Non-Commutative Key Exchange Protocol

Luis Adrián Lizama-Pérez[1][0000−0001−5109−2927] and J. Mauricio López R.[2]

Sección de Posgrado de la Universidad Politécnica de Pachuca,
Ex-Hacienda de Santa Bárbara, 43830, México
luislizama@upp.edu.mx
Cinvestav Querétaro, Libramiento Norponiente 2000,
Real de Juriquilla, 76230, Santiago de Querétaro, Querétaro, México
jm.lopez@cinvestav.mx

Abstract. We introduce a novel key exchange protocol based on non-commutative matrix multiplication. The security of our method does not rely on computational problems as integer factorization or discrete logarithm whose difficulty is conjectured. We claim that the unique opportunity for the eavesdropper to get the private key is by means of an exhaustive search which is equivalent to searching an unsorted database problem. Therefore, the algorithm becomes a promising candidate to be used in the quantum era to establish shared keys and achieve secret communication. Furthermore, to establish a 256-bit secret key the size of the public key only requires 256 bits while the private key occupies just 384 bits. Matrix multiplications can be done over a reduced 4-bit size modulo. Also, we show that in a generalized method, private numbers become indistinguishable and we discuss how to achieve Perfect Forward Secrecy (PFS). As a consequence, Lizama's protocol becomes a promising alternative for Internet-of-Things (IoT) computational devices in the quantum era.

Keywords: Non-commutative · matrix · cryptography

1 Introduction

In 2017 the National Institute of Standards and Technology (NIST) initiated a process to evaluate the cryptographic algorithms that will be used to support security in the quantum era. Unfortunately, most of the cryptosystems used today will become obsolete in the foreseeable future because they would be broken by quantum computers [1]. Shor's algorithm [2] solves the mathematical problems on which cryptography is supported: integer factorization and discrete logarithm. Although quantum principles have threatened the security of major cryptographic systems, they have raised a new technology known as quantum key distribution (QKD) that allows remote secret key establishment [3, 4, 5, 6].

Post-quantum crypto-systems under evaluation for public-key quantum-resistant [7] include cryptography based on lattices, multi-variate-based, hash-based [8, 9] and code-based [10]. After the third evaluation round, NIST has selected seven algorithms (and eight alternative candidates), four of them are public key encryption (and key-establishment) systems and three correspond to digital signature algorithms. In the first category, CRYSTALS-KYBER, NTRU-HPS, SABER are lattice-based while Classic McEliece is a code-based public key encryption system. Regarding digital signature schemes, CRYSTALS-DILITHIUM and FALCON are lattice-based and Rainbow is a multivariate-based algorithm [11, 12, 13]. According to the criteria defined by NIST, quantum algorithms must be resistant against classical and quantum adversaries, their security level must be comparable to the security of SHA-385 and AES-256. Issues to be considered are the size of the keys and the required computing resources and facility of implementation (in hardware and software). Versatility of the algorithm will be evaluated because of its ability to encrypt messages, perform digital signatures and/or allow key exchange.

As discussed in [14], Lizama's certification method is scalable and interoperable and can be exploited in the pre-quantum and quantum era because the protocol exhibits indistinguishability of the integers used in the public key and ciphertexts. Moreover, public keys size in Lizama's protocol has the smallest size: 0.256 kilobytes and 0.384 kilobytes for public key and certified key, respectively [14].

In this work, we will introduce a new key exchange algorithm based in non-invertible matrix multiplication that can be useful for secret communication in the pre-quantum but also in the quantum era. The article is organized as follows: in Section 2 we discuss some related protocols which include Lizama's non-invertible connectionless and the reduced version of this protocol. In Section 3 we introduce our Non-Commutative Key Exchange Protocol to later introduce in Section 4, a method to certificate the public keys. Finally, Section 5 describe our PFS method that guarantee secrecy of the new session keys.

2 Related protocols

2.1 Shamir-Rivest-Adleman three-pass protocol

The Shamir-Rivest-Adleman protocol allows two remote parties to exchange a secret message without sharing any initial secret as described in [15]. The protocol has the desired commutative property since $m^{ab} \bmod p = m^{ba} \bmod p$, where the exponents a and b satisfy the relation $e^{-1} \cdot e \equiv 1 \bmod p-1$ (see Fig. 1). If $x \equiv y \bmod p-1$, then $a^x \equiv a^y \bmod p$ because according to Fermat's Little Theorem $a^{p-1} = 1$ since p does not divide a. Unfortunately, this protocol is supported on Diffie-Hellman assumption [16] which makes it vulnerable to Shor's algorithm running in a quantum computer [2].

