QPP and HPPK: Unifying Non-Commutativity for Quantum-Secure Cryptography with Galois Permutation Group

-A Tribute to Évariste Galois (25.10.1811 – 31.05.1832)

Randy Kuang[1][0000−0002−5567−2192]

Quantropi Inc., 1545 Carling Av., Suite 620, Ottawa, Canada
randy.kuang@quantropi.com

Abstract. In response to the evolving landscape of quantum computing and the heightened vulnerabilities in classical cryptographic systems, our paper introduces a comprehensive cryptographic framework. Building upon the pioneering work of Kuang et al., we present a unification of two innovative primitives: the Quantum Permutation Pad (QPP) for symmetric key encryption and the Homomorphic Polynomial Public Key (HPPK) for Key Encapsulation Mechanism (KEM) and Digital Signatures (DS). By harnessing matrix representations of the Galois Permutation Group and inheriting its bijective and non-commutative properties, QPP achieves quantum-secure symmetric key encryption, seamlessly extending Shannon's perfect secrecy to both classical and quantum-native systems. Simultaneously, HPPK, free of NP-hard problems, relies on the security of symmetric encryption for the plain public key. This is accomplished by concealing the mathematical structure through arithmetic representations or modular multiplicative operators (arithmetic QPP) of the Galois Permutation Group over hidden rings, utilizing their partial homomorphic properties. This ensures secure computation on encrypted data during secret encapsulations, thereby enhancing the security of the plain public key. The integration of KEM and DS within HPPK cryptography results in compact key, cipher, and signature sizes, showcasing exceptional performance. This paper organically unifies QPP and HPPK under the Galois Permutation Group, marking a significant advance in laying the groundwork for quantum-resistant cryptographic protocols. Our contribution propels the development of secure communication systems in the era of quantum computing.

Keywords: Cryptography · Quantum Cryptography · Shannon Perfect · Galois Permutation Group · QPP · HPPK · KEM · Digital Signature.

1 Introduction

Quantum Key Distribution (QKD) stands at the forefront of modern cryptographic protocols, leveraging the foundational principles of quantum mechanics to establish secure communication in the face of emerging quantum computers [40]. Unlike classical methods relying on mathematical intricacies [36, 6,

29], QKD exploits unique quantum properties, particularly in photons. At its core, QKD relies on quantum superposition and entanglement, allowing cryptographic key exchange while promptly detecting eavesdropping attempts. Security is rooted in quantum indeterminacy, where measuring a quantum state disrupts it, revealing eavesdropping attempts. In the era of quantum computing threats to classical cryptography, QKD emerges as a promising secure communication avenue. Kuang and Barbeau demonstrated QKD photonic implementations, involving identity and XOR permutations within the Galois permutation group over the finite field $\{0, 1\}$ [14], framing QKD as a physical realization of Shannon's OTP scheme [37]. The natural non-commutativity of the Galois permutation group favors digital QPP implementations over physical QKD with more than one qubit [25]. Digital QKD becomes an economical, deployable, and scalable alternative for quantum-secure internet communication [32, 25, 26, 18, 15–17]. Kuang and Perepechaenko also demonstrated their QPP implementations in the native quantum computing system [19, 33, 34].

On the asymmetric cryptography front, Kuang in 2021 introduced Deterministic Polynomial Public Key (DPPK) [11], later enhanced by Kuang and Barbeau in 2021 as Multivariate Polynomial Public Key (MPPK) [13, 12]. To bolster MPPK security, Kuang, Perepechaenko, and Barbeau in 2022 partially encrypted MPPK public key over a hidden ring [22], using arithmetic permutations for security. In 2023, Kuang and Perepechaenko proposed variants with full encryption over two separate hidden rings [20] and a single hidden ring [10]. Concurrently, they introduced a digital signature, MPPK/DS, in 2022 [23], later optimized in 2023 [21], though Guo reported a forged signature in early 2023 [9]. Unable to rectify the inherent linear relationship in MPPK/DS, they recently extended the HPPK KEM scheme for digital signatures [24].

This paper unifies symmetric encryption with QPP and asymmetric HPPK under the Galois permutation group's umbrella. Leveraging non-commutativity in matrix representation (QPP) and arithmetic permutations over hidden rings (HPPK), we bridge both cryptographic realms. Related works will be discussed in Section 2, followed by the Galois permutation group's representation in Section 3. Section 4 covers symmetric cryptography with QPP, Section 5 addresses HPPK for asymmetric cryptography, and Section 6 provides a security brief. The conclusion is drawn in the final section.

2 Related Works

The realm of Post-Quantum Cryptography (PQC) encompasses a diverse array of standardized schemes outlined by the National Institute of Standards and Technology (NIST). This overview succinctly encapsulates notable schemes, categorized according to their cryptographic underpinnings.

For Key Encapsulation Mechanism (KEM), lattice-based contenders such as Kyber [2], BIKE [30], HQC [4], and code-based McEliece [28] take the spotlight. Additionally, in the domain of Digital Signatures (DS), lattice-based Falcon [35], Dilithium [27], and hash-based SPHINCS$^+$ [1] emerge as prominent choices.

In a significant development in 2022, NIST announced standardized algorithms [31], endorsing Kyber for KEM and propelling McEliece, BIKE, and HQC into round 4. Simultaneously, NTRU [3] and Saber [5] were excluded from further consideration, while novel submissions for generic digital signature schemes were introduced [31].

Lattice-based algorithms, exemplified by Kyber, BIKE, HQC, and Falcon, typically hinge on the Short-Vector Problem (SVP) as the linchpin of their security. Code-based algorithms, as showcased by McEliece, derive security from the intricate decoding of random linear codes, providing robust post-quantum security. Hash-based algorithms, as exemplified by SPHINCS$^+$, are constructed based on the security of one-way trapdoors in hash functions. These NP-hard problems lay the groundwork for security against the looming threat of quantum computing. In a departure from this trend, HPPK cryptography takes a distinctive approach, relying on the security of symmetric encryption, offering a unique and innovative trajectory in the landscape of post-quantum cryptographic solutions.

