Criptografía post-cuántica, desafíos y direcciones de investigación

Fernando Virdia

NOVA.ID.FCT
Universidade NOVA de Lisboa

Resumen

- Motivación: Criptografía y Computación Cuántica
- Fundamentos: Nuevas Suposiciones de Dificultad
- Estandares: El proceso del NIST de EE. UU.
- Implementación: Algunos desafíos

Diapositivas en https://fundamental.domains

Criptografía "Pre-Post-Cuántica"

Normalmente, la criptografía se presenta como compuesta por dos componentes:

- Criptografía simétrica, que se encarga de las comunicaciones seguras entre partes que comparten una clave secreta o una contraseña.
- Criptografía asimétrica (o criptografía de clave pública), que permite a partes distantes acordar una clave secreta compartida a través de un canal no seguro.

Juntas, posibilitan el despliegue en gran escala de la criptografía que vemos hoy en día en Internet y en los sistemas de pago electronico.

- Ambos tipos de primitivas se construyen utilizando diferentes grados de estructura matemática.
- La estructura usada debería implicar que un adversario que intenta romper la primitiva necesita resolver algún problema matemático difícil.
- Formalizamos estos problemas en "suposiciones de dificultad" concisas.
- Parte del trabajo de los criptógrafos es identificar suposiciones de dificultad, intentar romperlas y construir primitivas a partir de ellas.

Los sistemas de clave pública (PKC) de hoy en día se basan principalmente en suposiciones de dificultad relacionadas con dos problemas matemáticos:

Factorización

Sean p y q dos números primos aleatorios diferentes y de tamaño similar, $\log p \approx \log q$.

Dado $N = p \cdot q$, encontrar p y q.

Logaritmo discreto (DLOG)

Sean G un grupo finito y $g \in G$ un elemento que genera un subgrupo grande $\langle g \rangle \subset G$. Sea x un número entero al azar en $\{0, \ldots, |\langle g \rangle| - 1\}$.

Dado g^x, encontrar x.

Estos problemas han sido ampliamente estudiados y se utilizan en todas partes en software y hardware.

¿Qué entendemos por "la factorización es difícil"?

- Intuitivamente, resolver una instancia aleatoria debería requerir muchos recursos (cálculos, memoria, energía, dinero, *etc.*).
- Para determinar si esto es cierto, investigamos algoritmos para resolver el problema (criptoanálisis) y encontramos una fórmula para el costo te tales algorithms en función de los parámetros del problema (e.g., en funcion de $\log N$).
- Teniendo en cuenta los ataques conocidos, utilizamos estas fórmulas para elegir los parámetros del problema de manera que el costo sea "suficientemente alto" (por ejemplo, de manera que requiera $\geq 2^{128}$ ciclos de CPU para resolverlo).
- También investigamos las relaciones matemáticas del problema con otros similares.

NOTA: No podemos tener certeza absoluta de que el problema sea difícil. (Por ejemplo, tal vez $P = NP$).

Ejemplo: la dificultad de la factorización

- Sea $N = p \cdot q$ con p y q aleatorios, de manera que $\log p \approx \log q$.
- Para factorizar N, se requiere un máximo de $2^{\log p} \approx 2^{\frac{\log N}{2}}$ intentos de división (intentando adivinar p).
- Pero existen ataques mucho más rápidos, como la criba general de cuerpos de números (GNFS), que requiere

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln N)^{\frac{1}{3}}(\ln \ln N)^{\frac{2}{3}}\right) \text{ operaciones de CPU.}$$

- Al elegir adecuadamente $\ln N$, podemos asegurarnos de que GNFS sea demasiado costoso de ejecutar.

¿Sabemos con certeza que no existe un ataque mejor? ¡No! La única opción es hacer nuestro mejor esfuerzo para estudiar el problema y posibles ataques nuevos.

¿Preguntas hasta el momento?

Computación Cuántica

- Hasta ahora, las suposiciones de dificultad relacionadas con la factorización y el logaritmo discreto funcionaron bien. ¿Qué cambió?
- En la década de 1980, algunos físicos comenzaron a considerar el uso de fenómenos mecánicos cuánticos para realizar cálculos.
- Durante mucho tiempo, hubo mejoras prácticas muy pequeñas.
- En la última década, muchas inversiones de la industria se destinaron a esta tecnología [MQT18, MN18, AAB+19, Gib19, WFG21].

