Non-invertible key exchange protocol

Luis Adrian Lizama-Perez¹

Received: 29 October 2019 / Accepted: 17 April 2020 / Published online: 15 May 2020

© The Author(s) 2020

Abstract

We investigate a cryptosystem through what we call non-invertible cryptography. As a result of a continuous refinement process, we present a new key exchange method to establish a secret key between two remote parties. Non-invertible KEP is supported by Euler's theorem as RSA, it uses exponentiation to exchange a secret key as Diffie–Hellman, and it encrypts/decrypts through invertible multiplication as ElGamal. This method is public key; it allows secret key exchange and performs secret communication. Most remarkably, since it does not rely on computational problems as integer factorization or discrete logarithm whose difficulty is conjectured, non-invertible KEP becomes a promising candidate to protect communication in the quantum era. By contrast, the algorithm is supported on indistinguishability of public key and ciphertext so it achieves perfect secrecy. The protocol demonstrates minimum required time for encryption/decryption processes when is compared with the main public key algorithms as Diffie–Hellman, ElGamal or RSA.

Keywords Non-invertible · Cryptography · Ring

1 Introduction

Since its origin at the end of the 1970s, public key cryptography has become the main mechanism to support digital signatures and electronic internet services. By means of public/private keys, users and data can be authenticated during web transactions. In addition, users can verify the authenticity of web servers, and reciprocally, the service can approve or reject a transaction from an electronic payment card. A new software version can be verified before it is updated on a computer, mobile phone, or automobile system. Other capabilities include verification of electronic documents and issuance of digital receipts.

RSA and ECC are the most prominent algorithms used to implement electronic signature services. However, due to the advent of quantum computers, the security of such methods has been threatened by Shor's algorithm that solves, at least in theory, the computational problems on which are based those algorithms: integer factorization and discrete logarithm, respectively. 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 [1].

In view of the above, the National Institute of Standards and Technology (NIST) has initiated a process to evaluate cryptographic algorithms that could be used to achieve confidentiality and authentication services in the quantum era. Currently, the evaluation of candidate algorithms is in the second evaluation round. According to the criteria defined by NIST, algorithms must be resistant against classical and quantum adversaries, and their security level must be comparable to the security of SHA-385 and AES256. The size of the keys must be acceptable, as well as the required computing resources. Other criteria include facility of implementation (in hardware and software), whether it is an enhanced algorithm or a new one. The flexibility of the algorithm will be evaluated because of its ability to encrypt messages, perform digital signatures and/or allow key exchange.

✉ Luis Adrian Lizama-Perez, adrianlizama@gmail.com | ¹Sección de Posgrado de la Universidad Politécnica de Pachuca, Ex-Hacienda de Santa Bárbara, 43830 Pachuca, Mexico.

Even so, it is possible to continue using the algorithms RSA, ECDSA and DH, since their properties has been studied over the past years. The downside to this approach is that it can lead to increase the size of the keys to an impractical level in the quantum era. However, 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 [2, 3]. Post-quantum cryptosystems under evaluation for public-key quantum-resistant [4] include cryptography based on lattices, multivariate-based, hash-based [5, 6] or code-based [7]. We will describe them in the next section, for now let us briefly summarize some of the main algebraic cryptosystems used today which are based on integer factoring and discrete logarithm computational problems:

1. The key exchange protocol introduced by Diffie and Hellman [8]. The asymmetric key encryption algorithm for public-key cryptography of Taher ElGamal [9]. Both of them are based on the Diffie–Hellman assumption.
2. Cryptosystems that rely on the difficulty of the integer factorization problem: the RSA method for obtaining digital signatures by Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman [10]. Also, it can be cited the digitalized signatures by Rabin, Michael O Rabin [11].
3. Schemes based on the discrete logarithm with elliptic curve groups known as ECC developed by Miller and Koblitz [12, 13].

Just to make a simple comparison about their performance, Diffie–Hellman and El Gamal algorithms have similar temporal complexity. According to [14], RSA algorithm is the slowest one. In all cases, the computational speed is around 10^6 – 10^8 ms¹.

In this work, we will introduce a new key exchange method based in non-invertible cryptography that can be useful for secret communication in the quantum era. In Appendix, we include some definitions about modular arithmetic that we used to design the non-invertible key exchange protocol (KEP). We used a refinement design methodology so that at each stage we evaluated the security of the method against integer factorization. We describe KEP at three stages: multiplication-based, exponent-based and non-invertible KEP. In the first and second stages, the protocol is intended to pass secretly an integer number from Alice to Bob. However, we should note that they fail when trying to transmit securely an integer number from Alice to Bob. At the final stage, the

non-invertible KEP behaves as a secret key exchange algorithm and it achieves the security goal. Before we develop our protocols, let us briefly mention the current cryptosystems intended to allow secret key establishment. Also, in the next section we will describe some closely related protocols.

1.1 Quantum cryptography

A new approach has emerged for the quantum era which is based in quantum physics principles; it is called quantum key distribution and is useful for secret key establishment [2]. Quantum key distribution exploits the principle that an eavesdropper cannot alter quantum communication without producing a detectable noise. The technique has been enhanced to resist quantum attacks [15, 16].

1.2 Post-quantum cryptography

Some cryptographic protocols and key exchange methods have been conceived in the field of post-quantum cryptography for the quantum era. Let us introduce briefly some of the main cryptosystems [17, 18].

1.2.1 Code-based cryptography

Code-based cryptography was originally proposed by McEliece [19] in 1978, who described an asymmetric key cryptographic system based on the difficulty of decoding a generic linear code, which is an NP-difficult problem. A linear code is essentially a form of error correction code. The private key is a code C , which has the ability to correct t errors. When sending a message, the sender will encode the message with the public key and include t errors within the encoding; then, the ciphertext is obtained by adding an error vector to each codeword. The receiver with code C will be able to decode the message while accurately correcting the errors. Some properties of this system are high confidence, fast encryption but larger public keys. Some key exchange algorithms are BIKE, Classic McEliece, HQC, NTS-KEM, RQC [20].

1.2.2 Lattice-based key exchange

Wang et al. [21] proposed in 2014 a key exchange protocol which has no longer dependent on conventional computational problems. The key exchange protocol is based in what is named lattice-based cryptography [22]. The techniques are generally used in modern cryptography as a complex security in comparison with the conventional cryptographic techniques. Some of the notable attributes of lattice KEM algorithms generate short ciphertext and keys, short signatures, good performance,

¹ Computations in [14] were performed in a Pentium 133 MHz with 16 MB RAM.

but sometimes complex. Examples of KEM systems are FrodoKEM, NewHope, NTRU, FALCON and qTESLA.

