Algoritmos post-cuánticos: criptografía y ciberseguridad en la era cuántica

Seguridad en la era cuántica: una investigación de **NTT DATA** demuestra la complejidad computacional del algoritmo post-cuántico Crystals-Kyber, ante ciberataques que emplean ingeniería inversa y fuerza bruta.

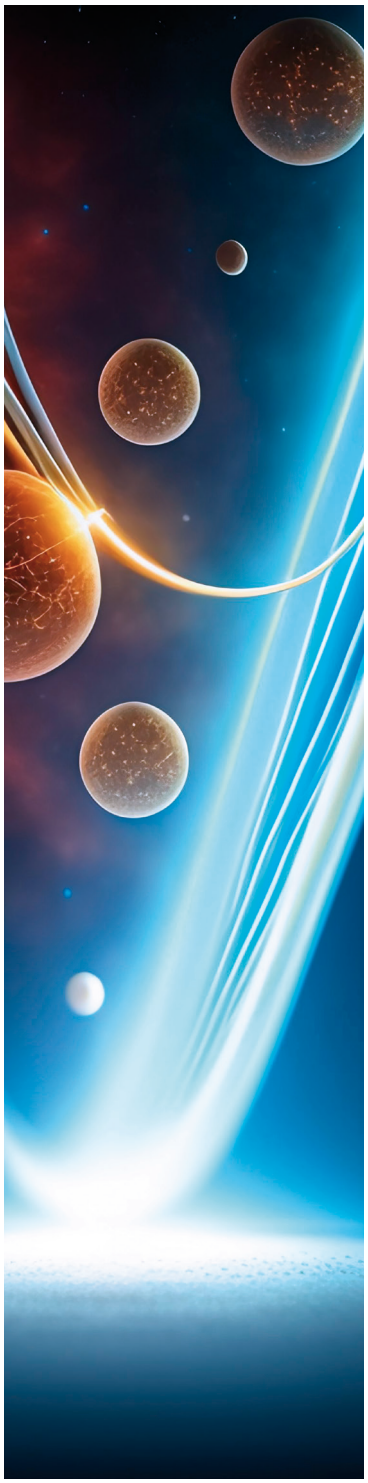
0. Resumen Ejecutivo

La computación cuántica es una tecnología con el potencial de impactar y revolucionar todas las industrias, ya que supera ampliamente ciertas limitaciones de la computación clásica. Este poder involucra también una nueva serie de desafíos: tiene la capacidad de vulnerar los métodos de criptografía tradicionales, pilares sobre los que se basó la seguridad de la información durante las últimas décadas.

NTT DATA está liderando la utilización de la computación cuántica para resolver problemas de altísima complejidad. En ese contexto, ha realizado también una investigación profunda, utilizando técnicas como la ingeniería inversa y la fuerza bruta, para analizar la complejidad computacional del algoritmo crip-tográfico post-cuántico Crystals Kyber, con el fin de garantizar el uso seguro de esta tecnología.

Este documento técnico relata los procesos de prueba y las conclusiones, que demuestran que el algoritmo en estudio es altamente robusto y muy difícil de vulnerar.