Las computadoras cuánticas representarían un nuevo tipo de "recurso" en manos de los atacantes.¿Cómo amenaza esto a la criptografía? Hasta ahora, en forma de dos algoritmos: Grover y Shor.

El algoritmo de Grover

- Supongamos que tenemos una lista L de N elementos diferentes, ordenados al azar.
- Digamos que sabemos que $x \in L$, pero necesitamos encontrar su índice.
- Clásicamente, esto requeriría $O(N)$ comparaciones. $L[0]{=}x?L[1]{=}x? \ldots$
- El algoritmo de Grover te permite encontrar x en $O(\sqrt{N})$ comparaciones superpuestas.

El algoritmo de Grover

¿Cómo afecta a la criptografía?

Ejemplo

- Supongamos que tienes un cifrado con 2^{128} posibles claves secretas.
- Clásicamente, encontrar la clave correcta requiere aproximadamente 2^{128} intentos.
- Cuánticamente, podría requerir aproximadamente $\sqrt{2^{128}} = 2^{64}$ intentos.

Cualquier cifrado se debilita automáticamente.

¡Podrías necesita claves el doble de largas!

El algoritmo de Shor

- Recordemos el tiempo de ejecución del mejor algoritmo de factorización, GNFS:
$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln N)^{\frac{1}{3}}(\ln \ln N)^{\frac{2}{3}}\right) \text{ ciclos de CPU.}$$
- En 1994, Peter Shor desarrolla un algoritmo cuántico que se ejecuta en
$$O\left((\log N)^2(\log\log N)(\log\log\log N)\right) \text{ operaciones cuánticas.}$$
- De subexponencial en $\log N$ (¡difícil!) a polilogarítmico (¡fácil!)
- Peor aún: no solo afecta la factorización, ¡sino también el logaritmo discreto!

- En el lapso de un solo algoritmo, perdimos dos familias de suposiciones de dificultad.
- En particular, las dos en las que la mayoría de la criptografía de clave pública comercial se basa.
- Esto significa que si/en cuanto una computadora cuántica capaz de ejecutar el algoritmo de Shor esté disponible, las futuras comunicaciones encriptadas estarán en riesgo.
- También significa que cualquier mensaje encriptado compartido hasta ese momento y almacenado estará en riesgo de descifrado, incluso si hoy son seguros.

Necesitamos nuevas suposiciones de dificultad que no puedan resolverse con computadoras cuánticas. Necesitamos criptografía "post-cuántica" (PQC).

Hacia la Criptografía Post-Cuántica

¿En qué consistiría esta actualización? Implicaría varios pasos.

- Identificar nuevas suposiciones de dificultad que sean resistentes a la computación cuántica.
- Diseñar primitivas criptográficas basadas en estas suposiciones, usarlas para actualizar protocolos más complejos.
- Producir implementaciones seguras y estándares legales.
- Desplegar en sistemas del mundo real.

¿Preguntas hasta el momento?

Suposiciones de dificultad

Hay muchos tipos, algunos más nuevos, otros más antiguos.

Se utilizan diversas estructuras matemáticas, por ejemplo

- Códigos de corrección de errores
- Anillos polinómicos y retículos algebraicos
- Sistemas de ecuaciones cuadráticas multivariables
- Problemas relativos a isogenias de curvas elipticas
- Problemas relativos a funciones de hash

Suposiciones de dificultad

Hay muchos tipos, algunos más nuevos, otros más antiguos.

Se utilizan diversas estructuras matemáticas, por ejemplo

- Códigos de corrección de errores
- Anillos polinómicos y retículos algebraicos
- Sistemas de ecuaciones cuadráticas multivariables
- Problemas relativos a isogenias de curvas elipticas
- Problemas relativos a funciones de hash

Suposiciones de dificultad

Hay muchos tipos, algunos más nuevos, otros más antiguos.

