Leveled Fully Homomorphic Encryption Schemes with Hensel Codes

David W. H. A. da Silva[1], Luke Harmon[1], Gaetan Delavignette[2], and Carlos Araujo[1]

[1] Algemetric, Colorado Springs, CO, 80919, USA
{dsilva,lharmon,caraujo}@algemetric.com
https://www.algemetric.com/

[2] University of Colorado at Colorado Springs, Colorado Springs, CO, 80918, USA
gdelavig@uccs.edu https://www.uccs.edu/

Abstract. We propose the use of Hensel codes (a mathematical tool lifted from the theory of p-adic numbers) as an alternative way to construct fully homomorphic encryption (FHE) schemes that rely on the hardness of some instance of the approximate common divisor (AGCD) problem. We provide a self-contained introduction to Hensel codes which covers all the properties of interest for this work. Two constructions are presented: a private-key leveled FHE scheme and a public-key leveled FHE scheme. The public-key scheme is obtained via minor modifications to the private-key scheme in which we explore asymmetric properties of Hensel codes. The efficiency and security (under an AGCD variant) of the public-key scheme are discussed in detail. Our constructions take messages from large specialized subsets of the rational numbers that admit fractional numerical inputs and associated computations for virtually any real-world application. Further, our results can be seen as a natural unification of error-free computation (computation free of rounding errors over rational numbers) and homomorphic encryption. Experimental results indicate the scheme is practical for a large variety of applications.

Keywords: Rational numbers · Fully homomorphic encryption · Hensel codes · Public-key encryption · Extended Euclidean algorithm.

1 Introduction

Homomorphic encryption (HE) is a type of encryption that enables meaningful and general computation over encrypted data. This notion, originally referred to as *privacy homomorphisms*, was introduced in 1978 [56]. Although every single instance of practical homomorphic computation can be interesting in itself, it is clear that the ultimate goal of HE was to enable computation of *any circuit*. Several constructions provided partial solutions [17, 25, 24, 53, 4] but it was not until 2009 that Craig Gentry proposed the first fully homomorphic encryption (FHE) scheme [22, 23]. Gentry's strategy consisted in first realizing a somewhat homomorphic encryption (SHE) scheme that enables the (homomorphic) evaluation of low-degree multivariate polynomials. Ciphertexts are embodied with noise, which grows slightly over addition and tremendously over multiplication, which compromises the limits of low-degree polynomials. To solve this problem, Gentry introduced a *bootstrapping* mechanism with which one can transform a SHE scheme that is able to homomorphically evaluate its own decryption function into a *leveled* FHE scheme, that is, an encryption scheme that is able to evaluate any circuit up to a predefined depth. The bootstrapping technique produces a “fresh” ciphertext: a ciphertext with an amount of noise equivalent to what it was prior to any homomorphic operation.

As remarked by Brakerski et al. [5], not only few SHE schemes are able to evaluate their own decryption function but also FHE schemes that follow Gentry’s blueprint suffer from poor performance. To put things into perspective, the complexity of performing bootstrapping is at least the complexity of decryption multiplied by the bit-length of the individual ciphertexts that are used to encrypt the bits of the secret key. In the context of Gentry’s blueprint this is necessary since the SWHE evaluates the decryption function using an encrypted secret key and each bit of the secret key is then replaced by a very large ciphertext that encrypts that bit [5]. To address this problem, Brakerski, Gentry, and Vaikuntanathan introduced two schemes [5] (known as BGV) which are conceived via an entirely new approach, with much better performance than Gentry’s original blueprint. This new approach consists in skipping the SWHE step and directly constructing leveled FHE schemes with the possibility of using bootstrapping as an optimization. The BGV scheme, as the vast majority of FHE schemes, is latticed-based and its security is based on some version of the learning with errors (LWE) assumption.

Dijk, Gentry, Halevi, and Vaikuntanathan, when introducing the scheme known as DGHV propose an interesting question: “What is the simplest encryption scheme for which one can hope to achieve security?”. Naturally, the simple will not always be secure so the reconciliation of simplicity and security is undoubtedly a much desired and sometimes hard-to-achieve property. Compared to any lattice-based FHE scheme, DGHV is significantly simpler: very small description with basic modular arithmetic. Similarly to Gentry-like constructions, it encrypts individual bits. Unlike lattice-based schemes (which work with vectors and matrices), it operates over the integers. DGHV’s security is based on both the single-source-shortest-paths (SSSP) and the approximate greatest common divisor (AGCD) assumption introduced by Howgrave-Graham in [34]. Several other contributions were able to improve DGHV’s efficiency [15, 16, 9, 8, 14].

Could a simpler FHE scheme be as secure as the lattice-based ones? A remarkable result by Cheon and Stehlé [10] introduces a reduction from LWE to AGCD which is demonstrated by constructing a FHE scheme with security based on the AGCD assumption by deriving the AGCD parameters from the LWE parameters. Among the similarities between DGHV and the scheme proposed by Cheon and Stehlé, we remark two facts: 1) they both encrypt bits and 2) they derive a public-key encryption scheme by first describing a private-key encryption scheme and then converted into its public-key counterpart by applying the method introduced by Rothblum [58] which is based on the fact that any additively homomorphic private-key encryption scheme that is *compact* can be converted into a public-key encryption scheme. (Informally, a FHE scheme is compact if the size of ciphertexts output by homomorphic evaluations is independent of the number of ciphertexts and/or operations from which it was created.) The combination of these two facts has, at least, the following implication: if γ is the bit length of ciphertexts generated by a FHE scheme with the aforementioned characteristics, for each n-bit message, their corresponding ciphertexts have length $n\gamma$. Since in that kind of encryption scheme the public key is a τ-tuple of ciphertexts encrypting n-bit messages, the length of the public-key is $\tau n\gamma$ bits.

1.1 Homomorphic Rational Arithmetic

The need for performing homomorphic operations with rational numbers has been recently investigated. This issue is usually addressed by adding an encoding scheme to the homomorphic encryption scheme so rational numbers can be encoded to, typically, polynomials over some ring. A clever solution was proposed in [6] where a technique from proposed by Hoffstein and Silverman [32] is combined with the Fan-Vercauteren homomorphic encryption scheme [18] so a new encryption scheme is derived where rational numbers can be encoded and then used as input. Another interesting solution was proposed in [12] where rational numbers are thought as continued fractions and then represented as a sequence of integers. It is not surprising, due to its simplicity, that some form of modular arithmetic is used to encode rational numbers for carrying computation over the integers [44]. Our contribution, at the very least, is distinct in the fact that the encoding of rational numbers into integers is the encryption function itself. Thus, we do not follow the blueprint of using a scheme for encoding rational numbers and another scheme for encrypting and evaluating homomorphic operations. Instead, Hensel codes are employed for both encoding and encryption. Another advantage of our constructions is that we show how to probabilistically encode rational numbers in a structure-preserving way so other homomorphic encryption schemes can use our encoding for performing rational arithmetic.

1.2 Our Contributions

Would it be possible to describe a leveled FHE scheme that conveniently evaluates ciphertexts over the integers and at the same time has a better ciphertext expansion? Would it be possible to work with a public key with length smaller than the length of corresponding ciphertexts? Furthermore, what if we wanted to further expand the message space from bits to not only large integers but also large (positive and negative) rational numbers? Properly expanding the message space of a FHE scheme to a more comprehensive set that includes rational numbers immediately enables the application of homomorphic encryption in scenarios that involve fractional data such as those associated with statistics, finance, machine learning, digital signal processing, among others, without any further need of data formatting. Besides the obvious benefits of such features, not having to format data at the bit level (for accommodating custom message spaces) represents, at the very minimum, less overhead. We believe that a leveled FHE scheme with these desired characteristics requires an approach that is distinct from those employed up to today.

We propose a new approach to construct a leveled FHE scheme that takes messages over a specialized set of rational numbers that can be sufficiently large to contain all rational numbers of interest for any real-world practical application. Our technique allows us to describe a private-key encryption scheme and turn it into a public-key encryption scheme where its public-key has length smaller than the ciphertext it generates. Moreover, both private-key and public-key leveled FHE schemes produce ciphertexts with the same length. We show that the security of our schemes can be clearly mapped to the AGCD assumption while we also introduce the notion of a new hardness assumption, which makes the security analysis clearer and more objective. We showcase a mathematical tool mostly used outside the context of cryptography, which enables our contributions, and we propose its use and further investigation in cryptography.

1.3 Hensel Codes

Between the end of the 19th and the beginning of the 20th centuries, Kurt Hensel introduced the p-adic numbers theory [31]. One of Hensel's main motivations was to relate the ring integers $\mathbb{Z}$ to the field of rationals $\mathbb{Q}$. For our purposes, it suffices to provide a brief discussion of the fundamental idea. If p is prime, any positive integer x can be represented uniquely as an expansion of the form $x = a_0 + a_1 p + a_2 p^2 + ... + a_n p^n$, where a_i is an integer with $0 \leq a_i < p$. In fact, one can similarly expand any rational number x/y by allowing negative powers of p. Such expansions are called p-adic numbers [26]. In the p-adic number system, the elements of $\mathbb{Q}$ are represented as infinite expansions $\alpha = \sum_{-\infty}^{\infty} a_i p^i$. Applications of p-adic numbers are varied, and include dynamical systems, theoretical physics, algebraic geometry, non-Archemdian analysis [35], differential calculus [42], topology [38, 37], and analytic functions [50, 57].

Between the 1970s and 1980s, Krishnamurthy, Rao, Subramanian [41], Alparslan [1], Hehner and Horspool [30] proposed the use of truncation of p-adic expansions to replace arithmetic operation on rational numbers by the corresponding operations on integers that represent those rational numbers. They named these special integers as *Hensel codes* and they established the foundation of the theory of Hensel codes as a solution to the problem of *error-free computation* [20, 27, 29, 51, 41, 55], that is, the computation over approximations of real numbers in such a way that rounding errors do not occur. This property is particularly relevant when working with ill-conditioned problems and numerically unstable algorithms.

Converting rational numbers into Hensel codes is rather trivial, however, the inverse mapping of Hensel codes was for many years an open problem [54] until Gregory identified the required boundaries in absolute value to the numerators and denominators of rational numbers so a Hensel code could be uniquely inverted [27–29]. Having these boundaries well-defined allowed Miola [48] to propose an efficient algebraic solution for inverting Hensel codes by applying a modified version of the Extended Euclidean Algorithm. Over the years, the theory of Hensel codes expanded to address a variety of areas benefited by error-free computation such as computation of Gröbner bases [19], overflow detection [47], matrix inversion [51], fast integer division [40], parallel computation [49], solving linear systems of equations [36], polynomial matrix computations [39], to cite a few.