1. Introducción

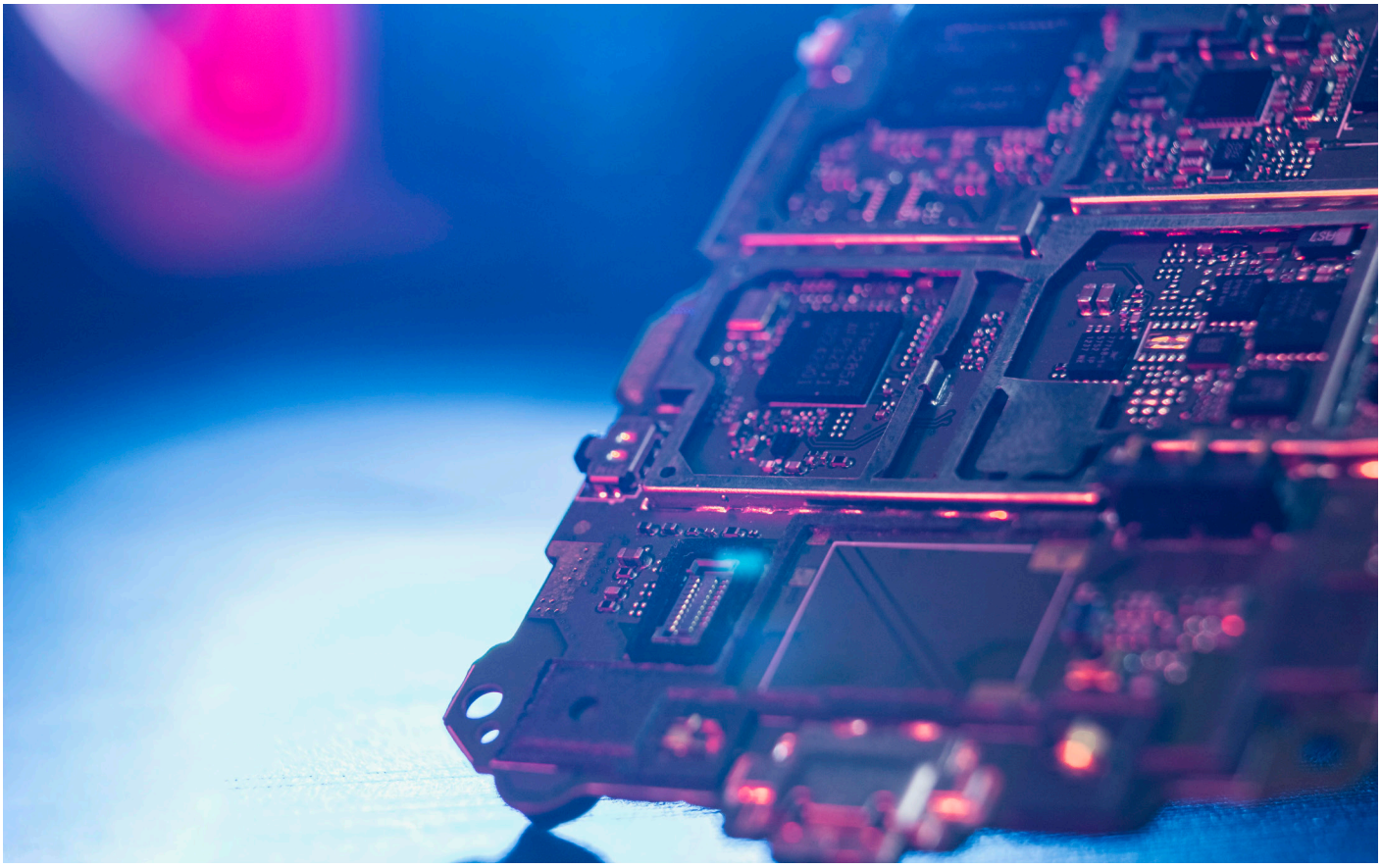


La creciente importancia de la digitalización ha forzado la necesidad de establecer canales de comunicación eficientes y seguros. Esto ha llevado a la formulación de algoritmos criptográficos avanzados para garantizar la seguridad de los datos en tránsito. RSA [1], AES [2], ECC [3], entre otros, han sido el pilar de la seguridad de los datos en las últimas décadas. Sin embargo, el surgimiento de la computación cuántica amenaza la seguridad ofrecida por métodos estandarizados de criptografía.

La computación cuántica posee el potencial de vulnerar los algoritmos criptográficos convencionales, especialmente aquellos basados en la factorización de números enteros muy grandes y logaritmos discretos [4]. En respuesta a esta amenaza, expertos en ciberseguridad han comenzado a explorar nuevas estrategias criptográficas para contrarrestar potenciales ataques cuánticos, conocido como criptografáa post-cuántica.

Entre los esquemas de criptografía post-cuántica propuestos, el algoritmo de criptografía de clave pública Crystals-Kyber (referido simplemente como Kyber en adelante) ha sido premiado entre los ganadores del concurso de criptografía post-cuántica del Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés) de Estados Unidos, y reconocido por su seguridad y velocidad de cifrado, como se puede observar en [5][6].