Se utilizan diversas estructuras matemáticas, por ejemplo

- Códigos de corrección de errores
- Anillos polinómicos y retículos algebraicos ← demos un ejemplo
- Sistemas de ecuaciones cuadráticas multivariables
- Problemas relativos a isogenias de curvas elipticas
- Problemas relativos a funciones de hash

Recordatorio matemático: polinomios

- Dado un anillo algebraico $\mathcal{R}$ (como los enteros, $\mathbb{Z}$, o los enteros módulo q, $\mathbb{Z}_q$)
- y una variable desconocida x,
- se puede definir el anillo de polinomios $\mathbb{Z}[x]$ con elementos $p(x)$ tales que:

$$p(x) = p_0 + p_1 \cdot x + p_2 \cdot x^2 + \cdots + p_n \cdot x^n,$$

  donde $p_0, \ldots, p_n \in \mathcal{R}$, $p_n \neq 0$. Decimos que n es el *grado* de p.
- Podemos sumar, multiplicar y dividir polinomios.

PQC utilizando anillos polinómicos [Reg05, SSTX09, LPR10]

Definamos

- $q \in \mathbb{Z}$, y $\phi = x^n + 1$ donde $n := 2^k$ para algun $k \in \mathbb{Z}_+$,
- $\mathcal{R} := \mathbb{Z}_q[x]/(\phi)$,
- $a \leftarrow U(\mathcal{R})$ y $s, e \in \mathcal{R}$ aleatorios de manera que los coeficientes sigan una distribución gaussiana redondeada al entero más cercano en $[-q/2, q/2)$.

Search Ring Learning With Errors (RLWE)

Dados $(a, b := a \cdot s + e \mod q) \in \mathcal{R} \times \mathcal{R}$, recuperar s.

Decision Ring Learning With Errors (RLWE)

Dados $(a, b) \in \mathcal{R} \times \mathcal{R}$, adivinar si $b \sim U(\mathcal{R})$ o si $b = a \cdot s + e \mod q$.

- Dadas las nuevas suposiciones, se necesitan nuevos diseños.
- A veces, las similitudes entre las suposiciones "precuánticas" y "poscuánticas" significan que los diseños pueden ser similares.
- Incluso en esos casos, se pueden introducir diferencias sutiles.

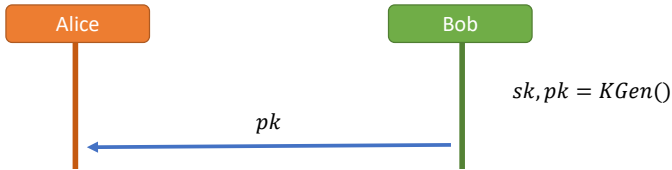
Similitud entre RLWE y DLOG

"dados $(a, a \cdot s + e)$, recuperar s" $\sim$ "dados (g, g^x), recuperar x"

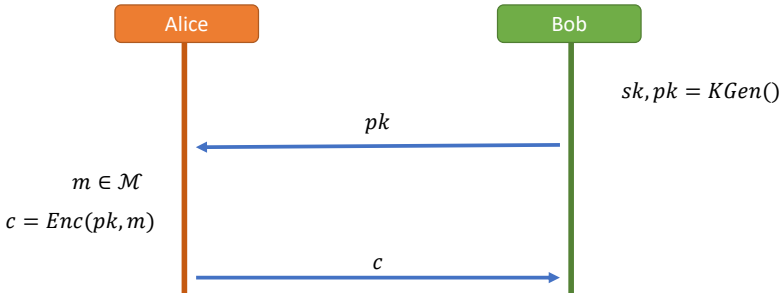
Intentemos usar esto para adaptar una primitiva DLOG a RLWE.

- Presentaré el cifrado ElGamal (seguro de contra atacantes pasiva).
- Este es un esquema clásico de cifrado de clave pública, muy similar al intercambio de claves Diffie-Hellman.

