Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

UNIVERSAL WISER
PUBLISHER

Research Article

Post-Quantum Key Agreement Protocols Based on Modified Matrix-Power Functions over Singular Random Integer Matrix Semirings

Juan Pedro Hecht[1], **Hugo Daniel Scolnik**[1,2*]

[1] Master's degree in Information Security, Faculties of Economic Science, Exacts and Natural Sciences and Engineering, University of Buenos Aires, Buenos Aires, Argentina
[2] Computer Science Department, Faculty of Exact and Natural Sciences, University of Buenos Aires, Buenos Aires, Argentina
E-mail: hugo@dc.uba.ar

Received: 21 November 2024; **Revised:** 6 December 2024; **Accepted:** 23 December 2024

Abstract: Post-quantum cryptography is essential for securing digital communications against threats posed by quantum computers. Researchers have focused on developing algorithms that can withstand attacks from both classical and quantum computers, thereby ensuring the security of data transmissions over public networks. A critical component of this security is the key agreement protocol, which allows two parties to establish a shared secret key over an insecure channel. This paper introduces two novel post-quantum key agreement protocols that can be easily implemented on standard computers using rectangular or rank-deficient matrices, exploiting the generalizations of the matrix power function, which is a generator of NP-hard problems. We provide basic concepts and proofs, pseudocodes, and examples, along with a discussion of complexity.

Keywords: key agreement protocol, non-commutative algebraic cryptography, post-quantum cryptography, rank-deficient matrices, matrix power function

1. Introduction

1.1 *Post-Quantum Cryptography (PQC)*

Post-quantum cryptography focuses on developing algorithms that are resistant to quantum computer attacks [1]. Quantum machines can solve problems exponentially faster than classical computers [2], thereby compromising the current cryptographic methods. This urgency has led to the development of quantum-safe cryptographic algorithms for securing data transmission over public networks. A key application is the key agreement protocol [3] that enables two parties to establish a shared secret key over an insecure channel. This paper introduces post-quantum key agreement protocols using rectangular or rank-deficient matrices instead of square ones, leveraging generalizations of the matrix power function [4, 5, 6, 7, 8, 9], which generates NP-hard problems [3]. These generalizations provide a novel approach for constructing secure key encapsulation mechanisms (KEMs) [10].

DOI: https://doi.org/10.37256/cnc.3120256112

1.2 *The history of post-quantum cryptography: Why and how?*

Quantum computing originated in the early 1980s when Richard Feynman proposed using quantum mechanics to simulate physical processes [2]. The significance of quantum computing in cryptography was recognized in the mid-1990s, notably when Peter Shor introduced an algorithm in 1994 that factors large integers exponentially faster than any known classical algorithm [11]. This breakthrough has alarmed the cryptography community because many cryptographic protocols (e.g., RSA and ECC) depend on the difficulty of factoring large numbers or solving elliptic-curve discrete logarithmic problems [3]. Grover's quantum search algorithm further threatens symmetric encryption systems by accelerating brute-force attacks and reducing their security [12]. These developments highlight the potential threat that quantum computers pose to current cryptographic systems, thereby prompting the emergence of postquantum cryptography. Researchers are currently exploring mathematical problems that are resistant to both classical and quantum attacks, leading to lattice-based, code-based, multivariate, and hash-based cryptographic schemes [1]. These new protocols aim to secure communication against quantum threats while maintaining efficiency in classical hardware.

1.3 *The need for KEM procedures in HTTPS*

A critical component of contemporary Internet security is Hypertext Transfer Protocol Secure (HTTPS), which is employed to encrypt data exchanged between web browsers and servers. HTTPS utilizes the Transport Layer Security (TLS) protocol to secure connections, with handshaking functioning as the primary TLS process [13]. Handshaking establishes a shared secret key between the client (browser) and server, thereby facilitating secure communication over an insecure channel. Currently, handshaking relies on asymmetric cryptographic primitives such as RSA or elliptic curve cryptography (ECC) to exchange secret keys [3]. However, both the RSA and ECC are vulnerable to Shor's algorithm, indicating that a sufficiently powerful quantum computer can effectively compromise these cryptographic schemes, thereby jeopardizing the security of all HTTPS communications [1]. This presents a significant challenge, as HTTPS is extensively utilized for securing financial transactions, personal data, and other sensitive online information. Key Encapsulation Mechanisms (KEMs) have emerged as post-quantum alternatives to current key exchange mechanisms employed in HTTPS [10]. KEM facilitates the establishment of a shared secret key between two parties, even when communicating over an insecure network, by encapsulating the key within a ciphertext that can only be decrypted by the intended recipient. By implementing KEMs based on quantum-resistant problems, it is possible to secure the handshake process in TLS protocols, thereby ensuring that the security of HTTPS remains intact, even in the quantum-enabled future [1].

1.4 *The limitations of Quantum Key Distribution (QKD)*

Quantum Key Distribution (QKD) [2] is frequently regarded as a theoretically optimal solution to the key exchange problem; however, it has practical limitations that render it unsuitable for widespread implementation in standard client-server communications. QKD protocols such as BB84 and B92 [14] utilize quantum mechanical principles to provide an unbreakable key exchange, assuming the absence of implementation flaws. These protocols are demonstrably secure because any attempt to intercept the quantum key would perturb the quantum states, thereby alerting the communicating parties to the presence of an eavesdropper [2]. Notwithstanding its theoretical advantages, QKD is not feasible for securing common internet communication. QKD systems require specialized hardware, such as single-photon emitters and detectors, and typically rely on direct point-to-point optical fiber connections between communicating parties. For global, large-scale communication networks such as the Internet, where thousands or millions of clients connect to servers across vast distances, the requirement for dedicated quantum channels is both impractical and prohibitively expensive. Furthermore, the infrastructure required to support QKD is incompatible with current Internet infrastructure, which relies on a complex mesh of interconnected routers and switches. Although QKD is optimal for certain niche applications requiring the highest level of security (e.g., government or military communications), it is not a viable solution for securing everyday Internet traffic between clients and servers.

1.5 *Post-quantum KEM for practical quantum-safe security*

Given the limitations of QKD, the development of post-quantum KEMs is a viable approach to securing Internet communication. In contrast to QKD, KEMs do not require specialized hardware and can be implemented using standard communication protocols. Through the design of KEMs that exploit the computational hardness of problems, such as those arising from matrix power functions with rectangular or rank-deficient matrices [15], it is possible to provide quantum-resistant security that integrates seamlessly into existing systems such as HTTPS and TLS.

1.6 *NIST PQC standardization program*

Recently, NIST (USA) concluded a PQC standardization program [16] by defining new standards for KEM exchanges and digital PQC signatures (FIPS 203, 204, 205) [17]; specifically, for key exchange, the selected protocols were Crystals-Kyber ML-KEM-512, ML-KEM-768, and ML-KEM-1024 [18]. However, this solution, although otherwise robust, is susceptible to AI-driven side-channel attacks [19]. Consequently, it is imperative to investigate alternative approaches, such as the algebraic-based solution presented herein, with particular emphasis on the latter. Furthermore, our protocols are significantly more code-compact than the learning-with-errors (LWE) lattices implemented by Crystals-Kyber, while maintaining the semantic security and quantum security levels mandated by NIST. Motivated by this objective, we developed novel KEMs that offer practical and resilient solutions for securing key exchanges in the post-quantum era. The proposed protocols can be implemented on a large scale, providing quantum-safe security for millions of users without necessitating substantial modifications to the internet infrastructure.

2. First protocol: using rectangular matrices

We define the Rectangular Matrix Power Function (RMPF), a generalization of the original Matrix Power Function (MPF) [4, 5, 6, 7, 8, 9] and outline its properties. We then present a Key Agreement Protocol (KAP) based on the RMPF and, later, a second protocol, the Rank-Deficient Matrix Power Function (RDMPF) variant, which exploits rank-deficient matrices with standard linear algebra operations. We used the same notation as in the literature. Although we refer only to RMPF here, it should be noted that the same arguments apply to the RDMPF solution. Furthermore, recall that any rectangular matrix can be transformed into an equivalent square rank-deficient matrix by adding linear combinations of rows.

