Non-invertible Public Key Certificates

Luis Adrian Lizama-Perez[1][0000−0001−5109−2927] and J. Mauricio López[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. Post-quantum public cryptosystems introduced so far do not define an scalable public key infrastructure for the quantum era. We demonstrate here a public certification system based in Lizama's non-invertible Key Exchange Protocol which can be used to implement a public key infrastructure (PKI), secure, scalable, interoperable and efficient. We show functionality of certificates across different certification domains. Finally, we discuss that non-invertible certificates can exhibit Perfect Forward Secrecy (PFS).

Keywords: Non-invertible · cryptography · certificate · PKI

1 Introduction

Since its origin at the late seventies, public key cryptography (PKC) has been exploited to support user authentication and digital signatures over the internet. In PKC, each user has two keys, the public P_u and the private key P_r, which are mutually inverses in some mathematical sense. Not taking into account formal details we would write that $P_r = {P_u}^{-1}$, thus to achieve confidentiality, a message m is encrypted using Bob's public key, symbolically we write $[m]_{P_u}$, then it is decrypted with the private key so $m = [m]_{P_u {P_u}^{-1}}$. In contrast, to guarantee message authentication m is encrypted with Alice's private key and decrypted with her public key. Symbolically we can write it as $m = [m]_{P_r {P_r}^{-1}}$.

Unfortunately, Shor's algorithm [1] solves over an hypothetical quantum computer, the mathematical problems on which PKC is supported: integer factorization and discrete logarithm. As fact, most of the public key cryptosystems used today will become obsolete in the foreseeable future because they would be broken by quantum computers [2]. For this reason, the National Institute of Standards and Technology (NIST) initiated in 2015 a process to evaluate cryptographic algorithms to choose the appropriate methods for the quantum era. To this date, the selection process is in the third evaluation round [3, 4].

In this work, we will enhance Lizama's non-invertible key exchange method [5] to be completely functional for PKC in the quantum era including Certification Authorities (CA) that issue digital certificates bounded to user's public keys. In addition, we will introduce a new method to achieve Perfect Forward Secrecy (PFS).

2 Cryptography in the Quantum Era

Cryptography in the quantum era can be divided into two main approaches: quantum and post-quantum cryptography. A formal discussion of such approaches is beyond the scope of the present article. Let us simply mention that quantum cryptography relies in quantum physics principles that allow to establish a secret key between two authenticated remote parties [6]. The eavesdropper cannot control quantum communication because she produces a detectable noise. Recent works has been done to resist quantum attacks [7, 8, 9].

On the other side, post-quantum cryptography encompasses cryptographic methods conceived to resist computational capacity of quantum computers [10, 4]. Several methods have been formulated based on computational problems whose complexity surpass the theoretical capacities of quantum computers. Not wishing to fully cover all cases, most promissory techniques include lattices, supersingular isogeny, multivariate equations, code and hash based cryptography.

Lattice based methods have demonstrated good performance, generate short ciphertext and keys and short signatures [11, 12]. Similar to Diffie-Hellman key exchange is the Supersingular Isogeny Diffie-Hellman (SIDH) method which is a quantum resistant key exchange algorithm [13, 14]. Supersingular Elliptic Curve Isogeny Cryptography (SIDH) produces very small key sizes but it shows slower performance. The representative algorithm is the Supersingular Isogeny Key Encapsulation (SIKE). The basic objects of multivariate cryptography are systems of nonlinear (usually quadratic) polynomial equations in several variables over a finite field. When performing a digital signature, the set of equations constitute the public key. The receiver computes the hash to verify that the output of the equations corresponds to the hash of the message that is signed [15].

A code-based cryptosystem is essentially a form of error correction code. The private key is a code C, which allows to correct t errors. The sender will encode the message with the public key and include t errors during encoding, then the ciphertext is obtained by adding an error vector to each codeword. The receiver with code C will be able to accurately correcting the errors so decoding the message. Hash based cryptography was introduced by Lamport, later it was enhanced using Merkle trees [16] and Lizama's hash based methods [17, 18].

3 Digital Certificates

A cryptographic certificate is basically, a verified public key signed by a third trusted party called Certification Authority (CA). By using this method each user can verify the origin of a request before accepting it. The importance of a certified key can be illustrated showing a Man In The Middle (MITM) attack over the Diffie-Hellman (DH) protocol, the first public key exchange algorithm [19]. In Fig. 1 we represent the steps required for this key exchange algorithm where the integer prime p and g are publicly known. A description in depth can be found in Appendix A.

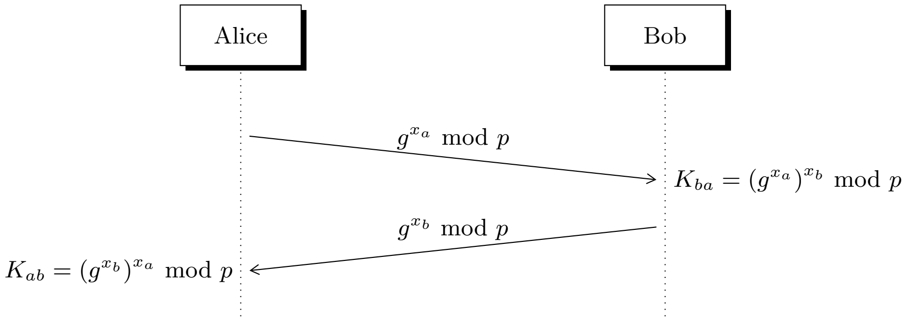
Alice
Bob
$g^{x_a} \mod p$
$K_{ba} = (g^{x_a})^{x_b} \mod p$
$g^{x_b} \mod p$
$K_{ab} = (g^{x_b})^{x_a} \mod p$

Fig. 1: Basic Diffie-Hellman protocol. All operations are performed module p.

Since there is no method to verify the origin of the integer numbers exchanged across the public channel, an eavesdropper can implement a man in the middle (MITM) attack over the Diffie-Hellman method as it is observed in Fig. 2.

