Quantum Information Processing (2022) 21:211
https://doi.org/10.1007/s11128-022-03557-y

Quantum permutation pad for universal quantum-safe cryptography

Randy Kuang[1] · **Michel Barbeau**[2]

Received: 23 March 2021 / Accepted: 16 May 2022 / Published online: 14 June 2022

Abstract

Classical cryptographic techniques are currently under the growing quantum computing threat. New techniques that quantum computing algorithms cannot break are urgently needed. We present such an encryption method. It builds upon quantum permutation logic gates or quantum permutation pads. It is universal in that it can be equally employed on classical computers, today's Internet, and the upcoming quantum Internet. While the cryptographic technique is formulated in a quantum computing framework, it does not rely on physical properties uniquely present at the quantum level, such as no-cloning or entanglement of data. It achieves with today's technology a level of security comparable to what will be possible to attain with tomorrow's quantum technology. The mathematics behind the cryptographic technique, quantum representations of a symmetric group over a computational basis, is surprisingly simple. However, the challenge faced by an adversary wishing to break the code is intractable and uninterpretable, a property of Shannon's perfect secrecy. We believe that the cryptographic technique presented in this article can be used in several different ways and modes. It can be integrated into numerous current Internet protocols, or the Internet of Things, making them quantum safe. In addition, it can be used to transition to the upcoming Internet quantum technology smoothly.

Keywords Cryptography · Post-quantum cryptography · Quantum computing · Quantum permutation pad

✉ Michel Barbeau
MichelBarbeau@cunet.carleton.ca; barbeau@scs.carleton.ca

Randy Kuang
randy.kuang@quantropi.com

[1] Quantropi Inc, Ottawa, ON K1Z 8P9, Canada

[2] School of Computer Science, Carleton University, Ottawa K1S 5B6, Canada

1 Introduction

In this article, the qualifier universal generically refers to the classical data and quantum data worlds. Building upon an encryption technique based on quantum algorithms of permutation logic gates called quantum permutation pad (QPP), we define a universal confidential communication framework. QPP is defined by an abstract model using the quantum mechanics formalism [36]. This model is represented by the box Abstract QPP in Fig. 1. It uses Hilbert vector spaces. A data item can be interpreted as a column vector or a qubit register in this framework. Figure 1 illustrates two representative actualizations of the abstract model. A data item is interpreted as a column vector in the upper part. A classical sender plaintext, in the column vector format, is encrypted by a classical data actualization of QPP. The encrypted classical data are transported to a receiver over a classical network, such as the current Internet. The receiver decrypts the classical data using the classical data actualization of QPP and restores the classical plaintext. The column vector representation is general enough that any classical data can be represented in that format. In the lower part of Fig. 1, a sender quantum plaintext, in the qubit register format, is encrypted by a quantum data actualization of QPP. The encrypted quantum data are transported to a receiver over a quantum network. The receiver decrypts the quantum data using the quantum data actualization of QPP and restores the quantum plaintext. The genericity of the encryption technique is such that its logic can be mapped to both the classical and quantum worlds. It does not rely on the no-cloning theorem of quantum data nor the ability of the sender and receiver to detect the presence of an interceptor, such as in quantum key distribution (QKD) [8]. It is independent of the implementation technology.

Contribution The article presents a new symmetric cryptosystem called QPP. First, it is important to highlight the relevance of symmetric cryptosystems. They work hand in hand with asymmetric cryptosystems. Asymmetric cryptography is used to establish and encrypt the random key that is significantly smaller than the messages.

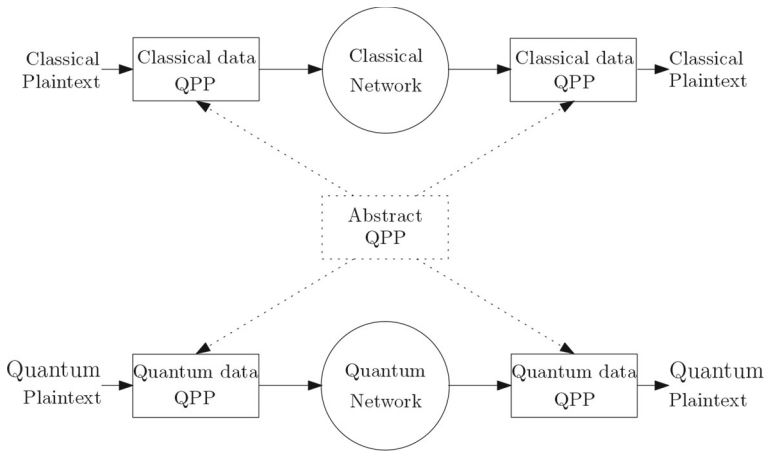

Fig. 1 The universal cryptography concept

Together with the short randomly generated key, messages are efficiently encrypted using symmetric cryptography. This hybrid approach to secure communications has been widely adopted in modern cybersecurity. QPP is one of a kind. At its core are unitary permutation matrices. It is a mathematical concept compatible with both classical and quantum computing. Hence, QPP can run on classical computing platforms, on quantum computing platforms, or a mix of both. In contrast to a quantum cryptosystem [6], the scope of QPP is not limited to quantum platforms and can run on classical platforms. In contrast to a classical cryptosystem [34], QPP can run on a quantum computing platform [40]. There are no classical symmetric encryption algorithms that are easy as QPP to implement on a quantum computing system.

QPP's security relies solely on the uninterpretable security following the use of bijective transformations. On the one hand, an adversary is faced with testing $2^n!$ equally likely bijection maps to crack the code, where n, a positive integer, is a security parameter. Conducting such an exploration requires an exponential quantity of resources. On the other hand, brute force tests performed on a given ciphertext yield the entire n-bit word plaintexts equally likely.

In Sect. 2, we review the state of the art in classical and quantum cryptography. In Sect. 3, we bridge classical and quantum cryptography. In Sect. 4, we present QPP. In Sect. 5, we describe our universal cryptography framework. Its security is analyzed in Sect. 6. Implementation aspects are reviewed in Sect. 7. We conclude with Sect. 8.

2 Related work

2.1 Classical cryptography

Today's information security builds upon asymmetric and symmetric cryptographic techniques. On the one hand, asymmetric cryptography refers to algorithms that establish a shared key between a pair of communication peers over an unsecure public channel. They include Rivest, Shamir and Adleman (RSA) [45], Diffie–Hellman [16] and Elliptic Curve Cryptography (ECC)) [31]. Public-key cryptographic techniques are based on specific computational difficulties, that is, RSA on the prime factorization problem, Diffie–Hellman and ECC on the discrete logarithm problem. In the average case, those problems are intractable and non-deterministic polynomial time (NP-hard). Their hardness is exponentially increasing with the key bit length. On the other hand, symmetric cryptographic techniques, such as data encryption standard (DES), triple DES [7] and advanced encryption standard (AES)) [34], achieve security based on a shared secret key. The key length is 64 bits in DES, 192 bits in triple DES and 128 or 256 bits in AES.

In 1994, Shor proposed a new algorithm to factorize large numbers leveraging the power of quantum bits (qubits) superposition [48]. Shor's algorithm reveals an exponentially increased capability of solving cryptographic NP-hard problems in polynomial time. However, the risk that such a discovery poses on public-key cryptography has been insignificant until a recent quantum computing breakthrough disclosed by Google [5, 56] . In 2015, Mosca [33] stated that "at present,... I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031."

In 1996, Grover invented a new quantum search algorithm [19] that solves the unstructured search problem of size n in $\mathcal{O}\left(\sqrt{n}\right)$ queries, while any classical algorithm needs $\mathcal{O}(n)$ queries. This speedup requires the key length of standard AES to be augmented from 128 bits to 256 bits with true randomness. In its report on post-quantum cryptography (PQC) [11], National Institute of Standards and Technology (NIST) clearly indicates that classical public-key algorithms are no longer secure and AES keys need to be true random and doubled in length.

