

On Black-Box Ring Extraction and Integer Factorization

Kristina Altmann, Tibor Jager, and Andy Rupp

Horst Görtz Institute for IT-Security
Ruhr-University Bochum
{kristina.altmann, tiber.jager}@nds.rub.de, arupp@crypto.rub.de

Abstract. The black-box extraction problem over rings has (at least) two important interpretations in the context of cryptography: An efficient algorithm for this problem implies (i) the equivalence of computing discrete logarithms and solving the Diffie-Hellman problem and (ii) the inexistence of secure ring-homomorphic encryption schemes.

For the special case of a finite field, Boneh/Lipton [BL96] and Maurer/Raub [MR07] show that there exist algorithms solving the black-box extraction problem in subexponential time. It is unknown whether there exist more efficient algorithms.

In this work, we consider the black-box extraction problem over finite rings of characteristic n , where n has at least two different prime factors. We provide a polynomial-time reduction from factoring n to the black-box extraction problem for integer rings and (multivariate) polynomial rings, assuming that computations can be performed efficiently in the given ring. Under the factoring assumption, this implies the inexistence of efficient generic reductions from computing discrete logarithms to the Diffie-Hellman problem on the one side, and might be an indicator that secure ring-homomorphic encryption schemes exist on the other side.

1 Introduction

Informally speaking, the black-box extraction problem over an algebraic structure A (like a group, ring, or a field) can be described as follows: Given an explicit representation of A (e.g., the cyclic group $(\mathbb{Z}_n, +)$ with the canonical binary representation of elements) as well as access to a black-box resembling the structure of A and hiding an element $x \in A$, the challenge is to recover the explicit representation of x . Algorithms that work on the black-box representation of an algebraic structure and thus on *any* concrete representation are called *generic* or *black-box* algorithms.

The black-box extraction problem has been studied in various variants and contexts, e.g., see [Nec94, Sho97, Mau05, BL96, MR07]. The case where the algebraic structure is a *cyclic group* (with given representation $(\mathbb{Z}_n, +)$), and the extraction problem is better known as the discrete logarithm problem, was considered by Nechaev [Nec94] and Shoup [Sho97]. They showed that the expected running time of any generic algorithm for this problem is $\Omega(\sqrt{p})$, where p is the largest prime factor of the group order n . Here, the integer n as well as its factorization is assumed to be publicly known. The provided bound essentially matches the running time of well-known generic algorithms for the discrete logarithm problem like the Pollard-Rho [Pol78] algorithm.

Boneh and Lipton [BL96] considered the black-box extraction problem over *prime fields* \mathbb{F}_p . Based on a result due to Maurer [Mau94] they developed an algorithm solving the problem in subexponential time (in $\log p$). Maurer and Raub [MR07] augmented this result to finite *extension fields* \mathbb{F}_{p^k} by providing an efficient reduction from the black-box extraction problem over \mathbb{F}_{p^k} to the black-box extraction problem over \mathbb{F}_p . Currently, it is unknown whether there exist more efficient algorithms for black-box extraction over fields.

In this paper we address the case where the underlying algebraic structure is a finite *commutative ring with unity*. More precisely, we consider the integer ring \mathbb{Z}_n as well as (multivariate) polynomial rings of characteristic n , where n is the product of at least two different primes. We provide an efficient reduction from computing a non-trivial factor of n to the black-box extraction problem

for virtually any such ring where computations (i.e., applying the ring operations $+$ and \cdot , equality tests and random sampling of elements) can be done efficiently. The black-box extraction problem over fields/rings has at least two important applications in cryptography, namely in the context of secure homomorphic encryption and regarding the equivalence of the discrete logarithm (DL) problem and the Diffie-Hellman (DH) problem. For $(\mathbb{Z}_n, +, \cdot)$ it can be interpreted as the problem of solving the DL problem given access to an oracle for the DH problem: $(\mathbb{Z}_n, +)$ forms a cyclic additive group. The black-box provides access to the common operations on this group as well as to the additional operation “ \cdot ”. This extra operation can be interpreted as an oracle solving the Diffie-Hellman problem in the group $(\mathbb{Z}_n, +)$. Hence, an efficient algorithm for the black-box extraction problem over $(\mathbb{Z}_n, +, \cdot)$ would correspond to an efficient *generic* reduction from computing discrete logarithms to solving the Diffie-Hellman problem over cyclic groups of order n . Such reductions are known for groups where the group order is prime and meets certain properties [dB88], or if a certain side information, depending on the respective group, is given [Mau94]. It is also known that no efficient generic reduction exists for groups with orders containing large multiple prime factor [MW98]. Bach [Bac84] has presented a reduction from factoring n to computing discrete logarithms modulo n , i.e. in the multiplicative group \mathbb{Z}_n^* of order $\phi(n)$, where $\phi(\cdot)$ is the Euler totient function.

Furthermore, the analysis of the black-box extraction problem sheds light on the existence of secure ring/field-homomorphic encryption schemes. Consider an encryption function $\text{enc} : K \times P \rightarrow C$, where K, P and C denotes the key, plaintext and ciphertext space, respectively. Moreover, assume that P and C exhibit an algebraic structure with respect to certain operations. If for any $k \in K$ the function $\text{enc}_k := \text{enc}(k, \cdot)$ is a homomorphism from P to C , the corresponding encryption scheme is said to be *homomorphic*. For instance, unpadded RSA is group-homomorphic, since the functions $\text{enc}_e : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, \text{enc}_e(a) := a^e$ satisfy

$$\text{enc}_e(a \cdot b) = (a \cdot b)^e = a^e \cdot b^e = \text{enc}_e(a) \cdot \text{enc}_e(b),$$

where “ \cdot ” denotes the multiplication modulo the RSA modulus n . Further well-known examples of group-homomorphic encryption schemes are native ElGamal [ElG85] and the Paillier cryptosystem [Pai99].

A natural question arising in this context is whether there exist secure *ring*-homomorphic encryption schemes, that is, schemes where P and C exhibit a ring structure, and enc_k is a ring-homomorphism. An efficient algorithm for the black-box extraction problem over the ring P would imply the inexistence of secure ring-homomorphic encryption schemes over P : The black-box can be considered as an idealization of the encryption functions enc_k and the problem of recovering the explicit representation of x as the problem of inverting enc_k . Note that since the black-box representation enables equality checking, also the class of considered encryption schemes allows for checking the equality of encrypted plaintexts. The results by Boneh and Lipton [BL96] and Maurer and Raub [MR07] imply that for the special case of a finite field any such scheme can be broken in subexponential time.

1.1 Our Contribution

In this work, we consider the black-box extraction problem over finite commutative rings with unity whose characteristic n is the product of at least two different primes. To the best of our knowledge, this case has not been treated in the literature yet. We present an efficient reduction from finding a non-trivial factor of n to the black-box extraction problem over the integer ring \mathbb{Z}_n and any (multivariate) polynomial ring R where computations can be performed efficiently. To this end,

we extend a technique due to Leander and Rupp [LR06] which was originally used to prove the equivalence of breaking RSA and factoring regarding generic ring algorithms.

We first provide a reduction for the case $R := \mathbb{Z}_n$. This case is especially interesting since Boneh and Lipton pointed out that their subexponential time black-box extraction algorithm for finite fields can be extended to finite rings \mathbb{Z}_n . Their extension involves the factorization of n . Our result implies that for \mathbb{Z}_n there are no better algorithms than those that factor n . Moreover, under the assumption that factoring n is hard, this implies the inexistence of efficient *generic* reductions from computing discrete logarithms to solving the Diffie-Hellman problem in cyclic groups of order n . Note that, in contrast to Bach [Bac84] who presented a reduction from factoring n to computing discrete logarithms in the multiplicative group of integers \mathbb{Z}_n^* , we consider *generic* algorithms in groups of order n .

In addition, we extend our reduction to rings of the form

$$R := \mathbb{Z}_n[X_1, \dots, X_t]/J,$$

where $t \geq 0$ and J is an ideal in $\mathbb{Z}_n[X_1, \dots, X_t]$ for which a Gröbner basis is known. (Note that for $t = 0$ we obtain the special case $R = \mathbb{Z}_n$.) If computation (i.e., applying the ring operations including reduction, equality testing and random sampling) in R is efficient the same holds for our reduction from finding a factor of n to black-box extraction over R .

Regarding secure homomorphic encryption our result has another interesting consequence: Boneh/Lipton and Maurer/Raub show that any field-homomorphic encryption scheme can be broken in subexponential time. It is open question if there exist more efficient generic algorithms. For a large class of rings, we can negate this question assuming that factoring the ring characteristic cannot be done better than in subexponential time. This might be seen as an indicator for the existence of secure ring-homomorphic encryption schemes.

2 Black-Box Rings and the Black-Box Ring Extraction Problem

Informally, black-box ring algorithms are the class of algorithms that operate on the structure of an algebraic ring, without exploiting specific properties of the representation of ring elements. We adapt Shoup's generic group model [Sho97] to formalize the notion of black-box ring algorithms:

Let $(R, +, \cdot)$ be a finite commutative unitary ring and $S \subset \{0, 1\}^{\lceil \log_2(|R|) \rceil}$ be a set of bit strings of cardinality $|R|$. Let

$$\sigma : R \rightarrow S$$

be a bijective encoding function which assigns ring elements to bit strings, chosen at random among all possible bijections. A *black-box ring algorithm* is an algorithm that takes as input an *encoding list* $(\sigma(r_1), \dots, \sigma(r_k))$, where $r_i \in R$. Note that depending on the particular problem the algorithm might take some additional data as input, such as the characteristic of R , for example. In order to be able to perform the ring operations on randomly encoded elements, the algorithm may query a *black-box ring oracle* $\mathcal{O}_{R,\sigma}$. The oracle takes two indices i, j into the encoding list and a symbol $\circ \in \{+, -, \cdot\}$ as input, computes $\sigma(r_i \circ r_j)$ and appends this bit string to the encoding list (to which the algorithm always has access).

We capture the notion of a black-box ring representation by the following definition:

Definition 1 (Black-Box Ring Representation). *Let $(R, +, \cdot)$ be a finite ring. We call the tuple $(\sigma, \mathcal{O}_{R,\sigma})$ consisting of a randomly chosen encoding function $\sigma : R \rightarrow S$, and a corresponding black-box ring oracle $\mathcal{O}_{R,\sigma}$ a black-box ring representation for R and denote it by R^σ .*

For short, we sometimes call R^σ a black box ring (meaning that we consider a ring exhibiting the structure of R but whose elements are encoded by random bit strings). As an abuse of notation we occasionally write $\sigma(x) \in R^\sigma$ meaning that the unique encoding $\sigma(x)$ of an element $x \in R$ is given. Moreover, when we say in the following that an algorithm \mathcal{A} performs operations on the black-box ring R^σ , we mean that \mathcal{A} interacts with the black-box ring oracle as described above.