Definition 1. *Equidimensional (m,n) rectangular matrices of integers (specifically p-prime $\mathbb{Z}p$ field elements) form an RM set, a (semi)ring structure with p-modular sums and p-modular Hadamard products (modular operations preserve numbers under a constant format).*

Definition 2. *Matrix elements of RM-set n powers are formed with the p-modular n powers of each element of the base matrix. Therefore, the product of the x-power of a W-element and y-power of the same element commutes (Wx.Wy = Wy.Wx) because the integer exponent products x and y commute. Hereafter, this paper deals only with RM sets when rectangular matrices are invoked.*

Definition 3. *Given any three matrices (X,W,C) of the same (m, n) RM set, the left-sided rectangular matrix power function (RMPF) exponential action of X over W is defined as the matrix* $C\!=\!\left\{\mathbf{c_{ij}}\right\}$*:*

$$\mathbf{X} \triangleright \mathbf{W} \equiv {}^{\mathbf{X}}\mathbf{W} = \mathbf{C},\ \mathbf{where}\ \mathbf{c_{ij}} = \prod_{\mathbf{k=1}}^{\mathbf{rank[X]}} \mathbf{w_{kj}}^{\mathbf{x_{ik}}} \tag{1}$$

Definition 4. *Given any three matrices (W, Y, D) of the same (m, n) RM set, the right RMPF exponential action of Y over W is equal to D;* $\mathbf{D}\!=\!\left\{\mathbf{d_{ij}}\right\}$:

$$\mathbf{W} \lhd \mathbf{Y} \ \equiv \mathbf{W}^{\mathbf{Y}} = \mathbf{D}, \ \textbf{where } \mathbf{d_{ij}} = \prod_{\mathbf{l=1}}^{\mathbf{rank[Y]}} \mathbf{w_{il}}^{\mathbf{y_{lj}}} \tag{2}$$

Definition 5. *Given any four matrices (X, W, Y, Q) of the same (m, n) RM set, the double-sided RPMF exponential action of matrix W with the left-sided X-matrix action and the right-sided Y-matrix action is defined as Q, such that* $\mathbf{Q} = \{\mathbf{q_{ij}}\}$*:*

$$\mathbf{X} \rhd \mathbf{W} \lhd \mathbf{Y} \equiv {}^{\mathbf{X}}\mathbf{W}^{\mathbf{Y}} = \mathbf{Q}, \textbf{where } \mathbf{q_{ij}} = \prod_{\mathbf{k=1}}^{\mathbf{rank(X)}} \prod_{\mathbf{l=1}}^{\mathbf{rank[Y]}} \mathbf{w_{kl}}^{\mathbf{x_{ik}} \cdot \mathbf{y_{lj}}} \tag{3}$$

Lemma 1. *The RPMF is unilaterally associative, as Sakalauskas proved [7], if the following identities hold.*

$${}^{\mathbf{Y}}({}^{\mathbf{X}}\mathbf{W}) = {}^{(\mathbf{YX})}\mathbf{W} = {}^{\mathbf{YX}}\mathbf{W}; \ (\mathbf{W}^{\mathbf{X}})^{\mathbf{Y}} = \mathbf{W}^{(\mathbf{XY})} = \mathbf{W}^{\mathbf{XY}} \tag{4}$$

and it is two-sided associative if:

$$({}^{\mathbf{X}}\mathbf{W})^{\mathbf{Y}} = {}^{\mathbf{X}}(\mathbf{W}^{\mathbf{Y}}) = {}^{\mathbf{X}}\mathbf{W}^{\mathbf{Y}} \tag{5}$$

whereas the RMPF is defined as associative if both conditions hold.

Lemma 2 (m, n)**.** *RM sets obey the associative properties of an RMPF. This is a special case of Sakalauskas' proof [14] because the square (m, m) matrices are replaced by the particular case of (m,n) rectangular matrices.*

Lemma 3. *If (X, Y, U, V, W) are (m, n) RM-set matrices acting as one-sided (left or right) RMPF actions over another W and (X, U), (Y, V) are pairs of outer (ordinary) products, then both commutative conditions hold:*

$$\mathbf{X^T} .\ \mathbf{U} \ = \mathbf{U^T} .\ \mathbf{X}; \mathbf{Y^T} .\ \mathbf{V} \ = \mathbf{V^T} . \mathbf{Y} \tag{6}$$

if the RMPFs of RM sets are associative (Equations (4) and (5)), then:

$${}^{\mathbf{U}}({}^{\mathbf{X}}\mathbf{W}^{\mathbf{Y}})^{\mathbf{V}} = {}^{\mathbf{UX}}\mathbf{W}^{\mathbf{YV}} = {}^{\mathbf{XU}}\mathbf{W}^{\mathbf{VY}} = {}^{\mathbf{X}}({}^{\mathbf{U}}\mathbf{W}^{\mathbf{V}})^{\mathbf{Y}} \tag{7}$$

Proof. If pairwise outer products commute, the elements of their square product matrix exponents can be interchanged (see Definition 2, properties applied to Equations (1)–(3)). Therefore, (7) holds. □

Lemma 4. *If* $(\lambda \mathbf{1}, \lambda \mathbf{2}) \in \mathbb{Z}^{\mathbf{2}}$ **and** $(\mathbf{X}, \mathbf{U}, \mathbf{W})$ *are members of the same RM set, then the scalar products* $\lambda \mathbf{1}.\mathbf{W} = \mathbf{X}$ **and** $\lambda \mathbf{2}.\mathbf{W} = \mathbf{U}$ *are matrices that satisfy condition (6).*

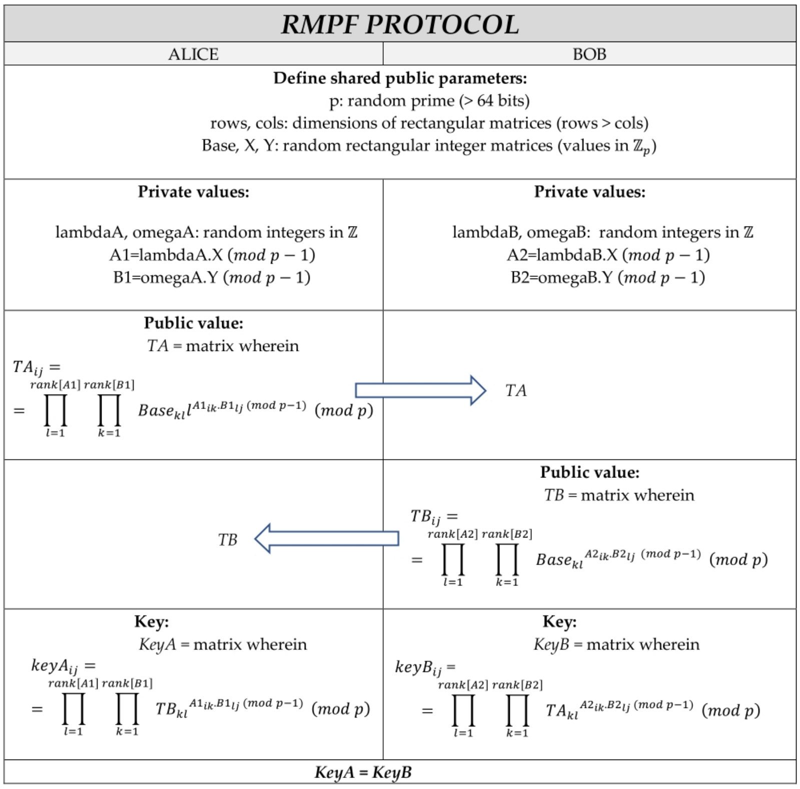
Proof. Given an RM-set matrix $\mathbf{W} = \{\mathbf{w_{ij}}\}$, then $\lambda \mathbf{W} = \{\lambda \mathbf{w_{ij}}\}$ and $\mathbf{oW} = \{\mathbf{ow_{ij}}\}$, as $(\boldsymbol{\lambda}, \mathbf{o}) \in \mathbb{Z}^2$ then $\lambda \mathbf{w_{ij}}.\mathbf{ow_{ij}} = \mathbf{ow_{ij}}.\ \lambda \mathbf{w_{ij}} = (\mathbf{o}.\lambda)\mathbf{w_{ij}^2} = (\lambda.\mathbf{o})\mathbf{w_{ij}^2}$ and therefore, condition (6, 7) hold. □