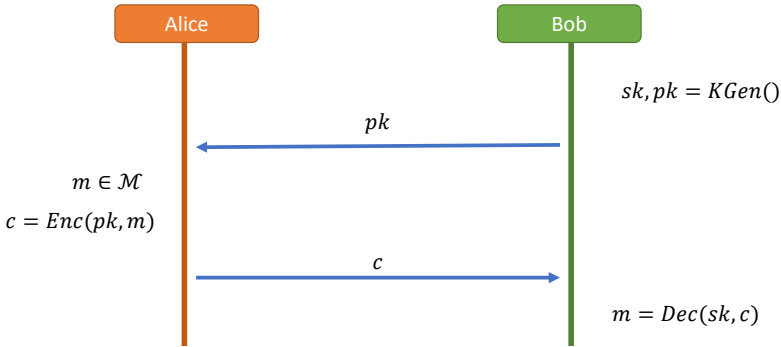
- Presentaré el cifrado ElGamal (seguro de contra atacantes pasiva).
- Este es un esquema clásico de cifrado de clave pública, muy similar al intercambio de claves Diffie-Hellman.



- Presentaré el cifrado ElGamal (seguro de contra atacantes pasiva).
- Este es un esquema clásico de cifrado de clave pública, muy similar al intercambio de claves Diffie-Hellman.



- Presentaré el cifrado ElGamal (seguro de contra atacantes pasiva).
- Este es un esquema clásico de cifrado de clave pública, muy similar al intercambio de claves Diffie-Hellman.



Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x}$

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x}$

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h := g^x)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h \coloneqq g^x)$, $pk \leftarrow (a, b \coloneqq a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h := g^x)$, $pk \leftarrow (a, b := a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h := g^x)$, $pk \leftarrow (a, b := a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$, $c_1 \leftarrow a \cdot r + f$,
- $c_2 \leftarrow h^y \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h := g^x)$, $pk \leftarrow (a, b := a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$, $c_1 \leftarrow a \cdot r + f$,
- $c_2 \leftarrow h^y \cdot m$, $c_2 \leftarrow b \cdot r + f' + \frac{q}{2} \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h := g^x)$, $pk \leftarrow (a, b := a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$, $c_1 \leftarrow a \cdot r + f$,
- $c_2 \leftarrow h^y \cdot m$, $c_2 \leftarrow b \cdot r + f' + \frac{q}{2} \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,
  $m' \leftarrow \quad c_2 - s \cdot c_1$

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h := g^x)$, $pk \leftarrow (a, b := a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$, $c_1 \leftarrow a \cdot r + f$,
- $c_2 \leftarrow h^y \cdot m$, $c_2 \leftarrow b \cdot r + f' + \frac{q}{2} \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,
  $m' \leftarrow \quad c_2 - s \cdot c_1 \quad = \quad (b \cdot r + f' + \frac{q}{2} \cdot m) - s \cdot (a \cdot r + f)$

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h \coloneqq g^x)$, $pk \leftarrow (a, b \coloneqq a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$, $c_1 \leftarrow a \cdot r + f$,
- $c_2 \leftarrow h^y \cdot m$, $c_2 \leftarrow b \cdot r + f' + \frac{q}{2} \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,
  $m' \leftarrow \quad c_2 - s \cdot c_1 \quad = \quad (b \cdot r + f' + \frac{q}{2} \cdot m) - s \cdot (a \cdot r + f) \quad = \quad \frac{q}{2} \cdot m + e \cdot r - s \cdot f + f'$

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h \coloneqq g^x)$, $pk \leftarrow (a, b \coloneqq a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$, $c_1 \leftarrow a \cdot r + f$,
- $c_2 \leftarrow h^y \cdot m$, $c_2 \leftarrow b \cdot r + f' + \frac{q}{2} \cdot m$,

Dec($sk, (c_1, c_2)$):

- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,
  $m' \leftarrow \lfloor c_2 - s \cdot c_1 \rceil = \lfloor (b \cdot r + f' + \frac{q}{2} \cdot m) - s \cdot (a \cdot r + f) \rceil = \lfloor \frac{q}{2} \cdot m + e \cdot r - s \cdot f + f' \rceil$

Sea $\langle g \rangle$ un subgrupo grande de $\mathbb{F}_q^\times$. Sea $a \sim U(\mathcal{R})$.

