Polly Cracker, Revisited

Martin Albrecht, Jean-Charles Faugère, Pooya Farshim, Gottfried Herold, Ludovic Perret

Polly Cracker, Revisited

Martin R. Albrecht[1], Jean-Charles Faugère[2], Pooya Farshim[3], Gottfried Herold[4], and Ludovic Perret[2]

[1] Technical University of Denmark, Denmark
[2] INRIA, Paris-Rocquencourt Centre, POLSYS Project
UPMC Univ. Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France
[3] Queen's University Belfast, Northern Ireland, UK
[4] Ruhr-Universität Bochum, Horst Görtz Institut für IT-Sicherheit, Germany

Abstract. We formally treat cryptographic constructions based on the hardness of deciding ideal membership in multivariate polynomial rings. Of particular interest to us is a class of schemes known as "Polly Cracker." We start by formalising and studying the relation between the ideal membership problem and the problem of computing a Gröbner basis. We show both positive and negative results. On the negative side, we define a symmetric Polly Cracker encryption scheme and prove that this scheme only achieves bounded CPA security under the hardness of the ideal membership problem. Furthermore, we show that a large class of algebraic transformations cannot convert this scheme to a fully secure Polly Cracker-style scheme. On the positive side, we formalise noisy variants of the ideal-theoretic problems. These problems can be seen as natural generalisations of the learning with errors (LWE) and the approximate GCD problems over polynomial rings. After formalising and justifying the hardness of the noisy assumptions, we show that noisy encoding of messages results in a fully IND-CPA-secure and somewhat homomorphic encryption scheme. Together with a standard symmetric-to-asymmetric transformation for additively homomorphic schemes, we provide a positive answer to the long-standing open problem of constructing a secure Polly Cracker-style cryptosystem reducible to the hardness of solving a random system of equations. Indeed, our results go beyond this and also provide a new family of somewhat homomorphic encryption schemes based on generalised hard problems. Our results also imply that Regev's LWE-based public-key encryption scheme is (somewhat) *multiplicatively* homomorphic for appropriate choices of parameters.

Key words. [94A60] Cryptography, [93C35] Multivariable systems, [68Q17] Computational difficulty of problems.

1 Introduction

Background. Fully homomorphic encryption [41] is a cryptographic primitive which allows performing arbitrary computations over encrypted data. In such a scheme, given a function f and a ciphertext c encrypting a plaintext m, it is possible to transform c into a new ciphertext c' which encrypts $f(m)$. From an algebraic perspective, this homomorphic feature can be seen as the ability to evaluate multivariate (Boolean) polynomials over ciphertexts. Hence, instantiating homomorphic encryption over the ring of multivariate polynomials is perhaps most natural, although not necessarily conceptually the simplest (cf. [68]).

Indeed, let $\mathcal{I}$ be some ideal in $P := \mathbb{F}[x_0, \ldots, x_{n-1}]$. Denote an injective function mapping bit strings to elements in the quotient ring $P/\mathcal{I}$ by $\mathbf{Encode}(\cdot)$, and its inverse by $\mathbf{Decode}(\cdot)$. If $\mathbf{Decode}(\mathbf{Encode}(m_0) \circ \mathbf{Encode}(m_1)) = m_0 \circ m_1$ for $\circ \in \{+, \cdot\}$, we can encrypt a message m as

$$c = f + \mathbf{Encode}(m), \text{ for } f \text{ randomly chosen in } \mathcal{I}. \tag{1}$$

The homomorphic features of this scheme follow from the definition of an ideal. Decryption is performed by computing remainders modulo $\mathcal{I}$. Alternative, if $\mathbf{Encode}(m) = 0$ for some message m, decryption is performed by deciding ideal membership. The problems of computing remainders modulo an ideal and deciding ideal membership were solved by Buchberger [20,21,22], where he introduced the notion of Gröbner bases, and gave an algorithm for computing such bases.

In fact, most known homomorphic schemes which support both addition and multiplication are based on variants of the ideal membership problem over various rings. For example in [68] the ring $\langle p \rangle \subseteq \mathbb{Z}$, for p an odd integer, is considered. In [41] ideals in a number field play the same role (cf. [65]). One can also view Regev's LWE-based public-key encryption scheme [59] as well as the homomorphic encryption scheme based on it [19] in this framework. Furthermore, if we instantiate the construction in [53] over P, we can view its multiplication operation as constructing the set of cross-terms appearing in multivariate polynomial multiplication. Finally, we note that the construction displayed above is essentially Polly Cracker [39,11,49], a family of cryptosystems dating back to the early 1990s. Despite their simplicity, our confidence in Polly Cracker-style schemes has been shaken as almost all such proposals have been broken [33]. This is partially due to the lack of formal treatment of security for such schemes in the literature. In fact, it is a long-standing open research problem to propose a secure Polly Cracker-style encryption scheme [11] (cf. [40, p. 41]).

Related works

Polly Cracker. In 1993, Barkee et al. wrote a paper [11] whose aim was to dispel the urban legend that "Gröbner bases are hard to compute." Another goal of this paper was to direct research towards *sparse* systems of multivariate equations. To do so, the authors proposed the most obvious dense Gröbner-based cryptosystem, namely an instantiation of the construction mentioned at the beginning of the introduction. In their scheme, the public key consists of a set of polynomials $\{f_0, \ldots, f_{m-1}\} \subset \mathcal{I}$ which are used to construct an element $f \in \mathcal{I}$. Encryption of messages $m \in P/\mathcal{I}$ are computed as $c = \sum h_i f_i + m = f + m$ for $f \in \mathcal{I}$. The private key is a Gröbner basis G which allows computing $m = c \bmod \mathcal{I} = c \bmod G$. As highlighted in [11] this scheme can be broken using results from [32] (cf. Section 5, Theorem 9).

At about the same time, and independently of Barkee et al., Fellows and Koblitz [39,49] proposed a framework for the design of public-key cryptosystems. The ideas in [39] were similar to Barkee et al.'s, but differed in two aspects. First, the polynomials generating the public ideal were derived from combinatorial or algebraic NP-complete problems (such systems were named CA-systems for "combinatorial-algebraic"). Second, the secret key was not a Gröbner basis of the public ideal, but rather a root of it, i.e. a Gröbner basis of a maximal ideal containing the public ideal. The main instantiation of such a system was the Polly Cracker cryptosystem. Fellows and Koblitz suggested several NP-complete problems, mainly based on graph-theoretic problems, for use in this context. The authors, however, did not investigate how one might generate "hard-on-average" instances of these problems with known solutions.

Subsequently, a variety of sparse Polly Cracker-style schemes were proposed. The focus on sparse polynomials aimed to prevent the attack based on Theorem 9 (Section 5), yet almost all of these schemes were broken. We point the reader to [33] for a good survey of various constructions and attacks. Currently, the only Polly Cracker-style scheme which is not broken is the scheme in [24]. This scheme is based on binomial ideals, which in turn are closely related to lattices.

Not only can our constructions be seen as instantiations of Polly Cracker (with and without noisy encoding of messages), they also allow security proofs based on the hardness of computational problems related to random systems. Our work presents a general treatment of problems related to ideals over multivariate polynomials – both with and without noise – and aims to provide a formal basis to assess the security of cryptosystems based on such problems.

Homomorphic encryption. In the last decades several different approaches to construct singly homomorphic schemes—with respect to both hardness assumptions and proofs of security—have been investigated. With respect to doubly (i.e. additively and multiplicatively) homomorphic schemes, a number of different hardness assumptions and constructions appeared in the literature. These include the ideal coset problem of Gentry [41], the approximate GCD problem over the integers (AGCD) of van Dijk et al. [68], the polynomial coset problem as proposed by Smart and Vercauteren in [65], the approximate unique shortest vector problem, the subgroup decision problem, and the differential knapsack vector problem all of which appear in the work of Aguilar Melchor et al. [53] as well as the learning with errors problem (LWE) of Brakerski and Vaikuntanathan [19]. There is a general agreement in the community that whilst the design of fully homomorphic encryption schemes is a great theoretical breakthrough, all schemes so far have remained rather impractical. However, research in this direction is progressing rapidly. Recently, Gentry and Halevi [43] have been able to implement all aspects of Gentry's scheme [41], including the bootstrapping step. In this work the authors also improve on the work of Smart and Vercauteren [65]. Later, Gentry, Halevi, and Smart implemented AES homomorphically [45]. However, the bootstrapping step still renders somewhat homomorphic schemes impractical (cf. [56]). Hence, some recent constructions aim to avoid it [18,42] and work is ongoing to improve this step [44].

Recently and independently of this work, Brakerski and Vaikuntanathan [19] gave an encryption scheme, SH, based on the LWE problem that can be seen as a linear variant of our noisy Polly Cracker scheme. Furthermore, the technique we propose in Section 8 was also independently proposed in this work. However, in contrast to our work, the authors of [19] have an explicit non-algebraic perspective. Also, a second scheme, BTS, was also proposed in [19], and it achieves *full* homomorphicity based on a "dimension-modulus reduction" technique, while our work only yields a somewhat homomorphic encryption scheme. We note that this technique also applies to our constructions. Finally, we note that improvements such as those proposed in [28] immediately apply to our constructions (which generalise the constructions considered there).

The main difference between our work—which can be seen as an instantiation of Gentry's ideal coset problem—and previous work is that we base the security of our somewhat homomorphic scheme on *new* computational problems related to ideals over multivariate polynomial rings which *generalise* previously considered problems [19,68]. Furthermore, our construction in Section 7 can be seen as a generalisation of a number of known schemes and their underlying hardness assumptions. As such, our work does not improve on such constructions in terms of efficiency, but provides a unified perspective on previous schemes and problems.

History of this Work. This work is based on two conference contributions. The first of which being "Polly Cracker, revisited" which appeared at ASIACRYPT 2011 [2]. While this work contains the bulk of theorems and lemmas also present in [2], there are some significant difference. Firstly, [2] did not contain most proofs due to space restrictions. Furthermore, it defined two problem IR and IRN which were dropped from this work, as they were found of little utility. However, most

importantly, [2] contains several errors which were pointed out and partly corrected in [48] which appeared at PKC 2012. In light of this, the authors of both publications [48] and [2] jointly revisited all theorems, proofs and constructions which resulted in this paper.

Contributions & organisation

Our contributions in this paper can be summarised as follows:

1. We conduct a formal treatment of Polly Cracker-style schemes over multivariate polynomial rings and characterise their security.
2. We demonstrate the impossibility of converting such schemes to fully IND-CPA-secure schemes through a large class of transformations.
3. We introduce natural noisy variants of classical problems related to Gröbner bases which also generalise previously considered noisy problems such as the LWE and the approximate GCD (AGCD) problems.
4. We present a new somewhat (and doubly) homomorphic encryption scheme based on these new hard problems.

In order of achieve these results, our work also provide some new results about the intractability of multivariate problems with noise.

After this introduction, we start by giving an overview of Gröbner bases in Section 2. The reader already familiar with commutative algebra can skip this part. In Section 3, we formalise various problems associated with ideals in polynomials rings in the language of code-based security definitions [15]. Namely, we define the Gröbner basis (GB) and the ideal membership (IM) problems in the code-based game-playing language [15]. It not difficult to show, in the game-based formalism, that IM is not harder than GB. We show also that deciding ideal membership with overwhelming probability is equivalent to compute Gröbner bases for zero-dimensional ideals for certain choices of parameters.

Lemma 1 (informal). *If* IM *is overwhelmingly easy, then* GB *is overwhelmingly easy. Conversely, if* GB *is easy then* IM *is easy as well.*

This allows us to introduce a symmetric variant of Polly Cracker, that we shall call $\mathcal{SPC}$, and precisely characterise its security guarantees. In particular, we show that this scheme achieves a weaker version of IND-CPA security where the total number of ciphertexts that the attacker can obtain is a priori bounded by a fixed polynomial. We prove this result under the assumption that computing Gröbner bases is hard if only a small number of polynomials are available to the attacker (Section 4). Bounded IND-CPA security is, in some sense, the best level of security that this scheme can possibly achieve: we give an attacker breaking the cryptosystem once enough ciphertexts are collected.

Theorem 1 (simplified). *The bounded* IND-BCPA *security of* $\mathcal{SPC}$ *is essentially equivalent to the hardness of* IM*.*

In Section 5, using results from computational commutative algebra, we show the security limitations of the constructed scheme are, in some sense, *intrinsic*. More precisely, we show that a

large class of algebraic transformation cannot turn this scheme into a fully IND-CPA-secure and additively homomorphic (public-key) Polly Cracker-type scheme. Our result captures both known symmetric-to-asymmetric conversion techniques for homomorphic schemes in the literature [61,68]. Furthermore, this result—due to the generality of Gröbner bases—implies that IND-CPA-secure homomorphic encryption is difficult to construct without noisy encoding of messages (further evidence for this is given in [17]). The main technical result of Section 5 is:

Theorem 2 (informal). *Sampling uniformly elements of a polynomial ideal $\mathcal{I}$ up to some degree is equivalent to computing a Gröbner basis for $\mathcal{I}$.*

In order to go beyond this barrier we consider constructions where the **Encode**() function introduced in (1) is *randomised*. To prove security for such schemes, we consider noisy variants of the ideal membership (IMN) and related problems (namely, WIMN a weaker version of ideal ideal membership with noise, and GBN a noisy variant of GB). These can be seen as natural generalisations of the (decisional) LWE and AGCD problems over polynomial rings (Section 6). Regarding the hardness of these new problems, we show that:

Lemma 2 (informal). *If* GBN *easy, then it holds that* WIMN *is easy.*

Note that WIMN is not harder that IMN, whilst the converse is not always true. However, we show that the equivalence holds in a relevant case. We also prove an average-case-to-worst-case reduction for IMN.

Lemma 3. *(informal) Assume that the ideal considered in the* IMN *games have a unique solution. Then, if we can solve the* IMN *problem for a polynomial fraction of instances, then we can also solve it for all instances.*

After formalising and justifying the hardness of the noisy assumptions in Section 7, we show that noisy encoding of messages can indeed be used to construct a fully IND-CPA-secure somewhat homomorphic scheme.

Theorem 3 (informal). *The* IND-CPA *security of $\mathcal{SPCN}$ scheme is essentially equivalent to the hardness of* WIMN.

Our result, together with a standard symmetric-to-asymmetric conversion for homomorphic schemes, provides a positive answer to the long-standing open problem proposed by Barkee et al. [11] asking for a Polly Cracker-style public-key encryption scheme whose security is based on the hardness of computing Gröbner bases for random systems of polynomials. This result also implies that Regev's LWE-based public-key scheme is *multiplicatively* homomorphic under appropriate choices of parameters.

In Section 8 we show that our scheme allows proxy re-encryption of ciphertexts. This re-encryption procedure can be seen as trading noise for degree in ciphertexts. In this section, we also show that our scheme achieves a limited form of key-dependent message (KDM) security in the standard model, where the least significant bit of the constant term of the key is encrypted. We leave it as an open problem to adapt the techniques of [4] to achieve full KDM security for the Polly Cracker with noise scheme. We conclude by discussing concrete parameter choices in Section 9, and give a reference implementation in Section 10.

2 Basics of Gröbner Bases

In this section we recall some basic definitions and results related to Gröbner bases [22,20,21]. For a more detailed treatment we refer the reader to [30]. We consider a polynomial ring $P = \mathbb{F}[x_0, \ldots, x_{n-1}]$ over some finite field (typically prime), some degree-compatible monomial ordering on the elements of P with $x_i > x_j$ if $i < j$, and a set of polynomials $f_0, \ldots, f_{m-1}$. We denote by $\mathrm{M}(f)$ the set of all monomials appearing in $f \in P$ and extend this definition to sets of polynomials in the natural way. By $\mathrm{LM}(f)$ we denote the leading monomial appearing in $f \in P$ according to the chosen term ordering. We denote by $\mathrm{LC}(f)$ the coefficient of $\mathrm{LM}(f)$ in f, and set $\mathrm{LT}(f) := \mathrm{LC}(f) \cdot \mathrm{LM}(f)$. We denote by $P_{<d}$ the set of polynomials of degree $< d$ (and analogously for $>, \leq, \geq$, and $=$ operations). We define $P_{=0}$ as the underlying field, including $0 \in \mathbb{F}$. We define $P_{<0}$ as zero. Finally, we denote by $M_{<m}$ the set of all monomials $< m$ for some monomial m (and analogously for $>, \leq, \geq$, and $=$ operations). We assume the usual power-product representation for elements of P.

Definition 1 (Generated ideal). *Let $f_0, \ldots, f_{m-1} \in P$ be polynomials. The set*

$$\mathcal{I} = \langle f_0, \ldots, f_{m-1} \rangle := \left\{ \sum_{i=0}^{m-1} h_i f_i \mid h_0, \ldots, h_{m-1} \in P \right\}$$

is called the ideal generated by $f_0, \ldots, f_{m-1}$.

It is known that every ideal $\mathcal{I}$ of P is finitely generated, i.e. there exists a finite number of polynomials $f_0, \ldots, f_{m-1}$ in P such that $\mathcal{I} = \langle f_0, \ldots, f_{m-1} \rangle$. A Gröbner basis of an ideal is a set of generators of the ideal which takes a particular form.

Definition 2 (Gröbner basis). *Let $\mathcal{I}$ be an ideal of $\mathbb{F}[x_0, \ldots, x_{n-1}]$ and fix a monomial ordering. A finite subset $G = \{g_0, \ldots, g_{m-1}\} \subset \mathcal{I}$ is said to be a Gröbner basis of $\mathcal{I}$ if for any $f \in \mathcal{I}$ there exists a $g_i \in G$ such that $\mathrm{LM}(g_i) \mid \mathrm{LM}(f)$.*

Remark. We note that for the vector space $\mathbb{F}^n$, the notion of a Gröbner basis coincides with that of a row echelon form, and Gröbner basis algorithms (see below) reduce to Gaussian elimination. For univariate polynomial rings, e.g. $\mathbb{F}[x]$ and $\mathbb{Z}[x]$, the notion of a Gröbner basis coincides with greatest common divisor, and running a Gröbner basis algorithm computes the GCD.

It is possible to extend the polynomial division algorithm to multivariate polynomials: we write $r = f \bmod G$ when r is a possible result of applying the multivariate division algorithm on f and G for the given monomial ordering. It holds that $f = \sum_{i=0}^{m-1} h_i \, g_i + r$ with $\mathrm{M}(r) \cap \langle \mathrm{LM}(G) \rangle = \emptyset$. When G is a Gröbner basis, r is unique and is called the *normal form* of f with respect to the ideal $\mathcal{I}$. In particular, we have that $f \bmod \mathcal{I} = f \bmod G = 0$ if and only if $f \in \mathcal{I}$. Given P and $\mathcal{I}$, we can define the quotient ring $P/\mathcal{I}$. By abuse of notation, we write $f \in P/\mathcal{I}$ if $f \bmod \mathcal{I} = f$ where the last equality is interpreted over the elements of P. That is, we identify elements of the quotient $P/\mathcal{I}$ with their minimal representation in P.

As defined above, a Gröbner basis is not unique. For instance, we can multiply any polynomial of a Gröbner basis by a nonzero constant. However, given any Gröbner basis we can compute the unique *reduced* Gröbner basis in polynomial time via $\mathsf{ReduceGB}(\cdot)$ given in Algorithm 1.

Definition 3 (Reduced Gröbner basis). *A reduced Gröbner basis for an ideal $\mathcal{I} \subset P$ is a Gröbner basis G such that: (1)* $\mathrm{LC}(g) = 1$*, for all $g \in G$, and (2) $\forall g \in G, \nexists\ m \in \mathrm{M}(g)$ such that m is divisible by some element of* $\mathrm{LM}(G \setminus \{g\})$.