Hensel codes can be represented and computed in many forms, from the "dotted" representation [54] to matrices of rational polynomials [39]. In this work we focus on the integer representation of Hensel codes using just the first coefficient of a conventional truncated p-adic expansion. We show that Hensel codes can be p-adic and g-adic (defined via single or multiple primes, respectively) and we expand the original special set of rational numbers to represent as Hensel codes in order to achieve a bijection between those special rational numbers and a finite set of integers reduced modulo a prime or a prime composite.

1.4 General Intuition

We were initially interested in Hensel codes solely for purpose of establishing a bijection between a subset of the rationals and a finite set of integers so we could construct a level FHE scheme with a more comprehensive message space. In the past, the use of Hensel codes for error-free computation was shown to be a more efficient solution in comparison to known alternatives [54, 28]. Could Hensel codes still provide advantages for error-free computation nowadays? In 2019, Barillas proposed an efficient machine learning classification approach based on Restricted Boltzmann Machines using Hensel codes. Barillas worked with limited hardware resources since the goal was to provide a solution suitable for embedded devices and classification problems over data containing a small to medium amount of features. Barillas' results over the MNIST dataset outperformed the current state-of-the-art of exact machine learning computations by a factor of 42 in terms of performance, and a factor of 62 in terms of energy efficiency [2]. So we were encouraged to proceed.

However, we identified an additional opportunity that is enabled by two facts: 1) The mapping we use to establish a connection between a special set of rational numbers and their corresponding Hensel codes has well-defined boundaries which are unique per prime or group of primes. Failure in observing these boundaries will lead to correctness violation. 2) The knowledge of the primes involved in the computation of Hensel codes is required for computing back their corresponding rational numbers. We then created a cryptosystem based on the hardness of inverting Hensel codes without the knowledge of the primes involved in that computation. We do it in such a way that trivial attempts will always violate the boundaries for correctness. Once the primes are unknown, so are the boundaries. This allows us to provide a new asymmetric encryption algorithm based on Hensel codes.

2 Hensel Codes

We now provide a sufficient and self-contained review of the theory of Hensel codes. While we omitted some portions of that theory (for lack of a direct connection with our contributions), we believe that more of the theory can not only be applied in future developments of our research. We hope that this work can motivate further study of Hensel codes as underlying tools for building cryptographic tools.

2.1 Hensel Codes and the Extended Euclidean Algorithm

It was shown by R.T. Gregory [28] that there is a one-to-one mapping from the so-called order-N Farey fractions

$$\mathbb{F}_N := \left\{x/y \mid |x| \leq N,\, 0 < |y| \leq N\right\},\ N = \left\lfloor\sqrt{(p-1)/2}\right\rfloor$$

to the finite field $\mathbb{Z}_p$, given via the mapping $x/y \mapsto xy^{-1} \pmod p$. The major drawback of the order-N Farey fractions is that they only correspond to a *subset* of $\mathbb{Z}_p$. We will use a modification of the extended Euclidean algorithm (EEA) to enlarge $\mathbb{F}_N$ to a set whose elements are in bijective correspondence with the elements of $\mathbb{Z}_p$. In particular, we construct a factor ring (isomorphic to the finite field of order p) from a subring of the rationals $\mathbb{Q}$ and then use the to-be-defined modification of the EEA to select one representative fraction from each coset of the factor ring. To this end, fix an odd prime p, and recall that the set $\left\{a/b \mid \gcd(p,b)=1\right\}$ can be realized as the localization of the integers $\mathbb{Z}$ at the prime ideal (p). We will denote this ring by $\mathbb{Z}_{(p)}$. Since $\gcd(p,b)=1$ guarantees that b^{-1} exists in $\mathbb{Z}_p$, we can define the map $H_p : \mathbb{Z}_{(p)} \to \mathbb{Z}_p$ by $a/b \mapsto ab^{-1} \pmod p$. It is easy to verify that this map is a surjective ring homomorphism. Consequently, we obtain an isomorphism $\mathbb{Z}_{(p)}/\ker(H_p) \cong \mathbb{Z}_p$. There are many ways to select representatives from the cosets of $\mathbb{Z}_{(p)}/\ker(H_p)$, but we will make our selection to guarantee that the set of representatives contains $\mathbb{F}_N$.

Recall that the Extended Euclidean Algorithm (EEA) calculates the greatest common divisor of two integers x_0, x_1 along with the associated Bézout coefficients. The computation generates the tuples $(x_2,\ldots,x_n)$, $(y_2,\ldots,y_n)$, $(z_2,\ldots,z_n)$, and $q_i = \lfloor x_{i-1}/x_i \rfloor$ such that:

$$\begin{aligned} x_{i+1} &= x_{i-1} - q_i x_i, && \text{where } x_0,\, x_1 \text{ are the input,} \\ y_{i+1} &= y_{i-1} - q_i y_i, && \text{with } y_0 = 1,\, y_1 = 0, \\ z_{i+1} &= z_{i-1} - q_i z_i, && \text{with } z_0 = 0,\, z_1 = 1. \end{aligned}$$

Moreover, for each $i \leq n$, we have $y_i x_1 + z_i x_0 = x_i$. The computation stops with $x_n = 0$, at which point $x_{n-1} = \gcd(x_0, x_1)$. We define a modified version of this algorithm, as follows:

Definition 1 (Modified Extended Euclidean Algorithm). *Let g be a product of distinct odd primes, $h \in \mathbb{Z}$, and $N = \left\lfloor\sqrt{(g-1)/2}\right\rfloor$. Run* EEA *with*

$x_0 = g$ and $x_1 = h$. Once $|x_i| \leq N$, output $(x, y) = \big((-1)^{i+1}x_i,\ (-1)^{i+1}y_i\big)$. We write this as $\mathsf{MEEA}(g, h) = (x, y)$. Observe that there is an integer z (namely, $(-1)^{i+1}z_i$) such that $yh + zg = x$.

Lemma 1. *Let g be a product of distinct, odd primes, $N = \left\lfloor\sqrt{(g-1)/2}\right\rfloor$, and $h, h' \in \mathbb{Z}_g$. The following hold:*

(i) If $\mathsf{MEEA}(g, h) = (x, y)$, then $|x| \leq N$ and $|y| \leq 2N$.
(ii) Let p be prime, $\mathsf{MEEA}(p, h) = (x, y)$, and $\mathsf{MEEA}(p, \alpha) = (x', y')$. $\alpha = h \pmod p$ if and only if $x = x'$ and $y' = y \pmod p$
(iii) $\mathsf{MEEA}(g, h) = (0, \cdot)$ if and only if $\gcd(g, h) > N$ or $h = 0$.

Proof. (i) Suppose $\mathsf{MEEA}(g, h) = \big((-1)^{i+1}x_i, (-1)^{i+1}y_i\big)$. That $|x| \leq N$ is immediate from the stopping condition in MEEA. The outputs of the EEA satisfy [59]

$$|y_k| \leq \frac{x_0}{x_{k-1}}, \text{ for all } k.$$

By definition, $x_{i-1} > N$. Whence, for $N' = \sqrt{(g-1)/2}$,

$$|y_i| \leq \frac{g}{x_{i-1}} < \frac{g}{N'} < \frac{2(N')^2 + 1}{N'} = 2N' + \frac{1}{N'}$$

It follows that $|y_i| \leq \left\lfloor 2N' + 1/N' \right\rfloor = 2N$, proving 1.
(ii) By hypothesis, there is an integer k such that $\alpha = h + kg$. Suppose that $\alpha \neq h$ (i.e., at least one of h,k is nonzero). Apply the EEA in two cases: (1) $x_0 = p$, $x_1 = h$, and (2) $x_0 = p$, $x_1 = h + kp$. After three iterations of (2), one observes that the values of x_i match those obtained after one iteration of (1). Moreover, MEEA applied to (1) and (2), respectively, will not terminate before the values of x_i match. This proves $x = x'$. The above, in conjunction with $yh + zp = x$ and $y'\alpha + z'p = x'$, yields $yh = y'\alpha$. Then $yh - y'h = y'kp$, and so $y'h = yh \pmod p$. Since $\gcd(p, h) = 1$, $y' = y \pmod p$. The converse follows easily.
(iii) Suppose $\gcd(g, h) > N$. Recall that the EEA with $x_0 = g$ and $x_1 = h$ terminates when $x_n = 0$, at which point $x_{n-1} = \gcd(g, h)$. Item 3 then follows from the stopping condition in MEEA. Conversely, if $\mathsf{MEEA}(g, h) = (0, \cdot)$, then $\gcd(g, h) > N$. For if not, then $\mathsf{MEEA}(g, h) = \big((-1)^{i+1}x_i, \cdot\big)$, where $i \leq n - 1$, a contradiction. □

Definition 2 (Order-(N, p) Farey Fractions). *Let p be an odd prime and $N = \left\lfloor\sqrt{(p-1)/2}\right\rfloor$. We define the set of order-(N, p) Farey fractions as*

$$\mathbb{F}_{N,p} := \left\{\frac{x}{y} : \exists h \in \mathbb{Z}_p \text{ s.t. } \mathsf{MEEA}(p, h) = (x, y)\right\}.$$

Throughout the paper, we will consider $\mathbb{F}_{N,p}$ with the familiar addition and multiplication on $\mathbb{Q}$. Note that $\mathbb{F}_{N,p}$ is not closed under these operations. The following lemma collects some important facts about $\mathbb{F}_{N,p}$.

Proposition 1. *Let p be an odd prime and $N = \left\lfloor \sqrt{(p-1)/2} \right\rfloor$.*
(i) $\mathbb{F}_N \subseteq \mathbb{F}_{N,p}$.
(ii) If $x/y \in \mathbb{F}_{N,p}$, then $|x| \leq N$ and $|y| \leq 2N$.
(iii) The elements of $\mathbb{F}_{N,p}$ are in lowest terms.
(iv) Distinct elements of $\mathbb{F}_{N,p}$ lie in distinct cosets of $\mathbb{Z}_{(p)}/\ker(H_p)$.
(v) $H_p : \mathbb{F}_{N,p} \to \mathbb{Z}_p$ is a bijection.

