Solving polynomial systems over non-fields and applications to modular polynomial factoring

Sayak Chakrabarti * Ashish Dwivedi † Nitin Saxena ‡

Abstract

We study the problem of solving a system of m polynomials in n variables over the ring of integers modulo a prime-power p^k. The problem over finite fields is well studied in varied parameter settings. For small characteristic $p = 2$, Lokshtanov et al. (SODA'17) initiated the study, for degree $d = 2$ systems, to improve the exhaustive search complexity of $O(2^n) \cdot \mathsf{poly}(m, n)$ to $O(2^{0.8765n}) \cdot \mathsf{poly}(m, n)$; which currently is improved to $O(2^{0.6943n}) \cdot \mathsf{poly}(m, n)$ in Dinur (SODA'21). For large p but constant n, Huang and Wong (FOCS'96) gave a randomized $\mathsf{poly}(d, m, \log p)$ time algorithm. Note that for growing n, system-solving is known to be *intractable* even with $p = 2$ and degree $d = 2$.

We devise a randomized $\mathsf{poly}(d, m, \log p)$-time algorithm to find a root of a given system of m integral polynomials of degrees bounded by d, in n variables, modulo a prime power p^k; when $n + k$ is constant. In a way, we extend the efficient algorithm of Huang and Wong (FOCS'96) for system-solving over Galois fields (i.e., characteristic p) to system-solving over Galois *rings* (i.e., characteristic p^k); when $k > 1$ is constant. The challenge here is to find a lift of *singular* $\mathbb{F}_p$-roots (exponentially many); as there is no efficient general way known in algebraic-geometry for resolving singularities.

Our algorithm has applications to factoring univariate polynomials over Galois rings. Given $f \in \mathbb{Z}[x]$ and a prime-power p^k ($k \geq 2$), finding factors of $f \bmod p^k$ has a curious state-of-the-art. It is solved for large k by p-adic factoring algorithms (von zur Gathen, Hartlieb, ISSAC'96); but unsolved for small k. In particular, no nontrivial factoring method is known for $k \geq 5$ (Dwivedi, Mittal, Saxena, ISSAC'19). One issue is that degree-δ factors of $f(x) \bmod p^k$ could be *exponentially* many, as soon as $k \geq 2$. We give the first randomized $\mathrm{poly}(\deg(f), \log p)$-time algorithm to find a degree-δ factor of $f(x) \bmod p^k$, when $k + \delta$ is constant. Our method has potential application in algebraic coding theory. In particular, extending algebraic geometric and Reed-Solomon codes to Galois rings could enable new and improved bounds on their underlying efficiency parameters.

2012 ACM CCS concept: Theory of computation– Algebraic complexity theory; Theory of computation– Problems, reductions and completeness; Computing methodologies– Algebraic algorithms; Computing methodologies– Hybrid symbolic-numeric methods.
Keywords: polynomial, factors, prime powers, efficient, roots, Nullstellensatz.

Contents

*Department of CSE, Indian Institute of Technology, Kanpur, India, Email: sayaksc@gmail.com
†Department of CSE, Indian Institute of Technology, Kanpur, India, Email: ashish02dwivedi@gmail.com
‡Department of CSE, Indian Institute of Technology, Kanpur, India, Email: nitin@cse.iitk.ac.in

1 Introduction

Deciding the existence of a common root of a system of multivariate polynomial equations (Hilbert's Nullstellensatz or HN) is a fundamental problem in algebraic geometry [CLO13]. In this paper, we study the search version of this problem, named as Search Hilbert's Nullstellensatz (SHN), which is about finding a common root of the system from the underlying coefficient ring. Over finite fields (characteristic p), the problem is well-studied and very important in cryptography [Din21a, KPG99, Pat96] even for $p = 2$ and systems of degree $d = 2$.

The problem has been studied mainly in two parameter settings over finite fields. In small characteristic $p = 2$, there has been a flurry of work [BFSS13, LPT+17, BKW19, Din21b, BDT21] to improve the brute force time complexity from $O(2^n) \cdot \mathsf{poly}(m, n)$ (for m quadratic polynomial equations in n variables) to finally $O(2^{0.6943n}) \cdot \mathsf{poly}(m, n)$ by Dinur [Din21b] which even outperforms $O(2^{0.792n}) \cdot \mathsf{poly}(n)$ complexity (where $m = n$) for random system of equations (Bardet et al. [BFSS13]).

For large p but constant n, Huang and Wong [HW99] gave an efficient randomized $\mathsf{poly}(d, m, \log p)$ time algorithm to find a common zero of a system of m-many degree-d polynomials in n variables. The decision version of the problem was derandomized by Kayal [Kay05] in same time complexity. Note that the problem is NP-complete for *un*bounded number of variables n, even if $p = 2$ and $d = 2$ [EK90, GGL08]. For growing prime p, the problem of detecting roots is *intractable* even for univariate polynomials ($n = 1$) [BCR16] and bivariate polynomials ($n = 2$) [vzGKS96] provided the encoding of the polynomials is "sparse".

Extending Huang and Wong [HW99], *we study the problem SHN for constant n but generalized over the ring of integers modulo a given prime power p^k ($k \geq 2$), called Galois rings.*

Galois rings are important in the study of algebraic codes [HKC+94]. Efficiently solving polynomial systems in Galois rings may be fruitful in the study of such codes. E.g., univariate root finding [BLQ13] has application in Guruswami-Sudan type list-decoding in Galois rings.

When $k > 1$, the classical methods of algebraic-geometry fail; which is perhaps why work in this setting is sparse. Starting $k = 2$, we are unaware of any efficient way to solve SHN; and there is no analogue of famous theorems like Hilbert's Nullstellensatz (see [Bro87, GS20] to read more about the rich machinery).

To understand the difficulty, consider a system with just one polynomial f such that $f(y_1, \ldots, y_n) =: \varphi^e \bmod p$, for *ramification* $e > 1$ and φ being *absolutely* irreducible (i.e., irreducible over all extension fields of $\mathbb{F}_p$). In this case, φ must have exponentially many (in $\log p$) roots and all those are *singular* roots of $f \bmod p$. So, there is no easy way to determine which root will lift, and which root will not, to modulo p^2 (i.e., they *ramify* in a complicated way).

Example 1. *$f(x, y) := (x - y)^2 + p \bmod p^2$. Thus, $f \bmod p$ has p roots (=exponential in $\log p$); but, none of them lifts $\bmod p^2$. So, $f(x, y)$ has no root $\bmod p^2$.*

Example 2. *Perturb the above example only slightly to $f(x, y) := (x - y)^2 + px \bmod p^2$. Again, $f \bmod p$ has p roots. However, now $(0, 0)$ is the unique root that lifts $\bmod p^2$. So, finding a random root modulo p and trying to lift it, does not work.*

This problem arises due to the existence of *singular* points (the points which are roots of the polynomial as well as all of its first order derivatives). Non-singular points can be lifted to modulo any power of p, in an analog to Hensel's lifting [Hen18]. However, quite like Hensel lifting fails in the case of ramification degree $e > 1$, lifting to higher powers of p fails for singular points. Thus, the problem of separating out the singular points and converting them to non-singular points in a different system, *desingularization* or *resolution of singularities*, has been a well-studied problem. It asks whether any algebraic variety `V` has a resolution which is a non-singular variety `H` such that the non-singular points of `H` can be birationally mapped to points of `V`. It is solved over characteristic 0 fields [Hir64], while it is still open for finite fields [Hau10]. We algorithmically circumvent the geometric obstruction of singular points and make the first progress towards SHN_{p^k}:

Theorem 1 (Informal)**.** *Let $n + k$ be constant. Given a system of m integral polynomials in n variables $\bmod p^k$, of degree at most d, we find a common root to this system in randomized $\mathsf{poly}(d, m, \log p)$ time.*

We thereby make progress towards another open problem of finding $\mathbb{Z}_p$ (i.e., p-adic) roots of a system of polynomial equations. [DS20] showed a bound on k for which roots of univariate polynomials modulo p^k correspond to unique p-adic roots. [RZ22] gave a more refined bound on k but for trinomial univariate polynomials. [CS23, Chi21] gave bounds for multivariate polynomials of the form $k = d^{2^{O(n)}}$. We find roots modulo p^k, while finding roots for large enough k (see [Gre66]) can solve the problem of finding p-adic roots.

Application to factoring in $\mathbb{Z}/\langle p^k \rangle$: Factoring a univariate polynomial over finite fields have many efficient algorithms known [Ber67, CZ81, Kal92, KU11, vzGP01] and have found many applications in mathematics and computing [FS15, Kal92, LN94, Sud97, vzGP01]. We consider the following (Galois) ring generalization of this question ($k > 1$):

Given $f \in \mathbb{Z}[x]$ and a prime power p^k, can we find a non-trivial factor of degree $\delta < \deg(f)$ in randomized $\mathsf{poly}(\deg(f), k \log p)$-time?

Though this problem is studied since the time of Hensel [Hen18] and it finds a section in many textbooks on elementary number theory [NZM13], yet there is no efficient algorithm known. The issue arises as $f \bmod p^k$ may possess *exponentially* many factors (in $\log p$); for e.g., $f = x^2 \bmod p^2$ has a factor $x + p\alpha$, for any $\alpha \in \{0, \ldots, p - 1\}$. This happens because the ring $\mathbb{Z}/\langle p^k \rangle$ is not a unique factorization domain. The following example illustrates the difficulty of lifting.

Example 3. *Let* $f = x^2 + p^2$ *and* $(p, k) := (5, 3)$. *The factorization* $f = x \cdot x \bmod p$ *lifts to* p *factorizations mod* p^2*, as discussed above. But* only *the factorization* $f = (x + 10) \cdot (x + 15) \bmod p^2$ *lifts to mod* p^3.

This example raises the question: How to efficiently determine which factorization (out of exponentially many) will lift to higher precision?

Hensel's lemma efficiently guarantees factoring when $f \bmod p$ has two *co-prime* factors. Thus, the hard case is to factor f which is power of an irreducible modulo p (as in the example above). Interestingly, in the hard case, using an extension of Hensel's lemma [BS86, vzGH98], one can solve the problem when k is large i.e., p^k does not divide the discriminant $\text{disc}(f)$ of f. In this case, [CL01, vzGH98] show that irreducible factors of $f \bmod p^k$ correspond to unique p-adic irreducible factors, which we get via efficient p-adic factoring algorithms [CG00, Chi87, Chi94, GNP12].

Thus, the major open question in factoring $f \bmod p^k$ is when k is *constant*. The main issue with the small k case is that a p-adic irreducible factor could become reducible modulo p^k and so factoring over p-adics does not help. Below is an example from [vzGH96]:

Example 4. $x^2 + 3^k$ *is irreducible mod* 3^{k+1} *and so over* $\mathbb{Q}_3$*, but is reducible mod* 3^k.

Currently, the best known methods [DMS21, Săl05] to find non-trivial factors of $f \bmod p^k$, for small k, require $k \leq 4$. On the other hand, degree $\delta = 1$ factors (i.e., actual roots) can be computed efficiently for any k [BLQ13, Pan95]. So, there seems to be a trade-off between the output-degree δ and input-exponent k. In this paper, we extend the constant k regime, to get low-degree factors of $f \bmod p^k$.

Theorem 2 (Informal)**.** *Let* $k + \delta$ *be constant. We give the first randomized poly-time algorithm to find a degree* δ *factor of given* $f(x) \bmod p^k$*. In particular, we can efficiently factorize constant-degree polynomials into irreducibles* $\bmod p^k$.

This is the first work to efficiently find a *ramified* factor (an irreducible factor which is not irreducible modulo p) for any constant k– the difficult case of the problem. We achieve this by efficiently reducing the problem of finding a constant-degree factor of $f \bmod p^k$, for constant k, to that of finding a root of system of polynomials in constant number of variables modulo p^k.

Other applications: Our methods have potential applications in algebraic coding theory. A breakthrough result of [HKC+94] showed that many known non-linear codes over finite fields, which have more codewords (higher information rate) than any linear code, are analogous to linear codes over the Galois ring $\mathbb{Z}/4\mathbb{Z}$. Thus many algebraic codes, e.g., algebraic-geometric (AG) [HKC+94, Wal99, Wal97] and Reed-Solomon (RS) [Arm05a, Arm05b] codes, were generalised over Galois rings to achieve better bounds on their parameters. Root finding of polynomial systems over Galois rings may help in the study of these generalised algebraic codes. Even univariate root finding over Galois rings [BLQ13] has applications to Guruswami-Sudan type list decoding algorithm for generalised RS codes.

AG codes and their higher dimensional generalisations (see [TV13]) are defined with respect to an absolutely irreducible variety, given by a system of polynomials over finite fields (see [HVLP98]). They are of significant interest as they beat Gilbert-Varshamov (GV) bound [Gop77, TVZ82]; a well-known lower bound on the maximum size of a code relative to a fixed rate and distance. After [HKC+94], these codes were generalised to various rings, in particular Galois rings [Wal99, Wal97]. These codes are defined with respect to a variety $\mathbf{X}$ over those Galois rings such that its associated variety $\mathbf{X}'$ over base field ($\mathbb{F}_p$) is absolutely irreducible (see [Wal99, Sec.5]). Given an arbitrary

system of polynomials, its variety over Galois ring may not have this property. Our method of system solving could be useful here as it returns a set of varieties over Galois ring (as ideals) which collectively contain all the zeros of the system and are *absolutely* irreducible modulo p.

Polynomial root-finding modulo 2^k has known applications in program analysis as real world programs perform modulo 2^k operations due to limited register size [ELS+14, MOS05, MOS07]. It also has applications in verifying equivalence of arithmetic datapaths [TKSG08].

Related works: Previously, only special systems of equations mod p^k have been studied. Chevalley-Warning and Ax-Katz type theorems [Kat09, MR75, SR14] are examples of this kind. These theorems study the conditions under which they can guarantee that a common root exists. Effective versions of some variant of Chevalley-Warning theorems help to get efficient quantum algorithms for important problems like, discrete logarithm and graph isomorphism [IR18].

Related to univariate polynomial factoring modulo p^k there are problems of univariate root finding and counting modulo p^k. These are of significant interest and finds applications in arithmetic-algebraic geometry [CS23, DH01, DS20, Zhu20, ZG03], factoring [CG00, Chi87, Chi94], coding theory [BLQ13, Săl05], and hyper/elliptic curve cryptography [Lau04]. [BLQ13] gave the first efficient randomized algorithm to find and count all the roots of $f(x)$ mod p^k. [DMS19] gave the first efficient *deterministic* algorithm to count all the roots of f mod p^k inspired by the ideals used in [CGRW19, KRRZ20] to store roots. These algorithms also give better understanding of *root-sets* modulo p^k (i.e., which subsets of $\mathbb{Z}/\langle p^k\rangle$ are zero-sets of some polynomial?). Their combinatorial properties are of significant interest in mathematics [Bha97, CP56, DM97, Mau01, Sie55].

1.1 Our results

The main result given in this paper gives an algorithm to find a common root of a system of polynomial equations. It can be summarized as the following theorem and proved in Section 3.3.

Theorem 1 (SHN_{p^k}). *Given a system of n-variate polynomials $f_1,\ldots,f_m \in \mathbb{Z}[z,\mathbf{x}]/\langle p^k,\varphi(z)\rangle$ of degrees at most d, for a prime p; and an irreducible polynomial $\varphi(z)\in\mathbb{F}_p[z]$ defining the Galois ring $\mathbb{G}:=\mathbb{Z}[z]/\langle p^k,\varphi(z)\rangle$. We can find a common root of the system in $\mathbb{G}$, in randomized $\mathsf{poly}(d^{c_{nk}},m,\deg(\varphi)\log p)$-time; where $c_{nk}\le (nk)^{O((nk)^2)}$.*

Remark 1. The following points can be noted about SHN_{p^k}:

- *Theorem 1 is efficient when $n+k$ is constant. Even if $k=1$, SHN is intractable for growing n.*
- *Theorem 1 efficiently extends the root-finding of [HW99] from Galois fields to Galois* rings *of characteristic p^k, for k constant. This extends the univariate results of [BLQ13, DMS19] to root-finding of constant-variate systems mod p^k.*
- *Theorem 1 resolves the open question asked in [DS20, RRZ21, Zhu20]— Efficiently find a point on a curve mod p^k, for fixed k. Our result is new for any constant $k>1$ and it utilizes the known root-finding method of [HW99] over finite fields ($k=1$).*

After a long series of efforts [BLQ13, CL01, DMS21, Kli97, KRRZ20, Săl05, Sir17, vzGH96, vzGH98], efficient modular factoring has remained elusive even for f mod p^5. In this direction, our result advances the state-of-the-art: to efficiently compute a *constant-degree* factor of f mod p^k, when k is *constant*. In particular, we can factor a *fixed* degree univariate polynomial into irreducibles.

Theorem 2 (Factoring)**.** *Given a univariate polynomial $f \in \mathbb{Z}[x]$ and a prime-power p^k, in binary, with k fixed. We can find a constant-degree factor g of $f \bmod p^k$ in randomized $\mathsf{poly}(\deg(f), \log p)$-time; or decide that none exists.*

Theorem 2 is proved is Section 3. The difficult case in factoring $f \bmod p^k$ happens when $f \bmod p$ has no two coprime factors: this obstructs Hensel lifting, which is the usual technique for lifting factors from $\mathbb{F}_p$ to $\mathbb{Z}/\langle p^k \rangle$ [Hen18]. For example, $f \equiv \varphi^e \bmod p$ for a $\varphi \in \mathbb{Z}[x]$ which is irreducible mod p. For such an f, we call e to be the *ramification-degree* of f. In fact, our proof method provides more general factors as summarized in the following corollary and proved in Section 3.

Corollary 3 (Low ramification factors)**.** *Given $f \in \mathbb{Z}[x]$ and prime-power p^k, with k constant. We can find a factor g of $f \bmod p^k$ in randomized $\mathsf{poly}(\deg(f), \log p)$-time, where the* ramification-degree *of g is at most a given constant; or decide that no such factor exists.*

Remark 2. We highlight the following points about factoring modular polynomials:

- *The brute-force approach takes time $p^{\Omega(k\delta)}$; which is clearly* exponential *(in $\log p$), even for fixed k and fixed ramification-degree δ.*
- *Thus, for constant k, our methods extend the results of [BLQ13, DMS19] from* un*ramified factors to ramified factors; albeit of 'low' ramification-degree.*
- *Our methods also extend [DMS21], from $k = 4$ to any fixed k, if the degree of f is fixed.*
- *Our algorithm is the first step towards factoring polynomials modulo p^k for any constant $k \geq 5$.*

1.2 Difficulty of the problems and techniques

As we have seen, solution to a system of polynomial equations in interesting fields/rings have been a well-studied problem. However, due to several difficulties to be discussed here, the problem has remained elusive in Galois rings. Here, we solve SHN_{p^k} for constant k.

The idea of univariate root-finding of Berthomieu et al. [BLQ13] fails even for a *single* integral bivariate $f(x_0, x_1) \bmod p^k$, for $k = 2$. For, in univariate case, the roots will have a p-adic expansion, which can be seen as a base-p expansion where each base-p precision coordinate is an element from 0 to p, denoted as a digit. [BLQ13] reduces the problem of finding each base-p digit to finding $\mathbb{F}_p$ roots, iterates over all these $\mathbb{F}_p$-roots to lift them. It could be shown that the number of iterations is bounded by d, the degree of the underlying univariate polynomial. This is not possible anymore with bivariate polynomial $f(x_0, x_1)$ as there could be as many as p (exponentially many) $\mathbb{F}_p$-roots, and bringing the complexity down to polynomial $\log p$ remains a challenge.

