ISSN 2007-9737

Post-Quantum Digital Signature for the Mexican Digital Invoices by Internet

Miguel Angel León Chávez[1], Francisco Rodríguez Henríquez[2]

[1] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

[2] CINVESTAV IPN,
Departamento de Computación,
México

miguel.leon@correo.buap.mx, francisco@cs.cinvestav.mx

Abstract. This paper presents an analysis of the Post-Quantum Cryptography (PQC) digital signature algorithms accepted for the third-round of NIST Post-Quantum Cryptography Competition. The digital signature primitive is the core of the Mexican Digital Invoices by Internet (CFDI), and it is based on the RSA algorithm but with the future arrival of Quantum Computers, it is vulnerable to cryptographic attacks. This paper points out the advantages and disadvantages of the NIST candidates for their potential adoption as the new core of CFDI.

Keywords. e-government, cryptography, post-quantum cryptography.

1 Introduction

In 1976 [1] Whitfield Diffie and Martin Hellman (2015 Turing award winners) were the first to publicly expose the basic concepts of Public-Key Cryptography (PKC) and a key-exchange protocol.

In PKC, each user has two keys, one secret or private and the other publicly known by all the users in the system. Both keys are mathematically related in such a way that knowing both, the public key and the encryption algorithm, it is computationally infeasible to obtain the private key; but it is computationally easy to encrypt/decrypt the plaintext using the keys, either of the two keys can be used for encryption while the other is used for decryption. PKC is based on computational problems that are difficult to solve, i.e., the algorithm solving the problem runs in a non-polynomial time.

Two illustrious examples that have resisted the test of time are the integer factorization problem (IFP) and the discrete logarithm problem (DLP).

In 1977 [2] Ron Rivest, Adi Shamir and Len Adleman (2002 Turing award winners) proposed to use the exponentiation in a mathematical finite (Galois) field over integers modulo a large prime number; where the exponentiation is easy to compute but the integer factorization is very difficult to compute. Such proposal is known as the RSA Algorithm and it has been the de facto standard of the PKC during the last four decades.

PKC together with Hash functions allow to implement digital signatures, which is the electronic analogous of the traditional handwritten signature, where the signer agrees to respect a contract and his/her signature is valid for legal purposes.

Roughly speaking, a Hash function is a function that for any input text of arbitrary length produces an output of fixed length in bits, known as the hash code or digest.

In 1994, Peter Shor [3] published two polynomial-time algorithms for solving both the IFP and the DLP on a quantum computer. Therefore, once that such large quantum computers become available, the security of all applications based on these problems will be vulnerable, e.g., the RSA digital signature, Digital Signature Algorithm

(DSA), and Elliptic Curve DSA (ECDSA), such as it has been shown in [4].

In 1996, Lov Grover [5] published a quantum algorithm to speed up the search in a database, and this algorithm can be used to attack the symmetric cryptographic algorithms, i.e., where two users share a secret key to encrypt/decrypt messages.

In Mexico, the RSA digital signature is the core of the digital invoices by Internet (CFDI) specified by the Secretary of Finance and Public Credit (SHCP) in the Annex 20 of the fiscal miscellany resolution published in the official journal of the federation in the years 2006, 2010, 2012, 2014, and 2017 [6].

The Annex 20 is a computing standard that specifies the structure, form, syntax, format, and cryptographic algorithms of the digital invoices issued electronically. The security requirements of the digital invoices by Internet (CFDI) are as follows: unforgeability, uniqueness, protection against modifications, protection against non-repudiation, and long-term storage. These requirements are met by means of two cryptographic digital signatures: one generated by the signer taxpayer, named digital stamp, and the other generated by the Tax Administration System (SAT), named SAT's stamp.

SAT has deployed one of the largest Public Key Infrastructure (PKI) in the world, where SAT is a Certification Authority that creates and signs two kinds of digital certificates for all the taxpayers in the country. One certificate for fiscal management purposes, named e-firma, and the other certificate for invoicing purposes, named digital stamp certificate. Both digital certificates comply with the X.509 [7] standard, and allow the widespread distribution of the taxpayers' public keys. A taxpayer is responsible for both keeping secret his/her private key and for the long-term storage of his/her issued digital invoices. SAT provides the software named *Certifica*, which permits to generate the taxpayer's private key and a requirements file that is verified at the SAT's office at the time of generation the digital certificate, which will be valid during a time interval, usually four years long.

