiber. FOLEAGE: $\mathbb{F}_4$OLE-based multi-party computation for boolean circuits. Cryptology ePrint Archive, Report 2024/429, 2024.

[BC23] Dung Bui and Geoffroy Couteau. Improved private set intersection for sets with small entries. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 190–220. Springer, Cham, May 2023.

[BCC+24] Dung Bui, Eliana Carozza, Geoffroy Couteau, Dahmun Goudarzi, and Antoine Joux. Short signatures from regular syndrome decoding, revisited. Cryptology ePrint Archive, Report 2024/252, 2024.

[BCCD23a] Maxime Bombar, Geoffroy Couteau, Alain Couvreur, and Clément Ducros. Correlated pseudorandomness from the hardness of quasi-abelian decoding. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 567–601. Springer, Cham, August 2023.

[BCCD23b] Maxime Bombar, Geoffroy Couteau, Alain Couvreur, and Clément Ducros. Correlated pseudorandomness from the hardness of quasi-abelian decoding. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 567–601. Springer Nature Switzerland, 2023.

[BCG+19a] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Peter Rindal, and Peter Scholl. Efficient two-round OT extension and silent non-interactive secure computation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 291–308. ACM Press, November 2019.

[BCG+19b] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 489–518. Springer, Cham, August 2019.

[BCG+20] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators from ring-LPN. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 387–416. Springer, Cham, August 2020.

[BCG+22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 603–633. Springer, Cham, August 2022.

[BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 896–912. ACM Press, October 2018.

[Beu21] Ward Beullens. Improved cryptanalysis of UOV and Rainbow. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 348–373. Springer, Cham, October 2021.

[BFP09] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *J. Math. Cryptol.*, 3(3):177–197, 2009.

[BFSS13] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic boolean systems. *J. Complex.*, 29(1):53–75, 2013.

[BH74] James R. Bunch and John E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 1974.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How 1 + 1 = 0 improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 520–536. Springer, Berlin, Heidelberg, April 2012.

[BM18] Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 25–46. Springer, Cham, 2018.

[BMRS21] Carsten Baum, Alex J. Malozemoff, Marc B. Rosen, and Peter Scholl. Mac'n'cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 92–122, Virtual Event, August 2021. Springer, Cham.

[BØ23] Pierre Briaud and Morten Øygarden. A new algebraic approach to the regular syndrome decoding problem and implications for PCG constructions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 391–422. Springer, Cham, April 2023.

[CC98] Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Trans. Inf. Theory*, 44(1):367–378, 1998.

[CCJ23] Eliana Carozza, Geoffroy Couteau, and Antoine Joux. Short signatures from regular syndrome decoding in the head. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 532–563. Springer, Cham, April 2023.

[CCNY12] Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, and Bo-Yin Yang. Solving quadratic equations with XL on parallel architectures. In Emmanuel Prouff and Patrick Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 356–373. Springer, Berlin, Heidelberg, September 2012.

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 392–407. Springer, Berlin, Heidelberg, May 2000.

[CLY+24] Hongrui Cui, Hanlin Liu, Di Yan, Kang Yang, Yu Yu, and Kaiyi Zhang. ReSolveD: Shorter signatures from regular syndrome decoding and VOLE-in-the-head. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part I*, volume 14601 of *LNCS*, pages 229–258. Springer, Cham, April 2024.

[Cop94] Don Coppersmith. Solving homogeneous linear equations over gf(2) via block wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994.

[CRR21] Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. Silver: Silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 502–534, Virtual Event, August 2021. Springer, Cham.

[CWYY23] Hongrui Cui, Xiao Wang, Kang Yang, and Yu Yu. Actively secure half-gates with minimum overhead under duplex networks. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 35–67. Springer, Cham, April 2023.

[DILO22] Samuel Dittmer, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Authenticated garbling from simple correlations. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 57–87. Springer, Cham, August 2022.

[DIO21] Samuel Dittmer, Yuval Ishai, and Rafail Ostrovsky. Line-point zero knowledge and its applications. In *2nd Conference on Information-Theoretic Cryptography*, 2021.

[Dum91] Ilya Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, 1991.

[DY09] Jintai Ding and Bo-Yin Yang. Multivariate public key cryptography. *Post-quantum cryptography*, pages 193–241, 2009.

[EB22] Andre Esser and Emanuele Bellini. Syndrome decoding estimator. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 112–141. Springer, Cham, March 2022.

[EKM17] Andre Esser, Robert Kübler, and Alexander May. LPN decoded. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 486–514. Springer, Cham, August 2017.

[ES24] Andre Esser and Paolo Santini. Not just regular decoding: Asymptotics and improvements of regular syndrome decoding attacks. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VI*, volume 14925 of *LNCS*, pages 183–217. Springer, Cham, August 2024.

[Fau99] Jean-Charles Faugere. A new efficient algorithm for computing gröbner bases (F_4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.

[Fau02] Jean Charles Faugere. A new efficient algorithm for computing gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 75–83, 2002.

[FK21] Hiroki Furue and Momonari Kudo. Polynomial XL: A variant of the XL algorithm using macaulay matrices over polynomial rings. Cryptology ePrint Archive, Report 2021/1609, 2021.

[Gal14] François Le Gall. Powers of tensors and fast matrix multiplication. In Katsusuke Nabeshima, Kosaku Nagasaka, Franz Winkler, and Ágnes Szántó, editors, *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*, pages 296–303. ACM, 2014.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 52–66. Springer, Berlin, Heidelberg, December 2001.

[HjLHD22] Zhicong Huang, Wen jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and fast secure two-party deep neural network inference. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022*, pages 809–826. USENIX Association, August 2022.