In this work, we resolve the above issue by taking the help of special data-structures in the form of ideals in $\mathbb{Z}_p$. Some points on these ideals are singular points, which might lead to roots too. However, since we work over the p-adic integers, lifting is not always possible for singular points. We handle singularity by 'carefully' modifying the ideals and lifting to $\mathbb{Z}_p$-ideals (see Lemma 4, Section 2.2). In a sense, we separate out the singular roots, the roots that do not lift, and modify the ideals such that they can henceforth be lifted– a method of *de-singularization* and extracting out the non-singular roots!

Another difficulty is getting a good bound on the number of ideals finally used. Dwivedi et al. [DMS19] bounds the number of ideals by $\deg(f)$, for any k, in their work on storing roots of

univariate polynomials. On the other hand, we bound the number of ideals by a quantity double exponential in $\ell := nk$ (Lemma 6). This forces us to assume ℓ constant, to get a practical algorithm.

The problem of finding the roots of these $\mathbb{Z}_p$-ideals, described above, poses several difficulties of its own. Especially, since exhaustively going over all the possible digits would make the algorithm exponential in $\log p$. For tackling this, we use [HW99] to solve for each base-p digit. This however is still insufficient, as given an $\mathbb{F}_p$-root, we need to lift them to p-adics; which is not always possible for even non-singular $\mathbb{F}_p$-roots. In order to perform this lift, we move to a birationally equivalent *hypersurface* (see Section 2.2), to represent the ideal containing several generator polynomials as having only one generator polynomial; and then consider the non-singular $\mathbb{F}_p$-roots of this single generator ideal (which lift by Hensel lifting!). In our proof method, the "project/lift" steps that the desired root takes, across the three rings, can be depicted simplistically as: $(\mathbb{Z}/p^k\mathbb{Z})^n \to (\mathbb{F}_p^k)^n \to (\mathbb{Z}_p^k)^n \to (\mathbb{Z}/p^k\mathbb{Z})^n$, while the previous attempts have been restricted only up to reductions to $\mathbb{F}_p$.

Example 5. *Let* $f := x^2 + 2 \bmod 3^2$. *Expanding root* $x =: y_0 + 3y_1$ *into the base-*3 *digits* y_i*'s*[1]*, we get* $f = (y_0^2 + 2) + 3 \cdot (2y_0y_1) \bmod 3^2$. *Consider the two base-*p *digits* $f_0 := y_0^2 + 2$ *and* $f_1 := 2y_0y_1$. *A common* $\mathbb{F}_3$*-root of* $\{f_0, f_1\}$ *is* $(y_0, y_1) = (1, 0)$. *But,* $1 + 3 \cdot 0 = 1$ *is not a root of* $f \bmod 3^2$.

This happens because we need $(y_0^2 + 2)/3 + (2y_0y_1) \equiv 0 \bmod 3$ *instead of* $f_1 = 0 \mod 3$*; which is not satisfied by* $(1, 0)$ *as* $(y_0^2 + 2)/3 \not\equiv 0 \bmod 3$*, while* $2y_0y_1 \equiv 0 \bmod 3$. *The correct strategy would be to first find* 3-adic $y_0 \in \mathbb{Z}_3$ *(not just over* $\mathbb{F}_3$*), and then fix* $y_1 = 0 \in \mathbb{Z}_3$. *This will enable us to correctly perform the division operation by* 3*; and take care of the 'carry-over' from* f_0 *to* f_1. *Finally* 3*-adic* y_i*'s give the correct* 3*-adic root* x.

As an application, we give an algorithm for finding certain factors of a univariate polynomial $f(x)$ modulo p^k. The main difficulty, with factoring $f(x) \bmod p^k$, is in finding ramified factors when k is small, i.e., $p^k \mid \mathrm{disc}(f)$. In addition, two irreducible factorizations can be very different, unlike the case of large k. For example, [vzGH96] shows that, $f = (x^2 + 243)(x^2 + 6) \bmod 3^6$ is an irreducible factorization of $x^4 + 249x^2$; while another irreducible factorization is $f = (x + 351)(x + 135)(x^2 + 243x + 249) \bmod 3^6$.

A connection due to [DMS21] shows: Finding (ramification-) degree-δ factors of $f(x) \bmod p^k$ reduces to root-finding of $E(y) \bmod \langle p^k, x^\ell \rangle$, for a special $E(y) \in (\mathbb{Z}[x])[y]$ and $\ell := \delta k$. This root-finding is still not easy to do; however, [DMS21] could do this if $k \leq 4$. Our new technique is to further reduce root-finding of $E(y) \bmod \langle p^k, x^\ell \rangle$ to root-finding of a system of ℓ-variate polynomials $\bmod p^k$ (i.e., SHN_{p^k}).

Practicalities. The most expensive part of the paper is where Algorithm 1 finds absolute-decomposition (Step 5), or finds a Gröbner basis (Step 9). As, we add *new* variables in each lifting-step, it is expected: In practice, the ideal will already be absolutely-irreducible and almost in Gröbner basis; so an implementation may run faster than our worst-case analysis. Basically, *our algorithm is especially fast in finding (p-adic)* $\mathbb{Z}_p$*-roots of an input* $\mathbb{Z}_p$*-ideal* $\hat{\mathtt{I}}$*, if* $\hat{\mathtt{I}}$ *is prime and* $\hat{\mathtt{I}} + \langle p \rangle$ *is absolutely irreducible* (see Lemma 9).

1.3 Proof overview of Theorem 1

In this section, we describe a method to find roots of a system of polynomials over a Galois ring. As we will see in Appendix 3, this has direct consequence on factoring.

Theorem 1 is proved by giving an algorithm that returns FALSE if the given system of n-variate polynomials $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}) \in \mathbb{Z}[z][\mathbf{x}]$ has no root in Galois ring $\mathbb{G} := \mathbb{Z}[z]/\langle p^k, \varphi(z) \rangle$, otherwise

[1] We will use the formal variables $\mathbf{y}_i$'s to denote the i-th base-p digits of $\mathbf{x} = (x_1, \ldots, x_n)$.

outputs a possible root. This is similar to the problem of solving Hilbert's Nullstellensatz over the Galois ring $\mathbb{G}$, which asks for a root of a system of polynomial equations in the closure of a finite field. As we have seen before, there have been several works on this problem in the setting of fields, but non-fields such as Galois rings have remained open.

Main ideas. The idea of the algorithm, as hinted before, works as a reduction to the problem of solving a system of polynomials over a base field $\mathbb{G}/\langle p\rangle = \mathbb{Z}[z]/\langle p, \varphi(z)\rangle \cong \mathbb{F}_p[z]/\langle\varphi(z)\rangle =: \mathbb{F}_q$ (where $q = p^b$ and $b := \deg(\varphi(z))$). The crux of this reduction is storing each base-p digit (in $\mathbb{F}_q$) in a data-structure, which will consist of special polynomial ideals. As we will see later, this is not sufficient, and we need to realize these ideals (initially over $\mathbb{F}_q$) as ideals over the *unramified p-adic integer* ring $\widehat{\mathbb{G}} := \mathbb{Z}_p[z]/\langle\varphi(z)\rangle$. Finally, we find the solution up to an accuracy of k base-p digits to find a root of the system of polynomials over $\mathbb{G}$.

Throughout this algorithm, we will use [HW99] to solve a system of polynomials over finite field $\mathbb{F}_q$, which requires an additional condition that q must be large enough, i.e., $q > d^{(nk)^{\Omega((nk)^2)}}$. If q is smaller, then the brute-force search for roots over $\mathbb{G}$ can be done in time q^{nk}.

Finding the base-p digits of the coordinates. The method of finding roots is performed iteratively on each p-adic coordinate (or *digit*). For an expansion of the form $\sum_{i=0}^{k-1} a_i p^i$, $a_i \in \{0,\ldots,p-1\}$, we refer to a_j as the j-th precision digit in the base-p notation. We find the (virtual) roots at each step, and using these roots, find those corresponding to digits of higher precision in the base-p notation. This method is essentially a reduction from modulo p^k to $\mathbb{F}_q$, and similar techniques have been used before, however restricted only to univariate polynomials [BLQ13, DMS19, DMS21, NRS17]. We generalize this lifting technique to multivariate polynomials. For a root $\mathbf{a} \in \mathbb{F}_q^n$ (also embedded in $\mathbb{G}^n$) of the system of polynomials modulo p, we transform each of the polynomials $f_j(\mathbf{x})$ to $f_j(\mathbf{a}+p\mathbf{x})$ for $j \in [m]$, inspired by the base-p digits of the p-adic notation. Since we have standard methods to find roots in $\mathbb{F}_q$, while the same is difficult in Galois rings, we divide-out the 'excess' powers of p, to bring this system back to $\mathbb{F}_q$. These excess powers of p will be given by $v_j = v(f_j(\mathbf{a}+p\mathbf{x}))$[2], which will be termed as *val-multiplicity* of the root $\mathbf{a}$ (Definition 20). The step thus discussed, given by transforming the polynomial $f_j(\mathbf{x})$ to $p^{-1}f_j(\mathbf{a}+p\mathbf{x})$ will be called the *lifting step*. The point $\mathbf{a}$ will be termed as the *local root* at that lifting step. We could try a more direct lift, $p^{-v_j}f_j(\mathbf{a}+p\mathbf{x})$ (since val-multiplicities can be ≥ 1); however this idea fails, if $v_j \geq 2$, due to some intermediate mod p arithmetic that our algorithm uses.

The modification to the polynomial during lifting will make sure that the $\mathbb{F}_q$ digits during the t-th step of lifting will return the t-th p-adic digit. For example, if $\mathbf{a}$ is an $\mathbb{F}_q$-root of $f_j(\mathbf{x})$, and after lifting, the polynomial becomes $\tilde{f}_j(\mathbf{x}) := p^{-1}f_j(\mathbf{a}+p\mathbf{x})$ which has an $\mathbb{F}_q$-root $\mathbf{b}$, then $(a_1+pb_1,\ldots,a_n+pb_n)$ is a root of $f_j(\mathbf{x})$ mod p^2. Note that some local roots might not have liftings while others can; as illustrated by the following example.

Example 6. *Consider $f(x_1,x_2) := x_1^3 - x_2^3 + 5$ and $p := 5$. $(0,0)$ and $(1,1)$ are its $\mathbb{F}_p$-roots. When we start the root $(0,0)$, the lifting step given by the transformation $(x_1,x_2) \mapsto (5x_1,5x_2)$ and subsequent division by 5, yields the polynomial $25x_1^3 - 25x_2^3 + 1$ which does not have $\mathbb{F}_5$-roots. However, restarting with the root as $(1,1)$ yields the polynomial $25x_1^3 - 25x_2^3 + 15x_1^2 - 15x_2^2 + 3x_1 - 3x_2 + 1$ after lifting, which now has $(3,0)$ as its $\mathbb{F}_5$-root! This anomaly is partially explained by $(0,0)$ being a singular root of f, while $(1,1)$ is a smooth point (*non-*singular).*

Virtual roots. The algorithm for univariate root finding [BLQ13] implicitly enumerates over all possible $\mathbb{F}_q$-roots, to check which one lifts. It is not possible for us to enumerate over all roots

[2] $v(\cdot)$ is called p-adic valuation function where $v(a)$ is the maximum power of p dividing a.

as already for curves, $\Omega(p)$ roots might exist, and there is no standard way of representing them 'compactly'. To tackle this problem, we introduce 'formal' variables $(y_{i,1}, \ldots, y_{i,n})$ for the roots corresponding to the i-th lifting step, instead of fixing them to $(a_1, \ldots, a_n) \in \mathbb{F}_q^n$, and lift to the next step to form a new polynomial $\tilde{f}$ in terms of this $\mathbf{y}_i$. We will denote this tuple $(y_{i,1}, \ldots, y_{i,n})$ as *virtual root*. At each step, we need the property of $\mathbf{y}_i$ that it must be a root modulo p. In order to track these properties together, we create a (p-adic) ideal $\hat{\mathtt{I}}$; which is a novel data-structure introduced by us that stores all possible roots, but in 'higher'-precision p-adics. Thus at every step, we include the polynomials $f_j(\mathbf{y}_i)$ mod p to this ideal, for $j \in [m]$; factorize, and lift again to p-adics. For instance, $\hat{\mathtt{I}}$ vanishing at a point $\mathbf{a} \in (\mathbb{Z}[z]/\langle\varphi(z)\rangle)^n$ implies: the virtual root $\mathbf{y}_i$ can be realized as an 'actual' root $\mathbf{a}$ of $f_j(\mathbf{x})$ which makes the quantity exactly zero. A similar idea of storing roots via ideals, though in a much simpler setting of 0-dimensional ideals, has been employed for univariate polynomials in [CGRW19, DMS19, DS20]. Our method uses any ideal, and gives the new idea of lifting multivariate polynomials modulo ideals.

Lifting and p-adics. In the first step, we have the system of (p-adic) polynomials $f_j(\mathbf{x})$, $j \in [m]$. If this system has a local root $\mathbf{a}$, then we perform lifting to get the polynomials $\tilde{f}_j(\mathbf{x}) := p^{-1} f_j(\mathbf{a} + p\mathbf{x})$ and move on to find the roots of $\tilde{f}_j$. As previously described, we use the virtual root $\mathbf{y}_0 = (y_{0,1}, \ldots, y_{0,n})$ instead of fixing the local root, and then add the polynomials $f_j(\mathbf{y}_0)$ mod p, $j \in [m]$, to the ideal $\mathtt{I}$. The polynomial system after this will be considered modulo $\mathtt{I}$, however with a slight modification due to the following obstruction.

In this process of forming next-precision polynomials and ideals, we use $\widehat{\mathbb{G}}$-arithmetic, instead of $\mathbb{F}_q$ in the base, as it handles division by p in a clean way. Operations over $\mathbb{F}_q$ would not have allowed division by p as new terms from the ideal might reappear in later steps when we divide by p.

The necessity for p-adics was also illustrated in Example 5. It is quite possible that a polynomial $r(\mathbf{y})$ is in the ideal is such that for an $\mathbb{F}_q$-root $\mathbf{a}$ of the system it, $r(\mathbf{a})$ is a non-zero multiple of p over $\mathbb{Z}$. Given an $f(\mathbf{x})$ in the system of polynomial equations, let the polynomial after taking $f(\mathbf{y} + p\mathbf{x})$ modulo $r(\mathbf{y})$ be of the form $\tilde{f}(\mathbf{y} + p\mathbf{x}) + q(\mathbf{y})r(\mathbf{y})$. Following this, we divide by p and recursively continue to $p^{-1}\tilde{f}(\mathbf{y} + p\mathbf{x})$ for finding roots. This can lead to an error if $p^{-1}q(\mathbf{a})r(\mathbf{a})$ is non-zero modulo p, in which case we should have continued to find roots of $p^{-1}\tilde{f}(\mathbf{y} + p\mathbf{x}) + c$, where c is a non-zero constant.

However, if the polynomial $r(\mathbf{y})$ were such that all of its roots were possible to lift to roots in $\widehat{\mathbb{G}}$, this problem would have been avoided as the lift $\hat{\mathbf{a}}$ of $\mathbf{a}$ to $\widehat{\mathbb{G}}$ would have made $r(\mathbf{y})$ vanish over p-adics. It turns out that we can indeed modify the ideals such that these lifts exist. This will be discussed in more details in Section 2.

Growing the p-adic ideal. We develop the idea of formation of p-adic ideals, which will store the roots of each step of lifting. We will use $\hat{\mathtt{I}}$ for the $\widehat{\mathbb{G}}$-ideal, which is a p-adic lifting of $\mathtt{I}$ to $\mathbb{Z}_p$ and perform our operations over $\widehat{\mathbb{G}}$. This lifting of the ideal from $\mathbb{F}_q$ to the ring $\widehat{\mathbb{G}}$ will be explicitly described later, but it can be roughly seen as considering the $\mathbb{F}_q$ elements as $\widehat{\mathbb{G}}$ elements, with the trailing base-p digits being zero (an integral lift of $\mathbb{F}_q$ to $\widehat{\mathbb{G}}$).

Assuming that we have the p-adic lift of the ideal, $\hat{\mathtt{I}}$, we describe a lifting step and the consecutive growing of the ideal. Let us assume that we have lifted the system of polynomials $f_j(\mathbf{x})$, $j \in [m]$, for ℓ-steps, to give the polynomials $\tilde{f}_j(\mathbf{x}) \in \widehat{\mathbb{G}}[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}][\mathbf{x}]$, and the ideal $\hat{\mathtt{I}} \subseteq \widehat{\mathbb{G}}[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}]$.

Next, we consider the local virtual root $\mathbf{y}_\ell =: (y_{\ell,1}, \ldots, y_{\ell,n})$ of the system $\tilde{f}_j(\mathbf{x})$, $j \in [m]$, and perform lifting to the $(\ell+1)$-th precision. First, we increase the precision of the root in the ideal by adding the polynomials $f_j(\mathbf{y}_\ell)$ mod p into $\hat{\mathtt{I}} + \langle p\rangle$ (the projection of $\hat{\mathtt{I}}$ unto $\mathbb{F}_q$), for $j \in [m]$, and redefining $\hat{\mathtt{I}}$— thus storing the information about the ℓ-th step and growing the precision of the

roots contained inside $\hat{\mathtt{I}}$. Subsequently, we obtain the new (lifted) system of polynomials given by $g_j(x) = p^{-1}\tilde{f}_j(\mathbf{y}_\ell + px) \bmod \hat{\mathtt{I}}$, on which we proceed recursively.

Finding a satisfying instance. After iteratively forming a chain of ideals while increasing the precision of the roots, we check if the system has a solution. The root of the polynomial which was, say, present at the beginning of the algorithm will be of the form

$$(a_{0,1} + a_{1,1}p + \cdots + a_{k-1,1}p^{k-1}\,,\,\ldots\,,\,a_{0,n} + a_{1,n}p + \cdots + a_{k-1,n}p^{k-1})\,. \tag{1}$$

Here, j-th digit $a_{i,j}$ is in $\widehat{\mathbb{G}}$, as described earlier. So, this expansion is *not* unique (eg. we can subtract any number t from $a_{1,1}$ and add $p \cdot t$ to $a_{0,1}$), but depends on the remaining digits. Also, it suffices, for our application, to find values $a_{i,j}$ of the virtual root $y_{i,j}$ only up to the precision of $\mathbb{G}$ (at most k-many digits).

2 Hilbert's Nullstellensatz over Galois rings: Proof of Theorem 1

In this section, we complete the algorithm that finds a common root of a system of polynomials over a Galois ring. The algorithm stores the roots in special ideals and modifies them such that they have roots that lift to p-adics; and we describe a method to extract an 'actual' root from each of these ideals returned by the algorithm. Some proofs of this section have been moved to Section B.

2.1 Main algorithm: Finding roots of a polynomial system

In order to complete the algorithm, we establish the missing details from Section 1.3. Our main objective is, given the ideals formed during lifting, we transform them as a set of 'special' ideals where each $\mathbb{F}_q$ point can be lifted to $\widehat{\mathbb{G}}$. As seen in Proposition 3, enabling val-multiplicity of each root would imply them to be lifted modulo any power of p. This also implies separating out the non-singular roots should guarantee p-adic roots, which is now the motive of the rest of the section.