Annex 20 [6] specifies the usage of the following cryptographic functions for signing and verifying the CFDI:

– SHA-2 256, it is a Hash function producing a digest of 256 bits length.
– RSAPrivateEncrypt, it is the RSA function that uses the signer's taxpayer private key to encrypt the digest of an invoice for generating the signature.
– RSAPublicDecrypt, it is the RSA function that uses the signer's taxpayer public key to decrypt the signature of an invoice in order to verifying the signature.

Section III.A of the Annex 20 specifies the usage of RSA-2048 bits with the Hash function SHA2-256.

From 2004 to 2020, SAT has generated 29 million 458 thousand 798 digital stamp certificates, and 16 million 405 thousand 29 taxpayers have completed the registration process for the e-firma. In 2020, 7.798 billion CFDI were issued in Mexico, i.e., an average of 247 invoices were processed per second [8]. From these, the Authorized Certification Providers (PAC) signed 96%. Notice that a PAC is a private service provider who receives from SAT both a SAT's private key and a SAT's certificate in order to generate the SAT's stamp.

In [9, 10, 11], the authors have identified some security vulnerabilities of the CFDI. Besides these vulnerabilities, now the digital signature based on RSA algorithm is vulnerable to attacks launched from potential large quantum computers.

On the other hand, in 2016 the National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce published a request for nominations for Public-Key Post-Quantum Cryptographic (PQC) Algorithms [12] i.e., quantum-resistant and classical-resistant algorithms, that will be standardized and augmented to Digital Signature Standard (NIST 186-4) as well as recommendations for Pair-Wise Key Establishment Schemes using both DLP (NIST 800-56A) and IFP (NIST 800-56B).

In 2017, 82 candidate algorithms were submitted to the NIST Post-Quantum Cryptography Competition. Among those, 69 were accepted as the first-round candidates. In 2019 [13] 26 algorithms were accepted as the second-round candidates, where 17 candidates for public-key encryption and key-establishment algorithms,

and 9 candidates for digital signatures were considered.

On July 22, 2020, NIST published the status report on the second-round candidates [14] and it announced three third-round finalists as well as three alternate candidate algorithms for standardization at the end of the third round in the spring or summer of 2021.

The PQC digital signatures accepted in the third-round candidates are as follows: Crystals-Dilithium, Falcon (Fast-Fourier Lattice-based Compact Signature over NTRU), and Rainbow. In addition, GeMSS (Great Multivariate Short Signature), Picnic, and SPHINCS$^+$ are considered as alternate candidates.

This paper analyses the PQC proposals accepted in the third-round candidates as digital signatures to generate the CFDI and points out their advantages and disadvantages. The rest of the paper is organized as follows: section 2 identifies the CFDI requirements. Section 3 briefly describes the hard computational problems that are at the core of the three PQC digital signature finalists. Section 4 presents the performance of the PQC digital signatures proposals reported by their respective authors. Section 5 presents the performance reported by the project eBATS. Finally, the conclusions and the future research work are drawn.

2 CFDI Requirements

This section presents an analysis of the functional, temporal, and security requirements of the CFDI based on public key cryptography. The analysis is developed according to the processes involved in the classical digital signature algorithms, i.e., key generation, signing, transfer, verifying, and storage.

2.1 CFDI Key Generation

The taxpayer's key generation process is as follows:

1. The user downloads the Certifica software from the SAT web page.
2. He/She generates his/her private key, its password, and the requirements file. The private key complies with the PKCS8 standard and it is stored in a file with format DER.
3. At the SAT office, the public key and the digital stamp certificate are generated. The public key complies with the PKCS10 standard and it is stored in a file with format DER.

It must be noted that the RSA algorithm produces a public key composed of both the product of two large prime numbers and an exponent. Thus, the public key length of RSA-2048 is 256 bytes and the value of the exponent is usually 65537.