3 Representations of Galois Permutation Group

The Galois Permutation Group over a finite field extension $\mathbf{F}_{2^n}$ with n bits plays a crucial role in finite field theory, particularly in cryptography and algebraic coding theory. This section explores various methods for representing elements within this group and elucidates their significance. The primary focus is on two representations of the Galois permutation group: matrix and arithmetic.

3.1 Matrix Representations

While our focus is on permutations over $\mathbf{F}_{2^n}$, binary unitary matrices serve as effective tools to represent the actions of Galois Group elements. These matrices facilitate the rearrangement of integers within the finite field, showcasing the reversible nature of Galois Group transformations. Key properties of these matrix representations include:

- Bijectiveness: Permutation representations highlight discrete actions on the integer set, capturing fundamental rearrangements induced by Galois Group elements. The bijective nature ensures a unique correspondence between the initial and final integer arrangements.
- Composition of Permutations: Group operations involve the composition of permutations, representing the sequential application of rearrangements. The properties of composition unveil overall symmetries and transformations within the Galois Permutation Group.
- Non-commutativity: Group operations or operators $\hat{P}_i$ and $\hat{P}_j$ generally meet $\hat{P}_i\hat{P}_j \neq \hat{P}_j\hat{P}_i$, adding a critical layer of security to cryptographic operations. This property, especially significant in symmetric-key cryptography where permutation operators are secret keys, implies that the order of permutation application matters, contributing complexity and enhancing security.

In the quantum realm, the non-commutativity of the Galois Permutation Group directly corresponds to the uncertainty principle in quantum mechanics. Therefore, this property of the Galois permutation group plays a crucial role in achieving perfect secrecy for both classical and quantum domains.

3.2 Arithmetic Representations

In the realm of finite fields, arithmetic permutations play a pivotal role in cryptographic operations. These permutations, often expressed through modular arithmetic, contribute to the non-commutative and intricate transformations inherent in the Galois Permutation Group. This subsection delves into some critical arithmetic permutations that form the foundation of cryptographic schemes.

XOR Operation In a binary finite field ($GF(2^n)$), the XOR operation (bitwise addition modulo 2) is a fundamental arithmetic operation. XORing two bits results in a bit set if the two input bits are different and cleared if they are the same. There are a total of 2^n permutations over ($GF(2^n)$) for XOR operations, proven to be the Shannon perfect scheme if the random key is used only once, leading to the concept of the One-Time-Pad (OTP).

Addition Operation Modular addition, expressed as $(a + b) \bmod 2^n$, is a fundamental arithmetic operation within finite fields. This operation involves adding two integers, a and b, and then taking the remainder when divided by a modulus 2^n. There are a total of 2^n permutations over ($GF(2^n)$) for this type of addition operations.

Modular Multiplication In the realm of modular arithmetic, the operation of modular multiplication, denoted as $((R \cdot b) \bmod S)$, holds significant importance when R and S form a coprime pair, with S being L-bits. This operation acts as a fundamental arithmetic permutation, rearranging residues within the ring.

The security of this permutation, particularly when S is kept secret, relies on the computational challenge posed by the brute-force search for the modulus S. In scenarios where S is public, the number of potential coprime pairs (R, S) is given by Euler's totient function, denoted as $\varphi(S)$. However, if S is confidential and possesses a known bit length L, the total count of potential arithmetic permutations becomes $\varphi(2^L)2^L$.

Modular Exponentiation The operation of modular exponentiation, expressed as $(a^b \bmod N)$, entails the iterative application of modular multiplications and stands as a potent arithmetic permutation. This permutation forms the basis of cryptographic systems such as RSA and Diffie-Hellman.

In RSA, the public key is represented by b and N, where N is the product of two large primes ($N = pq$). The security of RSA hinges on the computational complexity of the prime factorization problem. On the other hand, in the case

of Diffie-Hellman cryptography, the public key comprises a and N. Security is grounded in the difficulty of the discrete logarithm problem, requiring the knowledge of $c = a^b \bmod N$ and a to deduce b. Both the prime factorization problem and the discrete logarithm problem present significant computational challenges in classical computing. Shor's algorithm has demonstrated that the advent of quantum computing could render these cryptographic schemes vulnerable [39].

3.3 Classical Key Space to Quantum Key Space: Galois Permutation Group over $\mathbf{F}_{2^n}$

In the finite field, an n-bit key space refers to a finite field extension $\mathbf{F}_{2^n}$ with possible 2^n keys, each being an n-bit integer within $[0, 2^n)$. However, the Galois permutation group has an order of $2^n!$ permutation operators, operating on the finite field set in quantum computing. This makes the Galois permutation group the key space for quantum computing. The quantum key space holds an entropy of $e = log_2(2^n!) \approx (n - 0.42)2^n$ (for larger n), once retrieving those key operators with an equally-likely distribution.

3.4 Relevance to QPP and HPPK

The concept of the Shannon perfect OTP seamlessly extends to QPP, where each element is randomly selected from the Galois permutation group.

Additionally, the arithmetic permutation inherent in reversible modular multiplication presents the groundwork for another potential asymmetric cryptographic scheme, especially when the modulus S remains a secret. This notion forms the basis of HPPK, which will be explored in detail later.

By leveraging QPP and HPPK, we have the opportunity to unify symmetric and asymmetric cryptography within a cohesive framework provided by the Galois permutation group, with its matrix representation and arithmetic representation for symmetric and asymmetric schemes respectively.