1.2.3 Supersingular elliptic curve isogeny cryptography

Jao and De Feo [23] introduced an elliptic curve-based alternative to elliptic curve Diffie–Hellman (ECDH) which is not vulnerable to Shor’s quantum attack [24] called supersingular isogeny Diffie–Hellman (SIDH) key exchange protocol which is based in the difficulty of finding isogenies between supersingular elliptic curves. Isogeny computations constitute an algebraic map between elliptic curves, which appear to be resistant to quantum attacks. SIDH produces very small key sizes, but it shows slower performance. The representative algorithm is the Supersingular Isogeny Key Encapsulation (SIKE).

1.2.4 Hash-based cryptography

Cryptographic services have been conceived over the basis of hash functions. Lamport first introduced hash-based digital signature (OTS) in 1979 [25]. The signature is based on the difficulty of inverting a hash function. Unfortunately, signatures are ephemeral and a new public key must be computed for each signature. There have been some improvements to Lamport’s signature through variations in the use of Merkle trees [26] and Lizama’s hash-based methods [5, 6].

1.2.5 Zero knowledge

Zero knowledge proofs have their origin in a method by which the prover can demonstrate to the verifier that a mathematical statement is true, without revealing anything other than the truth of the statement. It was first introduced in 1985 [27]. Zero knowledge interactive proofs like ZK-STARKs (Zero-Knowledge Scalable Transparent ARguments of Knowledge) are post-quantum secure, which have application in voting systems, blockchain’s transactions and proving identity information [28].

1.2.6 Multivariate cryptography

Solving systems of multivariate polynomial equations is proven to be NP-complete. That is why multivariate cryptography is often considered to be post-quantum secure. The basic objects of multivariate cryptography are systems of nonlinear (usually quadratic) polynomial equations in several variables over a finite field. The problem is proven to be NP hard.

When performing the digital signature, the set of equations is the public key. A signature will be valid if at least a number of equations are satisfied so that a valid

authentication value or a valid signature. The verifier can apply to verify that the output of the equations corresponds to the hash of the message that is signed.

2 Related protocols

Some cryptographic protocols have significant mathematical properties closely related to our interests, specifically: modular invertible multiplication and modular exponentiation as well as Euler’s theorem in number theory. Let us explain succinctly such protocols.

2.1 Linear ciphers

Some ciphers have been conceived to exploit the invertible numbers in $\mathbb{Z}_n$. For example, a linear cipher encrypts a message M according to the operation $E(M) = k \cdot M \pmod{26}$, where k is relatively prime to 26. In a linear cipher, as the affine cipher, the encryption algorithm specifies multiplication of the plaintext by the key. Decryption involves multiplication by the multiplicative inverse of the key [29]. However, an opponent can derive the plaintext of two enciphered characters applying frequency analysis or a brute force attack.

2.2 Exponential ciphers

In an exponential cipher, a message is encrypted with an exponent k , so that $C = M^k \pmod{n}$. The decryption process is $M = C^{k^{-1}} \pmod{n}$ because k and k^{-1} are multiplicative inverses in $\phi(n)$, that is $k \cdot k^{-1} \equiv 1 \pmod{\phi(n)}$. The most remarkably exponential cipher is the Rivest–Shamir–Adleman (RSA) cryptosystem.

2.3 RSA cryptosystem

A ring $\mathbb{Z}_n$ is defined with $n = p \cdot q$ where p and q are kept as secret prime numbers [10]. A second ring is prepared taking $\phi(n)$ as the module. Inside this ring, two invertible numbers e and d are chosen, so that $e \cdot d \equiv 1 \pmod{\phi(n)}$. To encrypt a message, the following operation is executed $C = M^e \pmod{n}$. To recover the message is computed $M = C^d \pmod{n}$. The correctness of the algorithm relies in Euler’s theorem since $(M^e \pmod{n})^d \pmod{n} = M^{ed} \pmod{n}$ but $e \cdot d = k\phi(n) + 1$, so $M^{k\phi(n)+1} \pmod{n} = M^{k\phi(n)} \cdot M^1 \pmod{n}$ which reduces to M when $M < n$. The security of the protocol relies as mentioned at the beginning, in the difficulty of the integer factorization problem. RSA is an asymmetric system, thus allowing remote user authentication.

Table 1 Multiplication-based protocol

Purpose: Transmit a secret number from Alice to Bob		
Description	Operations	Math properties
A secret number can be transmitted from Alice to Bob when it is multiplied by a non-invertible integer because the product cannot be factored by Eve	$A \rightarrow B : v_a k_a$	The product of an invertible by a non-invertible integer gives a non-invertible integer
	$B \rightarrow A : v_a k_a k_b$	
	$A \rightarrow B : v_a k_a k_b k_a^{-1} = v_a k_b$	The product of an invertible integer by its inverse produces unity
	$B : v_a k_b k_b^{-1} = v_a$	

2.4 Diffie–Hellman key exchange

Another public algorithm broadly used today that uses exponential computation is Diffie–Hellman (DH) cipher [8]. In this case, a ring $\mathbb{Z}_p$ is used and two integers are publicly shared: the module p and the generator g which is primitive root in $\mathbb{Z}_p$. Alice chooses randomly an exponent x_a sending $k_a = g^{x_a} \bmod p$ to Bob. Similarly, Bob computes and responds to Alice with $k_b = g^{x_b} \bmod p$. Each one performs the exponentiation again using the received number, such that at Alice's side $(g^{x_b} \bmod p)^{x_a} \bmod p = g^{x_b x_a} \bmod p$. And at Bob's side $(g^{x_a} \bmod p)^{x_b} \bmod p = g^{x_a x_b} \bmod p$. Both results are equal since modular exponentiation obeys all of the normal rules of ordinary exponentiation. Unlike RSA, in the Diffie–Hellman algorithm there is no need to kept secret the prime number p . The security of the shared key depends on the difficulty that given g, k_a and k_b it is computationally intractable to compute the value of $g^{x_a x_b} \bmod p$.

2.5 ElGamal cipher

ElGamal cipher is based in the Diffie–Hellman key exchange algorithm. Suppose k_a and k_b are shared publicly, to send an encrypted message to Bob, Alice obtains $k = g^{x_b} \bmod p$. Now, she chooses a random x_s to get a one-time key $k_s = g^{x_s x_b} \bmod p$. Alice computes $C_1 = g^{x_s} \bmod p$ and encrypts the message as $C_2 = M \cdot k_s \bmod p$. Then she sends (C_1, C_2) to Bob who uses C_1 to compute $k_s = g^{x_s x_b} \bmod p$. Finally, Bob obtains the multiplicative inverse of k_s to retrieve the original message, so $M = M \cdot k_s \cdot k_s^{-1} \bmod p$.