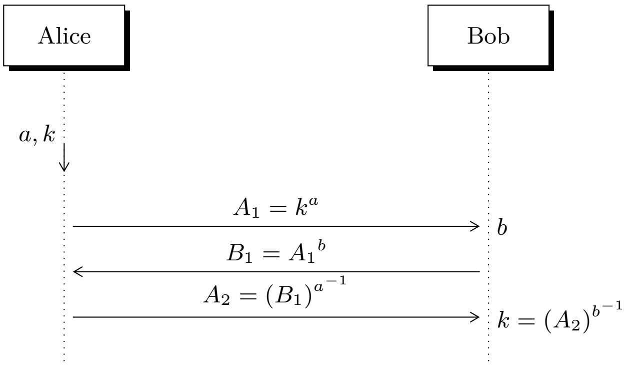

Fig. 1: Shamir-Rivest-Adleman three-pass protocol [15]. A_1, B_1 and A_2 and k are computed modulo p.

2.2 Non commutative cryptography

Stickel's key exchange protocol was motivated by the Diffie-Hellman protocol [17]. In the original formulation, the group used in the protocol was the group of invertible matrices over a finite field [18, 19]. Let G be a public non-abelian finite group. Let $a, b \in G$ be public elements such that $ab \neq ba$. Let the orders of a and b be N and M respectively:

1. Alice chooses two random natural numbers $n < N$, $m < M$ and sends $u = a^n b^m$ to Bob.
2. Bob picks two random natural numbers $r < N$, $s < M$ and sends $v = a^r b^s$ to Alice.
3. Alice derives the key as $K_A = a^n v b^m = a^{n+r} b^{m+s}$.
4. Bob computes $K_B = a^r u b^s = a^{n+r} b^{m+s}$.

Unfortunately a linear algebra attack to this protocol has been published [20, 19]. It is sufficient for the adversary to find matrices x and y such that $xa = ax$, $yb = by$, and $xu = y$, because x corresponds to a^{-n}, while y equals b^m [21].

2.3 Lizama's non-invertible connectionless protocol

Lizama's non-invertible key exchange protocol was introduced in [22]. In this section, we will enhance it to enable the secret key establishment when remote users (Alice and Bob) are not allowed to directly connect each other. We assume that the connection between users is later turned on to enable secret communication. As stated by the protocol, the public key of user i (a for Alice, b for Bob) has two components (P_i, Q_i) where $P_i \equiv p^{2x_i} k_i \bmod n$ and $Q_i \equiv q^{y_i} k_i \bmod n$ where n is the publicly known modulo n computed as $n = p\ q\ r$ so that p, q are small prime integers while r is a large prime integer and the symbol $\equiv$ represents the congruence relation. In the original formulation of the protocol is stated that $x_i + y_i = \phi(n) + 1$ where x_i is chosen randomly and y_i is derived from this relation. However, in this new approach, we will substitute it with $x_i + y_i = \phi(n)$. The private key of user i consists in the pair (x_i, k_i) where k_i is an invertible integer in $\mathbb{Z}_n$. Users upload their public keys (P_i, Q_i) to the cloud service where the public keys are stored. The next steps are described below:

1. Alice and Bob download each other's public key. Then, they perform exponentiation and multiplication mod n, as indicated in Table 1.

Table 1: Operations performed by users over the public keys.

User	Operation	k_{s_i}
Alice	$\left(p^{2x_b} \cdot k_b \bmod n\right)^{x_a} \cdot (q^{y_b} \cdot k_b \bmod n)^{y_a} \equiv$	$p^{2x_b x_a} q^{y_b y_a} \bmod n$
Bob	$\left(p^{2x_a} \cdot k_a \bmod n\right)^{x_b} \cdot (q^{y_a} \cdot k_a \bmod n)^{y_b} \equiv$	$p^{2x_a x_b} q^{y_a y_b} \bmod n$

2. The key that each user derives k_{s_i} has been written at the right hand of Table 1, both are equal because $x_i + y_i = \phi(n)$, thus according to Euler's theorem written in Equation 1 we derive ${k_i}^{x_i+y_i} = {k_i}^{\phi(n)} = 1$ since k_i is an invertible integer in $\mathbb{Z}_n$.

$${k_i}^{\phi(n)} \equiv 1 \bmod n \tag{1}$$

2.4 Reduced Lizama's key exchange

In the original formulation of the protocol, user i sends $k_s k_j \bmod n$ to user j over the public channel, where $k_s = p^{2x_i x_j} q^{y_i y_j} \bmod n$ and k_j is the private key of j who derives k_s multiplying by ${k_j}^{-1}$. The same is valid in the opposite direction from j to i. If $k_s \bmod n$ were an invertible integer in $\mathbb{Z}_n$, user j could compute ${k_s}^{-1}$ but previously she has computed $k_s k_i \bmod n$, thus she could obtain k_i, the private key of the remote user. This is the reason why k_s is chosen to be non-invertible in $\mathbb{Z}_n$.