Lifting mod irreducible components. As discussed before, given the polynomials $f_j(\mathbf{x}) \in \hat{G}[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}][\mathbf{x}]$ and the ideal $\hat{\mathtt{I}} \in \widehat{\mathbb{G}}[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}]$ after ℓ steps of lifting, we now increase the precision of the roots in $\hat{\mathtt{I}}$ to $\hat{\mathtt{I}}'$ given by adding $\mathbf{y}_\ell$'s, and lift the system to $\tilde{f}_j(\mathbf{x}) := p^{-1}f(\mathbf{y}_\ell + p\mathbf{x}) \bmod \hat{\mathtt{I}}'$ for $j \in [m]$. However, due to the motivation of having roots of the ideal that can be lifted to p-adics, instead of using $\hat{\mathtt{I}}$, we use an *irreducible component* of $\hat{\mathtt{I}} + \langle p\rangle$ lifted to p-adics, denoted as $\hat{\mathtt{C}}$, as we will see later on. These irreducible components of ideals can be seen as 'factors' of ideals corresponding to the factors of the polynomials generating the ideals. However, followed by this, we perform decomposition of this ideal into *absolutely* irreducible components (Definition 17), on which we perform our arithmetic, to redefine $\tilde{f}_j$. It will also guarantee val-multiplicity roots 1 (since roots are non-singular), and the roots having lifts to $\widehat{\mathbb{G}}$ (Proposition 3). Now, for the lifting step, now we will have the polynomial after lifting, denoted by $\tilde{f}_j(\mathbf{x}) := p^{-1}f_j(\mathbf{y}_0 + p\mathbf{x}) \bmod \hat{\mathtt{C}}$ instead of taking modulo $\hat{\mathtt{I}}$; where the ideal arithmetic is over $\widehat{\mathbb{G}}$.

In case this gives new constraints on previous variables, we *backtrack* the steps.

Branching out by absolutely irreducible components. Our objective is to obtain non-singular roots, via the birational hypersurface, as they lift all the way to p-adics. So, we 'replace' the ideal $\mathtt{I}$ by its absolutely irreducible components. At times, these components might correspond to individual points (single point ideals from 0-dimensional ideals), when again, each root lifts to $\widehat{\mathbb{G}}$ (Lemma 8).

Thus, as discussed in the above paragraph (on finding non-singular roots), we decompose the ideal $\mathtt{I} = \hat{\mathtt{I}} + \langle p\rangle$, the projection of $\hat{\mathtt{I}}$ to $\mathbb{F}_q$, using the decomposition algorithm of [HW99] (see

Theorem 24, Section A), and then lift them again to $\widehat{\mathbb{G}}$ using Lemma 4. The decomposition procedure may give 'many' absolutely irreducible components over $\mathbb{F}_q$, on each of which our SHN_{p^k} algorithm recurses. This procedure— of selecting an irreducible component $\mathtt{C}$ each time, growing that ideal to next precision, and again loop over its irreducible components —can be seen as a *tree* $\mathcal{T}$.

This tree has several nodes which correspond to absolutely irreducible ideals. The depth of the tree represents the precision of the roots formed until that point. This is also the number of times the polynomials have been lifted, and thus the ideal has grown. The tree gives branches which correspond to nodes containing irreducible components of the ideal formed from growing the parent ideal to the next precision digit in base-p notation. We deal with these ideals in the leaves, $\mathcal{L}$, of the tree, $\mathcal{T}$, returned by Algorithm 1. We extend the results to prove Theorem 5 which shows that all the roots are captured by the ideals in $\mathcal{L}$.

Backtracking. Whenever we arrive at a root given by some ideal present in a node of the tree, it must give a root with a precision equal to the depth of the tree, say ℓ. The path of the tree from the root to the given node should also give each of the previous digits of the root, i.e., all the previous base-p digits of precision $(\ell-1)$, must be present at the current node, while only some particular values give a valid root of the ℓ-th precision digit. This implies that the information about the $(\ell-1)$ precision digit in the ideal, before the decomposition, $\mathtt{I} \cap \mathbb{F}_q[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}]$ must be equal to that after decomposition, $\mathtt{C} \cap \mathbb{F}_q[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}]$. If this condition is not satisfied (i.e., a new root corresponding to the previous digits arrives out of the blue), then we backtrack to the earlier steps of the tree, discarding $\mathtt{I}$ and updating it by the new ideal $\mathtt{C}$.

Finding non-singular roots. After setting the virtual roots in order to achieve the required prime power k, say that we have the ideal $\hat{\mathtt{I}}$, birationally mapped to a hypersurface $\hat{\mathtt{H}}$, given by a polynomial $\hat{h}$ over $\widehat{\mathbb{G}}$, of $\dim = r$, for $0 \leq r \leq \dim(\mathtt{I})$. While, $\mathtt{I}$ $(= \hat{\mathtt{I}} + \langle p \rangle)$ is birationally mapped to a hypersurface $\mathtt{H}$, given by a polynomial h over $\mathbb{F}_q$. We find a random root of $\mathtt{H}$ [HW99, Thm.2.6] and map it to the 0-th precision digit of roots of $\hat{\mathtt{H}}$; this crucial property is proved in Lemma 4. After which we can lift to find a $\widehat{\mathbb{G}}$-root using an easy variant of Hensel's lifting (Proposition 3). This procedure gives us a $\widehat{\mathbb{G}}$-root from the $\mathbb{F}_q$-root.

Now, the density of non-singular roots on $\mathtt{H}$ will be much greater than singular roots (Lemma 7) if it is *absolutely irreducible*. Picking a non-singular $\mathbb{F}_q$-root, at random, we can lift it to a $\mathbb{G}$-root. If $\mathtt{H}$ is *relatively irreducible* (i.e., reduces in some field extension), we add the first-order derivative of h to the ideal (Lemma 22), as will be explained in Section 2.2. Thus, we lift a root whenever the hypersurface is absolutely irreducible and satisfiable.

Pseudocode. Using these ideas, and some technicalities on decomposition into absolutely irreducible components described in Section 2, we sketch our algorithm. Using the system of polynomials, it returns all the leaves of the tree described in the previous paragraph.

Input: The input consists of a system of n-variate polynomials $\{f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}) \mid f_j(\mathbf{x}) \in \widehat{\mathbb{G}}[\mathbf{y}_0, \ldots \mathbf{y}_{\ell-1}][\mathbf{x}]\}$ with the required exponent k, and an ideal $\hat{\mathtt{I}} = \hat{\mathtt{I}}_{\ell-1} \subseteq \widehat{\mathbb{G}}[\mathbf{y}_0, \ldots \mathbf{y}_{\ell-1}]$; where p is prime and $\varphi(z)$ is an $\mathbb{F}_p$-irreducible polynomial. We also maintain the ideal tree $\mathcal{T}$ and keep updating it along the algorithm.

Output: The algorithm outputs a list $\mathcal{L}$ of (absolutely irreducible) ideals, collectively containing the lift of the common roots of the system $f_j(\mathbf{x}) \equiv 0 \bmod \langle p^k \rangle + \hat{\mathtt{I}}$, for $j \in [m]$.

Initialization: We initialize the ideal as $\hat{\mathtt{I}} := \langle 0 \rangle$, $\ell := 0$, and the required exponent as k. A system of polynomials $\mathcal{F} := \{F_1(\mathbf{x}), \ldots, F_m(\mathbf{x})\}$ where $F_j(\mathbf{x}) \in \widehat{\mathbb{G}}[\mathbf{x}]$. We pass $\mathcal{F}$ to the algorithm so it starts with $\mathrm{SHN}_{p^k}(F_1, \ldots, F_m, k, \langle 0 \rangle)$.

Algorithm 1 Algorithm to find roots of a system of polynomial equations over a Galois ring.

procedure $\mathrm{SHN}_{p^k}(f_1, \ldots, f_m, k, \hat{\mathtt{I}}, \mathcal{T})$

  if Zeroset $\mathbf{V}_{\mathbb{F}_q}(\hat{\mathtt{I}} + \langle p \rangle) = \emptyset$ **then return** $\{\}$.

  if $k \leq 0$ **then return** $\{\hat{I}\}$.

  $\mathtt{I} \leftarrow \langle f_1(\mathbf{y}_\ell), \ldots, f_m(\mathbf{y}_\ell) \rangle + \hat{\mathtt{I}} + \langle p \rangle$, for (new) virtual root $\mathbf{y}_\ell := (y_{\ell,1}, \ldots, y_{\ell,n})$.

  $\mathcal{S} \leftarrow$ Abs_Decomp($\mathtt{I}$); absolutely irreducible ideals as computed by Algorithm 2.

  $\mathcal{L} \leftarrow \{\}$

  for each $\mathtt{C} \in \mathcal{S}$ **do**

    if $\mathtt{C} \cap \mathbb{F}_q[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}] = \mathtt{I} \cap \mathbb{F}_q[\mathbf{y}_0, \ldots, \mathbf{y}_{\ell-1}]$ **then**

      Find the special lift $\hat{\mathtt{C}}$ of $\mathtt{C}$ to $\widehat{\mathbb{G}}$ by computing Gröbner basis and lifting, using Lemma 4. /*$\hat{\mathtt{C}}$ is *prime*; reduced Gröbner basis w.r.t. $\mathbf{y}_0 < \ldots < \mathbf{y}_{k-1}$.*/

      Add $\mathtt{C}$ as a child of the current node to $\mathcal{T}$.

      For $j \in [m]$, compute $\tilde{f}_j(\mathbf{x}) := p^{-1} f_j(\mathbf{y}_\ell + p\mathbf{x}) \bmod \hat{\mathtt{C}}$, over $\widehat{\mathbb{G}}[\mathbf{y}_0, \ldots \mathbf{y}_\ell][\mathbf{x}]$.

      $\mathcal{L} \leftarrow \mathcal{L} \cup \mathrm{SHN}_{p^k}(\tilde{f}_1, \ldots, \tilde{f}_m, k-1, \hat{\mathtt{C}}, \mathcal{T})$. /*Maintain the recursion-tree $\mathcal{T}$.*/

    else /*Backtrack & repeat steps*/

      Find min $s \leq \ell - 1$ s.t. $\mathtt{C} \leftarrow \mathtt{C} \cap \mathbb{F}_q[\mathbf{y}_0, \ldots, \mathbf{y}_s] \supsetneq \mathtt{I} \cap \mathbb{F}_q[\mathbf{y}_0, \ldots, \mathbf{y}_s]$.

      Find special lift $\hat{\mathtt{C}}$ of $\mathtt{C}$ over $\widehat{\mathbb{G}}$ using Lemma 4.

      For all $j \in [m]$, compute $\tilde{f}_j(\mathbf{x}) := p^{-s-1} F_j(\mathbf{y_0} + \cdots + p^s \mathbf{y}_s + p^{s+1}\mathbf{x}) \bmod \hat{\mathtt{C}}$.

      $\mathcal{L} \leftarrow \mathcal{L} \cup \mathrm{SHN}_{p^k}(\tilde{f}_1, \ldots, \tilde{f}_m, k + \ell - 1 - s, \hat{\mathtt{C}}, \mathcal{T})$. /*Maintain $\mathcal{T}$ as before.*/

  return $\mathcal{L}$. /*Also, return the recursion-tree $\mathcal{T}$ whose leaves are ideals in $\mathcal{L}$.*/

Simple invariant. A node in the recursion-tree $\mathcal{T}$ either moves from $\hat{\mathtt{I}}_{\ell-1}$ to $\hat{\mathtt{I}}_\ell$, or *backtracks* to redefine $\hat{\mathtt{I}}_s$, $s < \ell$. In the former case, k reduces, while in the latter case $\dim(\mathbf{V}(\hat{\mathtt{I}}_s))$ reduces. Thus, in a path, $\hat{\mathtt{I}}_s$ can be redefined at most n times; thus bounding the length of any path in the tree by $\leq k + kn$.

2.2 Decomposition into absolutely irreducible components

As discussed in Section 1.3, we need the ideal in Algorithm 1 to have a (non-singular) point that lifts to $\widehat{\mathbb{G}}$ at every lifting step when we modify $\tilde{f}$. In order to ensure this, we modify the machinery developed in [HW99] to find the irreducible components of $\mathtt{I}$ whose birationally equivalent hypersurface is absolutely irreducible over $\mathbb{F}_q$. The crux of the decomposition algorithm of [HW99] is Lemma 22, which will be used to reduce the dimension of the components and iteratively continue the decomposition algorithm. (See Algorithm 2.)

Decomposing via a birationally equivalent hypersurface. First, given the ideal $\mathtt{C}$ irreducible over $\mathbb{F}_q$ which consists of several generators, we can construct a birationally equivalent hypersurface $\mathtt{H}$ obtained from random linear shift of variables of $\mathtt{C}$ which is given by a single polynomial h. We will use [HW99] for the construction of this hypersurface. Rational points on $\mathtt{H}$ correspond to roots of $\mathtt{C}$ and vice versa. If h is absolutely irreducible, we can lift an $\mathbb{F}_q$-root to $\widehat{\mathbb{G}}$ as will be proved in Proposition 3-(2). We also show a strong connection between the absolutely irreducible decomposition over $\mathbb{F}_q$ and that over $\widehat{\mathbb{G}}$ (Lemma 4) using a commutative diagram (Figure 1, Appendix B).

We follow the ideas of [HW99, Sec.3.3], and give a brief overview for the sake of clarity. Given an $\mathbb{F}_q$-ideal $\mathtt{C}$, we find the (finitely many) irreducible components of $\mathtt{C}$. As long as the irreducible components map to a hypersurface $\mathtt{H} = \mathbf{V}(h)$ which is *not* absolutely irreducible, we keep adding the pullback of a derivative of h, say h^*, into the ideal; since the roots will satisfy this equation as

well (Lemma 22). This procedure reduces the dimension of the ideal $\mathtt{C}$; thus, it continues for only few steps, and reaches absolutely irreducible components.

Loss of points. The map between a component and its birational equivalent hypersurface is given as rational functions $\psi_2 : \mathbf{V}(\mathtt{C}) \to \mathtt{H}$ and $\psi_1 : \mathtt{H} \to \mathbf{V}(\mathtt{C})$. (Contravariantly, a map ψ_1 can also be seen as a morphism on their function fields: $\mathbb{F}_q(\mathbf{V}(\mathtt{C})) \to \mathbb{F}_q(\mathtt{H})$.) Owing to this property of rationality, some points in $\mathtt{C}$ get lost when we perform the composition $\psi_1 \circ \psi_2$. These are precisely the zeroes of the functions which are in the denominator of ψ_1 (since ψ_2 is linear, like the construction idea of primitive element theorem, and no roots will be lost). Let us consider the polynomial e formed by multiplying the denominators of ψ_1. In order to include these points of $\mathtt{C}$ in our search, we consider the polynomial e^* which is the pullback of e unto $\mathtt{C}$, and include it to the ideal $\mathtt{C}$. As we will see, again the dimension of the ideal over $\mathbb{F}_q$ reduces, implying that this procedure occurs only few times. Thus, we avoid losing any $\mathbb{F}_q$-point, and simultaneously move to absolutely irreducible hypersurfaces!

Special lift of ideals and roots. We are forming the ideals by adding $f_j(\mathbf{y}_\ell)$ mod p into $\mathtt{I}$ and then 'lifting' the ideal to the p-adic one over $\widehat{\mathbb{G}}$. A question arises: Which $\mathbb{F}_q$-roots lift smoothly to $\mathbb{Z}_p$ and which don't? The latter ones are handled separately (as discussed above).

Lemma 4 (Connection of points via hypersurfaces)**.** *Given an $\mathbb{F}_q$-irreducible ideal $\mathtt{C}$ (resp. its birational equivalent hypersurface $\mathtt{H}$), we can lift it to a* prime *$\widehat{\mathbb{G}}$-ideal $\hat{\mathtt{C}}$ (resp. its birational equivalent hypersurface $\hat{\mathtt{H}}$), such that their morphism diagram commutes (Figure 1).*

In particular, for a non-singular $\mathbb{F}_q$-root of $\mathtt{H}$ (thus a root of $\mathtt{C}$), we can find a $\widehat{\mathbb{G}}$-root of $\hat{\mathtt{H}}$; which gives a root of $\hat{\mathtt{C}}$. This sets up the 'connection' between roots of $\mathtt{C}$ and $\hat{\mathtt{C}}$.

The proof of Lemma 4 is given in Appendix B. The basic idea in the proof of Lemma 4 is: Given $\mathbb{F}_q$-irreducible ideal $\mathtt{C}$, compute the reduced Gröbner basis of $\mathtt{C}$, using the block order $\mathbf{y}_0 < \ldots < \mathbf{y}_{k-1}$; and simply see it as a p-adic ideal $\hat{\mathtt{C}}$. This is a $\widehat{\mathbb{G}}$-irreducible ideal, which is the required special lift of $\mathtt{C}$. (Its localized version $B^{-1}\hat{\mathtt{C}}$ has a *triangular* Gröbner basis; where B is a transcendence basis of variables.) The proof closely follows Figure 1 (Appendix B); and proves its commutativity.

Now, the proof of Lemma 4 fails on a small set of points given by the roots of e, the product of denominators of ψ_1, as described before. So, we separately include these points, by including the pullback of e, which is e^*; and continue with our decomposition algorithm. Furthermore, the root-lifting technique of Lemma 4 uses non-singular points to lift to p-adics, which does not work for *singular* points (roots of h which are also roots of all first-order derivatives h'). Thus, again we need to add the pullback of h', say h^*, into the ideal, and continue with our decomposition Algorithm 2. Finally, we can find (non-singular) $\mathbb{F}_q$-roots of a (absolutely irreducible) system using [HW99].

In the end, we have absolutely irreducible ideals, which lift over $\widehat{\mathbb{G}}$. Note that sometimes the absolutely irreducible ideals might be single points as well, of the form $\langle \mathbf{y} - \mathbf{a} \rangle$. When the ideals are absolutely irreducible, it is easy to search for a $\widehat{\mathbb{G}}$-root (see Section 2.3).

Based on these ideas, we give Algorithm 2, Abs_Decomp($\mathtt{I}$), that decomposes the $\mathbb{F}_q$-ideal $\mathtt{I}$ to absolutely irreducible ideals $\mathtt{C} \in \mathcal{S}_{abs}$ in such a way that zeroset $\mathbf{V}_{\mathbb{F}_q}(\mathtt{I})$ remains unchanged, i.e, $\mathbf{V}_{\mathbb{F}_q}(\mathtt{I}) = \bigcup_{\mathtt{C} \in \mathcal{S}_{abs}} \mathbf{V}_{\mathbb{F}_q}(\mathtt{C})$.

Input: The algorithm takes as input, a radical ideal $\mathtt{I} \subseteq \mathbb{F}_q[y_1, \ldots, y_n]$.

Output: The algorithm outputs a set $\mathcal{S}_{abs}$ consisting of absolutely irreducible ideals $\mathtt{C}$, s.t. $\mathbf{V}(\mathtt{I}) = \bigcup_{\mathtt{C} \in \mathcal{S}_{abs}} \mathbf{V}(\mathtt{C})$.

