Journal of *Cybersecurity and Privacy*

MDPI

Article

D-HPPK KEM: A Defactorized Homomorphic Polynomial Public Key Encapsulation Mechanism for IND-CCA2 Security

Randy Kuang

Quantropi Inc., 1545 Carling Avenue, Suite 620, Ottawa, ON K1Z 8P9, Canada; randy.kuang@quantropi.com

Abstract

We present the **Defactorized Homomorphic Polynomial Public Key (D-HPPK)** Encapsulation Mechanism, a post-quantum KEM whose security rests on exhaustive-search problems with efficient verification for both attack paths. We adopt the NIST search-based convention, under which a primitive is assigned security level l when the adversary's best-known attack requires exhaustive search over 2^l candidates with efficient verification, exactly as AES-128, AES-192, and AES-256 are assigned NIST Levels I, III, and V. For D-HPPK, both attack paths—long-term key recovery and ephemeral-secret recovery—are exhaustive searches over spaces of size 2^l with polynomial-time verification. The long-term key-recovery path reduces to the Hidden Modulus Product Problem (HMPP), and the ephemeral-secret recovery path reduces to the scheme-induced Message Recovery Problem (MRP). The scheme uses a three-layer architecture of additive random masking and hidden-ring embeddings. The hidden-ring moduli and multipliers block the public-constraint attack; the random masking destroys the rank-2 algebraic regularity of the unmasked coefficient matrices, forcing the adversary into HMPP. Both attack paths have best-known classical cost $O(2^l \cdot \text{poly}(L))$. The IND-CCA2 proof uses a double-encryption consistency check with a five-class rejection set, together with a small number of explicitly stated hardness assumptions; the adaptive form of the decryption-collision property is derived through Structured D-HMPP. A unified parameter set $(n = 2, m = 3, l = \lceil \log_2 p \rceil)$ supports a single implementation across security levels. At Level V ($l = 256$), D-HPPK achieves a 1234-byte public key and a 392-byte CCA2 ciphertext—21% smaller public keys and 75% smaller ciphertexts than ML-KEM-1024—without SIMD or NTT acceleration.

Keywords: post-quantum cryptography; key encapsulation mechanism; polynomial-based cryptography; defactorization; hidden rings; random masking; Hidden Modulus Product Problem; Decisional HMPP; Message Recovery Problem; pairwise decryption collision resistance; IND-CCA2 security; search-based security

Check for updates

Academic Editor: Nikos Fotiou

Received: 18 August 2026
Revised: 16 September 2026
Accepted: 2 October 2026
Published: 8 October 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The *Homomorphic Polynomial Public Key (HPPK)* cryptosystem, introduced by Kuang and Perepechaenko in 2023 [1], evolved from the earlier Multivariate Polynomial Public Key (MPPK) framework [2]. Its construction relies on two private univariate polynomials $f(x), h(x) \in \mathbb{F}_p[x]$, which generate product polynomials

$$p(x, \mathbf{u}) = f(x)B(x, \mathbf{u}), \qquad q(x, \mathbf{u}) = h(x)B(x, \mathbf{u}),$$

where $B(x, \mathbf{u})$ is an ephemeral multivariate polynomial. The system preserves the homomorphic ratio

$$\frac{f(x)}{h(x)} \equiv \frac{p(x, \mathbf{u})}{q(x, \mathbf{u})} \pmod{p}, \quad (1)$$

with security originally derived from arithmetic permutations of coefficient vectors induced by two large hidden rings.

1.1. Motivation and the D-HPPK Approach

The history of multivariate cryptography is marked by a recurring pattern: elegant constructions are proposed, later found to harbor algebraic regularities, and then patched or replaced [3–6]. Rainbow, for instance, advanced to the NIST PQC finalists before a devastating attack exploited its hidden field structure [7,8].

The original HPPK construction offered compact keys and efficient operations, leveraging hidden-ring permutations to conceal its polynomial factorization. However, Johnson [9] has shown that the hidden-ring embedding itself is vulnerable to a polynomial-time key-recovery attack, the *Lattice Reconstitution Attack* (LRA). The public key has the form $\hat{P}_{ij} \equiv P_{ij} \cdot r \pmod{t}$, where a single scalar multiplier r is applied to all coefficients and t is a hidden modulus larger than the coefficients. An attacker can build a discriminator matrix from Bézout coefficients of pairs of public coefficients; row reduction and LLL lattice reduction then recover the hidden modulus t and hence the full secret key in $O((\log p)^3)$ classical time and $O(\log p)$ memory, with deterministic success. The attack exploits the combined structure of the single scalar multiplier and the low-rank factorization of the unmasked coefficient matrices, rather than the rank of the coefficients alone. The LRA invalidates the security assumptions of the original HPPK KEM.

To defend against the LRA and the algebraic-regularity attack more generally, we introduce the **Defactorized HPPK (D-HPPK) KEM**, which combines additive random masking with hidden-ring embeddings in a three-layer conjugated architecture. The random masking is applied *before* embedding into the hidden rings. The published coefficients become $\hat{P}_{ij} \equiv (P^{(0)}_{ij} + c^{(p)}_i d_j \pmod{p}) R_1 \pmod{S_1}$, where the rank-one perturbation $c^{(p)}_i d_j$ destroys the rank-two structure of $P^{(0)}$ and removes the low-dimensional structure on which the LRA depends. A linear constraint $\delta(\mathbf{u}) = \mathbf{d}^\top \mathbf{u} = 0$ ensures that the masking cancels exactly during decryption, restoring the homomorphic ratio invariant. Critically, this cancellation is enforced *inside* the hidden rings, so the adversary cannot evaluate the condition without knowledge of the secret moduli. The result is a pair of complementary mechanisms: the hidden rings block the public-constraint attack, and the random masking blocks the algebraic-regularity attack. An adversary is forced into the generic HMPP problem analyzed in Section 4.5, and the decryption-safety properties required for the CCA reduction are established in Section 4.4.

We emphasize that the masking provides a *single* barrier against the algebraic-regularity attack, not a second exponential barrier after HMPP. Once the hidden rings are broken, the factorization recovery is polynomial in L. This drives the equal-cost structure discussed next.

1.2. Security Model and Search-Based Framework

D-HPPK grounds security in five hardness assumptions, stated formally in Section 4.3.

Long-term key recovery. Reduces to the *Hidden Modulus Product Problem (HMPP)*: recover the hidden modulus S and multiplier R from the public samples $A_i \equiv R\tilde{a}_i \pmod{S}$. Once HMPP is broken, the factorization structure of the masked coefficient matrices is recoverable in polynomial time via projection onto the public subspace $\mathbf{d}^\perp$. Best-known classical cost: $O(2^l \cdot \mathrm{poly}(L))$.

Ephemeral secret recovery. Is the *Message Recovery Problem (MRP)*, in two forms. The single-ciphertext form MRP_1 defines the hardness of the base PKE: given **PK** and $C = \mathrm{Enc}(\mathbf{PK}, x; \mathbf{u})$, recover x. The double-ciphertext form MRP_2 matches the KEM challenge: given **PK** and $(C^{(1)}, C^{(2)})$ both encrypting the same x under independent randomness, recover x. Best-known attack on both: $O(2^l \cdot L^2)$.

Decryption safety. Is the scheme-induced *pairwise decryption-collision resistance* assumption: for every fixed ciphertext pair $C^{(1)} \neq C^{(2)}$ outside the rejection set, the two decryptions collide with probability at most $1/p + \mathrm{negl}(\lambda)$ over the secret key. This is a companion to MRP_2: where MRP_2 excludes recovery of x^* from the challenge pair, the collision assumption excludes oracle leakage through malformed queries. Marginal uniformity of each decryption output is proved from the hidden-ring modular reduction (Lemmas 1 and 2); the joint near-independence of two outputs is assumed and is the genuinely two-dimensional statement not proved here (Assumption 5). The adaptive version is derived through Structured D-HMPP (Lemma 3).

Search-based security convention. We adopt the same framework NIST uses for AES: a primitive is assigned security level l when the best-known attack requires exhaustive search over a space of size 2^l with efficient verification, and no structural attack reduces the search space. This is a definitional level assignment, not a theorem. The same convention applies to D-HPPK: both attack paths are exhaustive searches over 2^l candidates with polynomial-time verification, giving security level l and NIST categories $l = 128, 192, 256$ for Levels I, III, and V. D-HPPK is therefore a search-based cryptosystem in the sense of Merkle's puzzles [10] and AES. The cryptanalysis in Section 4 supports the level assignment by surveying known attack families on HMPP and MRP and finding none that reduces the search space.

1.3. Contributions

- **Two-barrier security architecture.** We separate the public-constraint attack (blocked by hidden rings) from the algebraic-regularity attack (blocked by masking), and show that neither mechanism subsumes the other.
- **Key-recovery analysis.** Stage 2 of the key-recovery attack is *polynomial-time* after HMPP, giving a best-known classical cost of $O(2^l \cdot \mathrm{poly}(L))$, matching the ephemeral-secret recovery cost.
- **Five hardness assumptions with a reduced decryption-safety property.** Marginal statistical uniformity of the decryption output is proved from the hidden-ring modular reduction; pairwise collision resistance is stated as a scheme-induced assumption with a companion role to MRP_2, and its adaptive form is derived through Structured D-HMPP. The IND-CCA2 proof rests on the conjunction of four of the five assumptions.
- **IND-CCA2 security via double encryption with a strengthened consistency check.** A union bound over both attack paths, with a five-class rejection set for correlated adversarial ciphertexts, a simulation argument justifying the game-level restriction, and the random-oracle advantage bound quantified.
- **Compact parameterization and algorithmic simplicity.** Unified parameter set $(n = 2, m = 3, l = \lceil \log_2 p \rceil)$ across security levels. At $l = 256$, D-HPPK achieves a 1234-byte public key and a 392-byte CCA2 ciphertext—21% and 75% smaller than ML-KEM-1024—without SIMD or NTT acceleration.

Scope and limitations. The security analysis is in the classical random oracle model; a quantum random-oracle proof is left for future work. The five assumptions are new problems or scheme-induced assumptions; the analysis reports best-known attacks and invites further cryptanalysis. The current reference implementation is not constant-time. These limitations are addressed explicitly in Sections 4 and 5.

2. Related Work

Public-key cryptography has historically relied on the hardness of integer factorization, discrete logarithms, and elliptic-curve discrete logarithms. Shor's quantum algorithm [11] breaks these assumptions, prompting the development of post-quantum cryptography (PQC). NIST initiated a multi-round standardization process [12] to select quantum-resistant algorithms for key encapsulation and digital signatures.

2.1. NIST Standardization and PQC Families

The first set of NIST standards, finalized in 2024, includes lattice-based ML-KEM [13] for key encapsulation, ML-DSA and FN-DSA [14,15] for signatures, and hash-based SLH-DSA [16,17]. Classic McEliece [18,19] continues as a fourth-round code-based candidate with large public keys, and NTRU Prime [20] provides an alternative lattice-based KEM. Earlier multivariate (Rainbow [7]) and isogeny-based (SIKE [21]) submissions were later broken [8,22], underscoring the community's effort to diversify hardness assumptions.

Lattice-based schemes operate in polynomial rings $\mathbb{Z}_q[x]/(x^n+1)$ using the Module-LWE problem and benefit from strong reductions to worst-case lattice problems, but face large key and ciphertext sizes (ML-KEM-1024: 1568-byte public key and ciphertext), reliance on NTT accelerators, and side-channel susceptibility. Code-based cryptography, exemplified by Classic McEliece, offers conservative security but requires public keys in the hundreds of kilobytes. The two families rest on different hardness assumptions and provide complementary foundations.

2.2. Multivariate Polynomial Cryptography

Multivariate polynomial cryptography provides an algebraic framework with security assumptions rooted in the hardness of solving nonlinear multivariate equations over finite fields. Traditional multivariate schemes [6] have historically found more success in signature constructions—exemplified by Rainbow [7], though recently broken [8]—than in practical key encapsulation mechanisms, due to large key sizes, computational cost, and the scarcity of IND-CCA2-secure designs that avoid complex transforms [6].

The security of multivariate schemes has been repeatedly challenged by algebraic cryptanalysis, and this history informs the design choices in D-HPPK. The additive random masking is specifically designed to destroy the algebraic regularities that have proven exploitable in other multivariate schemes, and the security proof in Section 4 explicitly analyzes the Gröbner basis attack surface of the resulting system.

The MMPS problem introduced in Section 4.6.1 belongs to the well-studied class of structured polynomial systems over finite fields. Systems that are linear in one set of variables and polynomial in another have been analyzed in the algebraic cryptanalysis literature. Faugère, Levy-dit-Vehel, and Perret studied the MinRank problem and its bilinear structure, showing that eliminating the linear variables yields a low-degree univariate relation [23]. Faugère's broader survey of structured polynomial systems situates these techniques within computational algebra [24], and Petit and Quisquater analyzed polynomial systems arising from Weil descent, which share the mixed linear-polynomial structure [25]. For the hidden-modulus aspect, the black-box field problem of Boneh and Lipton [26] and Coppersmith's algorithms for small solutions of modular polynomial equations [27] provide the closest precedents. The MMPS problem differs from all of these in that its coefficients are generated by the D-HPPK key generation and encryption process, rather than being drawn from an arbitrary or uniform distribution.

Finally, we note that alternative algebraic foundations for post-quantum cryptography have been explored outside the polynomial and lattice families. Selim, Saračević, and Ćatović recently proposed a homomorphic cryptographic scheme based on nilpotent Lie

algebras [28], using a different hardness foundation from the polynomial homomorphism and hidden-ring approach of D-HPPK.

2.3. KEM Transformations and Hidden-Modulus Problems

The standard method for converting a passively secure public-key encryption scheme into an IND-CCA2 secure KEM is the Fujisaki-Okamoto (FO) transform [29] and its modular reanalyses [30], widely used in ML-KEM [13] and extended to the quantum random oracle model [30,31]. Closely related are plaintext-aware and robust encryption notions [32,33], and generic KEM constructions [34]. The double-encryption construction used in D-HPPK belongs to this family. Its security relies on MRP_2 plus a pairwise decryption-collision assumption, rather than on the FO re-encryption check.

The HMPP introduced in this work is related in spirit to the Approximate Common Divisor (ACD) problem [35–38], but differs in structure. In ACD, the adversary receives samples $x_i = pq_i + r_i$ with a hidden large divisor p and small errors r_i, and recovers p by exploiting the small residues $x_i \bmod p$. In HMPP, the hidden quantities are the *modulus* S itself and a multiplier $R \in \mathbb{Z}_S^*$, and the public samples are $A_i \equiv R\tilde{a}_i \pmod{S}$ with small values $\tilde{a}_i \in [0, p-1]$. The samples A_i are uniform residues modulo S: no small-residue structure is visible until S is known. The standard ACD lattice template, which relies on small residues modulo the unknown divisor, does not transfer directly to HMPP; whether an adapted construction can recover S from a single guess of $\tilde{a}_1$ is examined in Section 4.5.