Having formalized the notion of a black-box ring, we can define the black-box ring extraction problem:

Definition 2 (BBRE Problem). *Let R be an explicitly given finite commutative ring with unity 1 and known characteristic n . Furthermore, let $\{1, r_1, \dots, r_t\}$ be an (explicitly given) set of generating elements of R , i.e., $R = \langle 1, r_1, \dots, r_t \rangle$, where $\langle 1, r_1, \dots, r_t \rangle$ denotes the ideal in R generated by these elements. The black-box ring extraction (BBRE) problem for R is the task of computing $x \in R$, where x is chosen uniformly random from R , given $\sigma(x), \sigma(1), \sigma(r_1), \dots, \sigma(r_t) \in R^\sigma$.*

3 The Relation between BBRE and Integer Factorization for \mathbb{Z}_n

In this section we consider the BBRE problem for rings which are isomorphic to \mathbb{Z}_n , where n has at least two different prime factors. We provide a reduction from factoring n to the BBRE problem in the following sense: If there exists an efficient algorithm solving the BBRE problem for \mathbb{Z}_n with non-negligible success probability, then there exists an efficient algorithm which finds a factor of n with non-negligible probability.

Theorem 1. *Let $R := \mathbb{Z}_n$ for some integer n having at least two different prime factors. Let \mathcal{A} be an algorithm for the BBRE problem that performs at most $m \leq n$ operations on R^σ . Assume that \mathcal{A} solves the BBRE problem with probability ϵ . Then there is an algorithm \mathcal{B} having white-box access to \mathcal{A} that finds a factor of n with probability at least*

$$\frac{\epsilon}{2} - \frac{1}{2n}$$

by running \mathcal{A} once and performing an additional amount of $O(m^2)$ random choices and $O(m^3)$ operations on R as well as $O(m^2)$ gcd computations on $\log_2(n)$ -bit numbers.

Remark 1. Assume we have a BBRE algorithm \mathcal{A} that works for all rings $R = \mathbb{Z}_n$ where n consists of at least two different prime factors. Then algorithm \mathcal{B} can be used to factor a given integer n completely. This is done by first running \mathcal{B} on n , i.e., \mathcal{B} runs \mathcal{A} on an instance of the BBRE problem over \mathbb{Z}_n and performs some additional operations, resulting in a factor d of n with a certain probability. If n/d is not a prime power, which can easily be determined, we can run \mathcal{B} on n/d and so on. If \mathcal{A} is efficient and solves the BBRE problem with non-negligible probability, then the same holds for the resulting factoring algorithm.

Proof Outline. In a nutshell, our proof works as follows: We replace the original black-box ring oracle \mathcal{O} with an oracle \mathcal{O}_{sim} that simulates \mathcal{O} without using the knowledge of the secret x . We call this setting the *simulation game*. Then we show that the behavior of \mathcal{O}_{sim} is perfectly indistinguishable from \mathcal{O} unless a certain simulation failure \mathbf{F} occurs. Denoting the success event of \mathcal{A} when interacting with \mathcal{O} and \mathcal{O}_{sim} by \mathbf{S} and \mathbf{S}_{sim} , respectively, it immediately follows that $\epsilon = \Pr[\mathbf{S}]$ is upper bound by $\Pr[\mathbf{S}_{\text{sim}}] + \Pr[\mathbf{F}]$. In other words, the probability $\Pr[\mathbf{F}]$ of a failure is at least $\epsilon - \Pr[\mathbf{S}_{\text{sim}}]$. Showing that $1/2 \Pr[\mathbf{F}]$ is in turn a lower bound on the the probability of revealing a factor of n completes our proof.

3.1 Detailed Proof of Theorem 1

Before introducing the actual simulation oracle, as announced in the proof outline, let us first define a slightly modified but *equivalent* version of the original black-box ring oracle \mathcal{O} : Instead of using the ring $R = \mathbb{Z}_n$ for the internal representation of ring elements, these elements are represented by polynomials in the variable X over R which are evaluated with x each time the encoding of a newly computed element must be determined.

Definition 3 (An Equivalent Oracle). *The oracle \mathcal{O} has an input and an output port as well as a random tape and performs computations as follows.*

Input. *As input \mathcal{O} receives the modulus n , an element $x \in_U R$, and the set of encodings S_n .*

Internal State. *As internal state \mathcal{O} maintains two lists $L \subset R[X]$ and $E \subset S_n$. For an index i let L_i and E_i denote the i -th element of L and E , respectively.*

Encoding of Elements. *Each time a polynomial P should be appended to the list L the following computation is triggered to determine the encoding of $P(x)$: \mathcal{O} checks if there exists any index $1 \leq i \leq |L|$ such that*

$$(P - L_i)(x) \equiv 0 \pmod{n}.$$

If this equation holds for some i , then the respective encoding E_i is appended to E again. Otherwise, the oracle chooses a new encoding $s \in_U S \setminus E$ and appends it to E . After \mathcal{O} has determined an encoding it writes it to its output port.

The computation of \mathcal{O} starts with an initialization phase, which is run once, followed by the execution of the query-handling phase:

Initialization. *The list L is initialized with the polynomials $1, X$ and the list E is initialized with corresponding encodings.*

Query-handling. *Upon receiving a query (\circ, i_1, i_2) on its input tape, where $\circ \in \{+, -, \cdot\}$ identifies an operation and i_1, i_2 are indices identifying the list elements the operation should be applied to, \mathcal{O} appends the polynomial $P := L_{i_1} \circ L_{i_2}$ to L and the corresponding encoding to E .*

A Simulation Game. Now we replace \mathcal{O} by a simulation oracle \mathcal{O}_{sim} . The simulation oracle is defined exactly like \mathcal{O} except that it determines the encodings of elements in a different way in order to be independent of the secret x .

Each time a polynomial P is appended to the end of list L (during initialization or query-handling), \mathcal{O}_{sim} does the following: Let $L_j = P$ denote the last entry of the updated list. Then for each $1 \leq i < j$ the simulation oracle chooses a *new* element $x_{i,j} \in R$ uniformly at random and checks whether

$$(L_i - L_j)(x_{i,j}) \equiv 0 \pmod{n}.$$

If the above equation is not satisfied for any i , the oracle chooses a new encoding $s \in_U S \setminus E$ and appends it to E . Otherwise, for the first i the equation is satisfied, the corresponding encoding E_i is appended to E again (i.e., $E_j = E_i$). The determined encoding is written to the output port.

The algorithm is successful in the simulation game if it outputs the element x (given as input to \mathcal{O}_{sim}). We denote this event by \mathbf{S}_{sim} .

Note that due to the modification of the element encoding procedure, it is now possible that both an element $L_i(x)$ is assigned to two or more different encodings and that different elements are assigned to the same encoding. In these cases the behavior of \mathcal{O}_{sim} differs from that of \mathcal{O} what may allow to distinguish between the oracles. In the case of a differing behaviour the following failure event \mathbf{F} occurred (i.e., the occurrence of \mathbf{F} is a necessary condition for a differing behavior): There exist $i < j \in \{1, \dots, |L|\}$ satisfying the equations

$$(L_i - L_j)(x) \equiv 0 \pmod{n} \text{ and } (L_i - L_j)(x_{i,j}) \not\equiv 0 \pmod{n}, \quad (1)$$

or the equations

$$(L_i - L_j)(x) \not\equiv 0 \pmod{n} \text{ and } (L_i - L_j)(x_{i,j}) \equiv 0 \pmod{n}. \quad (2)$$

Remark 2. There is a technical subtlety. If there is $i < j$ s.t. $(L_i - L_j)(x) \equiv 0 \pmod{n}$ but $(L_i - L_j)(x_{i,j}) \not\equiv 0 \pmod{n}$ then \mathcal{O}_{sim} does not necessarily determine different encodings for $L_j(x)$. There may be some $i < i' < j$ such that $(L_{i'} - L_j)(x) \equiv 0 \pmod{n}$ and $(L_{i'} - L_j)(x_{i',j}) \equiv 0 \pmod{n}$ and $E_i = E_{i'}$. So the simulation failure event as defined by us is just a necessary but not a sufficient condition for discriminative behaviour of \mathcal{O} and \mathcal{O}_{sim} .

It is important to observe that the original game and the simulation game proceed identically unless \mathbf{F} occurs: To this end consider the algorithm \mathcal{A} as deterministic Turing machine with identical input and random tape in both games. Also, consider the oracles \mathcal{O} and \mathcal{O}_{sim} as deterministic Turing machines receiving the same inputs and random tapes.¹ Assuming that \mathbf{F} does not occur, the algorithm receives the same sequence of encodings and thus issues the same sequence of queries in both games. Furthermore, it outputs the same element in the end of both games and thus wins the simulation game if and only if it wins the original game. Hence, we have the following relation between the considered events

$$\mathbf{S} \wedge \neg \mathbf{F} \iff \mathbf{S}_{\text{sim}} \wedge \neg \mathbf{F}$$

We can obtain an upper bound on $\Pr[\mathbf{S}]$ by applying the Difference Lemma (Lemma 1) and deriving upper bounds on $\Pr[\mathbf{S}_{\text{sim}}]$ and $\Pr[\mathbf{F}]$.

Lemma 1 (Difference Lemma [Sho06]). *Let $\mathbf{S}, \mathbf{S}_{\text{sim}}$, and \mathbf{F} be events over the same probability space. If $\mathbf{S} \wedge \neg \mathbf{F} \iff \mathbf{S}_{\text{sim}} \wedge \neg \mathbf{F}$, then it holds that*

$$\Pr[\mathbf{S}] \leq \Pr[\mathbf{S}_{\text{sim}}] + \Pr[\mathbf{F}].$$

Bounding the Probability of Success in the Simulation Game. Since all computations are independent of the uniformly random element $x \in R$, the algorithm \mathcal{A} can only guess x :

$$\Pr[\mathbf{S}_{\text{sim}}] \leq \frac{1}{|R|} = \frac{1}{n}.$$

Bounding the Probability of a Simulation Failure. Let $\mathfrak{D} = \{L_i - L_j | 1 \leq i < j \leq |L|\}$ denote the set of all non-trivial differences of polynomials in L after a run of \mathcal{A} . In the following we show how the probability that a polynomial $\Delta \in \mathfrak{D}$ causes a simulation failure is related to the probability of revealing a factor of n by simply evaluating Δ with a uniformly random element from R .