4. The certificate complies with the X.509 standard and it has several fields, such as certificate serial number, signature algorithm identifier, period of validity, subject's public key information, and the signature of the Certification Authority.
5. The taxpayer's private key and his/her certificate are stored in the taxpayer's USB memory.
6. The taxpayer's certificate is stored at the SAT office.

As was pointed out in the previous section, SAT is the Certification Authority that creates and signs (using the RSA digital signature algorithm) the taxpayers' certificates.

1. Both the private key length in bytes and the certificate length in bytes are transparent for the taxpayers.
2. The certificate length in bytes depends on the public key length. Larger public key length will require larger storage capacity at the SAT office.
3. The usage of a PQC digital signature will have no impact on the X.509 since it has the fields to identify a new algorithm.
4. The certificate security strength depends on the digital signature algorithm security strength. The security level or strength, in bits, is defined as the number of operations required to break the algorithm.
5. The key generation process has no time restrictions.

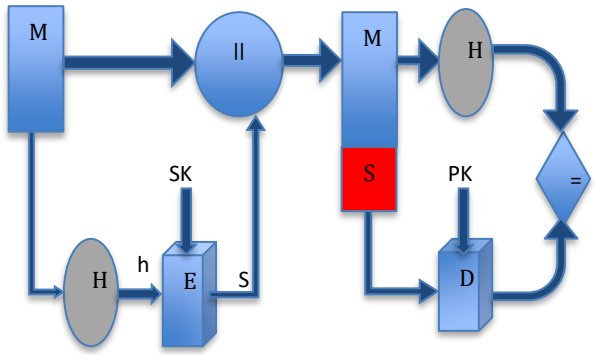

Fig. 1. Digital signature scheme

2.2 CFDI Signing

Figure 1 shows the basic scheme of the digital signature. During the signing process the signer ciphers (E), using both his/her private key (SK) and the asymmetric encryption algorithm, the hash value (h) of a message (M) produced by a Hash function (H). The signature (S) is concatenated (||) to the message and both are sent to the receiver or verifier.

In the CFDI case, all the accounting information of the tax receipt, i.e., M, is first transformed in an original chain (OC) and is represented in the encoding format UTF-8 before its hash value is calculated. Also, the signature is represented in Base 64 and is concatenated with both the OC and M.

It can be noted that the security strength of the digital signature algorithm depends on both the security level of the asymmetric cipher/decipher algorithm and the security level of the hash function, and both security levels must be equals, otherwise the security strength of the digital signature algorithm will be the lower of the two.

As it is the case of the CFDI where SHA-256 provides a security level of 128 bits and RSA-2048 provides a security level of 112 bits, according to the NIST SP 800-57 standard.

The future usage of a PQC digital signature in the CFDI requires a fast signing process able to sign at least 250 times per second and providing a security level of at least 128 bits.

2.3 CFDI Transfer

It is highly recommended to use a secure communication channel between the signer and the receiver, such as TLS.

2.4 CFDI Verifying

The verifier, i.e., SAT or an Authorized Certification Provider (PAC), will receive the accounting information of the tax receipt, the original chain, and digital stamp of the taxpayer.

Either SAT or PAC after verifies the received information, calculates the hash value of the OC and compares it against the deciphering (D) of the signature using the taxpayer's public key (PK) from his/her certificate. If both hash values are equal, the signature is accepted otherwise it is rejected.

Once accepted, either SAT or PAC signs again the original chain using the SAT's private key, and generates the CFDI with additional information such as fiscal folio number, both serial numbers of the SAT certificate and taxpayer certificate, date, time, and a QR code (Quick Response Code).

The CFDI is transmitted then to the taxpayer signer, receiver, and the SAT office.

Again, a new PQC digital signature in the CFDI requires a fast verification process able to verifies and sign at least 250 times per second.

2.5 CFDI Storage

At any time, during the storage process, a third party, usually the SAT, can verify the signature to resolve disagreements between the parties. According to the Annex 20, the taxpayer is responsible for both keeping secret his/her private key and for the long-term storage of his/her issued digital invoices for at least 5 years.