2.4. Search-Based Cryptography

Search-based cryptography reduces security to the size of a search space and the efficiency of verification, following the convention NIST uses for AES. For unstructured search problems, Grover's algorithm [39] provides a quadratic speedup, unlike Shor's algorithm [11] for factoring and discrete logarithms. This makes search-based assumptions a conservative foundation for post-quantum security. Traditional constructions such as Merkle's puzzles [10] have suffered from impractical key sizes; efficient CCA-secure extensions remain rare. D-HPPK embeds a structured algebraic layer—the homomorphic ratio invariant and masked polynomial representation—that aims to preserve search-based hardness while enabling compact keys, and uses a double-encryption consistency check to achieve IND-CCA2 security.

3. The Defactorized HPPK KEM Scheme

This section presents the complete construction of the Defactorized Homomorphic Polynomial Public Key (D-HPPK) Key Encapsulation Mechanism. The scheme employs a layered security architecture that combines additive random masking (over $\mathbb{F}_p$) with hidden ring embeddings.

We present the core mathematical objects and key generation (Sections 3.1 and 3.2), then define a base Public Key Encryption (PKE) scheme (Section 3.3), and finally obtain an IND-CCA2-secure KEM via a double-encryption construction with a consistency check (Section 3.4).

3.1. Core Mathematical Objects and Notation

All operations are defined over a large prime field $\mathbb{F}_p$ with prime modulus p of bit-length l, i.e., $2^{l-1} \leq p < 2^l$, or equivalently $l = \lceil \log_2 p \rceil$. Where only leading-order behavior matters, we write l and $\log_2 p$ interchangeably, since the two differ by at most 1.

Two secret hidden-ring moduli S_1, S_2 of bit-length $L = 2l + 8$ and two secret multipliers R_1, R_2 are used. The moduli satisfy $S_1, S_2 > 9p^2$, and the multipliers are sampled uniformly from $\mathbb{Z}_{S_1}^*$ and $\mathbb{Z}_{S_2}^*$. The bound $S_i > 9p^2$ is required for correct decryption (Section 3.3), and the choice $L = 2l + 8$ satisfies it for all relevant l, since

$2^{2l+8} = 256 \cdot 2^{2l} > 9p^2$ whenever $l \geq 1$. The moduli may be prime or composite; prime moduli simplify implementation, while composite moduli give faster key generation. The inverses $R_1^{-1} \bmod S_1$ and $R_2^{-1} \bmod S_2$ are computed at key generation.

The following objects form the mathematical foundation:

- **Secret polynomials:** $f(x) = f_1 x + f_0 \in \mathbb{F}_p[x]$ and $h(x) = h_1 x + h_0 \in \mathbb{F}_p[x]$.
- **Ephemeral random polynomial:** A random multivariate polynomial

$$B(x, \mathbf{u}) = \sum_{j=1}^{m} \left(\sum_{i=0}^{n_b} b_{ij} x^i \right) u_j$$

  with coefficients $\{b_{ij}\} \xleftarrow{\$} \mathbb{F}_p$. It is multilinear in $\mathbf{u}$. For the default $n_b = 1$, it is bilinear in $(x, \mathbf{u})$.
- **Total degree:** $p(x, \mathbf{u}) = f(x)B(x, \mathbf{u})$ and $q(x, \mathbf{u}) = h(x)B(x, \mathbf{u})$ have degree $n = 1 + n_b$. We set $n_b = 1$, giving $n = 2$.
- **Core secret:** The rational function $\Phi(x) = f(x)/h(x) \pmod p$, invariant under all transformations.

Notation conventions throughout: elements in $\mathbb{F}_p$ are lowercase letters; matrices in $\mathbb{F}_p$ are bold lowercase; integer lifts to $[0, p-1]$ carry a tilde; elements in the hidden rings $\mathbb{Z}_{S_i}$ carry a hat; $(\cdot \bmod p)$ returns an integer in $[0, p-1]$ and $(\cdot \bmod S_i)$ returns an integer in $[0, S_i - 1]$.

3.2. Key Generation

The key generation algorithm KeyGen() produces a public key **PK** and a private key **SK** through three sequential transformations, as illustrated in Figure 1.

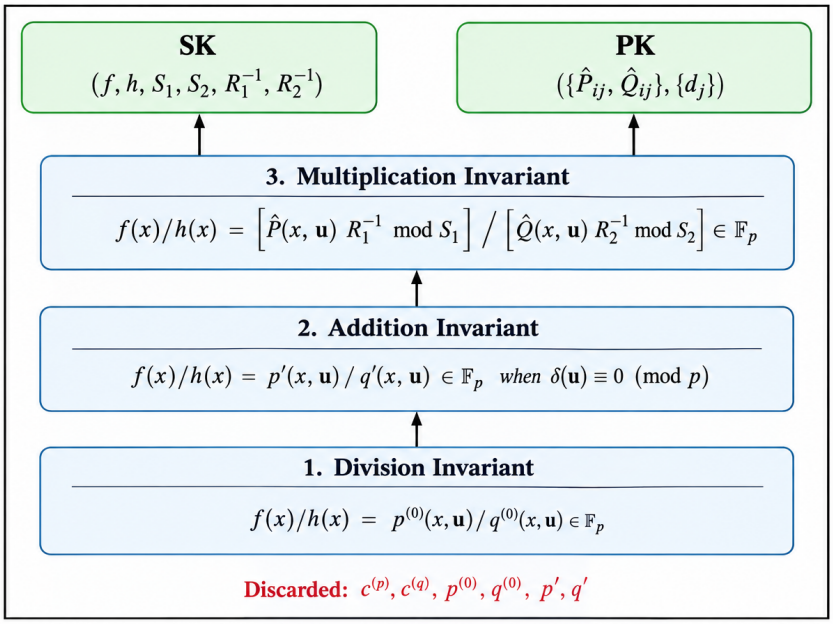


Figure 1. Three-layer key generation architecture for D-HPPK. The three sequential transformations (Division Invariant, Addition Invariant with random masking, Multiplication Invariant with hidden ring embedding) generate the public key **PK** and secret key **SK**. Intermediate values shown in the discarded block are permanently removed.

Transformation 1: Division Invariant (Embedding). Compute the product polynomials $p^{(0)}(x, \mathbf{u}) = f(x)B(x, \mathbf{u})$ and $q^{(0)}(x, \mathbf{u}) = h(x)B(x, \mathbf{u})$. Their coefficients form matrices $\mathbf{P}^{(0)}, \mathbf{Q}^{(0)} \in \mathbb{F}_p^{(n+1)\times m}$. By construction,

$$\Phi(x) \equiv \frac{p^{(0)}(x, \mathbf{u})}{q^{(0)}(x, \mathbf{u})} \pmod{p} \quad \text{for any } \mathbf{u}. \tag{2}$$

Transformation 2: Addition Invariant (Random Masking). Generate random ephemeral masking vectors $\mathbf{c}^{(p)}, \mathbf{c}^{(q)} \in \mathbb{F}_p^{n+1}$ and a public persistent constraint vector $\mathbf{d} \in \mathbb{F}_p^m$ with $d_m = 1$. Define rank-1 masking matrices $\mathbf{M}^{(p)} = \mathbf{c}^{(p)} \otimes \mathbf{d}$ and $\mathbf{M}^{(q)} = \mathbf{c}^{(q)} \otimes \mathbf{d}$, and compute

$$\mathbf{p}' = \mathbf{p}^{(0)} + \mathbf{M}^{(p)} \pmod{p}, \qquad \mathbf{q}' = \mathbf{q}^{(0)} + \mathbf{M}^{(q)} \pmod{p}. \tag{3}$$

Denote the elements of $\mathbf{p}'$ and $\mathbf{q}'$ as p'_{ij} and $q'_{ij} \in \mathbb{F}_p$. Define the constraint $\delta(\mathbf{u}) = \mathbf{d}^\top \mathbf{u}$. The masking terms cancel when $\delta(\mathbf{u}) \equiv 0 \pmod{p}$ because

$$\sum_{i=0}^{n} \sum_{j=1}^{m} c_i^{(p)} d_j x^i u_j = \Big(\sum_{i=0}^{n} c_i^{(p)} x^i \Big) \delta(\mathbf{u}). \tag{4}$$

Thus the invariant holds:

$$\Phi(x) \equiv \frac{p'(x, \mathbf{u})}{q'(x, \mathbf{u})} \pmod{p} \quad \text{when } \delta(\mathbf{u}) = 0 \pmod{p}, \tag{5}$$

where $p'(x, \mathbf{u}) = \sum_{i,j} p'_{ij} x^i u_j$ and $q'(x, \mathbf{u}) = \sum_{i,j} q'_{ij} x^i u_j$.

Transformation 3: Multiplication Invariant (Embedding into Hidden Rings). Embed the masked coefficients $p'_{ij}, q'_{ij} \in \mathbb{F}_p$ into the hidden rings by multiplying by the secret multipliers R_1, R_2 and reducing modulo the secret moduli S_1, S_2:

$$\hat{P}_{ij} \equiv p'_{ij} \cdot R_1 \pmod{S_1}, \qquad \hat{Q}_{ij} \equiv q'_{ij} \cdot R_2 \pmod{S_2}. \tag{6}$$

Key Pair. The following intermediate values are permanently discarded: the ephemeral masking vectors $\mathbf{c}^{(p)}, \mathbf{c}^{(q)}$; all basis polynomial coefficients $\{b_{ij}\}$; and the intermediate matrices $\mathbf{P}^{(0)}, \mathbf{Q}^{(0)}, \mathbf{p}', \mathbf{q}'$. Only the final values are retained:

$$\boxed{\begin{aligned} \mathbf{PK} &= \big(\{\hat{P}_{ij}, \hat{Q}_{ij}\}_{i=0,1,2;\, j=1,\ldots,m},\ \{d_j\}_{j=1,\ldots,m-1} \big), \\ \mathbf{SK} &= \big(f_0, f_1, h_0, h_1,\ S_1, S_2,\ R_1^{-1} \bmod S_1,\ R_2^{-1} \bmod S_2 \big). \end{aligned}} \tag{7}$$

The public key contains the masked coefficients as integers modulo the hidden moduli. The secret key holds the original polynomials, the hidden moduli, and the inverses of the multipliers.

3.3. Base Public Key Encryption Scheme

The base PKE scheme encrypts a message $x \in \mathbb{F}_p$ using a random noise vector $\mathbf{u}$ chosen from the kernel of the public constraint $\delta(\mathbf{u}) = 0$. The ciphertext consists of two integers c_p and c_q, computed as ordinary integer weighted sums of the public hidden-ring coefficients $\hat{P}_{ij}$ and $\hat{Q}_{ij}$, evaluated at the field values $x^i u_j$. The encrypter does **not** know the secret moduli S_1, S_2; thus the ciphertext is not reduced modulo these values and is typically much larger than S_1 or S_2. An adversary cannot simply reduce the ciphertext modulo p to recover the underlying field equation because the large S_1-scaled terms do not vanish without knowledge of the secret moduli, thereby blocking the fatal public-constraint attack.

Decryption recovers the masked field values by first reducing the ciphertext modulo S_1 and S_2, then unwrapping the hidden rings using the secret multipliers and moduli. The ratio of the recovered field values yields the invariant $\Phi(x) = f(x)/h(x) \pmod p$, from which x is uniquely recovered.

Encryption $\mathsf{Enc}(\mathbf{PK}, x; \mathbf{u})$**.** Given a message $x \in \mathbb{F}_p$ and randomness $\mathbf{u} = (u_1, \ldots, u_m) \in \mathbb{F}_p^m$ satisfying $\delta(\mathbf{u}) = 0$, compute the integer values

$$c_p = \sum_{i=0}^{n} \sum_{j=1}^{m} \hat{P}_{ij} \left(x^i \times u_j \pmod p \right),$$
$$c_q = \sum_{i=0}^{n} \sum_{j=1}^{m} \hat{Q}_{ij} \left(x^i \times u_j \pmod p \right), \tag{8}$$

where each product $(x^i \bmod p)(u_j \bmod p)$ is reduced modulo p to an integer in $[0, p-1]$, and the outer multiplication by $\hat{P}_{ij}$ is performed over the integers. Since $\hat{P}_{ij} \in [0, S_1 - 1]$ and each reduced product lies in $[0, p-1]$, every term of the sum lies in $[0, (S_1-1)(p-1)]$, and the ciphertext component satisfies

$$c_p \;\le\; (n+1)m\,(S_1-1)(p-1) \;<\; (n+1)m\,p\,S_1,$$

and similarly $c_q < (n+1)m\,p\,S_2$. For the default parameters $n = 2$, $m = 3$, this gives $c_p < 9pS_1$ and $c_q < 9pS_2$. Output $C = (c_p, c_q)$.

Decryption $\mathsf{Dec}(\mathbf{SK}, C)$**.** Given a ciphertext $C = (c_p, c_q)$:

1. Compute $a = R_1^{-1} \cdot (c_p \bmod S_1) \bmod S_1$ and $b = R_2^{-1} \cdot (c_q \bmod S_2) \bmod S_2$.
2. Compute $k = a/b \pmod p$. If $b \bmod p \equiv 0 \pmod p$, output $\perp$.
3. Solve for x from $k = (f_1 x + f_0)/(h_1 x + h_0) \pmod p$. If the denominator is zero, output $\perp$.
4. Output the recovered x.

Correctness. Correctness follows directly from the invariant (5): for honestly generated ciphertext with $\delta(\mathbf{u}) = 0 \pmod p$, decryption recovers $p'(x, \mathbf{u})/q'(x, \mathbf{u}) \equiv \Phi(x) \pmod p$, and the Möbius transformation $\Phi(x)$ is uniquely invertible. The scheme specifies that key generation rejects degenerate polynomial pairs satisfying $f_1 h_0 - f_0 h_1 = 0$, which occurs with probability at most $1/p$. The reference implementation does not perform this check, because the probability is below 2^{-128} at Level I.

3.4. Key Encapsulation Mechanism via Double Encryption with Consistency Check

We construct a KEM using a double-encryption transform illustrated in Figure 2. The same ephemeral secret x is encapsulated twice under independent randomness, and decapsulation accepts only if the two decrypted values coincide. We prove its IND-CCA2 security in the classical random oracle model, conditional on Assumptions 1 and 3–5 (Theorem 3). The security proof models the hash function H as a random oracle. The construction is related to generic KEM transformations [31,34], but our proof is specific to the properties of the D-HPPK base PKE. We explain at the end of this subsection why the standard Fujisaki–Okamoto transform does not apply to D-HPPK.

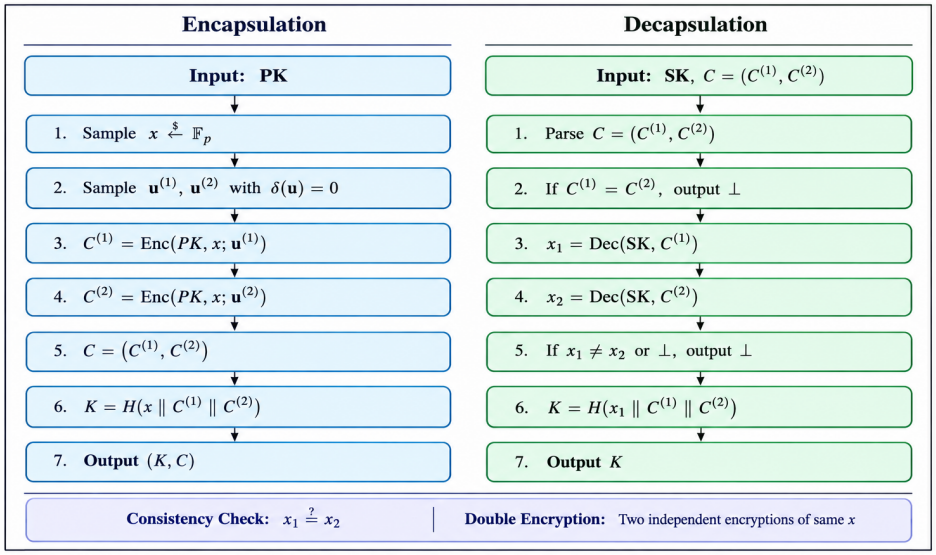


Figure 2. Encapsulation and decapsulation flow for the D-HPPK IND-CCA2 KEM. The double-encryption consistency check ensures that only ciphertexts decrypting to the same ephemeral secret x are accepted.

Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function modeled as a random oracle.

Encapsulation KEM.Encaps(**PK**).

1. Sample $x \xleftarrow{\$} \mathbb{F}_p$.
2. Sample two independent noise vectors $\mathbf{u}^{(1)}, \mathbf{u}^{(2)} \xleftarrow{\$} \{\mathbf{u} \in \mathbb{F}_p^m : \delta(\mathbf{u}) = 0 \pmod p\}$.
3. Compute $C^{(1)} = \mathsf{Enc}(\mathbf{PK}, x; \mathbf{u}^{(1)})$ and $C^{(2)} = \mathsf{Enc}(\mathbf{PK}, x; \mathbf{u}^{(2)})$.
4. Set $C = (C^{(1)}, C^{(2)})$ and $K = H(x \| C^{(1)} \| C^{(2)})$.
5. Output (K, C).

Decapsulation KEM.Decaps(**SK**, C).

1. Parse $C = (C^{(1)}, C^{(2)})$.
2. Reject if $C^{(1)} = C^{(2)}$ (Class 1).
3. Reject if $c_p^{(1)} \equiv c_p^{(2)} \pmod{S_1}$ and $c_q^{(1)} \equiv c_q^{(2)} \pmod{S_2}$ simultaneously (Class 2).
4. Reject if $C^{(2)} = \lambda C^{(1)}$ or $C^{(1)} = \lambda C^{(2)}$ for some integer λ (Class 3).
5. Reject if $\gcd(c_p^{(1)}, S_1) \neq 1$, $\gcd(c_p^{(2)}, S_1) \neq 1$, $\gcd(c_q^{(1)}, S_2) \neq 1$, or $\gcd(c_q^{(2)}, S_2) \neq 1$ (Class 5).
6. Recover $x_1 = \mathsf{Dec}(\mathbf{SK}, C^{(1)})$ and $x_2 = \mathsf{Dec}(\mathbf{SK}, C^{(2)})$.
7. If either decryption returns $\perp$ or $x_1 \neq x_2$, output $\perp$.
8. Otherwise output $K = H(x_1 \| C^{(1)} \| C^{(2)})$.

The rejection classes are analysed in Section 4.8. The decapsulation algorithm enforces Classes 1, 2, 3, and 5. Classes 1 and 3 are testable from the ciphertext alone; Classes 2 and 5 are secret-dependent and are evaluated using the hidden moduli before the main decryption computation. Class 4 is enforced by the IND-CCA2 game rather than by the decapsulation algorithm. In particular, Class 3 closes the integer-scalar-multiple collision channel that would otherwise allow an adversary to obtain a collision with non-negligible probability; Class 5 excludes the degenerate case in which one or both ciphertext components share a factor with the hidden modulus, which would otherwise allow a deterministic collision. The hidden moduli S_1, S_2 make residue-preserving shifts (Class 2) infeasible to construct without solving HMPP, and the enlarged rejection set therefore leaves only queries whose decryption outputs collide with probability at most $1/p + \mathrm{negl}(\lambda)$.

Applicability of the Fujisaki–Okamoto transform. The FO transform [29,30] requires that decapsulation be able to *reproduce* the encryption of the recovered message using only

public information. For D-HPPK this fails: the base PKE encrypts x using an ephemeral $\mathbf{u}$ sampled from the kernel of δ, and $\mathbf{u}$ is discarded. Decapsulation recovers x but not $\mathbf{u}$, so the re-encryption step cannot be executed. The two standard workarounds both require changing the base scheme and invalidating the current analysis. *Derandomizing* the encryption (replacing $\mathbf{u}$ with a deterministic function of x) makes re-encryption well-defined but changes the base PKE and invalidates the current MRP_1 analysis. *Including* $\mathbf{u}$ in the ciphertext changes the ciphertext format and the base PKE security game, so the current $\mathrm{MRP}_1/\mathrm{MRP}_2$ analysis no longer applies directly. We therefore adopt the double-encryption construction, which avoids both modifications at the cost of a $2\times$ ciphertext and a $2\times$ workload.

Table 1 compares the two approaches. The double-encryption construction is the correct choice *given the current base PKE*. If a future revision derandomizes the encryption and proves the base PKE remains MRP_1-hard, the FO transform would halve the ciphertext size. We note this as a direction for future work.

Table 1. Comparison of the double-encryption construction (this work) with a hypothetical FO transform applied to D-HPPK. The FO figures assume the encryption is derandomized.

Property	Double Encryption	FO (Hypothetical)
Encapsulation operations	2 Enc	1 Enc
Decapsulation operations	2 Dec + check	1 Dec + 1 re-Enc + check
Re-encryption available	No	Required
Base PKE security analysis	Unchanged	Must be redone
IND-CCA2 proof	Direct (this work)	Standard FO

4. Security Analysis of D-HPPK KEM

The security of D-HPPK rests on five assumptions: HMPP (Assumption 1), D-HMPP (Assumption 2), Structured D-HMPP (Assumption 3), MRP in its two forms (Assumption 4), and pairwise decryption-collision resistance (Assumption 5). The security framework is stated in Section 1.2. The technical content of this section is: (i) the search-problem framework (Section 4.1); (ii) the formal games (Section 4.2); (iii) the five assumptions (Section 4.3); (iv) the decryption-safety analysis (Section 4.4); (v) the cryptanalysis of HMPP and MRP (Sections 4.5 and 4.6); (vi) the two-phase key-recovery analysis (Section 4.7); (vii) the rejection set and the IND-CCA2 proof (Sections 4.8 and 4.9); and (viii) the concrete parameters and open problems.

4.1. Search Problems with Efficient Verification

We formalize the security framework in a form that applies to both problems underlying D-HPPK. The notion is standard: it is the framework NIST uses to assign security categories to AES and the framework underlying Merkle's puzzles [10].

Definition 1 (Search problem with efficient verification)**.** *A search problem with efficient verification is a pair* $(\mathcal{X}, V)$ *where*

1. $\mathcal{X}$ *is a finite search space of size* $|\mathcal{X}| = 2^l$*;*
2. $V : \mathcal{X} \times \mathcal{I} \to \{0,1\}$ *is a verification predicate, computable in time* $\mathrm{poly}(L)$ *for a public parameter* L*;*
3. *a distribution* $\mathcal{D}$ *over instances such that, for* $I \leftarrow \mathcal{D}$*, the set* $\{x \in \mathcal{X} : V(x, I) = 1\}$ *is non-empty with overwhelming probability.*

The enumeration-verification attack on $(\mathcal{X}, V)$ *enumerates every* $x \in \mathcal{X}$*, evaluates* $V(x, I)$*, and returns the first* x *with* $V(x, I) = 1$*. Its cost is* $O(2^l \cdot \mathrm{poly}(L))$*.*

Every search problem with efficient verification has a two-phase structure: a non-deterministic *guess phase* and a deterministic *verification phase*. This is the same structure as AES key search in the known-ciphertext setting: the guess is a candidate key $k \in \{0,1\}^{128}$, and the verification decrypts the public ciphertext under k and checks whether the result is a recognizable plaintext.

Both problems underlying D-HPPK are instances of Definition 1, each with the same two-phase structure. The HMPP instance is verified in Proposition 1 (Section 4.5); the MRP instances are verified in Proposition 4 (Section 4.6). Both share the 2^l search-space size and polynomial-time verification, which is why the IND-CCA2 proof must cover both paths via the union bound of Theorem 3.

Figure 3 summarizes the dependency structure. *Path A* (ephemeral-secret recovery) requires solving MRP_2, decomposing into a non-deterministic search over $x \in \mathbb{F}_p$ and a deterministic integer verification at cost $O(L^2)$ per candidate, for total cost $O(2^l \cdot L^2)$. *Path B* (long-term key recovery) requires solving HMPP (Theorem 2) followed by the polynomial-time Stage 2 projection onto $\mathbf{d}^\perp$, for cost $O(2^l \cdot \text{poly}(L))$. Structured D-HMPP underpins the adaptive reduction of the pairwise collision assumption (Lemma 3). The collision assumption itself bounds the collision probability of the fixed-pair oracle queries. Paths A and B are subsumed by the union bound of Theorem 3.

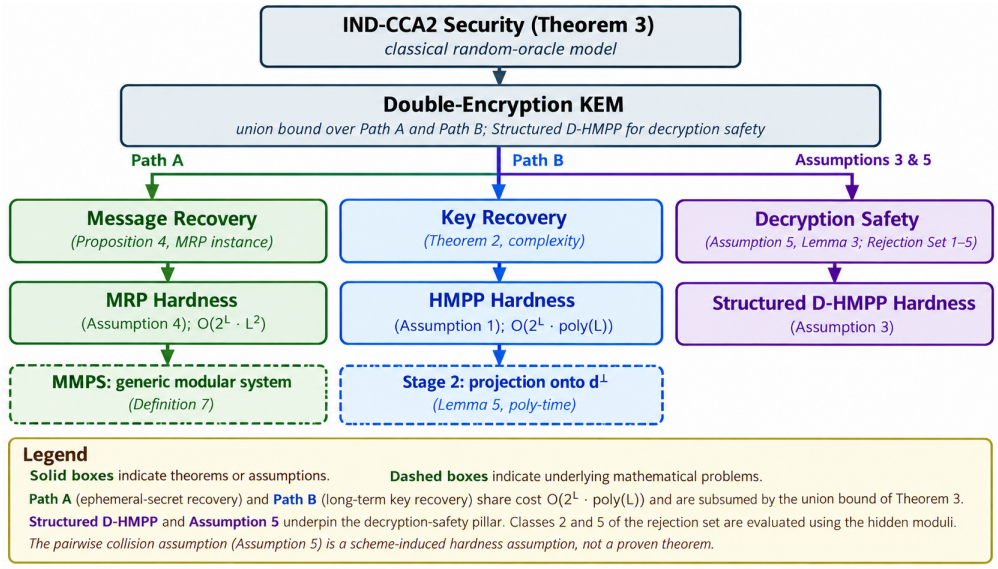


Figure 3. Security dependency structure for D-HPPK KEM. Four of the five assumptions support the IND-CCA2 proof of Theorem 3: HMPP and MRP_2 (the two search-based paths), Structured D-HMPP (adaptive reduction of the collision assumption), and pairwise decryption-collision resistance (Assumption 5). The fifth, standalone D-HMPP (Assumption 2), supports the cryptanalysis of Section 4.5 but does not appear directly in the bound. Marginal uniformity of the decryption output is proved from the hidden-ring modular reduction (Lemmas 1 and 2).

4.2. Security Model

We formalize the message-recovery games for the base PKE (single-ciphertext, MRP_1; double-ciphertext, MRP_2) and the standard IND-CCA2 game for the KEM. The parameter set is (n, m, p) with $l = \lceil \log_2 p \rceil$, $n = 2$, $m = 3$, and $L = 2l + 8$.

Definition 2 (Single-ciphertext message-recovery game $\text{Game}_{\text{MR}_1}$)**.** *The challenger runs* $(\mathbf{PK}, \mathbf{SK}) \leftarrow \text{KeyGen}(1^\lambda)$, *samples* $x^* \xleftarrow{\$} \mathbb{F}_p$ *and* $\mathbf{u}^* \xleftarrow{\$} \{\mathbf{v} \in \mathbb{F}_p^m : \delta(\mathbf{v}) = 0\}$, *computes* $C^* = (c_p^*, c_q^*) \leftarrow \text{Enc}(\mathbf{PK}, x^*; \mathbf{u}^*)$, *and sends* $(\mathbf{PK}, C^*)$ *to* $\mathcal{A}$. *The adversary* $\mathcal{A}$ *wins iff it outputs* $\hat{x} = x^*$. *The advantage is* $\text{Adv}^{\text{mr}_1}_{\text{PKE},\mathcal{A}}(\lambda) = \Pr[\hat{x} = x^*]$.

Definition 3 (Double-ciphertext message-recovery game $\mathrm{Game}_{\mathrm{MR}_2}$)**.** *Same as* $\mathrm{Game}_{\mathrm{MR}_1}$ *except the challenger samples two independent noise vectors* $\mathbf{u}^{*,(1)}, \mathbf{u}^{*,(2)}$*, computes* $C^{(i)*} = \mathrm{Enc}(\mathbf{PK}, x^*; \mathbf{u}^{*,(i)})$ *for* $i = 1, 2$*, and sends* $(\mathbf{PK}, C^{(1)*}, C^{(2)*})$ *to* $\mathcal{A}$*. The advantage is* $\mathrm{Adv}^{\mathrm{mr}_2}_{\mathrm{PKE},\mathcal{A}}(\lambda) = \Pr[\hat{x} = x^*]$.

Definition 4 (IND-CCA2 game $\mathrm{Game}_{\text{IND-CCA2}}$)**.** *The challenger runs* $(\mathbf{PK}, \mathbf{SK}) \leftarrow \mathrm{KeyGen}(1^\lambda)$ *and gives* $\mathbf{PK}$ *to* $\mathcal{A}$*, who may query* $\mathrm{Decaps}(\mathbf{SK}, \cdot)$ *adaptively.* $\mathcal{A}$ *outputs a challenge request; the challenger samples* $x^* \xleftarrow{\$} \mathbb{F}_p$ *and* $\mathbf{u}^{*,(1)}, \mathbf{u}^{*,(2)}$*, computes* $C^{(i)*} = \mathrm{Enc}(\mathbf{PK}, x^*; \mathbf{u}^{*,(i)})$*, sets* $C^* = (C^{(1)*}, C^{(2)*})$*,* $K_0 = H(x^* \| C^*)$*, samples* $K_1 \xleftarrow{\$} \{0,1\}^\ell$*, and sends* (C^*, K_b) *for* $b \xleftarrow{\$} \{0,1\}$*.* $\mathcal{A}$ *may continue querying* Decaps *subject to the rejection set* $\mathcal{R}$ *of Section 4.8, and wins iff* $\hat{b} = b$*. The advantage is* $\mathrm{Adv}^{\text{ind-cca2}}_{\mathrm{KEM},\mathcal{A}}(\lambda) = |\Pr[\hat{b} = b] - \frac{1}{2}|$.

4.3. Hardness Assumptions

The following five assumptions are the foundation of the security analysis.