NIST is currently in a review process of PQC techniques for the purposes of standardization. There were over two dozen techniques entered in the 2017 first run. The techniques can be classified into two categories: digital signature and key encapsulation mechanism (KEM). Within KEM, the techniques are lattice-based such as Nth degree truncated polynomial ring units (NTRU) [20] and ring-learning with error (R-LWE) [4], and code-based such as McEliece encryption system [13, 32] and random linear code encryption (RLCE) [53], multivariate [42] and super singular isogeny Diffie–Hellman (SIDH) [12]. The security of the different PQC techniques relies on computational difficulties, especially the NP-hard problem such as the shortest vector problem (SVP), in lattice-based techniques, and the error-correcting problem, in code-based techniques. In July 2020, NIST just entered its third round of candidate reviews.

2.2 Quantum cryptography

Leveraging the laws of quantum physics, Bennett and Brassard proposed the QKD protocol [8]. Shor and Preskill published a proof that QKD achieves unconditional secure key distribution [49]. QKD has been widely explored resulting in a variety of implementations and improvements (see a review article ref. [18]). In its initial design, QKD uses single photons as information carriers. In single photon QKD, extremely weak coherent photon pulses act as qubits, with an average photon number per laser pulse in the order of 0.2 or less. This constraint greatly impacts QKD's key rate and working distance. To overcome that, continuous-variable QKD (CV-QKD) proposed the use of weak coherent states as information carriers [28], with an average photon number per laser pulse in the order of hundreds to thousands. Diamanti, Lo, Qi and Yuan reviewed QKD challenges for a variety of practical implementations [15]. Recently, Xu et al. published a complete review of QKD security analysis addressing protocol, implementation, signal source and detection aspects [55].

Quantum cryptography generally refers to QKD. QKD and all its derivatives have shown that quantum mechanics can be used to secure classical data. Besides, quantum mechanics can be used to secure quantum data as well. Using Clifford groups, a cryptographic technique for quantum message authentication has been originally be introduced by Aharonov at al. [2], with follow-up work by Broadbent and Wainewright [10, 52]. Alagic et al. [3] and St-Jules [50] proposed asymmetric- and symmetric-key encryption techniques for quantum data.

A post of National Security Agency (NSA) explains their decision to discourage the use of QKD [35]. In their opinion, the added value of QKD does not compensate for the risks and limitations specific to the method. NSA favors PQC techniques.

On the other hand, three of the four PQC finalist algorithms are based on SVP, which has been recently reported solvable leveraging adiabatic quantum computation [23]. Are we in a kind of dead end?

What we propose in this article is a novel paradigm, i.e., a cryptographic technique that works for classical data and quantum data. The exact way it is being used is just a matter of interpretation. It relies solely on the uninterpretable security of the Shannon perfect secrecy extended over a quantum computational basis.

3 Bridging OTP and QKD

There are several different physical implementations of QKD. They mainly rely on polarization or phase encoding, with two non-orthogonal bases. In polarization encoding, the two bases are the rectilinear basis, where photon polarization is either vertical or horizontal, and the diagonal basis, where photon polarization is either diagonal or antidiagonal. In phase shift encoding, the first basis consists of phase shifts zero or π radians, while the second basis comprises the phase shifts $\pi/2$ or $3\pi/2$ radians.

Let us briefly review QKD using the Dirac notation, but without going to the details of physical encoding such as polarization or phase encoding. It uses two non-orthogonal bases. The first is the single-qubit computational basis $B_1 = \{|0\rangle, |1\rangle\}$, where $|0\rangle$ and $|1\rangle$ are, respectively, equal to the column vectors $[1, 0]^T$ and $[0, 1]^T$. It is a two-dimensional Hilbert space. The second is the Hadamard basis $B_2 = \{|-\rangle, |+\rangle\}$, where $|-\rangle$ and $|+\rangle$ are, respectively, equal to the column vectors $\left[1/\sqrt{2}, -1/\sqrt{2}\right]^T$ and $\left[1/\sqrt{2}, 1/\sqrt{2}\right]^T$. Note that the Hadamard basis vectors (B_2) are superpositions of the computational basis vectors (B_1). It can be easily verified that B_1 and B_2 are orthonormal bases, but mutually not orthogonal.

The QKD protocol proceeds as follows. There is a sender and a receiver. The sender randomly generates two equal-length strings σ_1 and σ_2 of classical bits. Every bit in string σ_1 controls the selection of a basis for encoding a key bit, value zero selects the basis B_1. Value one selects the basis B_2. String σ_2 is a key bit sequence. Value zero is encoded as the vector $|0\rangle$ in the basis B_1 or as the vector $|+\rangle$ in the basis B_2. Bit one is encoded as the vector $|1\rangle$ in the basis B_1 or as the vector $|-\rangle$ in the basis B_2.

The receiver randomly generates a bit string σ_3. The length of σ_3 is the same as the length of the strings σ_1 and σ_2. The string σ_3 controls the selection of the measurement bases for incoming vectors. Bit zero selects the basis B_1, while bit one selects the basis B_2. Measurement results are recorded in a fourth bit string σ_4.

The protocol concludes with the announcement of the selected measurement bases. This phase requires that the sender and receiver share a secret authentication key, aiming to mitigate Person-In-The-Middle (PITM) attacks. The receiver publicly announces the measurement basis selection string σ_3. The sender compares σ_3 with σ_1. The sender indicates to the receiver the positions where the sender randomly generated bases match the receiver randomly selected bases. The sender and receiver delete in σ_2 and σ_4, respectively, the bits where there are mismatch bases. The result is a common sequence of bits called the sifted key.

QKD encoding can be defined as well with quantum gates (see Appendix A). In this interpretation, the identity gate I and Hadamard gate H are used to secure quantum key distribution. However, there exists a relationship between those two gates (I and H) and permutation matrices in the symmetric group S_2 [54]. Furthermore, a slight variation of QKD can be defined with a pre-shared secret for both authentication and basis selection purposes. The public announcement of bases can be omitted which in return enhances its security with this trusted mode. This variation clearly reflects a quantum implementation of the classical one-time pad.

Table 1 connects QKD, a quantum encryption technique, with OTP, a classical encryption technique using the Exclusive OR (XOR) Boolean operation. On the left side, we have the QKD encoding. The basis bit (σ_1) and key bit (σ_2) together determine an encoding. On the right side, we have a quantum interpretation of OTP. Data bits 0 and 1 are mapped to computational basis members $|0\rangle$ and $|1\rangle$. Table 1 shows that bit a in the XOR operation ($\oplus$) can be interpreted as a control bit that selects a permutation matrix for encoding a data bit b, in a way similar to what the basis bit does in QKD (see Appendix A). The rightmost column shows the encoding of the computational basis members using permutation matrices. With this way of seeing things, QKD encoding can be interpreted as a quantum implementation of OTP, using the gates I and H. The public announcement of bases in QKD determines the shared bases for both encoding and measurement. The sifted key establishes a shared classical keypad used for the synchronization of encoding and measuring bases.