It can be noted that the reliability of the digital signature depends not only on the generation and verification processes but also on the transfer and storage processes. Any vulnerability on one process will affect all the security services. In addition, since the X.509 standard specifies a valid period for the digital certificates, SAT must guarantee the availability of the Certificate Revocation List (CRL) during the storage process. Table 1 resumes the CDFI requirements, where x means without restrictions.

Table 1. CFDI requirements

CFDI	Requirement
Digital signature security strength	□ 128 bits
Private key length	x
Public key length	SMALL
Signature length	SMALL
Keys generation time	x
Signing time	LOW
Verification time	LOW
Storage time	□ 5 years
X.509	comply

3 PQC

Even if the IFP and the DLP are vulnerable to large quantum computers, there are other computing problems difficult to solve using quantum and classical computers, such as [15]: Secret-key cryptography, Hash-based, Lattice-based, Multivariate-polynomial, and Code-based.

From the PQC digital signatures accepted in the third-round candidates both Crystals-Dilithium and Falcon are lattice-based cryptographic schemes, whereas Rainbow is Multivariate based. From the NIST point of view, these schemes appear to be the most promising general-purpose algorithms for digital signature algorithms. Nevertheless, NIST believes it is advisable to continue to study other schemes against progress in cryptanalysis. Lattice and Multivariate problems are briefly described below.

Lattice-based cryptographic schemes are based on the difficulty of solving several related problems such as the shortest vector problem, and the closest vector problem. Given n linearly independent vectors b_i defined over R^n, a lattice is defined as any linear combination with integer coefficients of such basis as in eq. (1):

$$L(b_0, \ldots, b_{n-1}) = \Sigma^{n-1}{}_{i=0}\, x_i b_i : x_i \in Z. \qquad (1)$$

The shortest vector problem consists of finding the shortest vector in the lattice considering its Euclidean norm. The fastest known solution takes $2^{O(n)}$ time and space. There also exist polynomial-space algorithm that takes $2^{O(n\log(n))}$ time. The lattice-based NIST candidate algorithms, such as Crystals-Dilithium and Falcon, exploit what is called structured lattice schemes, which allows them to achieve considerably efficiency for signing and verifying at the price of potential losses in security that must still be explored.

Multivariate-polynomial cryptographic, such as HFE^{v-}, Rainbow, and GeMSS, are based on multivariate polynomial problem over finite fields. The public key is the polynomial sequence P_1, P_2, …, $P_{2b} \in F_2[w_1,..w_{4b}]$, where b is the desired security level in bits, and where the $4b$ variables $w_1,\ldots,w_{4b}$ have coefficients in F_2.

Each polynomial is square-free of degree two and it's represented as a sequence of $1 + 4b + 4b(4b-1)/2$ bits. The public key is large, it has $16b^3 + 4b^2 + 2$ bits. However, the signature of a message is short, it has only $6b$ bits, namely, the $4b$ bits of the variables $w_1,..w_{4b} \in F_2$, and an $2b$-bit string r, satisfying eq. (2):

$$H(r, m) = (P_1(w_1,..w_{4b}), \ldots, P_{2b}(w_1,..w_{4b})), \qquad (2)$$

where H is a standard hash function. The verification process is simple as it requires b^3 bit operations to evaluate the polynomials P_1, …, P_{2b}. The main advantage of this crypto scheme is that the signature is small with respect to all the other NIST post-quantum candidates. The security of this scheme lies on the difficulty of finding the sequence $w_1,..w_{4b} \in F_2$ such that the polynomials P_1, …, P_{2b} can be produced. Using a brute-force approach, the probability of finding such sequence is 2^{-2b}. On the contrary, the signee takes advantage of a predefined structure for generating the polynomials. This problem is known as the Hidden Field Equation (HFE). It is possible that an attacker can also take advantage of this structure, although so far nobody has been able to find such line of attack.