Algorithm 2 Decomposing $\mathtt{I}$ into absolutely irreducible components over $\mathbb{F}_q$.

procedure ABS_DECOMP($\mathtt{I}$)
  Define $\mathcal{S}_{abs} := \{\}$ and $\mathcal{S}_{irr} := \{\}$.
  Decompose $\mathtt{I}$ into irreducible components over $\mathbb{F}_q$ using Theorem 24 and store them in $\mathcal{S}_{irr}$.
  while $\mathcal{S}_{irr} \neq \emptyset$ **do**
    $\mathtt{C} \leftarrow \mathtt{Pop}(\mathcal{S}_{irr})$.
    if $\dim(\mathbf{V}_{\mathbb{F}_q}(\mathtt{C})) = 0$ **then**
      Compute $\mathbf{V}_{\mathbb{F}_q}(\mathtt{C})$ using [HW99] and for each $\mathbf{a} \in \mathbf{V}_{\mathbb{F}_q}(\mathtt{C})$, update $\mathcal{S}_{abs} \leftarrow \mathcal{S}_{abs} \cup \{\langle \mathbf{y} - \mathbf{a} \rangle\}$.
    else
      if $\mathtt{C}$ is absolutely irreducible **then**
        $\mathcal{S}_{abs} \leftarrow \mathcal{S}_{abs} \cup \{\mathtt{C}\}$
      Let $\dim(\mathbf{V}_{\mathbb{F}_q}(\mathtt{C})) =: r$. Using [HW99] (Theorem 24) compute a birationally equivalent hypersurface $\mathtt{H} := \mathbf{V}_{\mathbb{F}_q}(h(l_1, \ldots, l_r, Y))$ and the rational maps $\psi_1 : \mathtt{H} \to \mathbf{V}(\mathtt{C})$ and $\psi_2 : \mathbf{V}(\mathtt{C}) \to \mathtt{H}$. ($\mathbf{l}, Y$ are linear forms in $\mathbf{y}$; also see Figure 1.)
      Compute $\mathtt{C}_1 := \mathsf{Rad}(\mathtt{C} + \langle h^* \rangle)$, where h^* is pullback of a first-order partial-derivative $h' \neq 0$.
      Compute $\mathtt{C}_2 := \mathsf{Rad}(\mathtt{C} + \langle e^* \rangle)$, where e^* is the pullback of e, which is a product of the denominators that appear– in rational functions $\psi_1 =: (\psi_{1,1}, \ldots, \psi_{1,n})$, or in the localization done in Lemma 4.
      Decompose the ideals $\mathtt{C}_1, \mathtt{C}_2$ into irreducible components over $\mathbb{F}_q$ using Theorem 24 ([HW99]), and push these components into $\mathcal{S}_{irr}$.
  return $\mathcal{S}_{abs}$

2.3 Recovering a $\mathbb{G}$-root of an ideal in $\mathcal{L}$ and $\mathcal{T}$ (of Algorithm 1)

An ideal $\hat{\mathtt{I}} \in \mathcal{L}$ has the property that modulo p, i.e., $\mathtt{I} := \hat{\mathtt{I}} + \langle p \rangle$, it is absolutely irreducible.

If the ideal consists of a single point, from Lemma 8, each $\mathbb{F}_q$-root of $\mathtt{I}$ lifts to $\mathbb{G}$ where the trivial lifting is the required root.

If the ideal $\mathtt{I}$ has dim > 0 (i.e., the points are 'dense' by Theorem 23), then its birationally equivalent hypersurface $\mathtt{H} = \mathbf{V}(h)$ is utilized. Let the map $\psi_2 : \mathbf{V}_{\mathbb{F}_q}(\mathtt{I}) \to \mathtt{H}$ be defined as $(\ell_1, \ldots, \ell_n) \to (\ell_1, \ldots, \ell_r, \ell_0)$, where $r \geq 1$ is the dimension of $\mathbf{V}_{\mathbb{F}_q}(\mathtt{I})$ and ℓ_i's are random linear forms (as in Figure 1). Next, consider any lift $\hat{\mathtt{H}} = \mathbf{V}(\hat{h})$ of $\mathtt{H}$ and compute the unique birational equivalence $\hat{\psi}_2 : \mathbf{V}(\hat{\mathtt{I}}) \to \hat{\mathtt{H}}$ and its inverse $\hat{\psi}_1$.

In order to find a root of $\hat{h}$, we pick a random non-singular $\mathbb{F}_q$-root of h, and lift it to a root of $\hat{h}$ (by Lemma 7). Finally, use $\hat{\psi}_1 : \hat{\mathtt{H}} \to \mathbf{V}(\hat{\mathtt{I}})$ to get the $\mathbb{G}$-root of $\hat{\mathtt{I}}$ (again by Lemma 4), which becomes the required output.

Let $\hat{\mathtt{I}}_0, \ldots, \hat{\mathtt{I}}_{k-2}, \hat{\mathtt{I}}_{k-1} = \hat{\mathtt{I}}$ be the eventual p-adic ideal definitions, in a path of the recursion tree $\mathcal{T}$. An important issue is: We need a $\mathbb{G}$-root common to these ideals. Lemma 9 shows that this condition is satisfied by randomly picking a root of $\hat{\mathtt{I}}$. The primality of these ideals together with a 'common' triangular Gröbner basis is key in this proof.

Using the ideas developed above, we formally design Algorithm 3 to recover a root from the set of ideals $\mathcal{L}$ (and recursion-tree $\mathcal{T}$) returned by Algorithm 1. We keep applying Algorithm 3 on each ideal of $\mathcal{L}$ until we find one whose variety is not null.

Input: Let $\mathfrak{I} := \{\hat{\mathtt{I}}_0, \ldots, \hat{\mathtt{I}}_{k-1} =: \hat{\mathtt{I}}\}$ be the eventual ideal definitions leading to the leaf $\hat{\mathtt{I}} \in \mathcal{L}$. Ideal $\hat{\mathtt{I}}_\ell \subseteq \widehat{\mathbb{G}}[\mathbf{y}_0, \ldots, \mathbf{y}_\ell]$ is *prime*, for $0 \leq \ell \leq k-1$, and the required prime-power precision is p^k.

Further, $\mathtt{I}_\ell := \hat{\mathtt{I}}_\ell + \langle p \rangle$ is absolutely irreducible.

Output: A 'generic' common $\mathbb{G}$-root $(\mathbf{a}_0, \ldots, \mathbf{a}_{k-1})$ of $\mathfrak{I}$, if it exists; ϕ otherwise.

Algorithm 3 Recovering p-adic or $\mathbb{G}$-root common to the ideals $\mathfrak{I}$.

procedure ROOTS($\mathfrak{I}, p^k$)
  if $\mathbf{V}_{\mathbb{F}_q}(\hat{\mathtt{I}}_{k-1} + \langle p \rangle) = \phi$ **then**
    return ϕ
  else if $\hat{I}_{k-1}$ contains a single point **then**
    return the single point $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_\ell)$ (Lemma 8).
  else
    Compute the birationally equivalent hypersurface $\mathtt{H}$ of $\hat{\mathtt{I}}_{k-1} + \langle p \rangle$, over $\mathbb{F}_q$, and the maps $\psi_1 : \mathtt{H} \to \mathbf{V}(\hat{\mathtt{I}}_{k-1} + \langle p \rangle)$ and $\psi_2 : \mathbf{V}(\hat{\mathtt{I}}_{k-1} + \langle p \rangle) \to \mathtt{H}$ using Thm. 24. (Also, see Fig. 1).
    Similar to Figure 1, compute the hypersurface $\hat{\mathtt{H}}$ birational to $\hat{\mathtt{I}}_{k-1}$ over $\widehat{\mathbb{G}}$, and the mappings $\hat{\psi}_1 : \hat{\mathtt{H}} \to \mathbf{V}(\hat{\mathtt{I}}_{k-1})$ and $\hat{\psi}_2 : \mathbf{V}(\hat{\mathtt{I}}_{k-1}) \to \hat{\mathtt{H}}$.
    Find a random $\mathbb{F}_q$-root $\mathbf{a}$ on $\mathtt{H}$ using [HW99].
    Map $\mathbf{a}$ to the 0-th precision digit of the corresponding root of $\hat{\mathtt{H}}$, using Lemma 4, to get the approximation $\mathbf{a}'$.
    Lift $\mathbf{a}'$ to $\hat{\mathbf{a}}$, using Hensel's lifting, which is a root of $\hat{\mathtt{H}}$ modulo p^k (Lemma 7).
    return the pullback of $\hat{\mathbf{a}} =: (\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{k-1})$ given by $\hat{\psi}_1(\hat{\mathbf{a}})$.

2.4 Correctness: $\mathcal{T}$ has all the $\mathbb{G}$-roots of the input system $\mathcal{F}$

Theorem 5 (Correctness)**.** *If an ideal in $\mathcal{L}$ returned by Algorithm 1 has $\widehat{\mathbb{G}}$-root $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{k-1})$ (given by Algorithm 3), then the system $\mathcal{F} := \{f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\}$ has $\mathbb{G}$-root $(\hat{\mathbf{a}}_0 + \ldots + p^{k-1}\hat{\mathbf{a}}_{k-1} \bmod p^k)$. Conversely, if $\mathcal{F}$ has $\mathbb{G}$-root $\mathbf{a}_0 + \ldots + p^{k-1}\mathbf{a}_{k-1}$, then an ideal in $\mathcal{L}$ has some $\widehat{\mathbb{G}}$-root $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{k-1})$, such that $\mathbf{a}_0 + \ldots + p^{k-1}\mathbf{a}_{k-1} \equiv \hat{\mathbf{a}}_0 + \ldots + p^{k-1}\hat{\mathbf{a}}_{k-1} \bmod p^k$.*

Proof. The method of finding $\mathbb{G}$-roots of a system of polynomials goes through three algorithms– Algorithm 1, 2 and 3. The primary Algorithm 1, as discussed, works recursively over a tree; with the branches corresponding to absolutely irreducible components of the $\mathbb{F}_q$-ideals (Step 5 of Algorithm 1 as seen in Section 1.3).

The invariant reduces. We first bound the depth of this tree by showing an easy property: val-multiplicity is at least 1 in every step (Lemma 21 in Section A). Thus, at each step of lifting, either a new block $\mathbf{y}_\ell$ is added, or some $\dim(\mathtt{I}_\ell)$ falls. So, the depth of the tree is $\leq k + kn$; and the number of the variables (that store virtual roots) is $\leq nk$. Furthermore, as the tree $\mathcal{T}$ grows the number of new branches produced at a node corresponding to $\mathtt{I}_\ell$ depends on the degree and dimension of $\mathtt{I}_\ell$. By computing the bound on the dimension and degree of the generators of $\mathtt{I}_\ell$ recursively at each level of the tree $\mathcal{T}$, we bound the size of tree $\mathcal{T}$ in the following lemma (proved in Section B).

Lemma 6 (Size of tree)**.** *The total number of leaves $\mathcal{L}$ of the recursion-tree $\mathcal{T}$, described in Section 1.3, is at most $d^{(nk)^{O((nk)^2)}}$.*

Roots that lift. We use [HW99] (Theorem 24, Section A) as a subroutine in Steps 3, 7, 14 of Algorithm 2 to obtain the irreducible components of the ideal over $\mathbb{F}_q$. After this, the ideals will

be lifted in the main algorithm (Steps 9, 14 of Algorithm 1) using Lemma 4. However, its proof needs to create a map from the $\mathbb{F}_q$-ideal (resp. its lift over $\widehat{\mathbb{G}}$) to a hypersurface. In this process, the map becomes undefined on 'few' roots of $\mathtt{H}$ (and hence misses some roots of $\mathtt{C}$ in the codomain). This happens because the map $\psi_1 : \mathtt{H} \to \mathbf{V}(\mathtt{C})$ is a *rational* function where the denominators might be zero for some points of $\mathtt{H}$, and (as already discussed in Section 2.2) we consider the polynomial e^* which captures these missed 'images' in $\mathbf{V}(\mathtt{C})$. We continue our procedure on $\mathtt{C} + \langle e^* \rangle$ (Step 13 of Algorithm 2). Lemma 6 shows that this reduces the dimension of the birationally equivalent hypersurface each time.

Apart from these points, some *singular* points of $\mathtt{H}$ might have lifts to $\widehat{\mathbb{G}}$, but we are unable to apply Hensel lifting directly. To cover these roots (and their missed 'images' in $\mathbf{V}(\hat{\mathtt{C}})$), we consider another ideal where we include the pullback, say h^*, of a suitable first-order derivative of the polynomial h (that defines $\mathtt{H}$).

Likewise, when the ideal is *relatively* irreducible, the roots will be shared with h', the derivative of the polynomial representing $\mathtt{H}$, and we add its pullback h^* (Lemma 22) in Step 12 of Algorithm 2. The dimension of the ideal thus formed reduces by exactly one [HW99], and we continue decomposing the ideal, until it returns an absolutely irreducible ideal. Thus, the while-loop (Steps 4-14 of Algorithm 2) runs for at most nk steps, which is the maximum possible dimension of the ideal.

Roots captured by $\mathcal{L}$. We want to show that all the roots of $f_1(\mathbf{x}) \equiv \cdots \equiv f_m(\mathbf{x}) \equiv 0 \bmod p^k$ are present in $\mathcal{L}$, and vice versa.

First, we show that roots of $\mathcal{L}$ always give rise to a root of the system $\mathcal{F}$ of polynomial equations $f_1(\mathbf{x}) \equiv \cdots \equiv f_m(\mathbf{x}) \equiv 0 \bmod \langle p^k, \varphi(z) \rangle$. We state the following two lemmas, which help in proving that the roots of the system can be constructed from the roots of the projection of the ideals unto $\mathbb{F}_q$.

Lemma 7 (dim >0 lift)**.** *Given an absolutely irreducible hypersurface $\mathtt{H}$ (resp. its lift $\hat{\mathtt{H}}$) over $\mathbb{F}_q$ of of positive dimension, its random $\mathbb{F}_q$-root is non-singular with high probability. Thus, we can lift a random root of $\mathtt{H}$ to $\widehat{\mathbb{G}}$-root of $\hat{\mathtt{H}}$.*

Lemma 8 (single-point lift)**.** *Given an $\mathbb{F}_q$-ideal $\mathtt{I}$ (resp. its lift $\hat{\mathtt{I}}$) that is radical and is a single point, we can uniquely lift it to $\widehat{\mathbb{G}}$-root of $\hat{\mathtt{I}}$.*

The proofs of Lemmas 7 and 8 can be found in Appendix B.

As we see, the lifting slightly changes when the ideal consists of a single point. Now, these single point ideals correspond to 0-dimensional components over $\mathbb{F}_q$, which are $d^{O(nk)}$-many. We can consider the trivial lifts of these single points and check if these finitely many points satisfy the system $\mathcal{F}$.

Using these Lemmas 7 and 8, we prove the correctness of Algorithm 3, which proves one side of the claim: $\mathcal{L}$ exactly captures the roots of the system $\mathcal{F}$.

Lemma 9 (Correctness of Algorithm 3)**.** *Given $\widehat{\mathbb{G}}$-ideal $\hat{\mathtt{I}}_{k-1}$ in a leaf of the tree $\mathcal{T}$, Algorithm 3 finds a generic common $\widehat{\mathbb{G}}$-root (if one exists) of the preceding ideals $\{\hat{\mathtt{I}}_\ell \mid \ell\}$.*

The proof of Lemma 9 is given in Appendix B. Using Lemma 9, we are now in a position to show that we can recover some root of the system $\mathcal{F}$ from $\mathcal{L}$.

Proposition 1 (Root in $\mathcal{L} \longrightarrow$ Root of $\mathcal{F}$)**.** *Given a root of a leaf in $\mathcal{L}$ (using $\mathcal{T}$ and Algorithm 3), we can find a common $\mathbb{G}$-root of the system $\mathcal{F}$ of polynomials f_j, for $j \in [m]$.*

Conversely, we can show that every $\mathbb{G}$-root of the system $\mathcal{F}$ has its p-adic lift present in some ideal of $\mathcal{L}$.

Proposition 2 (Root of $\mathcal{F} \longrightarrow$ Root in $\mathcal{L}$)**.** *If the system of polynomials, as described before, has a root in $\mathbb{G}$, then Algorithm 3 outputs a root for some leaf ideal $\hat{\mathcal{I}}_{k-1}$ in $\mathcal{L}$.*

Both the above propositions are proved in Appendix B. Therefore, we have shown that the roots of nodes in $\mathcal{L}$ *exactly* correspond to those of $\mathcal{F}$. Further, these can be realized by Algorithm 3. □

Proof of Theorem 1. We show in the proof of Theorem 5 in Appendix B that Algorithm 1 (using Algorithms 2–3), correctly returns a root (via an absolutely irreducible ideal), if and only if one exists.

Tree $\mathcal{T}$ built by Algorithm 1 has size $D := d^{(nk)^{O((nk)^2)}}$ and each of the ideal in $\mathcal{T}$ has at most nk variables with degree at most D (Lemma 6). At each step, we perform arithmetic with the reduced Gröbner basis of the ideal, which has polynomials of degree $\leq D$ and $\leq nk$ variables, and requires $\mathsf{poly}(D)$-time $\widehat{\mathbb{G}}$ arithmetic [Dub90]. After these arithmetic operations are performed, we check for an $\mathbb{F}_q$-root of the ideals using [HW99], which takes randomized $\mathsf{poly}(m, D^{(nk)^{O(nk)}}, \log q)$ time. Thus, the net time complexity is randomized $\mathsf{poly}(m, d^{c_{nk}}, \log p^b)$, where $c_{nk} \leq (nk)^{O((nk)^2)}$ and $q = p^b$ with $b := \deg(\varphi)$.

However, the algorithm uses [HW99] as a blackbox, which requires the additional condition that $q = p^b > d^{c_{nk}}$. If this condition is not satisfied, i.e., q is *small*, then we can deterministically find a root using exhaustive search of $q^{nk} \leq d^{c_{nk} \cdot nk}$ many iterations. This case has the time complexity as deterministic $\mathsf{poly}(m, d^{c'_{nk}}, \log p^b)$, where $c'_{nk} \leq c_{nk} \cdot nk \leq (nk)^{O((nk)^2)}$. This proves Theorem 1 in all cases. □

Using this, we show the reduction of factoring to SHN_{p^k} in Section 3; finishing Thm.2.

3 An application: Finding small factors of f mod p^k

In this section we will prove Theorem 2 and Corollary 3 i.e, we show how to efficiently find a 'low' ramification-degree factor of $f(x)$ mod p^k in randomized polynomial time. We achieve this via first reducing the problem, in Sections 3.1 and 3.2, to finding a common zero of a system of multivariate polynomial equations over a Galois ring of characteristic p^k.

Assume the input $f \in \mathbb{Z}[x]$ to be monic mod p^k (leading coefficient 1) as we can always remove the factors, which are units in the ring $(\mathbb{Z}/\langle p^k \rangle)[x]$, by division. Also assume $f \equiv \varphi(x)^e + p \cdot h(x)$ mod p^k (i.e, $f \equiv \varphi^e$ mod p), where $\varphi \in \mathbb{Z}[x]$ is an irreducible polynomial over $\mathbb{F}_p$. Otherwise, using co-prime factorization mod p, we can efficiently find a non-trivial factor of f mod p^k using Hensel's Lemma 18. Let $b := \deg(\varphi)$, with $\deg(f) = b \cdot e$ and $\deg(h) < \deg(f)$.

In this case, finding a ramification-degree δ factor is reduced to finding a root of an $E(y) \in \mathbb{Z}[x, y]$ modulo a bi-generated ideal $\langle p^k, \varphi(x)^\ell \rangle$ (due to [DMS21], Theorem 19) where $\deg_y(E) < k$ and $\ell = \delta \cdot k$. In Section 3.1, we will focus on this root-finding job and reduce this to root finding modulo a simpler ideal $\langle p^k, \varphi(z), (x - z)^\ell \rangle$. Then in Section 3.2, we further reduce this problem to solving a system of multivariate polynomial equations modulo $\langle p^k, \varphi(z) \rangle$ (namely, over the Galois ring).