2.5.1 Discussion

Several key exchange methods have been proposed to establish a secret integer between two remote parties [30–32]. Unfortunately, such protocols are supported on factorization problem and DH problem, which are known to be vulnerable to quantum computers [24].

Previously, we described RSA, DH and ElGamal because as we will discuss next, KEP is closely related to them.

Non-invertible KEP is supported by Euler's theorem as RSA, it uses exponentiation to exchange a secret key as DH, and it encrypts/decrypts through invertible multiplication as ElGamal. However, we have taken advantage of the main properties of each of them to achieve a post-quantum cryptosystem whose security does not rely on the integer factorization or the discrete logarithm problem.

3 Multiplication-based protocol

As stated in the previous section, in a ring with unity over $\mathbb{Z}_n$, an integer may or may not have a multiplicative inverse. The great majority of cryptographic systems based in number theory have not been interested in non-invertible integers. In contrast, we are not just interested in invertible but also in non-invertible (multiplicative) integers.

The multiplication-based protocol, is supported by the basic properties of modular multiplication. To simplify the explanation of this and subsequent protocols, we have omitted the operation symbol $\bmod n$ in text and figures (see Table 1).

1. The multiplication of an invertible integer denoted as k (the multiplicative inverse is written as k^{-1}) and a non-invertible number called v (which means that v^{-1} does not exist in $\mathbb{Z}_n$) gives a non-invertible integer, say w (therefore w^{-1} would not exist in $\mathbb{Z}_n$). To see that consider the multiplication $v \cdot k = w$, if w were invertible then $w \cdot w^{-1} = 1$. It implies that $v \cdot k w^{-1} = 1$; however, we have chosen v as a non-invertible number, so v does not meet a multiplicative integer that yields unity. Thus, we arrived to a contradiction that suggests w is a non-invertible integer in $\mathbb{Z}_n$. Following the same argumentation, we arrive to the next statements:
2. The multiplication of an invertible integer k_i and another invertible one k_j gives an invertible integer.
3. The multiplication of a non-invertible integer v and another non-invertible integer w gives a non-invertible integer.

Table 2 The integers used in the protocol

User	Non-invertible integer	Invertible integer
Alice	v_a	k_a
Bob	–	k_b

Alice transmits v_a to Bob

Fig. 1 Multiplication-based protocol. All computations in the protocol are performed mod pq . The prime numbers p and q are known publicly

From the previous statements, we can explain the multiplication-based protocol. The protocol is aimed to transfer a secret value generated by Alice, say v_a , to the remote side called Bob through a public channel in the presence of an eavesdropper named Eve. To initialize the protocol, Alice and Bob must generate the integer values listed in Table 2.

The general idea of the protocol can be described in the following words: Alice multiplies v_a with a integer key k_a to hide it; then, she sends the result to Bob who “encrypts” the input with his integer key k_b returning the resulting number to Alice. Now, she removes her key k_a using k_a^{-1} and sending the result to Bob. Finally, Bob removes his integer key from the input applying k_b^{-1} . At this point, Bob has obtained v_a . The protocol is illustrated in Fig. 1 and described in Table 1 where Alice transmits v_a to Bob. We remark that $v_a \cdot k_a$ does not have a corresponding inverse,

that is $(v_a \cdot k_a)^{-1}$ does not exist, otherwise the attacker could factorize the second message $v_a \cdot k_a \cdot k_b$, that is $(v_a \cdot k_a \cdot k_b) \cdot (v_a \cdot k_a)^{-1} = k_b$. Similarly, the attacker could not factorize it using the third message $v_a \cdot k_b$.

3.1 Security analysis

Eve could capture the message across the public channel $v_a k_a$, $v_a k_a k_b$, $v_a k_b$, then she proceeds as described by the following strategy:

1. Eve multiplies $v_a k_a k_b$ by each invertible integer in $\mathbb{Z}_n$ until she gets $v_a k_a$ or $v_a k_b$.
2. If she obtains $v_a k_a$ is because she has found k_b^{-1} . Then she computes $v_a k_b \cdot k_b^{-1}$ to get v_a .
3. If she obtains $v_a k_b$ implies she has found k_a^{-1} . Then she computes $v_a k_a \cdot k_a^{-1}$ to get v_a .

In this strategy, Eve must find a matching integer in a search space whose size is proportional to the length of $\phi(n)$. However, Eve would decide to apply a better strategy than exhaustive search, but before we examine it let us recall some important points:

- (i) We chose $n = p \cdot q$ where p and q are prime numbers defined in $\mathbb{Z}_n$.
- (ii) Non-invertible numbers are multiples of prime p and q , thus $v_i = \{p, 2p, \dots, q, 2q, \dots\}$. More conveniently, we can write v_i as $p \cdot x_i$ where x_i represents a multiple number of prime p inside $\mathbb{Z}_n$.
- (iii) Invertible numbers include all the remaining numbers in $\mathbb{Z}_n$. They are non-multiples of prime p or q , in other words they are co-primes of $p \cdot q$.

From modular arithmetic, it must be considered, that in a field, denoted as $\mathbb{Z}_p$, where p is a prime number, all integers are invertibles. Also, the division property in $\mathbb{Z}_n$ is written in Eq. (1).

$$\frac{p x_i}{p} \mod pq = x_i \mod q \quad (1)$$

The relation can be applied inversely; thus after multiplying the right hand of the equation by p , we recover $p \cdot x_i \mod pq$. Using those properties, Eve can attack the

Table 3 The attacker applies p division to compute the multiplicative inverses of the public numbers to derive the secret value $p x_a \mod pq$

Message	p division	First factorization	Second factorization	p multiplication
$p x_a k_a \mod pq$	$x_a k_a \mod q$			
$p x_a k_a k_b \mod pq$	$(x_a k_a) k_b \mod q$	$k_b \mod q$		
$p x_a k_b \mod pq$	$x_a k_b \mod q$		$x_a \mod q$	$p x_a \mod pq$

Table 4 Exponent-based protocol

Purpose: Transmit a secret number from Alice to Bob

Description	Operations	Math properties
A secret number can be transmitted from Alice to Bob when it is the exponent of a non-invertible integer because the power cannot be factored by Eve	$A \rightarrow B : p^{x_a} k_a$	The power of a non-invertible integer is a non-invertible integer
	$B \rightarrow A : p^{x_a} k_a k_b$	The product of an invertible by a non-invertible integer gives a non-invertible integer
	$A \rightarrow B : p^{x_a} k_a k_b k_a^{-1} = p^{x_a} k_b$	The product of an invertible integer by its inverse produces unity
	$B : p^{x_a} k_b k_b^{-1} = p^{x_a}$

Fig. 2 The exponent-based protocol. All computations in the protocol are performed mod pq . The non-invertible number p^{x_a} is multiplied by the invertible k_a before it is sent to Bob

multiplication-based protocol as indicated in Table 3 to derive the secret value $p \cdot x_a \bmod pq$.