Lemma 5. *keyA = keyB*

Proof. Considering (6) and (7), **KeyA** = **A1** ▷**TB**◁ **B1** = **A1** ▷ (**A2** ▷ **Base**◁ **B2**) ◁ **B1** = **A2** ▷ (**A1** ▷ **Base**◁ **B1**) ◁ **B2** = **A2** ▷**TA**◁ **B2** = **KeyB**. □

3. Full description of the proposed RMPF action

3.1 *First schematic outline: Key-Agreement Protocol (KAP): see Figure 1*



RMPF PROTOCOL

ALICE	BOB
Define shared public parameters: p: random prime (> 64 bits) rows, cols: dimensions of rectangular matrices (rows > cols) Base, X, Y: random rectangular integer matrices (values in $\mathbb{Z}_p$)	

Private values:
lambdaA, omegaA: random integers in $\mathbb{Z}$
A1=lambdaA.X $(mod\ p-1)$
B1=omegaA.Y $(mod\ p-1)$
	Private values:
	lambdaB, omegaB: random integers in $\mathbb{Z}$
	A2=lambdaB.X $(mod\ p-1)$
	B2=omegaB.Y $(mod\ p-1)$

Public value: TA = matrix wherein $TA_{ij} = \\ = \prod_{l=1}^{rank[A1]} \prod_{k=1}^{rank[B1]} Base_{kl}l^{A1_{ik} \cdot B1_{lj}\ (mod\ p-1)}\ (mod\ p)$	$\Rightarrow$ TA
TB $\Leftarrow$	**Public value:** TB = matrix wherein $TB_{ij} = \\ = \prod_{l=1}^{rank[A2]} \prod_{k=1}^{rank[B2]} Base_{kl}{}^{A2_{ik} \cdot B2_{lj}\ (mod\ p-1)}\ (mod\ p)$
Key: $KeyA$ = matrix wherein $keyA_{ij} = \\ = \prod_{l=1}^{rank[A1]} \prod_{k=1}^{rank[B1]} TB_{kl}{}^{A1_{ik} \cdot B1_{lj}\ (mod\ p-1)}\ (mod\ p)$	**Key:** $KeyB$ = matrix wherein $keyB_{ij} = \\ = \prod_{l=1}^{rank[A2]} \prod_{k=1}^{rank[B2]} TA_{kl}{}^{A2_{ik} \cdot B2_{lj}\ (mod\ p-1)}\ (mod\ p)$
KeyA = KeyB	

Figure 1. The proposed key agreement protocol, which is based on RMPF

3.2 *RMPF pseudocode description*

Setup

Both parties (Alice and Bob) agree:

A prime p.

The RM set dimensions (m, n), where m > n, and the applicable operations are governed by Hadamard algebra.

The parties share three RM random matrices with bases, X and Y, with values in ℤp.

Alice's private values

lambdaA, omegaA: random numbers in ℤ

A1 = lambdaA.X (mod p-1), B1 = omegaA.Y (mod p-1)

Alice's public token

Generate the TA matrix; $\{TA_{ij}\}$

$$\mathbf{TA_{ij}} = \prod\nolimits_{\mathbf{l=1}}^{\mathbf{rank[A1]}} \prod\nolimits_{\mathbf{k=1}}^{\mathbf{rank[B1]}} \mathbf{Base_{kl}}^{\mathbf{A1_{ik} \cdot B1_{lj}}\ \ (\mathbf{mod\ p-1})}\ (\mathbf{mod\ p})$$

TA is sent to Bob.

Bob's private values

lambdaB, omegaB: random numbers in $\mathbb{Z}$

A2 = lambdaB.X (mod p-1), B2 = omegaB.Y (mod p-1)

Bob's public token

Generate the TB matrix; $\{TB_{ij}\}$

$$\mathbf{TB_{ij}} = \prod_{l=1}^{\mathbf{rank}[\mathbf{A2}]} \prod_{k=1}^{\mathbf{rank}[\mathbf{B2}]} \mathbf{Base_{kl}}^{\mathbf{A2_{ik}.B2_{lj}}\ (\mathbf{mod\ p-1})}\ (\mathbf{mod\ p})$$

TB is sent to Alice.

Shared key

Alice and Bob generate keyA and keyB matching matrices, respectively. $\{keyA_{ij}, keyB_{ij}\}$

$$\mathbf{keyA_{ij}} = \prod_{l=1}^{\mathbf{rank}[\mathbf{A1}]} \prod_{k=1}^{\mathbf{rank}[\mathbf{B1}]} \mathbf{TB_{kl}}^{\mathbf{A1_{ik}.B1_{lj}}\ (\mathbf{mod\ p-1})}\ (\mathbf{mod\ p})$$

$$\mathbf{keyB_{ij}} = \prod_{l=1}^{\mathbf{rank}[\mathbf{A2}]} \prod_{k=1}^{\mathbf{rank}[\mathbf{B2}]} \mathbf{TA_{kl}}^{\mathbf{A2_{ik}.B2_{lj}}\ (\mathbf{mod\ p-1})}\ (\mathbf{mod\ p})$$

3.3 *A toy numeric example of the RMPF Key-Agreement Protocol (KAP)*

This example follows the first protocol.

Defining prime as **p** =**65537**, **rows** = **5**, **cols** = **3**, it follows:

$$\mathbf{Base} = \begin{pmatrix} 44664 & 10605 & 58177 \\ 37079 & 44866 & 49280 \\ 45409 & 15609 & 726 \\ 57731 & 9471 & 41234 \\ 52116 & 32253 & 872 \end{pmatrix} \tag{8}$$

$$\mathbf{X} = \begin{pmatrix} 25454 & 62439 & 63614 \\ 39060 & 9694 & 46468 \\ 6392 & 43055 & 15148 \\ 26377 & 42964 & 30474 \\ 55812 & 12484 & 59987 \end{pmatrix} \tag{9}$$

$$\mathbf{Y} = \begin{pmatrix} 32239 & 25090 & 32249 \\ 31554 & 15896 & 40908 \\ 53836 & 29341 & 55133 \\ 49046 & 44776 & 7840 \\ 53994 & 48994 & 62776 \end{pmatrix} \tag{10}$$

$$\mathbf{lambdaA} = \mathbf{60308} \tag{11}$$

$$\mathbf{A1} = \begin{pmatrix} \mathbf{30104} & \mathbf{3724} & \mathbf{21208} \\ \mathbf{4496} & \mathbf{44632} & \mathbf{7248} \\ \mathbf{5984} & \mathbf{24620} & \mathbf{39280} \\ \mathbf{54324} & \mathbf{41616} & \mathbf{65480} \\ \mathbf{46672} & \mathbf{7504} & \mathbf{43260} \end{pmatrix} \tag{12}$$

$$\mathbf{omegaA} \ = \mathbf{36605} \tag{13}$$

$$\mathbf{B1} \ = \begin{pmatrix} \mathbf{1843} & \mathbf{63482} & \mathbf{40213} \\ \mathbf{27706} & \mathbf{44472} & \mathbf{5276} \\ \mathbf{64796} & \mathbf{23337} & \mathbf{27881} \\ \mathbf{35646} & \mathbf{35656} & \mathbf{1056} \\ \mathbf{15682} & \mathbf{32730} & \mathbf{26712} \end{pmatrix} \tag{14}$$

$$\mathbf{TokenA} \ = \begin{pmatrix} \mathbf{19050} & \mathbf{55225} & \mathbf{32116} \\ \mathbf{20307} & \mathbf{33635} & \mathbf{46068} \\ \mathbf{50694} & \mathbf{64046} & \mathbf{51330} \\ \mathbf{1754} & \mathbf{3460} & \mathbf{4352} \\ \mathbf{50272} & \mathbf{26460} & \mathbf{52031} \end{pmatrix} \tag{15}$$