However, as described in the previous section, Lizama's connectionless protocol does not require the public exchange of $k_s k_i \bmod n$ neither $k_s k_j \bmod n$. On this new basis, let us introduce the reduced Lizama's non-invertible protocol. In this scheme the public key of user i is (P_i, Q_i) where $P_i = a^{2x_i} k_i \bmod n$ and $Q_i = a^{x_i} k_i \bmod n$ where $a = 2$, $n = 2p$ and p is a large prime integer. The private key of user i is (x_i, k_i) where x_i is a random integer and k_i is a random invertible integer in $\mathbb{Z}_n$. The protocol behaves according to the following steps:

1. Using the web service, Alice and Bob obtain a copy of their public keys each other. Then, they perform the operations indicated in Table 2.

Table 2: Operations performed by users over the public keys. The modulo n is computed as $2p$.

User	Operation	k_{s_i}
Alice	$\left(a^{2x_b} \cdot k_b \bmod n\right)^{x_a} \cdot \left(a^{x_b} \cdot k_b \bmod n\right)^{-x_a} \equiv$	$a^{x_a x_b} \bmod n$
Bob	$\left(a^{2x_a} \cdot k_a \bmod n\right)^{x_b} \cdot \left(a^{x_a} \cdot k_a \bmod n\right)^{-x_b} \equiv$	$a^{x_a x_b} \bmod n$

2. k_{s_i} is equal to both sides because ${k_i}^{x_j} {k_i}^{-x_j} \equiv 1 \bmod n$.

Unfortunately, this version of the protocol is insecure in the quantum era because an attacker can interact with Alice applying $x_e = 1$, thus getting $a^{x_a} \bmod n$ and changing the unsorted database problem to the discrete logarithm problem [23, 24].

3 Non-Commutative Key Exchange Protocol

In [22, 14] it has been proposed the following parameters for Lizama's non-invertible key exchange protocol: $\mid p \mid \sim 1024$, $k_i \sim 1024$, $x_i \sim 256$ which gives a public key size $\mid P_i \mid + \mid Q_i \mid \sim 2048$ and a private key size $\mid k_i \mid + \mid x_i \mid = 1280$. Although Lizama's keys size have the smallest when is compared against NIST Round 3 finalists [13] we will demonstrate that the required keys size and the modulo size can be reduced even more in order to operate IoT certificates in the quantum era.

Just to compare the keys size (in bits), let us considerate the reduced Lizama's key exchange. If we take $\mid p \mid$ as 256 then $\mid k_i \mid = 256$ and $\mid a^{2x_i} \mid = 256$ because x_i and k_i are kept secret. The size of the public key (P_i, Q_i) gives $2 \cdot 256 = 512$. The private key achieves $\mid x_i \mid + \mid k_i \mid = 256 + 256 = 512$. Therefore, the secret key k_{s_i} reaches 256 bits. Computation of the secret key requires raising a to an exponent whose size is 256 over a 256-bit modulo.

Now, we proceed to introduce the non-commutative Key Exchange Protocol (nc-KEP) which is based on classical non-commutative matrix algebra. The public key $[P_i]$ of user i is computed as $[P_i] \equiv [k_i] \cdot [u]^{x_i} \cdot [k_i]^{-1} \bmod p$ where matrix multiplication is represented by the symbol $\cdot$, which is performed using a publicly known prime modulo p. Exponentiation can be done since is known that $[P] = [k] \cdot [u] \cdot [k]^{-1} \rightarrow [P]^x = [k] \cdot [u]^x \cdot [k]^{-1}$. Also, $[k]$ and $[u]$ are random invertible square matrices. The exponent x_i is a random integer number, so the private key of a user i is the pair $(x_i, [k_i])$. The protocol behaves according the following steps:

1. Alice and Bob obtain a copy of their public keys from the web service. Then, they perform the operations indicated in Table 3. Exponentiation of $[k_i] \cdot [u]^{x_i} \cdot [k_i]^{-1} \bmod p$ to x_j can be performed applying exponentiation by squaring as illustrated in Equation 2.

$$[k_i] \cdot [u]^{x_i\, x_j} \cdot [k_i]^{-1} \equiv [k_i] \cdot [u]^{x_i} \cdot [k_i]^{-1} \cdot \\ [k_i] \cdot [u]^{x_i} \cdot [k_i]^{-1} \cdot \ldots \mod p \tag{2}$$

Furthermore, to compute the public key $[P_i] \equiv [k_i] \cdot [u]^{x_i} \cdot [k_i]^{-1} \mod p$ is required to raise $[u]$ to a big integer x_i (of at least 128 bits). For this purpose we choose that $[u]$ will be a diagonalizable matrix, therefore $[u] = [g] \cdot [d_u] \cdot [g]^{-1} \mod p$, where $[d_u]$ is the diagonal matrix, such that it holds Equation 3.

$$[u]^{x_i} \equiv [g] \cdot [d_u]^{x_i} \cdot [g]^{-1} \mod p \tag{3}$$

Table 3: Operations performed by users over the public keys.