4 Exponent-based protocol

We have indicated in Sect. 3 that after multiplying an integer number (invertible or not) by another non-invertible integer in $\mathbb{Z}_n$ always results a non-invertible number. Similarly, the multiplication of an invertible integer by another invertible integer always yields an invertible number. As a

consequence, the integer number that results after exponentiation say p^{x_a} gives always a non-invertible integer. The exponent-based protocol is presented in Table 4 and Fig. 2.

The exponent-based protocol is intended to pass securely p^{x_a} from Alice to Bob. Unfortunately for this version of the protocol, as it occurred in the multiplication-based, Eve can use the division property to find the secret number $p^{x_a} \bmod pq$. The steps are indicated in Table 5.

5 Non-invertible KEP

We must emphasize that the protocols discussed so far (multiplication and exponent-based) fail when trying to pass secretly an integer number from Alice to Bob over a public channel. To overcome the limitations described before we will introduce the Euler's theorem to the exponent-based protocol, what we call non-invertible KEP. By contrast to secret transfer, this protocol is a key exchange protocol (KEP). The non-invertible KEP is described in Table 6 and Fig. 3. As used before, the numbers $\{p^{x_a} k_a, q^{y_a} k_a, n\}$ and $\{p^{x_b} k_b, q^{y_b} k_b, n\}$ along with the prime numbers p, q and r are publicly known by Alice and Bob. On the other hand, $\{x_i, k_i\}$ constitutes the private key of Alice (or Bob, respectively). Now, let us detail the steps of the protocol.

1. Alice and Bob agree to use a module n , so they compute the integers represented in Table 7. Now, they exchange them through a public channel.

Table 5 The division attack over the exponent-based protocol

Message	p division	First factorization	Second factorization	p multiplication
$p^{x_a} k_a \bmod pq$	$p^{x_a-1} k_a \bmod q$			
$p^{x_a} k_a k_b \bmod pq$	$(p^{x_a-1} k_a) k_b \bmod q$	$k_b \bmod q$		
$p^{x_a} k_b \bmod pq$	$(p^{x_a-1}) k_b \bmod q$		$p^{x_a-1} \bmod q$	$p^{x_a} \bmod pq$

The attacker computes the multiplicative inverse of every public integer to get $p^{x_a} \bmod pq$

Table 6 Non-invertible KEP

Purpose: Establish a secret shared key between Alice and Bob

Description	Operations	Math properties
A secret number is established between Alice and Bob using non-invertible power and Euler's identity	$A \rightarrow B : p^{x_a} k_a, q^{y_a} k_a$	$x_a + y_a = \phi(n) + 1$
	$B \rightarrow A : p^{x_b} k_b, q^{y_b} k_b$	$x_b + y_b = \phi(n) + 1$
	$A \rightarrow B : p^{x_a x_b} p^{y_a y_b} k_b$	$(p^{x_b} k_b)^{x_a} (q^{y_b} k_b)^{y_a} = p^{x_a x_b} p^{y_a y_b} k_b$
	$B \rightarrow A : p^{x_a x_b} p^{y_a y_b} k_a$	$(p^{x_a} k_a)^{x_b} (q^{y_a} k_a)^{y_b} = p^{x_a x_b} p^{y_a y_b} k_a$
	$A : p^{x_a x_b} p^{y_a y_b} k_a k_a^{-1} = p^{x_a x_b} p^{y_a y_b}$	The product of an invertible integer by its inverse produces unity
	$B : p^{x_a x_b} p^{y_a y_b} k_b k_b^{-1} = p^{x_a x_b} p^{y_a y_b}$	

Fig. 3 Non-invertible KEP. All computations in the protocol are performed mod pqr . Here, k_s represents the shared secret key. The sum $x_i + y_i$ is equivalent to $\phi(n) + 1$

Table 7 The modulo n is computed as pqr , n and the prime numbers p , q and r are publicly known. The private key is (x, k_i) where $i = a, b$

User	Public Key
Alice	$\{P_{ap} = p^{x_a} \cdot k_a \bmod n, P_{aq} = q^{y_a} \cdot k_a \bmod n\}$
Bob	$\{P_{bp} = p^{x_b} \cdot k_b \bmod n, P_{bq} = q^{y_b} \cdot k_b \bmod n\}$

- Alice and Bob perform two operations over the numbers received: exponentiation and multiplication as indicated in Table 8.
- The results listed in Table 8 require us to apply the Euler's theorem defined in $\mathbb{Z}_n$, which is written in

Table 8 The operations performed at each side after the public keys are exchanged

User	Operation	Result
Alice	$(p^{x_b} \cdot k_b \bmod n)^{x_a} \cdot (q^{y_b} \cdot k_b \bmod n)^{y_a}$	$= p^{x_b x_a} q^{y_b y_a} \cdot k_b \bmod n$
Bob	$(p^{x_a} \cdot k_a \bmod n)^{x_b} \cdot (q^{y_a} \cdot k_a \bmod n)^{y_b}$	$= p^{x_a x_b} q^{y_a y_b} \cdot k_a \bmod n$

Eq. (2). Here, k and n are relative prime each other. Then, for correctness of the algorithm, x_i and y_i must sum up $\phi(n) + 1$. Thus, according to Eq.(2), we have $k^{\phi(n)+1} = k^{\phi(n)} \cdot k^1 = k$ because k is an invertible integer in $\mathbb{Z}_n$.

$$k^{\phi(n)} \equiv 1 \bmod n \quad (2)$$

4. Alice sends to Bob the resulting value $p^{x_b x_a} q^{y_b y_a} \cdot k_b \bmod n$ who applies k_b^{-1} to derive the number $p^{x_b x_a} q^{y_b y_a} \bmod n$. Similarly, Bob sends $p^{x_a x_b} q^{y_a y_b} \cdot k_a \bmod n$ to Alice who uses k_a^{-1} to get the shared secret $k_s = p^{x_a x_b} q^{y_a y_b} \bmod n$.