Table 2 shows sample choices of encoding gates by QKD and quantum OTP, when both the sender and receiver agree on the basis selection. This interpretation of the shifted key and OTP is applicable to both quantum and classical data (no need of the superposition concept). When the implementation meets the needed requirements for truly random key generation, this interpretation of OTP achieves Shannon perfect

Table 1 A parallel between QKD and OTP encodings

Encryption technique					
QKD			Quantum OTP		
Basis bit (σ_1)	Key bit (σ_2)	Encoding	Control bit a	Data bit b	$a \oplus b$
0	$\|0\rangle$	$I\|0\rangle = \|0\rangle$	0	0	$I\|0\rangle = \|0\rangle$
0	$\|1\rangle$	$I\|1\rangle = \|1\rangle$	0	1	$I\|1\rangle = \|1\rangle$
1	$\|0\rangle$	$H\|0\rangle = \|+\rangle$	1	0	$P_2\|0\rangle = \|1\rangle$
1	$\|1\rangle$	$H\|1\rangle = \|-\rangle$	1	1	$P_2\|1\rangle = \|0\rangle$

Table 2 Sample QKD and OTP(XOR) encoding

Sender basis bit (σ_1)	1	0	1	1	0	0	1	0	1	1	0
Receiver basis bit (σ_3)	0	0	1	0	0	1	1	0	1	1	1
Sifted key		0	1		0		1	0	1	1	
QKD encoding		I	H		I		H	I	H	H	
Quantum OTP encoding		I	P_2		I		P_2	I	P_2	P_2	

secrecy (ref. [46]). On the other hand, this interpretation of OTP needs trusted peers to mitigate PITM attacks.

It is very clear that the quantum interpretation of OTP is based on a single-qubit computational basis. Within this basis, the symmetric group is S_2 with only the two permutation matrices I and P_2. OTP can be considered as a pad consisting of permutation matrices selected from a single-qubit permutation group, with bit 0 as gate I and bit 1 as gate P_2. OTP is a single-qubit quantum permutation pad, implemented mathematically, while QKD is a single-qubit quantum permutation (corresponding to I or H) pad, implemented physically with photons.

Here is an interesting question. The computational basis B_1 is single qubit. Can the quantum interpretation of OTP be extended to a n-qubit computational basis ($n > 1$), using permutation matrices in the symmetric group S_{2^n}? The answer turns out to be yes!

4 Quantum permutation pad

In this section, we describe QPP. Let us consider the n-qubit computational basis $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$. Over this computational basis, there is a symmetric group S_{2^n} with $2^n!$ elements P_i, with $i = 1, 2, \ldots, 2^n!$ representing $2^n!$ permutation operators (ref. [54]). They are 2^n by 2^n unitary reversible matrices, also called permutation gates.

There is a message sender and a receiver. Confidential communication is required. A uniformly selected random permutation gate P in S_{2^n} is used by the sender, while its transpose permutation gate P^T is employed by the receiver. The pair of P and P^T is a sender–receiver shared secret key. The sender uses the permutation gate P to transform a plaintext state $|m\rangle$ into a ciphertext state $P|m\rangle$, denoted as $|c\rangle$. The ciphertext $|c\rangle$ is transmitted over a network to the receiver. The receiver applies the transposed permutation matrix P^T and restores back the plaintext state $P^T|c\rangle = P^T P|m\rangle = |m\rangle$.

Lemma 1 (perfect secrecy) *For n greater than one and uniform random key distribution, QPP achieves Shannon perfect secrecy (ref.* [46]*).*

Proof It follows from the fact that the key domain size ($2^n!$) is greater than the ciphertext domain size (2^n), which is greater than or equal to the message domain size (2^n). Keys are used with uniform probability $1/(2^n!)$. A permutation map is also a bijection over Galois fields $GF(2^n)$. For every message-cipher pair, there exist $(2^n - 1)!$ permutation matrices or keys. The overall probability for a message-cipher pair is still 2^{-n}, or Shannon's equally likely for perfect secrecy. Note that the constraint n greater than one is required because S_2 is a solvable group, leading to the Shannon perfect secrecy of OTP. □

Lemma 2 (non-commutability) *For a pair of permutations P_i and P_j in S_{2^n}, n greater than two, P_i generally does not commute with P_j.*

Proof It follows from the fact that permutations are generally not commutative. □

Corollary 1 *Given ciphertext $|c\rangle$ encrypted with permutation gate P, the application of any permutation gate Q in S_{2^n} is not equal to P^T, i.e., $Q|c\rangle$ equally likely yields any ciphertext $|c'\rangle$ with c' within a Galois field $GF(2^n)$.*

QPP was originally proposed by Kuang and Bettenburg [24]. Shannon perfect secrecy was extended over quantum Hilbert spaces. In Boolean algebra, OTP can be secure only for a single time use. Thanks to the non-commutability of permutation gates, QPP can be reused without compromising security. In fact, QPP can still be called OTP but with one time provision, forever use. QPP has been applied in a few use cases [25–27]. In this paper, a new universal quantum-safe cryptographic method is established leveraging QPP through the mathematical representation of quantum permutation gates over Hilbert spaces. For the first time, it puts forward the idea that we can build a quantum-safe Internet over a hybrid Internet infrastructure. QPP works in a quantum computing system with physical permutation gates between quantum computers. QPP also works in a classical computing system with the mathematical representation of quantum permutation gates. QPP works as well in hybrid, quantum and classical networks with pretty much today's Internet infrastructure consisting of copper, fiber, wireless, laser, etc. QPP offers a way to reuse today's existing trillion dollars Internet infrastructure for quantum-safe communications. We know that we have an urgent quantum threat problem, dubbed Years to Quantum (Y2Q). QPP is a good and economic candidate for a vaccine for Y2Q.

5 Universal encryption technique

QPP is a cryptography technique originally introduced by Kuang and Bettenburg (see Sect. 4). It can be interpreted in a classical data way, where the members of the computational basis $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ correspond to the 2^n element column vectors $[1, 0, \ldots, 0]^T, [0, 1, \ldots, 0]^T, \ldots, [0, 0, \ldots, 1]^T$ (ref. [36]). It may also be interpreted in a quantum data way, where the members of the computational basis $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ correspond to n-quit quantum states.

QPP is information-theoretically secure, as the QPP or Vernam Cipher (see ref. [30] Definition 1.39). At first glance, the conditions required to achieve this property reduce practicality of the quantum technique. The key size of QPP is in $\mathcal{O}(2^n!)$, i.e., space complexity is exponential. However, in contrast to OTP, a consequence of Corollary 1 is that key reuse in QPP does not invalidate Shannon perfect secrecy. Corollary 1 means that QPP is not vulnerable to attacks exploiting repeated use of the same key, for which OTP is vulnerable. A second application of permutation gate P to ciphertext $|c\rangle$, i.e., $P|c\rangle$, yields a new ciphertext $|c'\rangle$. By analogy with OTP, double encryption with the same keystream cancels keys and yields the plaintext message XOR with plaintext message. In OTP, key reuse enables cryptanalysis. Differential cryptanalysis is not enabled in QPP because plaintext differences are not available to the adversaries, such as in XOR-based cryptographic techniques [39]. Key reusability in QPP makes it practical.

QPP can be used in a block mode for M words of size n bits each. M uniformly selected random permutation gates $P_1, \ldots, P_M$ in S_{2^n} are used to encrypt a block of M words $m_1, \ldots, m_M$ of size n bits each into ciphertexts $P_1|m_1\rangle = |c_1\rangle, \ldots, P_M|m_M\rangle = |c_M\rangle$.

Table 3 shows that even for relatively small values of n, the corresponding number of permutation gates and Shannon entropy, a measure of uncertainty, are considerable.

Table 3 Number of permutation gates as a function of n

Message size (n)	Number of permutation gates ($2^n!$)	Shannon entropy (bits)
4	2.09×10^{13}	44.25
6	1.28×10^{89}	295
8	10^{507}	1684

Brute force attacks are unpractical. The random variable is the permutation gate with $2^n!$ possible outcomes. For a permutation group, the Shannon entropy is equal to $\log_2(2^n!)$ bits. For large values of n, it is approximately $2^n(n - 1.42)$ bits.