4 Symmetric Cryptography: QPP with Matrix Permutations

The extensive work of Kuang and Barbeau in 2022 [14] on Quantum Permutation Pad (QPP) cryptography is succinctly summarized in this section, with a predominant focus on the matrix representation of the Galois permutation group.

4.1 QPP Generation

In the context of an n-bit finite field extension $\mathbf{F}_{2^n}$, the classical key is initially generated as an n-bit binary string. However, in a quantum environment, this classical key undergoes transformation into permutation gates for implementation in quantum-native systems [19, 33] or permutation matrices for classical

implementations [14]. This transformation is facilitated by the Fisher and Yates algorithm [8].

Using the shuffling algorithm, a single permutation matrix is randomly selected with an input random key of size $n2^n$ bits. This algorithm leverages the random key string to induce shuffling on the ordered set of integers from $\mathbf{F}_{2^n}$. After shuffling, the integer set exists in a disordered state, enabling the creation of a binary mapping matrix derived from the classical key string.

The QPP generation process can be iterated to produce a set of permutation matrices, constituting the Quantum Permutation Pad (QPP) for symmetric encryption. The total classical key length is $M(n2^n)$ bits for a pad with M permutation matrices. The values of n and M are selected based on security requirements. Quantropi has developed its digital Quantum Key Distribution (QKD) platform with $n = 8$ and $M = 64$, delivering a total equivalent entropy of over 100,000 bits [25]. For a typical quantum-safe scenario with more than 256 bits of entropy, opting for $n = 4$ and $M = 8$ results in a total of 360 bits of entropy.

4.2 QPP Encryption

When employing QPP for encryption, information is represented in quantum computing format using Dirac ket notation $|i\rangle$ with $i = 0, 1, \ldots, 2^n - 1$ for an n-bit information. For example, an 8-bit string $''10001011''$ is expressed as $|139\rangle$. The complete set of $\mathbf{F}_{2^n}$ is denoted as $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$, referred to as a computational basis in quantum computing. In the computational basis, a state $|i\rangle$ is a vector expressed in terms of the entire basis vectors. Classical information is represented as a column vector with the row index corresponding to the decimal value of its bit string set to 1, and all other elements of the column vector set to zero. For instance, $|139\rangle$ is the 139^{th} basis vector in the 8-bit computational basis.

The encryption process follows the principles of quantum computing, as represented by the equation:

$$\hat{P}_i|m\rangle = |c\rangle \tag{1}$$

In this context, m signifies an n-bit plaintext in decimal form, and c signifies an n-bit ciphertext in decimal format. Upon transmission to a receiver, the decimal value is converted into a binary string.

In classical systems utilizing permutation matrices for implementation, the encryption process described in Eq. (1) designates the column index of the element $''1''$ at the m^{th} row in the permutation matrix $\hat{P}_i$ as the resulting ciphertext state $|c\rangle$.

4.3 QPP Decryption

The decryption process is equally straightforward, employing the reverse permutation operator $\hat{P}_i^{-1}$:

$$\hat{P}_i^{-1}|c\rangle = \hat{P}_i^{-1}\hat{P}_i|m\rangle = |m\rangle \tag{2}$$

This simplicity arises from the unitary and reversible properties inherent in permutation operators. Additionally, given that all elements of permutation matrices are either $''0''$ or $''1''$, the reverse permutation matrices are effectively their transposes: $\hat{P}_i^{-1} = \hat{P}_i^{\dagger}$. This characteristic significantly streamlines the implementation process, eliminating the need for matrix reversal operations. Once again, the decrypted decimal values are converted back to binary format to reconstruct the classical plaintext.

4.4 Confusion and Diffusion

Given the bijective mapping property of QPP, it possesses the capability to transform hidden structures in plaintexts into ciphertexts. To address this potential weakness, a common strategy involves pre-randomizing the plaintext to enhance both confusion and diffusion capabilities [14].

The pre-randomization process employs XOR operations with an n-bit plaintext and an n-bit sequence generated by a Pseudo-Random Number Generator (PRNG) seeded with a shared classical key. The resulting n-bit value is then dispatched to a selected permutation operator from the QPP pad based on an index generated by the PRNG. This random dispatching significantly further augments the diffusion capability, although a sequential dispatch, akin to AES block cipher, is also a viable alternative. Following the dispatching step, Eq. (1) comes into play for encryption.

On the decryption side, the dispatching step takes precedence to select the correct permutation operator, followed by the decryption using $\text{QPP}^{\dagger}$. Subsequently, the pre-randomizing step becomes a post-derandomizing operation aimed at retrieving the original plaintext.

4.5 Arithmetic QPP

Quantum Permutation Pad (QPP) typically requires a runtime storage of M permutation matrices or $Mn2^n$ bits for encryption and an equivalent storage for decryption. In resource-constrained scenarios, QPP can be reformulated in terms of arithmetic permutations, utilizing reversible modular multiplication denoted by $\hat{p}_i = R_i \cdot \square \bmod S_i$ with $\square$ representing the plaintext. The entropy of standard QPP, where matrices are randomly chosen, differs from that of arithmetic QPP: $e = M \log_2(2^n!) \longrightarrow e = M \log_2(\varphi(2^n)2^n)$. For example, in the case of $n = 8$, each randomly chosen permutation matrix holds 1684 bits of entropy, requiring 2048 bits of storage, whereas the entropy of arithmetic permutation is less than 15 bits, requiring 16 bits of storage. Nevertheless, the entropy of arithmetic QPP can be increased by enlarging the size of the QPP pad to meet our desired security level. Leveraging the security of symmetric encryption, arithmetic QPP proves more suitable for asymmetric cryptography, as exemplified in HPPK to be discussed.