For fixed $\Delta \in \mathfrak{D}$ let \mathbf{F}_Δ denote the event that Δ causes a simulation failure as defined by Equations (1) and (2). Furthermore, let \mathbf{D}_Δ denote the event that $\gcd(n, \Delta(a)) \notin \{1, n\}$ when choosing an element a uniformly at random from R .

Now, we are going to express the probabilities of both events using the same terms. Let $n = \prod_{i=1}^k p_i^{e_i}$ be the prime factor decomposition of n . Hence, R is isomorphic to $\mathbb{Z}_{p_1^{e_1}} \times \dots \times \mathbb{Z}_{p_k^{e_k}}$ by

¹ To be precise here, we actually should have defined \mathcal{O} to perform exactly the same random choices as \mathcal{O}_{sim} (i.e., letting \mathcal{O} also choose the elements $x_{i,j}$ but without using them).

the Chinese Remainder Theorem [Gri99, p.184]. Then we can write

$$\begin{aligned}
\Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod n] &= \Pr_{a \in U R} [(\Delta(a) \equiv 0 \pmod{p_1^{e_1}}) \wedge \dots \wedge (\Delta(a) \equiv 0 \pmod{p_k^{e_k}})] \\
&= \prod_{i=1}^k \Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod{p_i^{e_i}}] \\
&= \prod_{i=1}^k \nu_i,
\end{aligned} \tag{3}$$

where $\nu_i := \frac{|\{a \in R \mid \Delta(a) \equiv 0 \pmod{p_i^{e_i}}\}|}{|R|}$. Note that the second line of the above equation follows from the fact that the events defined by the predicates $\Delta(a) \equiv 0 \pmod{p_i^{e_i}}$ are mutually independent. Using Equation (3) we can express the probability of \mathbf{F}_Δ by

$$\begin{aligned}
\Pr[\mathbf{F}_\Delta] &= \Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod n] \left(1 - \Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod n]\right) \\
&\quad + \Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod n] \left(1 - \Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod n]\right) \\
&= 2 \left(1 - \prod_{i=1}^k \nu_i\right) \left(\prod_{i=1}^k \nu_i\right).
\end{aligned}$$

Similarly, we can write the probability of \mathbf{D}_Δ as

$$\begin{aligned}
\Pr[\mathbf{D}_\Delta] &= 1 - \Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod n] - \Pr_{a \in U R} [(\Delta(a) \not\equiv 0 \pmod{p_1^{e_1}}) \wedge \dots \wedge (\Delta(a) \not\equiv 0 \pmod{p_k^{e_k}})] \\
&= 1 - \Pr_{a \in U R} [\Delta(a) \equiv 0 \pmod n] - \prod_{i=1}^k \Pr_{a \in U R} [\Delta(a) \not\equiv 0 \pmod{p_i^{e_i}}] \\
&= 1 - \prod_{i=1}^k \nu_i - \prod_{i=1}^k (1 - \nu_i)
\end{aligned}$$

Now, the key observation is that we have the following relation between the probabilities of the events \mathbf{F}_Δ and \mathbf{D}_Δ :

Lemma 2. $\forall \Delta \in \mathfrak{D} : 2 \Pr[\mathbf{D}_\Delta] \geq \Pr[\mathbf{F}_\Delta]$

Proof. We have

$$\begin{aligned}
2 \Pr[\mathbf{D}_\Delta] - \Pr[\mathbf{F}_\Delta] &= 2 \left(1 - 2 \prod_{i=1}^k \nu_i - \prod_{i=1}^k (1 - \nu_i) + \prod_{i=1}^k \nu_i^2\right) \geq 0 \\
&\iff \left(1 - \prod_{i=1}^k \nu_i\right)^2 \geq \prod_{i=1}^k (1 - \nu_i)
\end{aligned}$$

It is easy to prove by induction over k that the inequality

$$\left(1 - \prod_{i=1}^k \nu_i\right)^k \geq \prod_{i=1}^k (1 - \nu_i)$$

holds for all $k \geq 1$. From this our claim follows immediately since

$$\left(1 - \prod_{i=1}^k \nu_i\right)^2 \geq \left(1 - \prod_{i=1}^k \nu_i\right)^k$$

holds for all $k \geq 2$. □

The Factoring Algorithm. Based on the relation given by Lemma 2, we construct an efficient factoring algorithm whose success probability is half the probability of a simulation failure. Consider an algorithm \mathcal{B} that runs the BBRE algorithm \mathcal{A} on an arbitrary instance of the BBRE problem over \mathbb{Z}_n . During this run it records the sequence of queries that \mathcal{A} issues, i.e., it records the same list L of polynomials as the black-box ring oracle. Then for each $\Delta \in \mathfrak{D}$ the algorithm \mathcal{B} chooses a new random element $a \in \mathbb{Z}_n$ (like the oracle is doing), and computes $\gcd(n, \Delta(a))$. There are at most $(m^2 + 3m + 2)/2$ such polynomials and each of them can be evaluated using at most $m + 1$ ring operations (since it is given as a straight-line program of length at most m). Thus, \mathcal{B} chooses $O(m^2)$ random elements and performs $O(m^3)$ operations on R as well as $O(m^2)$ gcd computations on $\log_2(n)$ -bit numbers. Let the event that at least one of these gcd computations yields a non-trivial factor of n be denoted by \mathbf{D} . Then we can show the following lower bound on the the success probability of \mathcal{B} :

Lemma 3. $2 \Pr[\mathbf{D}] \geq \Pr[\mathbf{F}]$

To prove this bound we make use of auxiliary Lemma 4.

Lemma 4. For $z \geq 2$ let A_1, \dots, A_z be mutually independent events over the same probability space. Then also the events $\bigcup_{i=1}^{z-1} A_i$ and A_z are independent.

Proof (Lemma 3). Let the set of difference polynomials induced by the run of \mathcal{A} be given by \mathfrak{D} . Note that the events $(\mathbf{F}_\Delta)_{\Delta \in \mathfrak{D}}$ are mutually independent and the same holds for the events $(\mathbf{D}_\Delta)_{\Delta \in \mathfrak{D}}$. The total failure probability can be written as

$$\Pr[\mathbf{F}] = \Pr \left[\bigcup_{\Delta \in \mathfrak{D}} \mathbf{F}_\Delta \right]$$

and the success probability of our factoring algorithm \mathcal{B} as

$$\Pr[\mathbf{D}] = \Pr \left[\bigcup_{\Delta \in \mathfrak{D}} \mathbf{D}_\Delta \right].$$

We show our claim

$$2 \Pr \left[\bigcup_{\Delta \in \mathfrak{D}} \mathbf{D}_\Delta \right] \geq \Pr \left[\bigcup_{\Delta \in \mathfrak{D}} \mathbf{F}_\Delta \right] \tag{4}$$

by induction over the size of \mathfrak{D} . For $|\mathfrak{D}| = 1$ the claim trivially holds due to Lemma 2. Now, let $\mathfrak{D} = \mathfrak{D}' \cup \{\hat{\Delta}\}$ and assume that Equation (4) holds for the smaller set \mathfrak{D}' instead of \mathfrak{D} . Then there exists $0 \leq \alpha_1 \leq 1$ such that

$$\Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{D}_\Delta \right] = \frac{1}{2} \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] + \alpha_1 \tag{5}$$

Similarly, Lemma 2 yields that there exists $0 \leq \alpha_2 \leq 1$ such that

$$\mathbf{D}_{\hat{\Delta}} = \frac{1}{2} \mathbf{F}_{\hat{\Delta}} + \alpha_2 \tag{6}$$

In this way we get

$$\begin{aligned}
2 \Pr \left[\bigcup_{\Delta \in \mathfrak{D}} \mathbf{D}_\Delta \right] &= 2 \Pr \left[\left(\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{D}_\Delta \right) \cup \mathbf{D}_{\bar{\Delta}} \right] \\
&= 2 \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{D}_\Delta \right] + 2 \Pr \left[\mathbf{D}_{\bar{\Delta}} \right] - 2 \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{D}_\Delta \right] \Pr \left[\mathbf{D}_{\bar{\Delta}} \right] \\
&= \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] + \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] - \frac{1}{2} \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] \\
&\quad + 2(\alpha_1 + \alpha_2) - \alpha_2 \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] - \alpha_1 \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] - 2\alpha_1\alpha_2 \\
&\geq \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] + \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] - \frac{1}{2} \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] \\
&\geq \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] + \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] - \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] \\
&= \Pr \left[\bigcup_{\Delta \in \mathfrak{D}} \mathbf{F}_\Delta \right]
\end{aligned}$$

The second line follows from the fact that the events $(\mathbf{D}_\Delta)_{\Delta \in \mathfrak{D}}$ are mutually independent and thus $\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{D}_\Delta$ and $\mathbf{D}_{\bar{\Delta}}$ by applying Lemma 4. To get the third line we apply Equations (5) and (6). The first inequality follows from

$$2(\alpha_1 + \alpha_2) - \alpha_2 \Pr \left[\bigcup_{\Delta \in \mathfrak{D}'} \mathbf{F}_\Delta \right] - \alpha_1 \Pr \left[\mathbf{F}_{\bar{\Delta}} \right] - 2\alpha_1\alpha_2 \geq \alpha_1 + \alpha_2 - 2\alpha_1\alpha_2 \geq \alpha_1\alpha_2 + \alpha_1\alpha_2 - 2\alpha_1\alpha_2 = 0$$

and the last line from the fact that the events $(\mathbf{F}_\Delta)_{\Delta \in \mathfrak{D}}$ are mutually independent. \square

To Summarize. Remember that $\epsilon = \Pr[\mathbf{S}]$ denotes the success probability of the BBRE algorithm \mathcal{A} . From Lemma 1 follows that

$$\Pr[\mathbf{F}] \geq \epsilon - \Pr[\mathbf{S}_{\text{sim}}] \geq \epsilon - \frac{1}{n}.$$

Deploying the above lower bound on $\Pr[\mathbf{F}]$ in Lemma 3 finally yields

$$\Pr[\mathbf{D}] \geq \frac{1}{2} \Pr[\mathbf{F}] \geq \frac{\epsilon}{2} - \frac{1}{2n}.$$

4 Extending our Reduction to Multivariate Polynomial Rings

In this section we are going to lift our reduction from the special case $R = \mathbb{Z}_n$ to the case

$$R = \mathbb{Z}_n[X_1, \dots, X_t]/J,$$

where $\mathbb{Z}_n[X_1, \dots, X_t]$ denotes the ring of polynomials over \mathbb{Z}_n in indeterminates X_1, \dots, X_t ($t \geq 0$) and J is an ideal in this polynomial ring such that R is finite. Note that any finite commutative unitary ring R with characteristic n has a representation of the form $R = \mathbb{Z}_n[X_1, \dots, X_t]/J$, as stated by the following lemma.