5.1 Cipher system

In Fig. 3, the number k_s is a non-invertible integer in $\mathbb{Z}_n$; thus, a convenient method to achieve a cipher system and secret communication is to divide the shared number $k_s = p^{x_a x_b} q^{y_a y_b} \bmod n$ by pq , so $k_r = p^{x_a x_b - 1} q^{y_a y_b - 1} \bmod r$. Now, it can be computed its multiplicative inverse k_r^{-1} . The enciphered message is obtained as $c = m \cdot k_r \bmod r$ 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 number, the integer m must be less than r .

6 Security analysis

6.1 Indistinguishability of the public key

To achieve indistinguishability of the public key (so it can be derived from every message in the ring, for details about indistinguishability see [33]), we need that p and q be primitive roots of r . Since the prime integers with primitive roots take the form as $2, 4, p^t$, and $2p^t$, where p is any prime and t is a positive integer [34]. We must redefine our prime numbers as it follows:

$$n = p \cdot q \cdot r, \text{ but making } p = q = 2$$

$$n = 4 \cdot r \text{ where } r \text{ is a large prime number such that } 2 \text{ is a primitive root of } r \text{ and it can be chosen as a usual}$$

safe prime. Therefore, the components of the public key (Alice or Bob) are:

$$P_p = 2^x k \bmod 4r,$$

$$P_q = 2^y k \bmod 4r$$

6.1.1 Multiplication-based attack

The selection for the prime integers as explained previously makes the protocol vulnerable to a multiplication attack. This attack occurs when $p = q$, thus $n = p^2 r$ but $\phi(p^2 r) = p^{2-1}(r-1)$, now the eavesdropper performs the product of the components P_p and P_q of the public key. If $p = q = 2$, then $\phi(4r) = 2r - 2$, so:

$$P_p = 2^x k \bmod 4r,$$

$$P_q = 2^{2r-1-x} k \bmod 4r \text{ because } y = 2r - 2 - x + 1$$

$$P_p \cdot P_q = 2^{2r-1} k^2 \bmod 4r$$

$$P_p \cdot P_q \cdot 2^{-2} = 2^{2r-3} k^2 \bmod r$$

$$k^2 \equiv P_p \cdot P_q \cdot 2^{-2} \cdot (2^{2r-3})^{-1} \bmod r$$

From here, Eve can derive the private key k . To avoid such attack, we must introduce the following change to the first component of the public key P_p :

$$P_p = 2^{2x} k \bmod 4r,$$

$$P_q = 2^{2r-1-x} k \bmod 4r \text{ where } x < 2r - 1$$

$$P_p \cdot P_q = 2^{x+2r-1} k^2 \equiv 2^{2r-1} 2^x k^2 \bmod 4r$$

Now, Eve cannot compute the multiplicative inverse of 2^x because she does not know x . Thus, she cannot obtain k . The relation between the secret key and the private components x_i ($i = a, b$) can be written as:

$$k_s = 2^{2x_a x_b} 2^{y_a y_b} \bmod 4r$$

$$k_s = 2^{2x_a x_b} 2^{((2r-1)-x_a)((2r-1)-x_b)} \bmod 4r$$

$$k_s = 2^{3x_a x_b - (2r-1)(x_a + x_b) + (2r-1)^2} \bmod 4r$$

To prove that indistinguishability is present in the first component of a user's public key $P_p = 2^{2x} k \bmod 4r$, we suggest the following procedure:

1. Derive the set of valid messages for P_p , say $\mathcal{M}_p$. Consider $m \cdot k \bmod 4r$ where m is computed as $m = 2^{2x} \bmod 4r$ and x is an integer taken from $\mathcal{X}_p = \{0, 1, 2, \dots, r-1\}$. Multiplying m by 2^{-1} we get $m' = 2^{2x-2} \bmod r$. Now, since 2 is a primitive root of r , the number of valid messages is $|\mathcal{M}_p| = (r-1)/2$ because the exponent $2x-2$ yields just even numbers. That is, $\mathcal{M}_p = \{2^2 \bmod 4r, 2^4 \bmod 4r, 2^6 \bmod 4r, \dots, 2^{r-1} \bmod 4r\}$ so $|\mathcal{M}_p| = (r-1)/2$.
2. Pick up a message m_a from $\mathcal{M}_p$. Then compute c_a from the relation $m_a \cdot k_a \bmod 4r = c_a$ where k_a is an invertible integer in $\mathbb{Z}_{4r}$.

3. To determine whether c_a can be derived from every message in the set $\mathcal{M}$, consider the relation $m_i \cdot k_i \bmod 4r = c_a$ where $m_i \in \mathcal{M}$ and k_i is an invertible integer in $\mathbb{Z}_{4r}$. Since $m_i = 2^{2x_i} \bmod 4r$, then $2^{2x_i} \cdot k_i \bmod 4r = c_a$. Therefore $2^{2x_i-2} \cdot k'_i \bmod r = c_a \cdot 2^{-2}$, then $k'_i = c_a \cdot 2^{-2} \cdot (2^{2x_i-2})^{-1} \bmod r$ and $k_i = 4k'_i \bmod 4r$. The multiplicative inverse of 2^{2x_i-2} always exists because in $\mathbb{Z}_r$ all the integers are invertibles.

From the last discussion, we realize that given a ciphertext it can be derived from every valid message in $\mathcal{M}_p$. In addition, we have $|\mathcal{M}_p| = |\mathcal{C}_p| = (r-1)/2$. From here $\Pr[M=m] = \Pr[C=c] = 2(r-1)^{-1}$. As stated before, a ciphertext c can be derived from every message m in P_p , thus we have $\Pr[M=m | C=c] = \Pr[M=m]$.

An equivalent formulation of indistinguishability is demonstrated that given two or more messages from $\mathcal{M}$ the distribution over the ciphertext they produce is the same [33]. In other words, for every message in $\mathcal{M}_p$ it can be derived the set of integers in $\mathcal{C}_p$. Symbolically, we must prove that given $m_a \in \mathcal{M}_p$ it holds that $m_a \cdot k_i \bmod 4r = c_i$ for every $c_i \in \mathcal{C}_p$. Since k_i is an invertible integer in $\mathbb{Z}_{4r}$, thus we have $m_a = c_i \cdot k_i^{-1} \bmod 4r$.