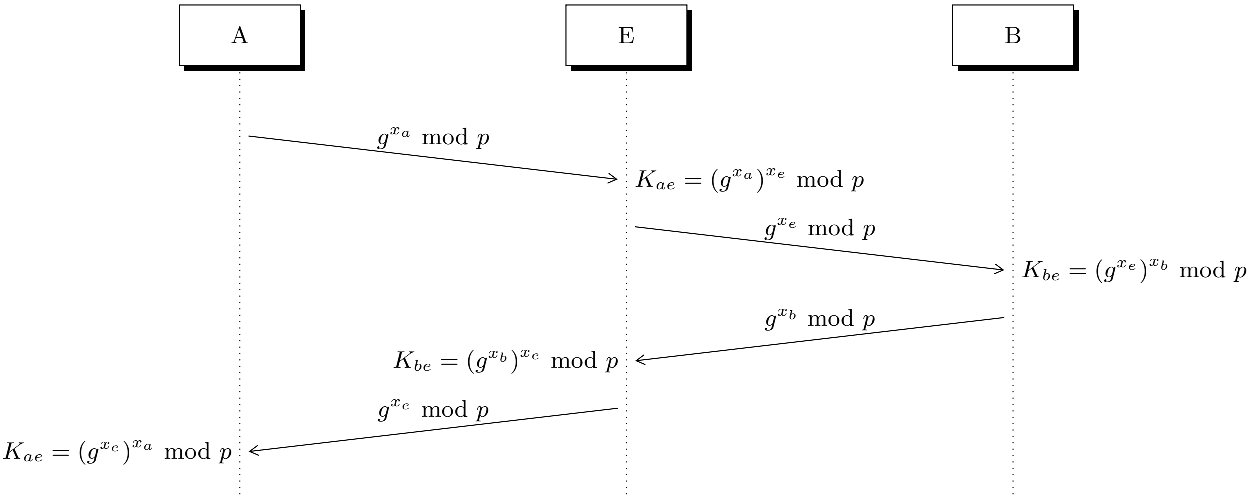
A
E
B
$g^{x_a} \bmod p$
$K_{ae} = (g^{x_a})^{x_e} \bmod p$
$g^{x_e} \bmod p$
$K_{be} = (g^{x_e})^{x_b} \bmod p$
$g^{x_b} \bmod p$
$K_{be} = (g^{x_b})^{x_e} \bmod p$
$g^{x_e} \bmod p$
$K_{ae} = (g^{x_e})^{x_a} \bmod p$

Fig. 2: The MITM attack over DH protocol. The eavesdropper obtains a key with Alice K_{ae} and other with Bob K_{be}. Legitimate users cannot verify the origin of exchanged numbers.

To avoid a MITM attack over DH protocol, it can be added the RSA algorithm to the exchange protocol. RSA is described in Appendix A. Another common method to protect DH key exchange algorithm is elliptic-curve cryptography [20, 21], however Lizama's protocol is closely related to RSA, thus we describe here RSA and DH.

Fig. 3 shows that Alice encrypts the DF constructor $g^{x_a} \bmod p$ with Bob's public key written as (e_b, n_b), so that only Bob can decrypt it using his private key d_b. Alice verifies the received message because it is attached a hash of the secret key computed by Bob as represented in Fig. 3.

In order Alice verifies Bob's public key preventing it does not come from an illegitimate user, Bob must register first his public key with the certification authority abbreviated as CA (a third trusted party). Generally speaking, Bob's obtains a certificate of his public key C_B after CA encrypts (signing) Bob's public key with CA's private key $P_{R_{CA}}$. In the next relations, encryption (or decryption) process is denoted as square brackets while the encryption (or decryption) key is outside the brackets:

$$C_B = [P_{U_B}]_{P_{R_{CA}}}$$

Every user can obtain and verify Bob's public key decrypting C_B with CA's public key $P_{R_{CA}}$:

$$P_{U_B} = [C_B]_{P_{U_{CA}}}$$

4 Certification Authority (CA)

As mentioned earlier, a certification authority (CA) is a trusted third party that signs a user public key using CA's private key therefore binding the subject's identity (and associated information

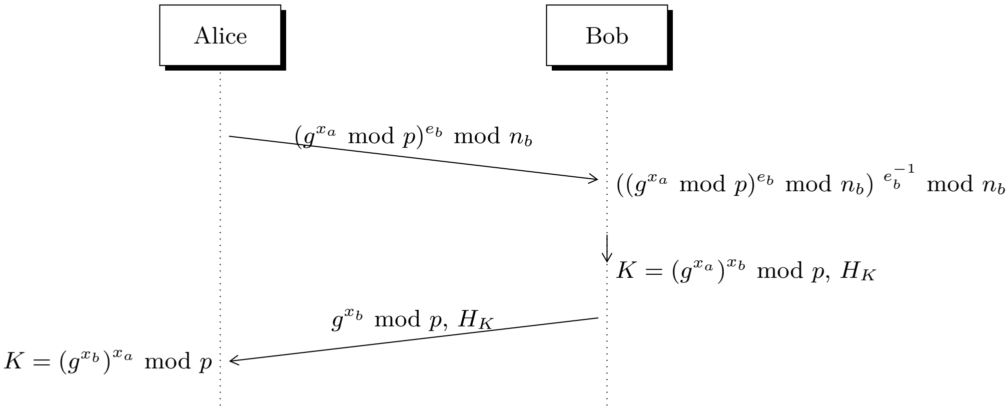
Alice
Bob
$(g^{x_a} \bmod p)^{e_b} \bmod n_b$
$((g^{x_a} \bmod p)^{e_b} \bmod n_b)^{e_b^{-1}} \bmod n_b$
$K = (g^{x_a})^{x_b} \bmod p,\ H_K$
$g^{x_b} \bmod p,\ H_K$
$K = (g^{x_b})^{x_a} \bmod p$

Fig. 3: Diffie-Hellman algorithm with RSA. Bob's public key is written as $P_{U_B} = (e_b, n_b)$, Bob's private key is ${e_b}^{-1}$ that indicates the inverse of e_b in $\mathbb{Z}_{\phi(n)}$. H_K represents the hash value of K which is used by Alice to verify the origin of the received number.

including the name of the owner) to the user's public key inside a cryptographic certificate. Cryptographic certificates can be exploited to achieve digital signatures in a wide broad of internet transactions and PKI: certificates (X.509), secure channels (TLS) and email (S/MIME).