User	Operation	Result
Alice	$\left([k_b] \cdot [u]^{x_b} \cdot [k_b]^{-1}\right)^{x_a} \equiv$	$[k_b] \cdot [u]^{x_a x_b} \cdot [k_b]^{-1} \mod p$
Bob	$\left([k_a] \cdot [u]^{x_a} \cdot [k_a]^{-1}\right)^{x_b} \equiv$	$[k_a] \cdot [u]^{x_a x_b} \cdot [k_a]^{-1} \mod p$

2. The resulting matrix $[k_i] \cdot [u]^{x_i x_j} \cdot [k_i]^{-1} \mod p$ is sent to the other user who applies the convenient multiplication (left and right hand sides) to get the shared key $[k_s] \equiv [u]^{x_i x_j} \mod p$ as depicted in Figure 2.

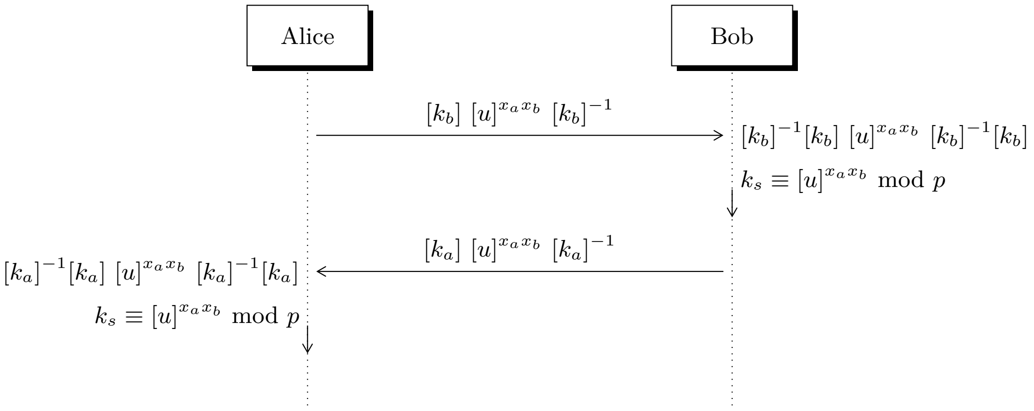
Alice
Bob
$[k_b]\ [u]^{x_a x_b}\ [k_b]^{-1}$
$[k_b]^{-1}[k_b]\ [u]^{x_a x_b}\ [k_b]^{-1}[k_b]$
$k_s \equiv [u]^{x_a x_b} \mod p$
$[k_a]\ [u]^{x_a x_b}\ [k_a]^{-1}$
$[k_a]^{-1}[k_a]\ [u]^{x_a x_b}\ [k_a]^{-1}[k_a]$
$k_s \equiv [u]^{x_a x_b} \mod p$

Fig. 2: Lizama's non-commutative Key Exchange Protocol (nc-KEP). The shared secret key between remote users is $k_s \equiv [u]^{x_a x_b} \mod p$. The operations denote matrix multiplication.

Cryptosystem. Encryption can be easily achieved as done by the Hill cipher system because the shared secret key $[k_s] \equiv [u]^{x_a x_b} \mod p$ can be properly inverted to decrypt a block message of size equal to $[k_s]$, as written in Equation 4 where symbol $\cdot$ denotes matrix multiplication.

Since not every possible matrix is an invertible matrix, users must restart the protocol when they derive a non-invertible matrix. It is known that the Hill cipher is vulnerable to a known-plaintext attack, so we will demonstrate in Section 5 how to safely generate a new secret key from the current one.

$$\begin{aligned}[c] &\equiv [m]\cdot[k_s] \bmod p\\ [m] &\equiv [c]\cdot[k_s]^{-1} \bmod p\end{aligned} \tag{4}$$

3.1 Security Analysis

The public key of user i is computed multiplying $[k_i]$, $[u]^{x_i}$ and $[k_i]^{-1}$ which are square matrices of size $t \times t$, that is $[u]^{x_i}{}_{t\times t}$ and $[k_i]_{t\times t}$. As a result, $\mid [u]^{x_i} \mid = \mid [k_i] \mid = t^2 \mid p \mid$ and we choose $\mid [k_i] \mid = 256$ to be resistant in the quantum era. The size of each matrix's integer is equal to $\mid p \mid$ which is the size of the modulo p and the size $\mid p \mid$ is obtained from $\frac{\mid [k_i] \mid}{t^2}$. The secret shared key is $[u]^{x_i x_j}$ mod p. From here, we deduced that the security level is $\mid x_i \mid + \mid x_j \mid$, thus for the quantum era we choose $\mid x_i \mid = \mid x_j \mid = 128$.