$$\mathbf{lambdaB} \ = \mathbf{25401} \tag{16}$$

$$\mathbf{A2} \ = \begin{pmatrix} \mathbf{44414} & \mathbf{41839} & \mathbf{3598} \\ \mathbf{13556} & \mathbf{18542} & \mathbf{30308} \\ \mathbf{30520} & \mathbf{40823} & \mathbf{12492} \\ \mathbf{27649} & \mathbf{23092} & \mathbf{24378} \\ \mathbf{5860} & \mathbf{42916} & \mathbf{17787} \end{pmatrix} \tag{17}$$

$$\mathbf{omegaB} \ = \mathbf{64763} \tag{18}$$

$$\mathbf{B2} \ = \begin{pmatrix} \mathbf{48469} & \mathbf{4086} & \mathbf{40739} \\ \mathbf{53686} & \mathbf{33160} & \mathbf{32004} \\ \mathbf{132} & \mathbf{60399} & \mathbf{46127} \\ \mathbf{32786} & \mathbf{56696} & \mathbf{34528} \\ \mathbf{9070} & \mathbf{7446} & \mathbf{36328} \end{pmatrix} \tag{19}$$

$$\mathbf{TokenB} \ = \begin{pmatrix} \mathbf{8616} & \mathbf{10721} & \mathbf{1187} \\ \mathbf{43735} & \mathbf{40329} & \mathbf{8281} \\ \mathbf{52007} & \mathbf{53646} & \mathbf{42109} \\ \mathbf{20747} & \mathbf{37614} & \mathbf{61557} \\ \mathbf{18153} & \mathbf{19017} & \mathbf{15289} \end{pmatrix} \tag{20}$$

$$\mathbf{KeyA} = \begin{pmatrix} \mathbf{23030} & \mathbf{13518} & \mathbf{44672} \\ \mathbf{8819} & \mathbf{10151} & \mathbf{12163} \\ \mathbf{21} & \mathbf{40471} & \mathbf{6436} \\ \mathbf{45352} & \mathbf{62662} & \mathbf{60452} \\ \mathbf{9532} & \mathbf{30007} & \mathbf{11905} \end{pmatrix} \tag{21}$$

$$\mathbf{KeyB} = \begin{pmatrix} \mathbf{23030} & \mathbf{13518} & \mathbf{44672} \\ \mathbf{8819} & \mathbf{10151} & \mathbf{12163} \\ \mathbf{21} & \mathbf{40471} & \mathbf{6436} \\ \mathbf{45352} & \mathbf{62662} & \mathbf{60452} \\ \mathbf{9532} & \mathbf{30007} & \mathbf{11905} \end{pmatrix} \tag{22}$$

3.4 *Real life parameters*

The success rate of a brute-force attack exponentially decreases as the matrix order increases. In this context, however, this phenomenon is not applicable because the (X, Y) matrices are public, and security relies on secret lambda and omega integers. It is recommended to utilize RMPF Protocol matrices with a rank on the order of 100 and p ~ 2^64 as the minimum values. Consequently, any attack against two random integers in ℤp corresponds to a 128-bit brute force attack.

4. Second protocol using rank-deficient matrices

4.1 *Second schematic outline: Multi-round Key-Agreement Protocol (KAP): see Figure* 2

4.2 *RDMPF pseudocode description*

We developed the multi-PQC-RDMPF-KAP algorithm, which generally obeys our rectangular protocol with the following variants:

o Replace rectangular matrices with equivalent rank-deficient matrices.

o Replace the use of Hadamard algebra with conventional linear algebra, which reduces the feasibility of linearization attacks by replacing numerical powers in private matrices with multilinear inner products.

o Operate in rounds and consolidate partial keys into a combined key.

o Add a cryptographically secure hash function (SHA3-512) to create a 512-bit shared session key, thereby adding a random oracle to the protocol [20].

Here we describe one round of the rank-deficient algorithm:

Public setup parameters:

p prime

dim = rows = cols ε Z

Full-Rank nucleus matrix: W (mod p)

Rank-deficient matrices {BaseXU, BaseYV} (mod p)

Rounds ε Z

expMax ε Z

Alice's private keys:

$X = BaseXU^{randX}$ (mod p-1), $Y = BaseYV^{randY}$ (mod p-1)

where {randX, randY} ε Z_{expMax}

Alice's public key: (token for Bob):

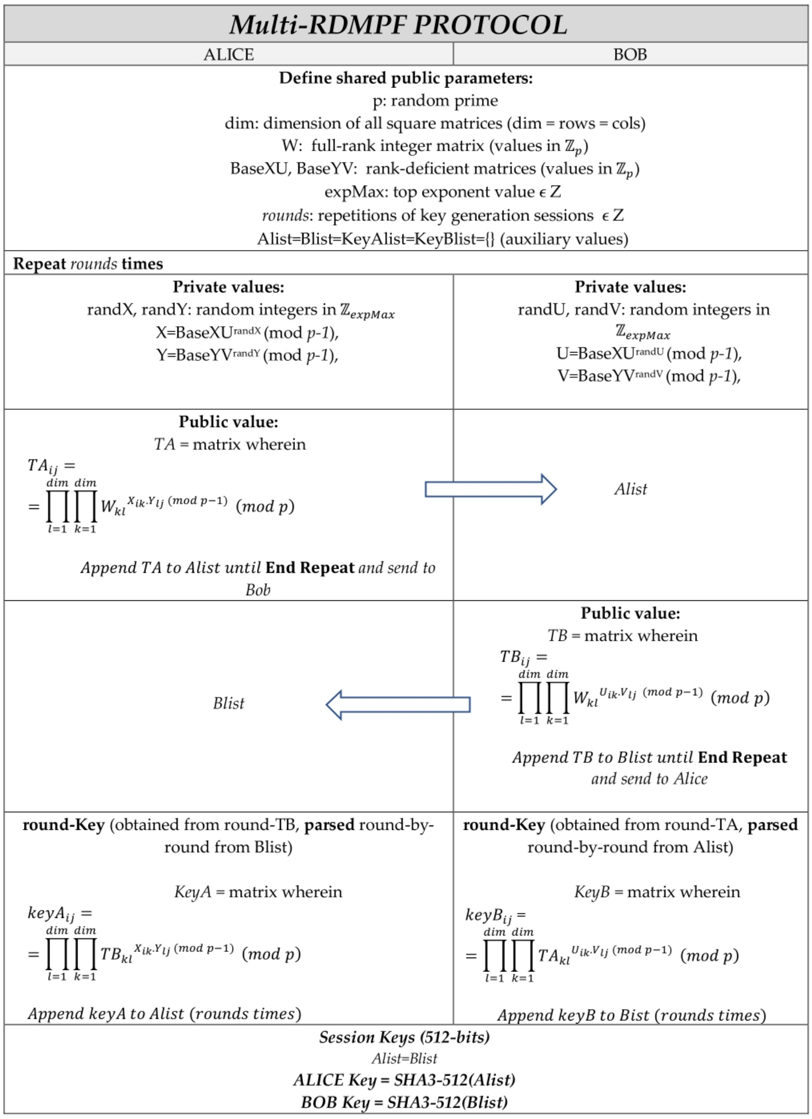
TA = RDMPF(X,W,Y) (mod p); RDMPF( ) is defined in Algorithm 1 pseudocode 1.

Bob's private keys:

$U = BaseXU^{randU}$ (mod p-1), $V = BaseYV^{randV}$ (mod p-1) where {randU, randV} ε Z_{expMax}

Bob's public key: (token for Alice):

TB = RDMPF(U,W,V) (mod p) ; RDMPF( ) is defined in Algorithm 1 pseudocode 1.