Al ser un potencial reemplazo de estándares criptográficos actuales, es importante poner a prueba el algoritmo de Kyber y explorar maneras de vulnerarlo. Las técnicas clásicas de hacking criptográfico, como el ataque por diccionario y el criptoanálisis diferencial [7], son ineficaces contra Kyber debido a su método de cifrado. En cambio, aplicando ingeniería inversa y fuerza bruta es posible obtener una mejor alternativa para vulnerar este algoritmo en comparación a los métodos ya existentes, con el fin de analizar la gran complejidad computacional que involucra tratar de romper con la seguridad de Kyber. Es importante mencionar que no se busca hackear el algoritmo



sino encontrar la mejor alternativa de vulnerabilidad para validar la robustez de Kyber frente a ciberataques por computadoras clásicas.

En esta investigación se implementó Kyber y se realizó un ciberataque con el método propuesto bajo una configuración de parámetros inseguros. Aunque tal configuración no proporciona una robustez suficiente para ser implementado en un escenario real, proporciona un entorno de prueba útil para entender mejor la resistencia de Kyber y para demostrar las dificultades inherentes al intento de romper su seguridad, incluso bajo condiciones favorecedoras. Además, esta investigación busca obtener una ecuación analítica que describa la complejidad computacional requerida para intentar vulnerar Kyber usando computadoras clásicas, con el fin de demostrar que el tiempo requerido para realizar un ciberataque a su cifrado sería prácticamente inviable.

> La computación cuántica posee el potencial de vulnerar los algoritmos criptográficos convencionales, especialmente aquellos basados en la factorización de números enteros muy grandes y logaritmos discretos

2. Criptografía post-cuántica: algoritmo Crystals-kyber

Kyber es un algoritmo criptográfico post-cuántico basado en el problema de Aprendizaje con Error (LWE, por sus siglas en inglés). En términos generales, LWE es un problema que involucra encontrar un vector secreto dado un conjunto de ecuaciones lineales ruidosas.

Este tipo de problemas son difíciles de resolver o vulnerar tanto para computadoras clásicas como para computadoras cuánticas [8].

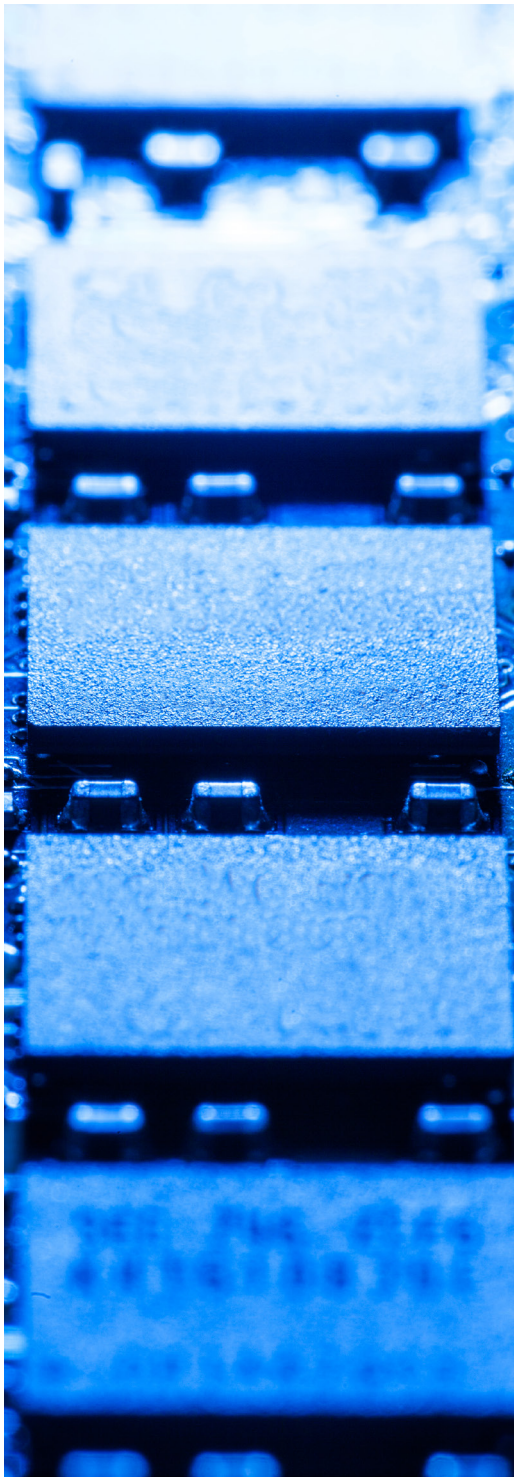
Kyber fue seleccionado entre los ganadores en la categoría de cifrado de clave pública en el concurso de criptografía post-cuántica del NIST, al demostrar alta robustez, velocidad en generación de claves, rendimiento y seguridad frente a otros candidatos [9].