For example, if we want a security level of 256 bits and we choose $t = 4$, then $\mid p \mid = 16$ because $\frac{256}{16}$. Also, $\mid [k_i] \mid = \mid [a_i]^{x_i} \mid = 256$ because $\mid [k_i] \mid = 16 \mid p \mid$ where each matrix's element takes 16 bits. The size of the public key is $\mid [P_i] \mid = 256$ and the private key occupies $\mid x_i \mid + \mid [k_i] \mid = 128 + 256 = 384$ bits. In this example, the computation of the key requires x_i (or x_j) matrix multiplications over a 16-bit size modulo. The matrix $[a_i]$ and the prime integer p are publicly shared: $[a_i]$ is initialized with t^2 random integers in $\mathbb{Z}_p$. Other parameter cases are written in Table 6.

Table 4: It is shown some parameter sizes when is chosen $\mid [k_s] \mid = 256$ as the security level for $t = 2, 4, 8$. Sizes are written in bits.

Parameter	2×2	4×4	8×8
$\mid [k_i]\cdot[u]^{x_i}\cdot[k_i]^{-1} \bmod p \mid$	256	256	256
$\mid x_i \mid + \mid [k_i] \mid$	384	384	384
$\mid p \mid$	64	16	4

We claim that our algorithm is post-quantum because the unique opportunity for the eavesdropper, in order to get the private key $(x_i, [k_i])$, is mounting an exhaustive search among those elements, which is equivalent to searching an unsorted database problem.

3.2 Generalized non-commutative KEP

Suppose a user acts as a malicious Eve, so after she establishes a key with the target user, say Alice, she obtains $[u]^{x_a x_e}$ mod p. However, Eve can compute $[u]^{x_e} \equiv [e]$ mod p then she would try to obtain x_a from $[e]^{x_a}$ mod p. If Eve gets x_a she derives $[a]^{x_a} \equiv [M]$ mod p, indeed she has:

— From the channel: $[x]\cdot[M]^{x_e}\cdot[x]^{-1} \equiv [c]$ mod p.
— From Alice's public key: $[x]\cdot[M]\cdot[x]^{-1} \equiv [s]$ mod p

Since $[c]$, $[s]$, $[M]$ and $[M]^{x_e}$ are known matrices, the opponent can solve the system for $[x]$ which is the Alice's private key thus impersonating her. We would suggest that the size of x_a must be increased to 256 bits, but this attack has changed the unsorted database problem to a hardest version of the discrete logarithm problem based on matrices [23, 24]. To avoid this attack we will introduce a generalized non-commutative KEP. Here, each user i has two public matrices $([P_i], [Q_i])$ as they are shown in Table 5. Thus, the secret key between i and j is deduced to be $[u]^{x_i x_j} \cdot [w]^{y_i y_j} \bmod p$ scaling the complexity problem to the generalized case $[e_1]^{x_a} \cdot [e_2]^{y_a} \bmod p$.

Indistinguishably of the secret key. Now, we want to demonstrate that the pair (x_a, y_a) is indistinguishable from other pairs, symbolically $[e_1]^{x_{a_1}} \cdot [e_2]^{y_{a_1}} \equiv [e_1]^{x_{a_2}} \cdot [e_2]^{y_{a_2}} \bmod p$, then for $t = 1, 2$:

— $[e_1]^{x_{a_t}} \equiv [u]^{x_{e_t} x_{a_t}} \bmod p$
— $[e_2]^{y_{a_t}} \equiv [w]^{y_{e_t} y_{a_t}} \bmod p$

Let us rewrite the last equation as $[u]^{x_1} \cdot [w]^{y_1} \equiv [u]^{x_2} \cdot [w]^{y_2} \bmod p$ where $x_t = x_{e_t} x_{a_t}$ and $y_t = y_{e_t} y_{a_t}$ for $t = 1, 2$. In order to be indistinguishable, we must establish $(x_1, y_1) \neq (x_2, y_2)$. But $[u]$ and $[w]$ are diagonalizable matrices, thus we can separate each equation's term into factors. If we take the first term of the left hand side, we have $[u]^{x_1} = [s_1] \cdot [s_2] \bmod p$ then:

— $[s_1] \equiv [g] \cdot [d_u]^{x_1 - \lambda} \cdot [g]^{-1} \bmod p$
— $[s_2] \equiv [g] \cdot [d_u]^{\lambda} \cdot [g]^{-1} \bmod p$

Because $x_1 - \lambda + \lambda = x_1$ for $\lambda = 0 \ldots x_1$. Provided $|x_1| = 256$, we can separate into several factors each equation's term. By separating them, we directly find (x_1, y_1) and (x_2, y_2) and private numbers (x_i, y_i) become indistinguishable.