Proof. (i) Let $x/y \in \mathbb{F}_N$ such that $h = xy^{-1} \pmod p$, and say $\mathsf{MEEA}(p,h) = (x_i, y_i)$. By construction, $x_i \leq N$ and $x_j > N$ for all $j < i$. As shown by Kornerup [28], implementing EEA with $x_0 = p$ and $x_1 = h$ will yield $x_k/y_k = x/y$ for some k. Now, suppose that $|y_i| > N$. One easily verifies inductively $|y_j| \leq |y_{j+1}|$ and $x_j > x_{j+1}$ for all j. Whence for all i, either $x_i > N$ or $|y_i| > N$, contradicting $x_k/y_k \in \mathbb{F}_N$. Thus $|y_i| \leq N$, and $x_i/y_i \in \mathbb{F}_N$. Finally, since $x_i y_i^{-1} \pmod p = xy^{-1} \pmod p = h$, and representations of elements of $\mathbb{F}_N$ in $\mathbb{Z}_p$ are unique, we conclude that $x_i/y_i = x/y$. This shows that $x/y \in \mathbb{F}_{N,p}$.
(ii) Use Lemma 1 with $g = p$.
(iii) $\mathsf{MEEA}(p, 0) = (0, \cdot)$, so let $h \in \mathbb{Z}_p$ be nonzero and $\mathsf{MEEA}(p,h) = (x,y)$. By definition, there is an integer z (output by EEA) such that $x = yh + zp$. Further, properties of greatest common divisors yield

$$\gcd(x, y) = \gcd(yh + zp, y) = \gcd(zp, y).$$

By (ii), $0 < |y| < p$. We deduce from Lemma 1(iii), that $0 < |z| < p$. Consequently,

$$\gcd(zp, y) = \gcd(z, y) \cdot \gcd(p, y) = \gcd(z, y).$$

Now, by [59, Theorem 4.3], $\gcd(z, y) = 1$, which proves (iii).
(iv) First, notice that (ii) implies $\mathbb{F}_{N,p} \subseteq \mathbb{Z}_{(p)}$. Let $x/y, x'/y' \in \mathbb{F}_{N,p}$ be distinct. Necessarily, $H_p\left(x/y\right) \neq H_p\left(x'/y'\right)$. Since H_p is a homomorphism, then $H_p\left(x/y - x'/y'\right) \neq 0$, which implies x/y and x'/y' lie in distinct cosets.
(v) The result follows immediately from (iv) and the isomorphism $\mathbb{Z}_{(p)}/\ker(H_p) \cong \mathbb{Z}_p$. □

We may now define the mapping that allows us to recover an element of $\mathbb{F}_{N,p}$ given an arbitrary integer.

Definition 3. *Let p be prime and $h \in \mathbb{Z}$. Define*

$$H_p^{-1} : \mathbb{Z} \to \mathbb{F}_{N,p} \;\; \text{by} \;\; h \mapsto \frac{x}{y \bmod p}, \tag{1}$$

where $\mathsf{MEEA}(p,h) = (x,y)$.

Remark 1. Lemma 1(ii) guarantees that the output $x/\big(y(\bmod p)\big)$ from the preceding definition is in $\mathbb{F}_{N,p}$. Moreover, by the definition of the order-(N,p) Farey fractions, H_p^{-1} is surjective.

Proposition 2. *If $x/y \in \mathbb{F}_{N,p}$ and $h \in \mathbb{Z}_p$, then $H_p^{-1}\left(H_p\left(x/y\right)\right) = x/y$ and $H_p\big(H_p^{-1}(h)\big) = h$.*

Proof. Obvious. □

The following results establish the compatibility of H_p^{-1} with arbitrary arithmetic circuits. For simplicity, we represent a circuit by the multivariate polynomial which it computes.

Lemma 2. *Let $h_1, \ldots, h_k \in \mathbb{Z}$. If P is a polynomial in k variables over $\mathbb{Z}$ which takes rational arguments, and $H_p^{-1}\big(P(h_1, \ldots, h_k)\big) = a/b$, then*

$$H_p\Big(P\big(H_p^{-1}(h_1), \ldots, H_p^{-1}(h_k)\big)\Big) = H_p\left(\frac{a}{b}\right).$$

Proof. Suppose $H_p^{-1}(h_i) = x_i/y_i$. Certainly $x_i y_i^{-1} = h_i (\bmod\ p)$, whence

$$P\left(H_p^{-1}(h_1), \ldots, H_p^{-1}(h_k)\right) = P\big(h_1, \ldots, h_k\big)(\bmod\ p).$$

The result follows, since $P(h_1, \ldots, h_k) = ab^{-1}(\bmod\ p)$. □

Proposition 3. *If $h_1, \ldots, h_k \in \mathbb{Z}$ and P is a polynomial in k variables over $\mathbb{Z}$ which takes rational arguments, then*

$$H_p^{-1}\big(P(h_1, \ldots, h_k)\big) = H_p^{-1}\Big(H_p\Big(P\big(H_p^{-1}(h_1), \ldots, H_p^{-1}(h_k)\big)\Big)\Big).$$

Proof. Since $\mathbb{F}_{N,p}$ is not closed under addition and multiplication then

$$\mathcal{P} = P\big(H_p^{-1}(h_1), \ldots, H_p^{-1}(h_k)\big)$$

need not be an element of $\mathbb{F}_{N,p}$. However, by Lemma 3, $\mathcal{P}$ and $H_p^{-1}\big(P(h_1, \ldots, h_k)\big)$ lie in the same coset (are equivalent modulo p). Consequently, $H_p^{-1}\big(H_p(\mathcal{P})\big) = H_p^{-1}\big(P(h_1, \ldots, h_k)\big)$. □

We now present the remaining maps which are fundamental to our scheme.

Definition 4. *Let $g = p_1 \cdots p_k$ be a product of at least two distinct primes. Define maps*

$$H_g : \mathbb{Q} \to \mathbb{Z}_g \ \textit{ by } \ \frac{x}{y} \mapsto \begin{cases} xy^{-1}(\bmod\ g), & \textit{if } \gcd(g, y) = 1 \\ 0, & \textit{if } \gcd(g, y) \neq 1 \end{cases}$$

$$\widetilde{H}_g^{-1} : \mathbb{Z} \to \mathbb{Q} \ \textit{ by } h \mapsto \frac{x}{y \bmod g}, \ \textit{ where } \mathsf{MEEA}(g, h) = (x, y).$$

Remark 2. If n is an integer, then $H_g(n) = n \pmod{g}$.

Remark 3. We write "$\widetilde{H}_g^{-1}(\cdot)$" instead of "$H_g^{-1}(\cdot)$" because $\widetilde{H}_g^{-1}$ is not the inverse of H_g when g is composite. This is because if $\widetilde{H}_g^{-1}(h) = x/y$ we may have $y|g$, in which case y is not invertible modulo g, and so (provided $x \neq 0$) $H_g\big(\widetilde{H}_g^{-1}(x) = 0 \neq x/y$.

Recall that the Chinese Remainder Theorem (CRT) simply describes an isomorphism $\mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_k} \cong \mathbb{Z}_{p_1 \cdots p_k}$. The image of $(h_1, \ldots, h_k)$ under this isomorphism will be denoted by $h = \mathsf{CRT}_{p_1,\ldots,p_k}(h_1, \ldots, h_k)$.
Henceforth, for primes $p_1, \ldots, p_k$ and $h_1, \ldots, h_k \in \mathbb{Z}$, we will denote

$$\widetilde{H}^{-1}_{p_1 \cdots p_k}\Big(\mathsf{CRT}_{p_1,\ldots,p_k}(h_1, \ldots, h_k)\Big)$$

by $\widetilde{H}^{-1}_{p_1,\ldots,p_k}(h_1, \ldots, h_k)$.

Lemma 3. *Let $g = p_1 \cdots p_k$ be a product of distinct primes. If $H_g\big(x/y\big) \neq 0$, then $H_g\big(x/y\big) = H_{p_i}\big(x/y\big) \pmod{p_i}$.*

Proof. To avoid confusion, we will denote the (multiplicative) inverse of y modulo n by y_n^{-1}. If $h = H_g\big(x/y\big) \neq 0$, then y is invertible modulo p_i for each i. Put $h_i = H_{p_i}\big(x/y\big)$. By definition, $h = xy_g^{-1} \pmod g$ and $h_i = xy_{p_i}^{-1} \pmod{p_i}$ for all i. Multiplying both sides of each congruence by y yields $hy = x \pmod g$ and $x = h_i y \pmod{p_i}$. It follows that $hy = h_i y \pmod{p_i}$ for all i. Finally, since y is invertible modulo each p_i, $h = h_i \pmod{p_i}$. □

Proposition 4. *If $g = p_1 \cdots p_k$ is a product of distinct primes, $H_g\big(x/y\big) \neq 0$, and $x/y \in \mathbb{F}_{N,p_i}$, then $H_{p_i}^{-1}\left(H_g\left(x/y\right)\right) = x/y$.*

Proof. By Lemma 1(ii), Lemma 3, and the definition of $H_{p_i}^{-1}$, we see that

$$H_{p_i}^{-1}\Big(H_g\big(x/y\big)\Big) = H_{p_i}^{-1}\Big(H_{p_i}\big(x/y\big)\Big).$$

The result then follows from Lemma 2. □

Lemma 4. *If g is a product of distinct primes and $g|n$, then $\widetilde{H}_g^{-1}(n) = 0$.*

Proof. Observe that $\gcd(g, n) = g > \Big\lfloor \sqrt{(g-1)/2} \Big\rfloor$. The result then follows from Lemma 1(iii). □

3 The AGCD Problem

Informally, the AGCD problem is defined as follows: given polynomially many samples of the form $x = r + qp$ for a randomly chosen odd prime p, find p. Since in the remainder of this paper we will refer to known (ρ, η, γ) AGCD parameters, a formal definition of the AGCD problem is reproduced below.