4 PQC Digital Signatures Performance

NIST defined a collection of security strength categories [11] as follows, listed in order of increasing strength:

1. Any attack that breaks the relevant security definition must require computational

Table 2. Length in bytes (or kilobytes (kB) or bits) of the Public Key (PK), Secret Key (SK), and Signature (S) reported by the authors

Digital Signature		AES 128	SHA3 256	AES 192	AES 256
Crystals-Dilithium	PK		1312	1952	2592
	SK		32	32	23
	S		2420	3293	4595
Falcon	PK	897			1793
	SK	32			32
	S	666			1280
Rainbow classical	PK	157.8 kB		861.4 kB	1885.4 kB
	SK	101.2kB		611.3 kB	1375.7 kB
	S	528 bits		1312 bits	1632 bits
CZ-Rainbow	PK	58.8 kB		258.4 kB	523.6 kB
	SK	101.2 kB		611.3 kB	1375.7 kB
	S	528 bits		1312 bits	1696 bits
Rainbow compressed	PK	58.8 kB		258.4 kB	523.6 kB
	SK	0.06 kB		0.06 kB	0.06 kB
	S	528 bits		1312 bits	1696 bits

resources comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g., AES128).

2. Any attack for collision search on a 256-bit hash function (e.g., SHA256/SHA3-256).
3. Any attack … for key search on a block cipher with a 192-bit key (e.g., AES192)
4. Any attack … for collision search on a 384-bit hash function (e.g., SHA384/SHA3-384).
5. Any attack … for key search on a block cipher with a 256-bit key (e.g., AES256).

Table 2 summarizes the performance metrics of the three candidates accepted in the third-round of the NIST competition in terms of both the security strength and the length in Bytes (or kilo Bytes (kB) or bits) of the Public key (PK), Private key (SK), and Signature (S) reported by the Authors in their proposals.

4.1 Remarks

1. Crystals-Dilithium is proposed to provide the security strength 2, 3, and 5.

   The secret key is a set of parameters, but the signer has the option of storing a 32-byte value and then re-deriving all the other elements of the secret key.

   The authors proposed a variant called Dilithium-AES that uses AES-256 in counter mode instead of SHAKE to both expand the matrix and the masking vectors, and to sample the secret polynomials.

2. Falcon provides the security levels 1 and 5. The secret key values are not reported in the third-round proposal, but their values are about three times that of a signature. According to the authors, the secret key value could be compressed down to a small PRNG seed of 32 bytes.
3. The authors of Rainbow propose three variants named classical, CZ (circumzenithal), and compressed for the security level 1, 3, and 5, respectively.

The cyclic Rainbow allows reducing the public key size of the classical scheme by up to 70% at a higher cost of signature verification. The compressed Rainbow stores the private key in the

form of a 512 bit seed, thus enabling storing the private key easily on small devices at the cost of the efficiency of the signature generation process.

The only difference between the compressed Rainbow and the cyclic Rainbow is the use of the PRNG function during the signature process.

Finally, it can be noted in Table 2 that no scheme proposes the security level 4, i.e., resistant to collision search on a 384-bit hash function.

4.2 PQC Computational Efficiency

The computational efficiency is measured in terms of clock cycles or time in milliseconds of the key generation, signing, and signature verification processes reported by the authors. The key generation process takes into account the time to generate the key pair, i.e., the secret key and the corresponding public key.

Even if NIST proposed to test the submissions on a PQC Reference Platform, an Intel x64 running Windows or Linux and supporting GCC compiler, each author measured the computational efficiency on his own platform, as follows:

1. Crystals-Dilithium on an Intel Core-i7 6600U CPU at 2600 MHz using SHAKE as the XOF.

   The values of the cycles reported are the medians of 1000 execution each, and signing a text of 32 bytes length. The authors reported also optimized versions based on AVX2, and AVX2+ AES.

2. Falcon on an Intel Core-i5 8259U CPU at 2.3 GHz. The authors do not indicate the length of the signing text.

3. Rainbow on an Intel(R) Xeon(R) CPU E3-1225 v5 at 3.60 GHz. The reported values do not indicate the length of the signing text.

It can be noted that it is not possible to compare the computational efficiency of the proposals, in terms of cycles or time, since each was implemented in different CPUs at different frequencies. Fortunately, there exist the effort of multiple researchers to develop a fair evaluation, as it is discussed in the next section.