KGen():

- $sk \leftarrow x \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $sk \leftarrow (s, e) \sim \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $pk \leftarrow (g, h \coloneqq g^x)$, $pk \leftarrow (a, b \coloneqq a \cdot s + e)$,

Enc(pk, m):

- $y \sim U(\mathbb{Z}_{|\langle g \rangle|})$, $(r, f, f') \sim \chi(\mathcal{R}) \times \chi(\mathcal{R}) \times \chi(\mathcal{R})$,
- $c_1 \leftarrow g^y$, $c_1 \leftarrow a \cdot r + f$,
- $c_2 \leftarrow h^y \cdot m$, $c_2 \leftarrow b \cdot r + f' + \frac{q}{2} \cdot m$,

Dec($sk, (c_1, c_2)$):

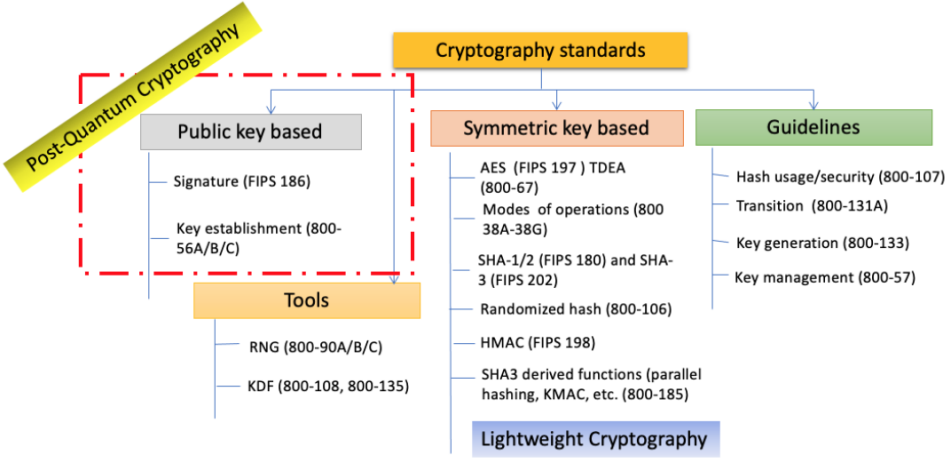
- $m' \leftarrow c_2 / c_1^x = h^y \cdot m / g^{y\,x} = (g^x)^y \cdot m / g^{y\,x} = m$,
  $m' \leftarrow \lfloor c_2 - s \cdot c_1 \rceil = \lfloor (b \cdot r + f' + \frac{q}{2} \cdot m) - s \cdot (a \cdot r + f) \rceil = \lfloor \frac{q}{2} \cdot m + e \cdot r - s \cdot f + f' \rceil$
  $= \frac{q}{2} \cdot m$ *con alta probabilidad.*

¿Preguntas hasta ahora?

Implementaciones seguras y estándares legales

- Las implementaciones seguras son un campo amplio en criptografía.
- No es específico de la criptografía poscuántica, por lo que no lo abordaré.
- Sin embargo, hay mucha investigación en criptografía poscuántica a medida que se acerca la implementación.
- Estén atentos a las publicaciones de la conferencia CHES: https://tches.iacr.org

- En términos de estandarización, se están llevando a cabo múltiples procesos.
- El esfuerzo más prominente ha sido dirigido por el Instituto Nacional de Estándares y Tecnología de los Estados Unidos (NIST).



- En 2016 realizaron una convocatoria abierta para propuestas de diseño de firmas digitales post-cuánticas (DSA) y mecanismos de encapsulación de claves (KEM, piensen en PKE)
- En 2017 se presentaron 69 propuestas.
- Después de múltiples rondas de revisión, en 2023 se han publicado los primeros estándares provisionales para comentarios en https://csrc.nist.gov/projects/post-quantum-cryptography

Se están estandarizando cuatro algoritmos:

- ML-KEM: un mecanismo de encapsulación de clave basado en retículos propuesto con el nombre Kyber.
- ML-DSA y NT-DSA: dos esquemas de firma basados en retículos propuestos como Dilithium y Falcon.
- SLH-DSA: un esquema de firma basado en funciones hash conocido como Sphincs+.