Assumption 1 (HMPP hardness)**.** *For every PPT adversary* $\mathcal{B}$*,*

$$\mathrm{Adv}^{\mathrm{hmpp}}_{\mathcal{B}}(\lambda) = \Pr\left[\begin{array}{l} (A_1, \ldots, A_t) \leftarrow \mathcal{D}_{\mathrm{HMPP}}(1^\lambda); \\ \hat{y} \leftarrow \mathcal{B}(A_1, \ldots, A_t) \end{array} : \hat{y} = (S, R) \right] \leq \mathrm{negl}(\lambda).$$

Assumption 2 (Decisional HMPP)**.** *For every PPT adversary* $\mathcal{B}$*,*

$$|\Pr[\mathcal{B}(A_1, \ldots, A_t) = 1] - \Pr[\mathcal{B}(U_1, \ldots, U_t) = 1]| \leq \mathrm{negl}(\lambda),$$

where $(A_1, \ldots, A_t) \leftarrow \mathcal{D}_{\mathrm{HMPP}}(1^\lambda)$ *and* $(U_1, \ldots, U_t) \xleftarrow{\$} (\mathbb{Z}^*_S)^t$*.*

Assumption 3 (Structured D-HMPP)**.** *For every PPT adversary* $\mathcal{B}$*,*

$$\left| \Pr[\mathcal{B}(\mathbf{PK}) = 1] - \Pr\left[\mathcal{B}(\mathbf{PK}^{\mathrm{ideal}}) = 1\right] \right| \leq \mathrm{negl}(\lambda),$$

where $\mathbf{PK}$ *is generated by* $\mathrm{KeyGen}(1^\lambda)$ *and* $\mathbf{PK}^{\mathrm{ideal}}$ *replaces the masked public coefficients* $(\hat{P}_{ij}, \hat{Q}_{ij})$ *by independent uniform elements of* $\mathbb{Z}^*_{S_1}$ *and* $\mathbb{Z}^*_{S_2}$*, with the constraint vector* $\mathbf{d}$ *and all public structure preserved.*

On the relationship between HMPP, D-HMPP, and Structured D-HMPP. The standalone D-HMPP of Assumption 2 applies to samples $A_i \equiv R\tilde{a}_i \pmod{S}$ with independent uniform $\tilde{a}_i$. The D-HPPK public key, however, exposes structured masked coefficients $p'_{ij} = p^{(0)}_{ij} + c^{(p)}_i d_j$ whose joint distribution is determined by the hidden polynomial structure of $p^{(0)}$ and the rank-one masking. Structured D-HMPP (Assumption 3) is the form actually invoked in the adaptive reduction of Lemma 3; the standalone form of Assumption 2 is used only in the cryptanalysis of Section 4.5 and is a weaker requirement. We state both separately so the reduction does not rely on an implicit extension. Structured D-HMPP is the standard pseudorandomness assumption used in the analysis of ACD-style and hidden-modulus schemes [35,36,40]. The D-HPPK security claim rests on the conjunction of all five assumptions.

On the structured D-HMPP distribution. The masking is designed so that the structured coefficients do not leak the low-dimensional structure of $p^{(0)}$: the Bézout-combination attack family of Section 4.5.4 shows that the LRA, which breaks the unmasked construction, does not apply to the masked coefficients. Structured D-HMPP asserts that, in addition to not revealing $p^{(0)}$, the structured coefficients are computationally indistinguishable from

independent uniform units of the hidden rings. This is the property required by the hybrid argument of Lemma 3.

Assumption 4 (MRP hardness)**.** *For every PPT adversary* $\mathcal{B}$*,*

$$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{mrp}}(\lambda) = \max\{\mathrm{Adv}_{\mathcal{B}}^{\mathrm{mr}_1}(\lambda),\ \mathrm{Adv}_{\mathcal{B}}^{\mathrm{mr}_2}(\lambda)\} \le \frac{1}{p} + \mathrm{negl}(\lambda).$$

On the nature of MRP. MRP is the scheme-induced search problem whose distribution is generated by D-HPPK KeyGen and Enc. By Proposition 4, both formulations are search problems with efficient verification on $\mathcal{X} = \mathbb{F}_p$ of size 2^l. The best-known analyzed attack on MRP_2 is identical to that on MRP_1 up to a constant factor: enumerate $x \in \mathbb{F}_p$, solve the linear system for each component's noise vector, and verify both integer ciphertext identities. We state MRP as an assumption rather than deriving it from a well-studied problem, because no such derivation is currently known. MRP_2 is the form actually used by the KEM proof; MRP_1 is stated for completeness.

Assumption 5 (Pairwise decryption-collision resistance)**.** *For every fixed ciphertext pair* $C^{(1)} \neq C^{(2)}$*, over the secret-key randomness* (S_1, R_1, S_2, R_2)*,*

$$\Pr\Big[(C^{(1)}, C^{(2)}) \notin \mathcal{R} \ \wedge\ \mathrm{Dec}(\mathbf{SK}, C^{(1)}) = \mathrm{Dec}(\mathbf{SK}, C^{(2)})\Big] \ \le\ \frac{1}{p} + \mathrm{negl}(\lambda),$$

except for pairs $(C^{(1)}, C^{(2)})$ *generated as two honest encryptions of the same plaintext under* Enc*.*

On the nature of Assumption 5. The assumption is stated in joint form: the probability that a fixed ciphertext pair is both admissible (outside $\mathcal{R}$) and colliding is at most $1/p + \mathrm{negl}(\lambda)$. This avoids a conditioning subtlety, because $\mathcal{R}$ contains the secret-dependent Classes 2 and 5, so admissibility itself depends on the secret-key sample; a conditional formulation would require the adversary to know the secret-dependent admissibility of its query, which is not the case in the game. The joint form automatically accounts for the cases in which a candidate pair is rejected by Class 2 or Class 5, since those events contribute to the complementary probability $(C^{(1)}, C^{(2)}) \in \mathcal{R}$ and are not counted.

The assumption is scheme-induced: its instance distribution is determined by KeyGen and Enc, and it is a companion to MRP_2. Where MRP_2 says that the adversary cannot recover x^* from the challenge pair, Assumption 5 says that the adversary cannot even produce a distinct admissible pair that decapsulation would accept as consistent. The assumption is consistent with Lemma 1, which shows that the marginal law of each decryption output is close to uniform; it additionally posits near-independence of the two outputs, which is the genuinely two-dimensional statement not proved here. We invite cryptanalysis on this point.

4.4. Decryption-Safety Properties

This subsection establishes the decryption-safety properties used by the IND-CCA2 proof: a marginal statistical-uniformity result for the decryption output (Lemmas 1 and 2) and the pairwise decryption-collision assumption for distinct ciphertext pairs outside the rejection set $\mathcal{R}$ (Assumption 5, stated in Section 4.3). We give a short hybrid reduction that derives the adaptive property from the fixed-pair assumption through Structured D-HMPP (Lemma 3). We state the pairwise property as an assumption rather than deriving it from the one-dimensional modular-reduction lemmas, because establishing the joint near-independence of two decryption outputs directly from the hidden-ring structure is

a genuinely two-dimensional equidistribution problem that we leave for future analysis (Section 6); the one-dimensional marginal uniformity is proved.

4.4.1. Analytic Preliminaries

The two lemmas below establish the statistical ingredients underlying the marginal uniformity of the decryption output for a fixed admissible ciphertext: Lemma 1 gives near-uniformity of each reduced intermediate, and Lemma 2 transfers near-uniformity through the field division that produces the decryption output.

Lemma 1 (Modular-reduction uniformity)**.** *Let p be prime, let $S > 9p^2$ be an L-bit integer, and let c be an integer with $\gcd(c, S) = 1$. Let Y be uniform on $\mathbb{Z}_S^*$. Then*

$$\Delta\big((cY \bmod S) \bmod p,\ U(\mathbb{F}_p)\big) \ \le\ \frac{c_0}{p},$$

where $c_0 = 2^{O(L/\log L)}\, O(\log L)$ is sub-exponential in L. In particular, $c_0/p = 2^{-\Omega(L)}$ for the D-HPPK parameter relation $L = 2l + 8$ with p an l-bit prime.

Proof. *Case 1:* $\gcd(S, p) = 1$. Since $\gcd(c, S) = 1$, multiplication by c is a permutation of $\mathbb{Z}_S^*$. Hence $a = cY \bmod S$ is uniform on $\mathbb{Z}_S^*$. For each $r \in \mathbb{F}_p$, let $N_r = |\{y \in \mathbb{Z}_S^* : y \equiv r \pmod p\}|$. By inclusion–exclusion over the squarefree divisors of S,

$$N_r = \frac{\varphi(S)}{p} + \eta_r, \qquad |\eta_r| \le 2^{\omega(S)}.$$

Therefore

$$\Delta\big(a \bmod p, U(\mathbb{F}_p)\big) \le \frac{p\, 2^{\omega(S)-1}}{\varphi(S)}.$$

By the maximal-order theorem for $\omega(S)$ [41], $\omega(S) = O(\log S/\log\log S)$, so $2^{\omega(S)} = 2^{O(L/\log L)}$. Using the standard lower bound on $\varphi(S)$ and $S > 9p^2$,

$$\Delta\big(a \bmod p, U(\mathbb{F}_p)\big) \le \frac{2^{O(L/\log L)}\, O(\log L)}{p}.$$

Thus $c_0 = 2^{O(L/\log L)}\, O(\log L)$ is sub-exponential in L.

Case 2: $p \mid S$. Write $S = pS'$. Since $\gcd(c, S) = 1$ and $Y \in \mathbb{Z}_S^*$, we have $p \nmid cY$, so $(cY \bmod S) \bmod p = cY \bmod p$. Multiplication by c is a bijection on $\mathbb{F}_p^*$, so $cY \bmod p$ is uniform on $\mathbb{F}_p^*$, and

$$\Delta\big((cY \bmod S) \bmod p,\ U(\mathbb{F}_p)\big) = \frac{1}{p}.$$

Combining both cases. The bound $\Delta \le c_0/p$ holds with $c_0 = 2^{O(L/\log L)}\, O(\log L)$. For $L = 2l + 8$, with $p \ge 2^{l-1} = 2^{L/2-5}$, Case 1 gives $c_0/p \le 2^{-\Omega(L)}$ and Case 2 gives $1/p = 2^{-\Omega(L)}$. □

Lemma 2 (Ratio of nearly-uniform variables)**.** *Let X, Y be independent random variables on $\mathbb{F}_p$ with $\Delta(X, U) \le \varepsilon$ and $\Delta(Y, U) \le \varepsilon$. Define $Z = X/Y$ when $Y \ne 0$ and $Z = 0$ otherwise, with the convention that X/Y is computed in $\mathbb{F}_p$. Then*

$$\Delta(Z, U) \le 2\varepsilon + \frac{p-1}{p^2} < 2\varepsilon + \frac{1}{p}.$$

Proof. Let (X', Y') be independent and uniform on $\mathbb{F}_p^2$, and define Z' from (X', Y') as Z is defined from (X, Y). By the product bound for total variation, $\Delta((X, Y), (X', Y')) \le 2\varepsilon$.

The data-processing inequality gives $\Delta(Z, Z') \le 2\varepsilon$. For $z \neq 0$, $\Pr[Z' = z] = (p-1)/p^2$, and $\Pr[Z' = 0] = (2p-1)/p^2$, so $\Delta(Z', U) = (p-1)/p^2$. The triangle inequality yields the stated bound. □

4.4.2. Pairwise Decryption-Collision

We invoke Assumption 5 and derive its adaptive form. The assumption is stated for fixed ciphertext pairs outside the rejection set $\mathcal{R}$, with the probability taken over the secret-key randomness. The adaptive version, in which the adversary chooses the pair as a function of **PK**, is reduced to the fixed version by Structured D-HMPP (Lemma 3).

Lemma 3 (Adaptive collision from the fixed-pair assumption)**.** *Assume Structured D-HMPP (Assumption 3) and Assumption 5. Then for every PPT adversary $\mathcal{A}$ that outputs a pair $C^{(1)} \neq C^{(2)}$ outside the rejection set $\mathcal{R}$,*

$$\Pr\Big[\mathrm{Dec}(\mathbf{SK}, C^{(1)}) = \mathrm{Dec}(\mathbf{SK}, C^{(2)})\Big] \;\le\; \frac{1}{p} + \mathrm{negl}(\lambda),$$

unless both $C^{(1)}$ and $C^{(2)}$ are honest encryptions of a common plaintext known to $\mathcal{A}$.

Proof. *Hybrid 0 (real).* $(\mathbf{PK}, \mathbf{SK}) \leftarrow \mathrm{KeyGen}$, $(C^{(1)}, C^{(2)}) \leftarrow \mathcal{A}(\mathbf{PK})$.

Hybrid 1 (ideal public key). Replace the masked public coefficients $(\hat{P}_{ij}, \hat{Q}_{ij})$ by uniform independent elements of $\mathbb{Z}^*_{S_1}, \mathbb{Z}^*_{S_2}$, keeping the constraint vector $\mathbf{d}$ and all other public structure unchanged. By Assumption 3, Hybrid 0 and Hybrid 1 are computationally indistinguishable, so the joint distribution of $(\mathcal{A}(\mathbf{PK}), \mathrm{Dec}(\mathbf{SK}, \cdot))$ changes by at most $\mathrm{negl}(\lambda)$.

Hybrid 2 (independent secret key). In the ideal world, the public key is independent of (S_1, R_1, S_2, R_2). Condition on $\mathbf{PK}^{\mathrm{ideal}}$ and on $(C^{(1)}, C^{(2)}) = \mathcal{A}(\mathbf{PK}^{\mathrm{ideal}})$; the secret-key randomness remains uniform and independent of the conditioning. Assumption 5 then gives

$$\Pr\Big[(C^{(1)}, C^{(2)}) \notin \mathcal{R} \;\wedge\; \mathrm{Dec}(C^{(1)}) = \mathrm{Dec}(C^{(2)})\Big] \;\le\; \frac{1}{p} + \mathrm{negl}(\lambda).$$

Conditioning on the admissible event $(C^{(1)}, C^{(2)}) \notin \mathcal{R}$ — which has probability bounded below by a non-negligible quantity in the relevant game regime — yields the conditional bound $\Pr[\mathrm{Dec}(C^{(1)}) = \mathrm{Dec}(C^{(2)}) \mid (C^{(1)}, C^{(2)}) \notin \mathcal{R}] \le 1/p + \mathrm{negl}(\lambda)$. Combining the two hybrid transitions yields the stated bound. The excluded honest-double-encryption case is handled as in Corollary 1. □

4.5. Hidden Modulus Product Problem

Definition 5 (HMPP instance distribution)**.** *Let p be the public prime with $l = \lceil \log_2 p \rceil$, let L be an integer with $L > 2l$, and let $t \ge 3$. The instance distribution $\mathcal{D}_{\mathrm{HMPP}}(1^\lambda)$ proceeds as follows:*

1. *Sample a secret modulus S uniformly from the set of L-bit integers satisfying $S > 9p^2$. This is the same distribution as* KeyGen *(Section 3.1); HMPP instances are therefore drawn from the exact distribution that D-HPPK uses, and the assumption is not strengthened by enlarging the instance set.*
2. *Sample a secret coprime multiplier $R \xleftarrow{\$} \mathbb{Z}^*_S$.*
3. *For $i = 1, \ldots, t$, sample $a_i \xleftarrow{\$} \mathbb{F}_p$ uniformly, and let $\tilde{a}_i \in [0, p-1]$ denote its canonical integer representative.*
4. *Compute the public samples $A_i \equiv R \cdot \tilde{a}_i \pmod{S}$.*

The secret trapdoor is $(S, R, a_1, \ldots, a_t)$. The public instance is $(A_1, \ldots, A_t)$.