Definition 5. [11] (AGCD). *Let $p, X \geq 1$, and ϕ a distribution over $\mathbb{Z}$. We define $A_{X,\phi}^{\mathrm{AGCD}}(p)$ as the distribution over $\mathbb{Z}$ obtained by sampling $q \leftarrow \mathbb{Z} \cap \big[0, X/p\big)$ and $r \leftarrow \phi$, and returning $x = qp + r$.*

Let $\mathcal{D}$ be a distribution over $\mathbb{Z} \cap [0, X)$. $\mathrm{AGCD}_{X,\phi}(\mathcal{D})$ consists in distinguishing, given arbitrarily many independent samples, between the uniform distribution over $\mathbb{Z} \cap [0, X)$ and the distribution $A_{X,\phi}^{\mathrm{AGCD}}(p)$ for $p \leftarrow \mathcal{D}$. We use *the notation* $\mathrm{AGCD}_{X,\phi}^{m}\left(\mathcal{D}\right)$ *to emphasize the number of samples* m *used by the eventual distinguisher. We say that an algorithm* $\mathcal{A}$ *is an* (ϵ_1, ϵ_2)*-distinguisher for* $\mathrm{AGCD}_{X,\phi}\left(\mathcal{D}\right)$ *if, with probability* $\geq \epsilon_2$ *over the choice of* $p \leftarrow \mathcal{D}$*, its distinguishing advantage between* $A_{X,\phi}^{\mathrm{AGCD}}\left(p\right)$ *and* $U\left(\mathbb{Z} \cap [0, X)\right)$ *is* $\geq \epsilon_1$*.*

For $\rho, \eta, \gamma \geq 1$*, the* (ρ, η, γ)*-AGCD problem is* $\mathrm{AGCD}_{2^{\gamma},\phi}\left(\mathcal{D}\right)$ *with* $\mathcal{D}$ *the uniform distribution over* η*-bit prime integers and* ϕ *the uniform distribution over* $\mathbb{Z} \cap (-2^{\rho}, 2^{\rho})$*.*

Cheon and Stehlé discuss a reduction from the Learning With Errors problem (LWE) to a variant of the AGCD where such search variant consists in finding the unknown p [11] while also introducing a reduction from the search variant to the decision variant. They arrive at a set of secure AGCD parameters via reduction of a LWE instance. For appreciating this reduction, we refer the reader to [11] since we shall not repeat that discussion in this work. Instead, we will use the proposed AGCD parameters in [11], the motivation for each one of them, together with some additional considerations from [60].

3.1 Recommended AGCD Parameters

Following [60, 11], we let ρ denote the size of the noise, η denote the size of the secret greatest common divisor, and γ denote the size of an AGCD sample. Cheon and Stehlé note that for the AGCD problem to be potentially hard, the parameters must satisfy the following: $\rho \geq \lambda$ in order to prevent brute force attacks on the noise as discussed in [7, 34], $\eta > \rho$, and $\gamma \geq \mathit{\Omega}\left(\left(\lambda/\log\lambda\right)\left(\eta-\rho\right)^{2}\right)$ in order to prevent lattice reduction attacks on AGCD such as orthogonal lattice attacks [52, 60], as well as the Lagarias' simultaneous Diophantine approximation attack [43], and the Cohn-Heninger attack [21, 13].

4 A Private-Key Leveled FHE Scheme

Now we introduce a private-key leveled FHE scheme based on Hensel codes. Our motivation is to provide a basic blueprint for a leveled FHE scheme with Hensel codes and then use it as the foundation of a public-key leveled FHE scheme by only applying an asymmetric property we have with Hensel codes. The reader can see this private encryption scheme as first step towards its public-key counterpart, which is the candidate scheme we want to highlight. For this reason, we will concentrate the discussions about correctness, security, and practical implications on the public-key version.

Given a parameter λ and the parameter d, define ρ, η, γ, and μ as follows: $\rho = \lambda$, $\eta = 2(d+2)\lambda$, $\mu = \gamma - \eta - 2\lambda$, and $\gamma = \frac{\lambda}{\log_2(\lambda)}\left(\eta - \rho\right)^2$. The encryption scheme is then given by:

- `Gen` takes λ and d as input and generates uniform primes $p_1, \ldots, p_5$ such that $|p_1|_{\mathrm{bits}} = \rho + 1$, $|p_3|_{\mathrm{bits}}, |p_3|_{\mathrm{bits}} = \frac{\rho}{2}$, $|p_4|_{\mathrm{bits}} = \eta$, $|p_5|_{\mathrm{bits}} = \mu$. We set $sk = (p_1, p_2, p_3, p_4)$ and $evk = g = \prod_{i=1}^{5} p_i$. Note that $|g|_{\mathrm{bits}} = \gamma$. The message space is defined as $\mathcal{P} = \mathbb{F}_{N,p_1}$. We write the syntax as $(sk, evk) \leftarrow \texttt{Gen}\left(1^\lambda, 1^d\right)$.

- $\texttt{Enc}$ takes a message $m \in \mathcal{P}$ and sk, evk as input, generates uniform and independent $s_1 \leftarrow \mathbb{Z}_{2^{\lambda-1}}$, $s_2 \leftarrow \mathbb{Z}_{p_2}$, and $s_3 \leftarrow \mathbb{Z}_{p_3}$, and $\delta \leftarrow \mathbb{Z}_{g/p_4}$, and then computes $c = \left| H_g\left(s_1 \cdot \widetilde{H}^{-1}_{p_1,p_2,p_3}(0, s_2, s_3) + m\right) + \delta p_4 \right|_g$. We write this as $c \leftarrow \texttt{Enc}_{sk,evk}(m)$.
- $\texttt{Dec}$ takes c and sk as input and computes $m = H^{-1}_{p_1}\left(H_{p_1}\left(H^{-1}_{p_4}(c)\right)\right)$. We write this as $m = \texttt{Dec}_{sk}(c)$.
- Addition and multiplication of ciphertexts are computed in the natural way over the integers modulo g.

Remark 4. In the encryption algorithm, let $x/y = \widetilde{H}^{-1}_{p_1,p_2,p_3}(0, s_2, s_3)$. Then, x/y is a rational encoding of zero, and so is $s_1 \cdot x/y$. Moreover, setting $|p_1|_{\text{bits}} = \rho + 1$ implies $p_1 > 2\left\lfloor\sqrt{(p_1p_2p_3 - 1)/2}\right\rfloor$, which guarantees (by Lemma 2.5(ii)) that $\gcd(p_1, y) = 1$, so $H_{p_1}(x/y)$ is defined.

Remark 5. In the decryption algorithm, for all c output by $\texttt{Enc}$, with high probability, it holds that $H^{-1}_{p_4}(c) \notin \mathbb{F}_{N,p_1}$. We address this issue by computing $H_{p_1}\left(H^{-1}_{p_4}(c)\right)$ which gives as a Hensel code in $\mathbb{Z}_{p_1}$. Then, we can just decode that Hensel code using p_1, which gives us the expression in the decryption algorithm, so we obtain the desired member of $\mathbb{F}_{N,p_1}$.

5 A Public-Key Leveled FHE Scheme

Now we introduce a public-key leveled FHE scheme that is similar to the previously described private-key encryption with the exception that we now explore asymmetric Hensel "encodings". The parameters we use are conservative in comparison with the ones recommended in [10]. The reason for employing a more conservative parameter definition is due to the fact that the ciphertext expansion of our construction is significantly more efficient than any leveled FHE construction where the message space is defined as $\{0, 1\}$. At the same time, we know there are room for optimizations which can further improve the already encouraging runtime results presented in Section 7.1. Given a parameter λ and the parameter d, define η, γ, and μ as follows:

$$\rho = \lambda, \eta = d\lambda, \mu = d^2\lambda \log_2(\lambda) - \eta - 2\lambda - 3, \gamma = 2\eta + (3\lambda)/2 + \mu + 3. \quad (2)$$

- $\texttt{Gen}$ takes 1^λ and 1^d as input and generates uniform and independent odd primes p_1, p_2, p_3, and p_4' such that $|p_1|_{\text{bits}} = \lambda$, $|p_2|_{\text{bits}} = \lambda + 3$, $|p_3|_{\text{bits}} = \eta$, $|p_4'|_{\text{bits}} = \eta$, so we compute $p_4 = (p_4')^{\mu/\eta+1}$. Let $g = p_1 \cdots p_4$ and $g' = p_3p_4$. For $t \leftarrow \mathbb{Z}_{2^{\lambda-1}}$ and $\delta_e \leftarrow \mathbb{Z}_{2^{\gamma-\eta}}$, we compute

$$e = H_g\left(H_{p_3}\left(\widetilde{H}^{-1}_{p_1,p_2}(0, t)\right) + \delta_e p_3\right) \quad (3)$$

The public key is $pk = (e, g', evk = g)$ and the secret key is $sk = (p_1, p_3)$.

- `Enc` encrypts a message $m \in \mathbb{F}_{N,p_1}$ by choosing $s_1, s_2 \leftarrow \mathbb{Z}_{2^{\lambda-1}}$ and $\delta \leftarrow [p_1p_2, p_1p_2p_4) \cap \mathbb{Z}$ and then computing

$$c \leftarrow \mathtt{Enc}_{pk,evk}(m) = H_g\left(H_{g'}\left(s_1e + m\right) + s_2g' + \delta\left(g'\right)^2\right). \tag{4}$$

- `Dec` takes a ciphertext c as input and computes m as follows:

$$m = \mathtt{Dec}_{sk,pk}\left(c\right) = H_{p_1}^{-1}\left(H_{p_1}\left(H_{p_3}^{-1}\left(c\right)\right)\right). \tag{5}$$

Remark 6. the constant e in the public key should never equal $\delta_e p_3$, else an adversary trivially computes $\gcd(e, g') = p_3$ which compromises the secret key. To this end, recall that $\widetilde{H}_{p_1,p_2}^{-1}(0,t) = \widetilde{H}_{p_1p_2}^{-1}(h)$, where $h = \mathsf{CRT}_{p_1,p_2}(0,t)$. Since $t \neq 0$, $h \neq 0$ and $\gcd(h, p_1p_2) = p_1$. Lemma 1(iii) then implies that $\widetilde{H}_{p_1p_2}^{-1}(h) \neq 0$ as long as $p_1 < \left\lfloor\sqrt{(p_1p_2-1)/2}\right\rfloor$. This is guaranteed since $|p_1|_{\text{bits}} = \lambda$ and $\left|\left\lfloor\sqrt{(p_1p_2-1)/2}\right\rfloor\right|_{\text{bits}} > \lambda$.

5.1 Correctness

Here we continue with the previously-adopted convention of using multivariate polynomials instead of arithmetic circuits.