Referring to Fig. 3, with the classical data actualization of QPP, the ciphertext, transmitted over a classical network, is a column vector or some compact representation of it such as an index. With the quantum data actualization of QPP, the cipher text is a quantum register state that can be transported from a sender to a receiver using entanglement swapping (ref [57]) and teleportation (ref. [9]).

A pre-shared key can be reused without compromising the security of QPP. QPP holds Shannon's perfect secrecy property, as OTP. Hence, QPP could be considered as an OTP extension for quantum communications. The reason why an OTP pre-shared key can only be used one time is due to the encryption operator being bitwise XOR. Let k, m_1, m_2, c_1 and c_2 be a pre-shared key, two messages and their corresponding ciphertexts. Given ciphertexts c_1 and c_2, the encryption key could be eliminated by performing the bitwise XOR calculation $c_1 \oplus c_2$ is equal to $(m_1 \oplus k) \oplus (m_2 \oplus k)$ is equal to $m_1 \oplus m_2$. When m_1 and m_2 are alphabet plaintexts, cracking m_1 and m_2 becomes possible. In the quantum world, the OTP encryption scheme can be represented by the two-qubit quantum CNOT gate. Conversely, a one-bit QPP is equivalent to OTP encryption. However, there is no classical counterpart of n-(qu)bit QPP. Permutation gates hold the generalized uncertainty principle: $[P_i, P_j]$ is not equal to zero. There is also no corresponding control-permutation gate, like the CNOT. There are many permutation gates: $2^n!$. They constitute a permutation space, which becomes our key space, with huge entropy: $log_2 2^n!$. That is the major difference between classical Boolean algebra and quantum linear algebra. A permutation is a typical bijective transformation that is Shannon perfect! The QPP's bijective property may be exploited by a statistical analysis attack. To address that eventuality, QPP encryption consists of multiple permutation gates, amplifying diffusion. Also, QPP does need to be equiped with pre-randomizing and dispatching to further augment diffusion.

6 Security analysis

6.1 Unsolvable permutation groups

Although QPP would work nicely with n equal two, n needs to be larger than two to avoid the solvable permutation group, as demonstrated by Galois. Indeed for n greater than two, S_{2^n} are unsolvable Galois groups. In general, two permutation operators P

and P' are non-commutative, that is, $[P, P']$ is not equal to zero. P and P' do not share the same eigenbasis. That means, a cipher text $|c\rangle$, obtained from a plaintext $|m\rangle$ transformed by a selected permutation gate P, does not reveal any information neither about both permutation gate P and P' nor about the plaintext state $|m\rangle$.

6.2 Pre-processing

In secure and trusted environments, QPP has the property of Shannon perfect secrecy. However, in a practical setting, QPP requires plaintext randomization and dispatching to a permutation matrix in QPP if the plaintext is not truly random, but statistically biased. This type of pre-processing strategy is present in numerous symmetric cryptographic algorithms, such as ShiftRows and MixColumns in AES. The preprocessing can be seeded with the pre-shared secret. The corresponding postprocessing is applied at the receiver side after decryption by QPP to restore the original plaintext.

6.3 Ciphertext attack

Given a ciphertext state $|c\rangle$, how challenging is for an adversary to find a transposed permutation matrix P^T that reveals the corresponding plaintext state $|m\rangle$? There are $2^n!$ candidates. The ciphertext state $|c\rangle$ does not provide any clue about what P^T can be. The amount of resources required for a brute force search is super exponential to n, together with un-interpretability. QPP can be considered safe with respect to brute force search attacks.

6.4 Plaintext attack

A known plaintext attack is an attack model for cryptanalysis used to reveal secret key information. Let us assume that an instance of QPP is selected with M permutation matrices, based on shared secret key material. A dispatcher driven by the shared key material dispatches a plaintext message to a permutation matrix within the QPP. Given a plaintext state $|m\rangle$ and a ciphertext state $|c\rangle$, with $P|m\rangle = |c\rangle$, what can be inferred about P? First, obviously we can infer that P maps $|m\rangle$ to $|c\rangle$, for a guessed P within a space of $2^n!$ permutation matrices. Secondly, because P is a bijection, if $P|m'\rangle$ is also equal to $|c\rangle$, then m and m' must identical if the dispatcher dispatches $|m'\rangle$ to the same permutation matrix P. Other than that, to determine what P does to the members of the computational basis different from $|m\rangle$, the adversary is left with exploring $(2^n - 1)!$ choices of permutation matrices for a given pair $|m\rangle$ and $|c\rangle$. Furthermore, the dispatcher dispatches $|m\rangle$ to any permutation gate within QPP, so the same $|m\rangle$ may be mapped to different $|c\rangle$'s. For a known plaintext–ciphertext pair for a QPP with M permutation gates, the uncertainty of the pad is super exponential in $\mathcal{O}\left((2^n - 1)!\right)^M\big)$. It means that QPP can be considered safe against plaintext attacks.

6.5 Indistinguishability under adaptive chosen ciphertext attack

Ciphertext indistinguishability is a great semantic security property that can be possessed by a cryptosystem. There are three types of indistinguishability: INDistinguishability under Chosen-Plaintext Attack (IND-CPA), INDistinguishability under (non-adaptive) Chosen-Ciphertext Attack (IND-CCA1) and INDistinguishability under adaptive Chosen-Ciphertext Attack (IND-CCA2). IND-CCA2 is the highest indistinguishability level. When a cryptosystem has IND-CCA2, it has IND-CPA and IND-CCA1 automatically.

Let us consider a QPP implementation with a simple preprocessing dispatcher function driven by the shared key material. On the sender side, the challenger maps the shared key material into M permutation gates for encryption. On the receiver side, the dispatcher maps shared key material to their transposed M permutation matrices for decryption. The adversary can obtain any ciphertext corresponding to a plaintext. They supply chosen plaintexts to form k plaintext–ciphertext pairs $(m_1, c_1), (m_2, c_2), \ldots, (m_k, c_k)$. The IND-CCA2 game proceeds as follows:

1. The adversary chooses two messages μ_0 and μ_1, not in $m_1, m_2, \ldots, m_k$. It requests encryption of μ_0 and μ_1 to the challenger.
2. The challenger randomly generates a bit b, encrypts the message μ_b and returns the corresponding ciphertext c.
3. The opponent must guess which message the received ciphertext c is encrypted from μ_0 or μ_1. The adversary can inquire a decryption oracle with chosen ciphertexts as much as they want, except for the ciphertext c.

For a bijective n-bit permutation map, O'Conner proved that the probability $p(\Delta m, \Delta c)$ of a differential characteristic is $2n/2^n$ for a single permutation mapping, where Δm is the XOR of two messages and Δc is the corresponding ciphertext XOR [39]. For a QPP implementation with M permutation matrices, this probability is equal to $(2n/2^n)^M$. For a typical case with n equal to eight and M equal to 16 [25, 26], the probability of a differential characteristic is equal to 2^{-128}, that is, extremely small. The opponent gets no significant benefit from making a biased decision given ciphertext c, whether it comes from μ_0 or μ_1, even though it has access to a decryption oracle. It must make a random guess among μ_0 and μ_1, that is, with uniform probability is $1/2$.