Definition 6 (D-HMPP instance distribution)**.** *The D-HMPP experiment proceeds as follows.*

1. *Sample a secret modulus S uniformly from the L-bit integers satisfying $S > 9p^2$, matching the HMPP instance distribution of Definition 5.*
2. *With probability $1/2$, sample a secret multiplier $R \xleftarrow{\$} \mathbb{Z}_S^*$ and independent $a_i \xleftarrow{\$} \mathbb{F}_p$, and set $A_i \equiv R\tilde{a}_i \pmod{S}$ for $i = 1, \ldots, t$.*
3. *Otherwise, sample $A_i \xleftarrow{\$} \mathbb{Z}_S^*$ independently and uniformly for $i = 1, \ldots, t$.*

The adversary receives $(A_1, \ldots, A_t)$ and outputs a bit $\hat{b}$. The D-HMPP advantage is

$$\left|\Pr[\hat{b} = 1 \mid \textit{structured}] - \Pr[\hat{b} = 1 \mid \textit{uniform}]\right|.$$

The modulus S is not given to the adversary; it is part of the secret trapdoor, exactly as in Definition 5.

HMPP is a new problem introduced in this work, related in spirit to ACD [35,36,38,40] but differing in that a single *large* multiplier $R \in \mathbb{Z}_S^*$ acts on all small values $\tilde{a}_i \in [0, p-1]$, and the resulting samples are uniform residues modulo S. We examine five attack families below, and identify a sixth class (full-sample lattice attacks) as open for further cryptanalysis.

Proposition 1 (HMPP is a search problem with efficient verification)**.** *Under the heuristic lattice model of Proposition 3, HMPP has the two-phase structure of Definition 1. The guess phase is a non-deterministic choice of $\hat{a}_1 \in [0, p-1]$, of size $2^{l-1} \le |\mathcal{X}| \le 2^l$. The verification phase applies lattice reduction to recover a candidate (S, R), then checks $R\tilde{a}_i \equiv A_i \pmod{S}$ for all i, in $O(t \cdot \mathrm{poly}(L))$ bit operations.*

4.5.1. Elementary GCD Attack (Rigorous)

Proposition 2 (Elementary attack on HMPP)**.** *For $t \ge 3$, there is an elementary algorithm that recovers (S, R) from $(A_1, \ldots, A_t)$ in time $O(2^{2l} \cdot \mathrm{poly}(L))$ classically.*

Proof. Fix guesses $\hat{a}_1, \hat{a}_j \in [0, p-1]$ for two of the small values, $j \in \{2, \ldots, t\}$. If both guesses are correct, $B_j = A_1\hat{a}_j - A_j\hat{a}_1$ is a nonzero multiple of S. Taking gcds over several j isolates S up to a small cofactor. Once S and a_1 are known, $R = A_1 a_1^{-1} \bmod S$ and $a_j = A_j R^{-1} \bmod S$. Wrong guesses are rejected by the consistency check. The dominant cost is the search over $(\hat{a}_1, \hat{a}_j)$. □

4.5.2. Lattice Attack (Heuristic)

A single-guess lattice attack has been proposed in the ACD literature. It reduces the guess space by half, at the cost of a lattice-reduction step whose applicability to HMPP is not proven.

Proposition 3 (Heuristic lattice attack on HMPP)**.** *If the ACD lattice template applies to the HMPP instance distribution, there is a classical algorithm that recovers (S, R) in time $O(2^l \cdot \mathrm{poly}(L))$.*

Proof. Fix a guess $\hat{a}_1 \in [0, p-1]$. For the correct guess, $B_j = A_1\tilde{a}_j - A_j\hat{a}_1$ is a multiple of S for all j. Under the heuristic assumption that the ACD template applies, a lattice of dimension t constructed from the public values and a scaling factor 2^ρ with $\rho \approx L$ contains the vector $(\tilde{a}_2, \ldots, \tilde{a}_t, S)$ as a short vector, which LLL/BKZ recovers in $\mathrm{poly}(L)$ time. The dominant cost is the search over $\hat{a}_1$. □

The proposition is conditional. If the ACD template applies, the best-known attack costs $O(2^l \cdot \mathrm{poly}(L))$; if it does not, the rigorous fallback is the elementary GCD attack of

Proposition 2 at $O(2^{2l} \cdot \mathrm{poly}(L))$. The true best attack lies in the range $[2^l, 2^{2l}]$. Parameter selection (Section 5.1) adopts the 2^l bound, so the claimed security level holds even if the heuristic is optimistic. The ACD template is not proven to apply: the standard technique requires small residues modulo the unknown modulus, whereas the HMPP residues $A_i \bmod S$ are uniform in the image of the multiplication-by-R map, and eliminating R requires knowing S. Whether the technique can be adapted is open (Section 4.12).

4.5.3. Algebraic and Statistical Attacks (No Improvement)

We examined three algebraic attack families. *Rational reconstruction* exploits the relation $A_i/A_j \equiv \tilde{a}_i/\tilde{a}_j \pmod{S}$, but reducing the fractions modulo S requires knowing S; without it, the relation is uninformative. *Simultaneous modular relations* uses the products $A_i\tilde{a}_j - A_j\tilde{a}_i$, which are multiples of S; eliminating S between two such products yields a bilinear relation in the unknown small values that does not isolate S. *Gröbner basis techniques* do not apply directly because the defining relations are congruences over $\mathbb{Z}$, not equations over a field. We found no algebraic attack that improves upon the elementary bound of Proposition 2.

We examined three statistical attack families. *Cofactor leakage* does not apply because the quotients $k_i = \lfloor R\tilde{a}_i/S \rfloor$ are uniform over a wide range and reveal nothing about S. *GCD-based combinations* do not isolate S because the products A_iA_j are uniform modulo S up to the shared factor R^2. *Small-coefficient bias* is defeated by the injectivity of multiplication by R: in the generic case $\gcd(S, p) = 1$, the image set $\mathcal{I}_R = \{R\tilde{a} \bmod S : \tilde{a} \in [0, p-1]\}$ has exactly p elements and density $p/S \approx 2^{-l-8}$, so no residue is preferred. We found no statistical attack that improves upon the elementary GCD bound of Proposition 2.

4.5.4. Bézout-Combination Attacks (Blocked by Masking)

The Lattice Reconstitution Attack (LRA) [9] recovers the secret key of the unmasked HPPK KEM in polynomial time. D-HPPK applies additive random masking *before* hidden-ring embedding, producing masked coefficients $p'_{ij} = p^{(0)}_{ij} + c^{(p)}_i d_j$. The rank-one perturbation destroys the rank-two structure of $p^{(0)}$, and for $m = n + 1$ the masked matrix is generically full-rank. The Bézout-combination construction that underlies the LRA therefore does not yield a usable discriminator on D-HPPK public samples.

4.5.5. Full-Sample Lattice Attacks (Open)

Using all t samples simultaneously is the strongest known approach to ACD [35,36]; whether this technique applies to HMPP, and with what complexity, is left open in Section 4.12.

4.5.6. Best-Known Attack Summary

The attacks surveyed above are summarized in Table 2. The strongest candidate attack is the heuristic lattice attack at $O(2^l \cdot \mathrm{poly}(L))$; the rigorous fallback is the elementary GCD attack at $O(2^{2l} \cdot \mathrm{poly}(L))$. The remaining attack families either do not improve on these bounds or are blocked by the masking. Parameter selection adopts the 2^l bound (Section 5.1).

Table 2. Summary of attacks on HMPP. Complexity is classical.

Attack	Complexity	Status
Elementary GCD (two guesses)	$2^{2l} \cdot \mathrm{poly}(L)$	Rigorous
Single-guess + lattice (ACD template)	$2^l \cdot \mathrm{poly}(L)$	Heuristic
Algebraic (rational, Gröbner)	No improvement	Examined
Statistical (cofactor, gcd, bias)	No improvement	Examined
Bézout-combination (LRA)	Blocked by masking	Not applicable
Full-sample lattice	Open	Invited

4.6. Message Recovery Problem

The problem an adversary faces against D-HPPK has two formulations. At the mathematical level, inverting a system of polynomial congruences over $\mathbb{F}_p$ is the **Modular Multivariate Polynomial System (MMPS)**. It is parameterized only by the number of equations k, the number of unknowns $m+1$, and the degrees of the coefficient polynomials. At the scheme level, the ciphertext recovery problem is the **Message Recovery Problem (MRP)** of Definitions 2 and 3. MRP is scheme-induced: its instance distribution references KeyGen and Enc, it inherits the constraint $\mathbf{d}^\top \mathbf{u} \equiv 0 \pmod{p}$, and its integer ciphertext identity retains the unreduced hidden-modulus information that pins down x^*.

MRP reduces to MMPS by reducing the ciphertext equations modulo p and appending the constraint. The hardness assumption is placed on MRP, not on the generic MMPS problem, because the modular system admits a solution for every free variable x and therefore does not by itself distinguish the true secret x^* (Section 4.6.4).

4.6.1. The Generic MMPS Problem

Definition 7 (MMPS)**.** *Let p be prime with $l = \lceil \log_2 p \rceil$, and let $n, m, k \geq 1$. An **MMPS instance** is a system of k polynomial congruences over $\mathbb{F}_p$ in $m+1$ unknowns $(x, u_1, \ldots, u_m)$,*

$$h_\ell(x, u_1, \ldots, u_m) \equiv b_\ell \pmod{p}, \qquad \ell = 1, \ldots, k,$$

where each h_ℓ is linear in $(u_1, \ldots, u_m)$ and polynomial in x of degree at most n:

$$h_\ell(x, \mathbf{u}) = \sum_{j=1}^{m} f_{\ell,j}(x)\, u_j, \qquad f_{\ell,j}(x) = \sum_{i=0}^{n} c_{\ell,j,i}\, x^i,$$

*with $c_{\ell,j,i}, b_\ell \in \mathbb{F}_p$. The **MMPS problem** is: given a valid MMPS instance, output $x^* \in \mathbb{F}_p$ for which the system admits a solution $(u_1^*, \ldots, u_m^*) \in \mathbb{F}_p^m$.*

The D-HPPK instantiation has $k = 3$ equations in $m+1$ unknowns: one constraint constant in x, and one equation from each of c_p and c_q. At the default parameters $m = 3$, the system is underdetermined by exactly one variable (x), and the solution variety has dimension one. For each fixed candidate x, the system reduces to three equations in (u_1, u_2, u_3), which is generically determined.

Remark 1 (Why MMPS is a natural problem)**.** *MMPS belongs to the well-studied class of structured polynomial systems over finite fields [23–27]. The D-HPPK instantiation differs from these generic constructions in its coefficient distribution and in the specific structure exploited by the enumeration-verification procedure described in Section 4.6.3.*

4.6.2. MRP: The Scheme-Induced Message Recovery Problem

Definition 8 (MRP_1 and MRP_2 instance distributions)**.** *The single-ciphertext instance distribution $\mathcal{D}_{\mathrm{MRP}_1}(1^\lambda)$ runs $(\mathbf{PK}, \mathbf{SK}) \leftarrow \mathrm{KeyGen}(1^\lambda)$, samples $x^* \xleftarrow{\$} \mathbb{F}_p$ and $\mathbf{u}^* \xleftarrow{\$} \{\mathbf{v} \in \mathbb{F}_p^m : \mathbf{d}^\top \mathbf{v} = 0\}$, computes $C^* \leftarrow \mathrm{Enc}(\mathbf{PK}, x^*; \mathbf{u}^*)$, and outputs the instance $(\mathbf{PK}, C^*)$ and the secret x^*. The double-ciphertext instance distribution $\mathcal{D}_{\mathrm{MRP}_2}(1^\lambda)$ is identical except it samples two independent noise vectors $\mathbf{u}^{*,(1)}, \mathbf{u}^{*,(2)}$ and outputs $(\mathbf{PK}, C^{(1)*}, C^{(2)*})$.*

The ciphertext identity is exact over the integers:

$$\begin{aligned} c_p^* &= \sum_{i=0}^{n} \sum_{j=1}^{m} \hat{P}_{ij} \left(x^{*i} u_j^* \bmod p \right), \\ c_q^* &= \sum_{i=0}^{n} \sum_{j=1}^{m} \hat{Q}_{ij} \left(x^{*i} u_j^* \bmod p \right). \end{aligned} \tag{9}$$

The integer structure is what distinguishes MRP from its modular reduction and is the only place where the hidden-modulus information is retained.

Proposition 4 (MRP is a search problem with efficient verification)**.** *Both* MRP_1 *and* MRP_2 *have the two-phase structure of Definition 1. The guess phase is a non-deterministic choice of a single* $x \in \mathcal{X} = \mathbb{F}_p$ *of size* 2^l*. The verification predicate* $V(x, C)$ *is defined as follows. If* $M(x)$ *is invertible, compute* $\mathbf{u}(x) = M(x)^{-1}\mathbf{b}$ *and check the integer ciphertext identity (9); output* 1 *if it holds and* 0 *otherwise. If* $M(x)$ *is singular, output* 0*. The determinant* $\det M(x)$ *vanishes for at most* 4 *values of* x*, so the exceptional cases remove at most* 4 *candidates from the search space and do not affect the asymptotic cost.*

For MRP_1*, the verification is* $O(L^2)$ *bit operations. For* MRP_2*, the same guess* x *is used to verify both components, each at* $O(L^2)$ *bit operations, for a total of* $O(L^2)$ *up to a constant factor of two. The search space in both cases is* 2^l *up to an additive constant; only the verification cost differs by a constant.*

4.6.3. Reduction to MMPS and Complexity

An MRP instance determines an MMPS instance by reducing the ciphertext equations modulo p and appending the constraint. Let $\bar{p}_{ij} = \hat{P}_{ij} \bmod p$, $\bar{q}_{ij} = \hat{Q}_{ij} \bmod p$, $\bar{c}_p = c_p^* \bmod p$, and $\bar{c}_q = c_q^* \bmod p$. The resulting MMPS instance has $k = 3$ equations in $m + 1$ unknowns:

$$\begin{aligned} \sum_{j=1}^{m} d_j u_j &\equiv 0 \pmod{p}, \\ \sum_{i=0}^{n} \sum_{j=1}^{m} \bar{p}_{ij} x^i u_j &\equiv \bar{c}_p \pmod{p}, \\ \sum_{i=0}^{n} \sum_{j=1}^{m} \bar{q}_{ij} x^i u_j &\equiv \bar{c}_q \pmod{p}. \end{aligned} \tag{10}$$

Lemma 4 (MRP solutions reduce to MMPS solutions)**.** *If* $(x^*, \mathbf{u}^*)$ *satisfies the MRP integer identity (9), then its reduction modulo* p *satisfies the MMPS system (10).*

Proof. Reduction modulo p is a ring homomorphism from $\mathbb{Z}$ to $\mathbb{F}_p$. The reverse direction does not hold in general: an MMPS solution need not lift to an MRP solution. □

Theorem 1 (Complexity of MRP and of the D-HPPK MMPS sub-family)**.** *At* $m = 3$*, the best-known analyzed attack on both* MRP_1 *and* MRP_2*, equivalently on MMPS instances of the D-HPPK form (10), has running time* $O(2^l \cdot L^2)$ *classical bit operations. This is an upper bound established by exhaustive search; no matching lower bound is claimed. For* $m > 3$*, the asymptotic complexity remains* $O(2^l \cdot \mathrm{poly}(L))$*.*

Proof. For each fixed guess $\hat{x} \in \mathbb{F}_p$, the system (10) becomes three linear equations in the three unknowns u_1, u_2, u_3. If the coefficient matrix $M(\hat{x})$ is non-singular, the system has a unique solution $\mathbf{u}$, which is then verified against the integer identity (9) at cost $O(L^2)$ bit operations. The determinant $\det M(x)$ vanishes for at most 4 values of $\hat{x}$, handled by a

constant-size exception list. For MRP_2, the procedure is applied to each component; the cost doubles, which does not change the O class. The search is over 2^l candidates. □

The two-phase structure makes the complexity chain explicit:

$$\underbrace{O(2^l)}_{\text{MMPS search over } x} \times \underbrace{O(L^2)}_{\text{integer verification}} = O(2^l \cdot L^2).$$

Under the search-based convention of Section 1.2, the security level of D-HPPK is l bits, determined by the size of the search space $|\mathbb{F}_p| = 2^l$ and the $O(L^2)$-time verification.