Indistinguishably of the private key. A public key $([P_{a_1}], [Q_{a_1}])$ is computed using the private key $(x_{a_1}, y_{a_1}, [k_{a_1}])$. Indeed $(x_{a_1}, [k_{a_1}])$ produces $[P_{a_1}]$ while $(y_{a_1}, [k_{a_1}])$ generates $[Q_{a_1}]$. Suppose we have found another pair $(x_{a_2}, [k_{a_2}])$ that also generates $[P_{a_1}]$. We would like to show that y_{a_2} exists such that $(y_{a_2}, [k_{a_1}])$ produces $[Q_{a_1}]$. In other words, the private key become indistinguishable from the opponent's point of view.

Assume the public key computed as $[P_{a_1}] \equiv [k_{a_1}] \cdot [g] \cdot [d_u]^{x_{a_1}} \cdot [g]^{-1} \cdot [k_{a_1}]^{-1} \bmod p$, $[Q_{a_1}] \equiv [k_{a_1}] \cdot [h] \cdot [d_w]^{y_{a_1}} \cdot [h]^{-1} \cdot [k_{a_1}]^{-1} \bmod p$. But provided $(x_{a_2}, [k_{a_2}])$ produces $[P_{a_1}]$ we must find y_{a_2} such that $[P_{a_1}] \equiv [k_{a_2}] \cdot [g] \cdot [d_u]^{x_{a_2}} \cdot [g]^{-1} \cdot [k_{a_2}]^{-1} \bmod p$ and $[Q_{a_1}] \equiv [k_{a_2}] \cdot [h] \cdot [d_w]^{y_{a_2}} \cdot [h]^{-1} \cdot [k_{a_2}]^{-1} \bmod p$. Our strategy consists in constructing $[d_w]$ such that each exponentiation produces a different matrix. Therefore, we will eventually arrive to the exponent that produces $[Q_{a_1}]$. So, we require that $[h] \cdot [d_w]^i \cdot [h]^{-1} \neq [h] \cdot [d_w]^j \cdot [h]^{-1} \ldots$ for $i \neq j$. Removing $[h]$ both sides we have $[d_w]^i \neq [d_w]^j \ldots$ which implies that each diagonal element satisfies the condition ${d_{w_i}}^i \neq {d_{w_i}}^j \bmod p \ldots$ for $i = 1 \ldots m$ where m is the matrix dimension. To surpass such requirement each diagonal element is computed as 2^{μ_i} but it meets $2^{\mu_i} < p$ for $i = 1 \ldots m$. Thus, each diagonal element of $[d_w]$ is a power of the primitive root inside $\mathbb{Z}_p$.

In this scenario, the size of the public key yields $\mid (P_i, Q_i) \mid = 512$, the private key $\mid (x_i, y_i, k_i) \mid = 512$ and the secret key raises its security level from 256 to 512 bits. As it can be concluded from this discussion, the generalized nc-KEP can be directly upgraded from its previous particular case. In the next sections we will use the non-generalized nc-KEP, so that a better explanation could be provided.

Table 5: The public keys in the generalized nc-KEP. The secret key between users will be $k_s = [u]^{x_a x_b} \cdot [w]^{y_a y_b} \bmod p$

User	P_i	Q_i
Alice	$[k_a] \cdot [u]^{x_a} \cdot [k_a]^{-1} \bmod p$	$[k_a] \cdot [w]^{y_a} \cdot [k_a]^{-1} \bmod p$
Bob	$[k_b] \cdot [u]^{x_b} \cdot [k_b]^{-1} \bmod p$	$[k_b] \cdot [w]^{y_b} \cdot [k_b]^{-1} \bmod p$

4 Certificated Keys

An indispensable property of public keys is to be authenticated by a Certification Authority (CA). The keys of the non-commutative Key Exchange Protocol (nc-KEP) can be certified if a CA raises the keys to her private key number x_{ca} as indicated in Table 6. Alice and Bob obtain a copy of their public certified keys from the web service. Then, they perform the usual exponentiation $\left([u]^{x_i x_{ca}}\right)^{x_j} \bmod p$. The secret shared key is derived as $[u]^{x_i x_{ca} x_j} \bmod p$.