As an example, consider $p = q = 2, r = 13$ (2 is a primitive root of 13), thus $n = 4r = 52$, $|k_i| = \phi(4r) = 24$, so there are 24 invertible integers and $52 - 24 = 28$ non-invertible integers in $\mathbb{Z}_{52}$. From a given ciphertext c_a , it can be obtained every message m_i in $\mathbb{Z}_{52}$. Conversely, given a message m_a it can be obtained every ciphertext c_i in $\mathbb{Z}_{52}$. Now, let us apply the same procedure to the second component of the public key $P_q = 2^{2r-1-x}k \bmod 4r$:

1. Let $\mathcal{M}_q$ be the set of valid messages for P_q . If we take $m \cdot k \bmod 4r$, $m = 2^{2r-1-x} \bmod 4r$ such that $x < 2r-1$. From here $m' = 2^{2r-3-x} \bmod r$. Once again, since 2 is a primitive root of r and taking x from $\{2r-1, 2r-2, \dots, r\}$ we get that the set of valid messages is $\mathcal{M}_q = \{2^{2r-1} \bmod 4r, 2^{2r-2} \bmod 4r, 2^{2r-3} \bmod 4r, \dots, 2^r \bmod 4r\}$ so $|\mathcal{M}_q| = r-1$.
2. Take a message m_a from $\mathcal{M}_q$, compute c_a from $m_a \cdot k_a \bmod 4r = c_a$ where k_a is an invertible integer in $\mathbb{Z}_{4r}$.
3. To verify that c_a can be derived from every message in the set $\mathcal{M}_q$ use $m_i \cdot k_i \bmod 4r = c_a$ where $m_i \in \mathcal{M}_q$ and k_i is an invertible integer in $\mathbb{Z}_{4r}$. Since $m_i = 2^{2r-1-x_i} \bmod 4r$ we can write the relation as $2^{2r-1-x_i} \cdot k_i \bmod 4r = c_a$. Multiplying both sides by 2^{-1} we have $2^{2r-3-x_i} \cdot k'_i \bmod r = c_a \cdot 2^{-2}$, then $k'_i = c_a \cdot 2^{-2} \cdot (2^{2r-3-x_i})^{-1} \bmod r$ and $k_i = 4k'_i \bmod 4r$. The inverse of 2^{2r-3-x_i} is guaranteed to exist because in $\mathbb{Z}_r$ all the integers are invertibles.

As it happened before, $\Pr[M=m] = \Pr[C=c] = (r-1)^{-1}$. Similarly, it can be derived every message m in P_q from a given a ciphertext c , thus we have $\Pr[M=m | C=c] = \Pr[M=m]$. We conclude that both components of the public key achieve indistinguishability.

6.2 Indistinguishability of the ciphertext

As stated before, the ciphersystem works under the following relations:

$$\begin{aligned} c &= m \cdot k_r \bmod r \\ m &= c \cdot k_r^{-1} \bmod r \end{aligned}$$

where m and k_r are integers in $\mathbb{Z}_r$. It can be written $m \in \mathbb{Z}_r$ where $\mathbb{Z}_r = \{0, 1, \dots, r-1\}$. However, the multiplication $c = m \cdot k_r \bmod r$ produces a permutation of the integers in $\mathbb{Z}_r$ because r is an integer prime, and hence, we have $c \in \mathbb{Z}_r$. As a result, we get that $\Pr[M=m] = \Pr[C=c] = (r-1)^{-1}$. In addition, from a ciphertext c it can be derived every message m in $\mathbb{Z}_r$, thus we have $\Pr[M=m | C=c] = \Pr[M=m]$ that accomplishes perfect secrecy.

6.3 Non-invertibility

We discuss if a message could be factored from any other message revealed by the protocol. After the public keys have been published, the key exchange protocol (KEP) requires the exchange of just two messages (in contrast to previous protocols discussed in Sects. 2 and 3 which require three messages Figs. 1 and 2). As a consequence, the protocol makes infeasible to perform factorization, because no message is a prefix of the other. It implies, symbolically, that $p^{x_a x_b} q^{y_a y_b} k_a$ is not a prefix of $p^{x_a x_b} q^{y_a y_b} k_b$ or vice versa. The integers exchanged across the public channel are:

$$\begin{aligned} T_{ab} &= p^{x_a x_b} q^{y_a y_b} k_b \bmod pqr \\ T_{ba} &= p^{x_a x_b} q^{y_a y_b} k_a \bmod pqr \end{aligned}$$

If T_{ab} were an invertible integer Alice could obtain T_{ab}^{-1} that is $\{p^{x_a x_b} q^{y_a y_b}\}^{-1} k_b^{-1} \bmod pqr$, from here she could multiply T_{ab}^{-1} with T_{ba} that is $\{p^{x_a x_b} q^{y_a y_b}\}^{-1} k_b^{-1} \cdot p^{x_a x_b} q^{y_a y_b} k_a \equiv k_b^{-1} \cdot k_a \bmod pqr$. Finally, Alice could perform $k_b^{-1} \cdot k_a \cdot k_a^{-1} \equiv k_b^{-1} \bmod pqr$ then she would obtain Bob's private key k_b . However, the last procedure cannot be executed since T_{ab} is a non-invertible integer in $\mathbb{Z}_n$. Moreover T_{ba} and the public keys are all of them non-invertible integers.

Table 9 Comparison between public cryptosystems

Method	Primes are public	Encryption system	Based in mathematical conjecture	Quantum Era
RSA	No	Yes	Yes $\rightarrow p \cdot q$	No
DH	Yes	No	Yes $\rightarrow g^x \bmod p$	No
ElGamal	Yes	Yes		
Lizama	Yes	Yes	No $\rightarrow g^x \cdot k \bmod pqr$	Yes

7 Performance

In the non-invertible KEP, we define $p = q = 2$ to guarantee indistinguishability of the ciphertext and public key. This is so because the prime integer r is chosen such that 2 is a primitive root of r . Fortunately, the condition $p = q = 2$ produces acceptable lengths of the public and private keys. Since the number of invertible integers in $\mathbb{Z}_{4r}$ is $2r - 2$ because $p = q = 2$ while the number of non-invertible integers is $4r - (2r - 2) = 2r + 2$. So, the quantity of invertible and non-invertible integers inside $\mathbb{Z}_{4r}$ is of the same size order around $2r$.

Let $2^{2x} \cdot k \bmod 4r$ be Alice's public key. The length of x can be computed using the relation $2^{2x} \bmod 4r$. Therefore, the length of x is around a half of the prime size, that is $|x| \sim \frac{|r|}{2}$, where the symbol $| \cdot |$ denotes the length of the integer in bits. Similarly, using $k \bmod 4r$ we get that $|k|$ is around $|r|$.

If we choose a prime integer such that $|r| = 1024$, we can use $|x| = 512$ and $|k| = 1024$. Thus, the size of the private key $P_r = \{x, k\}$ achieves 1536 bits², while the public key $P_u = \{2^{2x} \cdot k \bmod 4r, 2^y \cdot k \bmod 4r\}$ contains 2056 bits.