3.1 Factoring over the Galois ring

We have $f = \varphi^e + ph$ and prime power p^k. Consider the Galois ring $\mathbb{G} := \mathbb{Z}[z]/\langle p^k, \varphi(z) \rangle$ where $z \in \mathbb{G}$ be a root of the polynomial $\varphi(x)$. Denote the roots of $\varphi(x)$ in $\mathbb{G}$ by z_i with $z_0 := z$ for $i \in \{0, \ldots, b-1\}$ (recall $b = \deg(\varphi)$). Then, we know that $z_i \equiv z^{p^i}$ mod p for all $i \in \{0, \ldots, b-1\}$.

Let us denote the simpler Galois ring $\mathbb{Z}/\langle p^k\rangle$ by $\mathbb{G}_0$. Following property of $\mathbb{G}$ is useful (proved in [DMS19, Appendix A.1]).

Lemma 10 (Automorphisms of $\mathbb{G}$)**.** *The Galois ring $\mathbb{G} \cong \mathbb{Z}[z]/\langle p^k, \varphi(z)\rangle$ has exactly $\deg(\varphi) = b$ many automorphisms ψ_j, for $j \in \{0, \ldots, b-1\}$, fixing $\mathbb{G}_0 = \mathbb{Z}/\langle p^k\rangle$. Each ψ_j maps $z_0 \to z_j$ and fixes only $\mathbb{G}_0$ if $(j, b) = 1$.*

By Lemma 18, f in $\mathbb{G}$ factors as $f = \prod_{i=0}^{b-1} f_i$, where $f_i(x) = (x - z_i)^e + ph_i(x)$ in $\mathbb{G}$. In particular, $f_0 = (x - z)^e + ph_0(x)$. We now use Lemma 10 to prove the following two lemmas, for connecting ramified factors of f in $\mathbb{G}_0[x]$ to ramified factors of f_i's in $\mathbb{G}[x]$.
Notation: We often denote $u(x, z) \in \mathbb{G}[x]$ by $u(z)$ to highlight the relevant parameter 'z'.

Lemma 11. *If $(\varphi^\delta - py) \mid f(x) \bmod p^k$, for $y \in \mathbb{G}_0[x]$, then for some $u(x, z) \in \mathbb{G}[x]$, $((x - z_i)^\delta - pu(z_i)) \mid f_i(x) \bmod \langle p^k, \varphi(z)\rangle$, for each $i \in \{0, \ldots, b-1\}$.*

Proof. Let $g := \varphi^\delta - py$. Then $g \mid f$ in $\mathbb{G}_0[x]$ and so in $\mathbb{G}[x]$. Now $(x - z)^\delta$ is a factor of $g \bmod p$, as $g \equiv \varphi^\delta \bmod p$, and so there is an $u \in \mathbb{G}[x]$ such that $((x-z)^\delta - pu(z))$ is a factor of g (Hensel Lemma 18); thus factor of f (since $g \mid f$) in $\mathbb{G}[x]$. Applying Lemma 10, we see that $g_i := ((x - z_i)^\delta - pu(z_i))$ is a factor of f, for each $i \in \{0, \ldots, b-1\}$. Now, g_i divides only $f_i \bmod p$ (by Hensel Lemma 18); and this finishes the proof. □

Lemma 12. *If there exists $u \in \mathbb{G}[x]$ s.t. $((x - z)^\delta - pu(z)) \mid f(x) \bmod \langle p^k, \varphi(z)\rangle$ then we can compute a $y \in \mathbb{G}_0[x]$ such that $(\varphi^\delta - py) \mid f(x) \bmod p^k$.*

Proof. Let $g_0 := (x - z)^\delta - pu(z)$, and $g_i := (x - z_i)^\delta - pu(z_i)$, for all $i \in [b-1]$. By applying automorphisms ψ_i (Lemma 10) on g_0, for $i \in [b-1]$, we can easily compute all other g_i's. Also, by applying automorphisms ψ_i, for $i \in [b-1]$, we see that each g_i divides $f(x)$ in $\mathbb{G}[x]$ (since ψ_i keeps $\mathbb{G}_0$ fixed and $f \in \mathbb{G}_0[x]$).

Now define $g(x, z) := \prod_{i=0}^{b-1} g_i$ in $\mathbb{G}[x]$. We see that all g_i's are co-prime, since they are co-prime over the field $\mathbb{G}/\langle p\rangle$ (i.e, $(x - z_i)^\delta$ is co-prime to $(x - z_j)^\delta$ for $i \neq j$). Hence, $g(x, z) \mid f$ in $\mathbb{G}[x]$.

Applying map ψ_1 on $g(x, z)$ we see that $g(x, z)$ remains unchanged over $\mathbb{G}$; as g_i's permute among each other. But ψ_1 keeps $\mathbb{G}_0$, and only $\mathbb{G}_0$, fixed (Lemma 10); hence $g \in \mathbb{G}_0[x]$ of degree $\delta \cdot b$. So, we can rewrite g as $g =: \varphi^\delta - py$, for a $y \in \mathbb{G}_0[x]$. □

The following extension of Reduction Theorem 19 of [DMS21], from $\mathbb{G}_0$ to the Galois ring $\mathbb{G}$, is evident.

Theorem 13 (Extended Reduction [DMS21])**.** *We have $((x - z_i)^\delta - pu(z_i)) \mid f_i(x) \bmod \langle p^k, \varphi(z)\rangle$ iff $E(u) \equiv 0 \bmod \langle p^k, \varphi(z), (x - z_i)^\ell\rangle$, for all $i \in \{0, \ldots, b-1\}$; where $\ell := \delta \cdot k$ and $E(u) := f_i(x)[(x - z_i)^{\delta(k-1)} + (x - z_i)^{\delta(k-2)}(pu) + \ldots + (pu)^{k-1}]$.*

Thus we now focus on finding a root of $E(u)$ in the ring $\mathbb{G}[x]/\langle (x - z)^\ell\rangle$.

3.2 Reduce root-finding in non-Galois ring to root-finding in Galois ring

In this section we show that finding a root of polynomial $E(u)$, in the ring $\mathbb{G}[x]/\langle (x-z)^\ell\rangle$, is equivalent to solving a system of ℓ polynomial equations in ℓ variables of degree same as $\deg_y(E) \leq k - 1$ over Galois ring $\mathbb{G}$. We achieve this by simply *eliminating* the variable x.

Theorem 14 (Reduction to SHN)**.** *Given $E(u) \in (\mathbb{Z}[z,x])[u]$ and the ring $\mathbb{G}[x]/\langle (x-z)^\ell \rangle$ where $\mathbb{G} = \mathbb{Z}[z]/\langle p^k, \varphi(z)\rangle$ as before. For new variable tuple $\mathbf{u} = (u_0, \ldots, u_{\ell-1})$ define a polynomial $E_{new}(\mathbf{u}) \in (\mathbb{Z}[z,x])[\mathbf{u}]$ as $E_{new}(\mathbf{u}) := E(u_0 + (x-z)u_1 + \cdots + (x-z)^{\ell-1}u_{\ell-1})$.*

Let $\mathcal{F}(\mathbf{u}) := \{E_0, \ldots, E_{\ell-1}\}$ be a system of polynomial equations, where $E_i(\mathbf{u}) \in (\mathbb{Z}[z])[\mathbf{u}]$ with $\deg_z(E_i) < \deg(\varphi(z))$ and $\deg_{\mathbf{u}}(E_i) < k$, such that

$$E_{new}(\mathbf{u}) \equiv E_0(\mathbf{u}) + E_1(\mathbf{u}) \cdot (x-z) + \cdots + E_{\ell-1}(\mathbf{u}) \cdot (x-z)^{\ell-1} \bmod \langle p^k, \varphi(z), (x-z)^\ell \rangle.$$

Then for $\mathbf{a} \in \mathbb{G}^\ell$, $E_{new}(\mathbf{a}) \equiv 0 \bmod \langle p^k, \varphi(z), (x-z)^\ell\rangle$ iff $\mathcal{F}(\mathbf{a}) \equiv 0 \bmod \langle p^k, \varphi(z)\rangle$.

Proof. Following the definition of $E_{\mathsf{new}}(\mathbf{u})$, we can rewrite $E_{\mathsf{new}}(\mathbf{u})$, for some polynomials $E_i(\mathbf{u}) \in (\mathbb{Z}[z])[\mathbf{u}]$ as

$$E_{\mathsf{new}}(\mathbf{u}) \;=\; E_0(\mathbf{u}) + E_1(\mathbf{u})(x-z) + \cdots + E_{\ell-1}(\mathbf{u})(x-z)^{\ell-1}\,.$$

Now, $E_{\mathsf{new}}(\mathbf{a}) \equiv 0 \bmod \langle p^k, \varphi(z), (x-z)^\ell\rangle$
$\iff$ $E_{\mathsf{new}}(\mathbf{a}) =: t_x(x-z)^\ell$, for some $t_x \in \mathbb{G}[x]$.
$\iff$ $E_0(\mathbf{a}) + \cdots + (x-z)^{\ell-1}E_{\ell-1}(\mathbf{a}) \;=\; t_x(x-z)^\ell$.

Since degree wrt x of LHS is at most $\ell-1$, so $(x-z)^\ell$ can not divide it over $\mathbb{G}$. So $E_i(\mathbf{a})$ vanishes in $\mathbb{G}$, for each $i \in \{0, \ldots, \ell-1\}$. In other words, $\mathbf{a}$ is $\mathbb{G}$-root of the system $\mathcal{F}(\mathbf{u})$.

Now we prove the other direction. Given that, $E_i(\mathbf{a}) \equiv 0 \bmod \langle p^k, \varphi(z)\rangle$, for each $i \in \{0, \ldots, \ell-1\}$. We easily deduce: $E_{\mathsf{new}}(\mathbf{a}) \equiv 0 \bmod \langle p^k, \varphi(z), (x-z)^\ell\rangle$.

Moreover, this reduction is efficient when the parameter k is fixed; because $\deg_{\mathbf{u}}(E) < k$ and so E_{new} has at most $\binom{\ell+k}{\ell} \le (\ell+k)^k$ monomials. □

3.3 Algorithm: Proof of Theorem 2 & Corollary 3

Input: Given $f \in \mathbb{Z}[x]$ and a prime-power p^k such that $f \equiv \varphi^e \bmod p$, where $\varphi \in \mathbb{Z}[x]$ is irreducible mod p; and $\deg(f) = b \cdot e$, where $b := \deg(\varphi)$.

Output: A ramification-degree-δ factor $g(x)$ of $f(x) \bmod p^k$.

Algorithm 4 Factoring $f(x) \bmod p^k$

1: **procedure** FACTOR($f(x), p^k$)
2: Let $g = \varphi^\delta - p \cdot y$, where $y = y(x)$ is an unknown such that $g \mid f \bmod p^k$.
3: Consider Galois ring $\mathbb{G} := \mathbb{Z}[z]/\langle p^k, \varphi(z)\rangle$, where $\varphi(x)$ splits completely and z is a $\mathbb{G}$-root of $\varphi(x)$. (Other roots are conjugates of z, by Lemma 10.)
4: Factorize $\varphi(x)$ over $\mathbb{G}/\langle p\rangle$ into b linear (co-prime) factors using [CZ81] and lift to $\mathbb{G}$ using Hensel's lifting to obtain a co-prime factorization $f =: \prod_{i=0}^{b-1} f_i$.
5: Over $\mathbb{G}$, let $g =: \prod_{i=0}^{b-1} g_i$ be a co-prime factorizations, such that $g_i \mid f_i$ for all i (Lemma 11). Fix $j \in \{0, \ldots, b-1\}$ and consider $g_j =: (x-z)^\delta - pu$.
6: Using Theorem 13 reduce to root-finding question of $E(u) \equiv 0 \bmod \langle p^k, \varphi(z), (x-z)^\ell\rangle$, where $E(u) := f_j \cdot [(x-z)^{\delta(k-1)} + (x-z)^{\delta(k-2)}(pu) + \cdots + (pu)^{k-1}]$.
7: Substituting $u \to u_0 + (x-z)u_1 + \cdots + (x-z)^{\ell-1}u_{\ell-1}$, compute $E_0(\mathbf{u}), \ldots, E_{\ell-1}(\mathbf{u}) \in \mathbb{G}[\mathbf{u}]$ such that
$E(u) =: E_0 + (x-z)E_1 + \cdots + (x-z)^{\ell-1}E_{\ell-1} \bmod \langle p^k, \varphi(z), (x-z)^\ell\rangle$.
8: Find a $\mathbb{G}$-root $(a_0, \ldots, a_{\ell-1})$ of the system $\mathcal{F} := \{E_0, \ldots, E_{\ell-1}\}$ using Algorithm 1.
9: **if** no solution exists **then return** $\{\}$, i.e., no such factor g exists.

10: $u := a_0 + (x-z)a_1 + \cdots + (x-z)^{\ell-1}a_{\ell-1}$ is a solution of $E(u)$ mod $\langle p^k, \varphi(z), (x-z)^\ell\rangle$ (from Theorem 14). This gives us the factor $g_j = (x-z)^\delta - pu$ (Theorem 13).
11: Using $\mathbb{G}$-automorphisms (Lemma 12 & Step 4), we can compute $g = \varphi^\delta - py$ from g_j.
12: **return** g

Remark 3. *One can ask for a simpler Nullstellensatz approach: Why do we not reduce root-finding of $E(u)$ mod $\langle p^k, \varphi(z), (x-z)^\ell\rangle$ to directly solving a system of equations modulo p, instead of modulo p^k? For e.g., by further substituting $u_i \to u_{i,0} + pu_{i,1} + \ldots + p^{k-1}u_{i,k-1}$, for each $i \in \{0,\ldots,\ell-1\}$, $u_{i,j}$'s in $\mathbb{F}_p$?*

The issue is that we need to divide functions of $u_{i,j}$'s by p; and this only makes sense when we think of $u_{i,j}$'s as p-adic. See Example 5 for a more concrete discussion.

Now we prove Theorem 2 in a way that already subsumes Corollary 3.

Proof of Theorem 2. We have $f(x) = \varphi(x)^e + ph(x)$ and prime-power is p^k. A factor g of f mod p^k has the form $g = \varphi^\delta - py$ (ramification-degree δ) where we want to compute $y \in \mathbb{G}_0[x]$ such that $\deg(y) < \delta\deg(\varphi)$; to keep g monic.

Now over $\mathbb{G}$, f and g have co-prime factorizations as $f = \prod_{j=0}^{b-1} f_j$ and $\prod_{j=0}^{b-1} g_j$. By Lemma 11 if $g \mid f$ mod p^k then $g_j \mid f_j$ over $\mathbb{G}$, for all j. For a fixed $i \in \{0,\ldots,b-1\}$, let $f_i =: (x-z)^e + ph_i(x,z)$ and $g_i =: (x-z)^\delta - pu(x,z)$ (where u is unknown). Using Lemmas 11 and 12, it is sufficient to find unknown g_i. Computing factorizations of f and φ (using Hensel lifting 18) and getting g from g_i (Lemma 12) takes time $\mathsf{poly}(\deg(f), k\log p)$.

Using Theorem 13, finding g_i is reduced to finding a root, of $E(u) := f_i \cdot [(x-z)^{\delta(k-1)} + (x-z)^{\delta(k-2)}(pu) + \cdots + (pu)^{k-1}]$, in $\mathbb{G}[x]/\langle(x-z)^\ell\rangle$, where $\ell := \delta k$. Computing $E(u)$ takes time $\mathsf{poly}(\deg(f), \ell, \log p)$.

By Theorem 14, finding a root of $E(u)$ in $\mathbb{G}[x]/\langle(x-z)^\ell\rangle$ is reduced to finding $\mathbb{G}$-root of a system of ℓ-variate ℓ polynomial equations $\mathcal{F} := \{E_0(\mathbf{u}), \ldots, E_{\ell-1}(\mathbf{u})\}$ of degree at most $k-1$. Using Theorem 1, we get a solution of $\mathcal{F}$ in $\mathbb{G}$. This immediately gives us a root u of $E(u)$ mod $\langle p^k, \varphi(z), (x-z)^\ell\rangle$; thus we find the factor $g_i = (x-z)^\delta - pu$. The time complexity is dominated by time taken to find a solution of $\mathcal{F}$; which is $\mathsf{poly}(\deg(\mathcal{F})^{(\ell k)^{O((\ell k)^2)}}, \log p, \deg(f))$.

Since $\deg(\mathcal{F}) < k$ and $\ell = \delta k$, so the total time taken is $\mathsf{poly}(k^{(\delta k^2)^{O((\delta k^2)^2)}}, \log p, \deg(f))$. Since $\delta + k$ is constant, the time complexity becomes $\mathsf{poly}(\deg(f), \log p)$. □

4 Conclusion and future work

In this article, we deal with the problem of finding a common root of a system of polynomial equations, whose extension to Galois rings has been explored here. We extend the results of [HW99] to find roots of a system of equations in Galois rings. Furthermore, faster algorithms for polynomial system solving over $\mathbb{F}_q$ will lead to fine-grained improvements in the complexity of SHN_{p^k}.

We also make progress towards finding factors of univariate polynomials in prime-power rings. Interest in this problem developed after Hensel [Hen18] gave a method to lift (co-prime) factors to modulo any prime-powers. It is easier to factorize in fields, as seen before, but factorization modulo small prime powers has been elusive to computer scientists; owing to the fact that these rings are not integral domains and there can be exponentially many factors. This difficulty has been explained in [CL01, Sir17, vzGH96, vzGH98, vzGP01]. Overcoming some of these obstructions, we generalize [DMS21] to find factors of *small* ramification-degree modulo p^k, for large primes p, and *small* k.

This paper motivates and leaves the following questions open.

1. Test solvability of a system of n-variate polynomials mod p^k, for fixed n and arbitrary k (resp. over the p-adic integers $\mathbb{Z}_p$).
2. Count points on curves modulo p^k, over classical or quantum computers. (Some progress has been made in [CS23] recently.)
3. Find a large ramification-degree factor of $f(x)$ mod p^5; and extend it to any constant k.
4. Find a small degree factor of f mod p^k, for growing p, k.
5. Test irreducibility of f mod p^k.

Acknowledgements: We thank the anonymous reviewers for their helpful comments and pointing out related references. A part of this work appeared in the Master's thesis of the first author ([Cha22]) and the PhD thesis of the second author ([Dwi23]). N.S. thanks the funding support from DST-SERB (CRG/2020/000045) and N.Rama Rao Chair.