5 eBATS PQC Digital Signatures Performance

Two approaches to develop a fair evaluation of the PQC digital signatures proposals have been published: Implemented in hardware and hardware/software co-design [16] and software executing on multiple platforms [17]. This work reports the second approach.

ECRYT Benchmarking of Asymmetric Systems (eBATS) [17] is a project to measure the performance of the public-key systems. It is based on the SUPERCOP's work, which is a toolkit for measuring the performance of cryptographic software. With regard to the public-key signature algorithms it measures the cryptographic primitives according to different criteria, amongst they the following: Time (cycles) to generate a key pair (a secret key and the public key); time to sign a short message (59 bytes length); time to open a signed short message, i.e., to verify a (larger) signed message and recover the original short message.

eBATS has measured the PQC digital signature performance in both multiple CPU architectures and in multiple computers ranking from 1 up to 64 CPU cores at different frequencies. This work presents the PQC digital signature performance as reported by eBATS in two platforms, as follows: a slow computer using the Intel Core i3-2310M (2x2100 MHz), because most of the individual taxpayers in Mexico do not have a high performance computer, and a fast computer using AMD EPYC 7742 (64x2250 MHz) because the SAT and the most of the PACs have a higher computational infrastructure.

Tables 3 and 4 present the computational efficiency of the first level of security (AES128) in terms of cycles (median) of the following processes: key generation, signing of 59 bytes, and signature verification of 59 bytes reported by eBATS on an Intel Core i3-2310M (2x2100 MHz) and on an AMD EPYC 7742 (64x2250 MHz), respectively. It can be noted in Table 2 that the rows list the median of many speed measurements. But measurements with large variance are indicated with question mark (?). The computational efficiency for the security levels 3 and 5 can be consulted in [17] as well as for others CPU.

Table 3. Computational Efficiency in Cycles (Median) of the Key Generation (Key gen), Sign (S), and Verify (V) processes of 59 bytes on an Intel Core i3-2310M (2x2100 MHz) reported by eBATS for the security level 1 (AES128), except for Dilithium that is level 2

Digital Signature (AES128)	Key Gen	Sign	Verify
Dilithium-2	599972	2497172?	639972
Falcon-512 dyn	39969832?	1974036	188796
Rainbow-1a classic	27348024	252056	69336
Rainbow-1a compressed	28416812	20566500	13208016
Rainbow-1a cyclic	28281772	256664	13104888

Table 4. Computational Efficiency in Cycles (Median) of the Key Generation (Key gen), Sign (S), and Verify (V) processes of 59 bytes on an AMD EPYC 7742: 64x2250 MHz reported by eBATS for the security level 1 (AES128), except for Dilithium that is level 2

Digital Signature (AES128)	Key Gen	Sign	Verify
Dilithium-2	72045	182452?	79448
Falcon-512 dyn	14236740	555345	57600
Rainbow-1a classic	6624585	38655	21555
Rainbow-1a compressed	6991357	4638757	2458643
Rainbow-1a cyclic	7027402	39060	2483032

6 Conclusions

In the near future, the security of all applications based on either the integer factorization problem or the discrete logarithm problem, such as Mexican's digital invoices by Internet (CFDI), will be vulnerable to large quantum computers.

This paper has presented a security analysis of the post-quantum digital signature algorithms of NIST third-round candidates based on both the performance reported by the authors and a fair evaluation reported by the project eBATS.

The taxonomy reported by the authors based on publicly known information takes into account the security strength and the length of the public key, secret key, and signature. The data reported by eBATS takes into account the computational efficiency in cycles of the key generation, signing, and verification processes.

With regard to both Table 1 it can be noted the following:

1. Dilithium and Falcon provide the smallest secret key in bytes.
2. Falcon provides the smallest public keys in bytes.
3. Rainbow and its variants provide the smallest digital signatures in bits.