7 Implementation

QPP can be used as a key distribution or a plaintext encryption protocol. For key distribution purposes (see [29]), QPP can play the role of key expansion protocol, as QKD does. QPP can play that role for another encryption technique such as AES [25]. The selection of permutation gates must be truly random, with no statistical bias. A permutation gate is a bijection, with domain and codomain a computational basis. Any bias or statistical pattern in the keying material is reflected in the ciphered material. As a plaintext encryption protocol, confusion and diffusion (see ref. [30] Remark 1.36) of plaintext are required before encryption to remove any statistical pattern in it, which

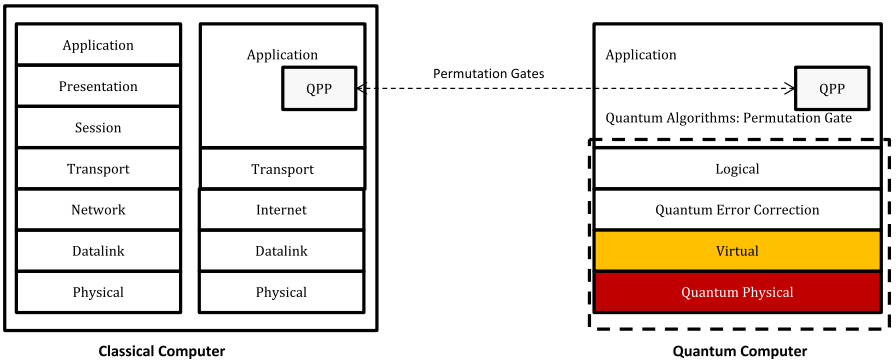

Fig. 2 Placement of QPP in classical and quantum architectures

is commonly naturally present. Randomization techniques, such as ShiftRows and MixColumns of AES, can be applied. This requirement is not unique to QPP. Indeed, most of the data encryption techniques make use of plaintext confusion and diffusion techniques. Figure 2 pictures, side by side, the placement of QPP in a classical network architecture, running on a classical computer, and a quantum computer architecture. The left side represents a five-layer classical protocol stack [41]. From bottom to top, the physical layer takes care of classical bit transmission, the datalink layer handles framing, the Internet layer does routing, the transport layer streams data process to process, while the application layer contains protocols specific to applications. QPP is placed in the application layer. It provides key distribution and encryption services to classical applications such as file transfer and email. The right side of Fig. 2 shows a quantum computer architecture running quantum algorithms [22]. From bottom to top, the quantum physical layer implements qubits and related concepts such entanglement, the virtual layer provides error cancellation, the quantum error correction layer implements logical qubits, the logical layer achieves quantum computing gates. Quantum algorithms are implemented in the application layer, together with QPP that provides key distribution and encryption services to applications. Both computer architectures share the same concept, i.e., permutation gates, but implemented in different ways. In the sequel, we discuss further QPP classical data implementation, quantum data implementation and secret key sharing.

7.1 Classical data implementation

The great advantage of QPP is the possibility to attain, with the current classical computing and Internet technology, a degree of confidentiality theoretically achievable with tomorrow's quantum technology. QPP builds upon quantum mechanics theory, while not requiring quantum-level physical properties such as entanglement and no cloning of data. QPP can run above classical computer memory and classical communication channels. A computational basis of dimension n qubits provides an alphabet of 2^n plaintext symbols for the representation of classical data.

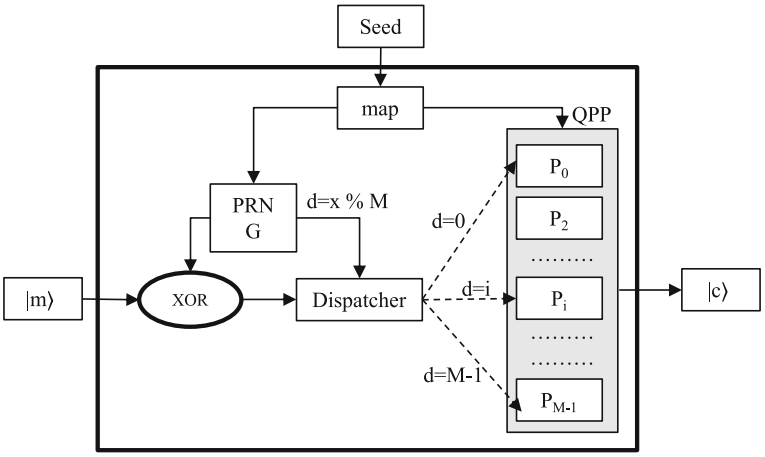

Fig. 3 Architecture of a QPP implementation

Figure 3 pictures the architecture of a stream cipher implementation of QPP, building upon classical technology. A pre-shared secret *Seed* is supplied to the module *map* that creates M permutation gates $P_0, \ldots, P_{M-1}$ and seeds a pseudo-random number generator PRNG. The latter is used to scramble the input plaintext m. A random binary sequence x is XORed with the plaintext message $|m\rangle$, yielding $|m'\rangle$. Seeded by x, and consequently the shared secret, a dispatcher D determines a randomly chosen permutation gate P_d, d in $0, \ldots, M-1$, used to encrypt the scrambled plaintext $|m'\rangle$. This permutation gate selection step smooths out statistical bias in the input. The output of the stream cipher is the column vector $P_d|m'\rangle = |c\rangle$. On the receiver side, the cipher text $|c\rangle$ is decoded as $P_d^T|c\rangle = P_d^T P_d|m'\rangle = |m'\rangle$. A postprocessing step unscrambles $|m'\rangle$ with an XOR operation and x to recover the original plaintext, i.e., $|m\rangle = |x\ XOR\ m'\rangle$. A detailed encryption and decryption example is provided in Appendix C.

In the implementation design of Fig. 3, parameter *Seed* is the shared secret. The word size n and number of permutation gates M are public security parameters. The size of *Seed* is determined by n, discussed in the sequel. Assuming truly uniform random input, because a permutation is a bijection every ciphertext word value of size n may occur with probability 2^{-n}. With a block of M words of size n bits each, every ciphertext block value may occur with probability 2^{-nM}.

Figure 4 plots the Shannon entropy as a function of the word size (n) and number of permutation gates (M) used in the design of Fig. 3. The random variable in the design of Fig. 3 is the arrangement of M permutation gates in S_{2^n}. As a function of n and M, the y-axis value is equal to M times $2^n(n-1.42)$ bits. Workable implementations may use an eight-bit word size, i.e., n, and M equal to 64. Each of them is 256-by-256 matrix. Shannon entropy is well above 10^5 bits. It means that the cipher uses 64 permutation gates. When this encryption method is used, we pre-randomized input, such as in a key distribution scenario, the scrambling–descrambling pre- and postprocessing can be omitted. A demonstration open-source software implementation together with a

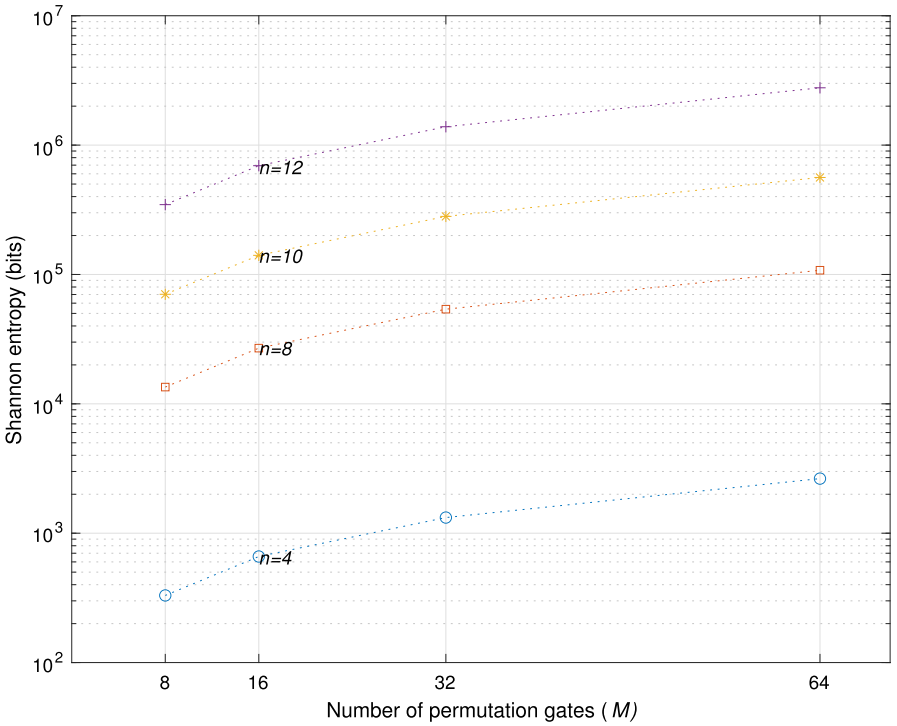

Fig. 4 Shannon entropy versus word size and number of permutation gates

performance analysis is available on a companion web page [44]. Test results using the NIST randomness test tools and Dieharder are also provided. Speed performance is equivalent to the single round of an AES-256 software implementation. It is twice the speed of AES-NI with hardware acceleration. With respect to energy consumption, QPP consumes up to 10% of the AES-256 consumption. Other implementation designs leveraging QPP can be envisioned, as a block cipher architecture.