Multi-RDMPF PROTOCOL	
ALICE	BOB
Define shared public parameters: p: random prime dim: dimension of all square matrices (dim = rows = cols) W: full-rank integer matrix (values in $\mathbb{Z}_p$) BaseXU, BaseYV: rank-deficient matrices (values in $\mathbb{Z}_p$) expMax: top exponent value ϵ Z *rounds*: repetitions of key generation sessions ϵ Z Alist=Blist=KeyAlist=KeyBlist={} (auxiliary values)	
Repeat *rounds* **times**	

Private values:
randX, randY: random integers in $\mathbb{Z}_{expMax}$
X=BaseXU^{randX} (mod *p-1*),
Y=BaseYV^{randY} (mod *p-1*),
	Private values:
	randU, randV: random integers in $\mathbb{Z}_{expMax}$
	U=BaseXU^{randU} (mod *p-1*),
	V=BaseYV^{randV} (mod *p-1*),

Public value: *TA* = matrix wherein $TA_{ij} = = \prod_{l=1}^{dim}\prod_{k=1}^{dim} {W_{kl}}^{X_{ik}.Y_{lj}\ (mod\ p-1)}\ (mod\ p)$ *Append TA to Alist until* **End Repeat** *and send to Bob* →	*Alist*
Blist ←	**Public value:** *TB* = matrix wherein $TB_{ij} = = \prod_{l=1}^{dim}\prod_{k=1}^{dim} {W_{kl}}^{U_{ik}.V_{lj}\ (mod\ p-1)}\ (mod\ p)$ *Append TB to Blist until* **End Repeat** *and send to Alice*
round-Key (obtained from round-TB, **parsed** round-by-round from Blist) *KeyA* = matrix wherein $keyA_{ij} = = \prod_{l=1}^{dim}\prod_{k=1}^{dim} {TB_{kl}}^{X_{ik}.Y_{lj}\ (mod\ p-1)}\ (mod\ p)$ *Append keyA to Alist (rounds times)*	**round-Key** (obtained from round-TA, **parsed** round-by-round from Alist) *KeyB* = matrix wherein $keyB_{ij} = = \prod_{l=1}^{dim}\prod_{k=1}^{dim} {TA_{kl}}^{U_{ik}.V_{lj}\ (mod\ p-1)}\ (mod\ p)$ *Append keyB to Bist (rounds times)*
Session Keys (512-bits) *Alist=Blist* ***ALICE Key = SHA3-512(Alist)*** ***BOB Key = SHA3-512(Blist)***	

Figure 2. The multi-round Key-Agreement Protocol, which is based on the RDMPF

Remark 1. *The sporadic occurrence of null matrices at the token level should be controlled, in which case the round is restarted. Once the tokens have been exchanged, both can compute the same key, Alice's key is KeyA = RDMPF(X,TB,Y) (mod p), and Bob's Key is KeyB = RDMPF(U,TA,V) (mod p)*

The tokens (public keys) of the rounds are concatenated sequentially, row by row, and exchanged:

Alice: Alist = A1 … Arounds, and Bob: Blist = B1 … Brounds

Based on the tokens (public keys) received from the counterpart and retrieved round-by-round, the respective keys of each round are obtained (see Remark 1).

The keys are concatenated sequentially row by row:

Alice: KeyAlist = KeyA1 … KeyArounds, and Bob: Blist = KeyB1 … KeyBrounds

The following 512-bit session keys were obtained:

KsessionA = SHA3-512(KeyAlist)

KsessionB = SHA3-512(KeyBlist)

Here we describe the Rank Deficient Matrix Power Function (RDMPF) using Mathematica (pseudocode-like) format. This algorithm illustrates the RDMPF in full detail. Note that there is no simple description of the entire matrix, and only element-by-element definitions. This function can be interpreted as the inner product exponential version of the ordinary matrix product that occurs when left and right matrix actions are applied. Expressed in simple terms, ordinary sums are transformed into products and ordinary products are transformed into exponential operations. This is the source of the overall complexity, which blocks both classical and quantum attacks.

Algorithm 1 pseudocode 1

```
RDMPFint[X_,W_,Y_]:= Module[{i, j, L, K, z, ex, pr}, (* local variables *)
Q = Rmat[rows, cols, prime]; (* modular random integer matrix, here rows = cols = dim *)
Do[
  Do[
    pr = 1;
    Do[
      Do[
        ex = Mod[Times[X[[i,K]],Y[[L,j]]], prime-1]; (*modular product *)
        z = IntFastPower[W[[K,L]], ex]; (* square-and-multiply power function*)
        pr = Times[pr, z];
      {L, 1, cols}]; (* end Do *)
    {K, 1, cols}]; (* end Do *)
    Q[[i, j]] = Mod[pr, prime];
  {j, cols}] (* end Do *)
{i, rows}]; (* end Do *)
Q]; (* output *)
```

Here is a small symbolic example:

Algorithm 2 pseudocode 2

```
dim = rows = cols = 2;
W = {{$w_{11}$, $w_{12}$}, {$w_{21}$, $w_{22}$}};
X = {{$x_{11}$, $x_{12}$}, {$x_{21}$, $x_{22}$}};
Y = {{$y_{11}$, $y_{12}$}, {$y_{21}$, $y_{22}$}};
Q = RDMPF[X,W,Y];
Q =
{{$w11^{x11\ y11}\ w12^{x11\ y21}\ w21^{x12\ y11}\ w22^{x12\ y21}$, $w11^{x11\ y12}\ w12^{x11\ y22}\ w21^{x12\ y12}\ w22^{x12\ y22}$},
{$w11^{x21\ y11}\ w12^{x21\ y21}\ w21^{x22\ y11}\ w22^{x22\ y21}$, $w11^{x21\ y12}\ w12^{x21\ y22}\ w21^{x22\ y12}\ w22^{x22\ y22}$}}
```

4.3 *A toy numeric example of the multi RDMPF Key-Agreement Protocol (KAP)*

This example follows the second protocol. Rows repetitions (here 1, 2) are at random positions.

Defining prime as **p = 65,537**, **dim = 5, expMax = 10,000, rounds = 2**, it follows:

$$\mathbf{W} = \begin{pmatrix} \mathbf{36671} & \mathbf{1524} & \mathbf{19050} & \mathbf{12061} & \mathbf{61140} \\ \mathbf{5366} & \mathbf{34773} & \mathbf{37275} & \mathbf{10709} & \mathbf{60768} \\ \mathbf{59994} & \mathbf{8372} & \mathbf{16513} & \mathbf{19213} & \mathbf{18024} \\ \mathbf{22554} & \mathbf{1387} & \mathbf{10646} & \mathbf{57542} & \mathbf{54414} \\ \mathbf{62130} & \mathbf{15684} & \mathbf{5868} & \mathbf{17933} & \mathbf{2855} \end{pmatrix} \tag{23}$$

$$\mathbf{BaseXU} = \begin{pmatrix} \mathbf{57543} & \mathbf{23480} & \mathbf{42992} & \mathbf{19549} & \mathbf{59890} \\ \mathbf{57543} & \mathbf{23480} & \mathbf{42992} & \mathbf{19549} & \mathbf{59890} \\ \mathbf{43343} & \mathbf{28960} & \mathbf{64751} & \mathbf{37741} & \mathbf{48337} \\ \mathbf{1091} & \mathbf{62357} & \mathbf{30242} & \mathbf{50955} & \mathbf{3101} \\ \mathbf{37839} & \mathbf{36136} & \mathbf{38757} & \mathbf{10107} & \mathbf{12470} \end{pmatrix} \tag{24}$$

$$\mathbf{BaseYV} = \begin{pmatrix} \mathbf{61098} & \mathbf{25692} & \mathbf{18347} & \mathbf{31256} & \mathbf{2737} \\ \mathbf{61098} & \mathbf{25692} & \mathbf{18347} & \mathbf{31256} & \mathbf{2737} \\ \mathbf{23628} & \mathbf{47854} & \mathbf{30452} & \mathbf{10898} & \mathbf{3201} \\ \mathbf{4055} & \mathbf{43906} & \mathbf{6517} & \mathbf{25648} & \mathbf{29018} \\ \mathbf{13622} & \mathbf{59502} & \mathbf{23730} & \mathbf{40601} & \mathbf{41483} \end{pmatrix} \tag{25}$$

