Polly Cracker, Revisited[⋆]

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

[1] INRIA, Paris-Rocquencourt Center, SALSA Project
UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France
[2] Information Security Group, Royal Holloway, University of London, UK
malb@lip6.fr, pooya.farshim@rhul.ac.uk, jean-charles.faugere@inria.fr,
ludovic.perret@lip6.fr

Abstract. In this work, we initiate the formal treatment of cryptographic constructions ("Polly Cracker") based on the hardness of computing remainders modulo an ideal. We start by formalising and studying the relation between the ideal remainder 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 IR 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 related problems. These problems can be seen as natural generalisations of the LWE problem and the approximate GCD problem 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 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 that by also providing a new family of somewhat homomorphic encryption schemes based on new, but natural, hard problems. Our results also imply that Regev's LWE-based public-key encryption scheme is (somewhat) *multiplicatively* homomorphic for appropriate choices of parameters.

Keywords. Polly Cracker, Gröbner bases, Learning with errors, Homomorphic encryption, Provable security.

1 Introduction

BACKGROUND. Homomorphic encryption [38] is a cryptographic primitive which allows to perform arbitrary computation 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 to 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, an instantiation of homomorphic encryption over the ring of

[⋆] An extended abstract of this work will appear in ASIACRYPT 2011.

multivariate polynomials itself is perhaps the most natural strategy, although not conceptually the simplest (cf. [57]).

Indeed, let $\mathscr{I} \subset P = \mathbb{F}[x_0, \dots, x_{n-1}]$ be some ideal and denote an injective function by $\mathbf{Encode}()$, with inverse $\mathbf{Decode}()$, that maps bit-strings to elements in the quotient ring $P/\mathscr{I}$. 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

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

Decryption is performed by computing remainders modulo $\mathscr{I}$. From the definition of an ideal the homomorphic features of this scheme follow. The problem of computing remainders modulo an ideal was solved by Buchberger in [19–21], where he introduced the notion of Gröbner bases, and gave an algorithm for computing such bases.

In fact, all known homomorphic schemes which support both addition and multiplication are based on variants of the ideal remainder problem over various rings. For example in [57] the ring $\langle p \rangle \in \mathbb{Z}$ for p an odd integer is considered. In [38] ideals in a number field play the same role (cf. [55]). One can even view Regev's LWE-based public-key encryption scheme [51] in this framework. Furthermore, if we instantiate the construction in [48] 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 [36, 10, 44], 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 [30]. 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 challenge to propose a secure Polly Cracker-style encryption scheme [10] (cf. also [37, p. 41]).

CONTRIBUTIONS & ORGANISATION. Our contributions in this paper can be summarised as follows: 1) we initiate the formal treatment of Polly Cracker-style schemes over multivariate polynomial rings and characterise their security; 2) we show 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; 4) we present a new somewhat (and doubly) homomorphic encryption scheme based on a new class of computationally hard problems.

More precisely, we start by settling notation in Section 2 and Section 3. In Section 4, we formalise various problems associated with ideals in polynomials rings in the language of game-based security definitions. In particular, we show that computing remainders modulo an ideal with overwhelming probability is equivalent to computing a Gröbner basis for zero-dimensional ideals. We then show that deciding ideal membership and computing remainders modulo an ideal are equivalent for certain choices of parameters. This allows us to introduce a symmetric variant of Polly Cracker 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 bounded by an a priori 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 5). Bounded IND-CPA security is the best level of security that this scheme can possibly achieve: we give an attacker breaking the cryptosystem once enough ciphertexts are obtained.

In Section 6, 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 [53, 57].

To go beyond this limitation, we consider a constructions where **Encode**$()$, as introduced in the beginning of this section, is randomised (and hence **Decode**$()$ is no longer injective). To prove security for such schemes, we consider noisy variants of the ideal membership and related problems. These can be seen as natural generalisations of the (decisional) LWE and the approximate GCD problems over polynomial rings (Section 7). After formalising and justifying the hardness of the noisy assumptions in Section 8, we show that noisy encoding of messages can indeed be used to construct a fully IND-CPA secure somewhat homomorphic scheme. This result also implies that Regev's LWE-based public-key scheme is *multiplicatively* homomorphic under appropriate choices of parameters. 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. [10], which asks for a public-key Polly Cracker-style encryption scheme whose security is based on the hardness of computing Gröbner bases for random systems of polynomials. In addition, we provide a new family of somewhat homomorphic schemes which are based on new natural variants of well-studied hard problems. In Section 9 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 [2] to achieve full KDM security for the Polly Cracker with noise scheme. We discuss concrete parameter choices in Section 10 and our reference implementation in Section 11.

1.1 Related Work

Polly Cracker. In 1993, Barkee et al. wrote a paper [10] 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 is a set of polynomials $\{f_0, \ldots, f_{m-1}\} \subset \mathscr{I}$ which is used to construct an element $f \in \mathscr{I}$. Encryption of messages $m \in P/\mathscr{I}$ are computed as $c = \sum h_i f_i + m = f + m$ for $f \in \mathscr{I}$. The private key is a Gröbner basis G which allows to compute $m = c \mod \mathscr{I} = c \mod G$. As highlighted in [10] this scheme can be broken using results from [28] (cf. Theorem 6).

At about the same time, and independently from the work of Barkee et al., Fellows and Koblitz [36, 44] proposed a framework for the design of public-key cryptosystems. The ideas in [36] were similar to Barkee et al.'s cryptosystem, 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 6, yet almost all of these schemes were broken. We point the reader to [30] for a good survey of various constructions and attacks. Currently, the only Polly Cracker-style scheme which is not broken is the scheme in [23]. 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 (multivariate) polynomial ideals with respect to random systems.

Homomorphic Encryption. In the last decades several different approaches to construct singly homomorphic schemes – with respect to 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 [38], the approximate GCD problem over the Integers of van Dijk et al. [57], the Polynomial Coset Problem as proposed by Smart and Vercauteren in [55], 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. [48] and recently the Learning with Errors Problem of Brakerski and Vaikuntanathan [18]. 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 [40] have been able to implement all aspects of Gentry's scheme [38], including the bootstrapping step. In this work the authors also improve on the work of Smart and Vercauteren [55]. However, the bootstrapping step still renders somewhat homomorphic schemes impractical (cf. [45]). Hence, recent constructions aim to avoid it [17, 39].

Recently and independently of this work, in [18] a construction based on LWE was proposed, denoted SH in [18], which can be seen as a linear variant of our noisy Polly Cracker scheme. Furthermore, the technique we propose in Section 9 is also independently proposed in [18]. However, in contrast to our work [18] has an explicit non-algebraic perspective. Also a second scheme in [18], denoted BTS, achieves *fully* homomorphic encryption based on a "dimension-modulus reduction" technique – while our work only achieves somewhat homomorphic encryption. We note that this technique also applies to some of our constructions. Finally, we note that improvements such as

[24] also immediatly apply to our constructions which generalise those constructions considered in [24].

The main difference between our work 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. Furthermore, due to the versatility of Gröbner basis theory, our work can be seen as a generalisation of a number of known schemes and their underlying hardness assumptions.

$\mathscr{MQ}$ Cryptography. Our work can also be seen in connection with public-key cryptosystems based on the hardness of solving multivariate quadratic equations ($\mathscr{MQ}$). The difference is that our cryptographic constructions enjoy strong reductions to the known and hard problem of solving a *random* system of equations, whereas the bulk of work in $\mathscr{MQ}$ cryptography relies on heuristic security arguments [58, 49, 16, 29]. In contrast, our work is more in the direction of research initiated by Berbain et al. [14, 3] who proposed a stream cipher whose security was reduced to the difficulty of solving a system of random multivariate quadratic equations over $\mathbb{F}_2$. Note also that the concept of adding noise to a system of multivariate equations has been also proposed by Gouget and Patarin in [41] for the design of an authentication scheme. Our work, however, presents a more general and complete 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.

2 Preliminaries

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 non-zero probability. We use ppt for probabilistic polynomial-time. We call a function $\eta(\lambda)$ negligible if $|\eta(\lambda)| \in \lambda^{-\omega(1)}$. 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 $\mathsf{MsgSp}(\mathsf{P})$.

GAMES-BASED SECURITY DEFINITIONS AND PROOFS. In this paper we use the code-based game-playing language [13]. Each game has some **Initialize** and a **Finalize** procedure. It also has specifications of procedures to respond to the adversary's various oracle queries. A game Game is run with an adversary $\mathscr{A}$ as follows. First **Initialize** runs and its outputs are passed to $\mathscr{A}$. Then $\mathscr{A}$ runs and its oracle queries are answered by the procedures of Game. When $\mathscr{A}$ terminates, its output is passed to **Finalize** which returns the outcome of the game y. This interaction is written as $\mathsf{Game}^{\mathscr{A}} \Rightarrow y$. In each game, we restrict our attention to legitimate adversaries, which is defined specifically for each game.

3 Basics of Gröbner Bases

In this section we recall some basic definitions related to Gröbner bases [21, 19, 20]. For a more detailed treatment we refer to, for instance, [26].

We consider a polynomial ring $P = \mathbb{F}[x_0,\dots,x_{n-1}]$ over some finite field (typically $\mathbb{F}_q$), some monomial ordering on elements of P, and a set of polynomials $f_0,\dots,f_{m-1}$. We denote by $\mathrm{M}(f)$ the set of all monomials appearing in $f \in P$. 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 $\in \mathbb{F}$ corresponding to $\mathrm{LM}(f)$ 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 underling 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 $f_0,\dots,f_{m-1}$ be polynomials in P. We define the set*

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

as the ideal generated *by $f_0,\dots,f_{m-1}$.*

It is known that every $\mathscr{I}$ ideal of P is finitely generated, i.e., there exists a finite number of polynomials $f_0,\dots,f_{m-1}$ in P such that $\mathscr{I} = \langle f_0,\dots,f_{m-1}\rangle$. Roughly speaking, a Gröbner basis is a particular generator set of an ideal.

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

$$\mathrm{LM}(g_i) \mid \mathrm{LM}(f).$$

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

It is possible to extend the division algorithm to multivariate polynomials: we write $r = f \mod G$ when $f = \sum_{i=0}^{m-1} h_i g_i + r$ 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 $\mathscr{I}$. In particular we have that $f \mod \mathscr{I} = f \mod G = 0$ if and only if $f \in \mathscr{I}$. Together P and $\mathscr{I}$ define the quotient ring $P/\mathscr{I}$ and, by abuse of notation, we write $f \in P/\mathscr{I}$ if $f \mod \mathscr{I} = f$ where equality is interpreted as those on elements of P. That is, we identify elements of the quotient $P/\mathscr{I}$ 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 non-zero constant. However, from any Gröbner basis we can compute the unique reduced Gröbner basis in polynomial time. The algorithm performing this transformation is denoted $\mathsf{ReduceGB}(\cdot)$ in this work and is given in Algorithm 1.

Algorithm 1: ReduceGB$(\cdot)$

```
Input: $Q$ – a set of polynomials forming a Gröbner basis
Result: the reduced Gröbner basis for $Q$
begin
    $\tilde{Q} \leftarrow \varnothing$;
    while $Q \neq \varnothing$ do
        $f \leftarrow$ the smallest element of $Q$ according to the term ordering;
        $Q \leftarrow Q \setminus \{f\}$;
        if $\mathrm{LM}(f) \notin \langle \mathrm{LM}(\tilde{Q}) \rangle$ then
            $\tilde{Q} \leftarrow \tilde{Q} \cup \{\mathrm{LC}(f)^{-1} \cdot f\}$;
    return $\left[h \mod \tilde{Q} \setminus \{h\} \mid h \in \tilde{Q}\right]$;
end
```

Definition 3 (Reduced Gröbner Basis). *A* reduced Gröbner basis *for an ideal* $\mathscr{I} \subset P$ *is a Gröbner basis G such that:*

1. $\mathrm{LC}(g) = 1$*, for all* $g \in G$*;*
2. $\forall g \in G, \nexists\, m \in \mathrm{M}(g)$ *such that m is divisible by some element of* $\mathrm{LM}(G \setminus \{g\})$.

Buchberger [19] proved that in order to compute a Gröbner basis from a given ideal basis, it is sufficient to consider 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 non-zero 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*
$$S(f,g) = \frac{x^\gamma}{\mathrm{LT}(f)} \cdot f - \frac{x^\gamma}{\mathrm{LT}(g)} \cdot g.$$

In particular, 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 in the form*
$$f = a_0 g_0 + \cdots + a_{s-1} g_{s-1},$$
such that whenever $a_i g_i \neq 0$*, we have* $a_i g_i \leq t$*. Furthermore, we write that* $f \xrightarrow[G]{} 0$ *("f reduces to zero") if and only if f has an* $\mathrm{LM}(f)$*-representation with respect to G.*

Note that $f \mod G = 0$ implies that $f \xrightarrow[G]{} 0$ while the converse is false.