Message-recovery security of the base PKE. By construction, the hardness of MRP_1 is the message-recovery security of the base PKE: the single-ciphertext game of Definition 2 and the MRP_1 instance distribution of Definition 8 are generated by the same KeyGen and Enc algorithms. The identification is definitional, not a reduction: MRP_1 hardness *is* the message-recovery security property of the base PKE under the search-based convention of Section 1.2. The double-ciphertext formulation MRP_2 is the form used by the IND-CCA2 proof of Section 4.9, and has the same $O(2^l \cdot L^2)$ best-known analyzed attack cost by Theorem 1.

4.6.4. Algebraic Attacks: Trivial Elimination Ideal

We bound the degree of regularity of the D-HPPK sub-family of MMPS. Let $I = \langle g_1, g_2, g_3 \rangle \subset \mathbb{F}_p[x, u_1, u_2, u_3]$ denote the ideal generated by the system (10), and write the system in matrix form $M(x)\mathbf{u} = \mathbf{b}$, where $\mathbf{b} = (0, \bar{c}_p, \bar{c}_q)^\top \in \mathbb{F}_p^3$.

The variety $V(I)$ is one-dimensional. For each x at which $M(x)$ is invertible, the system has a unique solution $\mathbf{u}(x)$. The determinant $\det M(x)$ vanishes for at most 4 values of x. For all other x, the projection $V(I) \to \mathbb{A}^1_x$ is a bijection, so $V(I)$ is a rational curve parameterized by x and has dimension 1.

Degree bound on the rational parametrization. Because the system is linear in $\mathbf{u}$, the Gröbner basis is computable by Gaussian elimination over $\mathbb{F}_p(x)$. The Leibniz expansion gives $\deg \det M(x) \leq 4$, and each entry of $\operatorname{adj} M(x)$ is of degree at most 4 in x. Consequently the rational parametrization $x \mapsto \mathbf{u}(x)$ has numerator and denominator degree at most 4, and Gröbner elimination introduces no high-degree condition that isolates x. We do not claim a Castelnuovo–Mumford regularity bound for the ideal I; the degree bound on the parametrization is what the cryptanalysis requires.

The elimination ideal is trivial. The elimination ideal $I \cap \mathbb{F}_p[x]$ is the zero ideal, because the projection $V(I) \to \mathbb{A}^1_x$ is surjective: every $x \in \mathbb{F}_p$ is the x-coordinate of some solution. Consequently, the adversary who computes the Gröbner basis learns nothing about the true x^* from the modular system alone. To recover x^*, the adversary must use the integer ciphertext identity (9) and enumerate over all $x \in \mathbb{F}_p$. This is why the search is over $x \in \mathbb{F}_p$ rather than over the algebraic variety, and why the hardness assumption is placed on MRP rather than on the generic MMPS problem.

4.7. Key Recovery: Two-Phase Attack

Recovering the long-term secret key from the public key decomposes into two stages: first breaking the hidden rings, then exploiting the factorization structure of the underlying coefficient matrices. The first stage is HMPP; the second is polynomial. Consequently the total key-recovery cost coincides with the HMPP cost, and Theorem 2 below makes this precise.

Stage 1: Breaking the hidden rings (HMPP). The public key contains two HMPP instances: $\{\hat{P}_{ij}\}$ with modulus S_1 and multiplier R_1, and $\{\hat{Q}_{ij}\}$ with modulus S_2 and multiplier R_2. The standalone cryptanalysis of Section 4.5 establishes that the best currently analyzed attack on HMPP has classical cost $O(2^l \cdot \mathrm{poly}(L))$, conditional on the ACD-

template heuristic, with the rigorous elementary GCD fallback at $O(2^{2l} \cdot \mathrm{poly}(L))$. The purpose of the present subsection is to show that once HMPP is broken, the remaining factorization recovery is polynomial, so the total key-recovery cost coincides with the HMPP cost.

Stage 2: Recovering the factorization. After Stage 1, the adversary knows $\mathbf{p}' = \mathbf{p}^{(0)} + \mathbf{c}^{(p)}\mathbf{d}^\top$ and $\mathbf{q}' = \mathbf{q}^{(0)} + \mathbf{c}^{(q)}\mathbf{d}^\top$. Let $\mathbf{v} \in \mathbb{F}_p^m$ be nonzero with $\mathbf{d}^\top\mathbf{v} = 0$. Then $\mathbf{p}'\mathbf{v} = \mathbf{p}^{(0)}\mathbf{v}$, and likewise for $\mathbf{q}'$. Since $\mathbf{p}^{(0)} = \mathbf{F}_p\mathbf{B}$ and $\mathbf{q}^{(0)} = \mathbf{F}_h\mathbf{B}$, the projection gives

$$\mathbf{p}'\mathbf{v} = \mathbf{F}_p(\mathbf{B}\mathbf{v}), \qquad \mathbf{q}'\mathbf{v} = \mathbf{F}_h(\mathbf{B}\mathbf{v}),$$

i.e., a pair of column vectors proportional to the coefficient vectors of f and h, up to unknown scalars. Taking a second, linearly independent $\mathbf{v}' \in \mathbf{d}^\perp$ recovers the ratio and hence the full Möbius transformation $\Phi(x) = f(x)/h(x)$ in $O(1)$ field operations. This requires the nondegeneracy condition that the projected vectors $(\mathbf{B}\mathbf{v}_1, \mathbf{B}\mathbf{v}_2)$ span a rank-two subspace and that the resulting linear system for the scale ratio has nonzero determinant. The condition holds for generic $\mathbf{B}$ and fails only on a measure-zero subset of key space; if it fails, key regeneration produces a valid key.

Lemma 5 (Polynomial-time key recovery after HMPP)**.** *Given $\mathbf{p}'$, $\mathbf{q}'$, and $\mathbf{d}$, the secret polynomials $f(x)$ and $h(x)$ are recoverable in $O(\mathrm{poly}(L))$ field operations.*

Proof. Choose a basis $(\mathbf{v}_1, \mathbf{v}_2)$ of $\mathbf{d}^\perp$. The projections $\mathbf{p}'\mathbf{v}_k = \mathbf{F}_p(\mathbf{B}\mathbf{v}_k)$ and $\mathbf{q}'\mathbf{v}_k = \mathbf{F}_h(\mathbf{B}\mathbf{v}_k)$, $k = 1, 2$, are computable in $O(m)$ field operations. The column space of $(\mathbf{p}'\mathbf{v}_1, \mathbf{p}'\mathbf{v}_2)$ equals the column space of $\mathbf{F}_p$, whose left null vector is proportional to $\mathbf{u}_f = (f_1^2, -f_0 f_1, f_0^2)^\top$. The same holds for $\mathbf{q}'$ with $\mathbf{u}_h = (h_1^2, -h_0 h_1, h_0^2)^\top$. The shared factor $\mathbf{B}$ fixes the relative scale, so comparing the recovered ratios f_0/f_1 and h_0/h_1 determines the full Möbius transformation in $O(1)$ field operations. The comparison is well-defined whenever $f_1 h_0 - f_0 h_1 \neq 0$, which is the nondegeneracy condition enforced by the KeyGen rejection described in Section 3.3. □

Theorem 2 (Key-recovery complexity)**.** *Under Assumption 1, the best-known classical complexity of key recovery is $O(2^l \cdot \mathrm{poly}(L))$, matching the best-known secret-recovery cost of MRP.*

Proof. Stage 1 is bounded by Assumption 1. Stage 2 is polynomial by Lemma 5. □

4.8. Rejection Set and Formal Justification

The property required by the IND-CCA2 proof is the pairwise decryption-collision assumption (Assumption 5), with its adaptive form provided by Lemma 3. The rejection set $\mathcal{R}$ comprises five classes. Classes 1 and 3 are computable from the ciphertext alone; Classes 2 and 5 are secret-dependent, evaluated using S_1, S_2; and Class 4 is enforced by the IND-CCA2 game.

Class 1 (duplicate components): $C = (D, D)$; rejected by direct comparison.

Class 2 (residue-preserving shifts): $c_p^{(1)} \equiv c_p^{(2)} \pmod{S_1}$ and $c_q^{(1)} \equiv c_q^{(2)} \pmod{S_2}$ simultaneously. A direct construction of such a pair requires knowledge of S_1, S_2, and under the current analysis this appears to require solving an HMPP-type hidden-modulus problem; we do not claim a proof that every construction must recover the moduli. Because $S_i > 9p^2$ while $c_p < 9pS_1$ and $c_q < 9pS_2$, any residue-preserving shift is necessarily a small multiple of the hidden modulus, and the moduli are not otherwise observable from the public key.

Class 3 (integer scalar multiples): $C^{(2)} = \lambda C^{(1)}$ or $C^{(1)} = \lambda C^{(2)}$ for some integer λ. Without this rejection, an adversary could submit $C^{(2)} = \lambda C^{(1)}$ and obtain a collision with probability roughly $1/\lambda^2$: the reduction $c_p^{(1)} \bmod S_1$ is uniform in $[0, S_1 - 1]$, so the probability that $\lambda(c_p^{(1)} \bmod S_1) < S_1$ is $1/\lambda$, and similarly for $c_q^{(1)} \bmod S_2$. For small λ this is non-negligible. The Class 3 rejection closes this channel.

Class 4 (challenge-component reuse): any query whose first or second component equals, or is an integer scalar multiple of, the corresponding challenge component.

Class 5 (non-unit components): reject any query $C = (c_p, c_q)$ for which $\gcd(c_p, S_1) \neq 1$ or $\gcd(c_q, S_2) \neq 1$. This class excludes the degenerate case $c_p \equiv 0 \pmod{S_1}$ (or symmetrically), in which the intermediate value a (resp. b) is identically zero and decryption produces a deterministic collision. The predicate $\gcd(c_p, S_1) \neq 1$ is secret-dependent—it cannot be evaluated without S_1—and constructing a ciphertext that satisfies it with non-negligible probability appears to require recovery of the corresponding hidden modulus (Class 5 may be narrowed in an implementation to $c_p \equiv 0 \pmod{S_1}$ or $c_q \equiv 0 \pmod{S_2}$. This suffices to block the deterministic collision at zero residue while not firing on honestly generated ciphertexts when S_1 or S_2 is composite. Since the adversary does not know S_1, S_2, it has no optimal choice of c_p, c_q to force a non-trivial common factor; any choice succeeds only with probability determined by the sampled moduli. The residual bias for components with $\gcd(c_p, S_1) = q > 1$ is $p\,2^{\omega(S_1/q)}/\varphi(S_1/q) + O(pq/S_1)$, negligible for small q and contributing $O(1/p)$ on average for large q, so Assumption 5 is preserved. The analysis in Section 4.9 uses the general gcd form).

The five classes are defined as separate rejection conditions; they need not be mutually disjoint. For example, a pair $C^{(2)} = \lambda C^{(1)}$ can simultaneously satisfy Class 3 (integer scalar multiple) and Class 4 (challenge-component reuse) when one component of $C^{(1)}$ matches the challenge, and Class 3 preserves Class 5 membership under multiplication by λ. The five conditions are enumerated separately because each admits a distinct rejection test, not because the sets are disjoint.

Corollary 1 (Collision bound)**.** *Under Assumption 5, let $\mathcal{A}$ be a PPT adversary that outputs a pair $C^{(1)} \neq C^{(2)}$. Then*

$$\Pr\left[(C^{(1)}, C^{(2)}) \notin \mathcal{R} \ \wedge\ \mathrm{Dec}(\mathbf{SK}, C^{(1)}) = \mathrm{Dec}(\mathbf{SK}, C^{(2)})\right] \ \leq\ \frac{1}{p} + \mathrm{negl}(\lambda),$$

except in the excluded honest-double-encryption case.

Proof. Immediate from Assumption 5 and Lemma 3. □

Formal justification for the game-level restriction. The standard IND-CCA2 game prohibits the adversary from submitting the complete challenge ciphertext C^*, but not from submitting a ciphertext sharing one component with C^*, nor from submitting an integer scalar multiple. We enforce a strictly stronger restriction. Let $\mathcal{A}$ be an adversary in the restricted game, and construct $\mathcal{A}'$ in the standard game that simulates $\mathcal{A}$'s queries. When $\mathcal{A}$ queries a ciphertext $C' = (C'^{(1)}, C'^{(2)})$ with $C'^{(1)} = C^{(1)*}$, $\mathcal{A}'$ already knows $C^{(1)*}$, whose decryption under $\mathbf{SK}$ is x^*. The only information extractable from the oracle's response is whether $C'^{(2)}$ decrypts to x^*, a candidate-testing query whose success probability over random $x^* \in \mathbb{F}_p$ is at most $1/p$ per query. The same argument covers integer scalar multiples $\lambda C^{(1)*}$ and Class 5 rejections: the Class 5 predicate is secret-dependent, and the adversary cannot construct a query satisfying it with non-negligible probability without recovering the hidden modulus. Hence the Class 3, Class 4, and Class 5 restrictions only remove redundant degrees of freedom; they do not improperly constrain the adversary.

4.9. IND-CCA2 Security of the Double-Encryption KEM

The double-encryption KEM of Section 3.4 achieves IND-CCA2 security in the random oracle model, under Assumptions 1 and 3–5. Assumption 2 (standalone D-HMPP) is used only in the cryptanalysis of Section 4.5 and does not appear directly in the CCA

bound. The decryption-safety property required by the proof is the adaptive pairwise decryption-collision property of Lemma 3. An adversary can break IND-CCA2 by either:

- **Path A (ephemeral secret).** Recovering the ephemeral secret x^* directly from $(C^{(1)*}, C^{(2)*})$ via MRP_2.
- **Path B (long-term key).** Recovering the secret key **SK** via HMPP followed by the polynomial-time Stage 2 attack, then decrypting $(C^{(1)*}, C^{(2)*})$ to obtain x^*.

The proof is in the random oracle model (ROM): the hash function H is treated as a random function that the adversary may query at most q_H times. The shared secret $K = H(x \| C^{(1)} \| C^{(2)})$ binds the key to the exact ciphertext and the ephemeral secret. The proof requires that, if the adversary has not queried H on $(x^* \| C^*)$, the value K appears uniformly random.