4.6 Implementations of QPP into Quantum Computing Systems

Due to its simplicity, QPP cryptography lends itself to straightforward implementation in physical quantum computing systems, such as IBMQ. Demonstrations of its viability have been conducted with toy examples utilizing 2- and 3-qubits [19, 34, 32]. Different quantum computing systems employ distinct decomposition mechanisms to transform an n-bit gate into fewer qubit circuits, typically consisting of 1- or 2-qubit gates.

In our toy examples on the IBMQ 5-qubit system, we generate a QPP pad, utilizing its matrix forms as inputs in the source codes. The compilation of a 2-qubit permutation gate results in a quantum circuit with a gate depth of about 15 layers. Achieving a fidelity of 99%, we can confidently execute our encryption and decryption processes.

However, as we escalate to 4-qubit permutation gates, the compilation yields a circuit with a depth exceeding 100 layers. At this point, a 99% fidelity becomes inadequate for meaningful results. Nevertheless, QPP remains relatively straightforward to implement in a quantum computing system with high fidelity or a substantial number of logic qubits, allowing for direct processing of encryption and decryption within the quantum computing system.

QPP facilitates encrypted communications across quantum-quantum, quantum-classical, and classical-classical channels. Over quantum channels, encrypted qubits can be directly decrypted by the receiving quantum system. Over classical channels, cipher qubits must be measured, producing a ciphertext that is then transmitted to the receiving side for decryption, either by a quantum system or a classical system equipped with the same $\text{QPP}^\dagger$ from the shared classical key.

5 Asymmetric Cryptography: HPPK with Arithmetic QPP

Asymmetric cryptography relies on challenging computational problems, such as the prime factorization problem in RSA, the discrete logarithm problem in Diffie-Hellman, and various problems such as the SVP in schemes like Kyber [2], Falcon [35], Dilithium [27], and the Multivariate Quadratic Problem or MQ in Multivariate Public Key Cryptography (MPKC) [7]. The security of these schemes hinges on the computational difficulty of these intricate problems. However, emerging algorithms, including Shor's algorithm [39], pose a threat by potentially solving some problems efficiently in classical public key schemes.

Recently, Sharp et al. introduced a novel computing technology based on self-organized gates [38], capable of breaking RSA classically using GPUs or silicon implementations. This emphasizes the need to explore alternative approaches to asymmetric cryptography.

Kuang and colleagues introduced a novel approach to asymmetric cryptography, proposing schemes that leverage symmetric encryption techniques for both Key Encapsulation (KEM)[20] and Digital Signature (DS)[24]. The symmetric encryption employed in these schemes is a specialized form of arithmetic Quantum Permutation Pad (QPP) featuring two permutation operators or modular

multiplication over hidden rings. This introduces non-commutability through arithmetic permutations. The subsequent sections of this paper delve into specific aspects, with Section 5.1 exploring homomorphic properties, Section 5.2 detailing Key Encapsulation, Section 5.3 focusing on Digital Signature, and Section 5.4 elucidating the key triple combination of KEM and DS.

5.1 Homomorphic Properties of Arithmetic QPP

Arithmetic permutations or arithmetic QPP, derived from modular multiplication over hidden rings, play a pivotal role in constructing a new form of asymmetric cryptography. Unlike pure symmetric encryption schemes where communication peers possess a shared secret key, asymmetric schemes aim to establish the shared secret during the process. This usually involves a roundtrip of key pair generation, public key encryption, and private key decryption.

An intriguing approach leverages self-shared symmetric keys between the key pair generator and cipher decryptor. This key is used to encrypt the plain public key into a cipher public key, subsequently decrypting the received cipher into an interim cipher associated with the plain public key encryption. This unique asymmetric scheme relies on the security of the self-shared symmetric encryption key or arithmetic QPP, necessitating symmetric encryption with certain homomorphic properties.

The arithmetic permutation of modular multiplication over hidden rings exhibits partial homomorphic properties for addition and scalar multiplication. The permutation operator $\hat{E}(R, S)$ is defined as follows:

$$\hat{E}(R, S) = R \circ \square \bmod S \tag{3}$$

for a coprime pair of L-bit R and S, where $\square$ represents an integer or a polynomial function. Demonstrating its addition property, let's choose two integers a and b:

$$\begin{aligned} &\hat{E}(R, S)a = R * a \bmod S = a' \\ &\hat{E}(R, S)b = R * b \bmod S = b' \\ &\hat{E}(R, S)(a + b) = R * (a + b) \bmod S = \hat{E}(R, S)a + \hat{E}(R, S)b \end{aligned} \tag{4}$$

and verify its scalar multiplication with a constant c:

$$\hat{E}(R, S)ca = R * ca \bmod S = ca' \bmod S = c\hat{E}(R, S)a \tag{5}$$

Equations (4) and (5) clearly demonstrate the partial homomorphic properties of the modular multiplication operator $\hat{E}(R, S)$. If S is a public ring, there exist $\varphi(S)$ permutation operators over the ring $\mathbf{Z}_S$. If S is a hidden ring with a bit size L, there exist potential $\varphi(S)2^L$ arithmetical permutation operators, corresponding to a key space of size $\varphi(S)2^L$. In a symmetric encryption scheme, the brute force search has a complexity of $\mathcal{O}(\varphi(S)2^L)$.