An important issue is mapping a classical key S to permutation gates. QPP needs uniform random generation of permutation gates. This can be done using the Fisher and Yates algorithm [17]. An alternative is the subgroup algorithm [14]. With either of these algorithms, a single permutation matrix is random in S_{2^n} selected with an input random key of size $n2^n$ bits. For the design of Fig. 3, $M \cdot n2^n$ uniformly generated random bits would be needed to produce the M permutation gates. With n equal to eight and M equal to 64, it means key size of 131, 072 bits, or 16K bytes. For the Internet of Things, choices such as n equal to eight and M equal to 16 can envisioned. In that case, the key size is 32, 768 bits, or 4K bytes. We can also choose $n = 4$ with $M = 16$ to reduce RAM space to 128 bytes for QPP and achieve the total entropy of 707 bits for quantum-safe IoT communications.

QPP can be seen as a logical quantum cryptographic technique over a n-bit computational basis. It is an alternative to QKD on non-photonic or digital QKD to establish a logical quantum Internet using the existing infrastructure. It can distribute keys

between endpoints that use standard cryptographic techniques such as AES or OTP. In contrast to OTP, where the pre-shared true random secret can be only used for one time, a QPP pre-shared secret distributed one time and can be reused multiple times. Of course, it can be updated automatically without invalidating the property of perfect secrecy, thanks to the uncertainty resulting from the use of permutation groups.

An interesting question is why not using any other classical symmetric cryptosystem, instead of QPP? Let us consider a classical symmetric cryptosystem such as AES. AES supports the key sizes 128, 192 and 256 bits. Usage of an algorithm with a maximum security of 256 bits of entropy implies that the security of key exchange is no more than 256 bits of entropy. A session key with less than 256 bits of entropy is not quantum safe for data encryption, according to the NIST recommendation [37]. A physical quantum key distribution system has theoretically infinite entropy. Therefore, it can be considered semantically secure for session keys, i.e., 256 bits of entropy from a 256-bit-long distributed key. Then, data encryption with 256-AES would be considered quantum safe. QPP with 64 8-bit permutation matrices holds 64 times 1, 684 bits ($\log_2(256!)$), a number greater than 100,000 bits, corresponding to more than $10^{32,448}$ states. Although it is not infinity, it can be treated as very close to infinity. Therefore, due to the achievable high entropy, QPP can be used to replace a physical quantum key distribution system (see [29]) over the existing Internet.

7.2 Quantum data implementation

QPP can universally work over classical or quantum networks. The only difference is the underneath implementation, which is either with permutation matrices or quantum permutation gates. At the outset, note that QPP does permutation of probability amplitudes of all possible states, in contrast to permutation of qubits that are only associated with transpositions of their positions . For example, with n equal to two, there are only two qubit permutation gates associated with their position transpositions: identity and SWAP. However, there are four probability amplitude permutations of their states.

In a n qubit system, there are $2^n!$ permutation gates. For n equal to three, there are 40, 320 different eight-by-eight permutation gates. For n equal to four, there are more than 2×10^{13} different 16-by-16 permutation gates. It is obvious that the physical implementation of such a quantum system is challenging for a quantum secure communication. The actualization would realize the actual permutation gate implementation of 4-qubits. Built on a layered quantum computer architectures [22, 43], QPP would be implemented as an application in a quantum application layer. A library would create the permutation gates and address the detailed implementation at the physical layer. Shende et al. proposed an algorithm to create generic permutation gates with NOT, CNOT and TOFFOLI gate [47].

Quantum implementations of QPP can be envisioned as the technology will evolve. For the sake of simplicity, QPP can be physically implemented for a few qubit systems, such as two or three qubits. QPP can also be implemented with one-qubit system, where permutation gates are randomly selected from the identity gate and Pauli gate X, which is the traditional QKD with a pre-shared pad for encoding and measuring bases.

7.3 Secret key sharing

In QPP, it is assumed that the two parties, e.g., a message sender and receiver, share a secret key. We discuss briefly how a secret key can be shared. There are indeed several different ways to establish a pre-shared secret, such as through a public key infrastructure (PKI), out-of-band communication or provisioning by a system admin. PKI leverages a trusted certificate authority (CA) and the transport layer security (TLS)/secure sockets layer (SSL) to share a secret key signed with a public key [1, 21, 51]. Hence, a public key exchange algorithm such as RSA [45] or Diffie–Hellman [16] can be used to establish a shared key with classical cryptography. One may also consider using one of the PQC algorithms for key exchange [38]. Out-of-band communication means that the secret is exchanged over a channel separate from the data channel, such as over a voice call. Provisioning by a system admin is a very common process in any typical organization for first time establishment of a trusted relationship. Note that QKD boxes do not work without an initial pre-configured shared secret at the system provisioning phase. The QKD postprocess requires authentication with a pre-shared secret. In our proposed QPP universal quantum-safe cryptographic system, we take that pre-shared secret is to create permutation gates P and P^T. P is a n-bit permutation gate, behaving like a n-qubit QKD transmission box. P^T, the inverse of P, behaves like a n-qubit QKD receiving box.

In a short, secret sharing is not simply an extra requirement but a part of a practical deployment of a trusted secure communication system. With that in mind, it is possible to implement quantum secure communications digitally with QPP.

8 Conclusion

This article presented the quantum-safe QPP cryptographic system. It runs either on the current classical Internet or the upcoming quantum Internet. This is the reason why we argue that QPP cryptography is universal, for both classical and quantum systems. It can be implemented on classical computer and communication technology. It can be used now! It is also ready for the upcoming quantum Internet technology. The QPP algorithm is indeed quantum. At its core, it uses quantum permutation gates. It is implementable on quantum computers with quantum circuits. It is defined using quantum computing notation, where data items can be interpreted as column vectors or qubit registers. It has two security parameters, n and M. The first parameter (n) determines the size of the input–output alphabet (2^n symbols). The second parameter (M) specifies the number of permutation gates randomly selected from $2^n!$ permutation matrices. Together, they determine the symmetric key size, that is, $M \log_2(2^n!)$ bits. It is described using quantum mechanics formalism but does not use quantum-level properties such as a no-cloning of entanglement of qubits.

qpp is resistant against brute force, known plaintext and ciphertext attacks. QPP ciphertext can be deterministically measured in a computational basis by adversaries. However, they cannot interpret the measurement results without knowing the QPP pad

secretly shared between the sender and receiver. QPP has Shannon's perfect secrecy. The results are uninterpretable. QPP is not sensitive to same-key double encryption.

Data availability All data generated or analyzed during this study are included in this published article (and its Supplementary Information files).