Lemma 5. *Let R be a finite commutative unitary ring of characteristic n . Then there is a number $t \leq \log_2 |R|$ and a finitely generated ideal J of $\mathbb{Z}_n[X_1, \dots, X_t]$ such that $R \cong \mathbb{Z}_n[X_1, \dots, X_t]/J$.*

Proof. Let $M = \{m_1, \dots, m_t\} \subset R$ be a generating subset of R , i.e., $R = \langle M \rangle$. Consider the mapping $\phi : \mathbb{Z}_n[X_1, \dots, X_t] \rightarrow R$ such that $1 \mapsto 1$ and $X_i \mapsto m_i$ for $1 \leq i \leq t$. Certainly, ϕ is a ring homomorphism implying that $J := \ker(\phi)$ is an ideal of $\mathbb{Z}_n[X_1, \dots, X_t]$. Applying the fundamental theorem on homomorphisms (e.g., see [Lan02, p.89]) yields that $\mathbb{Z}_n[X_1, \dots, X_t]/J \cong R$.

Since \mathbb{Z}_n trivially is a Noetherian ring it follows by Hilbert's basis theorem (e.g, see [Gri99, p.181]) that also the polynomial ring $\mathbb{Z}_n[X_1, \dots, X_t]$ is Noetherian. Thus, every ideal of $\mathbb{Z}_n[X_1, \dots, X_t]$, especially J , is finitely generated.

By the fundamental theorem of finitely generated abelian groups (e.g., see [Gri99, p.48]) the additive group of a finite ring decomposes uniquely (up to order) into a direct product of cyclic groups. Observe that a group of cardinality $|R|$ decomposes into a product of at most $\log_2 |R|$ groups. Hence, setting M to be the set of generators of these subgroups of $(R, +)$, we see that a number of $t \leq \log_2 |R|$ elements is sufficient to generate the entire ring. \square

We start this chapter with presenting a useful decomposition of such rings similar to the CRT-decomposition for \mathbb{Z}_n . Next, we state some facts about Gröbner bases over these rings and their component rings. Finally, we use these results in a reduction proof which is similar to the one for \mathbb{Z}_n .

4.1 A Prime-Power Decomposition for Multivariate Polynomial Rings

It is a well-known fact that any finite commutative ring is uniquely (up to order) decomposable into a direct product of *local* rings [McD74, p.95]. However, since this decomposition does not meet our requirements, we devise another simple way of decomposing R into a direct product of rings with prime-power characteristic, but not necessarily local rings.

Lemma 6. *Let $R = \mathbb{Z}_n[X_1, \dots, X_t]/J$ (where R is not necessarily finite) and $n = \prod_{i=1}^k p_i^{e_i}$ be the prime factor decomposition of the characteristic of R . Then R is decomposable into a direct product of rings*

$$R \cong R_1 \times \dots \times R_k,$$

where $R_i := \mathbb{Z}_{p_i^{e_i}}[X_1, \dots, X_t]/J$.

Proof. Let F be a set of polynomials generating the ideal J , we denote this by $J = \langle F \rangle$. Note that the ring R can equivalently be written as $\mathbb{Z}[X_1, \dots, X_t]/\langle n, F \rangle$. For $1 \leq i \leq k$ let $J_i := \langle p_i^{e_i}, F \rangle$ which is an ideal in $\mathbb{Z}[X_1, \dots, X_t]$. Then for each $1 \leq i < j \leq k$ it holds that $J_i + J_j := \{a + b | a \in J_i, b \in J_j\} = \mathbb{Z}[X_1, \dots, X_t]$. Moreover, we have $\bigcap_{i=1}^k J_i = \langle n, F \rangle$. Thus, by the generalized Chinese Remainder Theorem [Gri99, p.184], we obtain the isomorphism

$$\begin{aligned} \mathbb{Z}_n[X_1, \dots, X_t]/J &\cong \mathbb{Z}[X_1, \dots, X_t]/\langle n, F \rangle \\ &\cong \mathbb{Z}[X_1, \dots, X_t]/\langle p_1^{e_1}, F \rangle \times \dots \times \mathbb{Z}[X_1, \dots, X_t]/\langle p_k^{e_k}, F \rangle \\ &\cong \mathbb{Z}_{p_1^{e_1}}[X_1, \dots, X_t]/J \times \dots \times \mathbb{Z}_{p_k^{e_k}}[X_1, \dots, X_t]/J. \end{aligned}$$

\square

We call this way of decomposing R the *prime-power decomposition* of R .

Remark 3. Note that Lemma 6 extends to any finite commutative unitary ring of characteristic n since any such ring can be represented as polynomial ring $\mathbb{Z}_n[X_1, \dots, X_t]/J$ for some t and J .

4.2 Gröbner Bases for Polynomial Ideals over Rings

Roughly speaking, a Gröbner basis G is a generating set of an ideal J in a multivariate polynomial ring exhibiting the special property that reduction of polynomials from J modulo the set G always yields the residue zero. This property is not satisfied for arbitrary ideal bases and enables effective

computation in residue class rings modulo polynomial ideals in the first place. Gröbner bases were originally introduced by Buchberger [Buc65] for ideals J in $K[X_1, \dots, X_t]$ where the coefficient space K is a field. Later this notion were generalized to the case where K is a Noetherian ring such as \mathbb{Z}_n (e.g., see [AL94, Chapter 4]).

Let us introduce some notation. A *monomial* or *power product* in indeterminates X_1, \dots, X_t is a product of the form $\mathcal{X} = X_1^{a_1} \cdots X_t^{a_t}$ for some $(a_1, \dots, a_t) \in \mathbb{N}_0^t$. In the following let an arbitrary but admissible order $>$ on monomials be given. For instance, this could be the lexicographic order $>_{\text{lex}}$ defined as: $\mathcal{X}_1 = X_1^{a_1} \cdots X_t^{a_t} >_{\text{lex}} \mathcal{X}_2 = X_1^{b_1} \cdots X_t^{b_t}$ iff the leftmost non-zero entry of $(a_1 - b_1, \dots, a_t - b_t)$ is positive.

Let $f \in \mathbb{Z}_n[X_1, \dots, X_t]$ with $f \neq 0$. Then we can write f as $f = c_1 \mathcal{X}_1 + \dots + c_s \mathcal{X}_s$, where $c_1, \dots, c_s \in \mathbb{Z}_n \setminus \{0\}$ and $\mathcal{X}_1 > \dots > \mathcal{X}_s$. The *leading coefficient* $\text{lc}(f)$, the *leading monomial* $\text{lm}(f)$, and the *leading term* $\text{lt}(f)$ of f with respect to $>$ are defined as $\text{lc}(f) := a_1$, $\text{lm}(f) := \mathcal{X}_1$, and $\text{lt}(f) := a_1 \mathcal{X}_1$, respectively.

Now, we are able to define the reduction of a polynomial modulo a set of polynomials. To this end, we adopt the respective definitions from [AL94, Chapter 4]. In the following we do not mention the fixed monomial ordering explicitly anymore.

Definition 4 (Polynomial Reduction). *Let two polynomials f and h and a set of non-zero polynomials $F = \{f_1, \dots, f_s\}$ in $\mathbb{Z}_n[X_1, \dots, X_t]$ be given.*

- (a) *We say that f can be reduced to h modulo F in one step, denoted by $f \xrightarrow{F} h$, if and only if $h = f - (c_1 \mathcal{X}_1 f_1 + \dots + c_s \mathcal{X}_s f_s)$ for $c_1, \dots, c_s \in R$ and power products $\mathcal{X}_1, \dots, \mathcal{X}_s$ where $\text{lm}(f) = \mathcal{X}_i \text{lm}(f_i)$ for all i such that $c_i \neq 0$ and $\text{lt}(f) = c_1 \mathcal{X}_1 \text{lt}(f_1) + \dots + c_s \mathcal{X}_s \text{lt}(f_s)$.*
- (b) *We say that f can be reduced to h modulo F , denoted by $f \xrightarrow{F}_+ h$, if and only if there exist polynomials $h_1, \dots, h_{\ell-1} \in \mathbb{Z}_n[X_1, \dots, X_t]$ such that $f \xrightarrow{F} h_1 \xrightarrow{F} h_2 \xrightarrow{F} \dots \xrightarrow{F} h_{\ell-1} \xrightarrow{F} h$.*
- (c) *A polynomial h is called minimal with respect to F if h cannot be reduced modulo F .*
- (d) *We call h a (minimal) residue of f modulo F , denoted by $h = f \bmod F$, if $f \xrightarrow{F}_+ h$ and h is minimal.*

Note that there is an efficient algorithm computing a minimal residue of a polynomial f modulo a set F (provided that the representation of f and F is efficient) according to the above definition. For instance, see Algorithm 4.1.1 in [AL94].

Definition 5 (Gröbner Basis). *Let J be an ideal in $\mathbb{Z}_n[X_1, \dots, X_t]$ and $G = \{g_1, \dots, g_s\}$ be a set of non-zero polynomials such that $\langle G \rangle = J$. Then G is called a Gröbner basis for J if for any polynomial $f \in \mathbb{Z}_n[X_1, \dots, X_t]$ we have*

$$f \in J \iff f \bmod G = 0.$$

Fortunately, there always exists an ideal basis with this special property, as stated by Lemma 7. However, note that given an arbitrary ideal basis, a Gröbner basis for the corresponding ideal is not always easy to compute. In the following we always assume that Gröbner bases for the considered ideals are given.

Lemma 7. *Let J be a non-zero ideal of $\mathbb{Z}_n[X_1, \dots, X_t]$, then J has a finite Gröbner basis.*

The following lemma is crucial for proving that (similar to the \mathbb{Z}_n -case) an element $f \in R \cong R_1 \times \dots \times R_k$ that is congruent to zero over a component R_i but not congruent to zero over another component R_j (cf. Lemma 6) helps in factoring n . Observe that Lemma 8 requires that the leading coefficients of all given Gröbner basis elements are units. For our purposes, this is not a restriction

at all but a reasonable assumption since otherwise the given representation of R would immediately reveal a factor of n .

A proof for this lemma based on the notion of syzygies can be found in Appendix A.