Algorithm 1: ReduceGB(G)

```
1 begin
2     G̃ ← ∅;
3     while G ≠ ∅ do
4         f ← the smallest element of G according to the term ordering;
5         G ← G \ {f};
6         if LM(f) ∉ ⟨LM(G̃)⟩ then
7             G̃ ← G̃ ∪ {LC(f)^{-1} · f};
8     return [h mod G̃\{h} | h ∈ G̃];
```

Buchberger [20] proved that in order to compute a Gröbner basis from a given ideal basis, it is sufficient to consider so-called S-polynomials. From such a basis, it is easy to compute the (unique) reduced Gröbner basis using Algorithm 1.

Definition 4 (S-polynomial). *Let $f, g\ \in \mathbb{F}[x_0, \ldots, x_{n-1}]$ be nonzero polynomials.*

- *Let* $\mathrm{LM}(f) = \prod_{i=0}^{n-1} x_i^{\alpha_i}$ *and* $\mathrm{LM}(g) = \prod_{i=0}^{n-1} x_i^{\beta_i}$*, with $\alpha_i, \beta_i \in \mathbb{N}$, denote the leading monomials of f and g respectively. For every $0 \leq i < n$ set $\gamma_i := \max(\alpha_i, \beta_i)$ and denote by x^γ the polynomial $\prod_{i=0}^{n-1} x_i^{\gamma_i}$. Then x^γ is the least common multiple of* $\mathrm{LM}(f)$ *and* $\mathrm{LM}(g)$*:*

$$x^\gamma\ = \mathrm{LCM}(\mathrm{LM}(f), \mathrm{LM}(g)).$$

- *The S-polynomial of f and g is defined as*

$$S(f,g)\ =\ \frac{x^\gamma}{\mathrm{LT}(f)} \cdot f\ -\ \frac{x^\gamma}{\mathrm{LT}(g)} \cdot g.$$

Buchberger showed that a basis is a Gröbner basis if all S-polynomials "reduce to zero."

Definition 5 (Reduction to zero). *Fix a monomial order in P and let $G = \{g_0, \ldots, g_{s-1}\} \subset P$ be an unordered set of polynomials and let t be a monomial. Given a polynomial $f \in P$, we say f has a t-representation with respect to $\leq$ and G if f can be written as*

$$f = a_0 g_0 + \cdots + a_{s-1} g_{s-1},$$

such that whenever $a_i g_i \neq 0$, we have $a_i g_i \leq t$. Furthermore, we write that $f \underset{G}{\longrightarrow} 0$ ("f reduces to zero") if and only if f has an $\mathrm{LM}(f)$*-representation with respect to G.*

Note that $f \bmod G = 0$ implies that $f \underset{G}{\longrightarrow} 0$ while the converse is not necessarily the case.

Theorem 4 (Buchberger's criterion). *A basis $G = \{g_0, \ldots, g_{s-1}\}$ for an ideal $\mathcal{I}$ is a Gröbner basis if and only if for all $i \neq j$ we have $S(g_i, g_j) \xrightarrow[G]{} 0$.*

Theorem 4 (see [13, p.211f] for a proof) leads to an algorithm [20] which computes a Gröbner basis by constructing and reducing S-polynomials. However, this algorithm – Buchberger's algorithm – spends most of its time reducing elements to zero, a computation which is of no use. Buchberger also proposed two criteria which tell us *a priori* whether the S-polynomial of two polynomials reduces to zero. We make use of the first criterion in this work (see [13, p.222f] for a proof of this result):

Theorem 5 (Buchberger's first criterion). *Let $f, g \in P$ be such that*

$$\mathrm{LCM}(\mathrm{LM}(f), \mathrm{LM}(g)) = \mathrm{LM}(f) \cdot \mathrm{LM}(g),$$

i.e. f and g have disjoint leading terms. Then $S(f, g) \xrightarrow[\{f,g\}]{} 0$.

From this, we obtain the following corollary.

Corollary 1. *A set $\{g_0, \ldots, g_{n-1}\} \subset P$ with $\mathrm{LM}(g_i) = x_i^{d_i}$ with $d_i \geq 0$ for all $i, 0 \leq i < n$ is a Gröbner basis.*

All ideals considered in this work are zero-dimensional, i.e. their associated varieties have finitely many points. The following lemma establishes the equivalence between various statements about zero-dimensional ideals. This result will be required to analyse some algorithms introduced in Section 3 (a proof of this result can be found, for instance, in [30, p.234f]).

Lemma 4 (Finiteness criterion). *Let $\mathcal{I} = \langle f_0, \ldots, f_{m-1} \rangle \subset P := \mathbb{F}[x_0, \ldots, x_{n-1}]$ be an ideal. The following conditions are equivalent.*

1. *The system has only finitely many solutions in the algebraic closure of $\mathbb{F}$.*
2. *For any $i \in \{0, \ldots, n-1\}$, we have $\mathcal{I} \cap \mathbb{F}[x_i] \neq \varnothing$.*
3. *For any $i \in \{0, \ldots, n-1\}$, there exists $g_i \in \mathcal{I}$ such that $\mathrm{LM}(g_i) = x_i^{d_i}$ with $d_i > 0$.*
4. *The set of monomials $S(\mathcal{I}) := \mathrm{M}(P) \setminus \{\mathrm{LM}(f) \mid f \in \mathcal{I}\}$ is finite.*
5. *The $\mathbb{F}$-vector space $P/\mathcal{I}$ is finite-dimensional and a basis is given by $S(\mathcal{I})$.*

As soon as one of these conditions holds true, then we call the ideal $\mathcal{I}$ zero-dimensional*. Moreover, the number of solutions counted with multiplicities in the algebraic closure of $\mathbb{F}$ is exactly the cardinal of $S(\mathcal{I})$ which is the dimension of the vector space $P/\mathcal{I}$.*

We will be using reduction modulo an ideal to sample polynomials from some ideal. The following lemma will be helpful to assert that this sampling is uniform.

Lemma 5. *Let $\mathcal{I} \subset P$ be an ideal with a degree-compatible term ordering $\leq$. Then any $f \in P$ with $\deg(f) = b$ has a unique representation $f = \tilde{f} + r$ with $\tilde{f} \in \mathcal{I}$ and $r \in P/\mathcal{I}$ where $\deg(\tilde{f}) \leq b$ and $\deg(r) \leq b$. In particular, if $\mathrm{M}(P_{\leq b}/\mathcal{I})$ is the set of monomials in $P/\mathcal{I}$ with degree at most b, then for any $\tilde{f} \in \mathcal{I}_{\leq b}$ there are q^s elements f_i in $P_{\leq b}$ with $\tilde{f} = f_i - (f_i \bmod \mathcal{I})$ and $s = |\mathrm{M}(P_{\leq b}/\mathcal{I})|$.*

Proof. Given f we recover the unique r by computing $f \bmod G$ by a standard fact about Gröbner bases and get $\tilde{f} = f - r$. Since P has a degree-compatible ordering, r has degree at most b. To prove the second claim, note that the monomials in $P_{\leq b}$ span an $\binom{n+b}{b}$-dimensional vector space V over $\mathbb{F}_q$. The monomials in $P/\mathcal{I}$ up to degree b span a subspace of V of dimension $|\mathrm{M}(P_{\leq b}/\mathcal{I})|$, from which the claim follows. □

3 The Gröbner Basis and Ideal Membership Problems

In this section we formalise various problems associated with Gröbner bases.

NOTATION. We write $\mathsf{x} \leftarrow \mathsf{y}$ for assigning value y to a variable x, and $\mathsf{x} \leftarrow_{\$} \mathsf{X}$ for sampling x from a set X uniformly at random. If A is a probabilistic algorithm we write $\mathsf{y} \leftarrow_{\$} \mathsf{A}(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ for the action of running A on inputs $\mathsf{x}_1, \ldots, \mathsf{x}_n$ with uniformly chosen random coins, and assigning the result to y. For a random variable X we denote by $[\mathsf{X}]$ the support of X, i.e. the set of all values that X takes with nonzero probability. We use PPT for probabilistic polynomial-time. We call a function $\epsilon(\lambda)$ negligible if $|\epsilon(\lambda)| \in \lambda^{-\omega(1)}$. We say a function $f(\lambda)$ is overwhelming if $1 - f(\lambda)$ is negligible. We say that a function space $\mathsf{FunSp}(\mathsf{P})$ and a message space $\mathsf{MsgSp}(\mathsf{P})$, both parameterised by P, are compatible if for any possible value of P and for any $f \in \mathsf{FunSp}(\mathsf{P})$, the domain of f is $\mathsf{MsgSp}(\mathsf{P})$. We also denote by ω the matrix multiplication exponent (a.k.a. the linear-algebra constant) as defined in [69, Chapter 12]. We recall [70,67] that $\omega \in [2, 2.3727]$.

To formalise our problems, we use the code-based game-playing language [15]. Each game has an **Initialize** and a **Finalize** procedure. It also has specifications of procedures to respond adversary's various oracle queries. A game Game is run with an adversary $\mathcal{A}$ as follows. First **Initialize** runs and its outputs are passed to $\mathcal{A}$. Then $\mathcal{A}$ runs and its oracle queries are answered by the procedures of Game. When $\mathcal{A}$ terminates, its output is passed to **Finalize** which returns the outcome of the game y. This interaction is written as $\mathsf{Game}^{\mathcal{A}} \implies y$. In each game, we restrict our attention to legitimate adversaries, which are defined specifically for each game.

Following [31], we define *a computational polynomial ring scheme*. This is a general framework allowing to discuss in a concrete way the different families of rings that may be used in cryptographic applications. More formally, a computational polynomial ring scheme $\mathcal{P}$ is a sequence of probability distribution of *polynomial ring descriptions* $(\mathbf{P}_\lambda)_{\lambda \in \mathbb{N}}$. A polynomial ring description[1] P specifies various algorithms associated with P such as computing the ring operations, sampling of elements, testing membership, encoding of elements, ordering of monomials, etc. We assume each polynomial ring distribution is over $n = n(\lambda)$ variables, for some polynomial $n(\lambda)$, and is over a finite field of prime size $q(\lambda)$.

For q a prime, there is a one-to-one correspondence between ideals $\mathcal{I} \subset \mathbb{F}_{q^n}[x_0, \ldots, x_{n-1}]$ on polynomial rings over finite extension fields and over prime fields $\mathcal{J} \subset \mathbb{F}_q[x_0, \ldots, x_{n-1}, \alpha]$: map a root of $\mathbb{F}_{q^n}$ to α and add the characteristic polynomial of $\mathbb{F}_{q^n}$ to the generating basis. Hence, finite extension fields are covered by this definition. The ring $\mathbb{Z}[x_0, \ldots, x_{n-1}]$ is not covered by our definition for brevity, but it can easily be generalised [13, Ch. 10].

Once $\mathcal{P}$ is given and a concrete ring P is sampled, one can define various Gröbner basis generation algorithms on P. In this work we denote by $\mathsf{GBGen}(1^\lambda, P, d, \ell)$ any PPT algorithm which outputs

[1] Here we are slightly abusing notation and using P both for the polynomial ring and its description.

a reduced Gröbner basis G for some zero-dimensional ideal $\mathcal{I} \subset P$ such that the last ℓ elements of G have degree d and the remaining elements have degree 1 and such that $(P \setminus \mathcal{I})_{\leq b}$ is not empty. Of particular interest to us is the Gröbner basis generation algorithm shown in Algorithm 2 called $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$. Throughout this paper we assume an implicit dependency of various parameters associated with P on the security parameter. Thus, we drop λ to ease notation. Finally, we always assume that $\mathrm{LM}(G)$ and hence $S(\mathcal{I})$ is fixed by $\mathsf{GBGen}(\cdot)$ for each λ, and thus is known. We

Algorithm 2: Algorithm $\mathsf{GBGen}_{\mathsf{dense}}(1^\lambda, P, d, \ell)$

```
1  begin
2      if d = 0 then return {0} for i ∈ {0, ..., n − 1} do
3          if i > n − ℓ − 1 then
4              g_i ← x_i^d;
5          else
6              g_i ← x_i;
7          for m_j ∈ M_{<LM(g_i)} do
8              c_ij ←$ F_q;
9              g_i ← g_i + c_ij m_j;
10     return ReduceGB({g_0, ..., g_{n−1}});
```

note that using Buchberger's First Criterion in Algorithm 2 is a special case of using Macaulay's trick [55]. Also, $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$ for $d = 1$ or any $d > 1$ with $\ell = 0$ captures the usual case of a set of polynomials which have a (unique) common root in the base field, and where $\mathrm{LM}(g_i) = x_i$ for all $i, 0 \leq i < n$. This case is common in cryptographic applications such as algebraic cryptanalysis, e.g. [37,29,64] and is well studied. The next lemma – which is an easy consequence of Corollary 1 – establishes that $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$ returns a Gröbner basis with $\dim(P/\mathcal{I}) = d^\ell$.

Lemma 6. *Let $G = \{g_0, \ldots, g_{n-1}\} \subset P = \mathbb{F}[x_0, \ldots, x_{n-1}]$ be the set of polynomials defined as*

$$g_i := x_i^{d_i} + \sum_{m_j \in M_{<x_i^{d_i}}} c_{ij} m_j, \forall i, 0 \leq i < n$$

where $c_{ij} \in \mathbb{F}$. Then G is a Gröbner basis for the zero-dimensional ideal $\langle g_0, \ldots, g_{n-1} \rangle$. Additionally, the dimension of the $\mathbb{F}_q$-vector space $P/\langle g_0, \ldots, g_{n-1} \rangle$ is $\prod_{i=0}^{n-1} d_i$.

Proof. The Gröbner basis property follows from Corollary 1. Clearly, $S(\mathcal{I}) = \mathrm{M}(P) \setminus \{\mathrm{LM}(f) \mid f \in \mathcal{I}\}$ is the set of all monomials of the form $\prod_{j=0}^{n-1} x_j^{\gamma_j}$ where all $\gamma_j < d_j$. Since there are $\prod_{i=0}^{n-1} d_i$ such elements, this is also the dimension of the vector space by Lemma 4. □

Denote $Q = P_{\leq b}/\mathcal{I}$ for b some fixed parameter and note here that $P_{\leq b}/\mathcal{I} = (P/\mathcal{I})_{\leq b}$, since the monomial order sorts by total degree first. In this work we are mainly interested in the case where Q has polynomially many elements. In this case we require ℓ to be a constant but allow q to depend on λ. We note, however, that larger quotients are permitted by our definitions.

We now formally define the Gröbner basis problem, which is the problem of computing the Gröbner basis for some ideal $\mathcal{I}$ given a set of polynomials $f_0, \ldots, f_{m-1} \in \mathcal{I}$.

Definition 6 (The Gröbner basis (GB) problem). *The Gröbner basis problem is defined through game* $\mathsf{GB}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m}$ *shown in Figure 1. The advantage of a* PPT *algorithm* $\mathcal{A}$ *in solving the* GB *problem is defined by*

$$\mathbf{Adv}^{\mathsf{gb}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}}(\lambda) := \Pr\left[\mathsf{GB}^{\mathcal{A}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m}(\lambda) \Rightarrow \mathsf{True}\right].$$

An adversary is legitimate if it calls the **Sample** *procedure of Figure 1 at most* $m = m(\lambda)$ *times.*

Initialize$(1^\lambda, \mathcal{P}, d, \ell)$:	**Sample**():	**Finalize**(G'):
begin	**begin**	**begin**
$P \leftarrow_\$ \mathbf{P}_\lambda$;	$f \leftarrow_\$ P_{\leq b}$;	**return** $(G = G')$;
$G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d, \ell)$;	$f \leftarrow f - (f \bmod G)$;	**end**
return $(1^\lambda, P)$;	**return** f;	
end	**end**	

Fig. 1. Game $\mathsf{GB}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m}$.

It follows from Lemma 5 that the **Sample** procedure in Figure 1 returns elements of degree $\leq b$ which are uniformly distributed in $\langle G\rangle_{\leq b}$. We note that usually we must require $b \geq d$ in order to exclude the trivial case where **Sample** always returns zero or elements independent of some elements of the Gröbner basis.

We recall that given a Gröbner basis G of an ideal $\mathcal{I}$, $r = f \bmod \mathcal{I} = f \bmod G$ is the normal form of f with respect to the ideal $\mathcal{I}$. We sometimes drop the explicit reference to $\mathcal{I}$ when it is clear from the context which ideal we are referring to, and simply refer to r as the normal form of f. Furthermore $f \in \mathcal{I}$ if and only if $r = 0$. This is the well-known ideal membership problem formalised below. We mention that solving this problem was the original motivation which led to the discovery of Gröbner bases [20].

Definition 7 (The ideal membership (IM) problem). *The ideal membership problem is defined through game* $\mathsf{IM}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m}$ *shown in Figure 2. The advantage of a* PPT *algorithm* $\mathcal{A}$ *in solving* IM *is defined by*

$$\mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{IM}^{\mathcal{A}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m}(\lambda) \Rightarrow \mathsf{True}\right] - 1.$$

An adversary is legitimate if it calls the **Sample** *procedure of Figure 2 at most* $m = m(\lambda)$ *times.*

Initialize$(1^\lambda, \mathcal{P}, d, \ell)$:	**Sample**():	**Challenge**():	**Finalize**(c'):
begin	**begin**	**begin**	**begin**
$P \leftarrow_\$ \mathbf{P}_\lambda$;	$f \leftarrow_\$ P_{\leq b}$;	$f \leftarrow_\$ P_{\leq b}$;	**return** $(c = c')$;
$G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d, \ell)$;	$f' \leftarrow f \bmod G$;	$f \leftarrow f - (f \bmod G)$;	**end**
$c \leftarrow_\$ \{0,1\}$;	**return** $f - f'$;	**if** $c = 0$ **then**	
$Q \leftarrow (P/\langle G\rangle)_{\leq b}$;	**end**	$r \leftarrow_\$ Q \setminus \{0\}$;	
return $(1^\lambda, P)$;		$f \leftarrow f + r$;	
end		**return** f;	
		end	

Fig. 2. Game $\mathsf{IM}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m}$.

We note that in the above definition we have excluded the zero element in the sampling of the remainder when coin c takes value 0. This is to ensure than an algorithm can have an overwhelming advantage in solving the IM problem.

We define a game $\mathsf{IM}'_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m}$ similarly to the game in Figure 2 except that the zero element is allowed when $c = 0$ in the **Challenge** procedure (i.e., $r \leftarrow_\$ Q \setminus \{0\}$ is replaced by $r \leftarrow_\$ Q$). The advantage of any adversary $\mathcal{A}$ in a modified IM' game can be easily related to that in the IM game.