The partial homomorphic properties of the modular multiplication operators naturally favor any polynomial $\eta(x, y_1, \ldots, y_m) = \sum c_i \eta_i(x, y_1, \ldots, y_m)$ where c_i

are coefficients and $\eta_i(x, y_1, \ldots, y_m)$ are pure monomials over a prime field $\mathbf{F}_p$. The permutation operator $\hat{E}(R, S)$ maps all coefficients c_i from $\mathbf{F}_p$ to $\mathbf{Z}_S$ and treats all monomials as scalars:

$$\begin{aligned} &\hat{E}(R,S)\eta(x,y_1,\ldots,y_m) = \sum (R * c_i \bmod S)[\eta_i(x,y_1,\ldots,y_m) \bmod p] \\ &\longrightarrow \eta'(x,y_1,\ldots,y_m) = \sum c_i'[\eta_i(x,y_1,\ldots,y_m) \bmod p] \end{aligned} \tag{6}$$

where the computations of all monomials must be with mod p then directly scalar multiplications with their cipher coefficients $c_i' \in \mathbf{Z}_S$. To make the polynomial value or ciphertext decryptable at the decryptor holding the symmetric key $\hat{E}(R, S)$, the ring size L must hold p^2 for each polynomial term of $\eta'(x, y_1, \ldots, y_m)$, so $L \geq 2\log_2 p + \log_2 T$ with T being the total polynomial terms. Once these conditions are met, the permutation operator $\hat{E}(R, S)$ holds the homomorphic property for polynomials.

5.2 HPPK KEM

Once the security is ensured through symmetric encryption using modular multiplicative permutations, designing the plain public key scheme becomes relatively straightforward. This involves utilizing two simplified multivariate polynomials over $\mathbf{F}_p$:

$$\begin{aligned} p(x,u_1,\ldots,u_m) &= B(x,u_1,\ldots,u_m)f(x) = \vec{x}^T \cdot \mathbf{p} \cdot \vec{u} \\ q(x,u_1,\ldots,u_m) &= B(x,u_1,\ldots,u_m)h(x) = \vec{x}^T \cdot \mathbf{q} \cdot \vec{u} \end{aligned} \tag{7}$$

Here, λ denotes the order of univariate polynomials $f(x)$ and $h(x)$, and n represents the order of the variable x in $B(x, u_1, \ldots, u_m)$, with u_i being linear variables denoting monomials $\eta_i(x, y_1, \ldots, y_m)$ without variable x. The vector $\vec{x}^T = (x^0, x^1, \ldots, x^\lambda)$ represents a vector in a polynomial vector space, and the vector $\vec{u} = (u_1, \ldots, u_m)$ represents a vector in a multidimensional vector space. Eliminating the common factor polynomial $B(x, u_1, \ldots, u_m)$ is achieved by modular division:

$$\frac{p(x,u_1,\ldots,u_m)}{q(x,u_1,\ldots,u_m)} = \frac{f(x)}{h(x)} \bmod p \tag{8}$$

The matrices $\mathbf{p}$ and $\mathbf{q}$ in Eq. (7) are coefficient matrices of two multivariate polynomials $p(x, u_1, \ldots, u_m)$ and $q(x, u_1, \ldots, u_m)$, respectively. They inherit the mathematical structures of polynomial multiplications over a prime field, making them challenging to secure based on computational difficulty. These are referred to as plain public keys. To encrypt them, arithmetical modular multiplicative permutation operators in Eq. (3) are applied. An arithmetic QPP or two coprime pairs (R_1, S_1) for $p(x, u_1, \ldots, u_m)$ and (R_2, S_2) for $q(x, u_1, \ldots, u_m)$ are randomly chosen from a equally-likely distribution, and the encryption is performed as follows:

$$\begin{aligned} P(x,u_1,\ldots,u_m) &= \vec{x}^T \cdot [\hat{E}(R_1,S_1)\mathbf{p}] \cdot \vec{u} = \vec{x}^T \cdot \mathbf{P} \cdot \vec{u} \\ Q(x,u_1,\ldots,u_m) &= \vec{x}^T \cdot [\hat{E}(R_2,S_2)\mathbf{q}] \cdot \vec{u} = \vec{x}^T \cdot \mathbf{Q} \cdot \vec{u} \end{aligned} \tag{9}$$

with cipher public key matrices given by:

$$\mathbf{P} = \hat{E}(R_1, S_1)\mathbf{p} = R_1 * \mathbf{p} \bmod S_1, \quad \mathbf{Q} = \hat{E}(R_2, S_2)\mathbf{q} = R_2 * \mathbf{q} \bmod S_2 \quad (10)$$

Here, $\hat{E}(R_1, S_1)$ is applied to all matrix elements of $\mathbf{p}$, and $\hat{E}(R_2, S_2)$ is applied to all matrix elements of $\mathbf{q}$. The key pair is obtained as follows:

- **Public Key** PK_e**:** $\mathbf{P}[n+\lambda+1][m]$ and $\mathbf{Q}[n+\lambda+1][m]$.
- **Private Key** SK**:** $f[\lambda+1]$, $h[\lambda+1]$, $R_1, S1$, and R_2, S_2.

where symbol PK_e refers to the public key for encapsulation.

Encapsulation Then using the public key $\mathbf{P}$ and $\mathbf{Q}$, an encryptor can generate a ciphertext of a secret x randomly chosen from $\mathbf{F}_p$. Here are steps:

- Encapsulation of x: choose random noise $u_1, \ldots, u_m \in \mathbf{F}_p$ then evaluate $x_{ij} = x^i u_j \bmod p$ for $i = 0, 1, \ldots, n+\lambda, j = 1, 2, \ldots, m$.
- Evaluations of polynomials: $\overline{P} = \sum_{i=0}^{n+\lambda} \sum_{j=1}^{m} P_{ij} x_{ij}, \overline{Q} = \sum_{i=0}^{n+\lambda} \sum_{j=1}^{m} Q_{ij} x_{ij}$
- Ciphertext: $CT = \{\overline{P}, \overline{Q}\}$

Decapsulation With receiving ciphertext $CT = \{\overline{P}, \overline{Q}\}$, the decrypter can perform first symmetric decryption then the secret extraction as follows:

- Symmetric decryption: $\bar{p} = (\frac{\overline{P}}{R_1} \bmod S_1) \bmod p$ and $\bar{q} = (\frac{\overline{Q}}{R_2} \bmod S_2) \bmod p$
- Noise elimination: $k = \frac{\bar{p}}{\bar{q}} \bmod p = \frac{f(x)}{h(x)} \bmod p$
- Secret extraction: $f(x) - kh(x) = 0$. For linear univariate polynomial, $x = \frac{kh_0 - f_0}{f_1 - kh_1} \bmod p$.