In view of the imminent arrival of quantum computers is unpostponable to develop strategies in order to adapt the Public Key Infrastructure (PKI) for transition to the quantum era [3, 4]. Up to now, some few works have been published that adapt existing certificates to quantum certificates or hybrid certificates, which includes two public keys for the subject, one traditional and one post-quantum algorithm and two CA signatures [22, 23]. Other works have evaluated existing mechanisms to deal with large records like record fragmentation, segmentation, caching, and compression [24]. One the main challenges reported is the difficulty to manage larger certificates by some cryptographic software libraries.

ITU-T Recommendation X.509 defines the format of public key certificates as well as the provision of authentication services under a centralized control scheme that is represented by a directory [25, 26]. X.509 assumes a hierarchical system of certificate authorities (CAs) for issuing certificates. This contrasts with web of trust models, like PGP, where users sign others' key certificates to establish the authenticity of the binding between a public key and its owner [27].

A PKI is arranged hierarchically, so that there is always a direct path (a certificate chain) from the Root CA to every end-entity. Therefore, with many users, it may be more practical to have a series of CAs, each of which securely provides its public key to a fraction of the users.

If Alice has a certificate from CA$_1$ and Bob owns a certificate from CA$_2$ but Alice does not securely know the public key of CA$_2$, then Bobs's public certificate emitted by CA$_2$, cannot be used by Alice. However, if the two CAs have securely exchanged their own public keys, the following procedure will enable Alice to obtain Bob's public key:

1. Alice obtains the certificate of CA$_2$ signed by CA$_1$. Since Alice has the public key of CA$_1$, she can get the public key of CA$_2$ from its certificate and verify it using the signature of CA$_1$ on the certificate.
2. From the directory Alice obtains the certificate of Bob signed by CA$_2$. Since Alice now has the public key of CA$_2$, she can verify the signature, therefore getting Bob's public key.

5 Lizama's Key Exchange Protocol

Lizama's key exchange protocol was introduced in [5], there it can be found all details about the method and its security. The protocol is illustrated in Fig. 4. The public key of user i (a for Alice, b for Bob) has two components (P_i, Q_i) where $P_i = p^{2x_i}k_i \bmod n$ and $Q_i = q^{y_i}k_i \bmod n$. The value x_i is chosen randomly while $y_i = \phi(n)+1$. The module n is the product of tree public integer primes, so that $n = p \cdot q \cdot r$ where p and q are small integer primes and r is a big integer prime. To achieve indistinguishability p and q are suggested to be 2, since 2 is a primitive root module r (see [5]). The exponent is chosen to be $2x_i$ instead of x_i to avoid a multiplication attack. The x_i value constitutes along k_i the private key of user i where k_i is an invertible integer in the ring. Users share their public keys (P_a, Q_a) and (P_b, Q_b) as well as the integer module n. The steps of the protocols are summarized as follows:

1. Once public keys have been exchanged, Alice and Bob perform two operations over the numbers received: exponentiation and multiplication as indicated in Tab. 1.

Table 1: These operations (exponentiation and multiplication) are performed at each side after public keys of users are exchanged.

User	Operation	Result
Alice	$\left(p^{2x_b} \cdot k_b \bmod n\right)^{x_a} \cdot (q^{y_b} \cdot k_b \bmod n)^{y_a}$	$p^{2x_bx_a}q^{y_by_a} \cdot k_b \bmod n$
Bob	$\left(p^{2x_a} \cdot k_a \bmod n\right)^{x_b} \cdot (q^{y_a} \cdot k_a \bmod n)^{y_b}$	$p^{2x_ax_b}q^{y_ay_b} \cdot k_a \bmod n$

2. To derive the right hand results of Tab. 1 is applied Euler's theorem for $\mathbb{Z}_n$. The theorem is written in Eq.1 where r is an integer safe prime. Because $n = pqr$ we have that $\phi(n) = (p-1)(q-1)(r-1)$. Here, k and n are relative prime each other, so k is an invertible integer in $\mathbb{Z}_n$. The exponent x_i constitutes the private key, is chosen randomly, but x_i and y_i sum up $\phi(n)+1$, thus according to Eq.1 we have $k^{\phi(n)+1} = k^{\phi(n)} \cdot k^1 = k$ because k is an invertible integer in $\mathbb{Z}_n$.

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

3. Users exchange the resulting value $p^{2x_ax_b}q^{y_ay_b}k_i \bmod n$, which is multiplied by the corresponding inverse ${k_i}^{-1}$ at each side to derive the secret shared key $p^{2x_ax_b}q^{y_ay_b} \bmod n$ as depicted in Fig. 4.

As an example of the required bits for keys, if $|r| = 1024$ (the symbol $|\ |$ denotes the number of bits) the length of the private key is 1536 bits ($|x_i| = 512$ and $|k_i| = 1024$) while the public key (P_i, Q_i) contains 2056 bits [5]. In this example, the security level of the secret key is 1024.