Lemma 7. *It holds that* $\mathbf{Adv}^{\mathsf{im}'}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}}(\lambda) = \left(1 - \frac{1}{|Q|}\right) \cdot \mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}}(\lambda)$*, for any adversary* $\mathcal{A}$*.*

Proof. Let p be the probability that $\mathcal{A}$ outputs 0 when the remainder of the challenge polynomial modulo G is zero. Let p' denote the probability that $\mathcal{A}$ outputs 1 when this remainder is nonzero. We have:

$$\begin{aligned} 2 \cdot \Pr[\mathcal{A} \text{ wins the } \mathsf{IM} \text{ game}] - 1 &= p + p' - 1, \\ 2 \cdot \Pr[\mathcal{A} \text{ wins the } \mathsf{IM}' \text{ game}] - 1 &= p + \frac{1}{|Q|}(1-p) + \left(1 - \frac{1}{|Q|}\right) p' - 1 \\ &= \left(1 - \frac{1}{|Q|}\right) \cdot (p + p' - 1). \end{aligned}$$

□

We show below that under certain conditions the GB and IM problems are equivalent. Informally, the reduction of the GB problem to the IM problem works as follows. Consider an arbitrary element g_i in the Gröbner basis G. We can write g_i as $x_i^{d_i} - \tilde{g}_i$ for some $\tilde{g}_i < g_i$. Now, assume $x_i^{d_i} - r_i$ is in the ideal and that $r_i < x_i^{d_i}$, i.e. $\mathrm{LM}(x_i^{d_i} - r_i) = x_i^{d_i}$ and $x_i^{d_i} - r_i \in \langle G \rangle$. To find such an r_i we exhaustively search Q and hence require $|Q| = \mathrm{poly}(\lambda)$. Repeat this process for all $x_i^{d_i}$ and accumulate the results $x_i^{d_i} - r_i$ in a list $\tilde{G}$. The list $\tilde{G}$ is a list of elements in $\langle G \rangle$ with $\mathrm{LM}(\tilde{G}) = \mathrm{LM}(G)$ which implies that $\tilde{G}$ is a Gröbner basis. We note that this is the core idea behind the FGLM algorithm [36] which allows to efficiently change the ordering of a Gröbner basis given access to an oracle computing normal forms with probability 1 (and also "Bulygin's attack" in a different context [23]).

Lemma 8. (IM overwhelmingly easy $\implies$ GB overwhelmingly easy) *Suppose the quotient size $|Q|$ is polynomial in λ. Then for any* PPT *adversary $\mathcal{A}$ against the IM problem, there exists a* PPT *adversary $\mathcal{B}$ against the GB problem such that*

$$1 - \mathrm{poly}(\lambda) \cdot \left(1 - \mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}}(\lambda)\right) \leq \mathbf{Adv}^{\mathsf{gb}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m+1,\mathcal{B}}(\lambda),$$

where $\mathrm{poly}(\lambda) := |\mathrm{LM}(G)| \cdot (|Q| - 1)$*.*

(GB easy $\implies$ IM easy) *Conversely, for any* PPT *adversary $\mathcal{A}$ against the GB problem, there exists a* PPT *adversary $\mathcal{B}$ against the IM problem such that*

$$\mathbf{Adv}^{\mathsf{gb}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{B}}(\lambda).$$

Proof. Let us write $P_{\mathcal{A}}^{\mathsf{im}\text{-}0}$ (resp., $P_{\mathcal{A}}^{\mathsf{im}\text{-}1}$) for the success probability of any algorithm $\mathcal{A}$ against the IM problem conditioned on the event $c = 0$ (uniform challenge) (resp., $c = 1$). Various parameters are implicitly understood from the context. By the definition of advantage, we have

$$\mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}} = P_{\mathcal{A}}^{\mathsf{im}\text{-}0} + P_{\mathcal{A}}^{\mathsf{im}\text{-}1} - 1.$$

Now, to prove the first statement, we construct an algorithm $\mathcal{B}$ against the GB problem based on an algorithm $\mathcal{A}$ against the IM problem. This algorithm is described in Algorithm 3.

Algorithm 3: GB adversary $\mathcal{B}$ from IM adversary $\mathcal{A}$

begin
  $\mathcal{B}$ receives $(1^\lambda, P)$;
  $\tilde{G} \leftarrow \varnothing$;
  query GB.**Sample**() to get $f_0, \ldots, f_{m-1}$;
  query GB.**Sample**() to get f;
  for $m \in \mathrm{LM}(G)$ **do**
    for $b \in P/\mathcal{I}$ **do**
      run $\mathcal{A}(1^\lambda, P)$ as follows:
      if $\mathcal{A}$ *queries* IM.**Sample**() **then**
        answer $\mathcal{A}$'s ith query with f_i; `// we reuse` f_i `between different runs of` $\mathcal{A}$.
      if $\mathcal{A}$ *queries* IM.**Challenge**() **then**
        $\tilde{g}_i = m - b$;
        return $f + \tilde{g}_i$;
      if $\mathcal{A}$ *calls* IM.**Finalize**(c') **then**
        if $c' = 1$ **then** `//` $\tilde{g}_i$ `likely in` $\mathcal{I}$
          $\tilde{G} \leftarrow \tilde{G} \cup \{\tilde{g}_i\}$;
          break;
  call GB.**Finalize**$(\tilde{G})$;

We lower-bound the probability that algorithm $\mathcal{B}$ returns the correct Gröbner basis based on the success probability of $\mathcal{A}$. Note that if all of $\mathcal{A}$'s answers are correct, then $\mathcal{B}$'s output will be the Gröbner basis. Applying the union bound, we derive an upper bound on the failure probability of $\mathcal{B}$ by bounding the failure probability of $\mathcal{A}$ in each invocation. Let $\varepsilon = P_{\mathcal{A}}^{\mathsf{im}\text{-}0} + P_{\mathcal{A}}^{\mathsf{im}\text{-}1} - 1$ be $\mathcal{A}$'s advantage. Now consider invocations of $\mathcal{A}$ with $\tilde{g}_i = m - b \in \mathcal{I}$ within $\mathcal{B}$. Then on such a query, $\mathcal{A}$ is run in an environment with the challenge bit being 1. By definition, the probability of $\mathcal{A}$'s failure in this case is $1 - P_{\mathcal{A}}^{\mathsf{im}\text{-}1}$. Now consider invocations with $\tilde{g}_i \notin \mathcal{I}$. Since we iterate over all remainders, the *average* (over the choice of b, such that $\tilde{g}_i \notin \mathcal{I}$) failure probability for such invocations is $1 - P_{\mathcal{A}}^{\mathsf{im}\text{-}0}$. The union bound leads to an upper bound of

$$|\mathrm{LM}(G)|(1 - P_{\mathcal{A}}^{\mathsf{im}\text{-}1}) + |\mathrm{LM}(G)|(|Q| - 1)(1 - P_{\mathcal{A}}^{\mathsf{im}\text{-}0})$$

on the failure probability of $\mathcal{B}$, which in turn can be upper bounded by

$$|\mathrm{LM}(G)|(|Q| - 1)(2 - P_{\mathcal{A}}^{\mathsf{im}\text{-}1} - P_{\mathcal{A}}^{\mathsf{im}\text{-}0}) = |\mathrm{LM}(G)|(|Q| - 1)(1 - \varepsilon),$$

as desired.

Finally, it is easy to see that Algorithm 3 runs in polynomial time. The outer loop is repeated $|\mathrm{LM}(G)|$ and the inner loop $|P/\mathcal{I}|$ both of which are $\mathrm{poly}(\lambda)$. Algorithm $\mathcal{B}$ makes one additional query to **Sample** compared to $\mathcal{A}$ and hence needs $m+1$ samples.

Algorithm 4: IM adversary $\mathcal{B}$ from GB adversary $\mathcal{A}$

```
begin
    B receives (1^λ, P);
    Query IM.Challenge() to get h;
    Run A(1^λ, P) as follows:
    if A queries GB.Sample() then
        query IM.Sample() to get f;
        return f;
    if A calls GB.Finalize(G') then
        if G' is a red. Gröbner basis with the correct leading monomials then
            r ← h mod G';
            call IM.Finalize(1 − (r = 0));
        else
            c' ←$ {0,1};
            call IM.Finalize(c');
```

For the second statement, we construct $\mathcal{B}$ as in Algorithm 4. We use $\mathcal{A}$ to find a candidate Gröbner basis G'. If $G' = G$ we can compute the remainder r modulo the ideal spanned by the basis in polynomial time (cf. [30, p. 82]) and check if $r = 0$. So $\mathcal{B}$ will be successful whenever $\mathcal{A}$ is. By definition, the advantage of $\mathcal{B}$ is given by

$$\begin{aligned}\mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{B}}(\lambda) &= 2\cdot\Pr\left[\mathcal{B}\text{ successful}\right]-1\\ &= 2\left(\Pr\left[\mathcal{B}\text{ successful}\mid\mathcal{A}\text{ successful}\right]-\tfrac{1}{2}\right)\cdot\Pr\left[\mathcal{A}\text{ successful}\right]\\ &\quad+2\left(\Pr\left[\mathcal{B}\text{ successful}\mid\mathcal{A}\text{ not successful}\right]-\tfrac{1}{2}\right)\cdot\Pr\left[\mathcal{A}\text{ not successful}\right].\end{aligned}$$

The first summand is exactly what we need, so to finish the proof we need to show that the second summand is non-negative. This means, it remains to show that if $G' \neq G$, then $\mathcal{B}$ still has a non-negative advantage, i.e. $\mathcal{B}$ guesses c with probability at least $1/2$. Indeed, if G' does not have the correct form, $\mathcal{B}$ simply guesses the bit c (leading to a zero advantage). Moreover, if G' has the right form, reduction modulo G' gives rise to an $\mathbb{F}_q$-linear map $m_{G'} : P_{\leq b} \longrightarrow Q$, $f \mapsto f \bmod G'$. Since surjective linear maps preserve uniform distributions on finite-dimensional vector spaces, it follows that

$$\Pr_{f\leftarrow_{\$}P_{\leq b}}\left[m_{G'}(f)=0\right]=\frac{1}{|m_{G'}(P_{\leq b})|}\quad\text{and}\quad\Pr_{f\leftarrow_{\$}\mathcal{I}_{\leq b}}\left[m_{G'}(f)=0\right]=\frac{1}{|m_{G'}(\mathcal{I}_{\leq b})|}.$$

Since $\mathcal{I}_{\leq b} \subseteq P_{\leq b}$, we get

$$\Pr_{f\leftarrow_{\$}P_{\leq b}}\left[m_{G'}(f)=0\right]\leq\Pr_{f\leftarrow_{\$}\mathcal{I}_{\leq b}}\left[m_{G'}(f)=0\right].$$

Now, let

$$p_0 := \Pr_{f\leftarrow_{\$}P_{\leq b}}\left[f\in\mathcal{I}_{\leq b}\right]\quad\text{and}\quad p_1 := \Pr_{f\leftarrow_{\$}P_{\leq b}}\left[f\in P_{\leq b}\setminus\mathcal{I}_{\leq b}\right],$$

where $p_1 \neq 0$ since the quotient has positive dimension. Finally, let A be the event "$m_{G'}(f) = 0$." Then, since

$$\Pr_{f \leftarrow_\$ P_{\leq b}}[A] = p_0 \cdot \Pr_{f \leftarrow_\$ \mathcal{I}_{\leq b}}[A] \;+\; p_1 \cdot \Pr_{f \leftarrow_\$ P_{\leq b} \setminus \mathcal{I}_{\leq b}}[A]\,,$$

we get

$$\begin{aligned}
& p_0 \cdot \Pr_{f \leftarrow_\$ \mathcal{I}_{\leq b}}[A] + p_1 \cdot \Pr_{f \leftarrow_\$ P_{\leq b} \setminus \mathcal{I}_{\leq b}}[A] \leq \Pr_{f \leftarrow_\$ \mathcal{I}_{\leq b}}[A] \\
\iff \quad & p_1 \cdot \Pr_{f \leftarrow_\$ P_{\leq b} \setminus \mathcal{I}_{\leq b}}[A] \leq (1 - p_0) \cdot \Pr_{f \leftarrow_\$ \mathcal{I}_{\leq b}}[A] \\
\iff \quad & p_1 \cdot \Pr_{f \leftarrow_\$ P_{\leq b} \setminus \mathcal{I}_{\leq b}}[A] \leq p_1 \cdot \Pr_{f \leftarrow_\$ \mathcal{I}_{\leq b}}[A] \\
\iff \quad & \Pr_{f \leftarrow_\$ P_{\leq b} \setminus \mathcal{I}_{\leq b}}[A] \leq \Pr_{f \leftarrow_\$ \mathcal{I}_{\leq b}}[A]\,.
\end{aligned}$$

□

REMARK. Lemma 8 only proves a weak form of the equivalence between IM and GB. That is, for Lemma 8 to be meaningful we require that the IM adversary returns the correct answer with *overwhelming* probability. First, this is due to the restriction that **Sample** can only be called a bounded number of times, and thus we cannot amplify the success probability of the IM adversary through repetition. We note that it is possible to prove a stronger statement than Lemma 8 for $d = 1$ using the re-randomisation technique from [16]. Second, Lemma 8 does not address "structural errors" when $d > 1$, e.g. an IM oracle which decides based on partial information only. For example, assume $G = [x_0 + s_0 x_{n-1}, x_1 + s_1 x_{n-1}, \ldots, x_{n-1}^2 + s_{n-1} x_{n-1}]$ where $s_i \leftarrow_\$ \mathbb{F}_q$. This is a valid Gröbner basis generated by an algorithm satisfying the requirements for $\mathsf{GBGen}(\cdot)$. We have that $S(\mathcal{I}) = \{x_{n-1}, 1\}$ and by construction any $f \in P$ with a nonzero constant coefficient is not an element of $\mathcal{I} = \langle G \rangle$. Hence, it is easy – although not overwhelmingly so – to solve the IM problem by considering the constant coefficient only. On the other hand, the GB problem is still assumed to be hard, as it requires to recover all s_i.

3.1 Hardness assumptions

It is well known [12] that the worst-case complexity of the best algorithms of Gröbner bases computation is doubly exponential in the number of variables. However, in this work we are concerned with polynomial systems over finite fields, which do not achieve this worst-case complexity. In particular, we consider zero-dimensional ideals, i.e. ideals with a finite number of common roots. In order to make the paper self-content, we recall here a number of classical complexity results for these type of systems.

Lazard [50] showed that computing the Gröbner basis for a system of polynomials is equivalent to performing Gaussian elimination on so-called Macaulay matrices $\mathcal{M}_{d,m}^{\text{acaulay}}$ for $d, 1 \leq d \leq D$ for some D.

Definition 8 (Macaulay matrix). *For a set of m polynomials $f_0, \ldots, f_{m-1} \in P$, we define the Macaulay matrix $\mathcal{M}_{d,m}^{\text{acaulay}}$ of degree d as follows. Each column corresponds to one monomial. List "horizontally" all the degree $\leq d$ monomials from largest to smallest sorted by some fixed monomial*

ordering. The smallest monomial comes last. Multiply each f_i by all monomials $t_{i,j}$ of degree $d - d_i$ where $d_i = \deg(f_i)$. Finally, construct the coefficient matrix for the resulting system:

$$\mathcal{M}^{\text{acaulay}}_{d,m} := \begin{array}{c} \\ (t_{0,0}, f_0) \\ (t_{0,1}, f_0) \\ (t_{0,2}, f_0) \\ \vdots \\ (t_{1,0}, f_1) \\ \vdots \\ (t_{m-1,0}, f_{m-1}) \\ (t_{m-1,1}, f_{m-1}) \\ \vdots \end{array} \begin{array}{c} \text{monomials of degree } \leq d \\ \left(\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right) \end{array}.$$

Theorem 6. *Let $F = \{f_0, \ldots, f_{m-1}\}$ be a set of polynomials in P. There exists a positive integer D for which Gaussian elimination on all $\mathcal{M}^{\text{acaulay}}_{d,m}$ matrices for $d = 1, \ldots, D$ computes a Gröbner basis of $\langle F \rangle$.*

The F_4 algorithm [34] can be seen as another way to use linear algebra without knowing an a priori bound: it successively constructs and reduces matrices until a Gröbner basis is found. The same is true for the F_5 algorithm when considered in "F_4-style" [6,1]. Consequently, the complexity is bounded by the degree D and the number of polynomials considered at each degree. For F_5 [35] and the matrix-F_5 variant [38] we know that under some regularity assumptions all matrices have full rank which implies that the number of rows in the matrix is bounded by the number of columns. The number of monomials up to some degree d is bounded by $\binom{n+d}{n}$ and thus when considering some degree d the number of rows and columns of the matrices considered by F_5 is also bounded above by $\binom{n+d}{d}$. Thus, knowing the degree up to which F_5 has to compute provides an upper bound on the complexity of Gröbner bases. For this, the following definition [9] is useful.

Definition 9 (Semi-regular sequence of degree D). *Let $f_0, \ldots, f_{m-1}$ be homogeneous polynomials in P whose degrees are $d_0, \ldots, d_{m-1}$ respectively. We call this system a semi-regular sequence of degree D if:*

1. $\langle f_0, \ldots, f_{m-1} \rangle \neq \mathbb{F}[x_0, \ldots, x_{n-1}]$.
2. *For all $0 \leq i < m$ and $g \in \mathbb{F}[x_0, \ldots, x_{n-1}]$,*

$$(\deg(g \cdot f_i) < D \text{ and } g \cdot f_i \in \langle f_0, \ldots, f_{i-1} \rangle) \implies g \in \langle f_0, \ldots, f_{i-1} \rangle.$$

We call D the degree of semi-regularity of the system.

Definition 10 (Semi-regular sequence [9]). *Let $m > n$, and $f_0, \ldots, f_{m-1}$ be homogeneous polynomials of degree b in P generating an ideal $\mathcal{I}$. The system is said to be a semi-regular sequence if the Hilbert series [13] of $\mathcal{I}$ with respect to the degree reverse lexicographical order is*

$$H_{\mathcal{I}}(z) = \sum_{k \geq 0} c_k z^k = \frac{(1 - z^b)^m}{(1 - z)^n}.$$

Hence, for semi-regular sequences the degree of semi-regularity of the system is given by the index of the first non-positive coefficient of $H_{\mathcal{I}}(z)$.

This notion can be extended to affine polynomials by considering their homogeneous components of highest degree. It is conjectured that random systems are semi-regular with overwhelming probability. For semi-regular sequences, we have the following complexity result for F_5 [9,10,8].

Theorem 7. *Let $F = \{f_0, \ldots, f_{m-1}\}$ be a set of polynomials in P. Assuming that F is a semi-regular sequence, the complexity of the currently best known algorithm (i.e. F_5) to solve the Gröbner basis problem is given by*

$$\mathcal{O}\left(\binom{n+D}{D}^{\omega}\right)$$

where $2 \leq \omega < 3$ is the linear algebra constant, and D is the degree of semi-regularity of the system.

Asymptotic bounds for the degree of semi-regularity for semi-regular sequences of degree 2 can be found in [9]. These bounds for the degree of regularity lead to the following complexity estimates for Gröbner basis computations.

Corollary 2. *Let $c \geq 0$. Then for $m(\lambda) = c \cdot n(\lambda)$ (resp., $m(\lambda) = c \cdot n(\lambda)^2$) quadratic polynomials in some ideal $\mathcal{I} \subset \mathbb{F}_q[x_0, \ldots, x_{n-1}]$, the Gröbner basis of $\mathcal{I}$ can be computed in exponential (resp., polynomial) time in $n(\lambda)$.*

Lemma 8 states that the IM problem is equivalent to the GB problem if we have access to an IM oracle which succeeds with overwhelming probability. Although we cannot show this equivalence in general, we may assume that the two problems are indeed equivalent when $d = 1$ (cf. [16]):

Definition 11 (The GB and IM assumptions). *Let $\mathcal{P}$ be such that $n(\lambda) = \Omega(\lambda)$. Assume $b > 1$, $d = 1$, and that $m(\lambda) = c \cdot n(\lambda)$ for a constant $c \geq 1$. Then the advantage of any* PPT *algorithm in solving the GB or the IM problem is negligible as function of λ.*

4 Symmetric Polly Cracker: The Noise-Free Version

4.1 Homomorphic symmetric encryption

Syntax. An *arity-t homomorphic symmetric encryption scheme* is specified by four PPT algorithms as follows.