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

Proof. See [12, p.211ff]. □

From Theorem 1 an algorithm follows [19] 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 useless. 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:

Theorem 2 (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., they have disjoint leading terms. Then $S(f,g) \xrightarrow[\{f,g\}]{} 0$.*

Proof. See [12, p.222ff]. □

From this, we get:

Corollary 1. *A set $\{g_0, \dots, g_{n-1}\} \subset P$ 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.

Lemma 1 (Finiteness Criterion). *Let $\mathscr{I} = \langle f_0, \dots, f_{m-1} \rangle \subset P$ with $P = \mathbb{F}[x_0, \dots, x_{n-1}]$ 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 $i = 0, \dots, n-1$, we have $\mathscr{I} \cap \mathbb{F}[x_i] \neq \varnothing$.*
3. *For all $i, 0 \leq i < n$, there exists $g_i \in \mathscr{I}$ such that $\mathrm{LM}(g_i) = x_i^{d_i}$ with $d_i > 0$.*
4. *The set of monomials $S(\mathscr{I}) = \mathrm{M}(P) \setminus \{\mathrm{LM}(f) \mid f \in \mathscr{I}\}$ is finite.*
5. *The $\mathbb{F}$-vector space $P/\mathscr{I}$ is finite-dimensional and a basis is given by $S(\mathscr{I})$.*

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

Proof. See [26, p.234ff]. □

In this work we use reduction modulo an ideal to sample polynomials from some ideal. The following lemma will be helpful to assert that this sampling is uniform.

Lemma 2. *Let $\mathscr{I} \subset P = \mathbb{F}_q[x_0, \dots, x_{n-1}]$ be some ideal. Any element $f \in P$ with $\deg(f) = b$ has a unique representation $f = \tilde{f} + r$ with $\tilde{f} \in \mathscr{I}$ and $r \in P/\mathscr{I}$ where $\deg(\tilde{f}) \leq b$ and $\deg(r) \leq b$. In particular, if M is the set of monomials $\in P/\mathscr{I}$ with degree $\leq b$, then for any $\tilde{f} \in \mathscr{I}$ there are $q^{|M|}$ elements f_i in P with $f = f_i - (f_i \mod \mathscr{I})$.*

Proof. The monomials in $P_{\leq b}$ span a $\binom{n+b}{b}$-dimensional vector space V over $\mathbb{F}_q$. The monomials $\in P/\mathscr{I}$ up to degree b span a subspace of V with dimension $|M|$, from which the claim follows. □

4 Gröbner Basis and Ideal Membership Problems

In this section we formalise various problems associated with Gröbner bases. Following [27], 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 $\mathscr{P}$ 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 ring operations, sampling 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 prime field of size $q(\lambda)$.

REMARK. There is a one-to-one correspondence of ideals over polynomial rings over finite extension fields $I \subset \mathbb{F}_{q^n}[x_0,\dots,x_{n-1}]$ and ideal over polynomial rings over prime fields $J \subset \mathbb{F}_q[x_0,\dots,x_{n-1},\alpha]$ by mapping a root of $\mathbb{F}_{q^n}$ to α and adding the characteristic polynomial of $\mathbb{F}_{q^n}$ to the generating basis, hence finite extension fields are covered by this definition. The ring $\mathbb{Z}[x_0,\dots,x_{n-1}]$ is not covered by our definition, but it can easily be generalised.

Once $\mathscr{P}$ 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 $\mathsf{GBGen}(1^\lambda, P, d)$ any ppt algorithm which outputs a reduced Gröbner basis G for some zero-dimensional ideal $\mathscr{I} \subset P$ such that every element of G is of degree at most d. Of particular interest to this paper is the Gröbner basis generation algorithms 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. Note that $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$ for $d = 1$ captures the usual case of a set of polynomials which have a (unique) common root in the base field, and where $\mathrm{LM}(g_i) = x_i$

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

```
begin
    if $d = 0$ then return $\{0\}$;
    for $0 \le i < n$ do
        $g_i \leftarrow x_i^d$;
        for $m_j \in M_{<x_i^d}$ do
            $c_{ij} \leftarrow_\$ \mathbb{F}_q$;
            $g_i \leftarrow g_i + c_{ij} m_j$;
    return ReduceGB($\{g_0,\dots,g_{n-1}\}$);
end
```

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

for all $i, 0 \leq i < n$. This case is common in cryptographic applications such as algebraic cryptanalysis [34, 25] and a well-studied case. The next lemma – which is an easy consequence of Corollary 1 – establishes that $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$ returns a Gröbner basis.

Lemma 3. *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*

$$g_i = x_i^d + \sum c_{ij} m_j, \text{ for all } i, 0 \leq i < n, \text{ with } m_j \in M_{<x_i^d} \text{ and } c_{ij} \in \mathbb{F}.$$

Then G is a Gröbner basis for the zero-dimensional ideal $\langle g_0, \ldots, g_{n-1} \rangle$. In addition, the dimension of the $\mathbb{F}_q$-vector space $P/\langle g_0, \ldots, g_{n-1} \rangle$ is d^n.

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

REMARK. We note that using Buchberger's First Criterion in Algorithm 2 is a special case of using Macaulay's trick [50].

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

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

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

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

```
Initialize(1^λ, 𝒫, d):           Sample():                      Finalize():
begin                            begin                          begin
  P ←$ P_λ;                        f ←$ P_{≤b};                   return (G = G′);
  G ←$ GBGen(1^λ, P, d);           f ← f − (f  mod G);          end
  return (1^λ, P);                 return f;
end                              end
```

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

It follows from Lemma 2 that the **Sample** procedure in Figure 1 returns elements of degree b which are uniformly distributed in $\langle G \rangle$. We note that if we instantiate $\mathsf{GBGen}()$ with $\mathsf{GBGen}_{\mathsf{dense}}()$ we must require $b \geq d$ in order to exclude the trivial case where **Sample** always returns zero.

We recall that given a Gröbner basis G of an ideal $\mathscr{I}$, $r = f \mod \mathscr{I} = f \mod G$ is the normal form of f with respect to the ideal $\mathscr{I}$. We sometimes drop the explicit reference to $\mathscr{I}$ when it is clear from the context which ideal we are referring to, and simply refer to r as the normal form of f. Computing normal forms is the ideal remainder problem which we formalise below.

Definition 7 (Ideal Remainder (IR) **Problem).** *The ideal remainder problem is defined through game $\mathsf{IR}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m}$, shown in Figure 2. The advantage of a ppt algorithm $\mathscr{A}$ in solving the* IR *problem is defined by*

$$\mathbf{Adv}^{\mathsf{ir}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m,\mathscr{A}}(\lambda) := \Pr\left[\mathsf{IR}^{\mathscr{A}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m}(\lambda) \Rightarrow \mathsf{T}\right] - 1/c,$$

where $c = q^{\dim_{\mathbb{F}_q}(P/\langle G\rangle)}$. An adversary is legitimate if it calls the **Sample** *procedure described in Figure 2 at most $m = m(\lambda)$ times. We also note that in the above advantage term, q, P and G denote the finite field, the polynomial ring, and the Gröbner basis which are generated during the game respectively.*

```
Initialize($1^\lambda, \mathscr{P}, d$):
begin
    $P \leftarrow_\$ \mathbf{P}_\lambda$;
    $G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d)$;
    return $(1^\lambda, P)$;
end

Sample():
begin
    $f \leftarrow_\$ P_{\leq b}$;
    $f' \leftarrow (f \mod G)$;
    return $f - f'$;
end

Challenge():
begin
    $f \leftarrow_\$ P_{\leq b}$;
    return $f$;
end

Finalize($r'$):
begin
    $r \leftarrow f \mod G$;
    return $r = r'$;
end
```

Fig. 2. Game $\mathsf{IR}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m}$.

In fact, we show below that under certain conditions the two problems are equivalent. That is, not only do Gröbner bases allow to solve the IR problem, but we also have the reverse reduction. Lemma 4 proves a weak form of this equivalence. That is, for Lemma 4 below to be meaningful we require that the IR adversary returns the correct answer with an *overwhelming* probability. This is due to the restriction that **Sample** can only be called a bounded number of times, and thus one cannot amplify the success probability of the IR adversary through repetition. We note that it is possible to prove a stronger statement than Lemma 4 using a proof technique from [15]. However, the weaker and simpler statement is sufficient in our context.

Informally, the reduction of the GB problem to the IR problem works as follows. Consider an arbitrary element g_i in the Gröbner basis G. We can write g_i as $m_i + \tilde{g}_i$ for some $\tilde{g}_i < g_i$ and $m_i = \mathrm{LM}(g_i)$. Now, assume the normal form of m_i is r_i and suppose that $r_i < m_i$. This implies that $m_i = \sum_{j=0}^{n-1} h_j g_j + r_i$ for some $h_i \in P$. Hence, we have $m_i - r_i \in \langle G\rangle$: an element $\in \langle G\rangle$ with leading monomial m_i. Repeat this process for all monomials up to and including degree d and accumulate the results $m_i - r_i$ in a list $\tilde{G}$. The list $\tilde{G}$ is a list of elements $\in \langle G\rangle$ with $\mathrm{LM}(\tilde{G}) \supseteq \mathrm{LM}(G)$ which implies $\tilde{G}$ is a Gröbner basis. We note that this is the core idea behind the FGLM algorithm [33] which allows to efficiently change the ordering of a Gröbner basis (and also the "Bulygin attack" in a different context [22]).

Lemma 4 (IR **Hard** ⇔ GB **Hard).** *For any ppt adversary $\mathscr{A}$ against the* IR *problem, there exists a polynomial* poly() *and a ppt adversary $\mathscr{B}$ against the* GB *problem such that*

$$\mathbf{Adv}^{\mathsf{ir}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m,\mathscr{A}}(\lambda)^{\mathrm{poly}(\lambda)} \leq \mathbf{Adv}^{\mathsf{gb}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m,\mathscr{B}}(\lambda).$$

Conversely, for any ppt adversary $\mathscr{B}$ against the GB *problem, there exists a ppt adversary $\mathscr{A}$ against the* IR *problem such that*

$$\mathbf{Adv}^{\mathsf{gb}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m,\mathscr{B}}(\lambda) = \mathbf{Adv}^{\mathsf{ir}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m,\mathscr{A}}(\lambda).$$

Proof. The second statement is classical. It is proven for instance in [26, p. 82] which shows that a Gröbner basis allows computing remainders modulo the ideal spanned by the basis in polynomial time.

To prove the opposite direction, we construct an algorithm $\mathscr{B}$ against the GB problem based on an algorithm $\mathscr{A}$ against the IR problem. This algorithm is described in Algorithm 3.

Algorithm 3: GB adversary $\mathscr{B}$ from IR adversary $\mathscr{A}$

```
1  begin
2      $\mathscr{B}$ receives $(1^\lambda, P)$;
3      $\tilde{G} \leftarrow \varnothing$; $F, k \leftarrow [], 0$;
4      query Sample() to get $f$;
5      $M \leftarrow$ the list of monomials of degree $\leq d$, smallest first;
6      for $m \in M$ do
7          if $\exists g \in \tilde{G}$ s.t. $\mathrm{LM}(g) \mid m$ then continue;
8          $c, t \leftarrow 0, 0$;
9          for $\tilde{m} \in M_{<m}$ do
10             $c \leftarrow_{\$} \mathbb{F}_q$;
11             $t \leftarrow t + c \cdot \tilde{m}$;
12         $t \leftarrow t \mod \tilde{G}$;
13         $k \leftarrow 0$;
14         run $\mathscr{A}(1^\lambda, P)$ as follows:
15         if $\mathscr{A}$ queries Sample() then
16             if $k = \#F$ then
17                 query Sample() to get $h$;
18                 $F \leftarrow F \cup \{h\}$; $k \leftarrow k + 1$;
19             return $F_k$;
20         if $\mathscr{A}$ queries Challenge() then
21             return $f + m + t$;
22         if $\mathscr{A}$ calls Finalize$(r')$ then
23             set $r \leftarrow r' - t$;
24         if $r < m$ then
25             $\tilde{G} \leftarrow \tilde{G} \cup \{m - r\}$;
26     call Finalize$(\tilde{G})$;
27 end
```

First we consider correctness. If r' returned by $\mathscr{A}$ in line 23 satisfies $r' = f + m + t \mod G$ then $m - r = m + t - r'$ in line 25 is an element in $\langle G \rangle$ with leading monomial m. To see this recall that we have $f + m + t = \sum_{j=0}^{n-1} h_j g_j + r'$ for $0 \leq j < n$, $h_j \in P$ and $r' \notin G$ which implies – since $f \in \langle G \rangle$ – that $m + t - r' = \sum_{j=0}^{n-1} \tilde{h}_j g'_j$ for $0 \leq j < n$, $\tilde{h}_j \in P$ and hence $m + t - r' \in \langle G \rangle$. By construction $t < m$ and we only add elements to $\tilde{G}$ with $r < m$. We compute such elements for every monomial of degree $\leq d$. In particular, we compute such elements for every $\mathrm{LM}(g_i)$. Since $\mathrm{LM}(\tilde{G}) \supseteq \mathrm{LM}(G)$ we have that $\tilde{G}$ is a Gröbner basis for the ideal $\langle G \rangle$.