Este algoritmo posee tres variantes: Kyber512, Kyber768 y Kyber1024. La numeración se refiere al tamaño de la clave de seguridad, y cada variante ofrece diferentes niveles de seguridad. Kyber 512 ofrece un nivel de seguridad semejante a AES-128, Kyber768 a AES-192, y Kyber1024 a AES-256. Sin embargo, en teoría Kyber resiste ataques cuánticos como el algoritmo de Shor o el algoritmo de Grover.

La seguridad de Kyber radica en el problema de LWE, en el cual se emplean polinomios. Esto aumenta la robustez, la seguridad, y la velocidad de cifrado frente a otras alternativas [10].

Demostrado la capacidad del algoritmo de Shor en poder vulnerar los estándares actuales de ciberseguridad, dada una computadora cuántica lo suficientemente potente y que sea susceptible a errores, es evidente que la criptografía del futuro reside en algoritmos post-cuánticos, motivo por el cual vale la pena poner a prueba y realizar estudios de estos algoritmos.

3. Metodología de Ingeniería Inversa y Fuerza Bruta



La ingeniería inversa en criptografÍa, una técnica crucial en este estudio, consiste en analizar y desmontar sistemas criptográficos para descubrir sus mecanismos y vulnerabilidades. Este proceso involucra la descompilación del código, la identificación de algoritmos clave, el análisis de posibles debilidades y la realización de pruebas para verificar y explotar las vulnerabilidades encontradas.

En el caso específico del algoritmo criptográfico de Kyber, la táctica de ingeniería inversa se enfoca en analizar la matemática de cifrado con el propósito de encontrar vulnerabilidades. El objetivo es determinar cómo se puede atacar este algoritmo mediante fuerza bruta de la manera más eficiente posible, aplicando este enfoque metodológico para descubrir puntos débiles explotables en su estructura.

La seguridad de Kyber se basa en la multiplicación y la suma de matrices y vectores conformados por polinomios en lugar de números. Tanto la clave pública como la privada son vectores que también contienen polinomios. Esta suma y producto de polinomios se relaciona matemáticamente mediante la siguiente ecuación:

$$\mathbf{As} + \mathbf{e} = \mathbf{t}$$
$$\textit{Clave pública} = (\mathbf{A}, \mathbf{t}) \qquad (1)$$
$$\textit{Clave privada} = \mathbf{s}$$

donde **A** es una matriz cuadrada conocida, **s** es un vector que representa la clave privada que contiene polinomios con coeficientes pequeños, **t** es un vector de una sola columna que contiene polinomios, y **e** es un vector de errores (i.e. polinomio de coeficientes aleatorios pequeños). La matriz

A y **t** conforman la clave pública. A continuación, se muestra un ejemplo:

$$\mathbf{A}=\begin{bmatrix} 14x^3+5x^2-x+10 & 8x^3+5x^2+x-1 \\ -3x^3+x^2+8x-5 & x^3-x^2+7x-5 \end{bmatrix}$$

$$\mathbf{s}=\begin{bmatrix} x^3-x^2-x+1 \\ x^3+x^2+x+1 \end{bmatrix} \quad (2)$$

$$\mathbf{e}=\begin{bmatrix} x^3+x^2+x+1 \\ -x^3-x^2+x-1 \end{bmatrix}$$

La complejidad detrás del LWE reside en el error **e**, lo cual dificulta hallar la clave privada **s**, de manera que ambos satisfagan (1). Kyber emplea polinomios en todo su proceso de cifrado y descifrado, y el mensaje a cifrar también debe ser convertido a un polinomio. El primer paso consta en convertir el mensaje a binario. Por ejemplo:

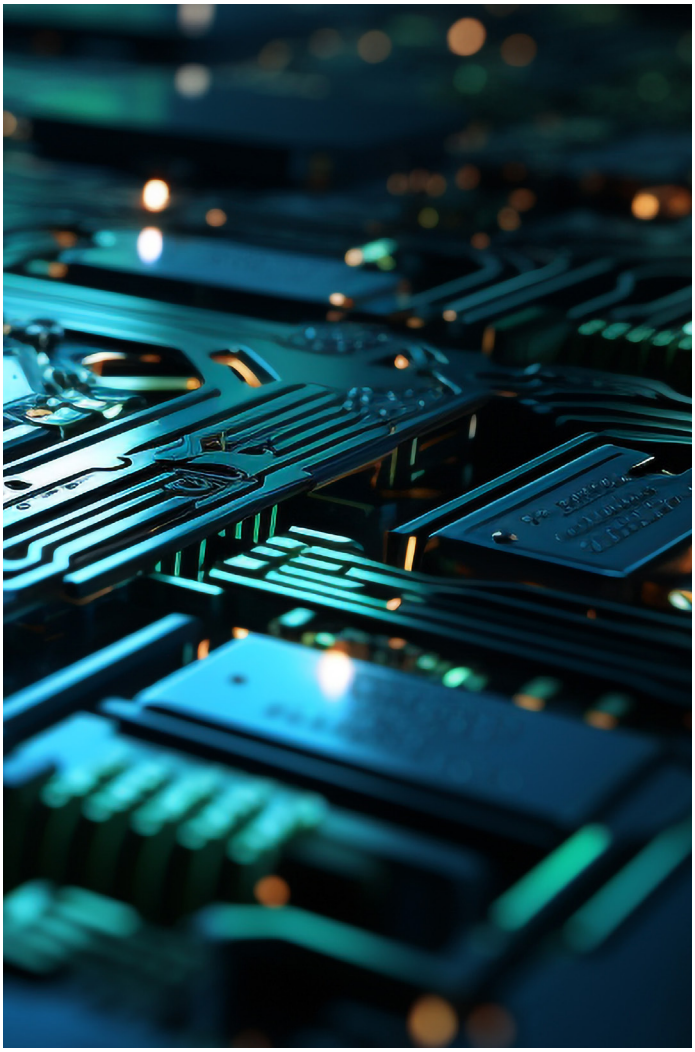
$$Mensaje = \text{"123"}$$
$$= \text{"1111011"}$$

Luego de obtener la representación binaria, se multiplica cada elemento binario por la variable que se esté empleando en los polinomios de Kyber. En el presente ejemplo, x sería la variable, la cuál se multiplica por cada número binario, a la vez que va aumentado el valor de su exponente. Al realizar este procedimiento, se obtiene el siguiente polinomio:

$$Mensaje = 1x^6+1x^5+1x^4+1x^3+0x^2+1x^1+1x^0$$
$$Mensaje = x^6+x^5+x^4+x^3+x+1 \quad (3)$$

Luego se procede a multiplicar la representación polinomial del mensaje por un escalar, como ejemplo usaremos el número 9, obteniendo:

$$m = 9(x^6+x^5+x^4+x^3+x+1)$$
$$m = 9x^6+9x^5+9x^4+9x^3+9x+9 \quad (4)$$



Para cifrar el mensaje se utiliza la clave pública, ya que Kyber se basa en un cifrado asimétrico. Parar cifrar matemáticamente el mensaje se realiza las operaciones mostradas en la siguiente ecuación: donde **u** y *v* conforman las partes del cifrado, **r**, $\mathbf{e}_1$ y $\mathbf{e}_2$ son errores adicionados.

$$\boldsymbol{u} = \boldsymbol{A}^T\boldsymbol{r} + e_1$$
$$v = \boldsymbol{t}^T\boldsymbol{r} + e_2 + m \quad (5)$$

$$Cifrado = (\mathbf{u}, v)$$

Una vez el mensaje cifrado llegue al punto destino, el receptor debe emplear la clave privada para descifrar el mensaje, utilizando:

$$m_o = v - \boldsymbol{s}^T\boldsymbol{u} \quad (6)$$

donde el resultado m_o será un polinomio, al cual se le evaluará sus coeficientes para ver qué tan cercanos son al número que hemos utilizado para escalar al mensaje en su forma polinomial.