Definition 6. *Let $\mathcal{P}_{k,n} \subseteq \mathbb{Q}[x_1, \ldots, x_k]$ be the family of polynomials of the form $P(x_1, \ldots, x_k) = y_1 * y_2 * \cdots * y_n$, where $y_i \in \{x_1, \ldots, x_k\}$ and $*$ is either $+$ or $\times$.*

Lemma 5. *Let α be a natural number, $P \in \mathcal{P}_{k,n}$, and $a_1/b_1, \ldots, a_k/b_k \in \mathbb{Q}$ be unknowns. Further, suppose $P\left(a_1/b_1, \ldots, a_k/b_k\right) = x/y$. If each $|a_i|, |b_i| \leq \alpha$, then $|x| \leq n\alpha^n$ and $|y| \leq \alpha^n$.*

Proof. If $P \in \mathcal{P}_{k,n}$ has $i \leq n-1$ additions, then the numerator of x is the sum of $i+1$ monomials, each being a product of the a_i,b_j. Moreover, the denominator y is simply a product of n (not necessarily distinct) of the b_j. Note that for each monomial m summand of x satisfies: m/y is a product (possibly with repeated factors) of some number of the a_i/b_i. It follows that each monomial in the numerator is a product of at most n of the a_j,b_j. For if there is a monomial m with more than n factors, then m/y reduces to a fraction with with more factors (the a_i,b_j) in the numerator than the denominator. Such a fraction cannot satisfy the above note, and so a contradiction is obtained. Now, since $|a_i|, |b_j| \leq \alpha$, we see that the denominator y and the monomial summands of x all have absolute value at most α^n. The result then follows since x has at most n monomial summands. □

Theorem 1 (Correctness). *For all sk,pk, and evk output by* `Gen` *and all $m \in \mathbb{F}_{N,p_1}$,*

$$\mathtt{Dec}_{sk,pk}\left(\mathtt{Enc}_{pk,evk}\left(m\right)\right) = m. \tag{6}$$

Let $P \in \mathcal{P}_{p_1,D}$, $m_1, \dots, m_k \in \mathbb{F}_{N,p_1}$ and $c_i \leftarrow \mathtt{Enc}_{pk,evk}(m_i)$.
If $P(m_1, \dots, m_k) \in \mathbb{F}_{N,p_1}$, $d \leq \lambda$, and $D \leq \big(d/5\big) - 1$, then

$$\mathtt{Dec}_{sk,pk}\Big(P\big(c_1, \dots, c_k\big)\Big) = P\left(m_1, \dots, m_t\right). \tag{7}$$

Proof. Let $m \in \mathbb{F}_{N,p_1}$, and suppose $c = \mathtt{Enc}_{pk,evk}(m)$. By construction,

$$c = H_g\Big(H_{g'}(se+m) + s_2 g' \delta (g')^2\Big) = H_{g'}(se+m) + \alpha p_3,\ \alpha \in \mathbb{Z},$$

where $e = H_g\Big(H_{p_3}\Big(\widetilde{H}^{-1}_{p_1,p_2}(0,t)\Big) + \delta_e p_3\Big)$ and $s \in \mathbb{Z}_{p_1}$.
Proceeding with $\mathtt{Dec}$ $\big($which computes $H^{-1}_{p_1}(H_{p_1}(H^{-1}_{p_3}(c)))\big)$ and applying Proposition 2 and Proposition 3 , we obtain

$$\begin{aligned} H^{-1}_{p_3}(c) &= H^{-1}_{p_3}\left(H_{g'}\left(se+m\right) + \alpha p_3\right), \ \text{ for some } \alpha \in \mathbb{Z} \\ &= H^{-1}_{p_3}\left(H_{p_3}\left(H^{-1}_{p_3}\left(H_{g'}\left(se+m\right)\right) + H^{-1}_{p_3}\left(\alpha p_3\right)\right)\right) \\ &= H^{-1}_{p_3}\left(H_{p_3}\left(H^{-1}_{p_3}\left(H_{g'}\left(se+m\right)\right)\right)\right) \\ &= H^{-1}_{p_3}\left(H_{g'}\left(se+m\right)\right) \end{aligned}$$

Put $x/y = \widetilde{H}^{-1}_{p_1,p_2}(0,t)$ and $N = \left\lfloor \sqrt{(p_3-1)/2} \right\rfloor$. We note that $|s|_{\text{bits}} \leq \lambda$, and deduce from Lemma 1(i) that $|x|_{\text{bits}} \leq \lambda + 1$ and $|y|_{\text{bits}} \leq \lambda + 2$. Further, since $(d\lambda - 2)/2 \leq |N|_{\text{bits}}$ (for $d \geq 5$), we see that $|sx|_{\text{bits}}, |y|_{\text{bits}} \leq |N|_{\text{bits}}$. Consequently, $sx/y \in \mathbb{F}_N \subseteq F_{N,p_3}$. Now, through repeated applications of the above observation, Lemma 1(iii), Proposition 3, and Lemma 4, we obtain

$$H^{-1}_{p_3}\left(H_{g'}\left(se+m\right)\right) = H^{-1}_{p_3}\left(H_{p_3}\left(\frac{sx}{y} + m\right)\right).$$

By comparing bit lengths (as above), we find that $sx/y + m \in \mathbb{F}_{N,p_3}$ (this time, as long as $d \geq 6$). Thus, $H^{-1}_{p_3}(c) = sx/y + m$.
Lastly, we compute

$$H^{-1}_{p_1}\left(H_{p_1}\left(\frac{sx}{y} + m\right)\right) = H^{-1}_{p_1}\left(H_{p_1}\left(\frac{sx}{y}\right) + H_{p_1}(m) - kp_1\right),\ k \in \{0,1\}.$$

Since H_{p_1} is a ring homomorphism and $s \in \mathbb{Z}_{2^{\lambda-1}} \subseteq \mathbb{Z}_{p_1}$, we have
$H_{p_1}\big(sx/y\big) = H_{p_1}(s)H_{p_1}\big(sx/y\big) \pmod{p_1} = sH_{p_1}\big(sx/y\big) - np_1$ for some $n \in \mathbb{Z}$.
Moreover, $H_{p_1}\big(x/y\big) = 0$ by construction.

Whence,

$$H^{-1}_{p_1}\left(H_{p_1}\left(\frac{sx}{y} + m\right)\right) = H^{-1}_{p_1}\Big(H_{p_1}(m) - (k+n)p_1\Big).$$

With a final application of Proposition 3 and Lemma 4, the above simplifies to $H^{-1}_{p_1}\big(H_{p_1}(m)\big) = m$. Thus (6) is established.

We now show that the scheme is compatible with homomorphic operations.

Let $c_1, \dots, c_k \in \mathbb{Z}_g$ be ciphertexts with corresponding messages $m_i \in \mathbb{F}_{N,p_1}$, and $P \in \mathcal{P}_{k,D}$. Then, as above, there are $s_i \in \mathbb{Z}_{2^{\lambda-1}}$ and an integer α' such that for each i, $c_i = H_{g'}(s_i e + m_i) + \alpha' g'$. Suppose further that $P(m_1, \dots, m_k) \in \mathbb{F}_{N,p_1}$. Proceeding as in the proof of (6), we compute

$$\begin{aligned}
&H_{p_3}^{-1}\big(P(c_1, \dots, c_k)\big)\\
&= H_{p_3}^{-1}\Big(P\big(H_{g'}(s_1 e + m_1), \dots, H_{g'}(s_k e + m_k)\big)\Big)\\
&= H_{p_3}^{-1}\Big(H_{p_3}\Big(P\left(H_{p_3}^{-1}\big(H_{g'}(s_1 e + m_1)\big), \dots, H_{p_3}^{-1}\big(H_{g'}(s_k e + m_k)\big)\right)\Big)\Big)\\
&= H_{p_3}^{-1}\Big(H_{p_3}\Big(P\left(s_1 \tfrac{x}{y} + m_1, \dots, s_k \tfrac{x}{y} + m_k\right)\Big)\Big).
\end{aligned}$$

Now, let $x_i/y_i = s_i x/y + m_i$ and $x^*/y^* = P\left(x_1/y_1, \dots, x_k/y_k\right)$. We will show that $x^*/y^* \in \mathbb{F}_{N,p_3}$. Recall that $|s_i|_{\text{bits}} \leq \lambda$, $|x|_{\text{bits}} \leq \lambda + 1$ and $|y|_{\text{bits}} \leq \lambda + 2$.

It follows that $|x_i|_{\text{bits}} \leq (5\lambda + 3)/2$ and $|y_i|_{\text{bits}} \leq (3\lambda + 3)/2$. For simplicity, we take $(5\lambda + 3)/2$ as the bound on bit length for both x_i and y_i.

By invoking Lemma 5, and the binary logarithm (to count bit lengths), we have

$$|x^*|_{\text{bits}}, |y^*|_{\text{bits}} \leq \log_2(D) + D\left(\frac{5\lambda + 3}{2}\right) + 1$$

Now, recalling that $\left|\left\lfloor\sqrt{(p_3 - 1)/2}\right\rfloor\right|_{\text{bits}} \geq (d\lambda - 2)/2$, we see that

$$\log_2(D) + D\left(\frac{5\lambda + 3}{2}\right) + 1 \leq \frac{d\lambda - 2}{2}$$

is a sufficient condition to guarantee that $x^*/y^* \in \mathbb{F}_N$, where $N = \left\lfloor\sqrt{(p_3 - 1)/2}\right\rfloor$. Easy algebraic manipulations verify that the above inequality reduces to

$$\frac{\log_2(D^2) + 3D + 3}{d - 5D} \leq \lambda.$$

The hypotheses that $d \leq \lambda$ and $D \leq \big(d/5\big) - 1$ guarantee that the above inequality is true. Whence, $x^*/y^* \in \mathbb{F}_N \subseteq \mathbb{F}_{N,p_3}$, and

$$H_{p_3}^{-1}\big(P(c_1, \dots, c_k)\big) = \frac{x^*}{y^*} = P\left(\frac{x_1}{y_1}, \dots, \frac{x_k}{y_k}\right).$$

All that remains is to compute $H_{p_1}^{-1}\Big(H_{p_1}\big(x^*/y^*\big)\Big)$. To this end, observe that since H_{p_1} is a homomorphism under addition and multiplication modulo p_1, $H_{p_1}\left(x_i/y_i\right) = H_{p_1}\left(s_i x/y\right) + H_{p_1}(m_i) - \alpha_i p_1 = H_{p_1}(m_i) - \alpha_i p_1$, and $H_{p_1}\Big(P\big(x_1/y_1, \dots, x_k/y_k\big)\Big) = P\Big(H_{p_1}\big(x_1/y_1\big), \dots, H_{p_1}\big(x_k/y_k\big)\Big) - \alpha p_1$, where $\alpha, \alpha_i \in \mathbb{Z}$. Now, by the preceding observations and Proposition 3, we obtain

$$\begin{aligned}
&H_{p_1}^{-1}\left(H_{p_1}\left(\tfrac{x^*}{y^*}\right)\right)\\
&= H_{p_1}^{-1}\left(P\left(H_{p_1}(m_1) - \alpha_1 p_1, \dots, H_{p_1}(m_k) - \alpha_k p_1\right) - \alpha p_1\right)\\
&= H_{p_1}^{-1}\left(H_{p_1}\left(P(m_1, \dots, m_k)\right)\right)
\end{aligned}$$

Finally, since $P(m_1, \dots, m_k) \in \mathbb{F}_{N,p_1}$,

$$H_{p_1}^{-1}\left(H_{p_1}\left(P\big(m_1, \dots, m_k\big)\right)\right) = P\big(m_1, \dots, m_k\big).$$

This completes the proof of (7).