Now, let us consider resources. Algorithm 3 runs in polynomial time. The outer loop is repeated $n(\lambda)^d$ times in the general case which is polynomial in λ by assumption. Note that if $\mathsf{GBGen}(\cdot) = \mathsf{GBGen}_{\mathsf{dense}}(\cdot)$ we can set $M \leftarrow [x_{n-1}^d, \ldots, x_0^d]$ in line 5 and thus repeat the outer loop only $n(\lambda)$ times. If $k-1$ is an upper bound on the number of queries to **Sample** that $\mathscr{A}$ makes, $\mathscr{B}$ makes at most k queries to its **Sample** oracle.

Finally, since we run $n(\lambda)^d$ independent copies of $\mathscr{A}$ for $n(\lambda)^d$ different challenges, and require all of them to return the correct results, the overall advantage is the product of the advantages of $\mathscr{A}$'s. □

The decisional variant of the IR problem is to decide whether the normal form of some element modulo an ideal is zero or not, i.e., whether this element is in the ideal or not. This is the well-known ideal membership problem formalised below. We note that solving this problem was the original motivation which lead to the discovery of Gröbner bases [19].

Definition 8 (Ideal Membership (IM) Problem). *The ideal membership problem is defined through the game $\mathsf{IM}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m}$ as shown in Figure 3. The advantage of a ppt algorithm $\mathscr{A}$ in solving IM is defined by*

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

An adversary is legitimate if it calls the **Sample** *procedure described in Figure 3 at most $m = m(\lambda)$ times.*

```
Initialize(1^λ, 𝒫, d):
begin
    P ←$ P_λ;
    G ←$ GBGen(1^λ, P, d);
    c ←$ {0,1};
    return (1^λ, P);
end

Sample():
begin
    f ←$ P_{≤b};
    f' ← f  mod G;
    return f − f';
end

Challenge():
begin
    f ←$ P_{≤b};
    if c = 1 then
        f ← f − (f  mod G);
    return f;
end

proc. Finalize(c'):
begin
    return (c = c');
end
```

Fig. 3. Game $\mathsf{IM}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m}$.

Clearly any adversary which can solve the IR problem can also solve the IM problem. However, if the search space of reminders modulo $\langle G \rangle$ is sufficiently small, i.e., when $q^{\dim_{\mathbb{F}_q}(P/\langle G \rangle)} = \mathrm{poly}(\lambda)$, and under similar assumptions as for Lemma 4, one can also perform the converse reduction. That is, one can solve the IR problem using an oracle for the IM problem. Lemma 5 below proves this equivalence for the special case of $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$. Once again, this is sufficient in our context. As before, for Lemma 5 to be meaningful we require that the IM adversary returns the correct answer with *overwhelming* probability.

Informally, the construction of an IR adversary from an IM adversary proceeds as follows. Let $\tilde{f}$ be the challenge polynomial. The attacker simply exhaustively searches all elements of the $\mathbb{F}_q$ vector space $P/\langle G \rangle$ until the right remainder r is found. This occurs if $f - r \in \langle G \rangle$ and can be then detected using an IM adversary. However, there is a technical difficulty here. In general, the attacker does not necessarily know the support (or the basis) of $P/\langle G \rangle$ and hence cannot know how to construct r. However, in our case we assume that $\mathsf{GBGen}(\cdot) = \mathsf{GBGen}_{\mathsf{dense}}(\cdot)$ and this difficulty does not arise. Indeed, a basis of $P/\langle G \rangle$ is given by the monomials $\prod_{i=0}^{n-1} x_i^{d_i}$, for all $d_i, 0 \le d_i < d$. In a more general setting, we would have to discover $P/\langle G \rangle$ as well (cf. proof of Lemma 7).

Lemma 5 (IM Hard $\Leftrightarrow$ IR Hard for poly-sized $q^{\dim_{\mathbb{F}_q}(P/\langle G \rangle)}$). *Assume that $q(\lambda)^{\dim_{\mathbb{F}_q}(P/\langle G \rangle)}$ is* $\mathrm{poly}(\lambda)$ *sized for any $P \in [\mathbf{P}_\lambda]$ and $G \in \mathsf{GBGen}(1^\lambda, P, d)$. Then for any ppt adversary $\mathscr{A}$ against the* IM *problem, there exists a ppt adversary $\mathscr{B}$ against the* IR *problem such that*

$$\mathbf{Adv}^{\mathsf{im}}_{\mathscr{P},\mathsf{GBGen}_{\mathsf{dense}}(\cdot),d,b,m,\mathscr{A}}(\lambda)^{\mathrm{poly}(\lambda)} \le \mathbf{Adv}^{\mathsf{ir}}_{\mathscr{P},\mathsf{GBGen}_{\mathsf{dense}}(\cdot),d,b,m,\mathscr{B}}(\lambda).$$

Conversely, for any ppt adversary $\mathscr{B}$ against the IR *problem, there exists a ppt adversary $\mathscr{A}$ against the* IM *problem such that*

$$\mathbf{Adv}^{\mathsf{ir}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m,\mathscr{B}}(\lambda) = \mathbf{Adv}^{\mathsf{im}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,m,\mathscr{A}}(\lambda).$$

Proof. The second statement is clearly true since $f \in \mathscr{I}$ iff $f \mod \mathscr{I} = 0$.

To prove the former direction, we construct an algorithm $\mathscr{B}$ against the IR problem based on an algorithm $\mathscr{A}$ against the IM problem. The procedure is described in Algorithm 4. By assumption we know that $p \notin \langle G \rangle$. When $\mathscr{A}$ returns $c = 1$ for $f - p$ we have that $f - p \in \langle G \rangle$ with some non-negligible probability. Hence, $f = \sum_{j=0}^{n-1} h_j g_j + p$ for $h_j \in P$ which implies $p = f \mod \langle G \rangle$ with non-negligible probability. If more than one run of $\mathscr{A}$ returns a candidate, we have a contradiction and simply pick a random candidate.

To consider resources, note that the outer loop in line 5 is iterated $q^{\#M}$ times where $\#M = \dim_{\mathbb{F}_q}(P/\langle G \rangle)$. By assumption, we have that $q^{\#M} = \mathrm{poly}(\lambda)$. Hence $\mathscr{B}$ runs in time polynomial in λ. If $\mathscr{A}$ makes at most k queries to its **Sample** oracle, since we reuse samples, $\mathscr{B}$ also makes k queries to its **Sample** oracle. □

4.1 Hardness Assumption

It is well-known [11] that the worst-case complexity of Gröbner bases is double 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., those ideals with a finite number of common roots. In this section, we recall a number of complexity results for these type of systems.

Lazard [46] showed that computing the Gröbner basis for a system of polynomials is equivalent to performing Gaussian elimination on the so-called Macaulay matrices $\mathscr{M}^{\mathrm{acaulay}}_{d,m}$ for $d, 1 \le d \le D$ for some D.

Definition 9 (Macaulay Matrix). *For a set of m polynomials $f_0, \ldots, f_{m-1}$ we define the* Macaulay matrix $\mathscr{M}^{\mathrm{acaulay}}_{d,m}$ *of degree d as follows: list "horizontally" all the degree*

```
Algorithm 4: IR adversary $\mathscr{B}$ from IM adversary $\mathscr{A}$
begin
    $\mathscr{B}$ receives $(1^\lambda, P)$;
    $L \leftarrow \varnothing$; $F, k \leftarrow \varnothing, 0$;
    $M \leftarrow$ an ordered list of all monomials $\prod_{i=0}^{n-1} x_i^{d_i}$ for each $d_i, 0 \leq d_i < d$;
    query Challenge() to get $f$ ;
    for $v \in \mathbb{F}_q^{\#M}$ do
        $p \leftarrow \sum v_i m_i$ for $0 \leq i < |M|, m_i \in M$;
        $k \leftarrow 0$;
        run $\mathscr{A}(1^\lambda, P)$ as follows:
        if $\mathscr{A}$ queries Sample() then
            if $k = \#F$ then
                query Sample() to get $h$;
                $F \leftarrow F \cup \{h\}$; $k \leftarrow k + 1$;
            return $F_k$;
        if $\mathscr{A}$ queries Challenge() then
            return $f - p$;
        if $\mathscr{A}$ calls Finalize($c$) then
            if $c = 1$ then $L \leftarrow L \cup \{p\}$;
    $p \leftarrow_\$ L$;
    call Finalize($p$);
end
```

d monomials from smallest to largest 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:

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

Theorem 3. *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 $\mathscr{M}^{\text{acaulay}}_{d,m}$ matrices for $d, 1 \leq d \leq D$ computes a Gröbner basis of $\langle F \rangle$.*

The F_4 algorithm [31] 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" [5, 1]. Consequently, the complexity is bounded by the degree D and the number of polynomials considered at each degree. For F_5 [32] and the matrix-F_5 variant [35] 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 [8] is useful.

Definition 10 (Semi-Regular Sequence of Degree D). *Let $f_0, \dots, f_{m-1} \subset P$ be homogeneous polynomials of degrees $d_0, \dots, d_{m-1}$ respectively. We call this system a* semi-regular sequence of degree D *if:*

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

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

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

Definition 11 (Semi-regular Sequence [8, 9, 7]). *Let $f_0, \dots, f_{m-1} \subset P$ be a system of homogeneous polynomials of degree b. We call this system a* semi-regular sequence *if the degree of semi-regularity of the system is given by the index of the first non-positive coefficient of:*

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

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 [8, 9, 7].

Theorem 4. *Assuming that F is a semi-regular sequence, the complexity of the currently best known algorithms (i.e., F_5) to solve the* GB *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 the degree of semi-regularity of the system.

Concrete (asymptotic) bounds for the degree of semi-regularity for semi-regular sequences of degree 2 can be found in [8]. 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 $m(\lambda) = c \cdot n(\lambda)$ or $m(\lambda) = c \cdot n(\lambda)^2$ quadratic polynomials in some ideal $\mathscr{I} \subset \mathbb{F}_q[x_0, \ldots, x_{n-1}]$, the Gröbner basis of $\mathscr{I}$ can be computed in exponential or polynomial time in $n(\lambda)$ respectively.*

This leads us to the following hardness assumption of the $\mathsf{GB/IR/IM}$ problems.

Definition 12 **($\mathsf{GB/IR/IM}$ Assumption).** *Let $\mathscr{P}$ be such that $n(\lambda) = \Omega(\lambda)$. Assume $b - d > 0$, $b > 1$, and that $m(\lambda) = c \cdot n(\lambda)$ for a constant $c \geq 1$. Then the advantage of any ppt algorithm in solving the* $\mathsf{GB/IR/IM}$ *problem is negligible as function of λ.*

5 Symmetric Polly Cracker: Noise-Free Version

5.1 Homomorphic Symmetric Encryption

SYNTAX. A *homomorphic symmetric-key encryption scheme* (HSKE) 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 a 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. An HSKE scheme is correct if for any $\lambda \in \mathbb{N}$, any $(\mathsf{SK}, \mathsf{PK}) \in [\mathsf{Gen}(1^\lambda)]$, any t messages $\mathsf{m}_i \in \mathsf{MsgSp}(\mathsf{PK})$, any $\mathsf{c} \in [\mathsf{Enc}(\mathsf{m}, \mathsf{SK})]$, any circuit $C \in \mathsf{FunSp}(\mathsf{PK})$, and any t ciphertexts $\mathsf{c}_i \in [\mathsf{Enc}(\mathsf{m}_i, \mathsf{PK})]$, 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 $\mathsf{Dec}(\mathsf{c}_{\mathsf{evl}}, \mathsf{SK}) = C(\mathsf{m}_0, \ldots, \mathsf{m}_{t-1})$. We do not necessarily require correctness 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 key-pair $(\mathsf{SK}, \mathsf{PK}) \in [\mathsf{Gen}(1^\lambda)]$, any circuit $C \in \mathsf{FunSp}(\mathsf{PK})$, any set of t messages $\mathsf{m}_i \in \mathsf{MsgSp}(\mathsf{PK})$, any ciphertext $\mathsf{c}_i \in [\mathsf{Enc}(\mathsf{m}_i, \mathsf{SK})]$, and any evaluated ciphertext $\mathsf{c}_{\mathsf{evl}} \in [\mathsf{Eval}(\mathsf{c}_0, \ldots, \mathsf{c}_{t-1}, C, \mathsf{PK})]$, 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 is similar to that of the an HSKE scheme, except that the encryption algorithm takes the public key as an input.

5.2 The Scheme

In this section we formally define the (noise-free) symmetric Polly Cracker encryption scheme. We present a family of schemes parameterised not only by the underlying computational polynomial ring scheme $\mathscr{P}$, but also by a Gröbner basis generation algorithm, which itself depends on a degree bound d, and a second degree bound b. Our parameterised scheme, which we write as $\mathscr{SPC}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b}$, is presented in Figure 4. The message space is $P/\mathscr{I}$.