5.3 HPPK DS

In a digital signature scheme within the context of HPPK, the process begins with the signer generating the hash code $x \leftarrow HASH(M)$ using a chosen cryptographic hash function for the signing message M. Subsequently, the signer employs the private key to sign x. Here, we first introduce the signature and then formulates the verification equation. The signature is defined as follows:

$$\begin{aligned} F &= R_2^{-1} * [\alpha f(x) \bmod p] \bmod S_2 \\ H &= R_1^{-1} * [\alpha h(x) \bmod p] \bmod S_1 \end{aligned} \quad (11)$$

where α is a randomly chosen integer from $\mathbf{F}_p$. The subsequent step involves developing the HPPK verification equation using cross-multiplication of Eq. (7) and Eq. (8):

$$\vec{x}^T \cdot (\bar{f}\mathbf{q}) \cdot \vec{u} = \vec{x}^T \cdot (\bar{h}\mathbf{p}) \cdot \vec{u} \bmod p \quad (12)$$

Here, $\bar{f} = \alpha f(x) \bmod p$ and $\bar{h} = \alpha h(x) \bmod p$. Utilizing the unitary and reversible encryption operators from Eq. (3), Eq. (12) is transformed into:

$$\begin{aligned} \{\vec{x}^T \cdot (F\mathbf{Q} \bmod S_2) \cdot \vec{u}\} \bmod p &= \{\vec{x}^T \cdot (H\mathbf{P} \bmod S_1) \cdot \vec{u}\} \bmod p \\ \longrightarrow \{\vec{x}^T \cdot V \cdot \vec{u}\} \bmod p &= \{\vec{x}^T \cdot U \cdot \vec{u}\} \bmod p \\ \longrightarrow V(x, u_1, \ldots, u_m) \bmod p &= U(x, u_1, \ldots, u_m) \bmod p \end{aligned} \tag{13}$$

where matrices $V = F\mathbf{Q} \bmod S_2 \bmod p$ and $U = H\mathbf{P} \bmod S_1 \bmod p$. Since the unknowns S_1 and S_2 cannot be determined, a verifier cannot perform the verification based on Eq. (13). These variables must be eliminated from all coefficients $V_{ij} = (FQ_{ij} \bmod S_2) \bmod p$ in polynomial $V(x, u_1, \ldots, u_m)$ and $U_{ij} = (HP_{ij} \bmod S_1) \bmod p$ in polynomial $U(x, u_1, \ldots, u_m)$.

To achieve this, the Barrett reduction algorithm is employed for modular multiplication, shifting (mod S_1) and (mod S_2) into division with the Barrett parameter $R = 2^K$:

$$a * b \bmod S = a * b - S\lfloor \frac{a\lfloor \frac{Rb}{S} \rfloor}{R} \rfloor = a * b - S\lfloor \frac{a\mu}{R} \rfloor \tag{14}$$

where $\mu = \lfloor \frac{Rb}{S} \rfloor$ and the result from the Barrett algorithm is within $[0, 2S)$ not $[0, S)$. Kuang et al [24] demonstrated that by significantly increasing K beyond the bit length $L = |S|_2$ of S, or $K \geq L + 32$, the result from the Barrett algorithm (14) could be compressed within $[0, S)$.

Leveraging the Barrett algorithm (14), U_{ij} and V_{ij} are redefined as follows:

$$\begin{aligned} V_{ij} &= \beta(FQ_{ij} \bmod S_2) \bmod p = Fq'_{ij} - s_2\lfloor \frac{F\nu_{ij}}{R} \rfloor \bmod p \\ U_{ij} &= \beta(HP_{ij} \bmod S_1) \bmod p = Hp'_{ij} - s_1\lfloor \frac{H\mu_{ij}}{R} \rfloor \bmod p \end{aligned} \tag{15}$$

with randomly chosen $\beta \in \mathbf{F}_p$ and

$$\begin{aligned} & q'_{ij} = \beta Q_{ij} \bmod p, p'_{ij} = \beta P_{ij} \bmod p \\ & \nu_{ij} = \lfloor \frac{RQ_{ij}}{S_2} \rfloor, \mu_{ij} = \lfloor \frac{RP_{ij}}{S_1} \rfloor \\ & s_1 = \beta S_1 \bmod p, s_2 = \beta S_2 \bmod p \end{aligned} \tag{16}$$

Eq. (16) represents the public key for signature verification or PK_v, and the verification equation is Eq. (13) with coefficients defined in Eq. (15) but without the symmetric keys S_1 and S_2.

Both polynomials $U(x, u_1, \ldots, u_m)$ and $V(x, u_1, \ldots, u_m)$ have coefficients determined by the received signature $Sig = \{F, H\}$, significantly restricting the possibility of forging a signature.

5.4 HPPK Key Triple

Combining the key pairs for Key Encapsulation Mechanism (KEM) (SK and PK_e) and Digital Signatures (DS) (SK and PK_v) into a key triple SK, PK_e,

and PK_v enables the same SK to be utilized for both decapsulation of ciphertext and the creation of a signature for a message. This versatility opens the door to various applications, including but not limited to blockchains and Zero-Knowledge Protocols.

6 Security Brief

Kuang and Barbeau conducted an in-depth security analysis of QPP in [14], while Kuang et al. provided security analyses for HPPK KEM in [20] and HPPK DS in [24]. Here, we provide concise summaries.