Remark 7. The set $\mathcal{P}_{p_1,D}$, $D = \big\lfloor \big(d/5\big) - 1 \big\rfloor$, does not contain all polynomials with which our scheme is compatible. In particular, we note that for any polynomial Q taking rational arguments: if $Q(m_1, \dots, m_k) \in \mathbb{F}_{N,p_1}$ and $Q\big(s_1 x/y + m_1, \dots, s_k x/y + m_k\big) \in \mathbb{F}_{N,p_3}$, then $\mathtt{Dec}_{sk,pk}\Big(Q\big(c_1, \dots, c_k\big)\Big) = Q\left(m_1, \dots, m_t\right)$.

Remark 8. The requirement that $P(m_1, \dots, m_k) \in \mathbb{F}_{N,p_1}$ may seem unreasonable since $\mathbb{F}_{N,p_1}$ is not closed under addition. However, one can always choose p_1 large enough to guarantee that the scheme is compatible with the requisite polynomials P. For example, if one only needs to work with fractions whose numerators and denominators are bounded (in absolute value) by M, then one simply chooses p_1 so that $N = \left\lfloor \sqrt{(p_1 - 1)/2} \right\rfloor \gg M$. This creates a "bounded closure".

6 Security Analysis

We present a discussion on the security properties of our construction from at least four perspectives: CPA indistinguishability, an analysis on encryptions of zero, factoring concerns, and an intrinsic hardness of Hensel codes that are meant to violate correctness boundaries.

6.1 Indistinguishability under Chosen Plaintext Attacks (CPA)

We present a variant of the AGCD assumption where the distinguisher is additionally given e, g, g'. For simplicity, we use (e, g, g')-AGCD to denote the variant of the AGCD problem and (e, g, g')-AGCD$\,(p)$ to denote its associated distribution.

Definition 7 **((e, g, g')-AGCD).** *Let u, v, p be primes such that $|u|_{\text{bits}} = \lambda$, $|v|_{\text{bits}} = \lambda + 3$, $|p|_{\text{bits}} = \eta$, and $|w'|_{\text{bits}} = \eta$ such that $w = (w')^{\gamma/\mu}$, $g = uvpw$ and $g' = pw$. We sample $r, s \leftarrow \mathbb{Z}_{2^{\lambda-1}}$, $q \leftarrow [uv, uvw) \cap \mathbb{Z}$, and $\sigma \leftarrow \mathbb{Z}_{2^{\gamma-\eta}}$ and we compute $e = H_g\left(H_p\left(\widetilde{H}_{u,v}^{-1}(0, t)\right) + \sigma p\right)$. Finally, we compute $x = H_g\left(r + qg'\right)$ and (e, g, g')-AGCD$\,(p)$ outputs x together with e, g, g'.*

Lemma 6. *For any message $m \in \mathbb{F}_{N,p_1}$, we can perfectly simulate $\mathtt{Enc}_{pk,evk}\left(m\right)$ by obtaining x, e, g, g' from (e, g, g')-AGCD$\,(p)$, sampling $t \leftarrow \mathbb{Z}_{2^{\lambda-1}}$, and outputting $c = H_g\left(xg' + H_{g'}\left(te + m\right)\right)$.*

Proof. Fix $m \in \mathbb{F}_{N,u}$. We claim that the following simulated ciphertext c_{sim} decrypts to m: $c_{\text{sim}} = H_{g'}(te+m) + qg' - kg$, for some integer k. We further simplify to get $c = H_{g'}(te+m) + \alpha g'$, where $\alpha = x - k(g/g')$. It now follows immediately from the proof of correctness $\big($Theorem 5.1, Equation (6)$\big)$ that $\texttt{Dec}_{sk,pk}(c_{\text{sim}}) = m$. Furthermore, we observe trivially that the simulated encryptions are distributed identically to the actual encryptions.

Definition 8. *Consider the following experiment: a uniform η-bit prime p is chosen along with a fixed message m from $\mathbb{Z}_{2^\lambda}$. Then a uniform bit $b \leftarrow \{0,1\}$ is chosen. A distinguisher $\mathcal{D}$ is given g, g' from (e,g,g')-AGCD$\,(p)$, and then:*

- *If $b = 0$, the distinguisher is given repeated random samples from (e,g,g')-AGCD$\,(p)$.*
- *If $b = 1$, the distinguisher is given repeated simulations of encryptions of m using samples from (e,g,g')-AGCD$\,(p)$ in the form of $H_g\left(xg' + H_{g'}(m)\right)$.*
- *The distinguisher outputs a guess b', and succeeds if $b' = b$. It ϵ-distinguishes if $\Pr[b' = b] = 1/2 + \epsilon$.*

Assumption 1 *For any probabilistic polynomial-time distinguisher $\mathcal{D}$, the probability that $\mathcal{D}$ is successful in the preceding experiment is negligible. That is, at best, $\mathcal{D}$ ϵ-distinguishes with $\epsilon = \epsilon(\lambda)$ negligible.*

Theorem 2. *The private-key leveled FHE scheme described in Section 5 is CPA-secure under Assumption 1.*

Proof. Fix an adversary $\mathcal{A}$ attacking the scheme. Construct an adversary $\mathcal{B}$ as follows: $\mathcal{B}$ is given repeated samples from (unknown) distribution. $\mathcal{B}$ runs $\mathcal{A}$. When $\mathcal{A}$ requests an encryption of m, then $\mathcal{B}$ does: 1) Get a sample x from the given distribution, 2) Return a simulated ciphertext c (dependent on x) to $\mathcal{A}$. When $\mathcal{A}$ outputs its challenge messages m_0, m_1 then $\mathcal{B}$ chooses a random bit b and does the exact same thing as above using the message m_b. When $\mathcal{A}$ outputs a guess b', then $\mathcal{B}$ outputs 1 iff $b = b'$.

Claim (1). When $\mathcal{B}$ is given samples from distribution (e,g,g')-AGCD$\,(p)$, then the probability that $\mathcal{B}$ outputs 1 is identical to $\mathcal{A}$'s success probability in the CPA experiment.

Proof (1). This follows since when $\mathcal{B}$ is given samples from distribution (e,g,g')-AGCD, then $\mathcal{A}$'s view is identical to its view in the CPA experiment.

Claim (2). When $\mathcal{B}$ is given random samples from (e,g,g')-AGCD$\,(p)$, then the probability that $\mathcal{B}$ outputs 1 is $1/2$.

Proof (2). This follows since $\mathcal{A}$'s view is independent of b. Indeed, for any message m, $\mathcal{B}$ can perfectly simulate a ciphertext for m using random samples from (e,g,g')-AGCD$\,(p)$. So, as per Lemma 6, the challenge ciphertext is computationally indistinguishable from a random element of (e,g,g')-AGCD$\,(p)$, which is independent of m regardless of the choice of b. It follows that $\mathcal{A}$ correctly outputs $b' = b$ with probability exactly $1/2$.

By Assumption 1, the difference in the probability that $\mathcal{B}$ outputs 1 when given samples from (e, g, g')-AGCD(p) and the probability that it outputs 1 when given uniform samples is negligible. It follows from this and the above claims that the probability that $\mathcal{A}$ succeeds in the CPA experiment is $1/2 + \mathsf{negl}$. This completes the proof. □

6.2 Further Discussion of Security

Here we show that, under a particular assumption, ciphertexts are indistinguishable from random elements of $\mathbb{Z}_g$.

Lemma 7. *There are $p_1^2 p_2$ distinct encryptions of* 0.

Proof. Encryptions of 0 are of the form $\mathtt{Enc}_{pk,evk}(0) = H_g\big(H_{g'}(se) + \delta g'\big)$, where $s \leftarrow [0, p_1) \cap \mathbb{Z}$ and $\delta \leftarrow [p_1p_2, p_1p_2p_4) \cap \mathbb{Z}$. We will show that each pair s, δ yields a unique encryption of 0. To this end, suppose $s \neq s'$ or $\delta \neq \delta' \pmod{p_1p_2}$. If $H_g\big(H_{g'}(se) + \delta g'\big) = H_g\big(H_{g'}(s'e) + \delta' g'\big)$, then we deduce that $s = s' \pmod{g'}$. Since $0 \leq s, s' < g'$, $s = s'$, a contradiction. All that remains is the case where $s = s'$ and $\delta \neq \delta' \pmod{p_1p_2}$. Again, if $H_g\big(H_{g'}(se)+\delta g'\big) = H_g\big(H_{g'}(s'e)+\delta' g'\big)$, then $H_{g'}(se) = H_{g'}(s'e)$ implies $\delta g' - kg = \delta' g' - k'g$, for some $k, k' \in \mathbb{Z}$. Rearranging the equation yields $\delta - \delta' = (k - k')p_1p_2$, also a contradiction. The result follows from that facts that there are p_1 choices for s, and p_1p_2 choices for δ. □

We now define an experiment in which a distinguisher tries to distinguish between a random subset of $[0, g) \cap \mathbb{Z}$ and a set of encryptions of 0.