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

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

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

$\mathsf{Eval}(c_0, \ldots, c_{t-1}, C, \mathsf{PK})$:
begin
    apply the $\mathsf{Add}$ and $\mathsf{Mult}$
      gates of $C$ over $P$;
    return the result;
end
```

Fig. 4. The (noise-free) Symmetric Polly Cracker scheme $\mathscr{SPC}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b}$.

CORRECTNESS OF EVALUATION. Consider the two ciphertexts $c_0 = \sum h_{0,j} g_j + m_0$ and $c_1 = \sum h_{1,j} g_j + m_1$. Addition and multiplication of the two ciphertexts c_0, c_1 are given by

$$\begin{aligned}
c_0 + c_1 &= \sum h_{0,j} g_j + m_0 + \sum h_{1,j} g_j + m_1 \\
&= \sum (h_{0,j} + h_{1,j}) g_j + m_0 + m_1 \\
c_0 \cdot c_1 &= \left(\sum h_{0,j} g_j + m_0\right) \cdot \left(\sum h_{1,j} g_j + m_1\right) \\
&= \left(\sum h_{0,j} g_j\right) \cdot \left(\sum h_{1,j} g_j\right) + \sum h_{0,j} g_j \cdot m_1 + \sum h_{1,j} g_j \cdot m_0 + \ m_0 m_1 \\
&= \sum \tilde{h}_j g_j + m_0 m_1, \text{ for some } \tilde{h}_i,
\end{aligned}$$

from which 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. Additions are free and do not increase the size of the ciphertext, whereas multiplications square the size of the ciphertext.

REMARKS. If $d = 1$ and $q(\lambda) = \text{poly}(\lambda)$ we have to 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 bit results in a ciphertext of length $\binom{n+b}{b} = \mathscr{O}\left(n^b\right)$ bits. The complexity of both encryption and decryption for fresh ciphertexts are $\mathscr{O}\left(n^b\right)$ ring operations.

5.3 Security

As we will show shortly, the above scheme only achieves a weak version of chosen-plaintext security, which allows access to a limited number of ciphertexts, as defined next.

Definition 13 (*m*-time $\mathsf{IND\text{-}BCPA}$ Security). *The m-time* $\mathsf{IND\text{-}BCPA}$ *security of a (homomorphic) symmetric-key encryption scheme* $\mathscr{SKE}$ *is defined by requiring that the advantage of any ppt adversary* $\mathscr{A}$ *given by*

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

is negligible as a function of the security parameter λ*. The game* $\mathsf{IND\text{-}BCPA}_{m,\mathscr{SKE}}$ *is shown in Figure 5. The difference with the usual* $\mathsf{IND\text{-}CPA}$ *security is that the adversary can query its encryption and left-or-right oracles at most* $m(\lambda)$ *times.*

```
Initialize($1^\lambda$):
begin
    $(\mathsf{SK},\mathsf{PK}) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$;
    $c \leftarrow_{\$} \{0,1\}$;
    $i \leftarrow 0$;
    return $\mathsf{PK}$;
end

Encrypt($\mathsf{m}$):
begin
    $i \leftarrow i+1$;
    if $i \geq m(\lambda)$ then
        return $\perp$;
    $\mathsf{c} \leftarrow_{\$} \mathsf{Enc}(\mathsf{m},\mathsf{SK})$;
    return $\mathsf{c}$;
end

Left-Right($\mathsf{m}_0,\mathsf{m}_1$):
begin
    $\mathsf{c} \leftarrow_{\$} \mathsf{Enc}(\mathsf{m}_c,\mathsf{SK})$;
    return $\mathsf{c}$;
end

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

Fig. 5. Game $\mathsf{IND\text{-}BCPA}_{m,\mathscr{SKE}}$. An adversary is legitimate if it calls oracle **Left-Right** exactly once on two message of equal lengths.

The security guarantees of this scheme are as follows.

Theorem 5. *Let* $\mathscr{A}$ *be a ppt adversary against the m-time* $\mathsf{IND\text{-}BCPA}$ *security of the scheme described in Figure 4. Then there exists a ppt adversary* $\mathscr{B}$ *against the* IM *problem such that for all* $\lambda \in \mathbb{N}$ *we have*[2]

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

Conversely, let $\mathscr{A}$ *be a ppt adversary against the* IM *problem. Then there exists a ppt adversary* $\mathscr{B}$ *against the m-time* $\mathsf{IND\text{-}BCPA}$ *security of the scheme described in Figure 4 such that for all* $\lambda \in \mathbb{N}$ *we have*

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

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 element of the quotient. Now deciding ideal membership directly leads to a distinguishing attack.

For the first part, we construct an algorithm $\mathscr{B}$ attacking the scheme based on an algorithm $\mathscr{A}$ attacking the IM problem as follows.

[2] We sometimes omit the subscript from schemes to ease notation. For example we have written $\mathscr{SPC}$ for $\mathscr{SPC}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b}$.

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

```
begin
    $\mathscr{B}$ receives $(1^\lambda, P)$;
    run $\mathscr{A}(1^\lambda, P)$ as follows;
    if $\mathscr{A}$ queries Encrypt$(m)$ then
        query Sample() to get $f$;    return $f + m$;
    if $\mathscr{A}$ queries Left-Right$(m_0, m_1)$ then
        query Challenge() to get $f$;    $c \leftarrow_\$ \{0,1\}$;    return $f + m_c$;
    if $\mathscr{A}$ calls Finalize$(b')$ then
        call Finalize$(b = b')$;
end
```

Now if the sample returned from the **Challenge** oracle to $\mathscr{B}$ 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 $\mathscr{A}$ is run in an environment which is identical to the IND-BCPA game. Hence in this case the probability that $c = c'$ is equal to the probability that $\mathscr{A}$ wins the IND-CPA game. The theorem follows. □

As a corollary, observer that when $m(\lambda) = \mathscr{O}(\lambda^b)$ one can use Corollary 2 to construct an adversary which breaks the IND-BCPA$_{m,\mathscr{SKE}}$ security of $\mathscr{SPC}$ in polynomial time. Thus we can only hope to achieve security in the bounded model for this scheme. In the remainder of the paper we show how to overcome this security limitation.

6 Symmetric-to-Asymmetric Conversion

Given the security limitation of the symmetric Polly Cracker scheme, our goal for the rest of the paper will be to convert the scheme to a scheme which is not only fully IND-CPA secure but also is (at least) additively homomorphic. Once we achieve this, then it is possible to construct a public-key scheme using the homomorphic features of the symmetric scheme by applying various generic conversions. In the literature there are two prominent such conversions:

(A) Publish many encryptions of zero F_0 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 [57].

(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 [53].

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 $\mathscr{SPC}$ is not IND-CPA secure the above transformations cannot be used.[3] However, one could envisage a larger class of transformations which might lead to a fully secure additively homomorphic SKE (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 $\mathscr{I} \subset P$. The decryption algorithm computes $c \mod \mathscr{I} = c \mod G$ (perhaps together with some post-processing such as a mod 2 operation). Thus, the message space is (essentially) $P/\mathscr{I}$. We assume that $P/\mathscr{I}$ is known.[4]
2. The public key consists of elements $f_i \in P$. We assume that the remainder of these elements modulo the ideal $\mathscr{I}$, i.e., $r_i := f_i \mod \mathscr{I}$, 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 (possibility depending on f_i), and r is an encoding in $P/\mathscr{I}$ of the message.
4. The construction of the ciphertext does not encode knowledge of $\mathscr{I}$ beyond f_i. That is, we have
$$\left(\sum_{i=0}^{N-1} h_i f_i + r\right) \mod \mathscr{I} = \sum_{i=0}^{N-1} h_i r_i + r.$$
Hence we have that $\left(\sum_{i=0}^{N-1} h_i r_i + r\right) \in P/\mathscr{I}$ 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 $P_{\leq b}$.

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 non-zero element in $\mathscr{I}$ using some arbitrary algorithm $\mathscr{A}$, we are unable to prove anything about the transformation. Furthermore, it is plausible that for any given $\mathscr{A}$ a similar impossibility result can be obtained if the remaining conditions hold (although we are unable to prove this at present).

Note that the two transformations listed above are special linear cases of this methodology. For transformation (A) we have that $f_i \in \mathscr{I}$ (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 [10] to discourage the use of Gröbner bases in the construction of public-key encryption schemes.

[3] As stated above, when applied to a *specific* scheme, the transformations might still result in secure schemes. However, it can be shown that the security of the transformed schemes are *equivalent* to that of the underlying scheme.

[4] For instance if $d = 1$ then $P/\mathscr{I} = \mathbb{F}_q$, or if $\mathsf{GBGen}_{\mathsf{dense}}(\cdot)$ is used then a basis for $P/\mathscr{I}$ as a vector space are the $\prod(x_i^{d_i})$ for $0 \leq d_i < d$.

Theorem 6 ([28]). *Let $\mathscr{I} = \langle f_0, \ldots, f_{m-1} \rangle$ be an ideal in the polynomial ring $P = \mathbb{F}[x_0, \ldots, x_{n-1}]$, h be such that $\deg(h) \le D$, and let $h - (h \mod \mathscr{I}) = \sum_{i=0}^{m-1} h_i f_i$, where $h_i \in P$ and $\deg(h_i f_i) \le D$. 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 $h \mod \mathscr{I}$ 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 compute a Gröbner basis for the ideal. Note that in itself Theorem 6 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 6 but with uniformly random output. Theorem 6 then allows us to compute normal forms which (because of the randomness of h) allows the computation of a Gröbner basis by Lemma 4. 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 7. *Let $G = \{g_0, \ldots, g_{s-1}\}$ be the reduced Gröbner basis of the zero-dimensional ideal $\mathscr{I}$ in the polynomial ring $P = \mathbb{F}[x_0, \ldots, x_{n-1}]$ where each $\deg(g_i) \le d$. Assume that $P/\mathscr{I}$ is known. Furthermore, let $F = \{f_0, \ldots, f_{N-1}\}$ be a set of polynomials with known $r_i := f_i \mod \mathscr{I}$. Let $\mathscr{A}$ be a ppt algorithm which given F produces elements $f = \sum h_i f_i + r$ with $\deg(f) \le b$, $h_i \in P$, $b \le B$, $\deg(h_i f_i) \le B$, and $(f \mod \mathscr{I}) = \sum h_i r_i + r$. Suppose further that the outputs of $\mathscr{A}$ are computationally indistinguishable from random elements in $P_{\le b}$. Then there exists an algorithm which computes a Gröbner basis for $\mathscr{I}$ from F in $\mathscr{O}\left(n^{3B}\right)$ field operations.*

Proof. Writing $\tilde{f}_i = f_i - r_i$ and $h = \sum_{i=0}^{N-1} h_i f_i + r$, we get $h = \sum_{i=0}^{N-1} h_i \tilde{f}_i + \tilde{r}$ for some $\tilde{r} \in P/\mathscr{I}$. Hence h satisfies the condition of Theorem 6, and we can compute the remainder of all elements of degree b produced by $\mathscr{A}$ by computing a Gröbner basis up to degree B. From Theorem 4 we know that this costs $\mathscr{O}\left(n^{\omega B}\right)$ field operations where $\omega < 3$ is the linear algebra constant.

We now have an algorithm which returns the remainder for arbitrary elements of $P_{\le b}$ with probability 1. This follows since h is computationally indistinguishable from random elements in $P_{\le b}$. More explicitly, we can generate the system parameters, including the Gröbner basis, and provide the algorithm which either an output of $\mathscr{A}$ 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 $\mathscr{A}$.

Now Lemma 4 shows that IR computation is equivalent to compute a Gröbner basis by making at most $\binom{n+b}{b} = \mathscr{O}\left(n^b\right)$ queries to the IR oracle. (Note that the above IR oracle has an overwhelming success probability.) Each such query costs at most $\binom{n+b}{b}^2 = \mathscr{O}\left(n^{2b}\right)$ field operations. Therefore the overall cost of the second step is $\mathscr{O}\left(n^{3b}\right)$.[5] Hence the overall complexity is $\mathscr{O}\left(n^{\omega B}\right)$ for the first step and $\mathscr{O}\left(n^{3b}\right)$ for

[5] 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 $\mathscr{I}$.

the second step with $b \leq B$ and $\omega < 3$ from which an overall complexity of $\mathscr{O}\left(n^{3B}\right)$ follows. □

REMARK. Although the above impossibility result is presented for public-key encryption schemes, due to the equivalence result of [53], it also rules out the existence of additively homomorphic symmetric Polly Cracker-style schemes with full $\mathsf{IND\text{-}CPA}$ security.

Therefore, if for some degree $b \geq d$ computationally uniform elements of $P_{\leq b}$ can be produced using the public key $f_0, \dots, f_{N-1}$, there is an attacker which recovers the secret key $g_0, \dots, g_{s-1}$ in essentially the same complexity. Hence, while conceptually simple and provably secure up to some bound, our symmetric Polly Cracker scheme $\mathscr{SPC}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b}$ does not provide a valid building block for constructing a fully-homomorphic public-key encryption scheme.

Our goal in the rest of the paper is to achieve full $\mathsf{IND\text{-}CPA}$ security for a symmetric Polly Cracker-type scheme. To this end, we introduce noisy variants of GB, IR and IM in the next section. These variants ensure that the conditions of Theorem 7 do not hold any more. In particular, the condition that $r_i := f_i \mod \mathscr{I}$ are known will be no longer valid.