1. $\mathsf{Gen}(1^\lambda)$. This is the key-generation algorithm, and is run by the receiver. On input a security parameter, it outputs a (secret) key SK and an (evaluation) public key PK. This algorithm also outputs the descriptions of a pair of compatible spaces FunSp and MsgSp.
2. $\mathsf{Enc}(\mathsf{m}, \mathsf{SK})$. This is the encryption algorithm, and is run by the sender. On input a message m, and a key SK, it returns a ciphertext c.
3. $\mathsf{Eval}(\mathsf{c}_0, \ldots, \mathsf{c}_{t-1}, C, \mathsf{PK})$. This is the evaluation algorithm, and is run by an evaluator. On input t ciphertexts $\mathsf{c}_0, \ldots, \mathsf{c}_{t-1}$, a circuit C, and the public key, it outputs a ciphertext $\mathsf{c}_{\mathsf{evl}}$.

4. $\mathsf{Dec}(\mathsf{c}_{\mathsf{evl}}, \mathsf{SK})$. This is the deterministic decryption algorithm, and is run by the receiver. On input an (evaluated) ciphertext $\mathsf{c}_{\mathsf{evl}}$, a key SK, it returns either a message m or a special failure symbol $\perp$.

Correctness. A homomorphic symmetric encryption scheme is *correct* if for any polynomial p, any $\lambda \in \mathbb{N}$, any $(\mathsf{SK}, \mathsf{PK}) \in [\mathsf{Gen}(1^\lambda)]$, any $t = p(\lambda)$ messages $\mathsf{m}_i \in \mathsf{MsgSp}(\mathsf{PK})$, any circuit $C \in \mathsf{FunSp}(\mathsf{PK})$ of arity t, any t ciphertexts $\mathsf{c}_i \in [\mathsf{Enc}(\mathsf{m}_i, \mathsf{SK})]$, and any evaluated ciphertext $\mathsf{c}_{\mathsf{evl}} \in [\mathsf{Eval}(\mathsf{c}_0, \ldots, \mathsf{c}_{t-1}, C, \mathsf{PK})]$, we have that $\mathsf{Dec}(\mathsf{c}_{\mathsf{evl}}, \mathsf{SK}) = C(\mathsf{m}_0, \ldots, \mathsf{m}_{t-1})$. Depending on the context, the correctness condition might also be imposed over freshly created ciphertexts.

Compactness. A homomorphic encryption scheme is *compact* if there exists a fixed polynomial bound $\mathrm{B}(\cdot)$ so that for any $(\mathsf{SK}, \mathsf{PK}) \in [\mathsf{Gen}(1^\lambda)]$, any circuit $C \in \mathsf{FunSp}(\mathsf{PK})$, any t messages $\mathsf{m}_i \in \mathsf{MsgSp}(\mathsf{PK})$, any $\mathsf{c}_i \in [\mathsf{Enc}(\mathsf{m}_i, \mathsf{SK})]$, and any $\mathsf{c}_{\mathsf{evl}} \in [\mathsf{Eval}(\mathsf{c}_0, \ldots, \mathsf{c}_{t-1}, C, \mathsf{PK})]$, the size of $\mathsf{c}_{\mathsf{evl}}$ is at most $\mathrm{B}(\lambda + |C(\mathsf{m}_0, \ldots, \mathsf{m}_{t-1})|)$ (independently of the size of C).

The syntax of a homomorphic public-key encryption scheme is defined similarly, with the exception that the encryption algorithm takes the public key rather than the secret key as an input.

4.2 Description of the scheme

In this section we formally define the (noise-free) symmetric Polly Cracker encryption scheme. We present a family of schemes parametrised not only by the underlying computational polynomial ring scheme $\mathcal{P}$, but also by a Gröbner basis generation algorithm, which itself depends on a degree bound d, and a second degree bound b. However, to satisfy our security assumption (cf. Definition 11) we require $d = 1$. Our parameterised scheme, which we write as $\mathcal{SPC}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b}$, is presented in Figure 3. The message space is $Q = P_{\leq b}/\mathcal{I}$. As a vector space, Q is determined by the leading terms $\mathrm{LM}(G)$ alone and hence independent of the randomness of $\mathsf{GBGen}(\cdot)$. However, as a ring, Q is only independent of the randomness of $\mathsf{GBGen}(\cdot)$ if $d = 1$; in that case $Q = \mathbb{F}_q$. Here, Q as a ring being independent of the randomness of $\mathsf{GBGen}(\cdot)$ means that we can perform ring operations in Q such that the result, represented as an element of $Q \subset P$, can be computed without knowledge of G. For $d > 1$ this is not the case for multiplication.

$\underline{\mathsf{Gen}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b}(1^\lambda)}$:
begin
  $P \leftarrow_\$ \mathbf{P}_\lambda$;
  $G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d, \ell)$;
  $\mathsf{SK} \leftarrow (G, P, b)$;
  $\mathsf{PK} \leftarrow (P, b)$;
  return $(\mathsf{SK}, \mathsf{PK})$;
end

$\underline{\mathsf{Enc}(\mathsf{m}, \mathsf{SK})}$:
begin
  $f \leftarrow_\$ P_{\leq b}$;
  $f' \leftarrow f \bmod G$;
  $f \leftarrow f - f'$;
  $\mathsf{c} \leftarrow \mathsf{m} + f$;
  return c;
end

$\underline{\mathsf{Dec}(\mathsf{c}, \mathsf{SK})}$:
begin
  $\mathsf{m} \leftarrow \mathsf{c} \bmod G$;
  return m;
end

$\underline{\mathsf{Eval}(\mathsf{c}_0, \ldots, \mathsf{c}_{t-1}, C, \mathsf{PK})}$:
begin
  apply the Add and Mult gates of C over P;
  return the result;
end

Fig. 3. The (noise-free) Symmetric Polly Cracker scheme $\mathcal{SPC}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b}$.

CORRECTNESS OF EVALUATION. Let $d = 1$ and consider the two ciphertexts $\mathsf{c}_0 = \sum h_{0,j} g_j + \mathsf{m}_0$ and $\mathsf{c}_1 = \sum h_{1,j} g_j + \mathsf{m}_1$. Addition and multiplication of the two ciphertexts $\mathsf{c}_0, \mathsf{c}_1$ are given by

$$\begin{aligned}
\mathsf{c}_0 + \mathsf{c}_1 &= \sum h_{0,j} g_j + \mathsf{m}_0 + \sum h_{1,j} g_j + \mathsf{m}_1 \\
&= \sum (h_{0,j} + h_{1,j}) g_j + \mathsf{m}_0 + \mathsf{m}_1, \\
\mathsf{c}_0 \cdot \mathsf{c}_1 &= (\sum h_{0,j} g_j + \mathsf{m}_0) \cdot (\sum h_{1,j} g_j + \mathsf{m}_1) \\
&= (\sum h_{0,j} g_j) \cdot (\sum h_{1,j} g_j) + \sum h_{0,j} g_j \cdot \mathsf{m}_1 + \sum h_{1,j} g_j \cdot \mathsf{m}_0 + \ \mathsf{m}_0 \mathsf{m}_1 \\
&= \sum \tilde{h}_j g_j + \mathsf{m}_0 \mathsf{m}_1, \text{ for some } \tilde{h_j}.
\end{aligned}$$

The homomorphic features follow. Correctness of addition and multiplication for arbitrary numbers of operands follow from the associative laws of addition and multiplication in P.

COMPACTNESS. This scheme is not compact for general circuits. Although additions do not increase the size of the ciphertext, multiplications square the size of the ciphertext.

EFFICIENCY. If $d = 1$ and $q(\lambda) = \text{poly}(\lambda)$ we set $n(\lambda) = \Omega(\lambda)$ to rule out exhaustive search for the Gröbner basis $\{x_0 - b_0, \ldots, x_{n-1} - b_{n-1}\}$ where $b_i \in \mathbb{F}_q$. Message expansion is n^b with $b \geq 1$. That is, encrypting a single field element results in a ciphertext of length $\binom{n+b}{b} = \mathcal{O}(n^b)$ field elements. The complexity of both encryption and decryption for fresh ciphertexts are $\mathcal{O}(n^b)$ ring operations. Decryption of ciphertexts with μ levels of multiplications require $\mathcal{O}(n^{2^\mu b})$ ring operations.

4.3 Security

As we will show shortly, the above scheme only achieves a weak form of chosen-plaintext security where a limited number of ciphertexts can be eavesdropped on.

Definition 12 (m-IND-BCPA security). *The m-*IND-BCPA *security of a (homomorphic) symmetric-key encryption scheme $\mathcal{SE}$ for a polynomial m is defined by requiring that the advantage of any* PPT *adversary $\mathcal{A}$ given by*

$$\mathbf{Adv}^{\mathsf{ind\text{-}bcpa}}_{m,\mathcal{SE},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{IND\text{-}BCPA}^{\mathcal{A}}_{m,\mathcal{SE}}(\lambda) \Rightarrow \mathsf{True}\right] - 1$$

is negligible as a function of the security parameter λ. Game $\mathsf{IND\text{-}BCPA}_{m,\mathcal{SE}}$ is shown in Figure 4. The difference with the usual IND-CPA *security is that the adversary can query its encryption oracle at most $m(\lambda)$ times.*

We sometimes omit the subscript from schemes to ease notation. For example, in the result below, we have written $\mathcal{SPC}$ for $\mathcal{SPC}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b}$. The security guarantees are as follows.

Theorem 8. *Let $\mathcal{A}$ be a* PPT *adversary against the m-*IND-BCPA *security of the scheme in Figure 3. Then there exists a* PPT *adversary $\mathcal{B}$ against the* IM *problem such that for all $\lambda \in \mathbb{N}$:*

$$\mathbf{Adv}^{\mathsf{ind\text{-}bcpa}}_{m,\mathcal{SPC},\mathcal{A}}(\lambda) = \frac{2\,|Q|}{|Q| - 1} \cdot \mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{B}}(\lambda).$$

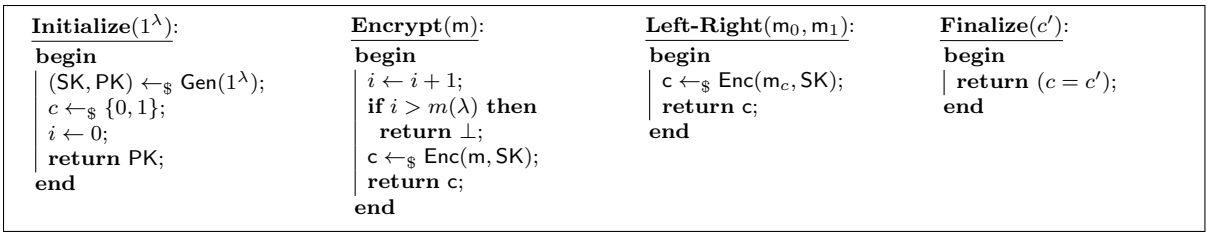

Fig. 4. Game $\mathsf{IND\text{-}BCPA}_{m,\mathcal{SE}}$. An adversary is legitimate if it calls oracle **Left-Right** exactly once on two message of equal lengths.

Conversely, let $\mathcal{A}$ be a PPT *adversary against the* IM *problem. Then there exists a* PPT *adversary $\mathcal{B}$ against the m-*IND-BCPA *security of the scheme in Figure 3 such that for all $\lambda \in \mathbb{N}$ we have*

$$\mathbf{Adv}^{\mathsf{im}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,m,\mathcal{A}}(\lambda) = \mathbf{Adv}^{\mathsf{ind\text{-}bcpa}}_{m,\mathcal{SPC},\mathcal{B}}(\lambda).$$

Proof. The second part of the lemma is clear: the **Sample** oracle is easily simulated by asking for encryptions of 0. The **Challenge** oracle is answered by querying **Left-Right** on $(0, r)$ where r is a uniformly chosen nonzero element of the quotient. Now deciding ideal membership directly leads to a distinguishing attack.

For the first part, we construct an algorithm $\mathcal{B}$ attacking the IM problem based on an algorithm $\mathcal{A}$ attacking the scheme as shown in Algorithm 5. To simplify the analysis, we compute the advantage of $\mathcal{B}$ in the IM' game and deduce the advantage of $\mathcal{B}$ in the IM game via Lemma 7.

Algorithm 5: IM adversary $\mathcal{B}$ from IND-BCPA adversary $\mathcal{A}$

```
1 begin
2     B receives (1^λ, P);
3     run A(1^λ, P) as follows;
4     if A queries IND-BCPA.Encrypt(m) then
5         query IM.Sample() to get f;   return f + m;
6     if A queries IND-BCPA.Left-Right(m_0, m_1) then
7         query IM.Challenge() to get f;   c ←$ {0,1};   return f + m_c;
8     if A calls IND-BCPA.Finalize(c') then
9         call IM.Finalize(c = c');
```

Now if the sample returned from the **Challenge** oracle in IM' to $\mathcal{B}$ is uniform in $P_{\leq b}$, then the probability that $c = c'$ is $1/2$. On the other hand, if the sample is an element of the ideal then adversary $\mathcal{A}$ is run in an environment which is identical to the m-IND-BCPA game. Hence in this case the probability that $c = c'$ is equal to the probability that $\mathcal{A}$ wins the m-IND-BCPA game. Switching from IM' to IM gives a factor $\frac{|Q|}{|Q|-1}$ by Lemma 7. The theorem follows. □

As a corollary, observe that when $m(\lambda) = \mathcal{O}(\lambda^b)$ one can use Corollary 2—which states that Gröbner bases are easy once $\mathcal{O}(n^b)$ elements from the ideal are available—to construct an adversary which breaks the $\mathsf{IND\text{-}BCPA}_{m,\mathcal{SE}}$ security of $\mathcal{SPC}$ in polynomial time. Thus we can only hope to achieve bounded security for this scheme.

5 Symmetric-to-Asymmetric Conversion

Given the security limitation of the symmetric Polly Cracker scheme, the goal for the rest of the paper is to convert the scheme to one which is not only fully IND-CPA-secure down to the problem of computing Gröbner bases but also is homomorphic and retains its generality. Once we achieve this, then it is possible to construct a public-key scheme using the additive homomorphic features of the symmetric scheme by applying various generic conversions. In section we pursue the less ambitious goal of constructing an additively homomorphic IND-CPA-secure public-key scheme from $\mathcal{SPC}$. In the literature there are two prominent conversions based on additive homomorphicity:

(A) Publish a set F_0 of encryptions of zero as (part of) the public key. To encrypt $m \in \{0,1\}$ compute $c = \sum_{f_i \in S} f_i + m$ where S is a sparse subset of F_0 [68].
(B) Publish two sets F_0 and F_1 of encryptions of zero and one as (part of) the public key. To encrypt $m \in \{0,1\}$ compute $c = \sum_{f_i \in S_0} f_i + \sum_{f_j \in S_1} f_j$, with S_0 and S_1 being sparse subsets of F_0 and F_1 respectively such that the parity of $|S_1|$ is m. Decryption checks whether $\mathsf{Dec}(c, \mathsf{SK})$ is even or odd [61].

The security of the above transformations rests upon the (computational) indistinguishability of asymmetric ciphertexts from those produced directly using the symmetric encryption algorithm. As noted above, since $\mathcal{SPC}$ is not IND-CPA-secure the above transformations cannot be used. However, when applied to a *specific* scheme, the transformations might still result in secure schemes. Yet, it can be shown that the security of the transformed schemes are *equivalent* to that of the underlying scheme. However, one could envisage a larger class of transformations which might lead to a fully secure additively homomorphic SE (or equivalently an additively homomorphic PKE) scheme. In this section we rule out a large class of such transformations. To this end, we consider PKE schemes which lie within the following design methodology.

1. The secret key is the Gröbner basis G of a zero-dimensional ideal $\mathcal{I} \subset P$. The decryption algorithm computes $c \bmod \mathcal{I} = c \bmod G$ (perhaps together with some post-processing such as a mod 2 operation). Thus, the message space is (essentially) Q. As before, we assume that $S(\mathcal{I})$—and hence Q as a vector space—is known.
2. The public key consists of elements $f_i \in P$. We assume that the remainders of these elements modulo the ideal $\mathcal{I}$, i.e. $r_i = f_i \bmod \mathcal{I}$, are known.
3. A ciphertext is computed using ring operations. In other words, it can be expressed as $f = \sum_{i=0}^{N-1} h_i f_i + r$. Here f_i are as in the public key, h_i are some polynomials (possibly depending on f_i), and r is an encoding of the message in Q.
4. The construction of the ciphertext does not encode knowledge of $\mathcal{I}$ beyond f_i. That is, we have

$$\left(\sum_{i=0}^{N-1} h_i f_i + r \right) \bmod \mathcal{I} = \sum_{i=0}^{N-1} h_i r_i + r.$$

   Hence we have that $\left(\sum_{i=0}^{N-1} h_i r_i + r \right) \in Q$ as an element of P.
5. The security of the scheme relies on the fact that elements f produced at step (3) are computationally indistinguishable from random elements in $P_{\leq b}$.

Although conditions 1–3 impose natural algebraic restrictions on the construction, and condition 5 provides a standard way to argue for security, condition 4 imposes some real restrictions on the set of allowed transformation, but strikes a reasonable balance between allowing a general statement without ruling out too large a class of conversions. It requires that the r_i and r do not encode any information about the secret key. We currently require this restriction on the "expressive power" of r_i and r so as to make a general impossibility statement. If r_i and r produce a nonzero element in $\mathcal{I}$ using some arbitrary algorithm $\mathcal{A}$, we are unable to prove anything about the transformation. Furthermore, it is plausible that for any given $\mathcal{A}$ a similar impossibility result can be obtained if the remaining conditions hold (although we were unable to prove this).

Note that the two transformations listed above are special linear cases of this methodology. For transformation (A) we have that $f_i \in \mathcal{I}$ (hence $r_i = 0$), $h_i \in \{0, 1\}$, and $r = m$. For transformation (B) we have $r_i = 0$ if $f_i \in F_0$, $r_i = 1$ if $f_i \in F_1$, $h_i \in \{0, 1\}$, and $r = 0$.

To show that any conversion of the above form cannot lead to an IND-CPA-secure public-key scheme, we will use the following theorem from commutative algebra which was already used in [11] to discourage the use of Gröbner bases in the construction of public-key encryption schemes.

Theorem 9 (Dickenstein et al. [32]). *Let $\mathcal{I} = \langle f_0, \ldots, f_{m-1}\rangle$ be an ideal in the polynomial ring $P = \mathbb{F}[x_0, \ldots, x_{n-1}]$, h be such that $\deg(h) \leq D$, and let $h - (h \bmod \mathcal{I}) = \sum_{i=0}^{m-1} h_i f_i$, where $h_i \in P$ and $\deg(h_i f_i) \leq D$. Let G be the output of some Gröbner basis computation algorithm up to degree D (i.e. all computations with degree greater than D are ignored and dropped). Then $h \bmod \mathcal{I}$ can be computed by polynomial reduction of h via G.*

The main result of this section is a consequence of the above theorem. It essentially states that uniformly sampling elements of the ideal up to some degree is equivalent to computing a Gröbner basis for the ideal. Note that Theorem 9 in itself does not provide this result, since there is no assumption about the "quality" of h. Hence, to prove this result we first show that the above methodology implies sampling as in Theorem 9 but with uniformly random output. Theorem 9 then allows us to compute normal forms, which in turn allows deciding ideal membership with success probability 1. This together with the fact that h is random allows us to compute a Gröbner basis by Lemma 8. Note that although we arrive at the same impossibility result using Corollary 2, the approach taken below better highlights the structure of the underlying problem.