Definition 9. *Let $\mathcal{D}$ be a probabilistic polynomial-time distinguisher which knows the public-key $pk = (e, g', g)$. A uniform bit $b \leftarrow \{0, 1\}$ and a random $k \leq p_1^2 p_2$ are chosen, and then:*

- *If $b = 0$, then $\mathcal{D}$ is given a random subset of $[0, g) \cap \mathbb{Z}$ with k elements.*
- *If $b = 1$, then $\mathcal{D}$ is given a set of k random encryptions of* 0.
- *$\mathcal{D}$ outputs a guess $b' \in \{0, 1\}$ and succeeds if $b' = b$. Say $\mathcal{D}$ ϵ-distinguishes if* $\Pr[b' = b] = 1/2 + \epsilon$.

Assumption 2 *The probability that $\mathcal{D}$ is successful in the preceding experiment is negligible. That is, $\mathcal{D}$ can only ϵ-distinguish if $\epsilon = \epsilon(\lambda)$ is negligible.*

Proposition 5. *For a fixed $m \in \mathbb{F}_{N,p_1}$, elements of the set $\{\mathtt{Enc}_{pk,evk}(m)\}$ are indistinguishable from random elements of $[0, g) \cap \mathbb{Z}$ under Assumption 2.*

Proof. Let $m \in \mathbb{F}_{N,p_1}$, $\mathtt{Enc}_{pk,evk}(m) = \mathtt{Enc}_{pk,evk}(m + 0) = \mathtt{Enc}_{pk,evk}(m) + \mathtt{Enc}_{pk,evk}(0)$. By Assumption 2, encryptions of 0 are indistinguishable from random elements of $[0, g) \cap \mathbb{Z}$, whence $\mathtt{Enc}_{pk,evk}(m) + \mathtt{Enc}_{pk,evk}(0) = \mathtt{Enc}_{pk,evk}(m)$ is indistinguishable from a random element of $[0, g) \cap \mathbb{Z}$.

6.3 Factoring

The most obvious threat to our construction is also the easiest to thwart. It is associated with factoring attacks since the public evaluation key evk corresponds to g, which is the product of $p_1, \dots, p_4$. Successfully factoring g leads to a total break of the scheme since decryption only uses the knowledge of p_1 and p_3 according to (5). Even a partial factorization of g might lead to a total break of our scheme, as long as p_1 and p_3 are recovered. The main threat could be provided by some variation ECM factoring method [3, 46] (since p_4 in our scheme is not prime) with running time on the size of the smallest prime factor as opposed of the size of g. To prevent ECM threats, one must set the size of individual primes to be at least 512 bits and preferably at least 768 bits. Additionally, if g is sufficiently large (e.g., greater than or equal to 4096 bits), index calculus methods such as the Number Field Sieve method [45] will not succeed.

6.4 Hensel Code Problem

We close this section with a proof that an adversary, knowing only $g = p_1 \cdots p_4$, $g' = p_3 p_4$, and $c = H_g\big(H_{g'}\big(x/y\big) + \delta g'\big)$, cannot deduce x/y using $\widetilde{H}_g^{-1}$ or $\widetilde{H}_{g'}^{-1}$. Furthermore, the range of the "noise parameter" δ can be restricted to guarantee that an adversary cannot even deduce the denominator y (a problem we noticed in some simulations).

Proposition 6. *If $\alpha = g'$ or g, then $\widetilde{H}_\alpha^{-1}\big(H_{g'}(se + m)\big) \neq m$.*

Proof. Suppose by way of contradiction that $H'_g(se + m) = H_\alpha(m)$.
Let $\alpha = g'$.
If we let $m = x/y$, then we get $H_{g'}\Big((sey + x)/y\Big) = H_{g'}\big(x/y\big)$. Since $p_1 \ll p_3, p_4$, y is invertible modulo g', whence $(sey + x)y^{-1} = xy^{-1} \pmod{g'}$. It follows that $se = 0 \pmod{g'}$. Since $s < p_1$, we also have s invertible modulo g', which means $e = 0 \pmod{g'}$. In particular, we note that $e = 0 \pmod{p_4}$. But, since $e = H^{-1}_{p_1,p_2}(0, s_2) + \delta_e p_4$, this implies $H^{-1}_{p_1,p_2}(0, s_2) = 0 \pmod{p_4}$. Put $H^{-1}_{p_1,p_2}(0, s_2) = x_0/y_0$. Since $s_2 \neq 0$, $x_0 \neq 0$. Moreover, since the inverse of y_0 modulo p_4 cannot be divisible by p_4, we conclude that $x_0 = 0 \pmod{p_4}$. This contradicts $0 < |x_0| \leq \big\lfloor \sqrt{(p_1 p_2 - 1)/2} \big\rfloor < p_4$.
Let $\alpha = g$.
Since $g'|g$, $H_{g'}\Big((sey + x)/y\Big) = H_g\big(x/y\big)$ implies $H_{g'}\Big((sey + x)/y\Big) = H_{g'}\big(x/y\big)$. The result follows. □

Lemma 8. *Suppose $\widetilde{H}_\alpha^{-1}\Big(H_{g'}\Big(x/y\Big)\Big) = x'/y'$, where $\alpha = g$ or g'. If $x \neq x'$, then $y \neq y'$ or $|x - x'| \geq g'$.*

Proof. Suppose $\alpha = g$. Then $H_{g'}\big(x/y\big) = H_g\big(x'/y'\big)$. We will prove the contrapositive. If $y = y'$ and $|x - x'| < g'$, then we use the fact that $g'|g$ to obtain $|x - x'|_{g'} = 0$. Since $|x - x'|$ is less than g', $x = x'$. The proof for $\alpha = g'$ is analogous.

Proposition 7. *Let $x/y \in \mathbb{F}_{N,p_1}$, $N = \left\lfloor\sqrt{(g'-1)/2}\right\rfloor$, and $\alpha = g$ or g'. If*

$$s \in \left(\frac{N+|x|}{e|y|}, \frac{g'-(N+|x|)}{e|y|}\right), \text{ and } \widetilde{H}_{\alpha}^{-1}\left(H_{g'}\left(se + \frac{x}{y}\right)\right) = \frac{x'}{y'}, \tag{8}$$

then $y \neq y'$.

Proof. In light of Lemma 8, it suffices to prove that $x + sey \neq x'$ and $|(x + se \cdot y) - x'| < g'$.
If $s > (N + |x|)/(e|y|)$, then

$$|x + sey| \geq \big||x| - se|y|\big| > \big||x| - (N + |x|)\big| = N.$$

Since $|x'| \leq N$, by definition of $\widetilde{H}_g^{-1}$ (MEEA, in particular), we see that $x + \delta p_4 y \neq x'$.
Similarly, if $0 < s < (g' - N - |x|)/(e|y|)$, then

$$|(x + sey) - x'| \leq |x| + se|y| + N < |x| + (g' - N - |x|) + N = g'.$$

This completes the proof. □

7 Practical Considerations

The security parameters of our construction are defined in observance of the results in [11, 60]. Among several alternative concrete parameter configurations we considered, the one we discuss in this work is not the most efficient in terms of ciphertext expansion. However, it is an instance that works as desired with respect to the homomorphic operations. As mentioned before, the noise growth on addition is still a concern. In our scheme, messages are members of a subset of the rational numbers so if we let two messages x_1/y_1 and x_2/y_2, be encrypted to two ciphertexts c_1 and c_2, then $c_1 \cdot c_2$ and $c_1 + c_2$ will decrypt, respectively, to

$$\frac{x_1 x_2}{y_1 y_2} \text{ and } \frac{x_1 y_2 + x_2 y_1}{y_1 y_2}. \tag{9}$$

We note that the space taken up by addition is similar to the space taken up by multiplication (in the sense that the scheme admits a similar number of additions and multiplications), and acknowledge that this might be a downside for some applications.

Advantages of Working with Hensel Codes Although there are some limiting aspects, we want to emphasize some of the advantages in working with Hensel codes. Recall that $\mathcal{P} = \mathbb{F}_{N,p_1}$. Considering that p_1 must be at least a 768-bit prime, its corresponding N is a 384-bit number and thus the message space will be sufficiently large to include the solutions for most applications that require rational numbers as inputs. For instance, the set of rational numbers $\mathbb{F}_{N,p_1}$ includes integers (negative and positive) up to 384 bits. This is a message space large enough to contemplate a large class of real-world applications. Obviously, the larger p_1 is, the larger will be the message space. Perhaps one of the greatest takeaways is that *we can merge error-free computation with homomorphic encryption* via Hensel codes. As an immediate consequence, we can naturally compute the arithmetic gates addition, subtraction, multiplication, and division as follows: $|c_1 + c_2|_g$, $|c_1 - c_2|_g$, $|c_1 c_2|_g$, and $\left|c_1 / c_2\right|_g$, respectively. Correctness follows the discussion in Section 5.1. It is clear that addition, subtraction, and multiplication are computed in the natural way modulo g. For all c_1, c_2 output by `Enc` or `Eval`, the division $\left|c_1 / c_2\right|_g$, will work as long as $\gcd(c_2, g) = 1$. One simple way to ensure that division is always defined is to allow an encryption of zero to be public and implement the following division algorithm: Given c_1, c_2, g and a public encryption of zero c_z, generate a uniform $r_c \leftarrow \mathbb{Z}_{2^{\lambda-1}}$ and update c_2 as $c_2 = r_c c_z + c_2$. If $\gcd(c_2, g) = 1$, compute and output $\left|c_1 / c_2\right|_g$, if not, repeat.

Given any homomorphic encryption that takes positive integers as valid messages, it is not surprising that one could easily provide a way for allowing that scheme to accept rational numbers as inputs. Given a rational number a/b (assumed to be positive, for simplicity), options include: 1) a simple modular encoding with a modulus $q > a, b$ such that you have $m = aq + b$, 2) CRT with two moduli $q_1, q_2 > a, b$ such that $m = \mathrm{CRT}_{q_1, q_2}(a, b)$, or yet 3) any pairing function, such as the Cantor pairing function [33], in which case we could obtain $m = 1/2\,(a + b)(a + b + 1) + b$. None of the above options, along with many other numeric manipulations, preserve operations in the message space. Thus, since Hensel codes create an operation-preserving correspondence between a set of rationals and a set of integers, they are preferable as a tool for modifying existing schemes to take rational inputs.