7 Gröbner Bases with Noise

In this section, we introduce noisy variants of the problems presented in Section 4. 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 $\mathscr{SPC}$. Similarly to [57, 52] we expect a problem which is efficiently solvable in the noise-free setting to be hard in the noisy setting. We will justify this assumption in Section 7.1 by arguing that our construction can be seen as a generalisation of [57, 52].

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

Definition 14 (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.*

As an example note that if $q = 2$ then $\chi_{\alpha,2}$ is a Bernoulli distribution with just one parameter $0 \leq p \leq 1$, 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 $\mathscr{I}$. However, we are now only given access to a noisy sample oracle which provides polynomials which are not necessarily in $\mathscr{I}$ but rather are “close” approximations to elements of $\mathscr{I}$. Here the term “close” is made precise using a noise distribution χ on $P/\mathscr{I}$.

Definition 15 (Gröbner Basis with Noise (GBN) Problem). *The Gröbner basis with noise problem is defined through the game $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$ as shown in Figure 6.*

The advantage of a ppt algorithm $\mathscr{A}$ in solving the GBN *problem is*

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

```
Initialize($1^\lambda, \mathscr{P}, d$):
begin
    $P \leftarrow_\$ \mathbf{P}_\lambda$;
    $G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d)$;
    return $(1^\lambda, P)$;
end

Sample():
begin
    $f \leftarrow_\$ P_{\leq b}$;
    $e \leftarrow_\$ \chi$;
    $f \leftarrow f - (f \mod G) + e$;
    return $f$;
end

Finalize($G'$):
begin
    return $(G = G')$;
end
```

Fig. 6. Game $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$.

The essential difference between the noisy and noise-free versions of the GB problem is that by adding noise we have eliminated the restriction on the adversary to call the **Sample** oracle a bounded number of times. Stated 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.

REMARK. When $d = 1$ the GBN problem is closely related to the Max-MQ problem, the problem of finding an assignment for m 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. In [42] it was shown that if all f_i are square-free it is NP-hard to approximate this problem to within a factor of $q - \varepsilon$ for ε a small positive number. Latter [59] proves that the minimal approximation ratio that can be achieved in polynomial time for Max-MQ is q. The most significant difference between the GBN problem 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 $d = 1$.

As in the noise-free setting, we can ask various questions about the ideal $\mathscr{I}$ spanned by G. One such example is solving the ideal remainder problem with access to noisy samples from $\mathscr{I}$.

Definition 16 (Ideal Remainder with Noise (IRN) Problem). *The ideal remainder with noise problem is defined through the game* $\mathsf{IRN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$ *as shown in Figure 7. The advantage of a ppt algorithm $\mathscr{A}$ in solving the* IRN *problem is*

$$\mathbf{Adv}^{\mathsf{irn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{A}}(\lambda) := \Pr\left[\mathsf{IRN}^{\mathscr{A}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}(\lambda) \Rightarrow \mathsf{T}\right] - 1/q(\lambda)^{\dim_{\mathbb{F}}(P/\langle G\rangle)}.$$

In fact, the above two problems are equivalent as shown in the lemma below. Compared to the noise-free version, we no longer need the IM adversary to be overwhelmingly successful, as there are no restrictions on the number of calls that can be made to the **Sample** procedure.

Lemma 6 (IRN Hard $\Leftrightarrow$ GBN Hard). *For any ppt adversary $\mathscr{A}$ against the* IRN *problem, there exists a ppt adversary $\mathscr{B}$ against the* GBN *problem such that*

$$\mathbf{Adv}^{\mathsf{irn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{A}}(\lambda) \leq \mathbf{Adv}^{\mathsf{gbn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{B}}(\lambda).$$

```
Initialize($1^\lambda, \mathscr{P}, d$):
begin
    $P \leftarrow_\$ \mathbf{P}_\lambda$;
    $G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d)$;
    return $(1^\lambda, P)$;
end

Sample():
begin
    $f \leftarrow_\$ P_{\leq b}$;
    $e \leftarrow_\$ \chi$;
    $f \leftarrow f - (f \mod G) + e$;
    return $f$;
end

Challenge():
begin
    $f \leftarrow_\$ P_{\leq b}$;
    return $f$;
end

Finalize($r'$):
begin
    $r" = f \mod G$;
    return $r' = r"$;
end
```

Fig. 7. Game $\mathsf{IRN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$.

Conversely, for any ppt adversary $\mathscr{B}$ against the GBN *problem, there exists a ppt adversary $\mathscr{A}$ against the* IRN *problem such that*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{B}}(\lambda) = \mathbf{Adv}^{\mathsf{irn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{A}}(\lambda).$$

Proof. To prove the first statement, we construct a procedure $\mathscr{B}$ against the GBN problem based on an algorithm $\mathscr{A}$ against the IRN as described in in Algorithm 6.

Algorithm 6: GBN adversary $\mathscr{B}$ from IRN adversary $\mathscr{A}$

```
begin
    $\mathscr{B}$ receives $(1^\lambda, P)$;
    $G \leftarrow \varnothing$;
    for $0 \leq d \leq b$ do
        $M_d \leftarrow$ all monomials of degree $d$ sorted ascendingly;
        for $m \in M_d$ do
            if $\nexists\, g \in G$ s.t. $\mathrm{LM}(g) \mid m$ then
                for $0 \leq j < \mathrm{poly}(\lambda)/\varepsilon$ do
                    query Sample() to get $f$;
                    run $\mathscr{A}(1^\lambda, P)$ as follows:
                    if $\mathscr{A}$ queries Sample() then
                        query Sample() to get $\tilde{f}$;
                        return $\tilde{f}$
                    if $\mathscr{A}$ queries Challenge() then
                        return $f + m$;
                    if $\mathscr{A}$ calls Finalize($r'$) then
                        set $r_j \leftarrow r'$;
                $r \leftarrow$ majority vote on $r_j$;
                if $r \neq m$ then $G \leftarrow G \cup \{m - r\}$;
    call Finalize($G$);
end
```

Algorithm 6 is correct: If $r = (f + m) \mod \mathscr{I}$ we have that $r = m + f - \sum h_i g_i$ for some $h_i \in P$ and thus we have $m - r = -\sum h_i g_i - f$ is an element of the ideal. Since we assume that $b \geq d$ we know that at some point we query $m = \mathrm{LM}(g)$ for all $g \in G$ and thus construct elements $(\mathrm{LM}(g) - r) \in \mathscr{I}$ with $r < \mathrm{LM}(g)$ which is sufficient to ensure G is a Gröbner basis.

Algorithm 6 is polynomial time: The outer loop in line 4 is repeated at most $\binom{n+b}{b} = \mathscr{O}(n^b)$ times as there are only $\binom{n+b}{b}$ monomials up to degree b. If k is an upper bound on the number of queries to **Sample** that $\mathscr{A}$ makes, $\mathscr{B}$ makes at most $n^b \cdot \mathrm{poly}(\lambda)/\varepsilon \cdot k$ queries to its **Sample** oracle, which is polynomial in λ if ε is not exponentially small.

To prove the second statement, we construct algorithm $\mathscr{A}$ against the IRN problem based on algorithm $\mathscr{B}$ against GBN in Algorithm 7.

Algorithm 7: IRN adversary $\mathscr{A}$ from GBN adversary $\mathscr{B}$

```
begin
    $\mathscr{A}$ receives $(1^\lambda, P)$;
    run $\mathscr{B}(1^\lambda, P)$ as follows:
    if $\mathscr{B}$ queries Sample() then
        query Sample() to get $f$;
        return $f$;
    if $\mathscr{B}$ calls Challenge() then
        query Challenge() to get $\tilde{f}$;
        return $\tilde{f}$
    if $\mathscr{B}$ calls Finalize($G$) then
        $r \leftarrow \tilde{f} \mod G$;
        call Finalize($r$);
end
```

Algorithm 7 is correct. This follows immediately from the property of a Gröbner basis G to allow to compute the unique remainder of any f module the ideal $\langle G \rangle$.

Algorithm 7 runs in polynomial time. It also makes exactly as many queries to its **Sample** oracle as $\mathscr{B}$ does to its own **Sample** oracle. Furthermore, the operation $f \mod G'$ is polynomial time in the size of f. □

Similarly to the noise-free setting, the ideal membership with noise (IMN) problem is the decisional variant of the IRN (and hence the GBN) problem. However, in the noisy setting we have the choice between a noisy and noise-free challenge polynomial. In the definition below noisy challenges are provided and the adversary wins the game if he can distinguish whether an element was sampled uniformly from $P_{\leq b}$ or from $\mathscr{I} + \chi$.

Definition 17 (Ideal Membership with Noise (IMN) Problem). *The ideal membership with noise problem is defined through* $\mathsf{IMN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$ *as shown in Figure 8. The advantage of a ppt algorithm* $\mathscr{A}$ *in solving the* IMN *problem is defined by*

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

```
Initialize($1^\lambda, \mathscr{P}, d$):
begin
    $P \leftarrow_\$ \mathbf{P}_\lambda$;
    $G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d)$;
    $c \leftarrow_\$ \{0,1\}$;
    return $(1^\lambda, P)$;
end

Sample():
begin
    $f \leftarrow_\$ P_{\leq b}$;
    $e \leftarrow_\$ \chi$;
    $f' \leftarrow f \mod G$;
    $f \leftarrow f - f' + e$;
    return $f$;
end

Challenge():
begin
    $f \leftarrow_\$ P_{\leq b}$;
    if $c = 1$ then
        $e \leftarrow_\$ \chi$;
        $f' \leftarrow f \mod G$;
        $f \leftarrow f - f' + e$;
    return $f$;
end

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

Fig. 8. Game $\mathsf{IMN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$.

Our definition of the IMN problem can be seen as an instantiation of Gentry's ideal coset problem [37] since both problems require distinguishing uniformly chosen elements in $P_{\leq b}$ from those in $\mathscr{I} + \chi$. Our problem, however, assumes noisy samples since it is clear from Section 4 that otherwise the problem is easy.

Again, we would like to have a decision-to-search reduction, that is, we would like to have an equivalence between the IRN and IMN problems. This equivalence holds when the search space of remainders is polynomial in λ, namely when

$$q(\lambda)^{\dim_{\mathbb{F}_q}(\mathscr{P}(\lambda)/\mathsf{GBGen}(\cdot))} = \mathrm{poly}(\lambda).$$

The intuition behind this reduction is that the adversary can exhaustively search the quotient ring and use the IMN oracle to verify his guess. Once again, a technical difficulty arises as the adversary does not know the search space $P/\mathscr{I}$ and thus has to discover it during the attack. Again, the IMN adversary provides an oracle to accomplish this. This is formalised in the lemma below.

Lemma 7 (IMN Hard $\Leftrightarrow$ IRN Hard for poly-sized $q^{\dim_{\mathbb{F}_q}(P/\langle G\rangle)}$). *Assume that $q(\lambda)^{\dim_{\mathbb{F}_q}(P/\langle G\rangle)}$ is $\mathrm{poly}(\lambda)$ sized for any $P \in [\mathbf{P}_\lambda]$ and $G \in \mathsf{GBGen}(1^\lambda, P, d)$. Then for any ppt adversary $\mathscr{A}$ against the IMN problem, there exists a ppt adversary $\mathscr{B}$ against the IRN problem such that*

$$\mathbf{Adv}^{\mathsf{imn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{A}}(\lambda) \leq \mathbf{Adv}^{\mathsf{irn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{B}}(\lambda).$$

Conversely, for any ppt adversary $\mathscr{B}$ against the IRN problem, there exists a ppt adversary $\mathscr{A}$ against the IMN problem s.t.

$$\mathbf{Adv}^{\mathsf{irn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{B}}(\lambda) = \mathbf{Adv}^{\mathsf{imn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathscr{A}}(\lambda),$$

if χ is efficiently distinguishable from the uniform distribution on $P/\mathscr{I}$.

Proof. The second claim holds as the adversary simply computes $r = f \mod \mathscr{I}$ and decides whether r is more likely to be from χ or from the uniform distribution in $P/\mathscr{I}$. To proof the first part of the theorem we construct an adversary $\mathscr{B}$ against IRN from adversary $\mathscr{A}$ against IMN in Algorithm 8. For the sake of compactness we omitted amplification in Algorithm 8. However, it is easy to see that we can amplify our confidence in the outputs of $\mathscr{A}$ (called in lines 15 and 26) by repeated calls to $\mathscr{A}$. We emphasise that this was not possible in the noise-free setting because of the limited number of samples allowed. Hence, in the noisy setting we can remove the $\mathrm{poly}(\lambda)$ exponent from advantage terms.

Algorithm 8 is correct. If $f - r \in \mathscr{I}$ we have $f - r = \sum_i h_i g_i$ for some $h_i \in P$. Hence, we have $f = \sum_i h_i g_i + r$. Furthermore, we have that r is minimal among all elements with

$f - r \in \mathscr{I}$ (cf. lines 4, 11 and 23). Finally, both calls to $\mathscr{A}$ (in lines 15 and 26) can be repeated to amplify the confidence in the result.

Algorithm 8 runs in polynomial time. By assumption

$$q(\lambda)^{\dim_{\mathbb{F}_q}(\mathscr{P}(\lambda)/\mathsf{GBGen}(\cdot))} = q(\lambda)^{|M|} \text{ is polynomial in } \lambda.$$

Furthermore, line 10 can only be executed logarithmically many times in λ. Also, Algorithm 8 will execute line 22 at most $|G| = \mathrm{poly}(\lambda)$ times, once for each $\mathrm{LM}(g)$ for $g \in G$. Hence the outer loop in line 7 is executed at most $q^{|M|} \cdot \log(\lambda) \cdot \mathrm{poly}(\lambda)$ times. If k is an upper bound on the number of calls that $\mathscr{A}$ makes to its **Sample** oracle, $\mathscr{B}$ will make at most $2 \cdot \mathrm{poly}(\lambda)/\varepsilon \cdot q^{|M|} \cdot \log(\lambda) \cdot \mathrm{poly}(\lambda) \cdot k$ calls to its **Sample** oracle. □

Hence GBN is equivalent to IRN and IRN is equivalent to IMN under some additional assumptions about the size $P/\mathscr{I}$. Finally, for $d = 1$ (but arbitrarily b) we show that if we can solve the GBN problem on average, then we can also solve it for worst-case instances. This is turn increases our confidence in hardness of the GBN problem.

Lemma 8 (Average-case to Worst-case). *Let $\mathscr{A}$ be a ppt adversary against $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),1,b,\chi}$. Then there exists a ppt adversary $\mathscr{B}$ which solves the Gröbner basis with Noise problem $\mathsf{GBN}_{\mathscr{P},G,1,b,\chi}$ 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:*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),1,b,\chi,\mathscr{A}}(\lambda) = \mathbf{Adv}^{\mathsf{gbn}}_{\mathscr{P},G,1,b,\chi,\mathscr{B}}(\lambda).$$

Proof. The proof is similar to the proof of [52, Lemma 3.2]. The difference is that we apply the transformation $L_{\mathbf{t}} : P \rightarrow P$ defined by $L_{\mathbf{t}}(f) := f(\mathbf{t})$ for any $\mathbf{t} := (\sum a_{0,i}x_i, \ldots, \sum a_{n-1,i}x_i)$ with randomly chosen $a_{i,j} \in \mathbb{F}_q$, such that the matrix $A = a_{i,j}$ has full rank. That is, we perform a random change of variables and hence re-randomise the secret G. Because A has full rank, this transformation is invertible and we can recover the original solution from the transformed secret by applying $\tilde{\mathbf{t}} := (\sum \tilde{a}_{0,i}x_i, \ldots, \sum \tilde{a}_{n-1,i}x_i)$ with $\tilde{a}_{i,j} = A^{-1}$. □

REMARK. The above proof strategy does not seem to extend to $d > 1$. This is because there are approximately q^{nd^n} secret keys compared to only $\prod_{i=0}^{n-1}(q^n - q^i) < q^{n^2}$ invertible maps $L_{\mathbf{t}}$. In other words, the maps $L_{\mathbf{t}}$ do not provide sufficient re-randomisation.

7.1 Hardness Assumptions and Justifications

In this subsection we investigate the hardness of the GBN, IRN, and IMN problems. We first consider the GBN problem and relate it to the well-established LWE problem [52]. Then, we discuss the relation between the GBN problem and various approximate GCD problems [57]. Third, we discuss the special case $q = 2$ by relating the GBN problem to the well-known $\mathsf{Max\text{-}SAT}$ problem. Finally, we consider known attacks against the GBN problem. We start by recalling the LWE problem.

Definition 18 (Learning with Errors (LWE) Problem). *The Learning with Errors problem is defined though game* $\mathsf{LWE}_{n,q,\chi}$ *shown in Figure 9. The advantage of a ppt algorithm* $\mathscr{A}$ *in solving* LWE *is*

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