6.1 QPP

QPP extends Shannon's perfect OTP, employing n-bit permutation matrices chosen from the quantum key space or permutation group, with a random classical key bit string of $n2^n$ bits. For a pad of M permutation matrices, the total classical key string is $Mn2^n$ bits long, providing entropy equivalent to $e = M\log_2(2^n!)$. This yields a best brute search complexity of $\mathcal{O}((2^n!)^M)$. The exponential complexity allows for relatively small values of n (e.g., $n = 4, 8$ bits).

Linear and differential cryptanalysis, effective against block ciphers like AES, are less potent against QPP due to its matrix representations. QPP eliminates certain arithmetic permutations present in AES, making attacks more challenging. Pre-randomizing and random dispatching further thwart potential cryptanalysis. Attackers are best served by attempting to obtain the initial shared classical random key material.

6.2 HPPK

In contrast to QPP's matrix form, HPPK employs a special arithmetic QPP, or modular multiplicative permutations over hidden rings, for symmetric encryption of plain public polynomials. While the symmetric encryption key must be pre-shared, the roundtrip public key scheme enables a self-shared symmetric key for both encryption and decryption. Attackers must obtain the symmetric encryption key or the parameters R_1, S_1, and R_2, S_2. Once acquired, attackers can compromise the entire HPPK for KEM and DS.

For HPPK KEM, the complexity of key recovery attack was estimated in [20] as $\mathcal{O}(\eta 2^L)$ with $\eta < 1$. We re-evaluate its complexity here. The total possible coprime pairs of R_1, S_1 and R_2, S_2 are:

$$\begin{aligned}\mathcal{O}\left(2\sum_{t=2^{L-1}}^{2^L}\varphi(t)\right) &\approx \mathcal{O}\left(\frac{3}{\pi^2}[2^{2L} - 2^{2(L-1)}]\right)\\ &\approx \mathcal{O}\left(\frac{9}{2\pi^2}2^{2L}\right)\end{aligned} \tag{17}$$

Equation (17) aligns with the estimate in [20]. Due to its random encapsulation, HPPK KEM holds the indistinguishable chosen plaintext attack or IND-CPA property.

In the case of HPP DS, the attacker doesn't need to find R_1 and R_2 since the signature F and H have their inverses. The attacker only needs to obtain S_1 and S_2; with intercepted true signatures, they can break the HPPK scheme. The overall complexity is thus $\mathcal{O}(2^L)$.

Table 1 illustrates all key sizes, cipher sizes, and signature sizes of HPPK KEM and DS based on their security complexities. Different configurations exist for a required security level, and the table shows typical configurations. In general, HPPK offers very compact sizes.

Table 1. This table compares public key, private key, and ciphertext sizes for HPPK KEM and DS across different security levels. Configurations are denoted as $(|p|_2, n, \lambda, m)$ with $L = |2|p|_2 + 8$ and $K = L + 32$.

	Size (Bytes)			
	PK_e/PK_v	***SK***	***Ciphertext/Signature***	***Secret/Hash***
			Security Level I	
KEM-(32,1,1,2)	108	52	224	32
KEM-(32,1,1,3)	162	52	224	32
DS-(64,1,1,1)	220	104	144	32
			Security Level III	
KEM-(48,1,1,2)	156	76	240	32
KEM-(48,1,1,3)	234	76	240	32
DS-(96,1,1,1)	300	152	208	48
			Security Level V	
KEM-(64,1,1,2)	204	100	208	32
KEM-(64,1,1,3)	306	100	208	32
DS-(128,1,1,1)	356	216	272	64

7 Conclusion

In summary, our research marks a substantial advancement in the pursuit of quantum-resistant cryptographic protocols. Through the introduction and convergence of two innovative primitives, the Quantum Permutation Pad (QPP) and the Homomorphic Polynomial Public Key (HPPK), both rooted in the robust Galois Permutation Group, we have established a cornerstone for secure communication systems in the era of quantum computing.

QPP's groundbreaking approach extends Shannon's perfect secrecy into the quantum realm, presenting a reusable and adaptable solution for symmetric key encryption. By harnessing the bijective and non-commutative properties of the Galois Permutation Group, QPP ensures quantum security in both classical and quantum-native systems.

In tandem with QPP, HPPK introduces a novel Homomorphic Polynomial Public Key designed for Key Encapsulation Mechanism (KEM) and Digital Signatures (DS). Exploiting the inherent partial homomorphic properties of modular multiplicative permutations, HPPK provides a robust symmetric encryption mechanism for asymmetric cryptography, independent of NP-hard problems. The seamless integration of KEM and DS within HPPK yields compact key sizes, cipher sizes, and signature sizes, showcasing exceptional performance across various cryptographic operations.

Our paper delves not only into the design and implementation of QPP and HPPK but also unifies these cryptographic primitives under the single umbrella of the Galois Permutation Group. This organic integration represents a significant stride in the ongoing endeavor to establish quantum-resistant cryptographic protocols. As quantum computing continues to progress, our work contributes significantly to the development of secure communication systems, addressing the vulnerabilities posed by quantum technologies.

8 Acknowledgments

The author wishes to express gratitude to Prof. Daniel Panario for insightful discussions and the invitation to submit this work.