First round

$$\mathbf{randX} = \mathbf{4267} \tag{26}$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{25300} & \mathbf{53591} & \mathbf{3358} & \mathbf{6302} & \mathbf{15971} \\ \mathbf{25300} & \mathbf{53591} & \mathbf{3358} & \mathbf{6302} & \mathbf{15971} \\ \mathbf{59640} & \mathbf{62777} & \mathbf{43012} & \mathbf{50996} & \mathbf{33510} \\ \mathbf{8272} & \mathbf{23015} & \mathbf{13985} & \mathbf{6756} & \mathbf{47019} \\ \mathbf{64853} & \mathbf{6353} & \mathbf{9303} & \mathbf{16909} & \mathbf{11272} \end{pmatrix} \tag{27}$$

$$\mathbf{randY} = \mathbf{4651} \tag{28}$$

$$\mathbf{Y} = \begin{pmatrix} \mathbf{50294} & \mathbf{15396} & \mathbf{2447} & \mathbf{20604} & \mathbf{46313} \\ \mathbf{50294} & \mathbf{15396} & \mathbf{2447} & \mathbf{20604} & \mathbf{46313} \\ \mathbf{52856} & \mathbf{57814} & \mathbf{29792} & \mathbf{40618} & \mathbf{1969} \\ \mathbf{25287} & \mathbf{53714} & \mathbf{4577} & \mathbf{4384} & \mathbf{26014} \\ \mathbf{24014} & \mathbf{4806} & \mathbf{32294} & \mathbf{48601} & \mathbf{23187} \end{pmatrix} \tag{29}$$

$$\mathbf{randU} = \mathbf{6066} \tag{30}$$

$$\mathbf{U} = \begin{pmatrix} 61917 & 24420 & 29078 & 47059 & 18070 \\ 61917 & 24420 & 29078 & 47059 & 18070 \\ 35935 & 20952 & 51333 & 41093 & 16163 \\ 41155 & 1979 & 10882 & 17171 & 37033 \\ 38861 & 15750 & 29077 & 7509 & 61114 \end{pmatrix} \tag{31}$$

$$\mathbf{randV} = \mathbf{8472} \tag{32}$$

$$\mathbf{V} = \begin{pmatrix} 37353 & 1020 & 59757 & 44920 & 18981 \\ 37353 & 1020 & 59757 & 44920 & 18981 \\ 38256 & 24936 & 25399 & 44464 & 10051 \\ 36307 & 16166 & 52913 & 49849 & 13652 \\ 51670 & 11528 & 54954 & 50615 & 6058 \end{pmatrix} \tag{33}$$

$$\mathbf{Token\ A} = \begin{pmatrix} 53838 & 27572 & 60974 & 49207 & 54423 \\ 53838 & 27572 & 60974 & 49207 & 54423 \\ 7986 & 15752 & 8069 & 40416 & 15771 \\ 41410 & 8254 & 42646 & 57132 & 64087 \\ 62119 & 17840 & 19622 & 20589 & 6234 \end{pmatrix} \tag{34}$$

$$\mathbf{Token\ B} = \begin{pmatrix} 29348 & 1649 & 29136 & 53009 & 60590 \\ 29348 & 1649 & 29136 & 53009 & 60590 \\ 47894 & 18698 & 17669 & 19542 & 31170 \\ 5356 & 9122 & 3736 & 17535 & 33881 \\ 46266 & 10907 & 21467 & 39257 & 36010 \end{pmatrix} \tag{35}$$

$$\mathbf{KeyA} = \begin{pmatrix} 20743 & 10836 & 64775 & 35222 & 44472 \\ 20743 & 10836 & 64775 & 35222 & 44472 \\ 49310 & 2062 & 65040 & 46960 & 24883 \\ 40381 & 25492 & 38040 & 58289 & 65195 \\ 43404 & 25602 & 54209 & 59994 & 36225 \end{pmatrix} \tag{36}$$

$$\mathbf{KeyB} = \begin{pmatrix} 20743 & 10836 & 64775 & 35222 & 44472 \\ 20743 & 10836 & 64775 & 35222 & 44472 \\ 49310 & 2062 & 65040 & 46960 & 24883 \\ 40381 & 25492 & 38040 & 58289 & 65195 \\ 43404 & 25602 & 54209 & 59994 & 36225 \end{pmatrix} \tag{37}$$

Second round

$$\mathbf{randX} = \mathbf{6171} \tag{38}$$

$$\mathbf{X} = \begin{pmatrix} 20687 & 43044 & 29876 & 65277 & 34570 \\ 20687 & 43044 & 29876 & 65277 & 34570 \\ 48043 & 42272 & 30547 & 16281 & 53097 \\ 64011 & 43209 & 15826 & 58203 & 65225 \\ 59031 & 50156 & 13641 & 54627 & 6418 \end{pmatrix} \tag{39}$$

$$\mathbf{randY} = \mathbf{2414} \tag{40}$$

$$\mathbf{Y} = \begin{pmatrix} 40891 & 39362 & 36749 & 34923 & 28810 \\ 40891 & 39362 & 36749 & 34923 & 28810 \\ 22763 & 63190 & 28195 & 33540 & 27134 \\ 56708 & 35280 & 14969 & 48184 & 42201 \\ 38364 & 57222 & 24807 & 17310 & 52808 \end{pmatrix} \tag{41}$$

$$\mathbf{randU} = \mathbf{7574} \tag{42}$$

$$\mathbf{U} = \begin{pmatrix} 61547 & 33968 & 56069 & 41953 & 50743 \\ 61547 & 33968 & 56069 & 41953 & 50743 \\ 29714 & 32573 & 36652 & 42508 & 7927 \\ 33931 & 35041 & 24823 & 50021 & 61711 \\ 38392 & 28428 & 60085 & 13340 & 4043 \end{pmatrix} \tag{43}$$

$$\mathbf{randV} = \mathbf{1456} \tag{44}$$

$$\mathbf{V} = \begin{pmatrix} 17998 & 3012 & 8841 & 26426 & 43907 \\ 17998 & 3012 & 8841 & 26426 & 43907 \\ 49148 & 8686 & 26452 & 55316 & 51969 \\ 64313 & 53978 & 52641 & 4196 & 14662 \\ 51704 & 8754 & 12104 & 61813 & 36643 \end{pmatrix} \tag{45}$$

$$\mathbf{Token\ A} = \begin{pmatrix} 21108 & 54710 & 20029 & 6255 & 14963 \\ 21108 & 54710 & 20029 & 6255 & 14963 \\ 28723 & 28942 & 9398 & 51028 & 3356 \\ 44003 & 6940 & 4827 & 50400 & 35084 \\ 54653 & 19386 & 46270 & 24516 & 19667 \end{pmatrix} \tag{46}$$

$$\textbf{Token B} = \begin{pmatrix} \mathbf{31055} & \mathbf{8992} & \mathbf{38240} & \mathbf{47046} & \mathbf{52571} \\ \mathbf{31055} & \mathbf{8992} & \mathbf{38240} & \mathbf{47046} & \mathbf{52571} \\ \mathbf{53708} & \mathbf{5236} & \mathbf{39748} & \mathbf{56283} & \mathbf{63932} \\ \mathbf{27273} & \mathbf{31500} & \mathbf{58981} & \mathbf{63915} & \mathbf{16157} \\ \mathbf{21773} & \mathbf{26963} & \mathbf{14715} & \mathbf{52520} & \mathbf{13589} \end{pmatrix} \tag{47}$$

$$\textbf{KeyA} = \begin{pmatrix} \mathbf{33253} & \mathbf{42632} & \mathbf{21998} & \mathbf{52285} & \mathbf{49951} \\ \mathbf{33253} & \mathbf{42632} & \mathbf{21998} & \mathbf{52285} & \mathbf{49951} \\ \mathbf{14086} & \mathbf{35325} & \mathbf{53116} & \mathbf{60717} & \mathbf{41037} \\ \mathbf{3238} & \mathbf{39606} & \mathbf{1643} & \mathbf{48792} & \mathbf{26310} \\ \mathbf{19481} & \mathbf{30394} & \mathbf{40594} & \mathbf{46821} & \mathbf{12282} \end{pmatrix} \tag{48}$$