5.1 Cipher-system

In Fig. 4, the secret shared key k_s is a non-invertible number in $\mathbb{Z}_n$, thus a convenient method to achieve a cipher-system and secret communication is to divide $k_s = p^{2x_ax_b}q^{y_ay_b} \bmod n$ by pq, so if we choose $p = q = 2$, then $k_r = p^{2x_ax_b-2}2^{(2r-1-x_a)(2r-1-x_b)} \bmod r$. Now, Alice and Bob can compute its multiplicative inverse ${k_r}^{-1}$. The enciphered message is obtained as $c = m \cdot k_r \bmod r$

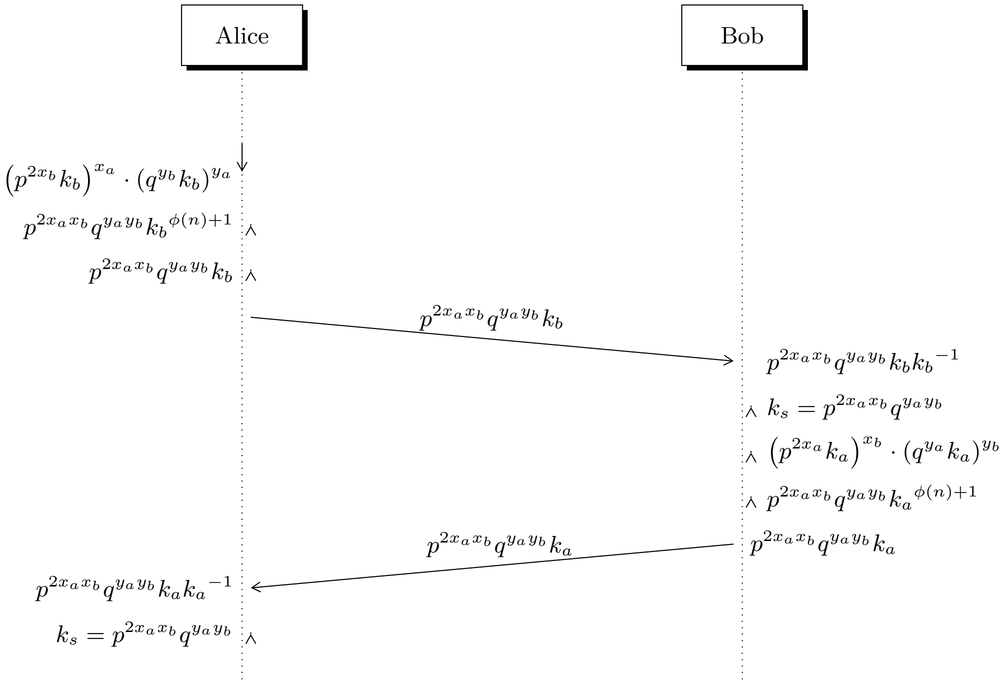
Alice
Bob
$\left(p^{2x_b}k_b\right)^{x_a} \cdot (q^{y_b}k_b)^{y_a}$
$p^{2x_ax_b}q^{y_ay_b}{k_b}^{\phi(n)+1}$
$p^{2x_ax_b}q^{y_ay_b}k_b$
$p^{2x_ax_b}q^{y_ay_b}k_b$
$p^{2x_ax_b}q^{y_ay_b}k_b{k_b}^{-1}$
$k_s = p^{2x_ax_b}q^{y_ay_b}$
$\left(p^{2x_a}k_a\right)^{x_b} \cdot (q^{y_a}k_a)^{y_b}$
$p^{2x_ax_b}q^{y_ay_b}{k_a}^{\phi(n)+1}$
$p^{2x_ax_b}q^{y_ay_b}k_a$
$p^{2x_ax_b}q^{y_ay_b}k_a$
$p^{2x_ax_b}q^{y_ay_b}k_a{k_a}^{-1}$
$k_s = p^{2x_ax_b}q^{y_ay_b}$

Fig. 4: Lizama's non-invertible KEP [5]. All operations are modulo n where $n = pqr$. According to Euler's theorem $k^{\phi(n)+1} \bmod n = k$ because k is an invertible integer in $\mathbb{Z}_n$.

and the original plaintext is recovered through the relation $m = c \cdot {k_r}^{-1} \bmod r$ because $m = m \cdot k_r{k_r}^{-1} \bmod r$. To send a message encoded as an integer in $\mathbb{Z}_r$, the number m must be less than r.

Table 2: When $p = q = 2$, we derived $k_r = 2^{2x_ax_b-2}2^{(2r-1-x_a)(2r-1-x_b)} \bmod r$ to encrypt/decrypt messages.

Message	Mathematical relation
Encrypted	$c = m \cdot k_r \mod r$
Decrypted	$m = c \cdot {k_r}^{-1} \mod r$

5.2 Prefix attack

Consider the protocol running with $n = 4r$ over a public channel. When an eavesdropper captures the integers from the public channel, where one of them, say w_a, is a prefix of the second number written as $w_{ab} = w_a \cdot k_b \mod 4r$. To derive k_b, the attacker computes the inverse of the prefix that is $(w_a)^{-1}$ to factorize it from the second number. However, in $\mathbb{Z}_{4r}$ w_a and w_{ab} are non-invertible integers, thus the attacker must perform first multiplication by 2^{-2} changing the module from $4r$ to r.

Therefore, if the eavesdropper has captured w_a and w_{ab} from the public channel, she proceeds dividing them by 4 thus getting $w_a{}'$ and $w_{ab}{}'$. Eve computes $(w_a{}')^{-1}$ and she gets $(w_a{}')^{-1} \cdot w_{ab}{}'$. As a consequence, Eve obtains $k_b \mod r$ provided $k_b < r$. To avoid a prefix attack k_b must be chosen to be greater than the integer prime r. The steps are indicated as follows:

$$
\begin{aligned}
&w_a = 4x_a \cdot k_a \mod 4r\\
&w_{ab} = w_a \cdot k_b = 4x_a \cdot k_a \cdot k_b \mod 4r\\
&w_a{}' = w_a \cdot 4^{-1} = x_a \cdot k_a \mod r\\
&(w_a{}')^{-1} = (x_a \cdot k_a)^{-1} \mod r\\
&k_b = (w_a{}')^{-1} \cdot w_{ab}{}' = k_b \mod r
\end{aligned}
$$