Lemma 8. *Let $A = \mathbb{Z}_n[X_1, \dots, X_t]$ and $n = \prod_{i=1}^k p_i^{e_i}$. Furthermore, let $G = \{g_1, \dots, g_s\}$ be a Gröbner basis for the ideal $J = \langle g_1, \dots, g_s \rangle$ in A such that $\text{lc}(g_i) \in \mathbb{Z}_n^*$ for all $1 \leq i \leq s$. Then for each $1 \leq \ell \leq s$ the set $G_\ell = \{p_\ell^{e_\ell}, g_1, \dots, g_s\}$ is a Gröbner basis for the ideal $J_\ell = \langle p_\ell^{e_\ell}, g_1, \dots, g_s \rangle$ in A .*

4.3 The Relation between BBRE and Integer Factorization for $\mathbb{Z}_n[X_1, \dots, X_t]/J$

We are going to lift our reduction from the special case $R = \mathbb{Z}_n$ to the more general case of finite multivariate polynomial rings $R = \mathbb{Z}_n[X_1, \dots, X_t]/J$, where J is given by a Gröbner basis. Let $n = \prod_{i=1}^k p_i^{e_i}$. In the case $R = \mathbb{Z}_n$ our factoring algorithm was successful if it was able to find an element $a \in R$ such that $a \in \langle p_i^{e_i} \rangle$ and $a \notin \langle p_j^{e_j} \rangle$ for some $1 \leq i < j \leq k$. The following theorem shows that a generalization of this fact holds for residue class rings modulo polynomial ideals given by Gröbner basis.

Theorem 2. *Let $A = \mathbb{Z}_n[X_1, \dots, X_t]$ where $n = \prod_{i=1}^k p_i^{e_i}$ and $k \geq 2$. Furthermore, let $G = \{g_1, \dots, g_s\}$ be a Gröbner basis for the ideal $J = \langle g_1, \dots, g_s \rangle$ in A such that $\text{lc}(g_i) \in \mathbb{Z}_n^*$ for all $1 \leq i \leq s$. Assume an element $f \in A$ is given, such that $f \in J_i = \langle p_i^{e_i}, g_1, \dots, g_s \rangle$ and $f \notin J_j = \langle p_j^{e_j}, g_1, \dots, g_s \rangle$ for some $1 \leq i < j \leq k$. Then computing $\text{gcd}(\text{lc}(r), n)$, where $r = f \bmod G$, yields a non-trivial factor of n .*

Proof. First of all, observe that since $f \notin J_j$ we have that $f \notin J$ and so $r = f \bmod G$ is not zero by Definition 5. Since r is a minimal residue and the leading coefficients $\text{lc}(g_i)$ of all Gröbner basis elements are units, it follows by Definition 4 that the leading monomial $\text{lm}(r)$ of r is not divisible by any leading monomial $\text{lm}(g_i)$ of a Gröbner basis element. Otherwise, r would be reducible to $r \xrightarrow{G} r - \text{lc}(g_i)^{-1} \text{lc}(r) \mathcal{X}_i g_i$, where $\mathcal{X}_i \text{lm}(g_i) = \text{lm}(r)$, using some g_i such that $\text{lm}(g_i)$ divides $\text{lm}(r)$. Moreover, by Lemma 8, the set $G_i = \{p_i^{e_i}, g_1, \dots, g_s\}$ is a Gröbner basis for the ideal J_i . Thus, since $f \in J_i$ also $r \in J_i$ and the reduction of r modulo G_i would yield the minimal residue zero. As the leading monomial $\text{lm}(r)$ is not divisible by any $\text{lm}(g_i)$, the leading coefficient $\text{lc}(r)$ must be divisible by $p_i^{e_i}$. Since $r \neq 0$ and $\text{lc}(r) \not\equiv 0 \pmod{n}$ (by definition of the leading coefficient) computing $\text{gcd}(\text{lc}(r), n)$ yields a non-trivial factor of n . \square

The followings example illustrates the result captured by the above theorem. Moreover, it shows that in the case where G is *not* a Gröbner basis, elements f satisfying the properties from Theorem 2 seem not to reveal a factor of n by considering their residues modulo G .

Example 1. Consider the finite ring $R = \mathbb{Z}_{225}[X_1, X_2]/J$ where $225 = 3^2 \cdot 5^2$ and $J = \langle X_1 X_2 + 1, X_2^2 + 224 \rangle$. Let us use the lexicographic order where $X_1 > X_2$ for polynomial reduction. Note that $F = \{X_1 X_2 + 1, X_2^2 + 224\}$ is a generating set, but not a Gröbner basis for J with respect to this order. Furthermore, note that $G = \{X_1 + X_2, X_2^2 + 224\}$ is a Gröbner basis for J . Let $f = X_1^4 X_2^2 + 224 X_1^4 + 2 X_1 + 7 X_2^2 + 38 X_2 + 218 = (4 X_2)(3^2) + (X_1^4 + 7)(X_2^2 + 224) + 2(X_1 + X_2) \pmod{225}$. It holds that $f \in J_1 = \langle 3^2, X_1 + X_2, X_2^2 + 224 \rangle$ but $f \notin J_2 = \langle 5^2, X_1 + X_2, X_2^2 + 224 \rangle$. The polynomial f can be reduced modulo F to the minimal residue $f \bmod F = 2 X_1 + 7 X_2^2 + 38 X_2 + 218 = f - (X_1^4 + 7)(X_2^2 + 224) \pmod{225}$. It is easy to see that no coefficient of this residue share a non-trivial factor with 225. Modulo the Gröbner basis G the polynomial f can be reduced to the minimal residue $r = f \bmod G = 36 X_2 = f - (X_1^4 + 7)(X_2^2 + 224) - 2(X_1 + X_2) \pmod{225}$. Computing $\text{gcd}(\text{lc}(r), 225) = 9$ yields a non-trivial factor of 225.

The above fact allows us to formulate and prove a theorem similar to Theorem 1.

Theorem 3. *Let $R := \mathbb{Z}_n[X_1, \dots, X_t]/J$ for some integer n having at least two different prime factors and ideal J in $\mathbb{Z}_n[X_1, \dots, X_t]$. Assume a Gröbner basis $G = \{g_1, \dots, g_s\}$ for J is given. Let \mathcal{A} be an algorithm for the BBRE problem that performs at most $m \leq |R|$ operations on R^σ . Assume that \mathcal{A} solves the BBRE problem with probability ϵ . Then there is an algorithm \mathcal{B} having white-box access to \mathcal{A} that finds a factor of n with probability at least*

$$\frac{\epsilon}{2} - \frac{1}{2n}$$

by running \mathcal{A} once and performing an additional amount of $O((m+t)^2)$ random choices and $O(m(m+t)^2)$ operations on R as well as $O((m+t)^2 + s)$ gcd computations on $\log_2(n)$ -bit numbers.²

Proof. We adapt the proof of Theorem 1. The description of the original and the simulation game almost carries over completely just by setting $R := \mathbb{Z}_n[X_1, \dots, X_t]/J$. There are only a few slight technical differences concerning the oracles \mathcal{O} and \mathcal{O}_{sim} considered in the original game (cf. Definition 3) and the simulation game:

- The list L maintained by both oracles is initialized with the $t+1$ generating elements $1, X_1, \dots, X_t$ of R , and with the variable X . As before, computed ring elements are represented by polynomials in $R[X] = (\mathbb{Z}_n[X_1, \dots, X_t]/J)[X]$.
- Whenever an element $P \in R[X]$ is appended to the list L , say as element $L_j = P$, \mathcal{O} checks whether there exists an element $L_i \in L$ such that $(L_i - L_j)(x) \in J$ which is equivalent to checking whether the residue $r = (L_i - L_j)(x) \bmod G$ is the zero polynomial over \mathbb{Z}_n . Instead of using the given secret x in the above evaluation, the simulation oracle \mathcal{O}_{sim} performs this check using a *new* random element $x_{i,j} \in R$ for each difference polynomial $L_i - L_j$ ($i < j \in \{1, \dots, |L|\}$).

The rest of the description of the games applies unchanged. Let the events \mathbf{S} , \mathbf{S}_{sim} and \mathbf{F} be defined analogously to the case $R = \mathbb{Z}_n$. We are left with deriving bounds for the success probability $\Pr[\mathbf{S}_{\text{sim}}]$ in the simulation game and for the probability $\Pr[\mathbf{F}]$ of a simulation failure. This also works similarly to the previous case except for some technical differences.

Bounding the Probability of Success in the Simulation Game. All computations in the simulation game are independent of the uniformly random element x . Thus, the algorithm \mathcal{A} can only guess x , resulting in

$$\Pr[\mathbf{S}_{\text{sim}}] \leq \frac{1}{|R|} \leq \frac{1}{n}.$$

Bounding the Probability of a Simulation Failure. Again let $\mathfrak{D} := \{L_i - L_j \mid 1 \leq i < j \leq |L|\}$ denote the set of all non-trivial differences of polynomials in L after a run of \mathcal{A} , and let Δ be some (fixed) element of \mathfrak{D} . Let $n = \prod_{i=1}^k p_i^{e_i}$ be the prime factor decomposition of n , then R has a prime power decomposition into

$$R \cong \mathbb{Z}[X_1, \dots, X_t]/\langle p_1^{e_1}, G \rangle \times \dots \times \mathbb{Z}[X_1, \dots, X_t]/\langle p_k^{e_k}, G \rangle$$

² We count the addition/multiplication of two ring elements together with the reduction modulo G as one ring operation.

according to Lemma 6. Let

$$\nu_i := \frac{|\{a \in R \mid \Delta(a) \in \langle p_i^{e_i}, G \rangle\}|}{|R|}$$

be the probability that $\Delta(a) \in \langle p_i^{e_i}, I \rangle$ for a uniformly random element $a \in R$. Using this notation, the probability that Δ causes a simulation failure is given by

$$\begin{aligned} \Pr[\mathbf{F}_\Delta] &= \Pr_{a \in \mathcal{U}R}[\Delta(a) \in \langle n, G \rangle] \left(1 - \Pr_{a \in \mathcal{U}R}[\Delta(a) \in \langle n, G \rangle] \right) \\ &\quad + \Pr_{a \in \mathcal{U}R}[\Delta(a) \in \langle n, G \rangle] \left(1 - \Pr_{a \in \mathcal{U}R}[\Delta(a) \in \langle n, G \rangle] \right) \\ &= 2 \prod_{i=1}^k \nu_i \left(1 - \prod_{i=1}^k \nu_i \right). \end{aligned}$$

By Theorem 2, Δ reveals a factor of n if we can find an element $a \in R$ such that $\Delta(a) \in \langle p_i^{e_i}, G \rangle$ and $\Delta(a) \notin \langle p_j^{e_j}, G \rangle$ for some $1 \leq i < j \leq k$. In this case computing $\gcd(\text{lc}(\Delta(a) \bmod G), n)$ yields a non-trivial factor provided that $\text{lc}(g) \in \mathbb{Z}_n^*$ for all $g \in G$. The probability of finding such an element a by sampling uniformly at random from R is given by

$$\begin{aligned} \Pr[\mathbf{D}_\Delta] &= 1 - \Pr_{a \in \mathcal{U}R}[\Delta(a) \in \langle n, G \rangle] - \Pr_{a \in \mathcal{U}R}[(\Delta(a) \notin \langle p_1^{e_1}, G \rangle) \wedge \dots \wedge (\Delta(a) \notin \langle p_k^{e_k}, G \rangle)] \\ &= 1 - \prod_{i=1}^k \nu_i - \prod_{i=1}^k (1 - \nu_i). \end{aligned}$$

Note that the equations for the probabilities of \mathbf{D}_Δ and \mathbf{F}_Δ equal the ones derived in the proof of Theorem 1.