Table 6: CA's public database. CA performs exponentiation over the public keys. The secret shared key is $k_s = [u]^{x_a x_{ca} x_b} \bmod p$

User	Public key	Certified key
CA	$[k_{ca}] \cdot [u]^{x_{ca}} \cdot [k_{ca}]^{-1} \bmod p$	-
Alice	$[k_a] \cdot [u]^{x_a} \cdot [k_a]^{-1} \bmod p$	$[k_a] \cdot [u]^{x_a x_{ca}} \cdot [k_a]^{-1} \bmod p$
Bob	$[k_b] \cdot [u]^{x_b} \cdot [k_b]^{-1} \bmod p$	$[k_b] \cdot [u]^{x_b x_{ca}} \cdot [k_b]^{-1} \bmod p$

4.1 Interdomain certificates

Users that have been certified with different Certification Authorities, say CA_1 and CA_2 can establish a secret key, if each CA certifies their keys with the converse CA. It means that after the second certification, the public key of users can be written as $[k_i] \cdot [u]^{x_i x_{ca_1} x_{ca_2}} [k_i]^{-1} \bmod p$ and $[k_j] \cdot [u]^{x_j x_{ca_1} x_{ca_2}} [k_j]^{-1} \bmod p$ of user i and j, respectively. The shared secret key between users will be $[u]^{x_i x_{ca_1} x_{ca_2} x_j} \bmod p$.

5 Perfect Forward Secrecy (PFS)

Remote users Alice and Bob may want to establish a new secret key $[k_t]$ based on the they already have $[k_s]$. However, if for some reason $[k_t]$ is compromised by an opponent, Perfect Forward Secrecy (PFS) is a property of key agreement protocols that guarantee that such leakage does not compromise the security of previously used keys. In Figure. 3 we depict our PFS protocol to produce new session secret keys. Provided the private keys $[k_a]$ and $[k_b]$ remain secret, the eavesdropper could capture $[k_t]$ but she does not know $t_a t_b$ thus she cannot derive $[k_s]$.

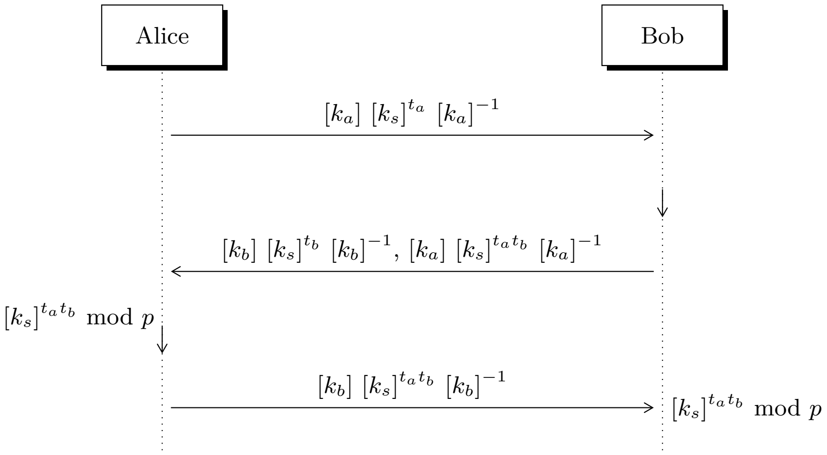
Alice
Bob
$[k_a]\ [k_s]^{t_a}\ [k_a]^{-1}$
$[k_b]\ [k_s]^{t_b}\ [k_b]^{-1},\ [k_a]\ [k_s]^{t_a t_b}\ [k_a]^{-1}$
$[k_s]^{t_a t_b} \bmod p$
$[k_b]\ [k_s]^{t_a t_b}\ [k_b]^{-1}$
$[k_s]^{t_a t_b} \bmod p$

Fig. 3: Perfect Forward Secrecy (PFS) in Lizama's non-commutative Key Exchange Protocol (nc-KEP). The new shared secret key between remote users $[k_t]$ is $[k_s]^{t_a t_b} \bmod p$. The operations denote matrix multiplication.

6 Conclusions

We introduced the non-commutative key exchange protocol (nc-KEP) which allows the secret key establishment between two remote parties to perform private communication. Lizama's nc-KEP does not rely on computational problems as integer factorization or discrete logarithm such that the security can be properly evaluated using ordinary rules of matrix multiplication without complex theoretical background. We have analyzed that in order to establish a 256-bit secret key the size of the public key takes 256 bits while the private key does require 384 bits. Remarkably, matrix multiplications can be done over a reduced 4-bit size modulo.

In addition, we have presented a generalized version of the protocol that guarantees that the eavesdropper must face the unsorted database problem because private numbers become indistinguishable. This enhanced version just requires that public and private keys be increased to 512 bits with same security level. Moreover, a method to achieve Perfect Forward Secrecy has been demonstrated.