$$\textbf{KeyB} = \begin{pmatrix} \mathbf{33253} & \mathbf{42632} & \mathbf{21998} & \mathbf{52285} & \mathbf{49951} \\ \mathbf{33253} & \mathbf{42632} & \mathbf{21998} & \mathbf{52285} & \mathbf{49951} \\ \mathbf{14086} & \mathbf{35325} & \mathbf{53116} & \mathbf{60717} & \mathbf{41037} \\ \mathbf{3238} & \mathbf{39606} & \mathbf{1643} & \mathbf{48792} & \mathbf{26310} \\ \mathbf{19481} & \mathbf{30394} & \mathbf{40594} & \mathbf{46821} & \mathbf{12282} \end{pmatrix} \tag{49}$$

$$\mathbf{A} - \textbf{Token combined list} = \{\mathbf{53838}, \ldots (\textbf{48 terms}) \ldots, \mathbf{19667}\} \tag{50}$$

$$\mathbf{B} - \textbf{Token combined list} = \{\mathbf{29348}, \ldots (\textbf{48 terms}) \ldots, \mathbf{13589}\} \tag{51}$$

$$\mathbf{A} - \textbf{Keys combined list} = \{\mathbf{20743}, \ldots (\textbf{48 terms}) \ldots, \mathbf{12282}\} \tag{52}$$

$$\mathbf{B} - \textbf{Keys combined list} = \{\mathbf{20743}, \ldots (\textbf{48 terms}) \ldots, \mathbf{12282}\} \tag{53}$$

$$K\ session\ key\ from\ Alice = \text{"}\mathbf{1f18d824829ebc368c13} \ldots (total\ length:\ 64\ bytes)\text{"} \tag{54}$$

$$K\ session\ key\ from\ Bob = \text{"}\mathbf{1f18d824829ebc368c13} \ldots (total\ length:\ 64\ bytes)\text{"} \tag{55}$$

4.4 *The security of the RDMPF protocol depends on the following considerations*

Generation of Private Matrices {X, Y. U, V} from public setup parameters (BaseXU and BaseYV). Because the goal is to achieve commutativity of these matrices, the mechanism must provide classical and quantum security. Note, however, that these matrices interpenetrate as exponents and in pairs (x, y); therefore, their impact on integral security is limited. One option is that the definition of the base matrices should consider the use of primitive characteristic polynomials of the matrices (in a semi-random selection) to ensure a long multiplicative order, thereby making it possible to define sufficiently long periods to avoid repetition over the power cycles.

The complexity of the RDMPF function can be approximated as if n = matrix dimension (and usually its rank is less than or equal to n − 1). Each element of the resulting matrix uses n products of modular exponentiations of a known base, and exponents resulting from the inner product of two unknown vectors.

The naive bit complexity [21] of a product (k = ab) is O(log2a. log2b) and the modular exponentiation m = b^k (mod s) is O((log2s)^2. log2k) [22]. Therefore, assuming only unit complexity in the computation of each exponent, n-exponentiations span O(n(log2s)^2.log2k) bits and, using "square and multiple operations," the final computation of the resulting RDMPF matrix (proportional to n^2) will be O(n^3. (log2s)^2 .log2k) bits. Further complexity can be added using field operations (F2^p) instead of regular arithmetic [20]. The number of rounds does not improve the complexity but adds proportionally more time to any practical attack.

4.5 *Parameters sensitivity of the RDMPK key-agreement in relative time units*

See Table 1, some results obtained with unoptimized interpreted Mathematica code of the full RDMPF on an Intel Core i5-5200U CPU @2.20 GHz, for different parameters and only one round (1 unit equals 265 ms).

Table 1. Relative sensitivity computation of the RDMPF. The time values are the mean of 10 runs

Dimension	Prime	expMax	Relative time
Effect of a 5-fold dim change			
5	997	1000	1
25	997	1000	577
Effect of an approximate 5-fold prime change			
5	997	1000	1
5	4973	1000	1.41
Effect of a 5-fold expMax change			
5	997	1000	1
5	997	5000	1.01

This table shows RDMPF's high sensitivity to dimensional change, relative sensitivity to prime change, and insensitivity to maximum exponent change due to the "square-and-multiply" algorithm [3] used.

4.6 *Using truly random integer matrices*

It is important to emphasize that the development of MPF represents a significant advance over the purely algebraic approach to PQC protocols owing to its inherent simplicity and security. Although the focus has primarily been on the overall structure, subtle refinements have improved both the efficiency and robustness. One of the most notable limitations of the original Matrix Power Function is the exclusive consideration of circulant matrices or those derived from special algebraic groups in pursuit of commutativity, rather than truly random integer matrices. This limitation was overcome using the algorithms presented in this paper.

5. Discussion

5.1 *Implementation suggestions*

It is imperative to adapt this protocol to a Key Encapsulation Mechanism (KEM) to ensure that its intrinsic security complies with the NIST post-quantum standardization proposals [16]. This requires the addition of a postquantum public-key cryptosystem [1]. Other issues such as constant time or uniform power consumption should be considered to achieve side-channel attack protection [23]. The benefit achieved is the adaptation of semantic security to the IND-CCA level [24, 25, 26].

The following procedure presents the proposed KEM based on the multiRDMPF key-agreement protocol (KAP). Note that it cryptographically conceals all otherwise public information except the setup parameters, thereby enhancing security.

1. **ALGORITHMS USED:**

multiRDMPF (RD:Rank-Deficient) MPF-KAP

HMAC(key, SHA3-512)

2. **CHAINED NONCES**: η0 (shared secret, root nonce) , ηm(subsequent random nonces), all nonces have 512-bits. (Note: η0 can be obtained by a previous multiRDMPF KAP session or pre-distributed by separate channel, the nonces can be renewed using Lamport scheme [2]).

(|| *concatenation symbol*, ⊕ bitwise XOR)

3. **AUTHENTICATION TAGS**: authA, authB, public values, of 256-bits

4. BOB Starts: (Obs: Points 4., 5., 6. are repeated round by round and the Token TA, TB are used as A-Token combined list and B-Token combined list following the multiRDMPF. Each TA, TB is padded to 512-bits.

Bob has η0

He generates PriB, TB (multiRDMPF KAP).

Compute and Send CloseB to ALICE:

CloseB = HMAC (η0, authA || authB) ⊕ TB

5. **ALICE (ENCAPSULATION):**

Alice has η0

Receives CloseB

Compute TB = CloseB ⊕ HMAC (η0, authA || authB)

Generate PriA, TA, KeyA (multiRDMPF KAP)

Compute

CloseA = HMAC (η0, authA || authB ⊕ ηm) ⊕ TA

Generate a random key K (512-bit)

Encapsulate

cipher = HMAC (KeyA, [authA || authB] ⊕ ηm) ⊕ K

Send {Encap, CloseA, ηm} to BOB

6. **BOB (DECAPSULATION):**

Receives {Encap, CloseA, ηm} from ALICE.

Retrieves

TA = CloseA ⊕ HMAC (η0, [authA || authB] ⊕ ηm).

Compute KeyB (multiRDMPF KAP)

Decrypts

K = Encap ⊕ HMAC (KeyB, [authA || authB] ⊕ ηm).

6. Summary and conclusions

The two protocols use rectangular or rank-deficient matrices instead of the original MPF matrices. Both block algebraic linearization and Gröbner basis attacks [27, 28, 29, 30, 31, 32]. In addition, the second method, using rank-deficient square matrices that allow conventional linear algebra over singular matrices, significantly enhances internal complexity and adds resilience to algebraic attacks. The main objective is to introduce quantum-safe protocols using new algorithms that require neither special hardware nor extended precision. Consequently, they can easily be implemented on commercially available computers.