Finally, we apply Lemma 3 and Lemma 1 to obtain

$$\Pr[\mathbf{D}] \geq \frac{1}{2} \Pr[\mathbf{F}] \geq \frac{\epsilon}{2} - \frac{1}{2n},$$

where $\mathbf{F} = \bigcup_{\Delta \in \mathfrak{D}} \mathbf{F}_\Delta$ and $\mathbf{D} = \bigcup_{\Delta \in \mathfrak{D}} \mathbf{D}_\Delta$.

The Factoring Algorithm. Consider an algorithm \mathcal{B} that first tries to find a factor of n by computing $\gcd(\text{lc}(g), n)$ for all $g \in G$. Then it runs the BBRE algorithm \mathcal{A} on an arbitrary instance of the BBRE problem over R and records the sequence of queries that \mathcal{A} issues. For each $\Delta \in \mathfrak{D}$ the algorithm \mathcal{B} chooses a new random element $a \in R$, and computes $\gcd(\text{lc}(\Delta(a) \bmod G), n)$. There are at most $(m+t+2)(m+t+1)/2$ such polynomials and each can be evaluated using at most $m+1$ ring operations. Thus, in total \mathcal{B} chooses $O((m+t)^2)$ random elements and performs $O(m(m+t)^2)$ operations on R as well as $O((m+t)^2 + s)$ gcd computations on $\log_2(n)$ -bit numbers. Clearly, the success probability of \mathcal{B} is at least $\Pr[\mathbf{D}]$. \square

Note that univariate polynomial quotient rings of the form $\mathbb{Z}_n[X_1]/J$ for some ideal J in $\mathbb{Z}_n[X_1]$ are covered by Theorem 3 as a special case. A Gröbner basis for J can always be easily determined: If J is given by a single polynomial g we are done. In the case where J is described by a set of polynomials $\{g_1, \dots, g_s\}$, a unique polynomial g generating J can be computed as $g = \gcd(g_1, \dots, g_s)$. Furthermore, we can use the standard polynomial division algorithm (for univariate polynomials) to implement reduction modulo g .

Let $(n, t, G) \leftarrow \text{RGen}(\kappa)$ be a ring instance generator that on input of a security parameter κ (in unary representation) outputs the description (n, t, G) of a ring $R = \mathbb{Z}_n[X_1, \dots, X_t]/\langle G \rangle$, where n is

an integer consisting of at least two different primes, t specifies the number of indeterminates, and G is a Gröbner basis for the ideal $\langle G \rangle$. Note that the parameters n, t , as well as the Gröbner basis (i.e., $|G|$ and the individual elements of the Gröbner basis) may all depend on κ . Let us assume that addition, subtraction, multiplication, reduction modulo G as well as sampling random elements in the rings R takes polynomial-time in κ . Furthermore, let there exist a non-constant polynomial $q(\cdot)$ over \mathbb{N} such that for all κ and possible outputs $(n, t, G) \leftarrow \text{RGen}(\kappa)$ it holds that $\log_2(n) \geq q(\kappa)$. Then Theorem 3 provides a polynomial-time (in κ) reduction from finding a factor of n to the black-box ring extraction problem for the family of rings described by RGen .

5 Extending our Reduction to Product Rings

Our reduction naturally extends to product rings where at least one component is a ring already covered by Theorem 1 or Theorem 3. In particular, this includes product rings of the form $\mathbb{Z}_n \times \dots \times \mathbb{Z}_n$ which might be of special interest.

Theorem 4. *Let $R := R_1 \times \dots \times R_\ell$ be the direct product of finitely many rings where $R_1 = \mathbb{Z}_n[X_1, \dots, X_t]/J$ or $R_1 = \mathbb{Z}_n$. Let the integer n consist of at least two different prime factors and let the ideal J occurring in the latter case be given by a Gröbner basis $G = \{g_1, \dots, g_s\}$. Let \mathcal{A} be an algorithm for the BBRE problem that performs at most $m \leq |R|$ operations on R^σ . Assume that \mathcal{A} solves the BBRE problem with probability ϵ . Then there is an algorithm \mathcal{B} having white-box access to \mathcal{A} that finds a factor of n with probability at least*

$$\frac{\epsilon}{2} - \frac{1}{2n}$$

by running \mathcal{A} once and performing an additional amount of $O((m+t)^2)$ random choices and $O(m(m+t)^2)$ operations on R_1 as well as $O((m+t)^2 + s)$ gcd computations on $\log_2(n)$ -bit numbers (where in the case $R_1 = \mathbb{Z}_n$ the parameters t and s are set to zero).

Proof. Given $\sigma(x) \in R^\sigma$ where $x = (x_1, \dots, x_\ell)$ is uniformly chosen from $R = R_1 \times \dots \times R_\ell$ the algorithm \mathcal{A} finds x , i.e., all components x_1, \dots, x_ℓ , with probability ϵ . Thus, given $\sigma(x)$ it outputs an element (y_1, \dots, y_ℓ) such that $y_1 = x_1$ with probability at least ϵ . Furthermore, observe that choosing an element x uniformly at random from R is equivalent to choosing each component x_i uniformly at random from R_i . Thus, informally speaking, \mathcal{A} solves the BBRE problem over each component ring separately. Hence, we can simply apply the ideas from the proof of Theorems 1 and 3 to this case, namely to the component R_1 .

More precisely, we can just generalize the description of the original and the simulation game to product rings (which is a straightforward task) except for one modification: instead of making the computations in the simulation game independent of x , i.e., all components x_1, \dots, x_ℓ , we make them only independent of the component x_1 . That means, when determining encodings, the simulation oracle \mathcal{O}_{sim} still evaluates polynomials over R_2, \dots, R_ℓ with the given inputs x_2, \dots, x_ℓ exactly as \mathcal{O} does and only chooses new random elements for evaluating polynomials over R_1 . In this way, only the modification over R_1 can lead to a difference in the behaviour of \mathcal{O}_{sim} and \mathcal{O} and so we can define the event of a simulation failure as before.

The success probability of \mathcal{A} in the simulation game is upper bounded by the probability that it outputs x_1 which is at most $\frac{\epsilon}{n}$. The probability of a simulation failure can be bounded exactly as before.

Similarly, the factoring algorithm \mathcal{B} runs \mathcal{A} on some instance of the BBRE problem over R and records the sequence of queries. The corresponding difference polynomials can be seen as

polynomials over R_1 and so \mathcal{B} does the same steps as the factoring algorithm described in the previous proof of Theorem 1 or 3 depending on whether $R_1 = \mathbb{Z}_n$ or $R_1 = \mathbb{Z}_n[X_1, \dots, X_t]/J$.

Hence, we can conclude that we obtain the same relation between the success probability of \mathcal{A} and \mathcal{B} and the same number of additional operations \mathcal{B} performs as in the case $R = R_1$. \square

Acknowledgements. We would like to thank Roberto Avanzi, Lothar Gerritzen, and Gregor Leander for helpful discussions.

References

- [AL94] William Adams and Philippe Loustau. *An introduction to Gröbner bases*. Graduate Studies in Math. Vol. 3. Oxford University Press, 1994.
- [Bac84] Eric Bach. Discrete logarithms and factoring. Technical Report UCB/CSD-84-186, EECS Department, University of California, Berkeley, Jun 1984.
- [BL96] Dan Boneh and Richard J. Lipton. Algorithms for black-box fields and their application to cryptography (extended abstract). In Neal Koblitz, editor, *Advances in Cryptology — CRYPTO '96, 16th Annual International Cryptology Conference*, volume 1109 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 1996.
- [Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal)*. PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965. (English translation in *Journal of Symbolic Computation*, 2004).
- [dB88] Bert den Boer. Diffie-Hellman is as strong as discrete log for certain primes. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO '88, 8th Annual International Cryptology Conference*, volume 403 of *Lecture Notes in Computer Science*, pages 530–539. Springer, 1988.
- [ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.
- [Gri99] Pierre A. Grillet. *Algebra (Pure and Applied Mathematics)*. John Wiley & Sons Inc., 1999.
- [Lan02] Serge Lang. *Algebra*. Springer-Verlag, 2002.
- [LR06] Gregor Leander and Andy Rupp. On the equivalence of RSA and factoring regarding generic ring algorithms. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology — ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security*, volume 4284 of *Lecture Notes in Computer Science*, pages 241–251. Springer, 2006.
- [Mau94] Ueli Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete algorithms. In Yvo Desmedt, editor, *Advances in Cryptology — CRYPTO '94, 14th Annual International Cryptology Conference*, volume 839 of *Lecture Notes in Computer Science*, pages 271–281. Springer, 1994.
- [Mau05] Ueli Maurer. Abstract models of computation in cryptography. In Nigel P. Smart, editor, *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.
- [McD74] Bernard R. McDonald. *Finite Rings with Identity*. Marcel Dekker, New York, 1974.
- [MR07] Ueli Maurer and Dominik Raub. Black-box extension fields and the inexistence of field-homomorphic one-way permutations. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security*, volume 4833 of *Lecture Notes in Computer Science*, pages 427–443. Springer, 2007.
- [MW98] Ueli M. Maurer and Stefan Wolf. Lower bounds on generic algorithms in groups. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1403 of *Lecture Notes in Computer Science*, pages 72–84. Springer, 1998.
- [Nec94] V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
- [Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.
- [Pol78] J. M. Pollard. Monte carlo methods for index computation mod p . *Mathematics of Computation*, 32:918–924, 1978.
- [Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997.