Posterior a ello, se tendría que realizar el proceso inverso de las ecuaciones (3) y (4). Esto quiere decir que se tendrá que convertir el polinomio en su representación binaria para luego ser convertidos a texto o a números y así obtener el mensaje original.

Tratar de descifrar el mensaje sin poseer la clave privada es una tarea de alto costo computacional.

Esto se debe a que se tendría que iterar sobre todos los posibles valores de **r**, $\mathbf{e}_1$ y $\mathbf{e}_2$ en (5). Además, **r** y $\mathbf{e}_1$ son vectores de polinomios, los cuales agregan complejidad al problema.

Respecto a la clave privada, es igualmente inviable tratar de iterar sobre sus posibles valores y utilizarlos en la ecuación (6), debido a que no se conoce el valor del polinomio m_o. Tratar de probar diferentes valores de mo es irrealizable, debido a que este polinomio corresponde al mensaje original en su forma polinomial, al cual se le han agregado polinomios de error en el proceso de cifrado. El mensaje original solo es conocido por la persona que está enviando la información, lo cual lo hace mucho más complejo aún. Esto implica que se tendría que probar la conversión de todas las palabras y oraciones posibles, convertirlos en forma polinomial y adicionar diferentes polinomios de error con un tamaño totalmente aleatorio, ya que no se conoce ninguna información de los errores añadidos.

Sin embargo, tratar de hallar el valor de la clave privada podría ser un procedimiento más realista si se analiza la ecuación correcta. Al realizar un análisis matemático simple, se puede apreciar que la clave privada está embebida en la clave pública (observar la ecuación (1)). Iterar sobre los posibles valores de las variables de solo esa ecuación genera un gran ahorro computacional, ya que no se requiere realizar todos los procedimientos ya mencionados anteriormente, analizando las ecuaciones (5) y (6). Solo será necesario hallar los valores de **s** y **e**, debido a que los valores de **A** y **t** son conocidos, los cuales conforman la clave pública.

Kyber posee 3 parámetros de seguridad internos: **n** indica el grado máximo de los polinomios, **q** indica el valor del escalar que se utiliza en la conversión del mensaje en su forma polinomial y **k** indica la cantidad de elementos en los vectores y matrices.

En el modelo Kyber512 se tiene un valor de $\boldsymbol{k} = 2$, en Kyber768 el valor de $\boldsymbol{k}$ es 3 y en Kyber1024 el valor de $\boldsymbol{k}$ es 4. La complejidad para hallar los valores de

s y **e** dependen del nivel de seguridad del modelo de Kyber. Por ejemplo, si se emplea la seguridad de Kyber512 se tendrá que hallar 4 polinomios, 2 que corresponden a los valores del vector **s** y 2 que corresponden al vector **e**. Si se emplea la seguridad de Kyber768 se tendrá que hallar 6 polinomios y para Kyber1024 se deben hallar 8 polinomios.

El costo computacional es muy elevado al tratar de hallar todos los elementos de **s** y **e** al variar entre los tipos de Kyber. Sin embargo, se puede simplificar aun más el análisis matemático si se analiza más a detalle la ecuación (1). Para este ejemplo se usa una seguridad de Kyber512, como se puede apreciar en la siguiente ecuación:

$$As + e = t$$

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad \textbf{(7)}$$

Se procede a analizar cada elemento del vector **t** para observar matemáticamente como se obtienen:

$$A_{1,1}s_1 + A_{1,2}s_2 + e_1 = t_1 \quad \textbf{(8)}$$

$$A_{2,1}s_1 + A_{2,2}s_2 + e_2 = t_2 \quad \textbf{(9)}$$

Como se puede apreciar, tanto la ecuación (8) y (9) contienen los valores de la clave privada. Solo se necesita analizar una de las dos ecuaciones para obtener el valor de la clave privada. Esto permite reducir el número de iteraciones, debido a que se está hallando el menor número de polinomios posibles.