- Mientras tanto, algunos esquemas de KEM están aún en consideración como parte del proceso original.
- NIST también inició un segundo proceso exclusivamente para firmas digitales adicionales.

Las discusiones sobre la estandarización se pueden seguir en https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List.

Bueno, tenemos nuevas suposiciones de dificultad, primitivas y estándares. ¿Podemos desplegarlos ahora, verdad?

Bueno, tenemos nuevas suposiciones de dificultad, primitivas y estándares. ¿Podemos desplegarlos ahora, verdad?

- No es tan fácil en la práctica.

Bueno, tenemos nuevas suposiciones de dificultad, primitivas y estándares. ¿Podemos desplegarlos ahora, verdad?

- No es tan fácil en la práctica.
- Los algoritmos PQC tienden a tener claves públicas y/o cifrados más grandes.

Bueno, tenemos nuevas suposiciones de dificultad, primitivas y estándares. ¿Podemos desplegarlos ahora, verdad?

- No es tan fácil en la práctica.
- Los algoritmos PQC tienden a tener claves públicas y/o cifrados más grandes.

	RSA	EC-DLOG	PQC
Cifrado	$\|pk\| = \|c\| = 384$ B	$\|pk\| = 32$ B, $\|c\| = 64$ B	

Table: Cifrados y firmas con 128-bits de seguridad.

Bueno, tenemos nuevas suposiciones de dificultad, primitivas y estándares. ¿Podemos desplegarlos ahora, verdad?

- No es tan fácil en la práctica.
- Los algoritmos PQC tienden a tener claves públicas y/o cifrados más grandes.

	RSA	EC-DLOG	PQC
Cifrado	$\|pk\| = \|c\| = 384$ B	$\|pk\| = 32$ B, $\|c\| = 64$ B	$\|pk\| = 800$ B, $\|c\| = 768$ B

Table: Cifrados y firmas con 128-bits de seguridad.

Bueno, tenemos nuevas suposiciones de dificultad, primitivas y estándares. ¿Podemos desplegarlos ahora, verdad?

- No es tan fácil en la práctica.
- Los algoritmos PQC tienden a tener claves públicas y/o cifrados más grandes.

	RSA	EC-DLOG	PQC
Cifrado	$\|pk\| = \|c\| = 384$ B	$\|pk\| = 32$ B, $\|c\| = 64$ B	$\|pk\| = 800$ B, $\|c\| = 768$ B
Firmas	$\|pk\| = \|c\| = 384$ B	$\|pk\| = 32$ B, $\|\sigma\| = 65$ B	

Table: Cifrados y firmas con 128-bits de seguridad.

Bueno, tenemos nuevas suposiciones de dificultad, primitivas y estándares. ¿Podemos desplegarlos ahora, verdad?

- No es tan fácil en la práctica.
- Los algoritmos PQC tienden a tener claves públicas y/o cifrados más grandes.

	RSA	EC-DLOG	PQC
Cifrado	$\|pk\| = \|c\| = 384$ B	$\|pk\| = 32$ B, $\|c\| = 64$ B	$\|pk\| = 800$ B, $\|c\| = 768$ B
Firmas	$\|pk\| = \|c\| = 384$ B	$\|pk\| = 32$ B, $\|\sigma\| = 65$ B	$\|pk\| = 32$ B, $\|\sigma\| = 7856$ B

Table: Cifrados y firmas con 128-bits de seguridad.

¿Por qué es esto un problema?

- Si tu protocolo envía muchas claves, texto cifrado o firmas, esto conlleva un aumento en los costos y retrasos.
- Aún peor: ¿qué sucede si la implementación de tu protocolo *asume* tamaños fijos?

```
unsigned char ciphertext[64]
```

- Una gran cantidad de código *sensible* requerirá ser reescrito, con todos los riesgos que esto conlleva. (E.g., CVE-2022-21449: Firmas Psíquicas en Java)

No solo hay problemas con el tamaño.

No todo problema ha recibido el mismo estudio. ¿Podrían romperse?