With regard to both Table 3 and Table 4 it can be noted the following:

1. Crystals-Dilithium combines their parameter sets to propose the security levels 2, 3, and 5. Both Tables show the performance of the security level (SHA3-256), and it can be noted that the signing process suffers of large variance in both platforms. Nevertheless, it has the best computing efficiency to compute the key generation process.
2. Rainbow classic has the best computing efficiency to sign and to verify 59 byes in both platforms. Nevertheless, the public key length is the longest of the three representing more than 600% of increase with regard of the current size using RSA.

As it can be noted, no PQC digital signature meets all the CFDI requirements identifies in Table 1. Therefore, new studies and analysis will be

required once the NIST announces the new standard. Meanwhile, the computational efficiency of the finalist will be evaluated with regard to the original chain of the CFDI, as our future research work.

References

1. **Diffie, W., Hellman, M. E. (1976).** New directions in cryptography. IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp. 644-654. DOI: 10.1109/TIT.1976.1055638.
2. **Rivest, R.L., Shamir, A., Adleman, L. (1978).** A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, Vol. 21, No. 2, pp. 120-126. DOI: 10.1145/359340.359342.
3. **Shor, P. (1997).** Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Computing, Vol. 26, No. 5, pp. 1484–1509. DOI: 10.1137/S0036144 598347011.
4. **Gidney, C., Ekerå, M. (2021).** How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. In Quantum, Vol. 5, pp. 433. DOI: 10.22331/q-2021-04-15-433.
5. **Grover, L.K. (1996).** A fast quantum mechanical algorithm for database search. ACM Symposium on the Theory of Computing, pp. 212–219. DOI: 10.1145/237814.237866.
6. **Anexo 20 SHCP (2017).** Annex 20 of the resolution of the fiscal miscellany for 2017. The Official Journal of the Federation by the Executive Power through the Ministry of Finance and Public Credit - Tax Administration Service.
7. **ITU-T Recommendation X.509 (2008).** Information technology – Open systems interconnection. The Directory: Public-key and attribute certificate frameworks.
8. **SHCP-SAT (2020).** Tax and Management Report - Fourth Trimester 2020. http://omawww.sat.gob.mx/gobmxtransparencia/Paginas/documentos/itg/ITG_2020_4T.pdf.
9. **González-García, V., Rodríguez-Henríquez, F., Cruz-Cortés, N. (2008).** On the security of Mexican digital fiscal documents. Computación y Sistemas, Vol. 12, No. 1, pp. 25−39. DOI: DOI:10.13053/CYS-12-1-1187.
10. **León-Chávez, M.A., Rodríguez-Henríquez, F. (2015).** Security vulnerabilities of the Mexican digital invoices by internet. IEEE International Conference on Computing Systems and Telematics. DOI: 10.1109/ICCSAT.2015.7362926.
11. **Domínguez-Pérez, L.J., Gómez-Trujillo, L.M., Cruz-Cortés, N., Rodríguez-Henríquez, F. (2019).** Sobre el impacto del colisionador SHA-1 en las firmas digitales mexicanas con valor legal. Computación y Sistemas, Vol. 23, No. 4, pp. 1181−1190. DOI: 10.13053/cys-23-4-3103.
12. **NIST (2016)**. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process.
13. **NISTIR 8240 (2019).** Status report of the first round of the NIST post-quantum cryptography standardization process.
14. **NISTIR 8309 (2020).** Status report of the second round of the NIST post-quantum cryptography standardization process.
15. **Bernstein, D., Buchmann, J., Dahmen, E. (2009).** Post-Quantum Cryptography. Springer.
16. **Ba Dang, V., Farahmand, F., Andrzejczak, M., Mohajerani, K., Tri Nguyen, D., Gaj, K. (2020).** Implementation and benchmarking of round 2 candidates in the NIST post-quantum cryptography standardization process using hardware and software/hardware co-design approaches. Cryptology ePrint Archive.
17. **eBATS (2021).** ECRYT Benchmarking of Asymmetric Systems. **Bernstein, D. J., Lange, T. (Ed).** eBACS: ECRYPT Benchmarking of Cryptographic Systems. https://bench.cr.yp.to.

Article received on 20/08/2020; accepted on 05/01/2021.
Corresponding author is Francisco Rodríguez Henríquez.