References

[Arm05a] Marc André Armand. Improved list decoding of generalized reed-solomon and alternant codes over galois rings. *IEEE transactions on information theory*, 51(2):728–733, 2005. 4

[Arm05b] Marc André Armand. List decoding of generalized reed-solomon codes over commutative rings. *IEEE transactions on information theory*, 51(1):411–419, 2005. 4

[BCR16] Jingguo Bi, Qi Cheng, and J Maurice Rojas. Sublinear root detection and new hardness results for sparse polynomials over finite fields. *SIAM Journal on Computing*, 45(4):1433–1447, 2016. 2

[BDT21] Charles Bouillaguet, Claire Delaplace, and Monika Trimoska. A simple deterministic algorithm for systems of quadratic polynomials over $\mathbb{F}_2$. *Cryptology ePrint Archive*, 2021. 2

[Ber67] Elwyn R Berlekamp. Factoring polynomials over finite fields. *Bell System Technical Journal*, 46(8):1853–1859, 1967. 3

[BFSS13] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic boolean systems. *Journal of Complexity*, 29(1):53–75, 2013. 2

[Bha97] Manjul Bhargava. P-orderings and polynomial functions on arbitrary subsets of dedekind rings. *Journal fur die Reine und Angewandte Mathematik*, 490:101–128, 1997. 5

[BKW19] Andreas Björklund, Petteri Kaski, and Ryan Williams. Solving systems of polynomial equations over GF(2) by a parity-counting self-reduction. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. 2

[BLQ13] Jérémy Berthomieu, Grégoire Lecerf, and Guillaume Quintin. Polynomial root finding over local rings and application to error correcting codes. *Applicable Algebra in Engineering, Communication and Computing*, 24(6):413–443, 2013. 2, 4, 5, 6, 8

[Bro87] W Dale Brownawell. Bounds for the degrees in the Nullstellensatz. *Annals of Mathematics*, 126(3):577–591, 1987. 3

[BS86] Zenon Ivanovich Borevich and Igor Rostislavovich Shafarevich. *Number theory*, volume 20. Academic press, 1986. 4

[Buc65] Bruno Buchberger. Ein algorithmus zum auffinden der basiselemente des restklassenringes nach einem nulldimensionalen polynomideal. *PhD thesis, Universität Innsbruck*, 1965. 31

[CG00] David G Cantor and Daniel M Gordon. Factoring polynomials over p-adic fields. In *International Algorithmic Number Theory Symposium*, pages 185–208. Springer, 2000. 4, 5

[CGRW19] Qi Cheng, Shuhong Gao, J Maurice Rojas, and Daqing Wan. Counting roots for polynomials modulo prime powers. *The Open Book Series (ANTS XIII)*, 2(1):191–205, 2019. 5, 9

[Cha22] Sayak Chakrabarti. Multivariate polynomials modulo prime powers: their roots, zeta-function and applications. Master's thesis, Dept of CSE, IIT Kanpur, India, 2022. 21

[Chi87] AL Chistov. Efficient factorization of polynomials over local fields. *Dokl. Akad. Nauk SSSR*, 293(5):1073–1077, 1987. 4, 5

[Chi94] AL Chistov. Algorithm of polynomial complexity for factoring polynomials over local fields. *Journal of mathematical sciences*, 70(4):1912–1933, 1994. 4, 5

[Chi21] Alexander L Chistov. An effective algorithm for deciding solvability of a system of polynomial equations over p-adic integers. *Algebra i Analiz*, 33(6):162–196, 2021. 3

[CL01] Howard Cheng and George Labahn. Computing All Factorizations in $\mathbb{Z}_N[x]$. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC'01*, pages 64–71, 2001. 4, 5, 20

[CLO13] David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013. 2, 31

[CP56] M Chojnacka-Pniewska. Sur les congruences aux racines données. In *Annales Polonici Mathematici*, volume 3, pages 9–12. Instytut Matematyczny Polskiej Akademii Nauk, 1956. 5

[CS23] Sayak Chakrabarti and Nitin Saxena. An effective description of the roots of multivariates mod p^k and the related Igusa's local zeta function. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC'23 (to appear), Tromsø, Norway*. ACM, 2023. 3, 5, 21

[CZ81] David G Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, pages 587–592, 1981. 3, 19

[DH01] Jan Denef and Kathleen Hoornaert. Newton polyhedra and Igusa's local zeta function. *Journal of number Theory*, 89(1):31–64, 2001. 5

[Din21a] Itai Dinur. Cryptanalytic applications of the polynomial method for solving multivariate equation systems over GF(2). In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I*, pages 374–403, 2021. 2

[Din21b] Itai Dinur. Improved algorithms for solving polynomial systems over GF(2) by multiple parity-counting. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2550–2564. SIAM, 2021. 2

[DM97] Bruce Dearden and Jerry Metzger. Roots of polynomials modulo prime powers. *European Journal of Combinatorics*, 18(6):601–606, 1997. 5

[DMS19] Ashish Dwivedi, Rajat Mittal, and Nitin Saxena. Counting Basic-Irreducible Factors Mod p^k in Deterministic Poly-Time and p-Adic Applications. In Amir Shpilka, editor, *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:29, 2019. 5, 6, 8, 9, 18, 32

[DMS21] Ashish Dwivedi, Rajat Mittal, and Nitin Saxena. Efficiently factoring polynomials modulo p^4. *Journal of Symbolic Computation*, 104:805 – 823, 2021. Preliminary version appeared in The 44th ACM International Symposium on Symbolic and Algebraic Computation (ISSAC) 2019. 4, 5, 6, 7, 8, 17, 18, 20, 28

[DS20] Ashish Dwivedi and Nitin Saxena. Computing Igusa's local zeta function of univariates in deterministic polynomial-time. 14*th Algorithmic Number Theory Symposium (ANTS XIV), Open Book Series*, 4(1):197–214, 2020. 3, 5, 9

[Dub90] Thomas W Dubé. The structure of polynomial ideals and gröbner bases. *SIAM Journal on Computing*, 19(4):750–773, 1990. 17

[Dwi23] Ashish Dwivedi. *Polynomials over composites: Compact root representation via ideals and algorithmic consequences*. PhD thesis, Dept of CSE, IIT Kanpur, India, 2023. 21

[EK90] Andrzej Ehrenfeucht and Marek Karpinski. *The computational complexity of (xor, and)-counting problems*. International Computer Science Inst., 1990. 2

[ELS+14] Matt Elder, Junghee Lim, Tushar Sharma, Tycho Andersen, and Thomas Reps. Abstract domains of affine relations. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 36(4):1–73, 2014. 5

[FS15] Michael A Forbes and Amir Shpilka. Complexity theory column 88: Challenges in polynomial factorization. *ACM SIGACT News*, 46(4):32–49, 2015. 3

[GGL08] Parikshit Gopalan, Venkatesan Guruswami, and Richard J Lipton. Algorithms for modular counting of roots of multivariate polynomials. *Algorithmica*, 50(4):479–496, 2008. 2

[GNP12] Jordi Guàrdia, Enric Nart, and Sebastian Pauli. Single-factor lifting and factorization of polynomials over local fields. *J. Symb. Comput.*, 47(11):1318–1346, November 2012. 4

[Gop77] Valerii Denisovich Goppa. Codes associated with divisors. *Problemy Peredachi Informatsii*, 13(1):33–39, 1977. 4

[Gre66] Marvin J Greenberg. Rational points in henselian discrete valuation rings. *Publications Mathématiques de l'IHÉS*, 31:59–64, 1966. 3

[GS20] Abhibhav Garg and Nitin Saxena. Special-case algorithms for blackbox radical membership, Nullstellensatz and transcendence degree. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*, pages 186–193, 2020. 3

[GTZ88] Patrizia Gianni, Barry Trager, and Gail Zacharias. Gröbner bases and primary decomposition of polynomial ideals. *Journal of Symbolic Computation*, 6(2-3):149–167, 1988. 31

[Hau10] Herwig Hauser. On the problem of resolution of singularities in positive characteristic (or: a proof we are still waiting for). *Bulletin of the American Mathematical Society*, 47(1):1–30, 2010. 3

[Hen18] Kurt Hensel. Eine neue theorie der algebraischen zahlen. *Mathematische Zeitschrift*, 2(3):433–452, Sep 1918. 3, 6, 20, 28

[Hir64] Heisuke Hironaka. Resolution of singularities of an algebraic variety over a field of characteristic zero: Ii. *Annals of Mathematics*, pages 205–326, 1964. 3

[HKC+94] A Roger Hammons, P Vijay Kumar, A Robert Calderbank, Neil JA Sloane, and Patrick Solé. The $\mathbb{Z}_4$-linearity of kerdock, preparata, goethals, and related codes. *IEEE Transactions on Information Theory*, 40(2):301–319, 1994. 2, 4

[HVLP98] Tom Høholdt, Jacobus H Van Lint, and Ruud Pellikaan. Algebraic geometry codes. *Handbook of coding theory*, 1(Part 1):871–961, 1998. 4

[HW99] M-D Huang and Y-C Wong. Solvability of systems of polynomial congruences modulo a large prime. *computational complexity*, 8(3):227–257, 1999. Preliminary version appeared in The IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS) 1996. 2, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 20, 28, 30, 31, 32, 33

[IR18] Gábor Ivanyos and Lajos Rónyai. Chevalley-Warning theorem in quantum computing. *ERCIM NEWS*, 112:28–29, 2018. 5

[Kal92] Erich Kaltofen. Polynomial factorization 1987–1991. In *Latin American Symposium on Theoretical Informatics*, pages 294–313. Springer, 1992. 3

[Kat09] Daniel Katz. Point count divisibility for algebraic sets over $\mathbb{Z}/p^\ell\mathbb{Z}$ and other finite principal rings. *Proceedings of the American Mathematical Society*, 137(12):4065–4075, 2009. 5

[Kay05] Neeraj Kayal. Solvability of a system of bivariate polynomial equations over a finite field. In *International Colloquium on Automata, Languages, and Programming*, pages 551–562. Springer, 2005. 2

[Kli97] Adam Klivans. Factoring polynomials modulo composites. Technical report, Carnegie-Mellon Univ, Pittsburgh PA, Dept of CS, 1997. 5

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 206–222. Springer, 1999. 2

[KRRZ20] Leann Kopp, Natalie Randall, J Maurice Rojas, and Yuyu Zhu. Randomized polynomial-time root counting in prime power rings. *Mathematics of Computation*, 89(321):373–385, 2020. 5

[KU11] Kiran S Kedlaya and Christopher Umans. Fast polynomial factorization and modular composition. *SIAM Journal on Computing*, 40(6):1767–1802, 2011. 3

[Lau04] Alan GB Lauder. Counting solutions to equations in many variables over finite fields. *Foundations of Computational Mathematics*, 4(3):221–267, 2004. 5

[LN94] Rudolf Lidl and Harald Niederreiter. *Introduction to finite fields and their applications*. Cambridge university press, 1994. 3

[LPT+17] Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, Ryan Williams, and Huacheng Yu. Beating brute force for systems of polynomial equations over finite fields. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2190–2202. SIAM, 2017. 2

[Mau01] Davesh Maulik. Root sets of polynomials modulo prime powers. *Journal of Combinatorial Theory, Series A*, 93(1):125–140, 2001. 5

[MOS05] Markus Müller-Olm and Helmut Seidl. Analysis of modular arithmetic. In *European Symposium on Programming*, pages 46–60. Springer, 2005. 5

[MOS07] Markus Müller-Olm and Helmut Seidl. Analysis of modular arithmetic. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 29(5):29–es, 2007. 5

[MR75] Murray Marshall and Garry Ramage. Zeros of polynomials over finite principal ideal rings. *Proceedings of the American Mathematical Society*, 49(1):35–38, 1975. 5

[NRS17] Vincent Neiger, Johan Rosenkilde, and Éric Schost. Fast computation of the roots of polynomials over the ring of power series. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, pages 349–356, 2017. 8

[NZM13] Ivan Niven, Herbert S Zuckerman, and Hugh L Montgomery. *An introduction to the theory of numbers*. John Wiley & Sons, 2013. 3

[Pan95] Peter N Panayi. *Computation of Leopoldt's P-adic regulator*. PhD thesis, University of East Anglia, 1995. 4

[Pat96] Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 33–48. Springer, 1996. 2

[RRZ21] Caleb Robelle, J Maurice Rojas, and Yuyu Zhu. Sub-linear point counting for variable separated curves over prime power rings. *arXiv preprint arXiv:2102.01626*, 2021. 5

[RZ22] J Maurice Rojas and Yuyu Zhu. Root repulsion and faster solving for very sparse polynomials over p-adic fields. *Journal of Number Theory*, 241:655–699, 2022. 3

[Săl05] Ana Sălăgean. Factoring polynomials over $\mathbb{Z}_4$ and over certain galois rings. *Finite fields and their applications*, 11(1):56–70, 2005. 4, 5

[Sch74] Wolfgang M Schmidt. A lower bound for the number of solutions of equations over finite fields. *Journal of Number Theory*, 6(6):448–480, 1974. 30

[Sie55] Wacław Sierpiński. Remarques sur les racines d'une congruence. *Annales Polonici Mathematici*, 1(1):89–90, 1955. 5

[Sir17] Carlo Sircana. Factorization of polynomials over $\mathbb{Z}/(p^n)$. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, pages 405–412. ACM, 2017. 5, 20

[SR14] Robert L Surowka and Kenneth W Regan. Polynomials modulo composite numbers: Ax-katz type theorems for the structure of their solution sets. *arXiv preprint arXiv:1404.4852*, 2014. 5

[Sud97] Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of complexity*, 13(1):180–193, 1997. 3

[TKSG08] Neal Tew, Priyank Kalla, Namrata Shekhar, and Sivaram Gopalakrishnan. Verification of arithmetic datapaths using polynomial function models and congruence solving. In *2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 122–128. IEEE, 2008. 5

[TV13] Michael Tsfasman and Serge G Vladut. *Algebraic-geometric codes*, volume 58. Springer Science & Business Media, 2013. 4

[TVZ82] Michael A Tsfasman, SG Vlădutx, and Th Zink. Modular curves, shimura curves, and goppa codes, better than varshamov-gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982. 4

[vzGH96] Joachim von zur Gathen and Silke Hartlieb. Factorization of polynomials modulo small prime powers. Technical report, Paderborn Univ, 1996. 4, 5, 7, 20

[vzGH98] Joachim von zur Gathen and Silke Hartlieb. Factoring modular polynomials. *Journal of Symbolic Computation*, 26(5):583–606, 1998. (Conference version in ISSAC'96). 4, 5, 20

[vzGKS96] Joachim von zur Gathen, Marek Karpinski, and Igor Shparlinski. Counting curves and their projections. *computational complexity*, 6(1):64–99, 1996. 2

[vzGP01] Joachim von zur Gathen and Daniel Panario. Factoring polynomials over finite fields: A survey. *Journal of Symbolic Computation*, 31(1-2):3–17, 2001. 3, 20

[Wal97] Judy L Walker. The nordstrom-robinson code is algebraic-geometric. *IEEE Transactions on Information Theory*, 43(5):1588–1593, 1997. 4

[Wal99] Judy L Walker. Algebraic geometric codes over rings. *Journal of pure and applied Algebra*, 144(1):91–110, 1999. 4

[Zas69] Hans Zassenhaus. On hensel factorization, I. *Journal of Number Theory*, 1(3):291–311, 1969. 28

[Zas78] Hans Zassenhaus. A remark on the hensel factorization method. *Mathematics of Computation*, 32(141):287–292, 1978. 28

[ZG03] WA Zuniga-Galindo. Computing Igusa's local zeta functions of univariate polynomials, and linear feedback shift registers. *Journal of Integer Sequences*, 6(2):3, 2003. 5

[Zhu20] Yuyu Zhu. *Trees, point counting beyond fields, and root separation*. PhD thesis, Texas A&M University, 2020. 5

A Preliminaries

A.1 Notations

For an n-tuple $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$ in $\mathbb{F}^n$, we denote $|\mathbf{a}| = \Sigma_i a_i$, and $\mathbf{a}! = a_1! \cdots a_n!$, where $\mathbf{a} \in \mathbb{Z}^n$.

Definition 15 (Taylor expansion/series)**.** *For a polynomial $f(\mathbf{x})$ of degree d over any field, we can express it as*

$$f(\mathbf{a}+\mathbf{x}) \;=\; \sum_{\ell=0}^{\infty} \left(\sum_{|\mathbf{i}|=\ell} \frac{\partial_{\mathbf{x}^\mathbf{i}} f(\mathbf{a})}{\mathbf{i}!} \cdot \prod_{j=1}^{n} x_j^{i_j} \right), \tag{2}$$

where $\partial_{\mathbf{x}^\mathbf{i}} f := \frac{\partial^{|\mathbf{i}|} f}{\partial x_1^{i_1} \ldots \partial x_n^{i_n}}$ is an order-$|\mathbf{i}|$ *partial derivative.*

Note that the value of $\frac{\partial_{\mathbf{x}^\mathbf{i}} f(\mathbf{a})}{\mathbf{i}!}$ is infact an integer, as it is in fact an integer, since it is merely a coefficient of a Taylor expansion about an integral shift.

We further define some terms which have been used frequently throughout the paper.

Definition 16 (Non-singular roots)**.** *A root $\mathbf{r}$ of a polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ is called a* non-singular root *if some first-order derivative $f'(\mathbf{x})$ (say $\partial_{x_j} f$) does not vanish at $\mathbf{r}$.*

Definition 17 (Absolutely irreducible)**.** *Let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a polynomial, where $\mathbb{F}$ is a field and $\overline{\mathbb{F}}$ is its algebraic closure. Then $f(\mathbf{x})$ is* absolutely irreducible *if it is irreducible over $\overline{\mathbb{F}}$.*

If $f(\mathbf{x})$ is irreducible over $\mathbb{F}$ but factorizes in $\overline{\mathbb{F}}$, we call it relatively irreducible*.*

We extend this definition to ideals as well. Given an irreducible ideal $\mathtt{I}$ over $\mathbb{F}$, using Theorem 24 ([HW99, Theorem 2.6]), we can map it to a hypersurface $\mathtt{H}$ given by the variety of a single polynomial h. If this polynomial h is absolutely irreducible, we refer to the ideal $\mathtt{I}$ as absolutely irreducible.

A.2 Simplifying the factors of $f(x)$ modulo p^k

Without loss of generality, we can assume given $f \in \mathbb{Z}[x]$ to be monic mod p^k (i.e., its leading coefficient is 1). Eg. $f = px^3 - x^2 + 1$ can be written as $(-x^2+1)\left(px^3/(-x^2+1)+1\right)$, where the second factor is clearly a *unit* mod p^k. Thus, for nontrivial factorization we only need to consider the monic polynomial $g := x^2 - 1$.

The following variant of Hensel's Lemma simplifies the task even more. It lets us assume $f(x) = \varphi^e + pg$, where $\varphi, g \in \mathbb{Z}[x]$ such that φ is monic, and $\varphi \bmod p$ is irreducible. Again, we can assume: $\deg(f) = e \cdot \deg(\varphi) > \deg(g)$.

Lemma 18 (Hensel's lemma for co-prime factors [Hen18, Zas69, Zas78])**.** *Let R be a commutative ring with unity, denote the polynomial ring over it by $R[x]$. Let $\mathtt{I}$ be an ideal of ring R. Given a polynomial $f(x) \in R[x]$, suppose f factorizes as $f = gh \bmod \mathtt{I}$, such that $gu + hv = 1 \bmod \mathtt{I}$ (for some $g, h, u, v \in R[x]$). Then, given any $l \in \mathbb{N}$, we can efficiently compute $g^*, h^*, u^*, v^* \in R[x]$, such that,*

$$f \;=\; g^* \cdot h^* \bmod \mathtt{I}^l\,.$$

Here $g^* = g \bmod \mathtt{I}$, $h^* = h \bmod \mathtt{I}$ and $g^*u^* + h^*v^* = 1 \bmod \mathtt{I}^l$ (i.e., the lift is pseudo *co-prime too). Moreover, g^* and h^* are unique up to multiplication by a unit.*

Factoring to Root-finding: Interestingly, [DMS21] showed that finding a factor $g(x) := \varphi^\delta - py$ of $f(x) = \varphi^e + ph(x) \bmod p^k$, where $\varphi, h \in \mathbb{Z}[x]$ with φ irreducible mod p and $\delta < e$, is equivalent to finding a root of a special polynomial $E(y) \in \mathbb{Z}[x, y]$ (E as defined in Theorem 19) modulo a *bi*-generated ideal $\langle p^k, \varphi^{\delta k}\rangle$.