[Sho06] Victor Shoup. Sequences of games: A tool for taming complexity in security proofs. IACR eprint report 2004/332, 2006.

A Detailed Proof of Lemma 8

In this section we provide a proof for Lemma 8. Our proof is based on an alternative but equivalent definition of Gröbner bases using the notion of syzygies. We partly make use of definitions and theorems given in Chapters 3.2 and 4.2 of [AL94].

Throughout this section let $A = D[X_1, \dots, X_t]$ where $D = \mathbb{Z}_n$ is a Noetherian ring. Let $I = \langle f_1, \dots, f_s \rangle$ be an ideal of A . Consider the A -module homomorphism

$$\begin{aligned} \phi : A^s &\rightarrow I \\ (h_1, \dots, h_s) &\mapsto \sum_{i=1}^s h_i f_i \end{aligned}$$

Then it holds that $I \cong A^s / \ker(\phi)$. Based on ϕ a syzygy is defined as follows:

Definition 6 (Syzygy). *The kernel of the map ϕ is called the syzygy module of the $1 \times s$ matrix $[f_1 \dots f_s]$ and is denoted by $\text{Syz}(f_1 \dots f_s)$. An element $(h_1, \dots, h_s) \in \text{Syz}(f_1 \dots f_s)$ is called a syzygy of $[f_1 \dots f_s]$ and satisfies*

$$h_1 f_1 + \dots + h_s f_s = 0.$$

Definition 7 (Homogeneous Syzygy). *Let power products $\mathcal{X}_1, \dots, \mathcal{X}_s$ and non-zero elements $c_1, \dots, c_s \in D$ be given. For a power product \mathcal{X} , we call a syzygy $h = (h_1, \dots, h_s) \in \text{Syz}(c_1 \mathcal{X}_1, \dots, c_s \mathcal{X}_s)$ homogeneous of degree \mathcal{X} if $\text{lt}(h_i) = h_i$, thus h_i is a term itself, and $\mathcal{X}_i \text{lm}(h_i) = \mathcal{X}$ for all i such that $h_i \neq 0$.*

As A is a Noetherian ring, $\text{Syz}(c_1 \mathcal{X}_1, \dots, c_s \mathcal{X}_s)$ has a finite generating set of homogeneous syzygies. Moreover, by [AL94] we have the following equivalent characterization of Gröbner bases:

Theorem 5 (Theorem 4.2.3 [AL94]). *Let $G = \{g_1, \dots, g_s\}$ be a set of non-zero polynomials in A . Let \mathfrak{B} be a homogeneous generating set for $\text{Syz}(\text{lt}(g_1) \dots \text{lt}(g_s))$. Then G is a Gröbner basis for the ideal $\langle g_1, \dots, g_s \rangle$ if and only if for all $(h_1, \dots, h_s) \in \mathfrak{B}$ we have*

$$\sum_{i=1}^s h_i g_i \xrightarrow{G}_+ 0.$$

Our proof for Lemma 8 will essentially be based on the above theorem. However, before we can actually give this proof we need to introduce two of auxiliary lemmas.

Lemma 9. *Let $\{f_1, \dots, f_s\}$ be a basis for an ideal I of A . For $1 \leq i \leq s$ let the leading term of f_i be denoted by $\text{lt}(f_i) = c_i \mathcal{X}_i$. If all leading coefficients c_i are units in D then*

$$\mathfrak{B}_{\text{Syz}(\text{lt}(f_1), \dots, \text{lt}(f_s))} = \left\{ S_{ij} = S_{ji} = \frac{\mathcal{X}_{ij}}{c_i \mathcal{X}_i} e_i - \frac{\mathcal{X}_{ij}}{c_j \mathcal{X}_j} e_j \mid 1 \leq i < j \leq s \right\},$$

where e_1, \dots, e_s form the standard basis for A^s and $\mathcal{X}_{ij} = \text{lcm}(\mathcal{X}_i, \mathcal{X}_j)$, is a generating set for $\text{Syz}(\text{lt}(f_1), \dots, \text{lt}(f_s))$ of homogeneous syzygies.

Proof. First of all, if $i \neq j$ then $\frac{\mathcal{X}_{ij}}{c_i \mathcal{X}_i} e_i - \frac{\mathcal{X}_{ij}}{c_j \mathcal{X}_j} e_j$ is a syzygy of $\text{Syz}(\text{lt}(f_1), \dots, \text{lt}(f_s))$ as $\frac{\mathcal{X}_{ij}}{c_i \mathcal{X}_i} \text{lt}(f_i) - \frac{\mathcal{X}_{ij}}{c_j \mathcal{X}_j} \text{lt}(f_j) = 0$. Furthermore, the non-zero polynomials $\frac{\mathcal{X}_{ij}}{c_i \mathcal{X}_i}$ and $\frac{\mathcal{X}_{ij}}{c_j \mathcal{X}_j}$ are terms and the syzygy $S_{ij} = S_{ji}$ is homogeneous of degree \mathcal{X}_{ij} as $\mathcal{X}_i \cdot \frac{\mathcal{X}_{ij}}{c_i \mathcal{X}_i} = \mathcal{X}_{ij} = \mathcal{X}_j \cdot \frac{\mathcal{X}_{ij}}{c_j \mathcal{X}_j}$.

Therefore we need to prove that $\mathfrak{B}_{\text{Syz}(\text{lt}(f_1), \dots, \text{lt}(f_s))}$ is a basis of the ideal $\text{Syz}(\text{lt}(f_1), \dots, \text{lt}(f_s))$. Let $h = (h_1, \dots, h_s) \in \text{Syz}(\text{lt}(f_1), \dots, \text{lt}(f_s))$ then

$$\sum_{i=1}^s h_i \text{lt}(f_i) = \sum_{i=1}^s c_i \cdot \mathcal{X}_i \cdot h_i = \sum_{i=1}^s c_i \cdot \mathcal{X}_i \left(\sum_{k=1}^{d_i} a_k^{(i)} \mathcal{X}_k^{(i)} \right) = 0$$

for $h_i = \sum_{k=1}^{d_i} a_k^{(i)} \mathcal{X}_k^{(i)}$. Let \mathcal{X} be any power product (in the variables X_1, \dots, X_t). Then the coefficient of \mathcal{X} in the polynomial $\sum_{i=1}^s c_i \cdot \mathcal{X}_i \cdot h_i$ must be zero. Let

$$\{\mathcal{Y}_1, \dots, \mathcal{Y}_d\} = \bigcup_{i=1}^s \left\{ \mathcal{X}_k^{(i)} \mid 1 \leq k \leq d_i \right\}$$

with $\mathcal{Y}_1 < \dots < \mathcal{Y}_d$. Then

$$h_i = \sum_{k=1}^{d_i} a_k^{(i)} \mathcal{X}_k^{(i)} = \sum_{\ell=1}^d b_\ell^{(i)} \mathcal{Y}_\ell \text{ with } \begin{cases} b_\ell^{(i)} = a_k^{(i)} & \text{if } \mathcal{Y}_\ell = \mathcal{X}_k^{(i)} \text{ for some index } 1 \leq k \leq d_i \\ b_\ell^{(i)} = 0 & \text{else} \end{cases}.$$

Moreover, let \mathcal{X} be a power product and $1 \leq m_i \leq d$ be the index such that $\mathcal{X}_i \cdot \mathcal{Y}_{m_i} = \mathcal{X}$ for $1 \leq i \leq s$. Then $\sum_{i=1}^s c_i \cdot b_{m_i} = 0$ and we have

$$\begin{aligned} (b_{m_1} \cdot \mathcal{Y}_{m_1}, \dots, b_{m_s} \cdot \mathcal{Y}_{m_s}) &= b_{m_1} \cdot \mathcal{Y}_{m_1} e_1 + \dots + b_{m_s} \cdot \mathcal{Y}_{m_s} e_s \\ &= b_{m_1} \cdot c_1 \cdot \frac{\mathcal{Y}_{m_1} \cdot \mathcal{X}_1}{c_1 \cdot \mathcal{X}_1} e_1 + \dots + b_{m_s} \cdot c_s \cdot \frac{\mathcal{Y}_{m_s} \mathcal{X}_s}{c_s \mathcal{X}_s} e_s \\ &= b_{m_1} \cdot c_1 \cdot \frac{\mathcal{X}}{\mathcal{X}_{12}} \left(\frac{\mathcal{X}_{12}}{c_1 \cdot \mathcal{X}_1} e_1 - \frac{\mathcal{X}_{12}}{c_2 \cdot \mathcal{X}_2} e_2 \right) \\ &\quad + (b_{m_1} \cdot c_1 + c_2 \cdot b_{m_2}) \cdot \frac{\mathcal{X}}{\mathcal{X}_{23}} \left(\frac{\mathcal{X}_{23}}{c_2 \cdot \mathcal{X}_2} e_2 - \frac{\mathcal{X}_{23}}{c_3 \cdot \mathcal{X}_3} e_3 \right) \\ &\quad + \dots + \left(\sum_{j=1}^{s-1} c_j \cdot b_{m_j} \right) \cdot \frac{\mathcal{X}}{\mathcal{X}_{s-1s}} \left(\frac{\mathcal{X}_{s-1s}}{c_{s-1} \cdot \mathcal{X}_{s-1}} e_{s-1} - \frac{\mathcal{X}_{s-1s}}{c_s \cdot \mathcal{X}_s} e_s \right) \\ &\quad + \left(\sum_{j=1}^s c_j \cdot b_{m_j} \right) \cdot \frac{\mathcal{X}}{c_s \mathcal{X}_s} e_s \\ &= \sum_{i=1}^{s-1} \left(\sum_{j=1}^i c_j \cdot b_{m_j} \frac{\mathcal{X}}{\mathcal{X}_{ii+1}} \right) S_{ii+1} \end{aligned}$$

□

Definition 8 (Saturation). For power products $\mathcal{X}_1, \dots, \mathcal{X}_s$ and a subset $J \subseteq \{1, \dots, s\}$ we set $\mathcal{X}_J = \text{lcm}(\mathcal{X}_j \mid j \in J)$. We say that J is saturated with respect to $\mathcal{X}_1, \dots, \mathcal{X}_s$ provided that for all $j \in \{1, \dots, s\}$ the index j is an element of J if \mathcal{X}_j divides \mathcal{X}_J . We call the subset $J' \subseteq \{1, \dots, s\}$ consisting of all j such that \mathcal{X}_j divides \mathcal{X}_J the saturation of J . We denote with $\text{Sat}(\mathcal{X}_1, \dots, \mathcal{X}_s)$ all saturated subsets of $\{1, \dots, s\}$ with respect to $\mathcal{X}_1, \dots, \mathcal{X}_s$.