- A pesar de que RSA y DLOG existen desde la década de 1970 y estándares como PKCS #1 v1.1 datan de 1992, su criptoanálisis no se estabilizó hasta mediados de la década de 1990 [Len93].
- De la misma manera, Rainbow (un DSA finalista de NIST definido por primera vez en 2005) fue vulnerado en 2022 [Beu22].
- Y el esquema SIKE (un KEM finalista de NIST, definido en 2011) fue vulnerado 2022 [CD23].
- ¡Mucho trabajo en criptoanálisis por hacer!

Pero necesitamos PQC lo antes posible!

- ¡Usa esquemas híbridos!
- Para PKE: cifra con EC-ElGamal y cifra el resultado con ML-KEM
- Para firmas: firma con (por ejemplo) EC-DSA y ML-DSA, verifica *ambas* firmas

Conclusiones

- PQC ha recibido un impulso significativo en investigación y esfuerzo industrial.
- Independientemente de si QC alguna vez sucede, los requisitos legales significan que PQC se implementará en un futuro cercano.
- Actualmente se está llevando a cabo mucha investigación: problemas teóricos y prácticos siguen abiertos, lo que brinda un buen espacio para realizar investigaciones.

Conclusiones

- PQC ha recibido un impulso significativo en investigación y esfuerzo industrial.
- Independientemente de si QC alguna vez sucede, los requisitos legales significan que PQC se implementará en un futuro cercano.
- Actualmente se está llevando a cabo mucha investigación: problemas teóricos y prácticos siguen abiertos, lo que brinda un buen espacio para realizar investigaciones.

Conclusiones

- PQC ha recibido un impulso significativo en investigación y esfuerzo industrial.
- Independientemente de si QC alguna vez sucede, los requisitos legales significan que PQC se implementará en un futuro cercano.
- Actualmente se está llevando a cabo mucha investigación: problemas teóricos y prácticos siguen abiertos, lo que brinda un buen espacio para realizar investigaciones.

Conclusiones

- PQC ha recibido un impulso significativo en investigación y esfuerzo industrial.
- Independientemente de si QC alguna vez sucede, los requisitos legales significan que PQC se implementará en un futuro cercano.
- Actualmente se está llevando a cabo mucha investigación: problemas teóricos y prácticos siguen abiertos, lo que brinda un buen espacio para realizar investigaciones.

Gracias

Diapositivas en https://fundamental.domains


Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, and William *et al.* Courtney.
Quantum supremacy using a programmable superconducting processor.
Nature, 574(7779):505–510, Oct 2019.


Ward Beullens.
Breaking rainbow takes a weekend on a laptop.
IACR Cryptol. ePrint Arch., page 214, 2022.


Wouter Castryck and Thomas Decru.
An efficient key recovery attack on sidh.
In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 423–447. Springer, 2023.


Elizabeth Gibney.
Quantum gold rush: the private funding pouring into quantum start-ups.
Nature, 574(7776):22–24, October 2019.


Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman.
NTRU: A ring-based public key cryptosystem.
In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *LNCS*, pages 267–288. Springer, Heidelberg, June 1998.


H. W. Lenstra.
The number field sieve: An annotated bibliography.

In Arjen K. Lenstra and Hendrik W. Lenstra, editors, *The development of the number field sieve*, pages 1–3, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

Vadim Lyubashevsky, Chris Peikert, and Oded Regev.
On ideal lattices and learning with errors over rings.
In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

Robert J. McEliece.
A public-key cryptosystem based on algebraic coding theory.
The deep space network progress report 42-44, Jet Propulsion Laboratory, California Institute of Technology, January/February 1978.
https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

Samuel K. Moore and Amy Nordrum.
Intel's new path to quantum computing.
IEEE Spectrum, 2018.

Microsoft Quantum Team.
Developing a topological qubit.
Cloud Perspectives Blog, 2018.

Oded Regev.
On lattices, learning with errors, random linear codes, and cryptography.
In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.


Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa.
Efficient public key encryption based on ideal lattices.
In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Heidelberg, December 2009.


Karl Wehden, Ismael Faro, and Jay Gambetta.
IBM's roadmap for building an open quantum software ecosystem.
IBM Research Blog, 2021.