Theorem 19 (Reduction [DMS21, Thm.11])**.** *With f, φ, g as above, g is a factor of f modulo p^k if and only if $E(y) := f \cdot (\varphi^{\delta(k-1)} + \varphi^{\delta(k-2)}(py) + \cdots + \varphi^\delta(py)^{k-2} + (py)^{k-1}) \equiv 0 \bmod \langle p^k, \varphi^{\delta k}\rangle$.*

A.3 Arithmetic on roots modulo p^k

As always, define the unramified extension $\widehat{\mathbb{G}} := \mathbb{Z}_p[z]/\langle\varphi(z)\rangle$; its Galois ring $\mathbb{G} := \mathbb{Z}[z]/\langle p^k, \varphi(z)\rangle$, and its Galois field $\mathbb{F}_q$. For a polynomial $f(x_1, x_2)$ over $\widehat{\mathbb{G}}$, we define the *effective polynomial* as $f(x_1, x_2) \bmod p$, where the coefficients are in $\mathbb{F}_q$ (w.l.o.g. $f(x_1, x_2) \bmod p$ is non-constant). Similarly, the *effective degree* of $f(x_1, x_2)$ is the degree of $f(x_1, x_2) \bmod p$.

The term *valuation* of a, $v(a)$, is defined as the largest integer $v \geq 0$ such that $p^v | a$. Now, we define a *local root* of $f(x_1, x_2)$ as a $\mathbb{F}_q$-root of the effective polynomial $f(x_1, x_2) \bmod p$. For a local root (a_1, a_2), *local valuation* is defined as $v(f(a_1, a_2))$.

Definition 20 (Val-multiplicity)**.** Val-multiplicity *of local root* $\mathbf{a}$ *(viewed as an element of $\widehat{\mathbb{G}}$) is defined as $v(f(\mathbf{a}+p\mathbf{x}))$, i.e., the minimum valuation of the coefficients of the polynomial thus formed.*

We now show that a non-singular $\mathbb{F}_q$ root can be lifted to the p-adic $\widehat{\mathbb{G}}$.

Proposition 3 (Hensel's p-adic lift of roots)**.** *Given a polynomial $f(\mathbf{x}) \in \widehat{\mathbb{G}}[\mathbf{x}]$, let $\mathbf{a} \in \mathbb{F}_q^n$ be a non-singular root of $f(\mathbf{x}) \bmod p$ over $\mathbb{F}_q$. Then, we have the following*

1. *Consider $\mathbf{a}$ as an element of $\widehat{\mathbb{G}}$ too. Val-multiplicity of the root is 1, which implies that the effective degree becomes 1 after lifting the polynomial by $\mathbf{a}$.*
2. *We can lift the root $\mathbf{a}$ to root $\hat{\mathbf{a}} \in \widehat{\mathbb{G}}^n$, such that $f(\hat{\mathbf{a}}) = 0$ over $\widehat{\mathbb{G}}$.*

Proof of Part 1. We can write the polynomial during lifting as

$$f(\mathbf{a} + p\mathbf{x}) \;=\; \sum_{\ell=0}^{d} \left(\sum_{|\mathbf{i}|=\ell} \frac{\partial_{\mathbf{x}^\mathbf{i}} f(\mathbf{a})}{\mathbf{i}!} \cdot (px_1)^{i_1} \dots (px_n)^{i_n} \right). \tag{3}$$

Since the root $\mathbf{a}$ is non-singular, $\partial_{x_j} f(\mathbf{a}) \neq 0 \bmod p$, which implies that the term $\Sigma_{j=1}^{n} (\partial_{x_j} f(\mathbf{a}))$ (px_j) is divisible by p, but not by p^2. By the above Taylor's expansion, we can infer that $p | f(\mathbf{a} + p\mathbf{x})$ but $p^2 \nmid f(\mathbf{a} + p\mathbf{x})$, implying that the val-multiplicity of the local-root $\mathbf{a}$ is 1.

Now, the terms in the multivariate Taylor's series with $\ell \geq 2$ are divisible by p^2. These terms, after division by p (owing to the val-multiplicity being 1), will vanish modulo p. Thus, the effective polynomial only contains *linear* terms; implying that the effective degree is 1. □

Proof of Part 2. Since the effective degree is currently 1, we can consider the polynomial as

$$f(\mathbf{x}) \;=\; l_1 x_1 + \dots + l_n x_n + m + p \cdot g(\mathbf{x}) \,, \tag{4}$$

where (w.l.o.g.) l_n is a *nonzero* constant modulo p, m, l_i's are constants, and $g(\mathbf{x})$ is a polynomial over $\widehat{\mathbb{G}}$. Since the effective polynomial is linear, we can fix any value $(a_{1,1}, \dots, a_{1,n-1})$ to the first $(n-1)$-precision digits, and find the unique value of x_n; given by $a_{1,n} = -(l_1 a_{1,1} + \dots + l_{n-1} a_{1,n-1} + m)/l_n$. We denote this root by $\mathbf{a}_1$, giving the root for two-steps of lifting as $\mathbf{a} + p\mathbf{a}_1$. The next lifting step starts with the substitution $f(\mathbf{a}_1 + p\mathbf{x})$, the polynomial will be lifted as:

$$\begin{aligned} &l_1(a_{1,1} + px_1) + \dots + l_{n-1}(a_{n-1,1} + px_{n-1}) + \\ &\qquad l_n(-(l_1 a_{1,1} + \dots + l_{n-1} a_{1,n-1} + m)/l_n + px_n) + m + pg(\mathbf{a}_1 + p\mathbf{x}) \,. \end{aligned}$$

In particular, any non-constant monomial from the pg part, will be divisible by p^2; while the linear terms which were initially in the effective polynomial will have valuation exactly p. Thus, the val-multiplicity of the root $\mathbf{a}_1$ is 1, and the effective degree remains the same (i.e., 1). The coefficients of x_j's in the effective polynomial will again be l_j's, for $j \in [n]$; the only change being that the constant m will now change.

We again continue our process of fixing the first $n-1$ precision digits to any value, finding the value of the n-th precision digit, and then lifting; whence the effective polynomial remains a linear of the form $l_1 x_1 + \dots l_n x_n + m'$, for possibly different constants m' everytime. This process can be continued to any number of steps, in order to obtain a p-adic root (*non*-unique) up to *any* finite precision. □

We give the following straightforward property, based on lifting, as described in Section 1.3. It will be useful later in proving the correctness of Algorithm 1.

Lemma 21 (Val-multiplicity ≥ 1)**.** *Given the lifting as defined in Algorithm 1, the val-multiplicity is at least* 1 *in each step; i.e.,* $p | f_j(\mathbf{y}_\ell + p\mathbf{x}) \bmod \hat{\mathtt{C}}$*, where* $\hat{\mathtt{C}}$ *is the* p*-adic lift of an irreducible component of the ideal* $\mathtt{I}$*, as described in the algorithm.*

Proof. Let us consider the multivariate Taylor's expansion (Definition 15) of the j-th polynomial $f_j(\mathbf{y}_\ell + p\mathbf{x})$ given by

$$f_j(\mathbf{y}_\ell + p\mathbf{x}) \;=\; f_j(\mathbf{y}_\ell) + p \sum_{i \in [n]} \partial_{x_i} f_j(\mathbf{y}_\ell) \cdot x_i + \dots \;, \tag{5}$$

where the terms of order-$|\mathbf{i}|$ (partial-derivative) are divisible by $p^{|\mathbf{i}|}$.

As we traverse along the depth of the tree, the polynomial $f_j(\mathbf{y}_\ell) \bmod p$ will be added to $\mathtt{I}$ (Step 4 of Algorithm 1). In Step 5, we consider the absolutely irreducible components of this ideal projected down to $\widehat{\mathbb{G}}/\langle p \rangle = \mathbb{F}_q$, and loop over them from Step 7 of Algorithm 1. These absolutely irreducible components are such that they are first the factors of $\mathtt{I}$, after which we lift them to $\widehat{\mathbb{G}}$. Thus, for any component $\hat{\mathtt{C}}$ of Step 9, we have $\mathtt{I} = \hat{\mathtt{I}} + \langle p \rangle \subseteq \hat{\mathtt{C}} + \langle p \rangle = \mathtt{C}$.

Now, when we add $f_j(\mathbf{y}_\ell) \bmod p$ to $\mathtt{I}$, while introducing a new set of virtual roots $\mathbf{y}_\ell$, in Step 4 of Algorithm 1, then Equation 5 modulo $\mathtt{I}$ is divisible by p. Therefore, from the previous paragraph, we get that Equation 5 modulo $\hat{\mathtt{C}}$ is also divisible by p, implying that the val-multiplicity is ≥ 1, and after division by p, the polynomial will still have coefficients in $\hat{\mathbb{G}}$. □

A.4 Commutative algebra preliminaries

Lemma 22 (Singular roots [HW99, Lem.2.1])**.** *If a polynomial* $h(\mathbf{x})$ *is irreducible over* $\mathbb{F}_q$*, but reducible over its algebraic closure* $\overline{\mathbb{F}}_q$*, then for any root* $\mathbf{a} \in \mathbb{F}_q^n$ *of* h*, we have*

$$h(\mathbf{a}) = h_{x_j}(\mathbf{a}) = 0$$

over $\mathbb{F}_q$*, for any first-order partial-derivative* h_{x_j} *of* h*.*

The following lemma gives an estimate on the number of roots of an *absolutely* irreducible polynomial, which has been used in [HW99] to find a root of a system of polynomial equations over $\mathbb{F}_q$. This is the reason why absolute irreducibility is crucial in this paper.

Theorem 23 (Number of roots [Sch74])**.** *An absolutely irreducible polynomial* $f(\mathbf{x})$ *(*d*-degree* n*-variate) has number of roots in the range,* $q^{n-1} \pm ((d-1)(d-2)q^{n-1.5} + 6d^2q^{n-2})$*, over a large finite field* $\mathbb{F}_q$ *(namely,* $q > \omega(n^3d^5)$*).*

Gröbner basis. We require some concepts of Gröbner basis in our algorithm to find 'special' lifts to $\widehat{\mathbb{G}}$ (in Lemma 4). Modulo multivariate polynomial ideals, the remainder on division is not always unique. Thus, we modify the ideal by adding some more generators, depending on a given *ordering* of variables, such that the remainder modulo the ideal is unique.

For a given ideal $\mathtt{I}$, the *S-polynomial* of two polynomials g_1, g_2 in $\mathtt{I}$ is defined as

$$S(g_1, g_2) = \frac{\mathrm{lcm}(\mathrm{LM}(g_1), \mathrm{LM}(g_2))}{\mathrm{LT}(g_1)} \cdot g_1 \;-\; \frac{\mathrm{lcm}(\mathrm{LM}(g_1), \mathrm{LM}(g_2))}{\mathrm{LT}(g_2)} \cdot g_2 \;, \tag{6}$$

where LM denotes the leading monomial and LT denotes the leading term.

Buchberger [Buc65] gave the famous algorithm to compute the (reduced) Gröbner basis; by considering every pair of current generators of the ideal and iteratively adding their S-polynomials; until the S-polynomials are zero. More properties of Gröbner basis, and their complexity, can be found in [CLO13].

Decomposition Algorithm [HW99]. We will heavily utilise an algorithm (and related theorems) due to [HW99, Section 3] which extracts all the irreducible components of an algebraic set described by a given polynomial system.

The algorithm takes as input a set of multivariate polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ each of total degree bounded by d. The algorithm outputs each irreducible component of the algebraic set $\mathbf{V}_{\mathbb{F}}(\langle f_1, \ldots, f_m \rangle)$ in the form of a birationally equivalent hypersurface. It is a classical result in algebraic geometry that an irreducible algebraic set $\mathtt{W}$ of dimension r is birationally equivalent to a hypersurface $\mathtt{H} := \mathbf{V}_{\mathbb{F}}(\langle h \rangle)$ where h is an irreducible polynomial in $r+1$ variables. Precisely, for each $r \leq n$ and for each r-dimensional component $\mathtt{W}$ of $\mathbf{V}_{\mathbb{F}}(\langle f_1, \ldots, f_m \rangle)$ the algorithm outputs a polynomial $h \in \mathbb{F}[z_0, z_1, \ldots, z_r]$ such that $\mathtt{H} := \mathbf{V}_{\mathbb{F}}(\langle h \rangle)$ is birationally isomorphic to $\mathtt{W}$. The algorithm also returns the birational morphism $\psi_2 : \mathtt{W} \to \mathtt{H}$ and its inverse $\psi_1 : \mathtt{H} \to \mathtt{W}$.

Following Theroem 24 ([HW99, Theorem 2.6]) gives the complexity of the decomposition algorithm. The complexity bounds hold true over any field $\mathbb{F}$ where a randomized polynomial time algorithm exists for polynomial factorization e.g., finite fields and p-adic fields. The theorem also assumes that the size of the field $|\mathbb{F}| \geq d^{cn^2}$ for some constant c. These assumptions are without loss of generality in our case as we work over finite field and p-adic field and smaller field size will make our main theorems work by brute force within the given time complexity.

Theorem 24 (Simplified [HW99, Theorem 2.6])**.** *Given a system of polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ of total degree bounded by d. Then the decomposition algorithm [HW99, Section 3.3] will construct a birational hypersurface $\mathtt{H}$ for each irreducible component $\mathtt{W}$ of $\mathbf{V}_{\mathbb{F}}(\langle f_1, \ldots, f_m \rangle)$ in $m^{O(1)} d^{O(n^2)}$ field operations. Furthermore, the total degree of $\mathtt{H}$ as well as polynomials appearing in rational functions ψ_1 and ψ_2 is upper bounded by $d^{O(n)}$.*

B Missing proofs from Section 2: Details of SHN_{p^k}

Lemma 4 (Connection of points via hypersurfaces)**.** *Given an $\mathbb{F}_q$-irreducible ideal $\mathtt{C}$ (resp. its birational equivalent hypersurface $\mathtt{H}$), we can lift it to a* prime *$\widehat{\mathbb{G}}$-ideal $\hat{\mathtt{C}}$ (resp. its birational equivalent hypersurface $\hat{\mathtt{H}}$), such that their morphism diagram commutes (Figure 1).*

In particular, for a non-singular $\mathbb{F}_q$-root of $\mathtt{H}$ (thus a root of $\mathtt{C}$), we can find a $\widehat{\mathbb{G}}$-root of $\hat{\mathtt{H}}$; which gives a root of $\hat{\mathtt{C}}$. This sets up the 'connection' between roots of $\mathtt{C}$ and $\hat{\mathtt{C}}$.

Proof. We have a prime ideal $\mathtt{C}$ given by generators in $\mathbb{F}_q[y_1, \ldots, y_N]$. Let $r > 0$ be the dimension of the variety of $\mathtt{C}$. By one of the definitions of dimension, there is a subset $B =: \{\ell_1 < \ldots < \ell_r\}$ of *least* possible variables in $\mathbf{y}$, such that the function field $\mathbb{F}_q(\mathtt{C})$ is a *finite* extension over the transcendental field $\mathbb{F}_q(B)$. So, we consider its defining maximal ideal $B^{-1}\mathtt{C}$; and compute its reduced Gröbner basis (using Buchberger's algorithm [Buc65]); with the graded lexicographical ordering $y_1 < \ldots < y_N$ and variables B *localized*. Let $B' := \mathbf{y} \setminus B =: \{\ell_{r+1} < \ldots < \ell_N\}$ be the remaining variables.

Triangular form. The localization $B^{-1}\mathtt{C}$ is a zero-dimensional prime ideal (= *maximal* ideal). Thus, by [GTZ88, Prop.5.9], $B^{-1}\mathtt{C}$ has exactly $N - r$ generators, the i-th one ($r < i \leq N$) corresponding to a monic *minpoly* (over $\mathbb{F}_q(B)$) for the variable ℓ_i in B' (in particular, having the

leading-monomial an ℓ_i-power). Thus, the Gröbner basis $\mathrm{GB}(B^{-1}\mathtt{C})$ is in a special form, that we call the *triangular form* in B' over B (see [DMS19, Def.4]).

p-adic lift. Compute the reduced Gröbner basis $\mathrm{GB}(\mathtt{C})$ too, and divide each generator by its leading coefficient (in $\mathbb{F}_q^*$) to make the polynomials *monic*; store them in reduced form where the coefficients are in $\{0,\ldots,p-1\}$. Define the p-adic lift $\hat{\mathtt{C}}$, of $\mathtt{C}$ to $\hat{\mathbb{G}}$, by considering the trivial integral embedding of each generator of $\mathtt{C}$. By Gröbner basis properties and the special generators, this special lift $\hat{\mathtt{C}}$ is a prime $\hat{\mathbb{G}}$-ideal.

Doing the same thing to $\mathrm{GB}(B^{-1}\mathtt{C})$, it is easy to deduce: the $\hat{\mathbb{G}}$-ideal thus obtained, called $B^{-1}\hat{\mathtt{C}}$, is a maximal ideal with a triangular (& reduced) Gröbner basis.

$\mathbb{F}_q$-map. By construction, $\mathbb{F}_q(B)[B']/\mathtt{C}$ is a field, denoted $\mathtt{R}$, of finite degree over $\mathtt{R}_0 := \mathbb{F}_q(B)$. We can compute a hypersurface $\mathtt{H}$ that is *birationally equivalent* to the variety of $\mathtt{C}$ using Theorem 24 ([HW99, Thm.2.6]). A standard algebraic way to compute it, is to pick a *random* linear form ℓ_0; assume q to be large enough for random sampling. Let $h(Y)$ be the *minpoly* of the *primitive* element $\ell_0 \in \mathtt{R}$ over the subfield $\mathtt{R}_0$. We can store a representation of h in $\mathbb{F}_q[B][Y]$ such that it gives an $\mathtt{R}_0$-isomorphism ψ_1 between the fields, $\mathtt{R} = \mathtt{R}_0[B']/\mathtt{C} \cong \mathtt{S} := \mathtt{R}_0[Y]/\langle h\rangle$; mapping $\ell_0 \mapsto Y$, and other ℓ_i $(i > r)$ to its implied image.

p-adic map. Take any p-adic lift $\hat{h}$ of h; clearly $\hat{h} \in \widehat{\mathbb{G}}(B)[Y]$. By definition, $\hat{h}(\ell_0) \in \mathtt{C} = \hat{\mathtt{C}} + \langle p\rangle$. Since ℓ_0 is a separable $\mathbb{F}_q$-root of $\hat{h}$, we can Hensel lift it to a $\widehat{\mathbb{G}}$-root $\ell_0' \in \widehat{\mathbb{G}}(B)[B'] =: \mathtt{R}_0'[B']$ such that $\hat{h}(\ell_0') \in \hat{\mathtt{C}}$. So, mapping $Y \mapsto \ell_0'$ gives a $\mathtt{R}_0'$-homomorphism $\hat{\psi}_2 : \mathtt{S}' := \mathtt{R}_0'[Y]/\langle\hat{h}\rangle \longrightarrow \mathtt{R}' = \mathtt{R}_0'[B']/\hat{\mathtt{C}}$; which is a map between integral domains. Moreover, it remains a nontrivial homomorphism if we localize the base ring from $\mathbb{Z}_p$ to $\mathbb{Q}_p$; making it a map between *fields*. Thus, $\hat{\psi}_2$ is an injective $\mathtt{R}_0'$-homomorphism.