Lemma 10. Let $G = \{g_1, \dots, g_s\}$ be a Gröbner basis for an ideal I of A and let $n = \prod_{\ell=1}^k p_\ell^{e_\ell}$ be the prime power decomposition of the characteristic n of D . For $1 \leq i \leq s$ let the leading term of

g_i be denoted by $\text{lt}(g_i) = c_i \mathcal{X}_i$. If all leading coefficients c_i are units in D then for each $1 \leq \ell \leq k$ the set

$$\begin{aligned} \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})} &= \{(S_{ij}, 0) \mid S_{ij} \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s))}\} \\ &\cup \left\{ \frac{p_\ell^{e_\ell} \mathcal{X}_J}{c_j \mathcal{X}_j} e_j - \mathcal{X}_J e_{s+1} \mid J \in \text{Sat}(\mathcal{X}_1, \dots, \mathcal{X}_s) \text{ and some } j \in J \right\} \end{aligned}$$

is a homogeneous generating set for $\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})$.

Proof. Certainly, each element of $\mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})}$ is vector of $\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})$ as

$$(S_{ij}, 0) \cdot (\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})^t = S_{ij} \cdot (\text{lt}(g_1), \dots, \text{lt}(g_s))^t = 0$$

and

$$\frac{p_\ell^{e_\ell} \mathcal{X}_J}{c_j \mathcal{X}_j} e_j - \mathcal{X}_J e_{s+1} \cdot (\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})^t = \frac{p_\ell^{e_\ell} \mathcal{X}_J}{c_j \mathcal{X}_j} \text{lt}(g_j) - \mathcal{X}_J \cdot p_\ell^{e_\ell} = 0$$

for $J \in \text{Sat}(\mathcal{X}_1, \dots, \mathcal{X}_s)$ and some $j \in J$. Furthermore the element $(S_{ij}, 0)$ is homogeneous of degree \mathcal{X}_{ij} and $\frac{p_\ell^{e_\ell} \mathcal{X}_J}{c_j \mathcal{X}_j} e_j - \mathcal{X}_J e_{s+1}$ is a homogeneous syzygy of degree \mathcal{X}_j .

Let $h = (h_1, \dots, h_{s+1}) \in \text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})$. If $h_{s+1} = 0$ then (h_1, \dots, h_s) is an element of $\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s))$, hence by Lemma 9 $(h_1, \dots, h_s, 0)$ is finite linear combination of

$$\{(S_{ij}, 0) \mid S_{ij} \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s))}\}.$$

Otherwise, using the notation from the proof Lemma 9, if $h_{s+1} \neq 0$ then for a power product \mathcal{X} , the coefficient of \mathcal{X} in the polynomial

$$h \cdot (\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})^t = \sum_{i=1}^s \text{lt}(g_i) h_i + p_\ell^{e_\ell} h_{s+1} = \sum_{i=1}^s c_i \mathcal{X}_i \left(\sum_{k=1}^{d_i} a_k^{(i)} \mathcal{X}_k^{(i)} \right) + p_\ell^{e_\ell} \left(\sum_{k=1}^{d_{s+1}} a_k^{(i)} \mathcal{X}_k^{(i)} \right)$$

must be zero for $h_i = \sum_{k=1}^{d_i} a_k^{(i)} \mathcal{X}_k^{(i)}$ and $1 \leq i \leq s+1$.

Let

$$\{\mathcal{Y}_1, \dots, \mathcal{Y}_d\} = \bigcup_{i=1}^{s+1} \left\{ \mathcal{X}_k^{(i)} \mid 1 \leq k \leq d_i \right\}$$

with $\mathcal{Y}_1 < \dots < \mathcal{Y}_d$. Then

$$h_i = \sum_{k=1}^{d_i} a_k^{(i)} \mathcal{X}_k^{(i)} = \sum_{\ell=1}^d b_\ell^{(i)} \mathcal{Y}_\ell \text{ with } \begin{cases} b_\ell^{(i)} = a_k^{(i)} & \text{if } \mathcal{Y}_\ell = \mathcal{X}_k^{(i)} \text{ for some index } 1 \leq k \leq d_i \\ b_\ell^{(i)} = 0 & \text{else} \end{cases},$$

for $1 \leq i \leq s+1$.

Furthermore, let \mathcal{X} be a power product and $1 \leq m_i \leq d$ be the index such that $\mathcal{X}_i \cdot \mathcal{Y}_{m_i} = \mathcal{X}$ for $1 \leq i \leq s+1$ with $\mathcal{X}_{s+1} = 1$. Then $(b_{m_1} \mathcal{Y}_{m_1}, \dots, b_{m_{s+1}} \mathcal{Y}_{m_{s+1}})$ is a homogeneous syzygy of degree \mathcal{X} . Moreover, we assume that $b_{m_{s+1}} \neq 0$ and consider the index set $J' = \{j \mid b_{m_j} \neq 0\} \setminus \{s+1\}$. Let $J \in \text{Sat}(\mathcal{X}_1, \dots, \mathcal{X}_s)$ such that $J' \subseteq J$. Then we fix an index $d \in J$ such that $\frac{p_\ell^{e_\ell} \mathcal{X}_J}{c_d \mathcal{X}_d} e_d - \mathcal{X}_J e_{s+1} \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})}$. It follows that

$$\begin{aligned} & (b_{m_1} \mathcal{Y}_{m_1}, \dots, b_{m_{s+1}} \mathcal{Y}_{m_{s+1}}) - (-b_{m_{s+1}}) \cdot \left(\frac{p_\ell^{e_\ell} \mathcal{X}_J}{c_d \mathcal{X}_d} e_d - \mathcal{X}_J e_{s+1} \right) \cdot \frac{\mathcal{X}}{\mathcal{X}_J} \\ &= \left(b_{m_1} \mathcal{Y}_{m_1}, \dots, b_{m_d} \mathcal{Y}_{m_d} + \frac{b_{m_{s+1}} \cdot p_\ell^{e_\ell} \mathcal{X}_J}{c_d \mathcal{X}_d} \cdot \frac{\mathcal{X}}{\mathcal{X}_J}, b_{m_{d+1}} \mathcal{Y}_{m_{d+1}}, \dots, b_{m_s} \mathcal{Y}_{m_s}, b_{m_{s+1}} \mathcal{Y}_{m_{s+1}} - b_{m_{s+1}} \mathcal{X}_J \cdot \frac{\mathcal{X}}{\mathcal{X}_J} \right) \\ &= \left(b_{m_1} \mathcal{Y}_{m_1}, \dots, \left(b_{m_d} + \frac{b_{m_{s+1}} \cdot p_\ell^{e_\ell}}{c_d} \right) \mathcal{Y}_{m_d}, b_{m_{d+1}} \mathcal{Y}_{m_{d+1}}, \dots, b_{m_s} \mathcal{Y}_{m_s}, 0 \right) \end{aligned}$$

is a homogeneous syzygy with zero in the $(s + 1)$ -th coordinate and a linear combination of the set $\{(S_{ij}, 0) \mid S_{ij} \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s))}\}$ by Lemma 9. \square

Now, we are able to actually prove Lemma 8 from Section 4.2.

Proof (Lemma 8). By Theorem 5 the set $G_\ell = \{p_\ell^{e_\ell}, g_1, \dots, g_s\}$ is a Gröbner basis for the ideal $J_i = \langle p_\ell^{e_\ell}, g_1, \dots, g_s \rangle$ if and only if for each element $h \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})}$ the relation $h \cdot (g_1, \dots, g_s, p_\ell^{e_\ell})^t \xrightarrow{G_\ell}_+ 0$ holds.

Let $h \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s), p_\ell^{e_\ell})}$, then by Lemma 10 either h is an element of

$$\{(S_{ij}, 0) \mid S_{ij} \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s))}\}$$

or h is an element of

$$\left\{ \frac{p_\ell^{e_\ell} \mathcal{X}_J}{\text{lt}(g_j)} e_j - \mathcal{X}_J e_{s+1} \mid J \in \text{Sat}(\text{lm}(g_1), \dots, \text{lm}(g_s)) \text{ and some } j \in J \right\}.$$

For the first case, we observe that

$$h \cdot (g_1, \dots, g_s, p_\ell^{e_\ell})^t = \sum_{i=1}^s h_i g_i = (h_1, \dots, h_s) \cdot (g_1, \dots, g_s)^t$$

with $(h_1, \dots, h_s) \in \mathfrak{B}_{\text{Syz}(\text{lt}(g_1), \dots, \text{lt}(g_s))}$. Since G is a Gröbner basis for the ideal I it follows by Theorem 5 that $h \cdot (g_1, \dots, g_s, p_\ell^{e_\ell})^t \xrightarrow{G}_+ 0$ and thus $(g_1, \dots, g_s, p_\ell^{e_\ell})^t \xrightarrow{G_\ell}_+ 0$.

In the other case, we have

$$h = \frac{p_\ell^{e_\ell} \mathcal{X}_J}{\text{lt}(g_j)} e_j - \mathcal{X}_J e_{s+1}$$

for some $J \in \text{Sat}(\text{lm}(g_1), \dots, \text{lm}(g_s))$ and some $j \in J$. Furthermore,

$$h \cdot (g_1, \dots, g_s, p_\ell^{e_\ell})^t = \frac{p_\ell^{e_\ell} \mathcal{X}_J}{\text{lt}(g_j)} g_j - p_\ell^{e_\ell} \mathcal{X}_J = \frac{p_\ell^{e_\ell} \mathcal{X}_J}{\text{lt}(g_j)} (\text{lt}(g_j) + g') - p_\ell^{e_\ell} \mathcal{X}_J = \frac{p_\ell^{e_\ell} \mathcal{X}_J}{\text{lt}(g_j)} g' = \frac{g' \cdot \mathcal{X}_J}{\text{lt}(g_j)} p_\ell^{e_\ell}$$

for $g_j = \text{lt}(g_j) + g'$, thus $h \cdot (g_1, \dots, g_s, p_\ell^{e_\ell})^t \xrightarrow{\{p_\ell^{e_\ell}\}}_+ 0$ implying that $h \cdot (g_1, \dots, g_s, p_\ell^{e_\ell})^t \xrightarrow{G_\ell}_+ 0$. \square