Theorem 10. *Let $G = \{g_0, \ldots, g_{n-1}\}$ be the reduced Gröbner basis of a zero-dimensional ideal $\mathcal{I} \subset P$ where each $\deg(g_i) \leq d$. Assume that $S(\mathcal{I})$ is known and that $Q = P_{\leq b}/\mathcal{I}$ has s elements. Furthermore, let $F = \{f_0, \ldots, f_{N-1}\}$ be a set of polynomials with known $r_i = f_i \bmod \mathcal{I}$. Let $\mathcal{A}$ be a* PPT *algorithm which given F produces elements $f = \sum h_i f_i + r$ with $\deg(f) \leq b, h_i \in P, b \leq B, \deg(h_i f_i) \leq B$, and $(f \bmod \mathcal{I}) = \sum h_i r_i + r$. Suppose further that the outputs of $\mathcal{A}$ are computationally indistinguishable from random elements in $P_{\leq b}$. Then there exists an algorithm which computes a Gröbner basis for $\mathcal{I}$ from F in $\mathcal{O}\big(n^{\omega B} + |\mathrm{LM}(G)| \cdot s \cdot n^{2b}\big)$ field operations.*

Proof. Let $f = \sum_{i=0}^{N-1} h_i f_i + r$. Writing $\tilde{f}_i = f_i - r_i$, we get that $h = f - (f \bmod \mathcal{I}) = \sum_{i=0}^{N-1} h_i \tilde{f}_i + \tilde{r}$ for some $\tilde{r} \in P_{\leq b}/\mathcal{I}$. Hence h satisfies the condition of Theorem 9, and we can compute the remainder of all elements of degree b produced by $\mathcal{A}$ by computing a Gröbner basis up to degree B. From Theorem 7 we know that this costs $\mathcal{O}\big(n^{\omega B}\big)$ field operations. □

We now have an algorithm which returns the remainder for arbitrary elements of $P_{\leq b}$ with probability 1. This follows since h is computationally indistinguishable from random elements in $P_{\leq b}$. More explicitly, we can generate the system parameters, including the Gröbner basis, and provide the algorithm with either an output of $\mathcal{A}$ or a random element. We can check for the correctness of the answer using the basis. Any non-negligible difference in algorithm's success rate translates to a break of the indistinguishability of the outputs of $\mathcal{A}$.

Now given an algorithm which computes normal forms, it is trivial to construct an algorithm which decides ideal membership: compute the normal form and compare with zero. Furthermore, by Lemma 8, deciding ideal membership with overwhelming probability is equivalent to computing a Gröbner basis by making at most $|\mathrm{LM}(G)| \cdot s$ queries to the IM oracle where both $|\mathrm{LM}(G)|$ and s are $\mathrm{poly}(n)$. Note that the IM oracle constructed here has success probability 1. Each IM query costs at most $\binom{n+b}{b}^2 = \mathcal{O}\big(n^{2b}\big)$ field operations. Therefore the overall cost of the second step is $\mathcal{O}\big(|\mathrm{LM}(G)| \cdot s \cdot n^{2b}\big)$. In fact, this last step is unnecessary, since it can be shown that the output of the Gröbner basis computation up to degree B *is* a Gröbner basis for $\mathcal{I}$. In any case, the overall complexity is $\mathcal{O}\big(n^{\omega B}\big)$ for the first step and $\mathcal{O}\big(|\mathrm{LM}(G)| \cdot s \cdot n^{2b}\big)$ for the second step with $b \leq B$ from which an overall complexity of $\mathcal{O}\big(n^{\omega B} + |\mathrm{LM}(G)| \cdot s \cdot n^{2b}\big)$ follows. □

Therefore, if for some degree $b \geq d$ computationally uniform elements of $P_{\leq b}$ can be produced using the public key $f_0, \ldots, f_{N-1}$, there is an attacker which recovers the secret key $g_0, \ldots, g_{n-1}$ in essentially the same complexity. Hence, while conceptually simple and provably secure up to some bound, our symmetric Polly Cracker scheme $\mathcal{SPC}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b}$ does not provide a valid building block for constructing a fully homomorphic public-key encryption scheme. We also stress that $\mathcal{SPC}$ is secure down to the IM problem with noticeable advantage, but in order to construct an adversary against the GB problem we need an IM oracle with overwhelming advantage.

Remark. Although the above impossibility result is presented for public-key encryption schemes, due to the equivalence result of [61], it also rules out the existence of additively homomorphic symmetric Polly Cracker-style schemes with full $\mathsf{IND\text{-}CPA}$ security.

Our goal now is to achieve full $\mathsf{IND\text{-}CPA}$ security for a symmetric Polly Cracker-type scheme down to the hardness of computing Gröbner bases. To this end, we introduce noisy variants of GB and IM in the next section. These variants ensure that the conditions of Theorem 10 do not hold any more. In particular, the condition that $r_i = f_i \bmod \mathcal{I}$ are known will be no longer valid.

6 Gröbner Bases with Noise

In this section, we introduce noisy variants of the problems presented in Section 3. The goal is to lift the restriction on the number of samples that the adversary can obtain and, following a similar design methodology to Polly Cracker, construct an $\mathsf{IND\text{-}CPA}$-secure scheme. Put differently, we consider problems that naturally arise if we consider noisy encoding of messages in $\mathcal{SPC}$. Similarly to [68,60] we expect a problem which is efficiently solvable in the noise-free setting to be also hard in the noisy setting. We will justify this assumption in Section 6.1 by arguing that our construction can be seen as a generalisation of [68,60].

The games below will be parametrised by a noise distribution χ. The discrete Gaussian distribution is of particular interest to us.

Definition 13 (Discrete Gaussian distribution). *Let $\alpha > 0$ be a real number and $q \in \mathbb{N}$. The discrete Gaussian distribution $\chi_{\alpha,q}$, is a Gaussian distribution rounded to the nearest integer and reduced modulo q with mean zero and standard deviation αq.*

In what follows we assume that χ is defined over Q, i.e. for $d > 1$ we have that χ is a multidimensional noise distribution. For example, χ may simply consist of $|S(\mathcal{I})_{\leq b}|$ independent discrete Gaussian distributions, one for each $m \in S(\mathcal{I})_{\leq b}$. However, as pointed out in [52] simply using the same Gaussian on each monomial is possibly not the best choice. Another notable special case is $q = 2$. In this case, $\chi_{\alpha,2}$ is a Bernoulli distribution with just one parameter $0 < p < 1$, the probability that 1 is returned.

We now define a noisy variant of the Gröbner basis problem. The task here is still to compute a Gröbner basis for some ideal $\mathcal{I}$. However, we are now only given access to a noisy sample oracle which provides polynomials which are not necessarily in $\mathcal{I}$ but rather are "close" approximations to elements of $\mathcal{I}$. Here the term "close" is made precise using a noise distribution χ on Q.

Definition 14 (The Gröbner basis with noise (GBN) problem). *The Gröbner basis with noise problem is defined through game $\mathsf{GBN}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}$ shown in Figure 5. The advantage of a* PPT *algorithm $\mathcal{A}$ in solving the* GBN *problem is*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{A}}(\lambda) := \Pr\left[\mathsf{GBN}^{\mathcal{A}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}(\lambda) \Rightarrow \mathsf{True}\right].$$

Initialize$(1^\lambda, \mathcal{P}, d)$:	**Sample**():	**Finalize**(G'):
begin	**begin**	**begin**
$P \leftarrow_\$ \mathbf{P}_\lambda$;	$f \leftarrow_\$ P_{\leq b}$;	**return** $(G = G')$;
$G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d, \ell)$;	$e \leftarrow_\$ \chi$;	**end**
return $(1^\lambda, P)$;	$f \leftarrow f - (f \bmod G) + e$;	
end	**return** f;	
	end	

Fig. 5. Game $\mathsf{GBN}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}$.

The essential difference between the noisy and noise-free versions of the Gröbner basis problem is that by adding noise we have eliminated the restriction on the adversary to call the **Sample** oracle a bounded number of times. Put differently, if χ is the delta distribution, the GBN problem degenerates to the GB problem with an unbounded number of samples. Hence, in this case the GBN problem is easy. On the other hand if χ is uniform, the GBN problem is information-theoretically hard. Thus, the choice of χ greatly influences the hardness of the GBN problem. We leave an in-depth investigation of the noise parameter to future work.

As in the noise-free setting, we can ask various questions about the ideal $\mathcal{I}$ generated by G. One such example is solving the ideal membership problem with access to noisy samples from $\mathcal{I}$. In our definition the adversary wins the game if it can distinguish whether an element was sampled uniformly from $P_{\leq b}$ or from $\mathcal{I}_{\leq b} + \chi$.

Definition 15 (The ideal membership with noise (IMN) problem). *The ideal membership with noise problem is defined through game $\mathsf{IMN}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}$ shown in Figure 6. The advantage of a* PPT *algorithm $\mathcal{A}$ in solving the* IMN *problem is defined by*

$$\mathbf{Adv}^{\mathsf{imn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{IMN}^{\mathcal{A}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}(\lambda) \Rightarrow \mathsf{True}\right] - 1.$$

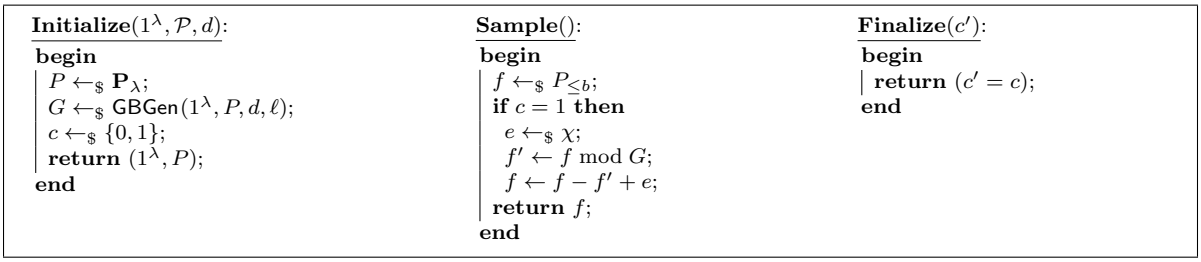

Fig. 6. Game $\mathsf{IMN}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}$. The adversary may call **Sample** multiple times.

Our definition of the IMN problem can be seen as an instantiation of Gentry's ideal coset problem [40] since both problems require distinguishing uniformly chosen elements in $P_{\leq b}$ from those in $\mathcal{I}_{\leq b} + \chi$.

Now, a pressing question is equivalence of the GBN and the IMN problem, i.e. decision-to-search reduction. We have been able to prove this equivalence in the special case $d = 1$ (see below). Intuitively, a straightforward reduction fails when $d > 1$ because an IMN oracle does not have to consider the coefficient of every monomial in $S(\mathcal{I})_{\leq b}$ when deciding ideal membership to be successful. On the other hand, a GBN oracle must recover coefficients of all monomial. For example, let $G = \{x_0 + s_0 x_{n-1}, \ldots, x_{n-2} + s_{n-2} x_{n-1}, x_{n-1}^2 + s_{n-1} x_{n-1}\}$ and hence $S(\mathcal{I})_{\leq b} = \{x_{n-1}, 1\}$. Assume the noise distribution χ is such that the coefficient for x_{n-1} of the noise is uniform $\in \mathbb{F}_q$, while the constant coefficient is always zero. For this shape and noise, it is easy to solve the IMN problem: any $f \in P_{\leq b}$ with a nonzero constant coefficient is not an element of $\mathcal{I} = \langle G \rangle$. However, turning this oracle into an adversary against the GBN problem would require to recover all s_i which are not even considered by IMN. Furthermore, the coefficients of x_{n-1} are information-theoretically hidden, so the distribution on $I_{\leq b} + \chi$ does not even depend on the s_i (cf. [48]).

In fact, this type of counterexample is essentially the only thing that can go wrong: for a weaker variant of the IMN problem, which we aptly call the weak IMN problem (and define below in such a way to ensure that the adversary has to consider all monomials) we are able to show a reduction to the GBN problem. In this definition we let $s := \dim(Q) = |S(\mathcal{I})_{\leq b}|$, which is independent of the randomness of $\mathsf{GBGen}(\cdot)$. Given χ, we also define the distributions χ^t for $t \in S(\mathcal{I})_{\leq b}$ by sampling an element e from Q according to χ and setting all but the coefficient corresponding to t to some independent uniform values.[2] Hence, all coefficients except that corresponding to t are information-theoretically blinded. Any algorithm which can distinguish samples following $\mathcal{I}_{\leq b} + \chi^t$ from uniform samples in $P_{\leq b}$ for *all* t can be used to solve the GBN problem.

Definition 16 (The weak ideal membership with noise (WIMN) problem). *The weak ideal membership with noise problem is defined through games* $\mathsf{WIMN}_{G,\chi,t^\star}$ *for* $t^\star \in S(\mathcal{I})_{\leq b}$ *shown in*

[2] Since the noise distribution χ only enters our construction via χ^t, this has the side effect of removing all dependencies between the coefficients. In particular, we may as well assume that χ samples the coefficients of all monomials t independently.

Figure 7. The advantage of a PPT *algorithm* $\mathcal{A}$ *in solving the* WIMN *problem is defined by*

$$\mathbf{Adv}^{\mathsf{wimn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{A}}(\lambda) := \mathop{\mathbb{E}}_{G}\left[\min_{t^\star}\left(2\cdot\Pr\left[\mathsf{WIMN}^{\mathcal{A}}_{G,\chi,t^\star}(\lambda)\Rightarrow\mathsf{True}\right]-1\right)\right],$$

where the expectation is taken over G *sampled from* $\mathsf{GBGen}(1^\lambda, P, d, \ell)$.

Initialize$(G,\chi,t^\star)$:
begin
  $c \leftarrow_\$ \{0,1\}$;
  return $(1^\lambda, P, t^\star)$;
end

Sample(t):
begin
  $f \leftarrow_\$ P_{\le b}$;
  $e \leftarrow_\$ \chi^t$;
  $f' \leftarrow f \bmod G$;
  $f \leftarrow f - f' + e$;
  return f;
end

Challenge$_{t^\star}()$:
begin
  $f \leftarrow_\$ P_{\le b}$;
  if $c = 1$ **then**
    $e \leftarrow_\$ \chi^{t^\star}$;
    $f' \leftarrow f \bmod G$;
    $f \leftarrow f - f' + e$;
  return f;
end

Finalize(c'):
begin
  return $(c' = c)$;
end

Fig. 7. Games $\mathsf{WIMN}_{G,\chi,t^\star}$. In each game, the adversary may call **Sample** with any monomials $t \in S(\mathcal{I})_{\le b}$ multiple times and **Challenge**$_{t^\star}$ once.

Our definition of advantage is somewhat non-standard but it bears similarities to game definitions in recent work on multi-instance security [14]. Indeed, we require that only those WIMN adversaries win the overall game which work for *all* $t^\star \in S(\mathcal{I})_{\le b}$ for a particular Gröbner basis G. As we shall see, only such adversaries allow us to recover the full Gröbner basis. Also, we note that the term "weak" is justified by the relation between WIMN and IMN. It is easy to see that if the IMN problem is hard, then so is the WIMN problem, while, as we have seen, the converse is not necessarily true. Finally, if $d = 1$ the IMN and WIMN problems are equivalent. We answer queries to WIMN's **Challenge**$_{t^\star}$ and **Sample** oracles with answers from IMN's **Sample** oracle. If IMN's **Sample** follows $I + \chi$, WIMN runs in the right environment. If IMN's **Sample** follows a uniform distribution on $P_{\le b}$ then WIMN's receives no information about the problem instance and hence has advantage zero.

We next show that when q and $|S(\mathcal{I})_{\le b}|$ are polynomial in λ and the WIMN problem is hard, the GBN problem is also hard. The intuition behind the reduction is that the adversary can exhaustively search for the coefficients for each monomial $t \in S(\mathcal{I})_{\le b}$ *independently* and then use the WIMN solver for t to verify its guess. This is formalised in the lemma below.

Lemma 9. (WIMN easy $\implies$ GBN easy) *Suppose the finite field size* q *and* $|S(\mathcal{I})_{\le b}|$ *are polynomial in* λ. *Then for any polynomial* p *and any* PPT *adversary* $\mathcal{A}$ *against the* WIMN *problem there exists a* PPT *adversary* $\mathcal{B}$ *against the* GBN *problem such that*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{B}}(\lambda) \ge (1-\mathrm{o}(1))\cdot\mathbf{Adv}^{\mathsf{wimn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{A}}(\lambda) + \mathrm{negl}(\lambda)$$

for all values of $\lambda \in \mathbb{N}$ *such that*

$$\mathbf{Adv}^{\mathsf{wimn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{A}}(\lambda) \ge \tfrac{1}{p(\lambda)}.$$

Proof. We construct an adversary $\mathcal{B}$ against GBN from adversary $\mathcal{A}$ against WIMN in Algorithm 6. Algorithm 6 runs $\mathcal{A}$ in the environment it expects. This is clear for calls to the **Sample**(t) oracle.

For the $\mathbf{Challenge}_{t^\star}()$ oracles, first assume that we guessed correctly and β is the coefficient of $t^\star$ of $g_i \in G$ with leading monomial m. Since g_i and $m + \beta \cdot t^\star$ differ only by an element in Q with coefficient 0 for $t^\star$, which is blinded by e, the challenge $f + a \cdot (m + \beta \cdot t^\star) + e$ follows the same distribution as $f + a \cdot g_i + e$. This in turn has the same distribution as $\mathcal{I} + \chi^{t^\star}$ as required. On the other hand, if we guessed incorrectly then $f + a \cdot (m + b \cdot t^\star) + e$ is uniform as a is independently uniform in $\mathbb{F}_q$ and e is independently uniform for all coefficients of the quotient $\neq t^\star$. Hence, $\mathcal{A}$ either sees elements following $\mathcal{I} + \chi^{t^\star}$ or uniform in $P_{\leq b}$.

Algorithm 6: GBN adversary $\mathcal{B}$ from WIMN adversary $\mathcal{A}$

begin
  $\mathcal{B}$ receives $(1^\lambda, P)$;
  $G' \leftarrow \varnothing$;
  for $m \in \mathrm{LM}(G)$ **do**
    $r \leftarrow 0$;
    for $t^\star \in S(\mathcal{I})_{\leq b}$ **do**
      for $\beta \in \mathbb{F}_q$ **do**
        $C_\beta \leftarrow 0$;
        for $i \in \{1, \ldots, \lambda p(\lambda)^2\}$ **do** `// amplification`
          run $\mathcal{A}(1^\lambda, P, t^\star)$ as follows:
          if $\mathcal{A}$ *queries* WIMN.**Sample**(t) **then**
            query GBN.**Sample**() to get f;
            $e \leftarrow$ random polynomial $\in Q$ but coefficient 0 for t;
            answer $\mathcal{A}$'s query with $f + e$;
          if $\mathcal{A}$ *queries* WIMN.$\mathbf{Challenge}_{t^\star}()$ **then**
            $a \leftarrow_\$ \mathbb{F}_q$;
            $e \leftarrow$ random polynomial $\in Q$ but coefficient 0 for $t^\star$;
            query GBN.**Sample**() to get f;
            return $f + a \cdot (m + \beta \cdot t^\star) + e$;
          if $\mathcal{A}$ *calls* WIMN.**Finalize**(c') **then**
            if $c' = 1$ **then** `//` $\beta \cdot t^\star$ `likely a correct guess`
              $C_\beta \leftarrow C_\beta + 1$;
              break;
      $r \leftarrow r + \widetilde{\beta} \cdot t^\star$ for a maximal $C_{\widetilde{\beta}}$; `// majority vote`
    $G' \leftarrow G' \cup \{m + r\}$;
  call GBN.**Finalize**(G');

Algorithm 6 runs in polynomial time. The outer loop terminates after at most $|\mathrm{LM}(G)| = n(\lambda)$ iterations. The two inner loops terminate after at most $|S(\mathcal{I})|$ and q iterations, both of which are polynomial in λ by assumption.