Theorem 3 (IND-CCA2 security, conditional on Assumptions 1 and 3–5)**.** *Assume the following four assumptions hold: Assumption 1 (HMPP), Assumption 3 (Structured D-HMPP), Assumption 4 (MRP, both* MRP_1 *and* MRP_2*), and Assumption 5 (pairwise decryption-collision resistance). Let* $\mathcal{A}$ *be a PPT adversary against* $\mathrm{Game}_{\text{IND-CCA2}}$ *(Definition 4) making at most* q_D *decapsulation queries and* q_H *random-oracle queries. Then there exist PPT adversaries* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$ *such that*

$$\begin{aligned}
\mathrm{Adv}^{\text{ind-cca2}}_{\mathrm{KEM},\mathcal{A}}(\lambda) &= \left|\Pr[\hat{b} = b] - \tfrac{1}{2}\right| \\
&\le \underbrace{\mathrm{Adv}^{\mathrm{mr}_2}_{\mathcal{B}_1}(\lambda)}_{\text{Path A: MRP}_2} + \underbrace{\mathrm{Adv}^{\text{hmpp}}_{\mathcal{B}_2}(\lambda)}_{\text{Path B: HMPP}} \\
&\quad + \underbrace{\mathrm{Adv}^{\text{collision}}_{\mathcal{B}_3}(\lambda)}_{\text{fixed-pair collision}} + \underbrace{\mathrm{Adv}^{\text{structured-dhmpp}}_{\mathcal{B}_4}(\lambda)}_{\text{adaptive reduction}} \\
&\quad + \frac{(q_D + q_H)}{p} + \mathrm{negl}(\lambda).
\end{aligned}$$

Under the four assumptions above, every term on the right-hand side is negligible in λ*: the two attack-path advantages by Assumptions 4 and 1, the fixed-pair collision term by Assumption 5, and the adaptive-reduction term by Assumption 3. The trailing* $\mathrm{negl}(\lambda)$ *absorbs the random-oracle simulation loss and the polynomial slack from the per-query bounds. Assumption 2 (standalone D-HMPP) supports the cryptanalysis of Section 4.5 but does not appear directly in the bound.*

Proof. Let E_i denote the event that $\mathcal{A}$ wins in Game i.

Game 0 (real game). This is $\mathrm{Game}_{\text{IND-CCA2}}$, so $\mathrm{Adv}^{\text{ind-cca2}}_{\mathrm{KEM},\mathcal{A}}(\lambda) = |\Pr[E_0] - 1/2|$.

Game 1 (explicit rejection). Modify the game to reject Classes 1–5 explicitly. Within Decaps, enforce Classes 1, 2, 3, and 5; Classes 1 and 3 are testable from the ciphertext alone, and Classes 2 and 5 are evaluated using the hidden moduli before the main decryption computation. Class 4 is enforced by the game. By Lemma 3, the adaptive form of the fixed-pair collision bound, the remaining admissible queries satisfy

$$\Pr\Big[(C^{(1)}, C^{(2)}) \notin \mathcal{R} \ \wedge\ \mathrm{Dec}(C^{(1)}) = \mathrm{Dec}(C^{(2)})\Big] \ \le\ \frac{1}{p} + \mathrm{negl}(\lambda)$$

per query. The two losses that appear in Lemma 3 are the fixed-pair collision bound of Assumption 5 and the Structured D-HMPP hybrid that reduces the adaptive case to the fixed-pair case. Over q_D queries,

$$|\Pr[E_0] - \Pr[E_1]| \ \le\ \mathrm{Adv}^{\text{collision}}_{\mathcal{B}_3} + \mathrm{Adv}^{\text{structured-dhmpp}}_{\mathcal{B}_4} + \frac{q_D}{p} + q_D \cdot \mathrm{negl}(\lambda).$$

Game 2 (random challenge key). Replace K_0 by a uniform K_{rand} independent of b; then $\Pr[E_2] = 1/2$. A transition between Games 1 and 2 requires $\mathcal{A}$ to query H on $(x^*\|C^*)$. Three disjoint events cover this: Path A (recover x^* via MRP_2) contributes $\text{Adv}^{\text{mr}_2}_{\mathcal{B}_1}$; Path B (recover **SK** via HMPP plus Stage 2) contributes $\text{Adv}^{\text{hmpp}}_{\mathcal{B}_2}$; and guessing x^* directly contributes at most q_H/p. Hence

$$|\Pr[E_1] - \Pr[E_2]| \;\le\; \text{Adv}^{\text{mr}_2}_{\mathcal{B}_1} + \text{Adv}^{\text{hmpp}}_{\mathcal{B}_2} + \frac{q_H}{p}.$$

Combining. By the triangle inequality, and using $\Pr[E_2] = 1/2$,

$$|\Pr[E_0] - \tfrac{1}{2}| \;\le\; |\Pr[E_0] - \Pr[E_1]| + |\Pr[E_1] - \Pr[E_2]|.$$

Substituting the two bounds above and folding the $q_D \cdot \text{negl}(\lambda)$ slack and the random-oracle simulation loss into the trailing $\text{negl}(\lambda)$ yields the stated result. □

Proof notes. The random oracle H is simulated by maintaining a shared table of queried values $(x\|C) \mapsto H(x\|C)$, and decapsulation queries are answered consistently with this table. Classes 1, 2, 3, and 5 are enforced by the decapsulation algorithm; Class 4 is enforced by the IND-CCA2 game. The simulation argument in Section 4.8 shows that the game-level restriction for Class 4 is without loss of generality up to q_D/p. The enlarged rejection set is what makes the joint form of Assumption 5 applicable to every remaining oracle query, so that the collision contribution is bounded by q_D/p rather than by a larger quantity. The proof is in the classical random oracle model; QROM security is left for future work.

4.10. Concrete Security Levels

Classical security level. Both attack paths are exhaustive searches over spaces of size 2^l: long-term key recovery via HMPP and ephemeral-secret recovery via MRP_2. The parameter set is assigned an l-bit search-based security level under the convention of Section 1.2, analogous to AES key search: $l = 128$ corresponds to Level I, $l = 192$ to Level III, and $l = 256$ to Level V. This is a definitional assignment from the search-space size and verification cost, not a computational lower bound. The polynomial factors L^2 (MRP) and $\text{poly}(L)$ (HMPP) are not counted in the security level.

Generic quantum cost of the best-known search attacks. The best-known attacks on HMPP and MRP both contain an unstructured search step, and Grover's algorithm applied to that step reduces the number of iterations to approximately $2^{l/2}$. This is a generic Grover cost on the search step alone. It does not account for the quantum cost of implementing the verification oracle (lattice reduction for HMPP, modular linear-system solving for MRP), nor for circuit depth, width, or memory. We therefore do not claim an established quantum security level; the numbers below are best read as the Grover-scaled cost of the classical search step. We emphasize that $l/2$, not l, is the binding exponent in this Grover-scaled estimate. A full quantum resource estimation is left for future work.

Assumption dependencies. Key recovery depends on Assumption 1 (Theorem 2); message recovery is defined by MRP (Assumption 4); the pairwise collision-resistance property is Assumption 5, with its adaptive form reduced through Structured D-HMPP (Lemma 3); IND-CCA2 depends on four of the five assumptions (Theorem 3), with standalone D-HMPP (Assumption 2) supporting the cryptanalysis.

4.11. Security Summary

D-HPPK rests on five assumptions: HMPP, D-HMPP, Structured D-HMPP, MRP (in both MRP_1 and MRP_2 forms), and pairwise decryption-collision resistance. Key recovery

costs $O(2^l \cdot \text{poly}(L))$ under HMPP; message recovery is defined by MRP; marginal uniformity of the decryption output is proved from the hidden-ring modular reduction; the pairwise collision property is assumed, with its adaptive form reduced through Structured D-HMPP; and IND-CCA2 security follows from a union bound over both attack paths in the classical random oracle model, conditional on four of the five assumptions. The five-class rejection set ensures the collision bound applies to every remaining oracle query. The security level is l bits under the search-based convention, determined by the size $|\mathbb{F}_p| = 2^l$ of the search space and the $O(L^2)$-time verification.

4.12. Open Problems

The security claim of D-HPPK is conditional on five assumptions, four of which are hardness statements about the underlying problems: HMPP, D-HMPP, Structured D-HMPP, and MRP. None of these is proved, and the security proof is conditional on all of them holding. The construction itself does not depend on their resolution: the scheme is well-defined and the correctness proof stands independently. But the assigned security level is meaningful only if the assumptions are, and the two most consequential open questions are therefore the hardness of the two problems on which the two attack paths depend.

1. **Is HMPP hard?** A polynomial-time classical or quantum algorithm that recovers (S, R) from the public samples $A_i \equiv R\tilde{a}_i \pmod{S}$ would break long-term key recovery and hence the scheme. HMPP is related in structure to the Approximate Common Divisor problem, and the strongest known attack family in that setting is the full-sample lattice technique [35,36]; whether it transfers to HMPP, and with what complexity, is open. A reduction of HMPP to a well-studied lattice problem would provide an independent foundation for the assigned security level; a polynomial-time algorithm would invalidate it.
2. **Is MRP hard?** A polynomial-time classical or quantum algorithm that recovers the ephemeral secret x^* from $(\mathbf{PK}, C^*)$ would break ephemeral-secret recovery and hence the scheme. MRP is a scheme-induced search problem whose instances are modular polynomial systems with the coefficient distribution of D-HPPK; the best-known attack is exhaustive search over $\mathbb{F}_p$ with an integer verification per candidate. Whether the algebraic structure of the resulting MMPS family admits an algorithm that reduces the search space below 2^l is open. The question applies to both forms of MRP (MRP_1 and MRP_2).

The cryptanalysis in Sections 4.5 and 4.6 constitutes the current evidence base for the hardness of HMPP and MRP respectively. We invite independent cryptanalysis and ask to be notified of any findings so that the security claims can be updated.

5. Implementation and Performance

This section reports the performance of the D-HPPK reference implementation on a single platform. Sizes and operation counts, which are implementation-independent, are also reported.

5.1. Parameter Selection

We fix $n = 2$ and $m = 3$; these choices ensure unique decryption, minimize the public key size, and enable efficient finite-field evaluation. The hidden moduli satisfy $S_1, S_2 > 9p^2$. Both key recovery and ephemeral-secret recovery have best-known classical cost $O(2^l \cdot \text{poly}(L))$, where $l = \lceil \log_2 p \rceil$ and $L = 2l + 8$. The resulting key and ciphertext sizes are summarized in Table 3.

Table 3. D-HPPK concrete parameters. The security level is $l = \lceil \log_2 p \rceil$ bits, assigned under the NIST search-based convention of Section 1.2.

	Level I	Level III	Level V
$l = \lceil \log_2 p \rceil$ (bits)	128	192	256
$L = 2l + 8$ (bits)	264	392	520
Field element size (B)	16	24	32
Hidden-ring element size (B)	33	49	65
Public key size (B)	626	930	1234
Secret key size (B)	196	292	388
Ciphertext size (CPA, B)	100	148	196
Ciphertext size (CCA2, B)	200	296	392

The shared secret is derived as $K = H(x \parallel C^{(1)} \parallel C^{(2)})$, where H outputs 256 bits.

Parameter selection uses the best currently analyzed HMPP attack cost $O(2^l \cdot \mathrm{poly}(L))$, conditional on the ACD-template heuristic, with the elementary GCD fallback at $O(2^{2l} \cdot \mathrm{poly}(L))$ providing additional margin.

The choice of $m = 3$. The parameter m is the number of noise variables in $B(x, \mathbf{u})$ and determines the kernel dimension of δ as $m - 1$. At $m = 3$, the kernel is two-dimensional and the MMPS system is uniquely solvable for each x. Increasing m grows the public key linearly in m and the per-candidate verification cost as $O(m^3)$, while the best-known attack on MRP remains 2^l. We therefore adopt $m = 3$ as the minimal choice.

5.2. Reference Implementation and Platform

We implemented a reference implementation of the D-HPPK IND-CCA2 KEM in C using GMP for large integer arithmetic and OpenSSL for SHA-256. The reference implementation performs the double encryption of Section 3.4, the consistency check, and the rejection handling for Classes 1 and 2 of Section 4.8. All measurements were obtained on a single platform: 12th Generation Intel Core i5-1240P (1.70 GHz base, 4.40 GHz turbo), Windows 11 64-bit, 16 GB RAM, GCC with `-O2 -march=native`. Measurements are averages over 10,000 iterations after a 1000-iteration warm-up; standard deviations across iterations were below 5% of the mean. The implementation has not been subjected to architecture-specific optimizations. It is an academic illustration of the construction, not a production-grade or constant-time implementation and not intended as a performance benchmark; see Section 5.4.

5.3. Operation Counts and Cycle Counts

The computational cost of D-HPPK is dominated by modular arithmetic in $\mathbb{F}_p$ and multiplication of L-bit integers. Table 4 summarizes the operation counts for the unified parameter set $(n = 2, m = 3)$.

Table 4. Field-operation counts for D-HPPK IND-CCA2 KEM $(n = 2, m = 3)$. The counts are algorithm-independent.

Operation	KeyGen	Encap	Decap
Random number generations	23	5	0
Modular multiplications (over $\mathbb{F}_p$)	77	18	8
Large integer multiplications	18	18	2
Large integer modular inversions	2	0	0
Hashing (SHA-256)	0	1	1

Table 5 reports the measured cycle counts, in k-cycles. Figure 4 plots the same data.

Table 5. D-HPPK IND-CCA2 KEM performance (k-cycles) on the reference platform. Averages over 10,000 iterations after a 1000-iteration warm-up; standard deviations below 5%.

Parameter	KeyGen	Encap	Decap
$l = 128$	23.7	6.6	11.0
$l = 192$	32.9	8.2	14.9
$l = 256$	41.7	9.3	18.0

At $l = 256$, the reference implementation takes approximately 9300 cycles for encapsulation and 18,000 cycles for decapsulation, corresponding to roughly 5.5 µs and 10.6 µs at 1.7 GHz. Scaling across security levels is roughly linear in l.

Scaling. The algorithmic structure of D-HPPK is identical across security levels; only l and $L = 2l + 8$ change. Total cost scales as $O(l^3)$ under schoolbook multi-precision arithmetic and $O(l^{2+\epsilon})$ with subquadratic multiplication. At very large l ($l > 1024$), the $O(l^2)$ multiplication becomes the dominant bottleneck. For $l \in [128, 512]$, this is not a constraint.

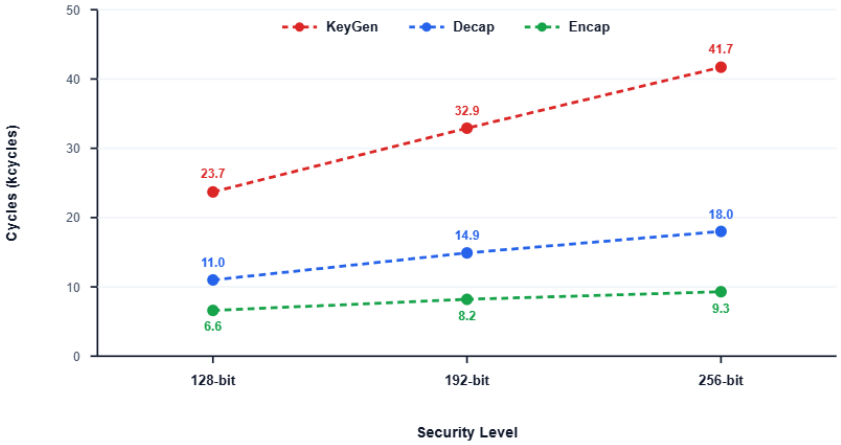


Figure 4. Cycle counts (k-cycles) for D-HPPK IND-CCA2 KEM across security levels. Values are the same as in Table 5.