[HOSS18] Carmit Hazay, Emmanuela Orsini, Peter Scholl, and Eduardo Soria-Vazquez. TinyKeys: A new approach to efficient multi-party computation. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 3–33. Springer, Cham, August 2018.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 663–680. Springer, Berlin, Heidelberg, December 2012.

[JV17] Antoine Joux and Vanessa Vitse. A crossbred algorithm for solving boolean polynomial systems. In Jerzy Kaczorowski, Josef Pieprzyk, and Jacek Pomykala, editors, *NuTMiC 2017*, volume 10737 of *LNCS*, pages 3–21. Springer, 2017.

[KL24] Jiseung Kim and Changmin Lee. Reduce and prange: Revisiting prange's information set decoding for LPN and RSD. Cryptology ePrint Archive, Report 2024/276, 2024.

[KS95] Erich Kaltofen and Victor Shoup. Subquadratic-time factoring of polynomials over finite fields. In *27th ACM STOC*, pages 398–406. ACM Press, May / June 1995.

[LWYY24] Hanlin Liu, Xiao Wang, Kang Yang, and Yu Yu. The hardness of LPN over any integer ring and field for PCG applications. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VI*, volume 14656 of *LNCS*, pages 149–179. Springer, Cham, May 2024.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, Berlin, Heidelberg, December 2011.

[MO15] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 203–228. Springer, Berlin, Heidelberg, April 2015.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8, 1962.

[RR22] Srinivasan Raghuraman and Peter Rindal. Blazing fast PSI from improved OKVS and subfield VOLE. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 2505–2517. ACM Press, November 2022.

[RRT23a] Srinivasan Raghuraman, Peter Rindal, and Titouan Tanguy. Expand-convolute codes for pseudorandom correlation generators from LPN. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 602–632. Springer, Cham, August 2023.

[RRT23b] Srinivasan Raghuraman, Peter Rindal, and Titouan Tanguy. Expand-convolute codes for pseudorandom correlation generators from LPN. In *Advances in Cryptology – CRYPTO 2023*, pages 602–632. Springer Nature Switzerland, 2023.

[RS21] Peter Rindal and Phillipp Schoppmann. VOLE-PSI: Fast OPRF and circuit-PSI from vector-OLE. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 901–930. Springer, Cham, October 2021.

[RS22] Rahul Rachuri and Peter Scholl. Le mans: Dynamic and fluid MPC for dishonest majority. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 719–749. Springer, Cham, August 2022.

[SGRR19] Phillipp Schoppmann, Adrià Gascón, Leonie Reichert, and Mariana Raykova. Distributed vector-OLE: Improved constructions and implementation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 1055–1072. ACM Press, November 2019.

[ST21] Igor Semaev and Andrea Tenti. Probabilistic analysis on macaulay matrices over finite fields and complexity of constructing gröbner bases. *Journal of Algebra*, 565:651–674, 2021.

[STA20] Chao Sun, Mehdi Tibouchi, and Masayuki Abe. Revisiting the hardness of binary error LWE. In Joseph K. Liu and Hui Cui, editors, *ACISP 20*, volume 12248 of *LNCS*, pages 425–444. Springer, Cham, November / December 2020.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications*, volume 388, 1988.

[Tho02] Emmanuel Thomé. Subquadratic computation of vector generating polynomials and improvement of the block wiedemann algorithm. *Journal of symbolic computation*, 33(5):757–775, 2002.

[vzGS92] Joachim von zur Gathen and Victor Shoup. Computing frobenius maps and factoring polynomials (extended abstract). In *24th ACM STOC*, pages 97–105. ACM Press, May 1992.

[Wie86] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theory*, 32(1):54–62, 1986.

[WMK16] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. EMP-toolkit: Efficient Multi-Party computation toolkit. https://github.com/emp-toolkit, 2016.

[WWY+25] Tianrui Wang, Anyu Wang, Kang Yang, Hanlin Liu, Yu Yu, Jun Zhang, and Xiaoyun Wang. A hybrid algorithm for the regular syndrome decoding problem. Cryptology ePrint Archive, Paper 2025/1284, 2025.

[WYKW21] Chenkai Weng, Kang Yang, Jonathan Katz, and Xiao Wang. Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In *2021 IEEE Symposium on Security and Privacy*, pages 1074–1091. IEEE Computer Society Press, May 2021.

[WYX+21] Chenkai Weng, Kang Yang, Xiang Xie, Jonathan Katz, and Xiao Wang. Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 501–518. USENIX Association, August 2021.

[YCBC07] Bo-Yin Yang, Owen Chia-Hsin Chen, Daniel J. Bernstein, and Jiun-Ming Chen. Analysis of QUAD. In Alex Biryukov, editor, *FSE 2007*, volume 4593 of *LNCS*, pages 290–308. Springer, Berlin, Heidelberg, March 2007.

[YCC04] Bo-Yin Yang, Jiun-Ming Chen, and Nicolas Courtois. On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis. In Javier López, Sihan Qing, and Eiji Okamoto, editors, *ICICS 04*, volume 3269 of *LNCS*, pages 401–413. Springer, Berlin, Heidelberg, October 2004.

[YH23] Yibin Yang and David Heath. Open source library for improved constant overhead zero-knowledge RAM, 2023. https://github.com/gconeice/improved-zk-ram.

[YSWW21] Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. QuickSilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2986–3001. ACM Press, November 2021.

[Yun76] David Y. Y. Yun. On square-free decomposition algorithms. In Richard D. Jenks, editor, *Proceedings of the third ACM Symposium on Symbolic and Algebraic Manipulation, SYMSAC*, pages 26–35. ACM, 1976.

[YWL+20] Kang Yang, Chenkai Weng, Xiao Lan, Jiang Zhang, and Xiao Wang. Ferret: Fast extension for correlated OT with small communication. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1607–1626. ACM Press, November 2020.