Para la implementación se realiza unos ajustes en los parámetros de seguridad de dicho algoritmo y se analiza la ecuación (8). En este caso se reduce el valor del parámetro **n** hasta llevarlo al valor mínimo. Esto permite reducir la cantidad de combinaciones posibles para hallar los valores de los polinomios. Además, se sabe que los coeficientes de los polinomios de **s** y **e** son números enteros pequeños cercanos al cero, es decir que podrían ser −1, 0 y 1. Para hallar los valores de s_1, s_2 y e_1 se debe de crear listas de todos los posibles valores de cada polinomio y se deben de probar de manera estratégica en la ecuación (8) hasta dar con el valor de t_1, el cuál ya es conocido y se usaría como punto de comparación.

La cantidad de iteraciones que se realicen dependerá del valor de n y del tipo de Kyber que se escoja, por ejemplo, para un $n = 2$ se tendrá 33 combinaciones posibles para cada polinomio. Esto se debe a que son 3 los posibles coeficientes que tendrá cada polinomio, en este caso son los valores 1, 0 y −1. Además, al saber que el grado máximo de los polinomios es de grado 2, indica que un polinomio cuadrático tiene 3 coeficientes $(n + 1)$. Esto quiere decir que para un $n = 2$ y usando Kyber512 se tendría 333333 = 39 iteraciones como máximo para hallar el valor de la clave privada.

Es importante mencionar que se está multiplicando el valor de 33 tres veces, debido a que se tiene que hallar los valores de s_1, s_2 y e_1.

> “El costo computacional es muy elevado al tratar de hallar todos los elementos de s y e al variar entre los tipos de Kyber. Sin embargo, se puede simplificar aun más el análisis matemático si se analiza más a detalle la ecuación

4. Resultados

Como era de esperarse, el método de ingeniería inversa, planteado para vulnerar al algoritmo post-cuántico, tuvo exito al variar sus parámetros de seguridad internos. Sin embargo, a medida que se aumenta el nivel de seguridad, es decir el valor de *n*, el tiempo para encontrar los valores de la clave privada aumenta. La **figura 1** muestra los resultados del tiempo de vulnerabilidad variando el valor de *n* y usando Kyber512, implementado en una computadora que posee un procesador Core i5-1135G7 a 2.40GHz, 16GB RAM y 259GB SSD. Además, se puede apreciar que el crecimiento del tiempo al variar el valor de n es exponencial. Asimismo, se observa que la tendencia de la gráfica corresponde a un coeficiente de determinación igual a 1. Los resultados demuestran que el método planteado es adecuado, no obstante, Kyber posee una seguridad muy robusta ante ataques con computadoras clásicas. Obteniendo la ecuación de la tendencia podemos estimar cuanto tomaría vulnerar el algoritmo utilizando los parámetros reales de Kyber.

$$n = 256$$

$$Tiempo = 4 \cdot 10^{-5} e^{8366592} \text{ segundos} \quad \textbf{(10)}$$

$$Tiempo = 6,93251 \cdot 10^{3633552} \text{ años}$$

Los resultados de la ecuación (10) demuestran que sería practicamente irrealizable llevar a cabo el ciberataque. Además, el tiempo de ejecución aumenta aún más al variar el tipo de Kyber, ya que el valor del parámetro k aumenta. Esto se puede apreciar en las figuras 2 y 3.

FIGURA 1:

Gráfica del tiempo de vulnerabilidad aumentando el valor de n usando Kyber512

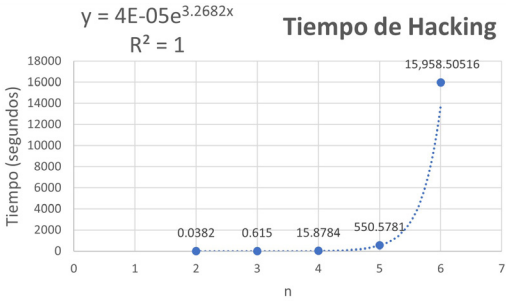


FIGURA 2:

Gráfica del tiempo de vulnerabilidad aumentando el valor de *n* usando Kyber768

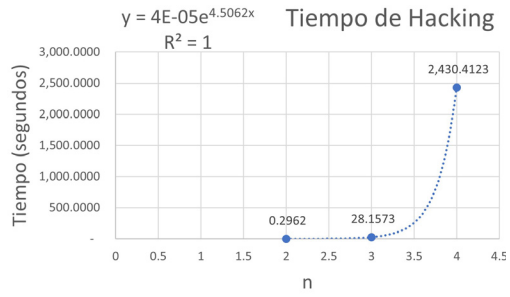
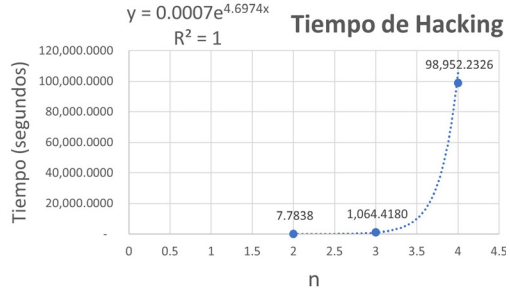


FIGURA 3:

Gráfica del tiempo de vulnerabilidad aumentando el valor de n usando Kyber1024



5. Conclusiones

5-A. Complejidad Computacional

Los resultados de este estudio, realizado por **NTT DATA**, demuestran claramente que si bien es posible vulnerar el algoritmo Kyber con parámetros de seguridad totalmente inseguros, el ataque se hace inviable cuando el valor de estos aumenta. Esto se puede demostrar al hallar la complejidad computacional analíticamente. La cantidad de iteraciones requeridas para vulnerar el algoritmo Kyber, empleando la metodología propuesta, está determinada por tres factores: en primer lugar, depende del grado de los polinomios empleados, representado por la variable *n*. En segundo lugar, varía según el tipo de Kyber utilizado, ya que esto influye en la cantidad de polinomios involucrados en el cifrado, modificando así el valor de la variable *k*. Por último, otro factor crucial es la cantidad de posibles valores que pueden tomar los coeficientes de cada polinomio, en este caso esta variable se denomina con el valor de N.

La ecuación presentada a continuación ilustra la complejidad computacional requerida para comprometer la seguridad del algoritmo Kyber mediante la metodología propuesta, tomando en cuenta los parámetros *k*, *n* y *N*:

$$O\left(N^{(n+1)\cdot(k+1)}\right) \qquad (11)$$

Como se puede apreciar, (11), posee un crecimiento exponencial en función al tamaño de las entradas *k*, *n* y N; lo cual indica que el tiempo de ejecución del programa también crece exponencialmente.

En resumen, esta investigación proporciona una comprensión profunda sobre la fortaleza inherente del algoritmo Crystals Kyber frente a distintos niveles de ataque que utilizan ingeniería inversa y fuerza bruta, lo que reafirma su robustez y resistencia ante ciberataques utilizando computadoras clásicas. La gran complejidad computacional que se debe de implementar demuestra que es un algoritmo totalmente seguro.

A medida que se avanza hacia un futuro en el que la computación cuántica se convierte en realidad, el uso de algoritmos criptográfico post-cuántico es esencial para mantener la seguridad de sistemas de información ante ciberataques, tanto de computadoras clásicas como cuánticas.

Referencias

[1] 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.

[2] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced encryption standard (aes)," 2001-11-26 2001.

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

[4] [4] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM Journal on Computing, vol. 26, no. 5, pp. 1484–1509, 1997.

[5] [5] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehl´e, "Crystals-kyber: a cca-secure module-lattice-based kem," in 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 353–367, IEEE, 2018.

[6] [6] National Institute of Standards and Technology (NIST), "Post-quantum cryptography standardization," 2021.

[7] [7] A. Joux, Algorithmic Cryptanalysis. CRC Press, 2009.

[8] [8] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," Journal of the ACM, vol. 56, no. 6, pp. 34:1–34:40, 2009.

[9] [9] N. I. of Standards and Technology, "Selected algorithms 2022," 2022.

[10] [10] C. Peikert, "A decade of lattice cryptography," Foundations and Trends® in Theoretical Computer Science, vol. 10, no. 4, pp. 283–424, 2016.

pe.nttdata.com