To safeguard against algebraic attacks [27, 28, 29, 30, 31, 32] and guarantee a well-defined numerical format, we incorporate p-modular operations in our protocols. We increase the entropy of key search spaces by replacing circulant matrices [4, 5, 6, 7, 8, 9] or restricted algebraic groups to achieve commutativity with unstructured random integers. An important aspect that should not be overlooked is the security of the pseudorandom bit generators. Therefore, we strongly recommend using algorithms that demonstrate resilience to side channels and quantum attacks, as described in [33]. One idea that can be implemented is the creation of a subliminal channel using an alternatively secure method, in which the parties privately agree upon a linear combination of rows to create rank-deficient matrices.

Supplementary materials

Mathematica 12 notebooks with all functions and numerical examples used in our KAP variants can be distributed upon request to phecht@dc.uba.ar.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] D. J. Bernstein, T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017. https://doi.org/10.1007/978-3-540-88702-7.

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, UK: Cambridge University Press, 2010.

[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.

[4] E. Sakalauskas, "Enhanced Matrix Power Function for Cryptographic Primitive Construction," *Symmetry*, vol. 10, no. 2, p. 43, 2018.

[5] E. Sakalauskas, N. Listopadskis, P. Tvarijonas, "The Key Agreement Protocol (KAP) Based on Matrix Power Function," *Adv. Stud. Softw. Knowl. Eng. Inf. Sci. Comput.*, vol. 2, no. 4, pp. 92–96, 2008.

[6] E. Sakalauskas and K. Luksys, "Matrix Power Function and Its Application to Block Cipher S-Box Construction," *Int. J. Innov. Comput. Inf. Control*, vol. 8, pp. 2655–2664, 2012.

[7] E. Sakalauskas and A. Mihalkovich, "Improved Asymmetric Cipher Based on Matrix Power Function Resistant to Linear Algebra Attack," *Informatica*, vol. 28, pp. 517–524, 2017.

[8] E. Sakalauskas and A. Mihalkovich, "MPF Problem Over Modified Medial Semigroup Is NP-Complete," *Symmetry*, vol. 10, no. 11, p. 571, 2018.

[9] E. Sakalauskas and A. Mihalkovich, "A New Asymmetric Cipher of Non-Commuting Cryptography Class Based on Enhanced MPF," *IET Inf. Sec.*, vol. 14, no. 4, pp. 410–418, 2020.

[10] S. Galbraith, "The KEM/DEM Paradigm," in *Mathematics of Public-Key Cryptography*. Cambridge, UK: Cambridge University Press, pp. 471–478, 2012.

[11] P. W. Shor, "Polynomial Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Rev.*, vol. 41, pp. 303–332, 1999.

[12] C. Zalka, "Grover's Quantum Searching Algorithm is Optimal," *Phys. Rev. A*, vol. 60, no. 4, p. 2746, 1999.

[13] I. McDonald and M. Sweet, "Internet Printing Protocol (IPP) Over HTTPS Transport Binding and URI Scheme," RFC 7472, 2015. Accessed: [Online]. Available: https://www.rfc-editor.org/rfc/rfc7472.

[14] M. H. Rempola, A. Smith, Y. Li, and L. Du, "Securing SDN Communication Through Quantum Key Distribution," in *Proc. 2024 IEEE Transp. Electr. Conf. Expo (ITEC)*, Chicago, IL, USA, Jun. 19–21, 2024, pp. 1–5.

[15] H. D. Scolnik and J. P. Hecht, "Post-Quantum Key Agreement Protocol Based on Non-Square Integer Matrices," Accessed: Jan. 1, 2023. [Online]. Available: https://arxiv.org/abs/2301.01586.

[16] NIST PQC, "Post-Quantum Cryptography," 2017–2024. Accessed: Oct. 25, 2024. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography.

[17] A. Marmebro and K. Stenbom, "Investigation of Post-Quantum Cryptography (FIPS 203 and 204) Compared to Legacy Cryptosystems, and Implementation in Large Corporations," 2024.

[18] W. Tan, Y. Lao, and K. K. Parhi, “Area-Efficient Matrix-Vector Polynomial Multiplication Architecture for ML-KEM Using Interleaving and Folding Transformations,” in *Proc. 2024 IEEE Int. Symp. Circ. Syst. (ISCAS)*, Singapore, May 19–22, 2024, pp. 1–5.

[19] G. Rajendran, P. Ravi, J. P. D’anvers, S. Bhasin, and A. Chattopadhyay, “Pushing the Limits of Generic Side-Channel Attacks on LWE-Based KEM-Parallel PC Oracle Attacks on Kyber KEM and Beyond,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 2, pp. 418–446, 2023.

[20] P. Hecht, “PQC: R-Propping of Public-Key Cryptosystems Using Polynomials over Non-Commutative Algebraic Extension Rings,” Accessed: Oct. 25, 2024. [Online]. Available: https://eprint.iacr.org/2020/1102.

[21] E. Bach and J. Shallit, *Algorithmic Number Theory: Efficient Algorithms*. Cambridge, MA, USA: MIT Press, vol. 1, 1997.

[22] H. Jiang, Z. Zhang, L. Chen, H. Wang, Z. Ma, “IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited,” Accessed: Oct. 25, 2024. [Online]. Available: https://eprint.iacr.org/2017/1096.pdf.

[23] Q. Guo, D. Nabokov, A. Nilsson, and T. Johansson, “SCA-LDPC: A Code-Based Framework for Key-Recovery Side-Channel Attacks on Post-Quantum Encryption Schemes,” in *The International Conference on The Theory and Application of Cryptology and Information Security*. Singapore: Springer Nature Singapore, pp. 203–236, Dec. 2023.

[24] R. Cramer and V. Shoup, “Universal Hash Proofs and a Paradigm for Adaptively Chosen Ciphertext Secure Public Key Encryption,” in *The International Conference on The Theory and Applications of Cryptographic Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 45–64, Apr. 2002.

[25] G. Barthe, B. Grégoire, Y. Lakhnech, and S. Zanella Béguelin, “Beyond Provable Security: Verifiable IND-CCA Security of OAEP,” in *Cryptographers’ Track at the RSA Conference*. Berlin, Heidelberg: Springer, pp. 180–196, Feb. 2011.

[26] H. Jiang, Z. Zhang, L. Chen, H. Wang, Z. Ma, “IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited,” Accessed: Oct. 25, 2024. [Online]. Available: https://eprint.iacr.org/2017/1096.pdf.

[27] J. Liu, H. Zhang, J. Jia, “A Linear Algebraic Attack on a Non-Commuting Cryptography Class Based on the Matrix Power Function,” in *International Conference on Information Security and Cryptology*. Cham, Switzerland: Springer, 2016, pp. 343–354.

[28] V. Roman’kov, “Cryptanalysis of a Combinatorial Public Key Cryptosystem,” *Groups Complex. Cryptol.*, vol. 9, no. 2, pp. 125–135, 2017.

[29] V. Roman’kov, “Nonlinear Decomposition Attack,” *Groups Complex. Cryptol.*, vol. 8, no. 2, pp. 197–207, 2016.

[30] A. Myasnikov and V. V. Roman, “A Linear-Decomposition Attack,” *Groups Complex. Cryptol.*, vol. 7, no. 1, pp. 81–94, 2015.

[31] A. G. Myasnikov, V. Shpilrain, A. Ushakov, “Non-Commutative Cryptography and Complexity of Group-Theoretic Problems,” *Math. Surv. Monogr.*, vol. 177, AMS, 2011.

[32] A. Ben-Zvi, A. Kalka, B. Tsaban, “Cryptanalysis Using Algebraic Spans,” in *Proc. Adv. Cryptol.—CRYPTO 2018: 38th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 19–23, 2018, Part I, pp. 255–274.

[33] J. Di Mauro, E. Salazar, and H. D. Scolnik, “Design and Implementation of a Novel Cryptographically Secure Pseudorandom Number Generator,” *J. Cryptogr. Eng.*, vol. 12, pp. 255–265, 2022. https://doi.org/10.1007/s13389-022-00297-8.