5.3 Multiplication-based attack

Consider again that $p = q = 2$, then $\phi(4r) = 2r - 2$. If the eavesdropper knows P which is computed as $P = 2^{2x}k \bmod 4r$, we affirm that she cannot derive $2^{2x}k \bmod r$ because she ignores $2^{2x}k$. However, after dividing P by 4 she gets $2^{2x-2}k \bmod r$. The eavesdropper can perform the product of the public components P and Q:

$$
\begin{aligned}
&P = 2^{x}k \bmod 4r,\\
&Q = 2^{2r-1-x}k \bmod 4r \text{ because } y = 2r - 2 - x + 1\\
&P \cdot Q = 2^{2r-1}k^2 \bmod 4r\\
&P \cdot Q \cdot 2^{-2} = 2^{2r-3}k^2 \bmod r\\
&k^2 \equiv P \cdot Q \cdot 2^{-2} \cdot (2^{2r-3})^{-1} \bmod r
\end{aligned}
$$

As a result, Eve can derive the private key k. To avoid such attack, the exponent is chosen to be $2x$ instead of x:

$$
\begin{aligned}
&P = 2^{2x}k \bmod 4r,\\
&Q = 2^{2r-1-x}k \bmod 4r \text{ where } x < 2r - 1\\
&P \cdot Q = 2^{x+2r-1}k^2 \equiv 2^{x}\ 2^{2r-1}k^2 \bmod 4r
\end{aligned}
$$

In this case, Eve cannot compute the multiplicative inverse of 2^x because she does not know x and she cannot obtain k. As a final remark, we highlight that the security of the non-invertible KEP is supported in the basis of perfect secrecy which means that the integers in the private key (x_i, k_i) are uniformly distributed among the integers in $\mathbb{Z}_n$. As a consequence, we devise that just implementing an exhaustive search of the private key could compromise the security of the system. It is known, that in the quantum era, Grover's algorithm would reduce the space search to half [28]. For us, this implies that the size of the private key must be chosen properly, but this feature allow us to claim that our method can be considered post-quantum.

5.4 Mathematical representation

In the rest of the paper we will use the following mathematical notation: (P_i, Q_i) constitutes the public key of user i. As stated before $P_i = p^{2x_i}k_i$ and $Q_i = q^{y_i}k_i$ where (x_i, k_i) constitute the private key of user i and $x_i + y_i = \phi(n) + 1$. As stated before, user j raises the public key of i to its private key. Then j returns to i the integer number $[k_{i,j}]\ k_i$ where $[k_{i,j}] = p^{2x_i x_j} q^{y_i y_j}$ and k_i is a component of the private key of user i, then he applies the inverse of k_i in order to derive the shared secret key $k_{i,j}$. The same procedure is applied in the opposite direction so user i sends to j the integer $[k_{i,j}]\ k_j$ to get the secret number $k_{i,j}$ (see Table 3).

Table 3: Mathematical representation. All operations are performed module n.

Short notation	Mathematical operation
(P_i, Q_i)	$P_i = p^{2x_i}k_i,\ Q_i = q^{y_i}k_i$
${P_i}^{x_j} \cdot {Q_i}^{y_j}$	$\left(p^{2x_i}k_i\right)^{x_j} \cdot (q^{y_i}k_i)^{y_j}$
$[k_{i,j}]\ k_i$	$p^{2x_ix_j}q^{y_iy_j}k_i$

6 Certification Authority (CA)

We will introduce the public key certification method so that a Certification Authority (CA) can certify the user's public keys using Lizama's non-invertible method:

1. To certify his key with certification authority CA, a user i sends to CA his public key (P_i, Q_i).
2. If CA approves the request of i, she generates and publishes the certified key $[k_{i,ac}]\ k_i$ where $[k_{i,ac}]\ k_i$ has been derived according to Table 3.
3. The CA's public database of certified keys can be seen in Tab. 4 that contains the certified keys of Alice and Bob.

Table 4: CA's public database. The certification authority CA publishes her public key (P_{ca}, Q_{ca}).

User	Public key	Certified key
CA	(P_{ca}, Q_{ca})	-
Alice	(P_a, Q_a)	$[k_{a,ca}]\ k_a$
Bob	(P_b, Q_b)	$[k_{b,ca}]\ k_b$

In order to Alice and Bob can establish a secret key with certified keys, Alice downloads Bob's certified key from CA's database and viceversa. The steps they follow are depicted in Fig. 5 and described as follows:

1. Using CA's public key (P_{ca}, Q_{ca}), Alice computes $[k_{a,ca}]\ k_{ca}$. Also, she computes $[k_{a,b}]\ k_b$ using Bob's public key (P_b, Q_b).
3. Alice multiplies them by Bob's certified key $[k_{b,ca}]k_b$ and sends the resulting integer number to Bob. The same procedure is applied by him.
4. Bob multiplies the received integer by ${k_b}^{-1}$ twice, thus she obtains the secret shared key $K_{ab} = [k_{a,b}][k_{b,ca}][k_{a,ca}]k_{ca}$ (see Fig. 5).
5. Applying this procedure Bob derives the same secret number K_{ab}.

It must be highlighted that in order to derive the secret key it must be applied the certified key of the intended user and the public key of the certification authority CA. In addition, once reception it must be applied (twice) the private key to get the shared secret key. Also, to avoid a prefix attack the relation $K_{ab} > r$ must be satisfied.