To achieve secrecy in the quantum era, we would consider $|x| \geq 256$ and $|k| \geq 256$ to resist an attack by exhaustive search by means of the quantum Grover algorithm [35]. A more realistic example using a safe big prime [36] is shown in "Appendix 2." Time to generate keys is around a millisecond.

8 Comparison

Results using Sagemath [37] demonstrate that encryption and decryption of the non-invertible KEP are faster compared to RSA and ElGamal. Here, the prime number generation process is avoided using a fixed safe prime integer. In addition, Table 9 shows a comparison of Lizama's protocol

and main cryptosystems used today. To compare them, we used the following criteria: The prime integers are publicly known, the system is allowed to perform encryption/decryption, and the computational problem is supported on prime factorization or discrete logarithm. Then due to Shor's quantum algorithm, we have that DH-ElGamal and RSA are not suitable for the quantum era.

We argue that the KEP protocol achieves perfect secrecy because public keys can be derived from any valid message in $\mathbb{Z}_n$. Thus, the non-invertible KEP seems to be resistant to quantum computing.

9 Conclusions

We introduced the non-invertible key exchange protocol (KEP). This protocol shows some important properties: is public key, allows secret key establishment and performs secret communication between two remote parties. Most remarkably, since it does not rely on computational problems as integer factorization or discrete logarithm, non-invertible KEP becomes a promising candidate for the quantum era.

Lizama's protocol shows indistinguishability of the public keys and encrypted texts. It requires minimum time for encryption/decryption when is compared with the main public key algorithms as DH-ElGamal or RSA. Another advantage of this system is the well understood theoretical basis when it is compared with other algorithms for the quantum era.

Compliance with ethical standards

Conflict of interest The authors declared that they have no conflicts of interest to this work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

² y cannot be considered as part of the private key because it is derived from x and $\phi(n)$, that is $y = \phi(n) - x + 1$

Appendix 1

In this appendix, we include some useful definitions we have used to design the non-invertible key exchange protocol.

Rings in number theory

In number theory, a ring $\mathbb{A}$ is an algebraic structure where it is possible to perform addition and multiplication. $\mathbb{A}$ with addition alone is an abelian group. In the set $\mathbb{A}$, there exists the zero element which is a neutral element for addition. However, every element has an additive inverse called its negative.

An illustrative example of ring is the set of integers $\mathbb{Z}$ where the operations addition and multiplication are allowed. $\mathbb{Z}$ is an example of an infinite ring. There are also finite rings $\mathbb{Z}_n$, for example in $\mathbb{Z}_{10}$ we have that $8 \cdot 4 = 2$ because $32 \bmod 10 = 2$. This operation is called multiplication modulo n . It can be stated that $\mathbb{Z}_n$ with addition and multiplication modulo n is a ring.

Multiplicative inverse

If $\mathbb{A}$ is a ring, there may be elements in $\mathbb{A}$ which have a multiplicative inverse. Such elements are said to be invertible [38]. In the set $\mathbb{R}$ of the real numbers, every nonzero element is invertible. However, in $\mathbb{Z}$ only 1 and -1 are invertibles. Zero is never an invertible number.

The integer k_a in $\mathbb{Z}_n$ has a multiplicative inverse if and only if $\gcd(k_a, n) \equiv 1 \bmod n$. The extended Euclidean algorithm is often used to compute the multiplicative inverse of e in $\mathbb{Z}_n$ provided the inverse exists. We can write the multiplicative inverse of k_a as k_a^{-1} so that $k_a k_a^{-1} = 1$ in the modulo n . For example, 7 is the inverse of 3 because $3 \cdot 7 = 1$ in the modulo 10. We must remark that provided the inverse of an integer exists in $\mathbb{Z}_n$ then such inverse is unique.

Euler's totient function

In number theory, Euler's totient function $\phi(n)$ gives the number of invertible integers in $\mathbb{Z}_n$, that is, the integers in $\mathbb{Z}_n$ which are relatively prime to n . We would like to emphasize here is that if n is a not prime number some integers in $\mathbb{Z}_n$ are invertibles and others are not. As an example take $n = 10$, in $\mathbb{Z}_{10}$ we have the invertible integers $\{1, 3, 7, 9\}$ and the non-invertibles numbers $\{0, 2, 4, 5, 6, 8\}$, so in this case $\phi(10) = 4$.

The Euler's function counts the number of integers with multiplicative inverses. It is known by Euler's theorem that if a is a unit in $\mathbb{Z}_n$, then $a^{\phi(n)} \equiv 1 \pmod{n}$ and $a^{\phi(n)-1} \pmod{n}$ returns the multiplicative inverse of a in $\mathbb{Z}_n$ [39].

Perfect security

According to [33], an encryption scheme (Gen, Enc, Dec) over a message space $\mathcal{M}$ is perfectly secret if for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $\Pr[C = c] > 0$ it is true that $\Pr[M = m \mid C = c] = \Pr[M = m]$.

Perfect indistinguishability

Another equivalent formulation of perfect secrecy [33] formalizes that the ciphertext contains no information about the plaintext, that is, the probability distribution over $\mathcal{C}$ is independent of the plaintext. Symbolically, for every $m_0, m_1 \in \mathcal{M}$, the distributions $\mathcal{C}(m_0)$ and $\mathcal{C}(m_1)$ are identical.

Appendix 2

Sagemath

Here is an example of the non-invertible KEP using Sagemath and applying a safe integer prime [36]:

```
import random
p=2
q=2
r=1686222770908721769746635450659841
680126920916639476373065622045210360
940178205881775365048552021766854406
549679363964455739790810585384768223
6925923395276526580734744951055642828
4856474735842563033798802739331551656
2805762132773704496780030485850709496
4895378080167642953566070121184958643
4652343660783933231
n=4*r
f=euler_phi(n)
xa=random.getrandbits(300)
xb=random.getrandbits(300)
ya=f-xa+1
yb=f-xb+1
ts=cputime()
ka=ZZ.random_element(n)
while gcd(ka,n) != 1:
ka=ZZ.random_element(n)
cputime(ts)
kb=ZZ.random_element(n)
while gcd(kb,n) != 1:
kb=ZZ.random_element(n)
Papre=power_mod(p,2*xa,n)
Pap=mod(Papre*ka,n)
Paqre=power_mod(p,ya,n)
```

```

Paq=mod(Paqr*ka,n)
Pbpre=power_mod(p,2*xb,n)
Pbp=mod(Pbpre*kb,n)
Pbqr=power_mod(p,yb,n)
Pbq=mod(Pbqr*kb,n)
kub=power_mod(Pap,xb,n)*power_mod(Paq,yb,n)
kua=power_mod(Pbp,xa,n)*power_mod(Pbq,ya,n)
ksa=Integer(kub*inverse_mod(ka,n))
ksb=Integer(kua*inverse_mod(kb,n))
kr=ksa/4
Ma=ZZ.random_element(r)
C=mod(Ma*kr,r)
Mb=C*inverse_mod(Integer(kr),r)
Time to generate ka 0.000557000000015
Time to generate Pua 0.001949
Time to encrypt 0.0002600000000026