Now we know: all the four rings in Figure 1 are domains (& two are fields). So, in case $\hat{\psi}_2$ is not an isomorphism, it is injective and non-surjective. Let $v_0 \in \mathtt{R}'$ be an element that is out of the image, but we know that some lift $v_0 + pv_1$ is in the image of $\hat{\psi}_2$ (by traversing the commutative diagram). Similarly, we have that some lift $v_1 + pv_2$, of v_1, is in the image of $\hat{\psi}_2$. Combining these two, we know: $v_0 - p^2v_2$ is in the image of $\hat{\psi}_2$. Doing this *ad infinitum*, we get v_0 in the image of $\hat{\psi}_2$; contradicting its choice. We conclude: $\hat{\psi}_2$ is an isomorphism, with the inverse map being (say) $\hat{\psi}_1$.

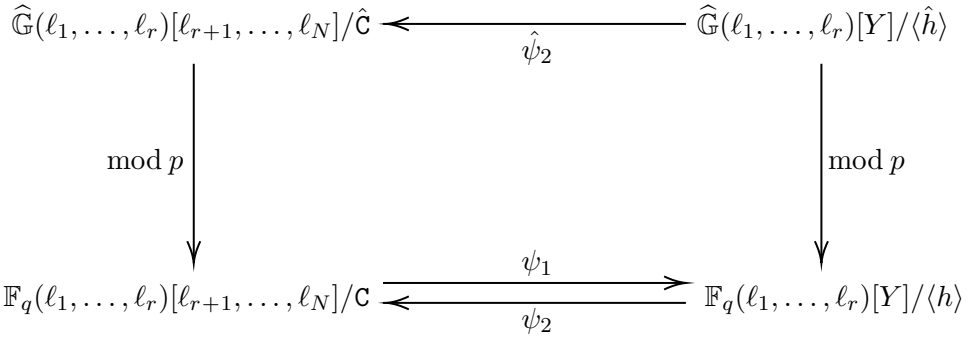

Figure 1: Commutative Diagram

In the above diagram let us start with a non-singular $\mathbb{F}_q$-root $\mathbf{a}$ of $\mathtt{H} := \mathbf{V}(h)$. With high probability, it will keep the relevant polynomials in $\ell_1,\ldots,\ell_r$ nonzero mod p; thus it would be consistent with the localization. It has 'pullback' via ψ_1, giving a root of $\mathtt{C}$. By the separability of the $\mathbb{F}_q$-root, $\mathbf{a}$ lifts to a root $\hat{\mathbf{a}}$ of $\hat{\mathtt{H}} := \mathbf{V}(\hat{h})$; from up there it has 'pullback' via $\hat{\psi}_1$, giving a $\widehat{\mathbb{G}}$-root

of $\hat{\mathtt{C}}$ too. This connects $\mathbf{V}(\mathtt{C})$ with $\mathbf{V}(\hat{\mathtt{C}})$. □

Lemma 6 (Size of tree)**.** *The total number of leaves $\mathcal{L}$ of the recursion-tree $\mathcal{T}$, described in Section 1.3, is at most $d^{(nk)^{O((nk)^2)}}$.*

Proof. We build tree $\mathcal{T}$ by first passing an ideal $\mathtt{I}_0$, in n variables, in Algorithm 2 with generators of degree at most $d_0 := d$. The set of absolutely irreducible ideals returned by Algorithm 2 forms the branches in the first level of $\mathcal{T}$. Each of these ideals (branches) at level-1, say $\mathtt{I}_1$, of degree d_1 (now in $2n$ variables) recurse in Algorithm 2, and produce more branches (ideals) at level-2. This process continues till $(k-1)$-th level.

The analysis of producing branches from an ideal at level $(\ell-1)$ to level ℓ is the same as that of [HW99]. This will allow us to use their estimates for number of branches and degree of new generators produced [HW99, Lem.2.7].

Similar to [HW99], we first decompose ideal $\mathtt{I}_{\ell-1}$ in $n\ell$ variables at Step 3 (Algorithm 2). However, we add h^* and e^* in the ideal at Steps 12-13 and then iterate. The idea of Step 12 is same as in [HW99] to capture the *singular* points of $\mathbf{V}(\mathtt{I})$ in separate absolutely irreducible ideals by adding h^* to the ideal. In [HW99], it was shown that the dimension of variety reduces when we add h^*.

When we add e^*, we make the 'free' variables $\ell_1, \ldots, \ell_r$ in the hypersurface (in Lemma 4) to satisfy an equation $e(\ell_1, \ldots, \ell_r) = 0$. Therefore, the transcendence degree reduces by 1, and it can reduce at most dimension-many times. So, complexity wise Step 13 is subsumed in Step 12 as degree of h^* and e^* have similar bound [HW99, Lem.2.7, Thm.2.6].

Applying analysis of [HW99] on ideal $\mathtt{I}_0$ at level-0, the number of branches (ideals) produced are $d_0^{n^{O(n)}}$ and the degree of generators at most $d_0^{n^{O(n)}} =: d_1$ at level-1. Each such branch (ideal) further produces (at level-2) $d_1^{(2n)^{O(2n)}}$ new branches with degree at most $d_1^{(2n)^{O(2n)}}$. By induction, the generator-set size, and degree, at level-nk (i.e., the leaves) is $\leq d^{(nk)^{O((nk)^2)}}$. □

Lemma 7 ($\dim > 0$ lift)**.** *Given an absolutely irreducible hypersurface $\mathtt{H}$ (resp. its lift $\hat{\mathtt{H}}$) over $\mathbb{F}_q$ of of positive dimension, its random $\mathbb{F}_q$-root is non-singular with high probability. Thus, we can lift a random root of $\mathtt{H}$ to $\widehat{\mathbb{G}}$-root of $\hat{\mathtt{H}}$.*

Proof. Let the hypersurface $\mathtt{H}$ be given by the polynomial $\langle h(Y)\rangle$ over $\mathbb{F}_q(\ell_1, \ldots, \ell_r)$ as before. Since it is absolutely irreducible, the variety $\mathbf{V}(h, h')$ has dimension one less than that of $\mathbf{V}(h)$, where $h' \neq 0$ is some first-order derivative of h. Therefore, the probability of a point being a non-singular root of $\mathtt{H}$, is around $(1 - q^{r-1}/q^r) = 1 - 1/q$ (by Theorem 23). Using a random non-singular root, we can lift it to modulo any p-power (by Proposition 3); thus, we get a $\widehat{\mathbb{G}}$-root. □

Lemma 8 (single-point lift)**.** *Given an $\mathbb{F}_q$-ideal $\mathtt{I}$ (resp. its lift $\hat{\mathtt{I}}$) that is radical and is a single point, we can uniquely lift it to $\widehat{\mathbb{G}}$-root of $\hat{\mathtt{I}}$.*

Proof. Since the ideal has a single point say $\mathbf{a}$; the ideal $\hat{\mathtt{I}}$ is just of the form $\langle \mathbf{y} - \hat{\mathbf{a}}\rangle$. So, we output $\hat{\mathbf{a}}$. □

Lemma 9 (Correctness of Algorithm 3)**.** *Given $\widehat{\mathbb{G}}$-ideal $\hat{\mathtt{I}}_{k-1}$ in a leaf of the tree $\mathcal{T}$, Algorithm 3 finds a generic common $\widehat{\mathbb{G}}$-root (if one exists) of the preceding ideals $\{\hat{\mathtt{I}}_\ell \,|\, \ell\}$.*

Proof. Using Lemma 4, we map an $\mathbb{F}_q$-root of ideal $\mathtt{I}$ to the 0-th precision digit of some (unknown) p-adic root of the ideal $\hat{\mathtt{I}}$. After this, if the root is random, we can lift to a $\widehat{\mathbb{G}}$-root using Hensel's lifting (Lemma 7). If the root comes from a single point ideal $\mathtt{I}$, then we use Lemma 8. Thus, Algorithm 3 correctly returns a $\widehat{\mathbb{G}}$-root of the given ideal $\hat{\mathtt{I}}$ (as long as, q is large enough for sampling). But what about the other ideals in the path in $\mathcal{T}$ leading to $\hat{\mathtt{I}}$?

We further need to show that the variety $\mathbf{V}(\hat{\mathtt{I}}_{\ell-1})$ extends to $\mathbf{V}(\hat{\mathtt{I}}_\ell)$ (where $\hat{\mathtt{I}}_{\ell-1}, \hat{\mathtt{I}}_\ell$ are the eventual definitions in recursion-tree $\mathcal{T}$). Consider a lift using Lemma 4; say in the $\mathbb{F}_q$-ideal $\mathtt{I}_{\ell-1}$ the variables $B_{\ell-1}$ are *localized* giving the triangular form (reduced Gröbner basis) in $\mathbb{F}_q(B_{\ell-1})[B'_{\ell-1}]$, where $B'_{\ell-1} := \cup_{i\le\ell-1}\mathbf{y}_i \setminus B_{\ell-1}$. Similarly, define B_ℓ and B'_ℓ, for I_ℓ. Recall the variable order, blockwise, $\mathbf{y}_{\ell-1} < \mathbf{y}_\ell$. By Algorithm 1 (Step 8), $B_{\ell-1} = B_\ell$; and $\mathtt{I}_{\ell-1} \subseteq \mathtt{I}_\ell$. Thus, the minpoly of the variables $B'_{\ell-1}$, over $\mathbb{F}_q(B_\ell)$, in $B_\ell^{-1}\mathtt{I}_\ell$; is the same as it was in $B_{\ell-1}^{-1}\mathtt{I}_{\ell-1}$. So, all the variables $\cup_{i\le\ell-1}\mathbf{y}_i$ have the same minpoly in the two Gröbner bases; and the triangular form is preserved in the new ideal $B_\ell^{-1}\mathtt{I}_\ell$.

Consequently, from Lemma 4 lifting, the generators of $B_{\ell-1}^{-1}\hat{\mathtt{I}}_{\ell-1}$ are contained inside those of $B_\ell^{-1}\hat{\mathtt{I}}_\ell$. Therefore, the variety of $\hat{\mathtt{I}}_{\ell-1}$ extends to that of $\hat{\mathtt{I}}_\ell$: For any generic root $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_\ell)$ of $\hat{\mathtt{I}}_\ell$, the projection $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{\ell-1})$ is a root of the predecessor ideal $\hat{\mathtt{I}}_{\ell-1}$.

With this induction step done, we can complete the proof for all $0 \le \ell \le k-1$. We use a monomial ordering, that is consistent with all steps, namely: $y_{0,1} < y_{0,2} < \cdots y_{0,n} < \cdots y_{k-1,1} < y_{k-1,2} < \cdots y_{k-1,n}$. Under localization of respective transcendence-basis, we keep the leading term of the generators of these ideals, to be $y_{i,j}$-powers; maintaining the triangular form of Gröbner basis. Thus, a generic root of the leaf $\hat{\mathtt{I}}_{k-1}$, is also a generic common root of the ideals $\{\hat{\mathtt{I}}_0, \ldots, \hat{\mathtt{I}}_{k-1}\}$ that led to the leaf $\hat{\mathtt{I}}_{k-1} \in \mathcal{T}$. □

Proposition 1 (Root in $\mathcal{L} \longrightarrow$ Root of $\mathcal{F}$)**.** *Given a root of a leaf in $\mathcal{L}$ (using $\mathcal{T}$ and Algorithm 3), we can find a common $\mathbb{G}$-root of the system $\mathcal{F}$ of polynomials f_j, for $j \in [m]$.*

Proof. We are given a prime ideal, say $\hat{\mathtt{I}}_{k-1} \in \mathcal{L}$, and the associated latest prime ideals $\mathfrak{I} := \{\hat{\mathtt{I}}_0, \ldots, \hat{\mathtt{I}}_{k-1}\}$ in the recursion-tree $\mathcal{T}$ of Algorithm 1. Let us assume that when the ℓ-th ideal $(\hat{\mathtt{I}}_{\ell-1})$ was defined the last (satisfying Step 8 of Algorithm 1), the j-th polynomial was $f_j^{(\ell)}(\mathbf{x}) \in \widehat{\mathbb{G}}[\mathbf{y}_0, \ldots \mathbf{y}_{\ell-1}][\mathbf{x}]$ (done at Step 10 of Algorithm 1).

From the p-adic root of $\mathfrak{I}$, say $(\mathbf{a}_0, \ldots, \mathbf{a}_{k-1}) \in \widehat{\mathbb{G}}^{nk}$ by Algorithm 3; we want to show that we can construct a common $\mathbb{G}$-root of $f_j(\mathbf{x})$'s, $j \in [m]$. We prove this by simply using the lifting-steps, one precision at a time, that designed the recursion-tree.

$$\begin{aligned} &p \mid f_j^{(0)}(\mathbf{y}_0) \bmod \hat{\mathtt{I}}_0\,, \\ &p \mid f_j^{(1)}(\mathbf{y}_0, \mathbf{y}_1) \bmod \hat{\mathtt{I}}_1\,, \\ &\quad\vdots \\ &p \mid f_j^{(k-1)}(\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{k-1}) \bmod \hat{\mathtt{I}}_{k-1}\,. \end{aligned} \tag{7}$$

Next, we can merge the divisibility properties of the key k lifting-steps (Algorithms 1 & 3), at the common p-adic point $(\mathbf{a}_0, \ldots, \mathbf{a}_{k-1})$ of Equation 7. This can be written as the following

cascading divisibilities:

$$\begin{aligned} p \,|\, f_j^{(0)}(\mathbf{a}_0) \;&\to p^2 \,|\, f_j^{(0)}(\mathbf{a}_0 + p\mathbf{a}_1) \\ &\to p^3 \,|\, f_j^{(0)}(\mathbf{a}_0 + p\mathbf{a}_1 + p^2\mathbf{a}_2) \\ &\to \ldots \to\; p^k \,|\, f_j^{(0)}(\mathbf{a}_0 + p\mathbf{a}_1 + \ldots + p^{k-1}\mathbf{a}_{k-1})\,, \end{aligned} \tag{8}$$

which provides the required precision p^k; thus giving the $\mathbb{G}$-root $(\mathbf{a}_0 + \ldots + p^{k-1}\mathbf{a}_{k-1})$ of f_j. This finishes proof. □

Proposition 2 (Root of $\mathcal{F} \longrightarrow$ Root in $\mathcal{L}$)**.** *If the system of polynomials, as described before, has a root in $\mathbb{G}$, then Algorithm 3 outputs a root for some leaf ideal $\hat{\mathtt{I}}_{k-1}$ in $\mathcal{L}$.*

Proof. Let us assume that the system of polynomials has a $\mathbb{G}$-root, given by $\mathbf{a} := (\mathbf{a}_0 + p\mathbf{a}_1 + \cdots + p^{k-1}\mathbf{a}_{k-1})$, with $\mathbf{a}_i$'s effectively 'in' $\mathbb{F}_q$. We use a technique, similar to that in the proof of Proposition 1, to inductively show that a root up to precision ℓ digits gives a p-adic root of the ideal grown for ℓ-steps (possibly with backtrackings). We use the same notation as of Proposition 1 for $f_j^{(\ell)}, \hat{\mathtt{I}}_\ell$.

For the *base case* of induction, let us consider the root $\mathbf{a}_0$ of $f_1(\mathbf{y}_0), \ldots, f_m(\mathbf{y}_0)$ over $\mathbb{F}_q$. Now, each of these equations were added to the ideal $\hat{\mathtt{I}}_0$, on which we performed the decomposition algorithm to find components $\hat{\mathtt{C}}$'s. Since, $\mathbf{a}_0$ is a root of $\hat{\mathtt{I}}_0 + \langle p \rangle$, it must also be a root of some $\hat{\mathtt{C}} + \langle p \rangle$; let us fix this ideal $\hat{\mathtt{C}}$. Now, by definition (Algorithm 2 & Lemma 4), $\hat{\mathtt{C}}$ is prime; and absolutely irreducible mod$p$. If $\mathbf{a}_0$ is a non-singular $\mathbb{F}_q$-root of $\hat{\mathtt{H}}$ (= hypersurface birationally equivalent to $\hat{\mathtt{C}}$), then it has a lift, say $\hat{\mathbf{a}}_0$, using Hensel's lifting (Proposition 3). Thus, we get a corresponding $\widehat{\mathbb{G}}$-root of $\hat{\mathtt{C}} \in \mathcal{T}$. On the other hand, if $\mathbf{a}_0$ is a singular root, or a root whose preimage does not exist in the hypersurface, then it will be present in some ideal of *lesser* dimension (eg. in another branch of recursion-tree $\mathcal{T}$). So, Algorithm 2 will locate $\mathbf{a}_0$ as a non-singular root of some other absolutely irreducible ideal of dimension ≥ 0. Thus, we always get a corresponding $\widehat{\mathbb{G}}$-root of some ideal, say $\hat{\mathtt{D}}$, in $\mathcal{T}$.

Now, for our *induction hypothesis*, assume that the root of the system modulo p^ℓ, $(\mathbf{a}_0 + \cdots + p^{\ell-1}\mathbf{a}_{\ell-1})$, gives a p-adic root, $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{\ell-1})$, of some $\hat{\mathtt{C}}$ which is an absolutely irreducible component of $\hat{\mathtt{I}}_{\ell-1}$ such that

$$\sum_{i=0}^{t-1} p^i \mathbf{a}_i \;\equiv\; \sum_{i=0}^{t-1} p^i \hat{\mathbf{a}}_i \bmod p^t, \text{ for } t \leq \ell\,. \tag{9}$$

Let us consider the *induction step*. Consider $f_j^{(\ell)}(\mathbf{y}_\ell) = f_j(\mathbf{y}_0 + \ldots + p^{\ell-1}\mathbf{y}_{\ell-1} + p^\ell \mathbf{y}_\ell) \bmod \hat{\mathtt{C}}$, where $\hat{\mathtt{C}}$ is the component where the p-adic root $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{\ell-1})$ can be found. After substituting the first ℓ variables by $(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{\ell-1})$, $f_j^{(\ell)}(\mathbf{y}_\ell)$ has a root, say $\hat{\mathbf{a}}_\ell$, modulo $p^{\ell+1}$; simply because—$f_j(\mathbf{a}_0 + \ldots + p^{\ell-1}\mathbf{a}_{\ell-1} + p^\ell \mathbf{x})$ has the root $\mathbf{a}_\ell$ modulo $p^{\ell+1}$, and by the induction hypothesis (esp. Equation 9). Like we did in the base case, we can consider two broad cases: $\hat{\mathbf{a}}_\ell$ is a non-singular $\mathbb{F}_q$-root, or it is a singular root (or a root whose preimage does not exist in the birationally equivalent hypersurface of Lemma 4). In the first case, we find a suitably lifted root in $\hat{\mathtt{C}} \in \mathcal{T}$ itself. While in the second case, we find a suitably lifted root in some lower-dimensional $\hat{\mathtt{D}} \in \mathcal{T}$ (though with the same $\{\hat{\mathtt{I}}_0, \ldots, \hat{\mathtt{I}}_{\ell-1}\}$). Thus, in all cases we ensure that

$$\hat{\mathbf{a}}_0 + \ldots + p^{\ell-1}\hat{\mathbf{a}}_{\ell-1} + p^\ell \hat{\mathbf{a}}_\ell \;\equiv\; (\mathbf{a}_0 + \ldots + p^\ell \mathbf{a}_\ell) \bmod p^{\ell+1},$$

for the lifted $\hat{\mathbb{G}}$-root of some ideal $\hat{\mathtt{I}}_\ell$ (which gets defined in Step 9 of Algorithm 1, possibly after many backtrackings); finishing the induction step.

Thus, with $\ell = k-1$, we deduce: $\mathbf{a}$ is represented as a $\widehat{\mathbb{G}}$-root of some ideal $\hat{\mathtt{I}}_{k-1}$ in $\mathcal{L}$. $\square$