Appendix A: Quantum gate interpretation of QKD

Let us consider the one-qubit computational basis $\{|0\rangle, |1\rangle\}$. QKD can be expressed by quantum gate operations. In the computational basis $B_1 = \{|0\rangle, |1\rangle\}$, encoding can be interpreted as the application the identity gate $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ to the basis vectors $|0\rangle = [1, 0]^T$ and $|1\rangle = [0, 1]^T$. The encoding in the Hadamard basis $B_2 = \{|-\rangle, |+\rangle\}$, can be interpreted as of the application of the Hadamard gate $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Encoding of $|0\rangle$ is performed by multiplying with the Hadamard gate, that is, $H|0\rangle = |+\rangle$. Similarly, the encoding $|1\rangle$ corresponds to the product $H|1\rangle = |-\rangle$.

Quantum encoding in the Hadamard basis is equivalent to the Hadamard gate operation on a state in the computational basis. It transforms basis vectors into states with superposition, for the purposes of secure communications. The receiver also applies the Hadamard gate to restore the superposition states back to a computational basis state, before measuring. A quantum state is prepared and measured in the computational basis, but a quantum gate transforms it into a superposition state for its secure communication. The reverse gate operation brings the superposition state back to its original state for measuring in the computational basis. However, a mismatched gate selection at the receiver leads to the measurement of a superposition state. This outcome must be avoided. Note that this interpretation of QKD uses two quantum gates, namely the identity gate (I) and Hadamard gate (H).

Appendix B: Symmetric group S_2

In S_2, the permutation matrices are $P_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $P_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ (ref. [46]). P_1 is also the identity permutation gate I, and P_2 is the Pauli permutation gate X.

Lemma 3 *The Hadamard basis is the eigenbasis of* P_2.

Proof Permutation gate P_2 is equal to the sum $|+\rangle - |1\rangle$. □

Appendix C: Key generation, encryption and decryption example

In the following example, classical key material is mapped to permutation gates, using Fisher–Yates shuffling algorithm. QPP is used for encryption and decryption. Security parameters n is 2 while M is 5. For the sake of simplicity, confusion and diffusion are omitted.

C.1 Encryption

Let us consider the plaintext `Hello World` as a toy example. Using an ASCII character table, the plaintext has the following binary representation:

01001000	01100101	01101100	01101100	01101111	00000000
00100000	01010111	01101111	01110010	01101100	01100100

The binary representation is segmented into two-bit segments with decimal values of segments as follows:

```
1, 0, 2, 0, 1, 2, 1, 1, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 3, 0, 0,
0, 0, 0, 2, 0, 0, 1, 1, 1, 3, 1, 2, 3, 3, 1, 3, 0, 2, 1, 2,
3, 0, 1, 2, 1, 0
```

For this example, we use a simple permutation selection algorithm. Suppose that we have randomly selected five permutation matrices/gates:

$$P_0 = \begin{pmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&0 \\ 0&0&0&1 \end{pmatrix}, P_1 = \begin{pmatrix} 0&0&1&0 \\ 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&1 \end{pmatrix}, P_2 = \begin{pmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 1&0&0&0 \end{pmatrix},$$
$$P_3 = \begin{pmatrix} 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \end{pmatrix} \text{ and } P_4 = \begin{pmatrix} 1&0&0&0 \\ 0&0&0&1 \\ 0&1&0&0 \\ 0&0&1&0 \end{pmatrix}$$

Mapping every decimal value 0, 1, 2, and 3 of input segments to the column vectors $|0\rangle = [1000]^T$, $|1\rangle = [0100]^T$, $|2\rangle = [0010]^T$ and $|3\rangle = [0001]^T$. Let us ignore the randomization for this simple example and also take the position of a dispatching segment modulo $M = 5$ to be the dispatching index of QPP. Then, we perform $P_d\ |m\rangle = |c\rangle$ with m as the decimal value of the dispatching segment, the ciphertext decimal values $|c\rangle$ of segments are as follows:

```
2, 2, 2, 0, 3, 0, 1, 3, 3, 1, 3, 2, 3, 2, 2, 1, 1, 2, 1, 2, 1, 2, 1,
0, 0, 0, 2, 1, 3, 3, 2, 3, 3, 2, 2, 3, 1, 0, 0, 1, 2, 0, 0, 0, 3, 0,
1, 1
```

The corresponding ciphertext binary representation is:

10101000	11000111	11011110	10010110	01100110	01100100
00001001	11111011	11101011	01000001	10000000	11000101

It is clearly shown that the bit randomness is improved: the number of zero bits is 53 and the number of one bits is 43 in the plaintext `Hello World`, but the number of zero bits is 49 and the number of one bits is 47 in the ciphertext.

C.2 Decryption

The corresponding inverse permutation gates are:

$$P_0^T = \begin{pmatrix} 0&0&1&0 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{pmatrix}, P_1^T = \begin{pmatrix} 0&0&1&0 \\ 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&1 \end{pmatrix}, P_2^T = \begin{pmatrix} 0&0&0&1 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \end{pmatrix},$$
$$P_3^T = \begin{pmatrix} 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \end{pmatrix} \text{ and } P_4^T = \begin{pmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&1&0&0 \end{pmatrix}$$

Decryption uses the same process as the encryption, but with transposed QPP and perform $P_d^T \ |c\rangle = |m\rangle$. The application of every corresponding inverse permutation gate yields the original plaintext, in a decimal form:

```
1, 0, 2, 0, 1, 2, 1, 1, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 3, 0, 0,
0, 0, 0, 2, 0, 0, 1, 1, 1, 3, 1, 2, 3, 3, 1, 3, 0, 2, 1, 2, 3, 0,
1, 2, 1, 0
```

It corresponds to the the ASCII binary:

01001000	01100101	01101100	01101100	01101111	00000000
00100000	01010111	01101111	01110010	01101100	01100100

That is the decrypted plaintext: `Hello World`.

The pre-shared key is a bit sequence. To achieve quantum-level security, the key length can be anything greater than 256 bits. The classical key material is expanded to determine a QPP pad. For 256 bits of entropy, a two-qubit QPP pad requires at least 56 permutation matrices with a classical key of 256 to 448 bits. A three-qubit QPP pad requires 17 permutation matrices with a key of 256 to 408 bits. A four-qubit QPP pad needs six permutation matrices with a key of 256 to 384 bits.

References

1. Adams, C., Lloyd, S.: Understanding PKI: Concepts, Standards, and Deployment Considerations. Addison-Wesley (2003)
2. Aharonov, D., Ben-Or, M., Eban, E., Mahadev, U.: Interactive proofs for quantum computations. arXiv preprint arXiv:1704.04487 (2017)
3. Alagic, G., Broadbent, A., Fefferman, B., Gagliardoni, T., Schaffner, C., Jules, M.S.: Computational security of quantum encryption. In: International Conference on Information Theoretic Security, pp. 47–71. Springer (2016)
4. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange'a new hope. In: 25Th USENIX Security Symposium (USENIX security 16), pp. 327–343 (2016)
5. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G., Buell, D.A., et al.: Quantum supremacy using a programmable superconducting processor. Nature **574**(7779), 505–510 (2019)