Therefore, Lizama's nc-KEP enables secret communication between restricted computational IoT devices in the quantum era. The algorithm would be further optimized in hardware/software since it only requires matrix-multiplication.

References

[1] M. A. Barreno, "The future of cryptography under quantum computers," *Dartmouth College Computer Science Technical Reports*, 2002.

[2] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science.* Ieee, 1994, pp. 124–134.

[3] H. Bennett Ch and G. Brassard, "Quantum cryptography: public key distribution and coin tossing int," in *Conf. on Computers, Systems and Signal Processing (Bangalore, India, Dec. 1984)*, 1984, pp. 175–9.

[4] L. A. Lizama-Pérez, J. M. López, and E. D. C. López, "Quantum flows for secret key distribution," *Advanced Technologies of Quantum Key Distribution*, p. 37, 2018. [Online]. Available: https://dx.doi.org/10.5772/intechopen.75964

[5] L. A. Lizama-Perez and J. M. López, "Quantum key distillation using binary frames," *Symmetry*, vol. 12, no. 6, p. 1053, Jun 2020. [Online]. Available: http://dx.doi.org/10.3390/sym12061053

[6] L. A. Lizama-Pérez and J. M. López R., "Beyond the limits of shannon's information in quantum key distribution," *Entropy*, vol. 23, no. 229, 2021. [Online]. Available: https://doi.org/10.3390/e23020229

[7] C. S. R. CENTER, "Post-Quantum Cryptography Standardization Conference," 2021, [Online; accessed March 30, 2021].

[8] L. A. Lizama-Perez, "Digital signatures over hash-entangled chains," *SN Applied Sciences*, vol. 1, no. 12, p. 1568, 2019. [Online]. Available: https://doi.org/10.1007/s42452-019-1618-6

[9] L. A. Lizama-Pérez, L. J. Montiel-Arrieta, F. S. Hernández-Mendoza, L. A. Lizama-Servín, and S.-A. Eric, "Public hash signature for mobile network devices," *Ingeniería, Investigación y Tecnología*, vol. XX, no. 2, pp. 1–10, 2019. [Online]. Available: http://dx.doi.org/10.22201/fi.25940732e.2019.20n2.018

[10] D. J. Bernstein, "Introduction to post-quantum cryptography," in *Post-quantum cryptography.* Springer, 2009, pp. 1–14.

[11] I. T. Laboratory, "PQC Standardization Process: Third Round Candidate Announcement," https://csrc.nist.gov/news/2020/pqc-third-round-candidate-announcement, 2020, [Online; accessed March 30, 2021].

[12] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Report on post-quantum cryptography.* US Department of Commerce, National Institute of Standards and Technology, 2016, vol. 12.

[13] E. Persichetti, "NIST Round 3 finalists," https://pqc-wiki.fau.edu/w/Special:DatabaseHome, 2020, [Online; accessed March 30, 2021].

[14] L. A. Lizama-Pérez and J. M. López R., "Non-invertible public key certificates," *Entropy*, vol. 23, no. 2, 2021. [Online]. Available: https://doi.org/10.3390/e23020226

[15] J. Clark and J. Jacob, "A survey of authentication protocol literature, 1997. version 1.0," *Unpublished Report, University of York, http://cs. york. ac. uk/˜ jac/papers/drareview. ps. gz.*

[16] B. Den Boer, "Diffie-hellman is as strong as discrete log for certain primes," in *Conference on the Theory and Application of Cryptography.* Springer, 1988, pp. 530–539.

[17] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[18] E. Stickel, "A new method for exchanging secret keys," in *Third International Conference on Information Technology and Applications (ICITA'05)*, vol. 2. IEEE, 2005, pp. 426–430.

[19] A. Myasnikov, V. Shpilrain, and A. Ushakov, *Group-based cryptography.* Springer Science & Business Media, 2008.

[20] V. Shpilrain, “Cryptanalysis of stickel’s key exchange scheme,” in *International Computer Science Symposium in Russia.* Springer, 2008, pp. 283–288.

[21] D. Grigoriev and V. Shpilrain, “Tropical cryptography,” *Communications in Algebra*, vol. 42, no. 6, pp. 2624–2632, 2014.

[22] L. A. Lizama-Perez, “Non-invertible key exchange protocol,” *SN Applied Sciences*, vol. 2, p. 1083, 2020. [Online]. Available: https://link.springer.com/content/pdf/10.1007

[23] C. Zalka, “Grover’s quantum searching algorithm is optimal,” *Physical Review A*, vol. 60, no. 4, p. 2746, 1999.

[24] D. Kahrobaei, C. Koupparis, and V. Shpilrain, “Public key exchange using matrices over group rings,” *arXiv preprint arXiv:1302.1625*, 2013.