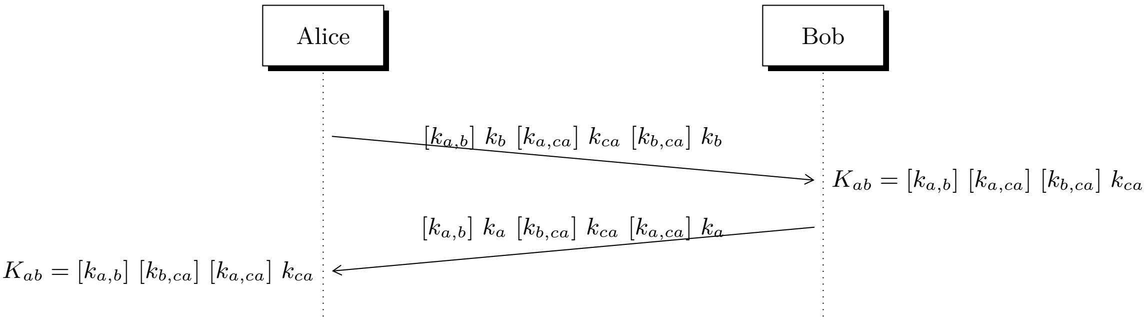
Alice
Bob
$[k_{a,b}]\ k_b\ [k_{a,ca}]\ k_{ca}\ [k_{b,ca}]\ k_b$
$K_{ab} = [k_{a,b}]\ [k_{a,ca}]\ [k_{b,ca}]\ k_{ca}$
$[k_{a,b}]\ k_a\ [k_{b,ca}]\ k_{ca}\ [k_{a,ca}]\ k_a$
$K_{ab} = [k_{a,b}]\ [k_{b,ca}]\ [k_{a,ca}]\ k_{ca}$

Fig. 5: Non-invertible KEP with certification authority (CA). All operations are performed module $4r$.

6.1 Indistinguishably of exchanged integers

Let us represent the exchanged messages across the public channel represented in Fig. 5 as $M_b \cdot k_b$ mod m from Alice to where $M_b = [k_{a,b}]k_b[k_{a,ca}]k_{ca}[k_{b,ca}]$. Similarly, $M_a \cdot k_a$ mod n implies that $M_a = [k_{a,b}]k_a[k_{b,ca}]k_{ca}[k_{a,ca}]$. Applying division between pq we have that:

$$(pq)^{-1} M_i \cdot k_i \bmod r$$

M_i mod r and k_i mod r are integers in $\mathbb{Z}_r$. Moreover, the multiplication $M_i \cdot k_i$ mod r produces a permutation of the integers in $\mathbb{Z}_r$ because r is an integer prime, thus the resulting integer is in $\mathbb{Z}_r$. As it was shown indistinguishably for encrypted messages in [5], as long as k_i remains secret, exchanged integers in the protocol accomplish perfect secrecy. Thanks to this property the unique opportunity for the eavesdropper is to find the secret key k_i by exhaustive search.

6.2 Multiple CA's

Suppose Alice has been registered with CA$_1$ while Bob has a certified key from CA$_2$. Also, Alice receives from Bob its certified key and viceversa but Alice does not have access to CA$_2$'s database neither Bob to CA$_1$'s database. As indicated in Tab. 5 CA$_1$'s database is accessible to Alice and CA$_2$'s database is reachable by Bob. However, as can be seen there, CA$_1$'s database contain the certified key of CA$_2$ and CA$_2$'s database contain the certificate of CA$_1$. Then, they follow the steps depicted in Fig. 6 and detailed below:

1. Using CA$_1$'s public key (P_{ca_1}, Q_{ca_1}), Alice computes $[k_{a,ac_1}]\ k_{ac_1}$, also she computes $[k_{a,b}]\ k_b$ with Bob's public key (P_b, Q_b).
3. Alice multiplies them by Bob's certificate $[k_{b,ca_2}]k_b$ and CA$_2$'s certificate $[k_{ca_1,ca_2}]\ k_{ca_2}$ and sends the resulting integer number to Bob. The same procedure is applied by Bob.
4. Alice multiplies the received integer by ${k_a}^{-1}$ twice, thus she obtains the secret shared key $K_{ab} = [k_{a,b}][k_{a,ca_1}]k_{ca_1}[k_{b,ca_2}]k_{ca_2}[k_{ca_1,ca_2}]$ (see Fig. 6).
5. Applying the same procedure Bob derives the secret shared number K_{ab}.

Table 5: Public databases of CA$_1$ and CA$_2$ which would be located distantly, so database of CA$_1$ is accessible to Alice and CA$_2$'s database is close to Bob.

CA	User	Public key	Certified key
CA$_1$	CA$_1$	(P_{ca_1}, Q_{ca_1})	-
	CA$_2$	(P_{ca_2}, Q_{ca_2})	$[k_{ca_1,ca_2}]\ k_{ca_2}$
	Alice	(P_a, Q_a)	$[k_{a,ca_1}]\ k_a$
CA$_2$	CA$_2$	(P_{ca_2}, Q_{ca_2})	-
	CA$_1$	(P_{ca_1}, Q_{ca_1})	$[k_{ca_1,ca_2}]\ k_{ca_1}$
	Bob	(P_b, Q_b)	$[k_{b,ca_2}]\ k_b$