7.1 Performance

The results presented in Table 1 were generated from experiments conducted with an implementation of our candidate scheme using Python 3.8.5, on a MacBook Pro 15-inch, MacOS High Sierra 10.13.6, 2.8 GHz Intel Core i7, 16 GB 1600 MHz DDR3, 500GB HD. Each runtime (in seconds) presented is the arithmetic mean of 100 runs.

We present practical results using two configurations. First, we set $\rho = \lambda = 512$, $d = 10$, which gives $\eta = 5120$, $\mu = 454653$, and $\gamma = 461571$, $|p_1|_{\text{bits}} = 512$, $|p_2|_{\text{bits}} = 515$, $|p_3|_{\text{bits}} = 5120$, $|p_4|_{\text{bits}} = 455672$, $|g'|_{\text{bits}} = 460791$, and $|g|_{\text{bits}} = 461818$ (slightly larger than γ). Second, we set $\rho = \lambda = 768$, $d = 20$, which gives $\eta = 15360$, $\mu = 2927601$, and $\gamma = 2950275$, $|p_1|_{\text{bits}} = 768$, $|p_2|_{\text{bits}} = 771$, $|p_3|_{\text{bits}} = 15360$, $|p_4|_{\text{bits}} = 2933708$, $|g'|_{\text{bits}} = 2949067$, and $|g|_{\text{bits}} = 2950606$. In Table 1 we display runtime results for the key generation, encryption, and decryption algorithms, and the homomorphic evaluation of the dot product of two 3D vectors.

Algorithm	Runtime for $\lambda = 512, d = 10$	Runtime $\lambda = 768, d = 20$
Key Generation	184.244738	927.6136270000001
Encryption	0.9383330000000001	35.50052400000004
Decryption	0.006788999999977135	0.09174399999994876
3D vector dot product	0.3805450000000121	12.849364999999807

Table 1. Runtime results.

References

1. Alparslan, E.: Finite p-adic number systems with possible applications. Ph.D. thesis, Ph. D. Dissertation. Department of Electrical Engineering, University of ... (1975)
2. Barillas, B.S.S.: Efficient Machine Learning Inference for Embedded Systems with Integer Based Restricted Boltzmann Machines Classifiers. Ph.D. thesis, University of Colorado Colorado Springs (2019)
3. Beullens, W., Smart, N., Vercauteren, F.: Security evaluation of x-logos factoring-based, private key fhe. personal communication (2020), on 2020-07-13
4. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Theory of cryptography conference. pp. 325–341. Springer (2005)
5. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 1–36 (2014)
6. Chen, H., Laine, K., Player, R., Xia, Y.: High-precision arithmetic in homomorphic encryption. In: Cryptographers' Track at the RSA Conference. pp. 116–136. Springer (2018)
7. Chen, Y., Nguyen, P.Q.: Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 502–519. Springer (2012)
8. Cheon, J.H., Coron, J.S., Kim, J., Lee, M.S., Lepoint, T., Tibouchi, M., Yun, A.: Batch fully homomorphic encryption over the integers. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 315–335. Springer (2013)
9. Cheon, J.H., Kim, J., Lee, M.S., Yun, A.: Crt-based fully homomorphic encryption over the integers. Information Sciences **310**, 149–162 (2015)
10. Cheon, J.H., Stehlé, D.: Fully homomophic encryption over the integers revisited. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 513–536. Springer (2015)
11. Cheon, J.H., Stehle, D.: Fully homomorphic encryption over the integers revisited. Cryptology ePrint Archive, Report 2016/837 (2016), https://eprint.iacr.org/2016/837
12. Chung, H., Kim, M.: Encoding rational numbers for fhe-based applications. IACR Cryptol. ePrint Arch. **2016**, 344 (2016)
13. Cohn, H., Heninger, N.: Approximate common divisors via lattices. The Open Book Series **1**(1), 271–293 (2013)
14. Coron, J.S., Lepoint, T., Tibouchi, M.: Scale-invariant fully homomorphic encryption over the integers. In: International Workshop on Public Key Cryptography. pp. 311–328. Springer (2014)

15. Coron, J.S., Mandal, A., Naccache, D., Tibouchi, M.: Fully homomorphic encryption over the integers with shorter public keys. In: Annual Cryptology Conference. pp. 487–504. Springer (2011)
16. Coron, J.S., Naccache, D., Tibouchi, M.: Public key compression and modulus switching for fully homomorphic encryption over the integers. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 446–464. Springer (2012)
17. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory **31**(4), 469–472 (1985)
18. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptol. ePrint Arch. **2012**, 144 (2012)
19. Fitzpatrick, P., Flynn, J.: A gröbner basis technique for padé approximation. Journal of Symbolic Computation **13**(2), 133–138 (1992)
20. Froment, A.: Error free computation: a direct method to convert finite-segment p-adic numbers into rational numbers. IEEE Computer Architecture Letters **32**(04), 337–343 (1983)
21. Galbraith, S.D., Gebregiyorgis, S.W., Murphy, S.: Algorithms for the approximate common divisor problem. LMS Journal of Computation and Mathematics **19**(A), 58–72 (2016)
22. Gentry, C.: A fully homomorphic encryption scheme. Stanford university (2009)
23. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 169–178 (2009)
24. Goldwasser, S., Micali, S.: Probabilistic encryption &amp; how to play mental poker keeping secret all partial information. In: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing. pp. 365–377 (1982)
25. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of computer and system sciences **28**(2), 270–299 (1984)
26. Gouvêa, F.Q.: p-adic numbers. In: p-adic Numbers, pp. 43–85. Springer (1997)
27. Gregory, R.T.: Error-free computation with finite number systems. ACM SIGNUM Newsletter **14**(3), 9–16 (1979)
28. Gregory, R.T.: Error-free computation: why it is needed and methods for doing it. RE Krieger (1980)
29. Gregory, R.: Error-free computation with rational numbers. BIT Numerical Mathematics **21**(2), 194–202 (1981)
30. Hehner, E.C.R., Horspool, R.: A new representation of the rational numbers for fast easy arithmetic. SIAM Journal on Computing **8**(2), 124–134 (1979)
31. Hensel, K.: Theorie der algebraischen Zahlen, vol. 1. BG Teubner (1908)
32. Hoffstein, J., Silverman, J.: Optimizations for ntru. public-key cryptography and computational number theory (2002)
33. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to automata theory, languages, and computation. Acm Sigact News **32**(1), 60–65 (2001)
34. Howgrave-Graham, N.: Approximate integer common divisors. In: International Cryptography and Lattices Conference. pp. 51–66. Springer (2001)
35. Khrennikov, A.Y., Nilsson, M.: P-adic deterministic and random dynamics, vol. 574. Springer Science & Business Media (2013)
36. Koç, Ç.K.: Parallel p-adic method for solving linear systems of equations. Parallel Computing **23**(13), 2067–2074 (1997)
37. Krasner, M.: Prolongement analytique uniforme et multiforme dans les corps values complets: preservation de l'analycite par la convergence uniforme et par la

derivation; theoreme de mittag-leffler generalise pour les elements analytiques'. CR Acad. Sci. Paris **244**, 2570–2573 (1957)

38. Krasner, M.: Nombres semi-réels et espaces ultramétriques. Comptes-Rendus de l'Académie des Sciences **2**, 219 (1944)
39. Krishnamurthy, E.V.: Error-free polynomial matrix computations. Springer Science & Business Media (2012)
40. Krishnamurthy, E., Murthy, V.K.: Fast iterative division of p-adic numbers. IEEE transactions on computers **32**(04), 396–398 (1983)
41. Krishnamurthy, E., Rao, T.M., Subramanian, K.: Finite segment p-adic number systems with applications to exact computation. In: Proceedings of the Indian Academy of Sciences-Section A. vol. 81, pp. 58–79. Springer (1975)
42. Kurt, M.: Introduction to p-adic numbers and their functions (1981)
43. Lagarias, J.C.: The computational complexity of simultaneous diophantine approximation problems. SIAM Journal on Computing **14**(1), 196–209 (1985)
44. Laine, K., Player, R.L., Chen, H.: Rational number arithmetic in homomorphic encryption (Jun 25 2019), uS Patent 10,333,695
45. LENSTRA, A.K.: The development of the number field sieve. Lecture Notes in Mathematics (LNM) (1993)
46. Lenstra Jr, H.W.: Factoring integers with elliptic curves. Annals of mathematics pp. 649–673 (1987)
47. Li, X., Lu, C., Sjogren, J.A.: A method for hensel code overflow detection. ACM SIGAPP Applied Computing Review **12**(1), 6–11 (2012)
48. Miola, A.: Algebraic approach to p-adic conversion of rational numbers. Information Processing Letters **18**(3), 167–171 (1984)
49. Morrison, J.F.: Parallel p-adic computation. Information processing letters **28**(3), 137–140 (1988)
50. Motzkin, E., Robba, P.: Prolongement analytique en analyse *p*-adique. Séminaire de théorie des nombres de Bordeaux pp. 1–47 (1968)
51. Murthy, V.: Exact parallel matrix inversion using para-hensel codes with systolic processors. Applied optics **27**(10), 2022–2024 (1988)
52. Nguyen, P.Q., Stern, J.: The two faces of lattices in cryptology. In: International Cryptography and Lattices Conference. pp. 146–180. Springer (2001)
53. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International conference on the theory and applications of cryptographic techniques. pp. 223–238. Springer (1999)
54. Rao, T.M., Gregory, R.T.: The conversion of hensel codes to rational numbers. In: 1981 IEEE 5th Symposium on Computer Arithmetic (ARITH). pp. 10–20. IEEE (1981)
55. Rao, T.M., Subramanian, K., Krishnamurthy, E.: Residue arithmetic algorithms for exact computation of g-inverses of matrices. SIAM Journal on Numerical Analysis **13**(2), 155–171 (1976)
56. Rivest, R.L., Adleman, L., Dertouzos, M.L., et al.: On data banks and privacy homomorphisms. Foundations of secure computation **4**(11), 169–180 (1978)
57. Robba, P.: Fonctions analytiques sur les corps valués ultramétriques complets. Prolongement analytique et algèbres de Banach ultramétriques (1973)
58. Rothblum, R.: Homomorphic encryption: From private-key to public-key. In: Theory of cryptography conference. pp. 219–234. Springer (2011)
59. Shoup, V.: A computational introduction to number theory and algebra. Cambridge university press (2009)

60. Van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 24–43. Springer (2010)