```
Initialize($1^\lambda$):
begin
    $n \leftarrow n(\lambda)$;
    $s \leftarrow_\$ \mathbb{Z}_q^n$;
    return $(1^\lambda, n)$;
end
```

```
Sample():
begin
    $a \leftarrow_\$ \mathbb{Z}_q^n$;
    $e \leftarrow_\$ \chi$;
    $b \leftarrow e + \sum_i a_i s_i$; // < a, s > +e
    return $(a, b)$;
end
```

```
Finalize($s'$):
begin
    return $s = s'$;
end
```

Fig. 9. Game $\mathsf{LWE}_{n,q,\chi}$.

From the definition of LWE it is easy to see that GBN can be considered as a non-linear 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 9 (LWE Hard ⇒ GBN Hard for $d = 1, b = 1$). *Let q be a prime number. Then for any ppt adversary* $\mathscr{A}$ *against the* GBN *problem*[6] *with* $b = d = 1$, *there exists a ppt adversary* $\mathscr{B}$ *against the* LWE *problem such that*

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

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

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

□

In the noise-free setting we assume that solving systems of equations of degree greater than 1 is harder than solving those of degree 1. More generally, we assume that equations of degree $b > b'$ are harder to solve than those of degree b'. Intuitively, equations of degree b' can be seen as those of degree b where the coefficients of higher

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

degree monomials are set to zero. However, formalising this intuition for an adversary which expects uniformly distributed equations of degree b seems futile since producing such equations is equivalent to solving the system by Theorem 7.

In the noisy setting this equivalence (i.e., Theorem 7) between sampling and solving no longer holds. However, we still need to deal with the distribution of noise. One strategy to show that difficulty increases with the degree parameter b is to allow for an increase of the noise level in the samples. We formalise this below.

Lemma 10 (GBN **Hard for** $2b \Rightarrow \mathsf{GBN}$ **Hard for** b**).** *Let* $N = \binom{n+b}{b}$*. For any ppt adversary* $\mathscr{A}$ *against the* GBN *problem at degree b with noise* $\chi_{\alpha,q}$*, there exists a ppt adversary* $\mathscr{B}$ *against the* GBN *problem at degree* $2b$ *with noise* $\chi_{\sqrt{N}\alpha^2 q,q}$ *such that*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi_{\alpha,q},\mathscr{A}}(\lambda) = \mathbf{Adv}^{\mathsf{gbn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,2b,\chi_{\sqrt{N}\alpha^2 q,q},\mathscr{B}}(\lambda).$$

Proof (Sketch). Let $f_0 = \sum_j h_{0,j} g_j + e_0$ and $f_1 = \sum_i h_{1,i} g_i + e_1$ be samples from $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi_{\alpha,q}}$. We have

$$\begin{aligned}
f_0 \cdot f_1 &= (\sum_j h_{0,j} g_j + e_0) \cdot (\sum_i h_{1,i} g_i + e_1) \\
&= (\sum_j h_{0,j} g_j) \cdot (\sum_i h_{1,i} g_i + e_1) + e_0 \cdot (\sum_i h_{1,i} g_i) + e_0 e_1 \\
&= \sum_j (\sum_i h_{1,i} g_i + e_1) h_{0,j} g_j + e_0 \cdot (\sum_i h_{1,i} g_i) + e_0 e_1 \\
&= \sum_j (\sum_i (h_{1,i} h_{0,j} g_i + e_1 h_{0,j}) g_j) + \sum_i e_0 h_{1,i} g_i + e_0 e_1 \\
&= \sum_j \tilde{h}_j g_j + e_0 e_1 \text{ for some } \tilde{h}_j.
\end{aligned}$$

It follows that $f_{0,1} := f_0 \cdot f_1$ is a polynomial of degree $2b$ whose error term $e_0 e_1$ follows a discretised Normal product distribution with mean 0 and standard deviation $\alpha^2 q^2$. More generally, let $f_{i,j} := f_i \cdot f_j$, with f_i, f_j samples from $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,2b,\chi_{\alpha,q}}$, be a product of polynomials. The elements $f_{i,j}$ are not random elements of degree $2b$ in $\mathscr{I}$. In particular, all $f_{i,j}$ factor into at least two polynomials. Let h be a sample from $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,2b,\chi_{\alpha,q}}$. To "destroy" this algebraic structure we may consider the sum

$$\hat{f} = \sum_{i=0}^{m-1} f_{2i,2i+1} + h \text{ for some } m \in \mathbb{N}.$$

The addition of h – which has a small error term – ensures that $\hat{f}$ is indeed an element of $\mathscr{I}$ and not just $\mathscr{I}^2$. In order to estimate the required magnitude of m to render $\hat{f}$ indistinguishable from uniform $\in \mathscr{I}$ at degree $2b$, we write $f_j = \sum c_{ik} m_k$ for $c_{ik} \in \mathbb{F}_q$ and m_k monomials of degree $\leq b$. Hence, we can write $f_i \cdot f_j = \sum_{k=0}^{N-1} c_{ik} m_k f_i$. We now apply the leftover hash lemma for each value of k independently. That is, we consider the affine group $\mathbb{G} = \mathbb{F}_q^N$ of coefficient vectors of polynomials $m_k \cdot f_i$ and apply a variant of a special case of the leftover hash lemma [43], i.e., that $\sum_{i=0}^{\ell-1} b_i g_i$ has statistical distance bounded from above by $\sqrt{|\mathbb{G}|/q^\ell} = \sqrt{q \cdot N/q^\ell}$ to the uniform distribution for

$b_i \in [-q/2, q/2)$ and $g_i \in \mathbb{G}$. For $\ell = N$ this magnitude is exponentially small. Finally, we note that the parallel applications of the leftover hash lemma are independent because of the independence of c_{ik}. Hence, $\hat{f} = \sum_{i=0}^{N-1} f_{2i,2i+1}$ is a indistinguishable from a random element in $\mathscr{I}$ of degree $2b$.

Finally, we need to consider the distribution on the noise. By the Central Limit Theorem $\sum_{i=0}^{N-1} e_{2i,2i+1}$ converges to a discretised normal distribution centred at zero and with standard deviation of $\sqrt{N}\alpha^2 q^2$. More precisely, the value $\sum_{i=0}^{N-1} |e_{2i,2i+1}|$ is smaller than samples from $\chi_{\sqrt{N}\alpha^2 q,q}$ with very high probability. Hence, we may add noise from the appropriate noise distribution χ' (i.e., the difference of the two distributions) to $\hat{f}$ such that $\hat{f}$ has an error term distributed close to $\chi_{\sqrt{N}\alpha^2 q,q}$.

This means that $\hat{f} = \sum_{i=0}^{N-1} f_{2i,2i+1} + h + e$ for $e \leftarrow_\$ \chi'$ is a random-looking sample for $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,2b,\chi_{\sqrt{N}\alpha^2 q,q}}$. It follows from the definition of ideals that the Gröbner basis for polynomials $\sum f_{2i,2i+1} + h$ is the same as that for f_{2i}, f_{2i+1}. From this, it is easy to see that the adversary $\mathscr{B}$ will return a Gröbner basis which is valid for samples presented to $\mathscr{A}$. □

On the other hand, if we do not want to tolerate this noise increase, another strategy would be to consider a "sparse" variant of $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi_{\alpha,q}}$ where **Sample** returns samples whose higher-degree terms only involve a subset of $\log(\lambda)$ variables similar to [15]. This strategy is pursued in Appendix A.

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. [57]), this problem has not yet received much attention, and hence it is unclear under which parameters it is hard. However, as mentioned in Section 3, the notion of a Gröbner basis can been extended to $\mathbb{Z}[x_0, \ldots, x_{n-1}]$, which 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 CASE $q = 2$. Recall that if $b = d = 1$ we have an equivalence with the LWE problem (or the well-known problem of learning parity with noise (LPN) 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 $\mathscr{A}$ 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-SAT instance (more precisely Partial Max-Sat) by running an ANF to CNF conversion algorithm [6].

KNOWN ATTACKS. Finally, we consider known attacks to understand the difficulty of the GBN problem. Recall, that if $b = 1$ Lemma 9 states that we can solve the LWE problem if we can solve the GBN problem. The converse also applies. Indeed, 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 non-linear systems of equations in the noise-free setting. Let $N = \binom{n+b}{b}$ be the number of monomials up to degree b. Let $\mathscr{M} : P \longrightarrow \mathbb{F}_q^N$ be a function which maps polynomials in P to vectors in $\mathbb{F}_q^N$ by assigning the i-th component of the image vector the coefficient of the i-th 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 the GBN **Sample** oracle to retrieve f, compute $v = \mathscr{M}(f)$ and return (a,b) with $a = (v_{N-1},\ldots,v_1)$ and $b = -v_0$. When the LWE adversary queries **Finalize** on s, query the GBN **Finalize** with $[x_0 - s_0,\ldots,x_{n-1} - s_{n-1}]$. Correctness follows from the correctness of linearisation in the noise-free setting [4]. 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 $\alpha \geq \frac{3}{2}\cdot \max\left(\frac{1}{q}, 2^{-2\sqrt{N\log q \log d}}\right)$ for an appropriate choice of δ which is the quality of the approximation for the shortest vector problem. With current lattice algorithms $\delta = 1.01$ is hard and 1.005 infeasible [49].

Perhaps the most interesting attack on LWE from the perspective of this work is that due to Arora and Ge [4] which reduces the problem of solving linear systems with noise to the problem of solving (structured) non-linear noise-free systems. We may apply this technique directly to GBN, i.e., without going to 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 hence a noise-free system) must be imposed.

We conclude this section by explicitly stating our hardness assumption.

Definition 19 (GBN/IRN/IMN **Assumptions).** *Let $b,d \in \mathbb{N}$ with $b \geq d \geq 1$. Let $\mathscr{P}$ 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. *$n \geq \sqrt[b]{\lambda}$;*
2. *$(\alpha q)^{nd^n} \approx 2^\lambda$ so exhaustive search over the noise or the secret key space is ruled out;*
3. *$\alpha q \geq 8$ as suggested in [47]; and*
4. *For $N := \binom{n+b}{b}$, and $\delta := 1.005$ we have $\alpha \geq \frac{3}{2}\cdot \max\left\{\frac{1}{q}, 2^{-2\sqrt{N\log q \log \delta}}\right\}$, and hence the best known attacks against the* LWE *problem are ruled out [49, 54].*

The advantage of any ppt algorithm in solving the GBN, IRN, *and* IMN *problems with the above parameters is negligible as a function of λ.*

8 Polly Cracker with Noise

In this section we present a fully IND-CPA secure Polly Cracker-style symmetric encryption scheme. Our parameterised scheme, $\mathscr{SPCN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$, is shown in Figure 10. Here we represent elements in $\mathbb{F}_q$ as integers in the interval $(-\lfloor\frac{q}{2}\rfloor,\lfloor\frac{q}{2}\rfloor]$. This representation 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 $\gcd(2,q) = 1$. This condition is needed for the correctness and the security of our scheme. The message space is $\mathbb{F}_2$ (although we remark that this can be generalised to other small fields).

CORRECTNESS OF EVALUATION. We restrict our attention to $d = 1$. This greatly simplifies the discussion of correctness below due to a simpler notion of “size” of the noise. That is, we define the size of the noise as $\log_2$ of the distance to zero over the integers. Addition and multiplication of the two ciphertexts $c_0 = \sum h_{0,j} g_j + 2e_0 + m_0$ and $c_1 = \sum h_{1,j} g_j + 2e_1 + m_1$ are given by

$$\begin{aligned}
c_0 + c_1 &= \sum h_{0,j} g_j + 2e_0 + m_0 + \sum h_{1,j} g_j + 2e_1 + m_1 \\
&= \sum (h_{0,j} + h_{1,j}) g_j + 2(e_0 + e_1) + (m_0 + m_1) \\
c_0 \cdot c_1 &= (\sum h_{0,j} g_j + 2e_0 + m_0) \cdot (\sum h_{1,j} g_j + 2e_1 + m_1) \\
&= (\sum h_{0,j} g_j) \cdot (\sum h_{1,j} g_j + 2e_1 + m_1) \\
&\quad + (2e_0 + m_0) \cdot (\sum h_{1,j} g_j) \\
&\quad + (4e_0 e_1 + 2e_0 m_1 + 2e_1 m_0 + m_0 m_1) \\
&= \sum \tilde{h}_j g_j + 2(2e_0 e_1 + e_0 m_1 + e_1 m_0) + m_0 m_1 \text{ for some } \tilde{h}_j
\end{aligned}$$

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.

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

$\mathsf{Enc}(m, \mathsf{SK})$:
begin
    $f \leftarrow_\$ P_{=b}$;
    $f' \leftarrow f \mod G$;
    $f \leftarrow f - f'$;
    $e \leftarrow_\$ \chi$;
    $c \leftarrow f + 2e + m$;
    return $c$;
end

$\mathsf{Dec}(c, \mathsf{SK})$:
begin
    $m' \leftarrow c \mod G$;
    $m \leftarrow m' \mod 2$;
end

$\mathsf{Eval}(c_0, \ldots, c_{t-1}, C, \mathsf{PK})$:
begin
    apply Add and Mul gates
    of $C$ over $P$;
    return the result;
end
```