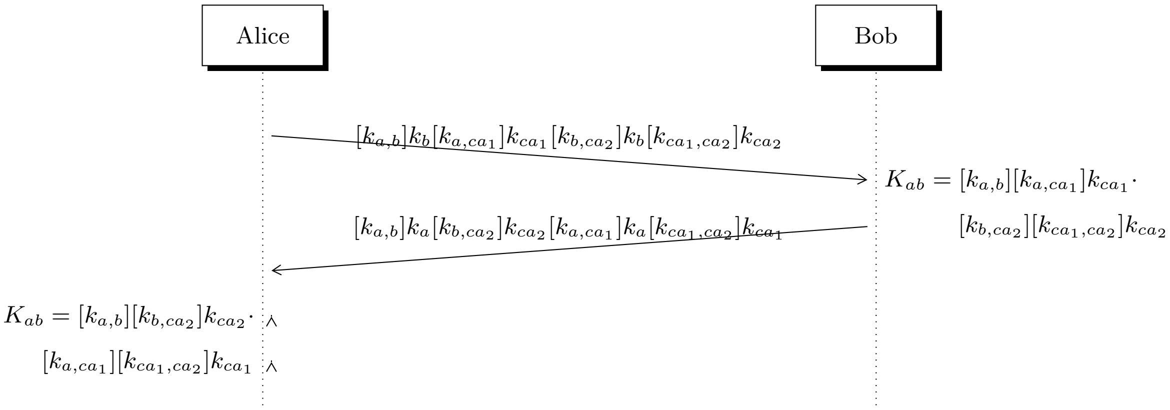
Alice
Bob
$[k_{a,b}]k_b[k_{a,ca_1}]k_{ca_1}[k_{b,ca_2}]k_b[k_{ca_1,ca_2}]k_{ca_2}$
$K_{ab} = [k_{a,b}][k_{a,ca_1}]k_{ca_1}\cdot$
$[k_{b,ca_2}][k_{ca_1,ca_2}]k_{ca_2}$
$[k_{a,b}]k_a[k_{b,ca_2}]k_{ca_2}[k_{a,ca_1}]k_a[k_{ca_1,ca_2}]k_{ca_1}$
$K_{ab} = [k_{a,b}][k_{b,ca_2}]k_{ca_2}\cdot$
$[k_{a,ca_1}][k_{ca_1,ca_2}]k_{ca_1}$

Fig. 6: Non-invertible KEP with two CAs. Operations are performed module $4r$.

7 Perfect Forward Secrecy (PFS)

Suppose Alice and Bob, require to establish a new confidential communication. However, they do not want to use the same secret key of the last session. Perfect forward secrecy (PFS) is a feature of key agreement protocols that guarantee that if the currently key was compromised it does not compromise the security of previously used keys. Therefore, the security of encrypted messages using old keys persists. When a system has a perfect forward secret, the system is said to be forward secure.

Table 6: Useful mathematical operations to achieve perfect secrecy (PFS).

Short notation	Mathematical operation
(P_i, Q_i)	$P_i = p^{2x_i} k_i,\ Q_i = q^{y_i} k_i$
$P_i{}^{x_j} \cdot Q_i{}^{y_j}$	$(p^{2x_i} k_i)^{x_j} \cdot (q^{y_i} k_i)^{y_j}$
$[k_{i,j}]\ k_i$	$p^{2x_i x_j} q^{y_i y_j} k_i$
$P_i{}^{k_s x_j} \cdot Q_i{}^{k_s y_j}$	$(p^{2x_i} k_i)^{k_s x_j} \cdot (q^{y_i} k_i)^{k_s y_j}$
$[k_{i,j}]^{k_s}\ k_i{}^{k_s}$	$p^{2k_s x_i x_j} q^{k_s y_i y_j} k_i{}^{k_s}$

In the next procedure, we demonstrate that Lizama's non-invertible KEP is enhanced to exhibit PFS (see Tab. 6 and Fig. 7).

1. Alice and Bob share a certified key K_i from a previous exchange.
2. Using CA's public key (P_{ca}, Q_{ca}), Alice computes $[k_{a,ca}]\ k_{ca}$. Also, according to Tab. 6, Alice computes $[k_{a,b}]^{K_i}\ k_b{}^{K_i}$ using Bob's public key (P_b, Q_b).
4. Alice multiplies them by Bob's certificate $[k_{b,ca}]\ k_b$ and sends the resulting number to Bob. The same procedure is applied by Bob.
5. Bob multiplies the received integer by $k_b{}^{-K_i-1}$, thus she obtains the secret shared key $K_{i+1} = [k_{a,b}]^{K_i}[k_{a,ca}][k_{b,ca}]k_{ca}$ (see Fig. 7).
6. Conversely, Alice multiplies the received integer by $k_a{}^{-K_i-1}$, thus she gets the secret shared key $K_{i+1} = [k_{a,b}]^{K_i}[k_{b,ca}][k_{a,ca}]k_{ca}$.

The eavesdropper cannot derive K_i from K_{i+1} and the procedure can be repeated as many times as required to derive K_{m+1} from K_m.

8 Conclusions

We have demonstrated that Lizama's non-invertible key exchange protocol can be used to implement a public key cryptosystem with single and multiple certification domains. The required size of keys in non-invertible certificates seems to be manageable to attend some issues as fragmentation, segmentation and caching.

In addition, we have discussed a method to achieve Perfect Forward Secrecy (PFS) for session keys which can be used as many times as required.

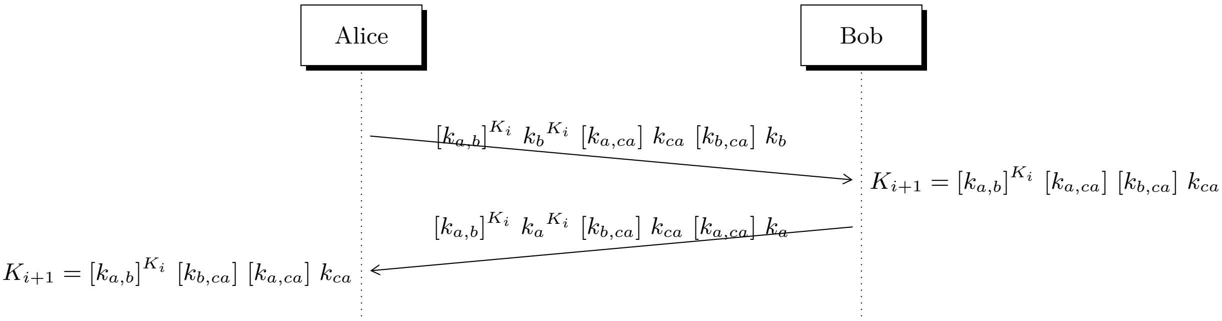
Alice
Bob
$[k_{a,b}]^{K_i}$ $k_b{}^{K_i}$ $[k_{a,ca}]$ k_{ca} $[k_{b,ca}]$ k_b
$K_{i+1} = [k_{a,b}]^{K_i}$ $[k_{a,ca}]$ $[k_{b,ca}]$ k_{ca}
$[k_{a,b}]^{K_i}$ $k_a{}^{K_i}$ $[k_{b,ca}]$ k_{ca} $[k_{a,ca}]$ k_a
$K_{i+1} = [k_{a,b}]^{K_i}$ $[k_{b,ca}]$ $[k_{a,ca}]$ k_{ca}

Fig. 7: Alice and Bob require to establish a new secret key K_{i+1}. However, they do not want to use the last secret key K_i. This procedure is repeated to derive K_{i+2} from K_{i+1}.