Algorithm 6 is correct and returns the reduced Gröbner basis G' such that $\langle G' \rangle = \mathcal{I}$. For this, first note that whenever we make a fixed wrong guess β, the distribution of the challenge $f + a \cdot (m + \beta \cdot t^\star) + e$ presented to $\mathcal{A}$ does not depend on the value of β due to the multiplication by the uniform a. As a consequence, we can amplify the success probability of $\mathcal{A}$ by $r(\lambda) = p(\lambda)^2\lambda$-fold repetition: Call G good if the minimal (over $t^\star$) conditional advantage of $\mathcal{A}$ (conditioned on G) is at least $\frac{1}{p(\lambda) \log \lambda}$. By a standard argument, the probability that G is good must then be at least

$(1-\frac{1}{\log\lambda})\mathbf{Adv}^{\mathsf{wimn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{A}}(\lambda)$. For those good G, our choice of number of repetitions r is large enough to amplify the advantage $\mathcal{A}$ to overwhelming using Chernoff–Hoeffding bounds. Hence in each majority vote we will pick the correct value with overwhelming probability, so $G' = G$ holds with overwhelming probability for good G, which finishes the proof. □

REMARK. In the lemma above, we were able to amplify the success probability of any adversary which solves the WIMN problem with non-negligible advantage to one which has an overwhelming advantage via Chernoff bounds, since we have no a priori bound on the number of queries as we had in the noiseless setting. In contrast to Lemma 8 we also do not require $|Q|$ to be polynomial in λ but only $|S(\mathcal{I})_{\leq b}|$ and the field size q. Finally, because WIMN treats every monomial $t^\star$ independently, "structural errors" as in described after Lemma 8 are ruled out.

We note that when $d = 1$ Lemma 9 implies IMN is hard if GBN is hard, as in this case WIMN and IMN are equivalent. Furthermore, it is easy to see that in this case a converse reduction can also be constructed because a WIMN adversary expects samples from $\mathbf{Sample}(1)$ which GBN's **Sample** oracle returns. Hence, for $d = 1$ we an equivalence between the IMN, WIMN, and GBN problems holds. Moreover, in this case, we can also demonstrate an average-case-to-worst-case reduction analogous to that known for the LWE problem. That is, for $d = 1$ we show below that if we can solve the IMN problem for a polynomial fraction of instances, then we can also solve it for all instances.

Lemma 10 (Average-case-to-worst-case reduction for $d = 1$). *Let $\mathcal{A}$ be a* PPT *adversary against $\mathsf{GBN}_{\mathcal{P},\mathsf{GBGen}_{\mathsf{dense}}(\cdot),1,\ell,b,\chi}$ that is successful for a polynomial fraction of all secrets. Then there exists a* PPT *adversary $\mathcal{B}$ which solves $\mathsf{GBN}_{\mathcal{P},G,1,\ell,b,\chi}$ on* all *instances G. That is, the basis is no longer sampled at random, but is fixed to be a specific value G. More precisely*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},G,1,\ell,b,\chi,\mathcal{B}}(\lambda) = 1 - \mathrm{negl}(\lambda)$$

for all values of G, provided that $\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}_{\mathsf{dense}}(\cdot),1,\ell,b,\chi,\mathcal{A}}(\lambda) > \frac{1}{p(\lambda)}$ for a polynomial p.

Proof. The proof is similar to the proof of [60, Lemma 3.2]. The idea is to find a suitable class of transformations which allow us to randomise a specific Gröbner basis G. We remark that G should be a valid output of $\mathsf{GBGen}_{\mathsf{dense}}()$ for $d = 1$: it is of the form $G = G_{\mathbf{s}} := [x_0 - s_0, \ldots, x_{n-1} - s_{n-1}]$ where $s_i \in \mathbb{F}$. We also denote by $\mathcal{I}_{\mathbf{s}}$ the ideal generated by $G_{\mathbf{s}}$ and by $\mathcal{J} = \mathcal{I}_{\mathbf{s},\leq b} + \chi$ the probability distribution on $P_{\leq b}$ presented to $\mathcal{B}$. We consider the transformation $L_{\mathbf{t}} : P \longrightarrow P$ defined by $L_{\mathbf{t}}(f) := f(\mathbf{t})$ for any $\mathbf{t} := (x_0 - t_0, x_1 - t_1, \ldots, x_{n-1} - t_{n-1})$ with $t_i \in \mathbb{F}_q$.

We remark that the image $\mathcal{I}'$ of $\mathcal{I}_{\mathbf{s}}$ under $L_{\mathbf{t}}$ is $\mathcal{I}_{\mathbf{s}+\mathbf{t}}$ (i.e. the ideal generated by $G_{\mathbf{s}+\mathbf{t}}$). Indeed, since $L_{\mathbf{t}}$ is a bijection, there is a one-to-one correspondence between the zeroes of $\mathcal{I}'$ and $\mathcal{I}_{\mathbf{s}}$. This implies that the variety corresponding to $\mathcal{I}'$ only consists of the single element $\mathbf{s} + \mathbf{t}$. Therefore, the unique Gröbner basis of $\mathcal{I}'$ is $G_{\mathbf{s}+\mathbf{t}}$, and $\mathcal{I}' = \mathcal{I}_{\mathbf{s}+\mathbf{t}}$. Therefore, $L_{\mathbf{t}}$ allows to map $\mathcal{I}_{\mathbf{s}} + \chi$ to $\mathcal{I}_{\mathbf{s}+\mathbf{t}} + \chi$. It is clear that $G_{\mathbf{s}+\mathbf{t}}$ is a valid output of $\mathsf{GBGen}_{\mathsf{dense}}()$. Moreover, the distribution of $G_{\mathbf{s}+\mathbf{t}}$ is uniform on the image of $\mathsf{GBGen}_{\mathsf{dense}}()$.

Now, we use $\mathcal{A}$ a polynomial number of times on $L_{\mathbf{t}}(\mathcal{J})$, each using a freshly chosen and uniform $\mathbf{t} \leftarrow_{\$} \mathbb{F}^n$. With overwhelming probability, $\mathcal{A}$ will output the correct Gröbner basis $G_{\mathbf{s}+\mathbf{t}}$ at least once, from which we can recover $\mathbf{s}$ (and hence $G_{\mathbf{s}}$) and verify against $\mathcal{J}$.

Note that we can verify in PPT whether a given $\mathbf{s}'$ is correct for $\mathcal{J} = \mathcal{I}_{\mathbf{s},\leq b} + \chi$. Indeed, as soon as $\mathbf{s} = \mathbf{s}'$ then $\mathcal{J}(\mathbf{s}')$—the evaluation of polynomials distributed according to $\mathcal{J}$ on $\mathbf{s}'$—is distributed according to χ, because $\mathcal{I}_{\mathbf{s}}(\mathbf{s}') = 0$. In contrast, whenever $\mathbf{s} \neq \mathbf{s}'$, $\mathcal{J}(\mathbf{s}') = \mathcal{I}_{\mathbf{s},\leq b}(\mathbf{s}') + \chi$ is uniform. This is the case, because the evaluation of a polynomial at a given point is a surjective linear map from $\mathcal{I}_{\mathbf{s},\leq b}$ to $\mathbb{F}$, and such maps preserve the uniform distribution. Hence, to verify that a given $\mathbf{s}'$ is correct for $\mathcal{J}$ we have to decide whether each polynomial in $\mathcal{J}(\mathbf{s}')$ is uniform or follows χ. We can obtain overwhelming confidence in polynomial time if χ is a Gaussian distribution [59, Lemma 3.6].

More generally, we may use $\mathcal{A}$ to verify in PPT whether a given $\mathbf{s}'$ is correct for $\mathcal{J} = \mathcal{I}_{\mathbf{s},\leq b} + \chi$ for any distribution χ. Indeed, to verify $\mathbf{s}'$ against $\mathbf{s}$, we test whether the probability that $\mathcal{A}$ returns $G_{\mathbf{u}}$ when presented with samples from $\mathcal{I}_{\mathbf{u},\leq b} + \mathcal{J}(\mathbf{s}')$ for a uniform $\mathbf{u} \in \mathbb{F}^n$ is $\geq \frac{1}{p}$ or equal to $\frac{1}{q^n}$. Since $\frac{1}{p} - \frac{1}{q^n}$ is noticeable, we can obtain overwhelming confidence in polynomial time. □

We note that this proof strategy does not apply to $d > 1$ for two reasons. First, it is not necessarily true that we have more maps $L_{\mathbf{t}}$ than secrets as the space of the secrets increases with d but the number of maps does not. Second, if we have noise on non-constant coefficients our maps change the noise distribution.

6.1 Hardness assumptions and justifications

In this subsection we investigate the hardness of the GBN, WIMN, and IMN problems. We first consider the GBN problem and relate it to the well-established LWE problem [60]. Then, we discuss the relation between the GBN problem and various approximate GCD problems [68]. Third, we discuss the special case $q = 2$ by relating the GBN problem to the well-known Max-3SAT problem, and more generally when $d = 1$ to Max-MQ, the problem of finding an assignment for polynomials $f_0, \ldots, f_{m-1} \in \mathbb{F}_q[x_0, \ldots, x_{n-1}]$ such that the majority of them evaluate to zero. Finally, we consider known attacks against the GBN problem. We start by recalling the LWE problem.

Definition 17 (The learning with errors (LWE) problem). *The* LWE *problem is defined though game* $\mathsf{LWE}_{n,q,\chi}$ *shown in Figure 8. The advantage of a* PPT *algorithm* $\mathcal{A}$ *in solving* LWE *is*

$$\mathbf{Adv}^{\mathsf{lwe}}_{n,q,\chi,\mathcal{A}}(\lambda) := \Pr\left[\mathsf{LWE}^{\mathcal{A}}_{n,q,\chi}(\lambda) \Rightarrow \mathsf{True}\right].$$

Initialize(1^λ):	**Sample**():	**Finalize**(s'):
begin	**begin**	**begin**
$n \leftarrow n(\lambda)$;	$a \leftarrow_\$ \mathbb{Z}_q^n$;	**return** $s = s'$;
$s \leftarrow_\$ \mathbb{Z}_q^n$;	$e \leftarrow_\$ \chi$;	**end**
return $(1^\lambda, n)$;	$b \leftarrow e + \sum_i a_i s_i$;	
end	**return** (a, b);	
	end	

Fig. 8. Game $\mathsf{LWE}_{n,q,\chi}$.

From the definition of LWE it is easy to see that GBN can be considered as a nonlinear generalisation of LWE if q is a prime. In other words, we have equivalence between these problems if we consider $b = d = 1$ in GBN. This is formalised in the next lemma.

Lemma 11 (LWE hard $\implies$ GBN hard for $b = d = 1$). *Let q be a prime. Then for any* PPT *adversary $\mathcal{A}$ against the* GBN *problem*[3] *with $b = d = 1$, there exists a* PPT *adversary $\mathcal{B}$ against the* LWE *problem such that*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),1,\ell,1,\chi,\mathcal{A}}(\lambda) = \mathbf{Adv}^{\mathsf{lwe}}_{n,q,\chi,\mathcal{B}}(\lambda).$$

Proof. We construct an adversary $\mathcal{B}$ against the LWE problem based on an adversary $\mathcal{A}$ against the GBN problem for $d = 1$ and $b = 1$. Algorithm $\mathcal{B}$ initialises $\mathcal{A}$ with P. Whenever $\mathcal{A}$ calls its **Sample** oracle, $\mathcal{B}$ queries its own **Sample** oracle to obtain (a, b) where $a = (a_0, \ldots, a_{n-1})$. It returns $\sum a_i x_i - b$ to $\mathcal{A}$. This is a valid GBN sample of degree $b = 1$. The **Challenge** oracle is answered similarly. When $\mathcal{A}$ calls its **Finalize** on G, since $d = 1$, we can assume w.l.o.g. that G is of the form $[x_0 - s_0, \ldots, x_{n-1} - s_{n-1}]$ with $s_i \in \mathbb{F}_q$. Algorithm $\mathcal{B}$ terminates by calling its **Finalize** oracle on $s = (s_0, \ldots, s_{n-1})$.

Adversary $\mathcal{B}$ is successful whenever $\mathcal{A}$ is. Indeed, from $\sum a_i x_i - b = 0$ it follows that $\sum a_i s_i = e$ and hence that s satisfies the LWE samples $(a, \sum a_i s_i + e)$. Finally, it is easy to see that $\mathcal{B}$ runs in polynomial time and uses polynomially many samples. □

This result can be generalised to any $b = d$ if we allow $\ell = n$ and consider the amortised variant of Regev's LWE scheme from [57] where each monomial m of $Q_{<b}$ corresponds to one parallel instance of Regev's original scheme. Note, however, that $\ell = n$ implies an exponentially large quotient.

Lemma 12 (LWE hard $\implies$ GBN hard for $b = d$, $\ell = n$, and $\mathsf{GBGen}(\cdot) = \mathsf{GBGen}_{\mathsf{dense}}(\cdot)$). *Let q be a prime, and assume χ outputs $e \leftarrow_{\$} \chi$, $e = \sum_{m \in Q} e_m \cdot m$, where the e_m are chosen independently, their distribution possibly depending on m. Then for any* PPT *adversary $\mathcal{A}$ against the* GBN *problem with $b = d$, $\exists$ PPT adversary $\mathcal{B}$ against the amortised* LWE *from [57] such that*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}_{\mathsf{dense}}(\cdot),d,n,b,\chi,\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathsf{lwe}}_{n,q,\chi,\mathcal{B}}(\lambda).$$

Proof (Sketch). Samples from GBN if $b = d$ are necessarily of the form $c = \sum t_i g_i + e$ where $t_i \in \mathbb{F}_q$. Write $c = \sum c_m m$ where the sum is over all monomials in c. The coefficients $\in Q$ of c are $c_{\tilde{m}} = \sum t_i g_{i,\tilde{m}} + e_{\tilde{m}}$ where $\tilde{m} \in Q$, $g_{i,\tilde{m}}$ is the coefficient for $\tilde{m}$ in g_i, and $e_{\tilde{m}}$ is the coefficient of $\tilde{m}$ in e. These are noisy random linear combinations of the secrets $g_{i,\tilde{m}}$ as in LWE. For $\tilde{m} \in Q_{<b}$, the $g_{i,\tilde{m}}$ are uniform. All monomials $\notin Q$ are of the form x_i^d and we have that the coefficient of $x_i^d = \mathrm{LM}(g_i)$ is t_i, exactly as in the amortised construction from [57]. □

Note that for coefficients $\tilde{m} \in Q_{=b}$, we can get LWE instances where some of the secret coefficients $g_{i,\tilde{m}}$ are always zero and so these instances will be easier, which is why we don't have equality between the advantages above.

Relation to the approximate GCD problem. The GBN problem for $n = 1$ is the approximate GCD problem over $\mathbb{F}_q[x]$. Contrary to the approximate GCD problem over the integers (cf. [68,27,25,26]), this problem has not yet received much attention (a variant of this problem is investigated in [26]), and hence it is unclear under which parameters it is hard. However, as mentioned in Section 2, the notion of a Gröbner basis can be extended to $\mathbb{Z}[x_0, \ldots, x_{n-1}]$, which

[3] Here $\mathcal{P}$ is a distribution which returns $P = \mathbb{F}_q[x_0, \ldots, x_{n-1}]$ with q as in the LWE game. Algorithm $\mathsf{GBGen}(\cdot)$ returns $[x_0 - s_0, \ldots, x_{n-1} - s_{n-1}]$ for some $s_i \in \mathbb{F}_q$, which is the only choice for $d = 1$.

in turn implies a version of the GBN problem over $\mathbb{Z}$. This can be seen as a direct generalisation of the approximate GCD problem in $\mathbb{Z}$.

THE $q = 2$ CASE. Recall that if $b = d = 1$ we have an equivalence with the LWE problem (or the well-known problem of learning parity with noise if $q = 2$). More generally, for $d = 1$ we can reduce Max-3SAT instances to GBN instances by translating each clause individually to a Boolean polynomial. However, in Max-3SAT the number of samples is bounded and hence this reduction only shows the hardness of GBN with a bounded number of samples. Still, the Gröbner basis returned by an arbitrary algorithm $\mathcal{A}$ solving GBN using a bounded number of samples will provide a solution to the Max-3SAT problem. Vice versa, we may convert a GBN instance for $d = 1$ to a Max-3SAT instance (more precisely a Partial Max-SAT instance where some clauses must be satisfied) by running an ANF-to-CNF conversion algorithm [7].

THE $d = 1$ CASE. When $d = 1$ the GBN problem is closely related to the Max-MQ problem. In [47] it was shown that if all f_i are square-free it is NP-hard to approximate this problem to within a factor of $q-\epsilon$ for ϵ a small positive number. Latter [71] proves that the minimal approximation ratio that can be achieved in polynomial time for Max-MQ is q. The most significant difference between GBN for $d = 1$ and Max-MQ is that the latter treats polynomials either as correct or incorrect, and no notion of "smallness" of noise exists. It follows from the properties of the Gaussian distribution that a Max-MQ oracle solves the GBN problem for $d = 1$.

KNOWN ATTACKS. Finally, we consider known attacks to understand the difficulty of the GBN problem. Recall, that if $b = d = 1$ Lemma 11 states that we can solve the LWE problem if we can solve the GBN problem. Conversely, for any $b \geq d$ and $d = 1$ the best known attack against the GBN problem is to reduce it to the LWE problem similarly to the linearisation technique used for solving nonlinear systems of equations in the noise-free setting. Let $N = \binom{n+b}{b}$ be the number of monomials up to degree b. Let $\mathcal{M} : P \longrightarrow \mathbb{F}_q^N$ be a mapping of polynomials in P to vectors in $\mathbb{F}_q^N$ by assigning to the ith component of the image vector the coefficient of the ith monomial $\in M_{\leq b}$. Then, in order to reduce GBN with n variables and degree b to LWE with N variables, reply to each LWE **Sample** query by calling GBN's **Sample** oracle to retrieve f, computing $v = \mathcal{M}(f)$ and returning (a, b) with $a := (v_{N-1}, \ldots, v_1)$ and $b := -v_0$. When the LWE adversary queries **Finalize** on s, query GBN's **Finalize** oracle with $[x_0 - s_0, \ldots, x_{n-1} - s_{n-1}]$. Correctness follows from the correctness of linearisation in the noise-free setting [5]. Furthermore, the LWE problem in N variables and with respect to the discrete Gaussian noise distribution $\chi_{\alpha,q}$ is considered to be hard if

$$\alpha \geq \frac{3}{2} \cdot \max\left(\frac{1}{q}, 2^{-2\sqrt{N \log q \log \delta}}\right)$$

for an appropriate choice of δ, which is the quality of the approximation for the shortest vector problem. With the current lattice algorithms $\delta = 1.01$ is hard, and $\delta = 1.005$ is infeasible [54].

Perhaps the most interesting attack on the LWE problem from the perspective of this work is that due to Arora and Ge [5]. This attack reduces the problem of solving linear systems with noise to the problem of solving (structured) nonlinear noise-free systems. We may apply this technique directly to GBN, i.e. without going through LWE first, and reduce it to GB with large b. However, it seems this approach does not improve the asymptotic complexity of the attack. Finally, certain conditions to rule out exhaustive search for the noise (and reduction to a noise-free system) must be imposed. We conclude this section by explicitly stating our hardness assumption.