Fig. 10. The Symmetric Polly Cracker with Noise scheme $\mathscr{SPCN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$.

PERMITTED CIRCUITS. Circuits composed of Add and Mul 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., the coefficients are in $\mathbb{F}_q$). In order to define the set of permitted circuits (which will be parameterised by $\alpha > 0$) we first embed the Boolean polynomials into the ring of polynomials over $\mathbb{Z}$. For $\chi_{\alpha,q}$ we have that the probability of the noise being larger than $k\alpha q$ is $< \exp(-k^2/2)$. We now say that a circuit is valid if for any $(s_0, \ldots, s_{t-1})$ with $s_i \leq t\alpha q$ we have that the outputs are less than q for some parameter t. This restriction ensures that no overflows occurs when polynomials are evaluated over $\mathbb{F}_q$. Section 10 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 the bit-size of the the noise by approximately $\log(5e_0e_1)$ bits. Section 9 contains a discussion on how to trade ciphertext size with noise. The theorem below states the security properties of the above scheme.

Theorem 8. *Let $\mathscr{A}$ be a ppt adversary against the* IND-CPA *security of the scheme in Figure 10. Then there exists a ppt adversary $\mathscr{B}$ against the* IMN *problem such that for all $\lambda \in \mathbb{N}$ we have*

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

Proof. We construct an algorithm $\mathscr{B}$ against the IMN problem based on $\mathscr{A}$ attacking the IND-CPA security of the scheme in Algorithm 9.

Now if the sample returned from the **Challenge** oracle to $\mathscr{B}$ is 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 $\mathscr{A}$ is run in an environment which is identical to the IND-CPA game. Note that since $\gcd(2,q) = 1$, multiplications by 2 at lines 6 and 10 do not affect the distribution of f. Hence in this case the probability that $c = c'$ is equal to the probability that $\mathscr{A}$ wins the IND-CPA game. The theorem follows. □

The above theorem together with the recent results in [53] which establish the equivalence of symmetric and asymmetric homomorphic encryption schemes leads to the first provably secure public-key encryption scheme from assumptions related to Gröbner bases for random systems. This provides a positive answer to the challenges raised by Barkee et al. [10] (and later also by Gentry [37]). We note here that the transformation – as briefly described in Section 6 – only use the additive features of the scheme and does not require full homomorphicity.

9 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 $2b$ to polynomials of degree b by performing a 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 $K_{A\to B}$.

We discuss how one can achieve the above functionality for our scheme.[7] 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 $\mathscr{I}_A \subset P$ and $\mathscr{I}_B \subset P$. Finally, suppose $P/\mathscr{I}_A = P/\mathscr{I}_B$. To re-encrypt a ciphertext intended for G_A under key G_B we generate the re-encryption key $G_{A\to B}$ as in Algorithm 10. This key will then be used in Algorithm 11, which is the actual re-encryption algorithm.

The central ideal behind these algorithms is the equivalence between different representations of elements in $P/\mathscr{I}$. While for the most part of this work we identify elements in $P/\mathscr{I}$ with elements $f \mod \mathscr{I}$, Algorithms 10 and 11 make use of different representations of elements in $P/\mathscr{I}$. 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 $P/\mathscr{I}_A$ since f

[7] Since the construction only uses additions, this feature also applies to the LWE-based encryption scheme as previously observed in `http://xagawa.net/pdf/20100120_SCIS_PRE.pdf`

$\mod G_A = r$, i.e., $x \mod G_A = -1$. Hence, if we are interested in $P/\mathscr{I}_A$ (our messages live in $P/\mathscr{I}$) 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., $m+2e = f \mod \mathscr{I}_A$, as $\sum(c_i m_i \mod \mathscr{I}_A)$. Furthermore, since $P/\mathscr{I}_A = P/\mathscr{I}_B$, we may encrypt the encoded message $m+2e$ for G_B by computing

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

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

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 11. All elements in $G_{A\to B}$ are of degree at most b. Hence, the degree of the output of Algorithm 11 is at most b. Furthermore, given a polynomial of degree $b' = 2b$, this algorithm performs at most $\log q \cdot \binom{n+b'}{b'}$ additions of polynomials. If e is the maximal noise in any of the polynomials in $G_{A\to B}$, reducing the degree from $2b$ to b adds a noise of at most e^2. On top of that, Algorithm 11 will "copy" the noise from f, and hence, it does not reduce it: we are trading degree for noise.

To consider security, we first discuss re-encryption under the same key, i.e., $G_A = G_B$. If $b' = b$, the key $G_{A\to A}$ can be constructed publicly 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 $(f \mod \mathscr{I}) = 2e$ for some small error term e it holds that $f + 2^j \cdot m \mod \mathscr{I} = (2^j \cdot m \mod \mathscr{I}) + 2e$. Hence, $G_{A\to A}$ is a correct re-encryption key which can be generated given only access to encryptions zero, i.e., no additional information is leaked. This implies limited key-dependent message security for our scheme in the standard model; limited in the sense that only the least significant bit of the constant term of each Gröbner basis element is encrypted.

However, this argument does not go through for $b' > b$. While it is easy to construct elements f which satisfy $f \mod \mathscr{I} \approx 2^j \cdot m \mod \mathscr{I}$ for m a monomial of degree $> b$ for anyone with access to encryptions of zero, it is not easy to produce such elements with degree $\leq b$ and small noise.

Yet, for $G_A \to G_B$ with $G_A \neq G_B$ security of this re-encryption can be shown under the IMN assumption. That is, any adversary breaking the IND-CPA security of this game with access to the re-encryption key $G_{A\to B}$ can be turned into an adversary breaking against the IMN problem. A full proof for this is presented for the special case of LWE in [18] where this technique was independently proposed.

10 Parameters

In this section we give concrete suggestions for various parameters that are involved in our scheme. These suggestions are based on the currently best known attacks – instead of theoretical hardness results – in order to stimulate research on the concrete hardness of our underlying assumptions.

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 1% decryption error probability, we must have $\Pr[|e^{\mu}| \geq q/2] < 1/100$. Hence (cf. Section 8) we need to ensure that

$$\exp(k^2/2) > 100 \text{ and } k(\alpha q) < 1/2 \cdot \sqrt[\mu]{q}.$$

Another set of restrictions comes from the conditions stated in our intractability assumption in Definition 19. For this, we make the somewhat arbitrary choice of $b = 2$ and denote by $N = \binom{n+2}{2}$ the number of monomials in a fresh ciphertext. 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

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

This allows us to simplify the condition needed to ensure the hardness of the LWE problem in Definition 19 to:

$$\lambda^{(\mu+\frac{1}{2})} \log^2(\lambda) \leq \frac{2}{3} \cdot 2^{2\sqrt{\binom{n+2}{2}(\mu+2)\log\lambda \log 1.005}}.$$

Based on these inequalities, we give example choice for parameters in Table 1. In this table we have also included whether the theoretical bound $\alpha q > 2\sqrt{N}$ is satisfied. This inequality allows quantum reductions between the LWE problem and certain lattice-based problems to go through.

11 Reference Implementation

We implemented our scheme using the Sage mathematics software [56].[8] 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.