A Appendix

A.1 RSA cryptosystem

The security of RSA cryptosystem relies in the difficulty of the integer factorization problem. Two invertible numbers e and d are chosen inside the ring defined by $\mathbb{Z}_{\phi(n)}$, so that $e \cdot d \equiv 1 \bmod \phi(n)$. The other ring $\mathbb{Z}_n$ is prepared with $n = p \cdot q$ where p and q are secret prime integers [29]. The encrypted message is computed as $C = M^e \text{mod } n$ while $M = C^d \text{mod } n$ returns the original cleartext M. The cryptosystem works because of Euler's theorem since $(M^e \text{mod } n)^d \text{mod } n = M^{ed} \text{mod } n$ but $e \cdot d = k\phi(n) + 1$, so $M^{k\phi(n)+1} \text{mod } n = M^{k\phi(n)} \cdot M^1 \text{mod } n$ which yields M provided $M < n$.

A.2 Diffie-Hellman key exchange

Diffie-Hellman key exchange (DH) was the first public key exchange algorithm [19]. The integer prime p defines a ring $\mathbb{Z}_p$ and the generator g is a primitive root module p. The integers p and g are publicly known.

Alice chooses randomly the exponent x_a and she computes $k_a = g^{x_a} \bmod p$ which she sends to Bob over a public channel. On the other side, Bob obtains $k_b = g^{x_b} \bmod p$, then he communicates this integer number to Alice across the channel.

Alice and Bob execute exponentiation over the received number, such that Alice's gets $(g^{x_b} \bmod p)^{x_a} \bmod p = g^{x_b x_a} \bmod p$. Conversely Bob gets $(g^{x_a} \bmod p)^{x_b} \bmod p = g^{x_a x_b} \bmod p$. The two operations yield the same integer number because multiplication of exponents is commutative. The security of the secret shared key relies on the difficulty that given g, k_a and k_b it is computationally infeasible to derive $g^{x_a x_b} \bmod p$.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pp. 124–134, IEEE, 1994.

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

[3] I. T. Laboratory, "PQC Standardization Process: Third Round Candidate Announcement." `https://csrc.nist.gov/news/2020/pqc-third-round-candidate-announcement`, 2020. [Online; accessed 08-October-2020].

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

[5] L. A. Lizama-Perez, "Non-invertible key exchange protocol," *SN Applied Sciences*, vol. 2, p. 1083, 2020.

[6] 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)*, pp. 175–9, 1984.

[7] L. A. Lizama-Pérez, J. M. López, E. De Carlos-López, and S. E. Venegas-Andraca, "Quantum flows for secret key distribution in the presence of the photon number splitting attack," *Entropy*, vol. 16, no. 6, pp. 3121–3135, 2014.

[8] L. A. Lizama-Pérez, J. M. López, and E. De Carlos López, "Quantum key distribution in the presence of the intercept-resend with faked states attack," *Entropy*, vol. 19, no. 1, p. 4, 2016.

[9] L. J. Lizama-Perez LA, "Quantum key distillation using binary frames.," *Symmetry*, vol. 12, no. 6, p. 1053, 2020.

[10] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[11] S. Wang, Y. Zhu, D. Ma, and R. Feng, "Lattice-based key exchange on small integer solution problem," *Science China Information Sciences*, vol. 57, no. 11, pp. 1–12, 2014.

[12] "Criptoanálisis del protocolo de intercambio de claves basado en celosía de wang et al.," *Perspectives in Science*.

[13] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *International Workshop on Post-Quantum Cryptography*, pp. 19–34, Springer, 2011.

[14] C. Costello, P. Longa, and M. Naehrig, "Efficient algorithms for supersingular isogeny diffie-hellman," in *Annual International Cryptology Conference*, pp. 572–601, Springer, 2016.

[15] T. Matsumoto and H. Imai, "Public quadratic polynomial-tuples for efficient signature-verification and message-encryption," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 419–453, Springer, 1988.

[16] R. C. Merkle, "Method of providing digital signatures," Jan. 5 1982. US Patent 4,309,569.

[17] L. A. Lizama-Perez, "Digital signatures over hash-entangled chains," *SN Applied Sciences*, vol. 1, no. 12, p. 1568, 2019.

[18] 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.

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

[20] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[21] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the theory and application of cryptographic techniques*, pp. 417–426, Springer, 1985.

[22] N. Bindel, U. Herath, M. McKague, and D. Stebila, "Transitioning to a quantum-resistant public key infrastructure," in *International Workshop on Post-Quantum Cryptography*, pp. 384–405, Springer, 2017.

[23] G. Pradel and C. J. Mitchell, "Post-quantum certificates for electronic travel documents," 2019.

[24] P. Kampanakis, P. Panburana, E. Daw, and D. Van Geest, "The viability of post-quantum x. 509 certificates.," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 63, 2018.

[25] W. Polk, R. Housley, and L. Bassham, "Algorithms and identifiers for the internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile," *Algorithms*, vol. 2, p. 26, 2002.

[26] E. Gerck *et al.*, "Overview of certification systems: x. 509, ca, pgp and skip," 1997.

[27] A. Abdul-Rahman, "The pgp trust model," in *EDI-Forum: the Journal of Electronic Commerce*, vol. 10, pp. 27–31, 1997.

[28] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, ACM, 1996.

[29] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.