6. Barbeau, M., Kranakis, E., Perez, N.: Authenticity, integrity, and replay protection in quantum data communications and networking. ACM Trans. Quantum Comput. **3**(2), 1–22 (2022)
7. Barker, E., Mouha, N.: NIST special publication 800–67 revision 2: recommendation for the triple data encryption algorithm (TDEA) block Cipher. Natl. Inst. Stand. Technol. (NIST) (2017). https://doi.org/10.6028/NIST.SP.800-67r2
8. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. Theor. Comput. Sci. **560**, 7–11 (2014)
9. Bennett, C.H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., Wootters, W.K.: Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels. Phys. Rev. Lett. **70**(13), 1895 (1993)
10. Broadbent, A., Wainewright, E.: Efficient simulation for quantum message authentication. In: International Conference on Information Theoretic Security, pp. 72–91. Springer (2016)
11. Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on Post-Quantum Cryptography. National Institute of Standards and Technology, https://nvlpubs.nist.gov/nistpubs/ir/2016/ NIST.IR.8105.pdf. Access: 2021-01-0
12. Costello, C., Jao, D., Longa, P., Naehrig, M., Renes, J., Urbanik, D.: Efficient compression of sidh public keys. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 679–706. Springer (2017)
13. Daniel, A., Lejla, B., et al.: Initial recommendations of long-term secure post-quantum systems. PQCRYPTO. EU. Horizon **2020** (2015)
14. Diaconis, P., Shahshahani, M.: The subgroup algorithm for generating uniform random variables. Probab. Eng. Inf. Sci. **1**(1), 15–32 (1987)
15. Diamanti, E., Lo, H.K., Qi, B., Yuan, Z.: Practical challenges in quantum key distribution. NPJ Quantum Inf. **2**(1), 1–12 (2016)
16. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (1976)
17. Fisher, R.A., Yates, F.: Statistical tables: For biological, agricultural and medical research. Oliver and Boyd (1938)
18. Giampouris, D.: Short review on quantum key distribution protocols. GeNeDis **2016**, 149–157 (2017)
19. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)
20. Hirschhorn, P.S., Hoffstein, J., Howgrave-Graham, N., Whyte, W.: Choosing NTRUEncrypt parameters in light of combined lattice reduction and MITM approaches. In: International Conference on Applied Cryptography and Network Security, pp. 437–455. Springer (2009)
21. Housley, R., Polk, T.: Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure. Wiley, Networking Council (2001)
22. Jones, N.C., Van Meter, R., Fowler, A.G., McMahon, P.L., Kim, J., Ladd, T.D., Yamamoto, Y.: Layered architecture for quantum computing. Phys. Rev. X **2**(3), 031007 (2012)
23. Joseph, D., Ghionis, A., Ling, C., Mintert, F.: Not-so-adiabatic quantum computation for the shortest vector problem. Phys. Rev. Res. **2**(1), 013361 (2020)
24. Kuang, R., Bettenburg, N.: Shannon perfect secrecy in a discrete hilbert space. In: 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 249–255. IEEE (2020)
25. Kuang, R., Lou, D., He, A., Conlon, A.: Quantum safe lightweight cryptography with Quantum Permutation Pad. In: 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), pp. 790–795. IEEE (2021)
26. Kuang, R., Lou, D., He, A., Conlon, A.: Quantum safe lightweight cryptography with quantum permutation pad. Adv. Sci., Technol. Eng. Syst. J. **6**, 401–405 (2021)
27. Kuang, R., Lou, D., He, A., McKenzie, C., Redding, M.: Pseudo quantum random number generator with quantum permutation pad. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 359–364. IEEE (2021)
28. Laudenbach, F., Pacher, C., Fung, C.H.F., Poppe, A., Peev, M., Schrenk, B., Hentschel, M., Walther, P., Hübel, H.: Continuous-variable quantum key distribution with gaussian modulation-the theory of practical implementations. Adv. Quantum Technol. **1**(1), 1800011 (2018)
29. Lou, D., Kuang, R., He, A.: Entropy transformation and expansion with Quantum Permutation Pad for 5G secure networks. In: The IEEE 21st International Conference on Communication Technology. IEEE (2021)

30. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. Discrete Mathematics and Its Applications, CRC Press (2018)
31. Menezes, A.J., Okamoto, T., Vanstone, S.A.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Trans. Inf. Theor. **39**(5), 1639–1646 (1993)
32. Misoczki, R., Tillich, J.P., Sendrier, N., Barreto, P.S.: MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In: 2013 IEEE international symposium on information theory, pp. 2069–2073. IEEE (2013)
33. Mosca, M.: Cybersecurity in an era with quantum computers: will we be ready? IEEE Secur. Priv. **16**(5), 38–41 (2018)
34. National Institute of Standards and Technology: Advanced Encryption Standard (AES). https://csrc.nist.gov/publications/detail/fips/197/final. Access: 2021-01-0
35. National Security Agency – Central Security Service: Quantum Key Distribution (QKD) and Quantum Cryptography (QC). National Institute of Standards and Technology, https://www.nsa.gov/what-we-do/cybersecurity/quantum-key-distribution-qkd-and-quantum-cryptography-qc/. Access: 2021-01-01 (2020)
36. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2010)
37. NIST: Report on post-quantum cryptography. Online: https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf (2016)
38. NIST: Post-quantum cryptography. Online: https://csrc.nist.gov/projects/post-quantum-cryptography (2021)
39. O'Connor, L.: On the distribution of characteristics in bijective mappings. In: Workshop on the Theory and Application of of Cryptographic Techniques, pp. 360–370. Springer (1993)
40. Perepechaenko, M., Kuang, R.: Quantum encrypted communication between two IMBQ systems using quantum permutation pad. In: 11th International Conference on Communications, Circuits and Systems (ICCCAS) (2022)
41. Peterson, L., Davie, B.: Computer Networks: A Systems Approach. Elsevier Science (2011)
42. Petzoldt, A., Bulygin, S., Buchmann, J.: Selecting parameters for the rainbow signature scheme. In: International Workshop on Post-Quantum Cryptography, pp. 218–240. Springer (2010)
43. Pirker, A., Dür, W.: A quantum network stack and protocols for reliable entanglement-based networks. New J. Phys. **21**(3), 033003 (2019)
44. Quantropi Toolkit Starter. https://github.com/quantropi/quantropi-toolkit-starter. Access: 2021-01-0 (2021)
45. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
46. Shannon, C.E.: Communication theory of secrecy systems. Bell Syst. Techn. J. **28**(4), 656–715 (1949)
47. Shende, V.V., Prasad, A.K., Markov, I.L., Hayes, J.P.: Synthesis of reversible logic circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **22**(6), 710–722 (2003)
48. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134. IEEE (1994)
49. Shor, P.W., Preskill, J.: Simple proof of security of the BB84 quantum key distribution protocol. Phys. Rev. Lett. **85**(2), 441 (2000)
50. St-Jules, M.: Secure Quantum Encryption. Master's thesis, School of Graduate Studies and Research, University of Ottawa, Ottawa, Ontario, Canada. (2016)
51. van Oorschot, P.: Computer Security and the Internet: Tools and Jewels from Malware to Bitcoin. Springer International Publishing, Information Security and Cryptography (2021)
52. Wainewright, E.: Ecient Simulation for Quantum Message Authentication. Master's thesis, School of Graduate Studies and Research, University of Ottawa, Ottawa, Ontario, Canada. (2016)
53. Wang, Y.: Revised quantum resistant public key encryption scheme RLCE and IND-CCA2 security for McEliece schemes. IACR Cryptol. ePrint Arch. **2017**, 206 (2017)
54. Weyl, H.: The Classical Groups: Their Invariants and Representations (PMS-1). Princeton Landmarks in Mathematics and Physics. Princeton University Press (2016)
55. Xu, F., Ma, X., Zhang, Q., Lo, H.K., Pan, J.W.: Secure quantum key distribution with realistic devices. Rev. Mod. Phys. **92**(2), 025002 (2020)
56. Zhong, H.S., Wang, H., Deng, Y.H., Chen, M.C., Peng, L.C., Luo, Y.H., Qin, J., Wu, D., Ding, X., Hu, Y., et al.: Quantum computational advantage using photons. Science **370**(6523), 1460–1463 (2020)

57. Zukowski, M., Zeilinger, A., Horne, M.A., Ekert, A.K.: “Event-ready-detectors” Bell experiment via entanglement swapping. Phys. Rev. Lett. **71**(26) (1993)