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). Pooya Farshim was funded in part by the US Army Research laboratory and the UK Ministry of Defense and was accomplished

[8] `https://bitbucket.org/malb/algebraic_attacks/src/e70e02bb456d/noisy-polly-cracker.py`

under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Algorithm 8: IRN adversary $\mathscr{B}$ from IMN adversary $\mathscr{A}$

```
begin
    $\mathscr{B}$ receives $(1^\lambda, P)$;
    $M, G \leftarrow \varnothing, \varnothing$;
    **Gen**$()$ := a function taking $n \geq 0$ and returning a of generator producing $v \in \mathbb{F}_q^n$ for
    lex. sorted;
    $gen, c_m \leftarrow$ **Gen**$(|M|), 1$;
    query **Challenge**$()$ to get $f$;
    while $True$ do
        $v \leftarrow gen()$;
        if $v = \perp$ then
            $M \leftarrow M \cup \{c_m\}$;
            $c_m \leftarrow$ the min. monomial $m$ with $m \notin M$ and $m \notin \langle \mathrm{LM}(G) \rangle$;
            $gen \leftarrow$ **Gen**$(|M|)$;
            $v \leftarrow gen()$;
        $r \leftarrow c_m + \sum v_i m_i$ for $0 \leq i < |M|, m_i \in M$;
        run $\mathscr{A}(1^\lambda, P)$ as follows:
        if $\mathscr{A}$ *queries* **Sample**$()$ then
            query **Sample**$()$ to get $h$ and return $h$;
        if $\mathscr{A}$ *queries* **Challenge**$()$ then
            query **Sample**$()$ to get $h$; return $h + r$;
        if $\mathscr{A}$ *calls* **Finalize**$(c')$ then
            if $c' = 0$ then
                $G \leftarrow G \cup r$;
                $c_m \leftarrow$ the min. monomial $m$ with $m \notin M$ and $m \notin \langle \mathrm{LM}(G) \rangle$;
                $gen \leftarrow$ **Gen**$(|M|)$;
            else
                run $\mathscr{A}(1^\lambda, P)$ as follows:
                if $\mathscr{A}$ *queries* **Sample**$()$ then
                    query **Sample**$()$ to get $h$;
                    return $h$;
                if $\mathscr{A}$ *queries* **Challenge**$()$ then
                    return $f - r$;
                if $\mathscr{A}$ *calls* **Finalize**$(b)$ then
                    if $b = 0$ then
                        call **Finalize**$(r)$;
end
```

Algorithm 9: IMN adversary $\mathscr{B}$ from IND-CPA adversary $\mathscr{A}$

```
begin
    $\mathscr{B}$ receives $(1^\lambda, P)$;
    run $\mathscr{A}(1^\lambda, P)$ as follows;
    if $\mathscr{A}$ queries Encrypt$(m)$ then
        query Sample$()$ to get $f$;
        return $2f + m$ ;
    if $\mathscr{A}$ queries Left-Right$(m_0, m_1)$ then
        $c \leftarrow_\$ \{0,1\}$;
        query Challenge$()$ to get $f$;
        return $2f + m_c$ ;
    if $\mathscr{A}$ calls Finalize$(c')$ then
        call Finalize$(c = c')$;
end
```

Algorithm 10: Generating the re-encryption key

```
Input: $G_A$ – a Gröbner basis
Input: $f_0, \dots, f_{m-1}$ – encryptions of zero under $G_B$
Input: $b'$ – a bound on the degree of polynomials
begin
    $G_{A \to B} \leftarrow \varnothing$;
    for $m \in M_{\leq b'}$ do
        $m' \leftarrow m \mod G_A$;
        for $0 \leq j < \lceil \log_2(q/2) \rceil$ do
            $s \leftarrow_\$$ a sparse subset of $[0, \dots, m-1]$;
            $f \leftarrow \sum_s f_s$;
            $G_{A \to B}[2^j \cdot m] \leftarrow f + 2^j \cdot m'$;
    return $G_{A \to B}$;
end
```

Algorithm 11: 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
    $f' \leftarrow 0$;
    for $m \in f$ do
        $c \leftarrow$ the coefficient in $f$ of $m$ represented as an integer in $(-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor]$;
        $m' \leftarrow 0$;
        for $0 \leq j < \lceil \log_2(q/2) \rceil$ do
            if the $j$-th bit of $|c|$ is set then
                $m' \leftarrow m' + G_{A \to B}[2^j \cdot m]$;
        if $c < 0$ then
            $m' \leftarrow -1 \cdot m'$;
        $f' \leftarrow f' + m'$;
    return $f'$;
end
```

λ	μ	n	N	α	q	$\alpha q > 2\sqrt{N}$	ciphertext size
40	1	15	136	0.00558254200346408	1999	False	$\approx$ 0.2 kbytes
40	2	20	231	0.000139563550086602	92893	False	$\approx$ 0.5 kbytes
40	3	24	325	3.48908875216505e-6	3842401	False	$\approx$ 0.9 kbytes
80	1	16	153	0.00279740858078175	12227	True	$\approx$ 0.3 kbytes
80	2	21	253	0.0000349676072597719	594397	False	$\approx$ 0.6 kbytes
80	3	26	378	4.37095090747149e-7	54771113	False	$\approx$ 1.2 kbytes
128	1	23	300	0.00180384382955752	29501	True	$\approx$ 0.6 kbytes
128	2	22	276	0.0000140925299184181	4025909	True	$\approx$ 0.8 kbytes
128	3	27	406	1.10097889987642e-7	456626039	True	$\approx$ 1.4 kbytes
256	1	41	903	0.000976562500000000	81971	True	$\approx$ 1.6 kbytes
256	2	38	780	3.81469726562500e-6	28191413	True	$\approx$ 2.5 kbytes
256	3	42	946	1.49011611938477e-8	5005092413	True	$\approx$ 3.2 kbytes
512	1	68	2415	0.000545607084248879	347539	True	$\approx$ 5.2 kbytes
512	2	65	2211	1.06563883642359e-6	239518691	True	$\approx$ 8.2 kbytes
512	3	69	2485	2.08132585238983e-9	85332320813	True	$\approx$ 11.8 kbytes

Table 1. Example parameter choices for $b = 2$, $k = \sqrt{2\log(100)}$

References

1. Martin Albrecht and John Perry. F4/5. *CoRR*, abs/1006.4933v2, 2010.
2. 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.
3. David Arditti, Côme Berbain, Olivier Billet, Henri Gilbert, and Jacques Patarin. QUAD: Overview and recent developments. In Eli Biham, Helena Handschuh, Stefan Lucks, and Vincent Rijmen, editors, *Symmetric Cryptography*, volume 07021 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.
4. Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP 2011*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415, Berlin, Heidelberg, New York, 2011. Springer Verlag.
5. Gwenole Ars. *Applications des bases de Gröbner à la cryptographie*. PhD thesis, Université de Rennes I, 2005.
6. 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`.
7. Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Paris VI, 2004.
8. 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.
9. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. 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.
10. 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.
11. Dave Bayer and Mike Stillman. On the complexity of computing syzygies. *Computational Aspects of Commutative Algebra*, page 1–13, 1988.
12. Thomas Becker and Volker Weispfenning. *Gröbner Bases - A Computational Approach to Commutative Algebra*. Springer Verlag, Berlin, Heidelberg, New York, 1991.
13. 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.
14. Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A multivariate stream cipher with provable security. *J. Symb. Comput.*, 44(12):1703–1723, 2009.
15. Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A practical stream cipher with provable security. In *Advances in Cryptography - EUROCRYPT 2006*, Lecture Notes in Computer Science, pages 109–128, Berlin, Heidelberg, New York, 2006. Springer Verlag.
16. Olivier Billet and Jintai Ding. Overview of cryptanalysis techniques in multivariate public key cryptography. 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.

17. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277, 2011. `http://eprint.iacr.org/`.
18. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. To appear in FOCS 2011, 2011.
19. Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. PhD thesis, Universität Innsbruck, 1965.
20. Bruno Buchberger. Gröbner bases: an algorithmic method in polynomial ideal theory. In N. K. Bose, editor, *Multidimensional Systems Theory*. D. Reidel Publishing Company, 1985.
21. 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.
22. Stanislav Bulygin. Chosen-ciphertext attack on noncommutative Polly Cracker. *CoRR*, abs/cs/0508015, 2005.
23. Massimo Caboara, Fabrizio Caruso, and Carlo Traverso. Lattice Polly Cracker cryptosystems. *Journal of Symbolic Computation*, 46:534–549, May 2011.
24. Jean-Sebastien Coron, David Naccache, and Mehdi Tibouchi. Optimization of fully homomorphic encryption. Cryptology ePrint Archive, Report 2011/440, 2011. `http://eprint.iacr.org/`.
25. 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.
26. David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, Berlin, Heidelberg, New York, 3rd edition, 2005.
27. 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.
28. 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.
29. Jintai Ding and Bo-Yin Yang. Multivariate public key cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 193–234. Springer Verlag, Berlin, Heidelberg, New York, 2009.
30. 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.
31. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner basis (F4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.
32. 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.
33. 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.
34. 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.

35. 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.
36. 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.
37. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. Available at `http://crypto.stanford.edu/craig`.
38. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC 09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178, 2009.
39. Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. Cryptology ePrint Archive, Report 2011/279, 2011. `http://eprint.iacr.org/`.
40. 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.
41. Aline Gouget and Jacques Patarin. Probabilistic multivariate cryptography. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 2006*, volume 4341 of *Lecture Notes in Computer Science*, pages 1–18, Berlin, Heidelberg, New York, 2006. Springer Verlag.
42. Johan Håstad, Steven Phillips, and Shmuel Safra. A well-characterized approximation problem. *Inf. Process. Lett.*, 47:301–305, October 1993.
43. Russell Impagliazzo and David Zuckerman. How to recycle random bits. In *Proceedings of 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, page 248–253, 1989.
44. Neal Koblitz, Alfred J. Menezes, Yi-Hong Wu, and Robert J. Zuccherato. *Algebraic aspects of cryptography*. Springer Verlag, Berlin, Heidelberg, New York, 1998.
45. Kristin Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? Cryptology ePrint Archive, Report 2011/405, 2011. `http://eprint.iacr.org/`.
46. 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.
47. 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.
48. 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.
49. 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.
50. 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.
51. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56:34:1–34:40, September 2009.

52. Oded Regev. The learning with errors problem. In *IEEE Conference on Computational Complexity 2010*, pages 191–204, 2010.
53. Ron Rothblum. Homomorphic encryption: from private-key to public-key. In *Theory of Cryptography – TCC 2011*, volume 6597 of *Lecture Notes in Computer Science*, pages 219–234, Berlin, Heidelberg, New York, 2011. Springer Verlag.
54. 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`.
55. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443, Berlin, Heidelberg, New York, 2010. Springer Verlag.
56. William Stein et al. *SAGE Mathematics Software*. The Sage Development Team (Version 4.7.0), 2011. Available at `http://www.sagemath.org`.
57. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43, 2010.
58. Christopher Wolf. *Multivariate quadratic polynomials in public key cryptography*. Univ. Leuven Heverlee, 2005.
59. 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.

A Alternative Strategy for Hardness of Higher Degrees

Let V be a subset of $\{x_0, \ldots, x_{n-1}\}$. We denote by $\mathsf{GBN}'_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b',b,\chi,V}$ a game which is similar to the $\mathsf{GBN}_{\mathscr{P},\mathsf{GBGen}(\cdot),d,b,\chi}$ game except that the **Sample** procedure is replaced by the **Sample**$'$ procedure given in Figure 11.

```
proc. Sample'():
begin
    $f \leftarrow_{\$} P_{\leq b'}$;
    $f' \leftarrow_{\$} P_{\leq b}$ restricted to the variables in $V$;
    $e \leftarrow_{\$} \chi$;
    $f \leftarrow f + f' - (f + f' \mod G) + e$;
    return $f$;
end
```

Fig. 11. Procedure **Sample**$'$ returning sparse samples.

Now, if $|V| = \log \lambda$ we can exhaustively search for a configuration which will satisfy these terms. This is formalised in the next lemma for the case $d = 1$ but for any χ.

Lemma 11 (GBN **Hard for** $b' \Rightarrow \mathsf{GBN}'$ **Hard for** $b > b'$ **if** $|V| = \log(\lambda)$**).** *For any ppt adversary* $\mathscr{A}$ *against* GBN' *at degree* $b > b'$*, there exists a ppt adversary* $\mathscr{B}$ *against*

GBN *at degree* b' *such that*

$$\mathbf{Adv}^{\mathsf{gbn}'}_{\mathscr{P},\mathsf{GBGen}(\cdot),1,b',b,\chi,V,\mathscr{A}}(\lambda) = \mathbf{Adv}^{\mathsf{gbn}}_{\mathscr{P},\mathsf{GBGen}(\cdot),1,b',\chi,\mathscr{B}}(\lambda),$$

where $|V| = \log(\lambda)$.

Proof. We construct a GBN adversary $\mathscr{B}$ at degree b' from a GBN$'$ adversary $\mathscr{A}$ at degree $b > b'$ in Algorithm 12. When $\mathscr{A}$ queries **Finalize** on $G \neq \{1\}$ this means that our guess v was correct and that the actual solution agrees with our guess. Thus, G is the Gröbner basis we are looking for. Since $|V| = \log(\lambda)$ we have that $\mathbb{F}_q^{|V|}$ is poly(λ) and the outer loop is repeated poly(λ) times. Hence, Algorithm 12 only uses resources polynomial in λ. □

Algorithm 12: GBN adversary $\mathscr{B}$ for b' from GBN$'$ adversary $\mathscr{A}$ for $b > b'$

```
begin
    $\mathscr{B}$ receives $(1^\lambda, P)$;
    replace $b'$ by $b$ and run $\mathscr{A}(1^\lambda, P)$ as follows;
    $|V| \leftarrow$ log of the number of variables in $P$;
    $V \leftarrow_\$ n$ variables from $P$;
    for $v \in \mathbb{F}_q^{|V|}$ do
        $I_v \leftarrow \{V_0 - v_0, \dots, V_{|V|-1} - v_{|V|-1}\}$;
        run $\mathscr{A}(1^\lambda, P)$ as follows:
        if $\mathscr{A}$ queries Sample$'()$ then
            query Sample$()$ to get $f$;
            $f' \leftarrow_\$ P_{\leq d}$ restricted to the variables $V$;
            $f' \leftarrow f' - (f' \mod I_v)$;
            return $f + f'$;
        if $\mathscr{A}$ queries Challenge$()$ then
            query Challenge$()$ to get $f$;
            $f' \leftarrow_\$ P_{\leq d}$ restricted to the variables $V$;
            $f' \leftarrow f' - (f' \mod I_v)$;
            return $f + f'$;
        if $\mathscr{A}$ calls Finalize$(G)$ then
            if $G \neq \{1\}$ then call Finalize$(G)$;
end
```