References

1. Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, and Bas Westerbaan. SPHINCS+. Tech. rep. available at https://csrc.nist.gov/projects/post- quantum-cryptography/round-3-submissions, 2020. National Institute of Standards and Technology.
2. Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Stehlé Damien. CRYSTALS-KYBER. Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions, 2020. National Institute of Standards and Technology.
3. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU prime: Reducing attack surface at low cost. In Carlisle Adams and Jan Camenisch, editors, *Selected Areas in Cryptography – SAC 2017*, pages 235–260, Cham, 2018. Springer International Publishing.
4. et al. Carlos Aguilar Melchor. Hamming quasi-cyclic (hqc). http://pqc-hqc.org/doc/hqc-specification_2021-06-06.pdf, 2021.
5. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Ml wr-based kem. https://www.esat.kuleuven.be/cosic/pqcrypto/saber/index.html. Online; accessed 1 November 2023.
6. Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

7. Jintai Ding and Bo-Yin Yang. *Multivariate Public Key Cryptography*, pages 193–241. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
8. Ronald A Fisher and Frank Yates. *Statistical tables: For biological, agricultural and medical research*. Oliver and Boyd, 1938.
9. Hao Guo. An algebraic attack for forging signatures of mppk/ds. Cryptology ePrint Archive, Paper 2023/453, 2023.
10. R Kuang and M Perepechaenko. A novel homomorphic polynomial public key encapsulation algorithm [version 1; peer review: awaiting peer review]. *F1000Research*, 12(1347), 2023.
11. Randy Kuang. A deterministic polynomial public key algorithm over a prime galois field gf(p). In *2021 2nd Asia Conference on Computers and Communications (ACCC)*, pages 79–88, 2021.
12. Randy Kuang and Michel Barbeau. Indistinguishability and non-deterministic encryption of the quantum safe multivariate polynomial public key cryptographic system. In *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–5. IEEE, 2021.
13. Randy Kuang and Michel Barbeau. Performance analysis of the quantum safe multivariate polynomial public key algorithm. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 351–358. IEEE, 2021.
14. Randy Kuang and Michel Barbeau. Quantum permutation pad for universal quantum-safe cryptography. *Quantum Information Processing*, 21:211, 2022.
15. Randy Kuang and Nicolas Bettenburg. Shannon perfect secrecy in a discrete hilbert space. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 249–255. IEEE, 2020.
16. Randy Kuang, Dafu Lou, Alex He, and Alexandre Conlon. Quantum safe lightweight cryptography with Quantum Permutation Pad. In *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, pages 790–795. IEEE, 2021.
17. Randy Kuang, Dafu Lou, Alex He, and Alexandre Conlon. Quantum safe lightweight cryptography with Quantum Permutation Pad. *Advances in Science, Technology and Engineering Systems Journal*, 6:401–405, 2021.
18. Randy Kuang, Dafu Lou, Alex He, Chris McKenzie, and Michael Redding. Pseudo quantum random number generator with quantum permutation pad. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 359–364. IEEE, 2021.
19. Randy Kuang and Maria Perepechaenko. Quantum encryption with quantum permutation pad in ibmq systems. *EPJ Quantum Technology*, 9(26), 2022.
20. Randy Kuang and Maria Perepechaenko. Homomorphic polynomial public key encapsulation over two hidden rings for quantum-safe key encapsulation. *Quantum Information Processing*, 22:315, 2023.
21. Randy Kuang and Maria Perepechaenko. Optimization of the multivariate polynomial public key for quantum safe digital signature. *Scientific Reports*, 13:6363, 2023.
22. Randy Kuang, Maria Perepechaenko, and Michel Barbeau. A new post-quantum multivariate polynomial public key encapsulation algorithm. *Quantum Information Processing*, 21:360, 2022.
23. Randy Kuang, Maria Perepechaenko, and Michel Barbeau. A new quantum-safe multivariate polynomial public key digital signature algorithm. *Scientific Reports*, 12:13168, 2022.

24. Randy Kuang, Maria Perepechaenko, Mahmoud Sayed, and Dafu Lou. Homomorphic polynomial public key cryptography for quantum-secure digital signature, 2023.
25. Dafu Lou, Alex He, Michael Redding, Marc Geitz, Ryan Toth, Ronny Döring, Richard Carson, and Randy Kuang. Benchmark performance of digital qkd platform using quantum permutation pad. *IEEE Access*, 10:107066–107076, 2022.
26. Dafu Lou, Randy Kuang, and Alex He. Entropy transformation and expansion with Quantum Permutation Pad for 5G secure networks. In *The IEEE 21st International Conference on Communication Technology*. IEEE, 2021.
27. V Lyubashevsky, L Ducas, E Kiltz, T Lepoint, P Schwabe, G Seiler, D Stehlé, and S Bai. CRYSTALS-DILITHIUM. Tech. rep. available at https://csrc.nist.gov/projects/post- quantum-cryptography/round-3-submissions, 2020. National Institute of Standards and Technology.
28. R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, January 1978.
29. Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *iEEE Transactions on information Theory*, 39(5):1639–1646, 1993.
30. et al. Nicolas Aragon. Bit flipping key encapsulation. https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.04.1.pdf, 2022.
31. NIST. Status report on the third round of the nist post-quantum cryptography standardization process. https://csrc.nist.gov/publications/detail/nistir/8413/final, July 2022.
32. Maria Perepechaenko and Randy Kuang. Quantum encrypted communication between two ibmq systems using quantum permutation pad. In *2022 11th International Conference on Communications, Circuits and Systems (ICCCAS)*, pages 146–152, 2022.
33. Maria Perepechaenko and Randy Kuang. Quantum encryption of superposition states with quantum permutation pad in ibm quantum computers. *EPJ Quantum Technology*, 10(7), 2023.
34. Maria Perepechaenko and Randy Kuang. Quantum encryption of superposition states with quantum permutation pad in IBM quantum computers. *EPJ Quant. Technol.*, 10(1):7, 2023.
35. T Prest, P-A Fouque, J Hoffstein, P Kirchner, V. Lyubashevsky, T Pornin, T Ricosset, G Seiler, W Whyte, and Z Zhang. FALCON. Tech. rep. available at https://csrc.nist.gov/projects/post- quantum-cryptography/round-3-submissions, 2020. National Institute of Standards and Technology.
36. Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
37. Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949.
38. Tristan Sharp, Rishabh Khare, Erick Pederson, and Fabio Lorenzo Traversa. Scaling up prime factorization with self-organizing gates: A memcomputing approach, 2023.
39. Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.
40. Peter W Shor and John Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical review letters*, 85(2):441, 2000.