Definition 18 (The GBN and WIMN assumptions). *Let $b, d \in \mathbb{N}$ with $b \geq d$. Let $\mathcal{P}$ be a polynomial ring distribution and $\chi_{\alpha,q}$ be the discrete Gaussian distribution. Suppose the parameters n, α, and q (all being a function of λ) satisfy the following set of conditions:*

1. *$n \geq \sqrt[b]{\lambda}$ to rule out linearisation attacks.*
2. *$\mathsf{GBGen}(\cdot)$ is instantiated with $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$, and q and $|S(\mathcal{I})_{\leq b}|$ are $\mathrm{poly}(\lambda)$ such that Lemma 9 applies.*
3. *$(\alpha q)^{nd^\ell} \approx 2^\lambda$ so exhaustive search over the noise or the secret key is ruled out.*
4. *For $N := \binom{n+b}{b}$, α and q are chosen such that the advantage of any PPT algorithm in solving the LWE problem with dimension N, modulus q and noise distribution $\chi_{\alpha,q}$ is negligible as a function of λ.*

Then the advantage of any PPT algorithm in solving the GBN or the WIMN problem is negligible as a function of λ.

7 Polly Cracker with Noise

We present a fully IND-CPA-secure Polly Cracker-style symmetric encryption scheme and prove it secure down to the hardness of the GBN problem. Our parameterised scheme, $\mathcal{SPCN}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}$, is shown in Figure 9. Here we represent elements in $\mathbb{F}_q$ as integers in the interval $[-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor]$. This convention is also used in the definition of noise. All the computations are performed in the ring P as generated by Gen. Furthermore, we assume that $\gcd(q, 2) = 1$. This condition is needed for the correctness and the security of our scheme. The message space is $\mathbb{F}_2$ (although we note that this can be generalised to other small fields).

Correctness of evaluation. For any choice of d, $\mathcal{SPCN}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}$ is additively homomorphic. However, to achieve multiplicative homomorphicity we need to set $d = 1$ as in Section 4. Hence, we restrict our attention to $d = 1$ and define the size of the noise as the logarithm of distance to zero over the integers. Addition and multiplication of the two ciphertexts $\mathsf{c}_0 = \sum h_{0,j} g_j + 2e_0 + \mathsf{m}_0$ and $\mathsf{c}_1 = \sum h_{1,j} g_j + 2e_1 + \mathsf{m}_1$ are given by

$$\begin{aligned}
\mathsf{c}_0 + \mathsf{c}_1 &= \sum h_{0,j} g_j + 2e_0 + \mathsf{m}_0 + \sum h_{1,j} g_j + 2e_1 + \mathsf{m}_1 \\
&= \sum (h_{0,j} + h_{1,j}) g_j + 2(e_0 + e_1) + (\mathsf{m}_0 + \mathsf{m}_1), \\
\mathsf{c}_0 \cdot \mathsf{c}_1 &= (\sum h_{0,j} g_j + 2e_0 + \mathsf{m}_0) \cdot (\sum h_{1,j} g_j + 2e_1 + \mathsf{m}_1) \\
&= (\sum h_{0,j} g_j) \cdot (\sum h_{1,j} g_j + 2e_1 + \mathsf{m}_1) \\
&\quad + (2e_0 + \mathsf{m}_0) \cdot (\sum h_{1,j} g_j) \\
&\quad + (4e_0 e_1 + 2e_0 \mathsf{m}_1 + 2e_1 \mathsf{m}_0 + \mathsf{m}_0 \mathsf{m}_1) \\
&= \sum \tilde{h}_j g_j + 2(2e_0 e_1 + e_0 \mathsf{m}_1 + e_1 \mathsf{m}_0) + \mathsf{m}_0 \mathsf{m}_1 \text{ for some } \tilde{h_j}.
\end{aligned}$$

The homomorphic features follow. Correctness of addition and multiplication for arbitrary numbers of operands follow from the associative laws of addition and multiplication in P up to overflows.

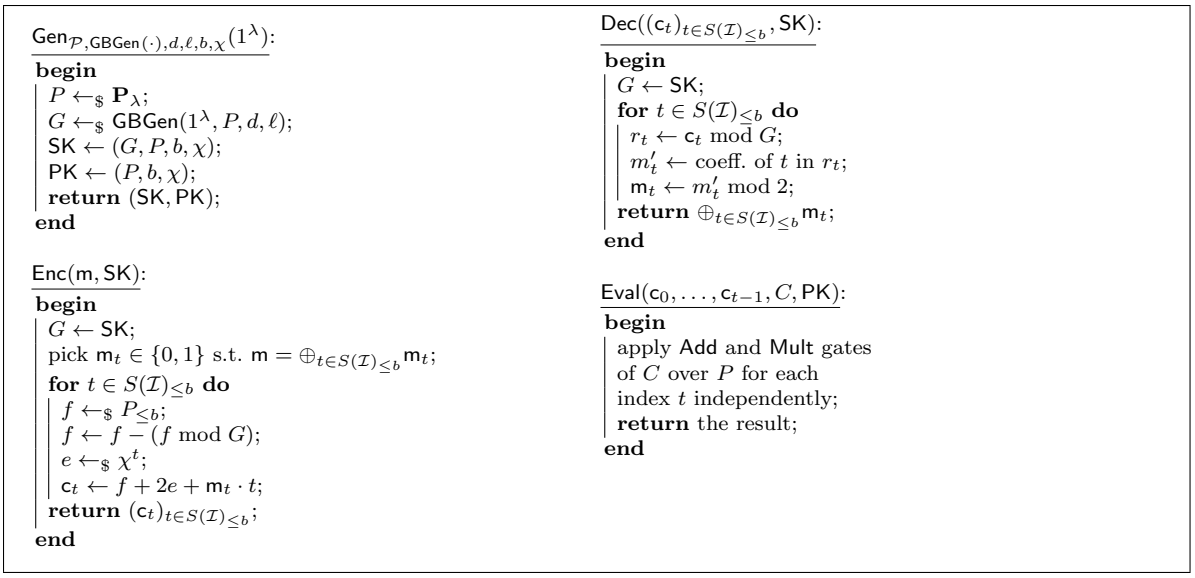

Fig. 9. The symmetric Polly Cracker with noise scheme, $\mathcal{SPCN}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi}$.

Permitted circuits. Circuits composed of Add and Mult gates can be seen as multivariate Boolean polynomials in t variables over $\mathbb{F}_2$. We can consider the generalisation of this set of polynomials to $\mathbb{F}_q$ (i.e., when the coefficients are in $\mathbb{F}_q$). In order to define the set of permitted circuits (which will be parametrised by $\alpha > 0$) we first embed the Boolean polynomials into the ring of polynomials over $\mathbb{Z}$. For $\chi_{\alpha,q}$, the probability of noise being larger than $k\alpha q$ is at most $\exp(-k^2/2)$. We say that a circuit is valid if for any $(s_0, \ldots, s_{t-1})$ with $s_i \leq t\alpha q$ the outputs are less than q for some parameter t. This restriction ensures that no overflows occur when polynomials are evaluated over $\mathbb{F}_q$. Section 9 discusses how to set α and q in order to allow for evaluation of polynomials of some fixed degree μ.

Compactness. Additions do not increase the size of the ciphertext, but they do increase the size of the error by at most one bit. Multiplications square the size of the ciphertext and increase the bit size of the noise by approximately $\log(8e_0e_1)$ bits The theorem below states the security properties of the above scheme.

Theorem 11. *Let $b \geq d$ be arbitrary and let $\mathcal{A}$ be a* PPT *adversary against the* IND-CPA *security of the scheme in Figure 9. Then there exists a* PPT *adversary $\mathcal{B}$ against the* WIMN *problem such that for all $\lambda \in \mathbb{N}$ we have*

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{SPCN},\mathcal{A}}(\lambda) = 2 \cdot \mathbf{Adv}^{\mathsf{wimn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,\ell,b,\chi,\mathcal{B}}(\lambda).$$

Proof. We construct an algorithm $\mathcal{B}$ against the WIMN problem for some arbitrary but fixed $t^\star \in S(\mathcal{I})_{\leq b}$ based on $\mathcal{A}$ attacking the IND-CPA security of the scheme. Roughly speaking, this algorithm runs $\mathcal{A}$ and answers its encryption queries using the provided sample oracle. Algorithm $\mathcal{B}$ answers $\mathcal{A}$'s left-or-right query by constructing the ciphertext components for $t \in S(\mathcal{I})_{\leq b}$ using its sample oracle when $t \neq t^\star$, and its challenge oracle when $t = t^\star$. See Algorithm 7 for the details.

Algorithm 7: $\mathsf{WIMN}_{t^\star}$ adversary $\mathcal{B}$ from IND-CPA adversary $\mathcal{A}$

```
1  begin
2      B receives (1^λ, P);
3      run A(1^λ, P) as follows;
4      if A queries IND-CPA.Encrypt(m) then
5          pick m_t ∈ {0,1} s.t. m = ⊕_{t∈S(I)_{≤b}} m_t;
6          for t ∈ S(I)_{≤b} do
7              query WIMN.Sample(t) to get f;
8              c_t ← 2f + m_t · t;
9          return (c_t)_{t∈S(I)_{≤b}};
10     if A queries IND-CPA.Left-Right(m_0, m_1) then
11         c ←$ {0,1};
12         pick m_t ∈ {0,1} s.t. m_c = ⊕_{t∈S(I)_{≤b}} m_t;
13         for t ∈ S(I)_{≤b} do
14             if t = t* then
15                 query WIMN.Challenge_{t*}() to get f;
16             else
17                 query WIMN.Sample(t) to get f;
18             c_t ← 2f + m_t · t;
19         return (c_t)_{t∈S(I)_{≤b}};
20     if A calls IND-CPA.Finalize(c') then
21         call Finalize(c = c');
```

Algorithm 7 is correct. If the samples returned by the $\mathbf{Challenge}_{t^\star}$ oracle to $\mathcal{B}$ are uniform in $P_{\leq b}$, then the probability that $c = c'$ is $1/2$. On the other hand, if the sample is a noisy element of the ideal, then adversary $\mathcal{A}$ is run in an environment which is identical to the IND-CPA game. Note that since $\gcd(q, 2) = 1$, multiplications by 2 at lines 9 and 19 do not affect the distribution of f (apart from doubling the noise, which is necessary to get the IND-CPA game environment). Hence in this case the probability that $c = c'$ is equal to the probability that $\mathcal{A}$ wins the IND-CPA game. The theorem follows. □

The above theorem, Lemma 9 and the recent results in [61] which establish the equivalence of symmetric and asymmetric homomorphic encryption schemes leads to the first provably secure public-key encryption scheme reducible to the hardness of computing Gröbner bases for random systems. This provides a positive answer to the challenges raised by Barkee et al. [11] (and later also by Gentry [40]). We note here that the transformation – as briefly described in Section 5 – only uses the additive features of the scheme and does not require full homomorphicity.

8 Trading Degrees for Noise

The product of two polynomials of degree b is a polynomial of degree $2b$, and hence the size of the ciphertext squares if two ciphertexts are multiplied together. In this section, we discuss how to reduce polynomials of degree b to polynomials of degree b' by performing *proxy re-encryption*. Proxy re-encryption allows to transform a ciphertext intended for a party A to a ciphertext for a party

B with the help of a (unidirectional) re-encryption key $K_{A\to B}$. Hence, after each multiplication we can apply this re-encryption for $K_{A\to A}$ to reduce the size of our ciphertexts at the cost of increasing the noise.

We discuss how one can achieve the above functionality for our scheme. Since the construction only uses additions, this feature also applies to the LWE-based encryption scheme as previously observed in [4]. Let $P = \mathbb{F}_q[x_0, \ldots, x_{n-1}]$ and suppose that $G_A = \{g_0, \ldots, g_{n-1}\}$ and $G_B = \{h_0, \ldots, h_{n-1}\}$ are two (possibly distinct) Gröbner bases for ideals $\mathcal{I}_A \subset P$ and $\mathcal{I}_B \subset P$. Finally, suppose $P/\mathcal{I}_A = P/\mathcal{I}_B$ as vector spaces (the equality always holds for $d = 1$). To re-encrypt a ciphertext intended for G_A under key G_B we first generate a re-encryption key $G_{A\to B}$ using Algorithm 8, and then use this key in Algorithm 9 – the re-encryption algorithm – to obtain a ciphertext under G_B.

The central idea behind these algorithms is the equivalence between different representations of elements in $P/\mathcal{I}$. While for the most part of this work we identify elements in $P/\mathcal{I}$ with elements $f \bmod \mathcal{I}$, Algorithms 8 and 9 make use of different representations of elements in $P/\mathcal{I}$. For example, if $x+1$ is an element of a Gröbner basis G_A, both $f = x$ and $r = -1$ represent the same element in $P/\mathcal{I}_A$ since $f \bmod G_A = r$, i.e. $x \bmod G_A = -1$. Hence, if we are interested in $P/\mathcal{I}_A$ (our messages live in $P/\mathcal{I}$) we can use f and r interchangeably. That is, for some $f = \sum c_i m_i$ with monomials m_i and coefficients $c_i \in \mathbb{F}_q$, we can compute the first decryption step, i.e. $\mathsf{m} + 2e = f \bmod \mathcal{I}_A$, as $\sum(c_i m_i \bmod \mathcal{I}_A)$. Furthermore, since $P/\mathcal{I}_A = P/\mathcal{I}_B$, we may encrypt the encoded message $\mathsf{m} + 2e$ for G_B by computing

$$f' = (f \bmod \mathcal{I}_A) + \tilde{f} = \sum(c_i m_i \bmod \mathcal{I}_A) + \tilde{f} = \mathsf{m} + 2e + \tilde{f} \text{ for } \tilde{f} \in \mathcal{I}_B.$$

Hence, we get that $f' \bmod \mathcal{I}_B = f \bmod \mathcal{I}_A$.

Algorithm 8: Generating a re-encryption key (Re-encryptionKey)

Input: G_A – a Gröbner basis
Input: $f'_0, \ldots, f'_{s-1}$ – polynomials of degree b' encrypting zero under a Gröbner basis G_B
Input: b – a bound on the degree of polynomials
Input: y – sparsity parameter

```
1 begin
2     G_{A→B} ← ∅;
3     for m ∈ M_{≤b} do
4         m' ← m mod G_A;
5         for 0 ≤ j < ⌈log_2(q/2)⌉ do
6             s ←$ a sparse subset of {0, ..., s − 1} of size y;
7             f'_{2^j · m} ← Σ_{i∈s} f'_i;
8             G_{A→B}[2^j · m] ← f'_{2^j · m} + 2^j · m';
9     return G_{A→B};
```

Now, using the key $G_{A\to B}$ we may re-encrypt a ciphertext f under G_A to a ciphertext f' under G_B using Algorithm 9.

Lemma 13. *Let G_A be a Gröbner basis. Let $f'_0, \ldots, f'_{s-1}$ be polynomials of degree b' encrypting zero under a Gröbner basis G_B. We set $G_{A\to B} = \mathsf{Re\text{-}encryptionKey}(G_A, [f_0, \ldots, f_{s-1}], b, y)$ for some*

[4] http://xagawa.net/pdf/20100120_SCIS_PRE.pdf

Algorithm 9: Re-encryption

Input: f – a polynomial in P of degree at most b
Input: $G_{A\to B}$ – a re-encryption key from key G_A to key G_B

```
begin
    f' ← 0;
    for monomials m appearing in f do
        c ← the coefficient in f of m, represented as an integer in (−⌊q/2⌋, ⌊q/2⌋];
        m' ← 0;
        for 0 ≤ j < ⌈log2(q/2)⌉ do
            if the jth bit of |c| is set then
                m' ← m' + G_{A→B}[2^j · m];
        if c < 0 then
            m' ← −1 · m';
        f' ← f' + m';
    return f';
```

$b > 0$ *and* $y > 0$. *Finally, let* f *be an encryption of* m *under key* G_A *with* $\deg(f) \leq b$. *It holds that* $f' = \mathsf{Re\text{-}encryptionKey}(f, G_{A\to B})$ *is a re-encryption of message* m *under* G_B *with* $\deg(f') \leq b'$. *Furthermore,* $\mathsf{Re\text{-}encryptionKey}(f, G_{A\to B})$ *adds a noise of bit size* $\log\log(q) + b\log(n+1) + \log(y) + |e'_{max}|$, *where* $|e'_{max}|$ *is the maximum bit size of the noise in any of the* f'_i*'s.*

Proof. Let f be an encryption of a message m under the key G_A and $G_{A\to B}$ be a re-encryption key generated using Algorithm 8. We want to show that $\mathsf{Re\text{-}encryption}(f, G_{A\to B})$ is an encryption of a message m under the key G_B. Let then m be a monomial of f, and c be the coefficient of m in f, represented as an integer in $(-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor]$. To simplify the notation, we set $t = \log_2(\lfloor \frac{q}{2} \rfloor)$. By definition, we can write $c \cdot m = s(c) \cdot \sum_{j=0}^{t-1} c_j \cdot 2^j \cdot m$, the c_j's being the binary decomposition of the absolute value of c, and $s(c) \in \{-1, +1\}$ the sign of c. It is clear that $\mathsf{Re\text{-}encryption}$ will transform each term $c \cdot m$ (c is a constant and m is a monomial) as follows:

$$\mathsf{Re\text{-}encryptionKey}(c \cdot m, G_{A\to B}) = s(c) \cdot \sum_{j=0}^{t-1} c_j \cdot G_{A\to B}[2^j \cdot m].$$

For any $0 \leq j < t$, by definition we have: $G_{A\to B}[2^j \cdot m] = f'_{2^j \cdot m} + 2^j \cdot (m \bmod G_A)$, where the $f'_{2^j \cdot m}$ are as in Algorithm 8. Since $P/\mathcal{I}_A = P/\mathcal{I}_B$, it holds that $(m \bmod G_A) \in P/\mathcal{I}_B$. As a consequence, each $G_{A\to B}[2^j \cdot m] \bmod G_B$ is a noisy encoding of $2^j(m \bmod G_A)$. More precisely,

$$\begin{aligned}
\mathsf{Re\text{-}encryptionKey}(c \cdot m, G_{A\to B}) \bmod G_B &= \left(s(c) \cdot \sum_{j=0}^{t-1} c_j \big(f'_{2^j \cdot m} + 2^j (m \bmod G_A) \big) \right) \bmod G_B \\
&= \left(c \cdot (m \bmod G_A) + s(c) \cdot \sum_{j=0}^{t-1} \big(c_j f'_{2^j \cdot m} \big) \right) \bmod G_B \\
&= c \cdot (m \bmod G_A) + s(c) \cdot \sum_{j=0}^{t-1} \big(2\, c_j e'_{2^j \cdot m} \big),
\end{aligned}$$

where $2\,e'_{2^j \cdot m}$ is the noisy part of $f'_{2^j \cdot m}$, namely $2\,e'_{2^j \cdot m} = f'_{2^j \cdot m} \bmod G_B$. Now, for any polynomial f, we denote by $\mathrm{T}(f)$ the *terms* of f. Recall that a term is a monomial times a constant. We have $\mathsf{Re\text{-}encryptionKey}(f, G_{A\to B}) = \sum_{c\cdot m \in \mathrm{T}(f)} \mathsf{Re\text{-}encryptionKey}(c \cdot m, G_{A\to B}) =$

$$\sum_{c\cdot m \in \mathrm{T}(f)} \left(s(c) \cdot \sum_{j=0}^{t-1} \left(c_j f'_{2^j \cdot m} \right) + c \cdot (m \bmod G_A) \right).$$

Hence, $\mathsf{Re\text{-}encryptionKey}(f, G_{A\to B}) \bmod G_B = \sum_{c\cdot m \in \mathrm{T}(f)} \mathsf{Re\text{-}encryptionKey}(c \cdot m, G_{A\to B}) \bmod G_B$

$$\begin{aligned}
&= \sum_{c\cdot m \in \mathrm{T}(f)} \left(s(c) \cdot \sum_{j=0}^{t-1} \left(c_j f'_{2^j \cdot m} \right) + c \cdot (m \bmod G_A) \right) \bmod G_B \\
&= \sum_{c\cdot m \in \mathrm{T}(f)} \left(s(c) \cdot \sum_{j=0}^{t-1} \left(2\, c_j e'_{2^j \cdot m} \right) + c \cdot (m \bmod G_A) \right) \\
&= \sum_{c\cdot m \in \mathrm{T}(f)} \left(s(c) \cdot \sum_{j=0}^{t-1} \left(2\, c_j e'_{2^j \cdot m} \right) \right) + \sum_{c\cdot m \in \mathrm{T}(f)} c \cdot (m \bmod G_A) \\
&= \sum_{c\cdot m \in \mathrm{T}(f)} \left(s(c) \cdot \sum_{j=0}^{t-1} \left(2\, c_j e'_{2^j \cdot m} \right) \right) + 2e + \mathsf{m} = 2e' + \mathsf{m},
\end{aligned}$$

where $e' = \sum_{c\cdot m \in \mathrm{T}(f)} \left(s(c) \cdot \sum_{j=0}^{t-1} \left(2\, c_j e'_{2^j \cdot m} \right) \right) + 2e \in P/\mathcal{I}_B$. Also, note that $\mathsf{m} + 2e = \sum_{c\cdot m \in \mathrm{T}(f)} c \cdot (m \bmod G_A) = f \bmod G_A$ holds because of the additive/multiplication-by-a-constant homomorphic features of the encryption scheme. All elements in $G_{A\to B}$ are of degree at most b'. Hence, the degree of the output of Algorithm 9 is at most b'.