```

```

ka=479351571554246382607035953516495276
256284692806766779542594493710796144484
836267534122175838993984551889869643385
687952825184675359826672066813985539069
333222704138576507904087177573398726571
887612804318005740873810954065809532546
766331568129430875015963415078707122213
254541779711399007111477127220974898475
kb=39880841418626545649895503195996734
725939836802083040334083431504459330799
235471882446249744097329166050067262726
825863437815738284832874522027696038033
029151370669318137960844116931201202959
088514932504797237842821543310830669719
797377443090542649815256563914025099829
2389020765225220483084104253887118354311
ksa=1586973641296677855761844562081647163898
4742208777647694589481552447542573782823364486
4915706909540739655079287908008955858680021751
9522510535317995934014864949489787166868225111
5165009996279538826716713595704015975145714674
7319821251516367069899129468016244700471346457
923154622726710623244830242896538535052

```

References

- Barreno MA (2002) The future of cryptography under quantum computers. Dartmouth College Computer Science Technical Reports
- Bennett CH, Brassard G (1984) Quantum cryptography: public key distribution and coin tossing. In: Conf. on Computers, Systems and Signal Processing, Bangalore, India, pp 175–186
- Lizama-Pérez LA, López JM, López EDC (2018) Quantum flows for secret key distribution. In: Advanced technologies of quantum key distribution, p 37
- CENTER CSR (2018) Post-Quantum Cryptography Standardization Conference, <https://csrc.nist.gov/projects/post-quantum-cryptography/>. Accessed 2020/05/09
- Lizama-Pérez LA (2019) Digital signatures over hash-entangled chains. SN Appl Sci 1(12):1568
- Lizama-Pérez LA, Montiel-Arrieta LJ, Hernández-Mendoza FS, Lizama-Servín LA, Eric S-A (2019) Public hash signature for mobile network devices. Ingeniería, Investigación y Tecnología XX(2):1–10
- Bernstein DJ, Buchmann J, Dahmen E (2009) Post-quantum cryptography. Springer, p 245
- Diffie W, Hellman M (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654
- ElGamal T (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory 31(4):469–472
- Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126
- Rabin MO (1979) Digitalized signatures and public-key functions as intractable as factorization. Tech. Rep., Massachusetts Inst of Tech Cambridge Lab for Computer Science
- Miller VS (1985) Use of elliptic curves in cryptography. In: Conference on the theory and application of cryptographic techniques. Springer, New York, pp 417–426
- Koblitz N (1987) Elliptic curve cryptosystems. Math Comput 48(177):203–209
- Alvarez BR (2004) Avances en criptología y seguridad de la información. Ediciones Díaz de Santos
- Lizama-Pérez LA, López JM, De Carlos-López E, Venegas-Andraca SE (2014) Quantum flows for secret key distribution in the presence of the photon number splitting attack. Entropy 16(6):3121–3135
- Lizama-Pérez LA, López JM, De Carlos-López E (2016) Quantum key distribution in the presence of the intercept-resend with faked states attack. Entropy 19(1):4
- Bernstein DJ, Lange T (2017) Post-quantum cryptography. Nature 549(7671):188–194
- Chen L, Chen L, Jordan S, Liu Y-K, Moody D, Peralta R, Perlner R, Smith-Tone D (2016) Report on post-quantum cryptography, vol 12. US Department of Commerce, National Institute of Standards and Technology
- McEliece RJ (1978) A public-key cryptosystem based on algebraic. Coding Thv 4244:114–116
- Ott D, Peikert C et al (2019) Identifying research challenges in post quantum cryptography migration and cryptographic agility. arXiv preprint [arXiv:1909.07353](https://arxiv.org/abs/1909.07353)
- Wang S, Zhu Y, Ma D, Feng R (2014) Lattice-based key exchange on small integer solution problem. Sci China Inf Sci 57(11):1–12
- Criptoanálisis del protocolo de intercambio de claves basado en celosía de wang et al Perspectives in Science
- Jao D, De Feo L (2011) Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: International workshop on post-quantum cryptography. Springer, New York, pp 19–34
- Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: 35th annual symposium on foundations of computer science, 1994 Proceedings, IEEE, pp 124–134
- Lamport L (1979) Constructing digital signatures from a one-way function. Tech. Rep., Technical Report CSL-98, SRI International Palo Alto

26. Merkle RC (1982) Method of providing digital signatures, US Patent 4,309,569
27. Goldwasser S, Micali S, Rackoff C (1989) The knowledge complexity of interactive proof systems. *SIAM J Comput* 18(1):186–208
28. Ben-Sasson E, Bentov I, Horesh Y, Riabzev M (2018) Scalable, transparent, and post-quantum secure computational integrity. *IACR Cryptol ePrint Arch* 2018:46
29. Forouzan BA, Mukhopadhyay D (2011) *Cryptography and network security (Sie)*. McGraw-Hill Education, New York
30. Krawczyk H (1996) Skeme: a versatile secure key exchange mechanism for internet. In: *Proceedings of internet society symposium on network and distributed systems security*, IEEE, pp 114–127
31. Steiner M, Tsudik G, Waidner M (1996) Diffie–Hellman key distribution extended to group communication. In: *Proceedings of the 3rd ACM conference on computer and communications security*, pp 31–37
32. Bellare M, Canetti R, Krawczyk H (1998) A modular approach to the design and analysis of authentication and key exchange protocols. In: *Proceedings of the thirtieth annual ACM symposium on theory of computing*, pp 419–428
33. Katz J, Lindell Y (2014) *Introduction to modern cryptography*. Chapman and Hall/CRC, London
34. Ore O (1988) *Number theory and its history*. Courier Corporation, North Chelmsford
35. Grover LK (1996) A fast quantum mechanical algorithm for database search. In: *Proceedings of the twenty-eighth annual ACM symposium on theory of computing*, ACM, pp 212–219
36. Harkins D, Carrel D (1998) Rfc2409: the internet key exchange (ike)
37. Stein W (2020) SageMath. <https://www.sagemath.org/>. Accessed 21 January 2020
38. Pinter CC (2010) *A book of abstract algebra*. Courier Corporation, North Chelmsford
39. Alkam O, Osba EA (2008) On the regular elements in z_n . *Turk J Math* 32(1):31–39

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.