5.4. Implementation Status and Constant-Time Roadmap

The current implementation is a prototype for correctness and performance indication. It uses GMP for large integer arithmetic and is not constant-time. We document below the specific operations that would require hardening in a constant-time reference implementation, together with the standard techniques that would be used.

Decapsulation (security-critical path). The six operations that involve secret values are: (i) reduction of c_p, c_q modulo S_1, S_2 (Barrett or Montgomery); (ii) multiplication by R_1^{-1}, R_2^{-1} (fully unrolled routine with no early termination); (iii) modular inversion of b in $\mathbb{F}_p$ (fixed-window exponentiation b^{p-2}); (iv) the division for $x = \Phi^{-1}(k)$; (v) the consistency check $x_1 = x_2$ (branch-free selector via $(x_1 - x_2)^{p-1}$); and (vi) the five-class rejection comparisons, including the Class 3 divisibility check and the Class 5 constant-time gcd of c_p with S_1 and c_q with S_2.

Key generation. Rejection sampling of S_1, S_2; sampling of R_1, R_2 with fixed-time coprimality testing; modular inversion of R_1, R_2; and the computation of $\hat{P}_{ij}, \hat{Q}_{ij}$ with constant-time modular multiplication.

Encapsulation. In the standard KEM threat model, encapsulation is executed by the sender on a trusted machine. If the sender's platform is compromised, the shared secret could leak through timing, but this is outside the standard IND-CCA2 model.

The techniques listed are standard and have been implemented in production libraries such as BearSSL, BoringSSL, and libsodium. The work required for D-HPPK is engineering rather than research: adapting known constant-time primitives to the specific multi-precision bit-lengths used here ($L = 2l + 8$ bits). We leave this implementation work for a subsequent revision, and we state clearly that the current prototype should not be deployed as-is.

Implementation-level attacks (timing, cache, fault injection, power analysis) are outside the scope of this work.

6. Conclusions

We have introduced D-HPPK, the **Defactorized Homomorphic Polynomial Public Key** key encapsulation mechanism, a post-quantum cryptosystem whose security rests on five hardness assumptions: HMPP, its decisional form D-HMPP, Structured D-HMPP, the scheme-induced MRP in its single- and double-ciphertext forms, and a scheme-induced pairwise decryption-collision resistance assumption. The scheme combines a three-layer architecture of additive random masking and hidden-ring embeddings: the hidden rings prevent the public-constraint attack, and the random masking destroys the rank-2 algebraic regularity that the original HPPK construction exposed.

D-HPPK is a search-based cryptosystem in the sense of AES: its security level is assigned from the size of the search space and the efficiency of verification, following the NIST convention. Both key recovery and ephemeral-secret recovery have best-known classical cost $O(2^l \cdot \mathrm{poly}(L))$, and the IND-CCA2 proof rests on a union bound over both paths, conditional on four of the five assumptions. The consistency check relies on the pairwise decryption-collision assumption, whose adaptive form is reduced through Structured D-HMPP; the marginal uniformity of the decryption output is proved from the hidden-ring modular reduction. At Level V, D-HPPK achieves a 1234-byte public key and a 392-byte CCA2 ciphertext, more compact than lattice-based KEMs at comparable security levels, with a single implementation scaling across security levels without SIMD or NTT.

The IND-CCA2 proof is not established in the quantum random oracle model, and further cryptanalysis of HMPP, MRP, and the pairwise collision assumption is invited. Directions for future research: further cryptanalysis of the two hardness problems; a proof of the two-dimensional equidistribution lemma or a reduction of Structured D-HMPP to the standalone form, either of which would reduce the assumption count; a constant-time implementation; a quantum random-oracle proof; and adaptation of the conjugated masking architecture to other primitives.

D-HPPK establishes that polynomial-based cryptography can simultaneously deliver compact representations and a simple algorithmic structure, offering a transparent alternative to lattice- and code-based post-quantum KEMs.

Funding: The APC was funded by Quantropi Inc.

Data Availability Statement: The reference implementation for the D-HPPK IND-CCA2 KEM is available at https://github.com/quantropi/dhppk-kem-jcp accessed on 1 October 2026. It is a feasibility implementation, not a performance benchmark.

Acknowledgments: The author used AI language models (DeepSeek-V3 and Gemini 2.5 Pro) for language refinement and as a discussion aid during the revision process. Every scientific claim, proof, cryptanalytic assessment, and final conclusion in this work was verified, decided, and written by the author. The author also thanks the anonymous reviewers for their detailed and constructive feedback, which substantially improved the rigor of the security analysis.

Conflicts of Interest: R.K. is the founder and Chief Scientist of Quantropi Inc., a company developing post-quantum cryptographic technologies. The author declares no other conflicts of interest.

References

1. Kuang, R.; Perepechaenko, M. Homomorphic polynomial public key encapsulation over two hidden rings for quantum-safe key encapsulation. *Quantum Inf. Process.* **2023**, *22*, 315. [CrossRef]
2. Kuang, R.; Perepechaenko, M.; Barbeau, M. A New Post-Quantum Multivariate Polynomial Public Key Encapsulation Algorithm. *Quantum Inf. Process.* **2022**, *21*, 360. [CrossRef]
3. Matsumoto, T.; Imai, H. Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In *Proceedings of the Advances in Cryptology—EUROCRYPT '88*; Barstow, D., Brauer, W., Brinch Hansen, P., Gries, D., Luckham, D., Moler, C., Pnueli, A., Seegmüller, G., Stoer, J., Wirth, N., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 1988; pp. 419–453. [CrossRef]
4. Patarin, J. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In *Proceedings of the Advances in Cryptology—EUROCRYPT '96*; LNCS; Maurer, U., Ed.; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1070, pp. 33–48. [CrossRef]
5. Ding, J. A New Variant of the Matsumoto-Imai Cryptosystem through Perturbation. In *Proceedings of the Public Key Cryptography—PKC 2004*; LNCS; Bao, F., Deng, R., Zhou, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 2947, pp. 305–318. [CrossRef]
6. Ding, J.; Yang, B.Y. Multivariate Public Key Cryptography. In *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 193–241. [CrossRef]
7. Ding, J.; Schmidt, D. Rainbow, a New Multivariable Polynomial Signature Scheme. In *Proceedings of the Applied Cryptography and Network Security*; LNCS; Ioannidis, J., Keromytis, A.D., Yung, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3531, pp. 164–175. [CrossRef]
8. Beullens, W. Breaking Rainbow Takes a Weekend on a Laptop. Cryptology ePrint Archive, Report 2022/214. 2022. Available online: https://eprint.iacr.org/2022/214 (accessed on 19 August 2026).
9. Johnson, D. Evolutions in Knapsack Cryptology. Ph.D. Thesis, Carleton University, Ottawa, ON, Canada, 2025. [CrossRef]
10. Merkle, R.C. Secure communications over insecure channels. *Commun. ACM* **1978**, *21*, 294–299. [CrossRef]
11. Shor, P.W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134. [CrossRef]
12. National Institute of Standards and Technology. *Post-Quantum Cryptography Standardization: Minimum Acceptability Requirements*; NIST Computer Security Resource Center: Gaithersburg, MD, USA, 2017. Available online: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/minimum-acceptability-requirements (accessed on 19 August 2026).
13. National Institute of Standards and Technology. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. ML-KEM (Formerly CRYSTALS-Kyber). 2024. Available online: https://csrc.nist.gov/pubs/fips/203/final (accessed on 19 August 2026).
14. National Institute of Standards and Technology. FIPS 204: Module-Lattice-Based Digital Signature Standard. ML-DSA (Formerly CRYSTALS-Dilithium). 2024. Available online: https://csrc.nist.gov/pubs/fips/204/final (accessed on 19 August 2026).
15. Fouque, P.A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU. Specification v1.2, NIST PQC Round 3 Submission. 2020. Available online: https://falcon-sign.info/falcon.pdf (accessed on 19 August 2026).
16. Aumasson, J.P.; Bernstein, D.J.; Beullens, W.; Dobraunig, C.; Eichlseder, M.; Fluhrer, S.; Gazdag, S.L.; Hülsing, A.; Kampanakis, P.; Kölbl, S.; et al. SPHINCS+: Stateless Hash-Based Signatures. NIST Post-Quantum Cryptography Standardization Round 3 Submission, v3.1. 2022. Available online: https://sphincs.org/data/sphincs+-r3.1-specification.pdf (accessed on 19 August 2026).
17. National Institute of Standards and Technology. FIPS 205: Stateless Hash-Based Digital Signature Standard. SLH-DSA (formerly SPHINCS+). 2024. Available online: https://csrc.nist.gov/pubs/fips/205/final (accessed on 19 August 2026).
18. McEliece, R.J. A Public-Key Cryptosystem Based on Algebraic Coding Theory. DSN Progress Report 42–44, 114–116. 1978. Available online: https://tda.jpl.nasa.gov/progress_report/42-44/44N.PDF (accessed on 19 August 2026).
19. Albrecht, M.R.; Bernstein, D.J.; Chou, T.; Cid, C.; Gilcher, J.; Lange, T.; Maram, V.; von Maurich, I.; Misoczki, R.; Niederhagen, R.; et al. Classic McEliece: Conservative Code-Based Cryptography. NIST Post-Quantum Cryptography Standardization Round 3 Submission. 2020. Available online: https://classic.mceliece.org (accessed on 19 August 2026).
20. Bernstein, D.J.; Chuengsatiansup, C.; Lange, T.; van Vredendaal, C. NTRU Prime: Round 2. In *Technical Report NIST PQC Standardization Process*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2019. Available online: https://ntruprime.cr.yp.to/nist/ntruprime-20190330.pdf (accessed on 19 August 2026).
21. Jao, D.; Azarderakhsh, R.; Campagna, M.; Costello, C.; Feo, L.D.; Hess, B.; Jalali, A.; Koziel, B.; LaMacchia, B.; Longa, P.; et al. Supersingular Isogeny Key Encapsulation (SIKE). NIST Post-Quantum Cryptography Standardization Round 2 Submission. 2020. Available online: https://sike.org (accessed on 19 August 2026).

22. Castryck, W.; Decru, T. An Efficient Key Recovery Attack on SIDH (Preliminary Version). Cryptology ePrint Archive, Report 2022/975. 2022. Available online: https://eprint.iacr.org/2022/975 (accessed on 19 August 2026).
23. Faugère, J.C.; Levy-dit Vehel, F.; Perret, L. Cryptanalysis of MinRank. In *Proceedings of the Advances in Cryptology—CRYPTO 2008*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5157, pp. 280–296. [CrossRef]
24. Faugère, J.C. Solving Structured Polynomial Systems and Applications to Cryptology. In *Proceedings of the Computer Algebra in Scientific Computing (CASC 2009)*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5743, pp. 79–80. [CrossRef]
25. Petit, C.; Quisquater, J.J. On Polynomial Systems Arising from a Weil Descent. In *Proceedings of the Advances in Cryptology—ASIACRYPT 2012*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7658, pp. 451–466. [CrossRef]
26. Boneh, D.; Lipton, R.J. Algorithms for Black-Box Fields and Their Application to Cryptography. In *Proceedings of the Advances in Cryptology—CRYPTO '96*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1109, pp. 283–297. [CrossRef]
27. Coppersmith, D. Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *J. Cryptol.* **1997**, *10*, 233–260. [CrossRef]
28. Selim, A.; Saračević, M.; Ćatović, A. Homomorphic Cryptographic Scheme Based on Nilpotent Lie Algebras for Post-Quantum Security. *Symmetry* **2025**, *17*, 1666. [CrossRef]
29. Fujisaki, E.; Okamoto, T. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Proceedings of the Advances in Cryptology—CRYPTO '99*; LNCS; Wiener, M., Ed.; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1666, pp. 537–554. [CrossRef]
30. Hofheinz, D.; Hövelmanns, K.; Kiltz, E. A Modular Analysis of the Fujisaki-Okamoto Transformation. In *Proceedings of the 15th Theory of Cryptography Conference (TCC 2017)*; LNCS; Katz, J., Shacham, H., Eds.; Springer: Cham, Switzerland, 2017; Volume 10677, pp. 341–371. [CrossRef]
31. Jiang, H.; Zhang, Z.; Chen, L.; Wang, H.; Ma, Z. IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited. In *Proceedings of the Advances in Cryptology—CRYPTO 2018*; LNCS; Shacham, H., Boldyreva, A., Eds.; Springer: Cham, Switzerland, 2018; Volume 10993, pp. 96–125. [CrossRef]
32. Jiang, S.; Wang, H. Plaintext-Awareness of Hybrid Encryption. In *Proceedings of the Topics in Cryptology—CT-RSA 2010*; LNCS; Pieprzyk, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 5985, pp. 57–72. [CrossRef]
33. Seurin, Y.; Treger, J. A Robust and Plaintext-Aware Variant of Signed ElGamal Encryption. In *Proceedings of the Topics in Cryptology—CT-RSA 2013*; LNCS; Dawson, E., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7779, pp. 381–397. [CrossRef]
34. Chen, K.; Miyaji, A.; Chen, J. Generic CCA Secure Key Homomorphic KEM and Updatable Public Key Encryption. In *Proceedings of the Information Security Practice and Experience—19th International Conference, ISPEC 2024, Wuhan, China, 25–27 October 2024*; LNCS; Xia, Z., Chen, J., Eds.; Springer: Cham, Switzerland, 2024; Volume 15053, pp. 168–185. [CrossRef]
35. Cohn, H.; Heninger, N. Approximate Common Divisors via Lattices. Cryptology ePrint Archive, Report 2011/437. 2011. Available online: https://eprint.iacr.org/2011/437 (accessed on 19 August 2026).
36. Galbraith, S.D.; Gebregiyorgis, S.W.; Murphy, S. Algorithms for the Approximate Common Divisor Problem. Cryptology ePrint Archive, Report 2016/215. 2016. Available online: https://eprint.iacr.org/2016/215 (accessed on 19 August 2026).
37. Xu, J.; Sarkar, S.; Hu, L. Revisiting Orthogonal Lattice Attacks on Approximate Common Divisor Problems and their Applications. Cryptology ePrint Archive, Report 2018/1208. 2018. Available online: https://eprint.iacr.org/2018/1208 (accessed on 19 August 2026).
38. Notarnicola, L.; Wiese, G. The Hidden Lattice Problem. Cryptology ePrint Archive, Report 2021/1491. 2021. Available online: https://eprint.iacr.org/2021/1491 (accessed on 19 August 2026).
39. Grover, L.K. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*; ACM: New York, NY, USA, 1996; pp. 212–219. [CrossRef]
40. Howgrave-Graham, N. Approximate Integer Common Divisors. In *Proceedings of the Cryptography and Lattices (CaLC 2001)*; LNCS; Silverman, J.H., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2146, pp. 51–66. [CrossRef]
41. Wigert, S. Sur l'ordre de grandeur du nombre des diviseurs d'un entier. *Ark. Mat. Astron. Fys.* **1907**, *3*, 1–9.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.