Finally, if $|e'_{\max}|$ is the maximal bit size of noise in any of the f'_i used to generate $G_{A\to B}$ by Algorithm 8, then entries of $G_{A\to B}$ have maximal noise of bit size $\log(y) + |e'_{\max}|$. Now, given a polynomial of degree b, Algorithm 9 performs at most $\log(q) \binom{n+b}{b} \leq \log(q)\,(n+1)^b$ additions of polynomials with noise of size $\log(y) + |e'_{\max}|$. The bit size of the noise added in Algorithm 9 will be $(\log\log(q) + b\log(n+1) + \log(y) + |e'_{\max}|)$. Additionally, Algorithm 9 will "copy" the noise from f. □

For the security, we first discuss re-encryption under the same key, i.e. when $G_A = G_B$. If $b = b'$, the key $G_{A\to A}$ can be publicly constructed given access to encryptions of zero by requesting a fresh encryption of zero f and storing $G_{A\to A}[2^j \cdot m] = 2^j \cdot m + f$. Since $(f \bmod \mathcal{I}) = 2e$ for some small error term e, it holds that $f + 2^j \cdot m \bmod \mathcal{I} = (2^j \cdot m \bmod \mathcal{I}) + 2e$. Hence, $G_{A\to A}$ is a correct re-encryption key which can be generated given access only to encryptions of zero, and no additional information is leaked. This implies a limited form of key-dependent message security in the standard model: the least significant bits of the constant terms of the Gröbner basis elements are encrypted.

However, this argument does not go through for $b > b'$. While it is easy to construct elements f' which satisfy $f' \bmod \mathcal{I} \approx 2^j \cdot m \bmod \mathcal{I}$ for m a monomial of degree at least $b'+1$ and at most b with access to encryptions of zero, it is not easy to produce such an element f' with degree $\leq b'$ and

small noise. Yet, for $G_A \to G_B$ with $G_A \neq G_B$ security of re-encryption can be shown under the WIMN assumption. That is, any adversary breaking the IND-CPA security of the scheme with access to the re-encryption key $G_{A\to B}$ can be turned into an adversary breaking the IMN problem. A full proof of this for the special case of LWE is presented in [19], where this technique was independently proposed.

9 Parameters

Picking parameters for $\mathcal{SPCN}$ essentially reduces to fulfilling our correctness requirements and to ensuring that the linearised LWE instance is hard. We emphasise that a public-key version of $\mathcal{SPCN}$ is not competitive with Ring-LWE based schemes which have a public-key at least a factor of $N = \binom{n+b}{b}$ smaller. Furthermore, recent constructions apply additional techniques to manage parameter growth as the multiplication depth increases such as modulus switching, which we do not consider in this work. However, for completeness, we now give concrete suggestions for various parameters that are involved in our scheme. This section is to be understood as a sanity check showing how all parts fit together and not as a proposal of $\mathcal{SPCN}$ as a contender for an efficient somewhat homomorphic encryption scheme – it could not be such a contender as it attempts to provide a more unified view on such schemes.

We denote by μ the maximal degree of the Boolean polynomials corresponding to the circuits that we wish to support, and by λ the security parameter as before. One restriction on our choice of parameters is imposed by the requirement that decryption error probability on evaluated ciphertexts should be low. Since additions have a small effect on the noise, we concentrate on the degree of polynomials. This means that in order to allow for polynomials of degree up to μ and at most a, say, 2^{-20} decryption error probability, we must have $\Pr\left[|e^\mu| \geq q/2 : e \leftarrow_\$ \chi\right] < 2^{-20}$. Hence (cf. Section 7) we need to ensure that

$$\exp(k^2/2) > 2^{20} \quad \text{and} \quad k(\alpha q) < 1/2 \cdot \sqrt[\mu]{q}.$$

Another set of restrictions comes from the conditions stated in our intractability assumption in Definition 18, i.e. that the linearised LWE instance should be hard. For this, we make the somewhat arbitrary choice of $b = 2$ and denote by $N = \binom{n+b}{b} = \binom{n+2}{2}$ the number of monomials in a fresh ciphertext. We pick $d = 1$ because this case is best understood and allows for multiplicative homomorphicity. We set the parameters in a way which keeps q independent of b and allow for dependency on λ and μ only. (This is compatible with the definitional framework that we have set up.) We pick:

$$q \approx \lambda^{(2+\mu)} \quad \text{and} \quad \alpha = 1/(\lambda^\mu \log^2(\lambda)\sqrt{\lambda}).$$

Under these choices, we now have to establish the conditions under which the LWE instance characterised by N, q and $\chi_{\alpha,q}$ is hard. Since a detailed treatment of algorithms for solving LWE is out of scope for this work, we here restrict ourselves to a simple distinguishing attack for which we follow [51]. Denote by Λ the lattice spanned by our linearised polynomials and by Λ^T its dual lattice scaled by q. For the distinguishing attack we need a vector of length $1/\alpha$ in the dual lattice

Λ^T, i.e. we require a lattice reduction algorithm with root hermite factor δ satisfying:

$$\begin{aligned} 1/\alpha &= \delta^m \cdot \det(\Lambda^T) \\ 1/\alpha &= \delta^m \cdot q^{N/m} \\ \delta &= (1/\alpha \cdot q^{N/m})^{1/m}. \end{aligned}$$

We want to ensure that achieving this δ costs at least 2^λ operations. We set $m = \sqrt{n \cdot \log q / \log \delta}$ [54] and assume $\lambda \approx 1.8/\log \delta - 110$ [51] to be compatible with related work. This allows to estimate parameters for $\mathcal{SPCN}$. To turn it into a public-key scheme, we may use a standard symmetric-to-assymetric conversion. In particular, picking strategy (A) from Section 5, we require $2\,N \log_2 q$ encryptions of zero. Putting all this together, we arrive at the example choices of parameters given in Table 1 which was generated using the reference implementation from Section 10.

λ	μ	n	N	$\log_2 \alpha$	$\log_2 q$	$\log_2 \|\text{sk}\|$	$\log_2 \|\text{enc}\|$	$\log_2 \|\text{pk}\|$
40	1	11	78	-7.48	11.27	6.95	9.78	20.56
40	2	15	136	-12.81	16.94	7.99	11.17	23.34
40	3	18	190	-18.13	22.16	8.64	12.04	25.08
40	4	21	253	-23.45	27.19	9.16	12.75	26.50
40	5	23	300	-28.77	32.64	9.55	13.26	27.52
80	1	18	190	-8.48	12.96	7.87	11.27	23.53
80	2	18	190	-14.80	19.60	8.46	11.86	24.73
80	3	22	276	-21.13	25.97	9.16	12.81	26.61
80	4	25	351	-27.45	32.38	9.66	13.47	27.94
80	5	29	465	-33.77	38.32	10.12	14.12	29.24
128	1	26	378	-9.11	14.04	8.51	12.37	25.75
128	2	25	351	-16.11	21.28	9.06	12.87	26.73
128	3	25	351	-23.11	28.61	9.48	13.29	27.59
128	4	29	465	-30.11	35.77	10.02	14.02	29.04
128	5	33	595	-37.11	42.62	10.46	14.63	30.26

Table 1. Example parameter choices for $b = 2$, $k = \sqrt{2\log(2^{20})}$ and $N = \binom{n+2}{2}$. The column $\log_2 \|\text{sk}\|$ expresses the logarithm of the bitsize of the secret key, the column $\log_2 \|\text{enc}\|$ the logarithm of the bitsie of a fresh ciphertext, the column $\log_2 \|\text{pk}\|$ the logarithm of the bitsize of the public key. For $\mathcal{SPCN}$ the last column is ignored.

From Table 1 it is easy to see that $\mathcal{SPCN}$ and its public-key variant only supports very limited levels of multiplicative homomorphicity. This is to be expected, as the public key has size $\mathcal{O}\left((N \log q)^2\right)$ with $q \approx \lambda^{2+\mu}$, i.e. $\mathcal{O}\left((N\mu)^2\right)$.

10 Reference Implementation

We implemented our scheme using the Sage mathematics software [66].[5] Although this implementation is not efficient, the code not only concretely demonstrates the correctness of the scheme, it also shows that if basic mathematical structures are available, it can be easily implemented.

[5] See `https://bitbucket.org/malb/research-snippets/src/tip/noisy-polly-cracker.py` .

Acknowledgements

We would like to thank Carlos Cid for valuable feedback and discussions on this work. We would also like to thank Frederik Armknecht for helpful discussions on an earlier draft of this work. The work described in this paper has been supported by the Royal Society grant JP090728 and by the Commission of the European Communities through the ICT program under contract ICT-2007-216676 (ECRYPT-II).

Martin R. Albrecht, Jean-Charles Faugère, and Ludovic Perret are also supported by the French ANR under the Computer Algebra and Cryptography (CAC) project (ANR-09-JCJCJ-0064-01) and the EXACTA project (ANR-09-BLAN-0371-01).

References

1. Martin Albrecht and John Perry. F4/5. *CoRR*, abs/1006.4933v2, 2010.
2. Martin R. Albrecht, Pooya Farshim, Jean-Charles Faugère, and Ludovic Perret. Polly Cracker, revisited. In *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 179–196, Berlin, Heidelberg, New York, 2011. Springer Verlag.
3. Martin R. Albrecht, Jean-Charles Faugère, Dongdai Lin, and Ludovic Perret. Polynomials With Errors (PWE). in preparation, 2012.
4. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptography - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618, Berlin, Heidelberg, New York, 2009. Springer Verlag.
5. Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.
6. Gwenole Ars. *Applications des bases de Gröbner à la cryptographie*. PhD thesis, Université de Rennes I, 2005.
7. Gregory V. Bard, Nicolas T. Courtois, and Chris Jefferson. Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over GF(2) via SAT-solvers. Cryptology ePrint Archive, Report 2007/024, 2007. Available at `http://eprint.iacr.org/2007/024`.
8. Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Paris VI, 2004.
9. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *International Conference on Polynomial System Solving - ICPSS*, pages 71 –75, Nov 2004.
10. Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Bo-Yin Yang. Asymptotic expansion of the degree of regularity for semi-regular systems of equations. In P. Gianni, editor, *The Effective Methods in Algebraic Geometry Conference, Mega 2005*, pages 1 –14, May 2005.
11. Boo Barkee, Deh Cac Can, Julia Ecks, Theo Moriarty, and R. F. Ree. Why you cannot even hope to use Gröbner bases in Public Key Cryptography: An open letter to a scientist who failed and a challenge to those who have not yet failed. *Journal of Symbolic Computations*, 18(6):497–501, 1994.
12. Dave Bayer and Mike Stillman. On the complexity of computing syzygies. *Computational Aspects of Commutative Algebra*, page 1–13, 1988.
13. Thomas Becker and Volker Weispfenning. *Gröbner Bases - A Computational Approach to Commutative Algebra*. Springer Verlag, Berlin, Heidelberg, New York, 1991.
14. Mihir Bellare, Thomas Ristenpart, and Stefano Tessaro. Multi-instance security and its application to password-based cryptography. In Safavi-Naini and Canetti [63], pages 312–329.
15. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Advances in Cryptology – EUROCRYPT 2004*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, Berlin, Heidelberg, New York, 2006. Springer Verlag.
16. Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A multivariate stream cipher with provable security. *J. Symb. Comput.*, 44(12):1703–1723, 2009.

17. Zvika Brakerski. When homomorphism becomes a liability. Cryptology ePrint Archive, Report 2012/225, 2012. http://eprint.iacr.org/.
18. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS*, pages 309–325. ACM, 2012.
19. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 97–106. IEEE, 2011.
20. Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. PhD thesis, Universität Innsbruck, 1965.
21. Bruno Buchberger. Gröbner bases: an algorithmic method in polynomial ideal theory. In N. K. Bose, editor, *Multidimensional Systems Theory*. D. Reidel Publishing Company, 1985.
22. Bruno Buchberger. An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of Symbolic Computation*, 41(3-4):475–511, 2006.
23. Stanislav Bulygin. Chosen-ciphertext attack on noncommutative Polly Cracker. *CoRR*, abs/cs/0508015, 2005.
24. Massimo Caboara, Fabrizio Caruso, and Carlo Traverso. Lattice Polly Cracker cryptosystems. *Journal of Symbolic Computation*, 46:534–549, May 2011.
25. Yuanmi Chen and Phong Q. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In Pointcheval and Johansson [58], pages 502–519.
26. Henry Cohn and Nadia Heninger. Approximate common divisors via lattices. Cryptology ePrint Archive, Report 2011/437, 2011. http://eprint.iacr.org/.
27. Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer, 2011.
28. Jean-Sebastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2011/440, 2011. http://eprint.iacr.org/.
29. Nicolas T. Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287, Berlin, Heidelberg, New York, 2002. Springer Verlag.
30. David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, Berlin, Heidelberg, New York, 3rd edition, 2005.
31. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal of Computing*, pages 167–226, 2003.
32. Alicia Dickenstein, Noaï Fitchas, Marc Giusti, and Carmen Sessa. The membership problem for unmixed polynomial ideals is solvable in single exponential time. *Discrete Appl. Math.*, 33(1-3):73–94, 1991.
33. Françoise Levy dit Vehel, Maria Grazia Marinari, Ludovic Perret, and Carlo Traverso. A survey on Polly Cracker systems. In Massimiliano Sala, Teo Mora, Ludovic Perret, Shojiro Sakata, and Carlo Traverso, editors, *Gröbner Bases. Coding and Cryptography*, pages 285–305. Springer Verlag, Berlin, Heidelberg, New York, 2009.
34. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner basis (F4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.
35. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 75–83, New York, 2002. ACM.
36. Jean-Charles Faugère, Patrizia M. Gianni, Daniel Lazard, and Teo Mora. Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering. In *Journal of Symbolic Computation 16*, pages 329–344. Academic Press, 1993.
37. Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 2003. Springer Verlag.
38. Jean-Charles Faugère and Sajja Rahmany. Solving systems of polynomial equations with symmetries using SAGBI-Gröbner bases. In *ISSAC '09: Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, ISSAC '09, pages 151–158, New York, NY, USA, 2009. ACM.
39. Mike Fellows and Neal Koblitz. Combinatorial cryptosystems galore! In G. L. Mullen and P. J.-S. Shiue, editors, *Finite Fields: Theory, Applications, and Algorithms*, volume 168 of *Contemporary Mathematics*, pages 51–61. AMS, 1994.
40. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. Available at http://crypto.stanford.edu/craig.

41. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.
42. Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In Rafail Ostrovsky, editor, *FOCS*, pages 107–109. IEEE, 2011.
43. Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In Kenneth Paterson, editor, *Advances in Cryptology — EUROCRYPT 2010*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148, Berlin, Heidelberg, New York, 2011. Springer Verlag.
44. Craig Gentry, Shai Halevi, and Nigel P. Smart. Better bootstrapping in fully homomorphic encryption. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography – PKC 2012*, volume 7293 of *Lecture Notes in Computer Science*. Springer Verlag, 2012.
45. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Safavi-Naini and Canetti [63], pages 850–867.
46. Henri Gilbert, editor. *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*. Springer, 2010.
47. Johan Håstad, Steven Phillips, and Shmuel Safra. A well-characterized approximation problem. *Inf. Process. Lett.*, 47:301–305, October 1993.
48. Gottfried Herold. Polly cracker, revisited, revisited. In *Public Key Cryptography – PKC 2012*, volume 7293 of *Lecture Notes in Computer Science*, pages 17–33, Berlin, Heidelberg, New York, 2012. Springer Verlag.
49. Neal Koblitz, Alfred J. Menezes, Yi-Hong Wu, and Robert J. Zuccherato. *Algebraic aspects of cryptography*. Springer Verlag, Berlin, Heidelberg, New York, 1998.
50. Daniel Lazard. Gröbner-bases, Gaussian elimination and resolution of systems of algebraic equations. In *Proceedings of the European Computer Algebra Conference on Computer Algebra*, volume 162 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 1983. Springer Verlag.
51. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.
52. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology — EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*. Springer Verlag, 2010.
53. Carlos Aguilar Melchor, Philippe Gaborit, and Javier Herranz. Additively homomorphic encryption with d-operand multiplications. In *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 138–154, Berlin, Heidelberg, New York, 2010. Springer Verlag.
54. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer Verlag, Berlin, Heidelberg, New York, 2009.
55. Ferdinando Mora. De Nugis Groebnerialium 2: Applying Macaulay's trick in order to easily write a Gröbner basis. *Applicable Algebra in Engineering, Communication and Computing*, 13(6):437–446, 2003.
56. Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In Christian Cachin and Thomas Ristenpart, editors, *CCSW*, pages 113–124. ACM, 2011.
57. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Advances in Cryptology – CRYPTO 2008*, pages 554–571, Berlin, Heidelberg, 2008. Springer Verlag.
58. David Pointcheval and Thomas Johansson, editors. *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*. Springer, 2012.
59. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56:34:1–34:40, September 2009.
60. Oded Regev. The learning with errors problem. In *IEEE Conference on Computational Complexity 2010*, pages 191–204, 2010.
61. Ron Rothblum. Homomorphic encryption: From private-key to public-key. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 219–234. Springer, 2011.
62. Markus Rückert and Michael Schneider. Estimating the security of lattice-based cryptosystems. Cryptology ePrint Archive, Report 2010/137, 2010. Available at `http://eprint.iacr.org/2010/137`.
63. Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012.
64. M. Sala, T. Mora, L. Perret, S. Sakata, and C. Traverso. *Gröbner Bases, Coding, and Cryptography*. Springer, Berlin, Heidelberg, New York, 2009.

65. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.
66. William Stein et al. *SAGE Mathematics Software*. The Sage Development Team (Version 4.7.0), 2011. Available at `http://www.sagemath.org`.
67. A. J. Stothers. *On the Complexity of Matrix Multiplication*. PhD thesis, University of Edinburgh, 2010.
68. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Gilbert [46], pages 24–43.
69. Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra (2. ed.)*. Cambridge University Press, 2003.
70. Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In Howard J. Karloff and Toniann Pitassi, editors, *STOC*, pages 887–898. ACM, 2012.
71. Shang-Wei Zhao and Xiao-Shan Gao. Minimal achievable approximation ratio for MAX-MQ in finite fields. *Theor. Comput. Sci.*, 410(21-23):2285–2290, 2009.