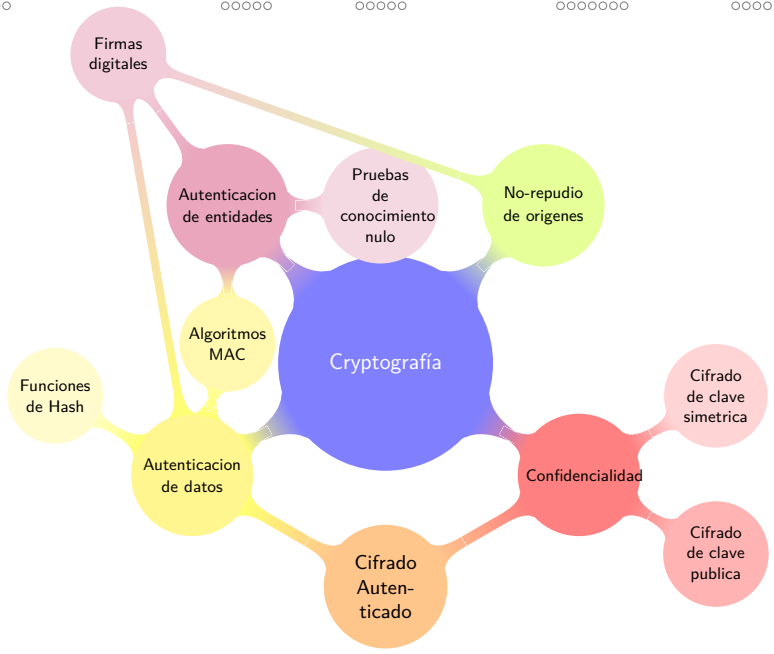
Criptografía post-cuántica, desafíos y direcciones de investigación

Fernando Virdia

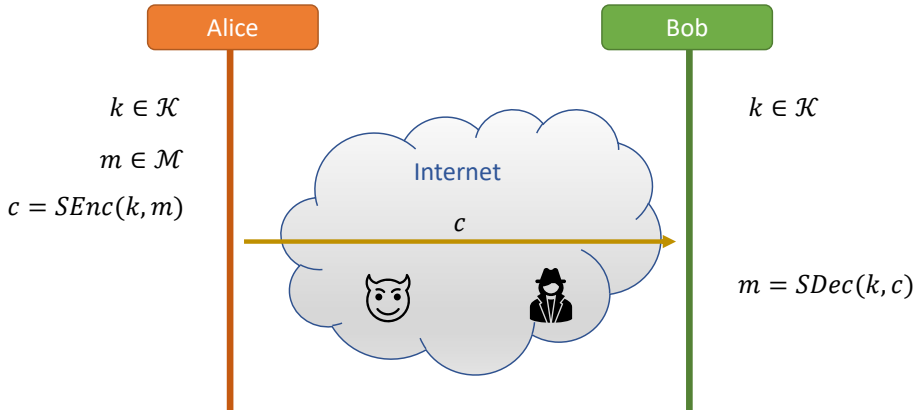
Applied Cryptography Group
ETH Zurich

- En esta charla voy a hablar de algunos problemas abiertos en criptografía.
- Mi objetivo es dar una visión general, de manera que haya algo para gente de M, P, CS y EE.

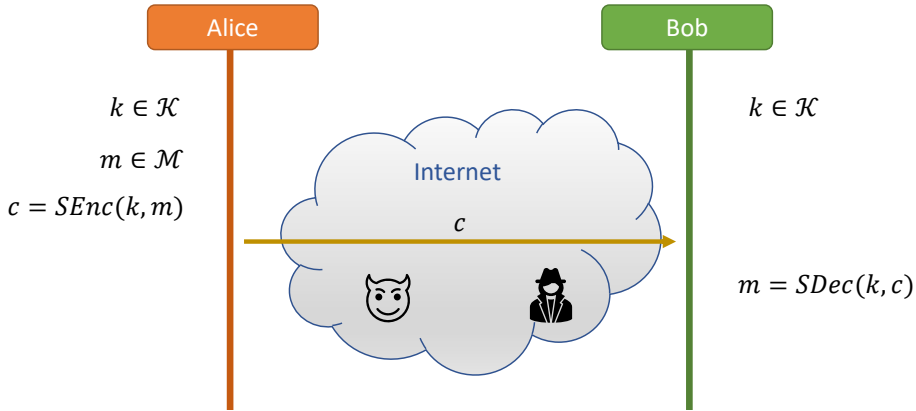
- En esta charla voy a hablar de algunos problemas abiertos en criptografía.
- Mi objetivo es dar una visión general, de manera que haya algo para gente de M, P, CS y EE.
- La criptografía es interdisciplinaria! Es raro que en un equipo de investigación criptográfica todos entiendan todos los ángulos y detalles. Yo seguro no!
- Siéntanse libres de hacer preguntas durante la charla.


Firmas digitales
Autenticacion de entidades
Pruebas de conocimiento nulo
No-repudio de origenes
Algoritmos MAC
Cryptografía
Funciones de Hash
Autenticacion de datos
Cifrado de clave simetrica
Confidencialidad
Cifrado de clave publica
Cifrado Auten-ticado

Cifrado de clave simétrica



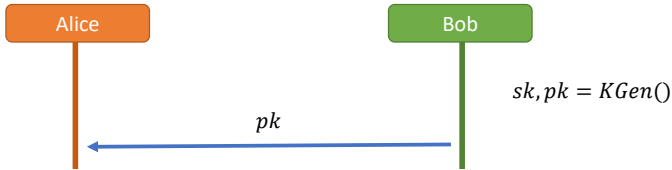
Cifrado de clave simétrica



- Problema: Alice y Bob necesitan compartir la clave secreta k antes de iniciar la comunicación.

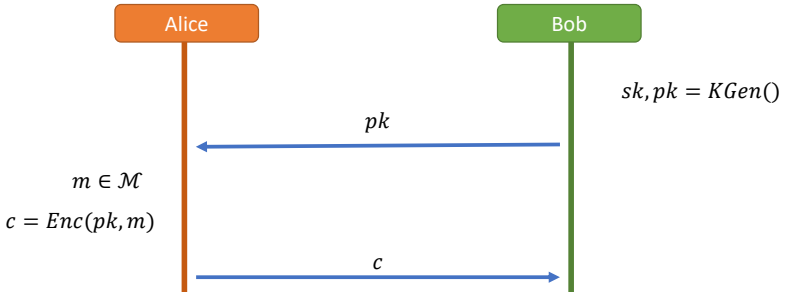
- [DH76] propone la primera solución al problema del establecimiento de claves (KEX).
- Introducen la noción de criptografía asimétrica, o de clave publica (PKC).

- [DH76] propone la primera solución al problema del establecimiento de claves (KEX).
- Introducen la noción de criptografía asimétrica, o de clave publica (PKC).



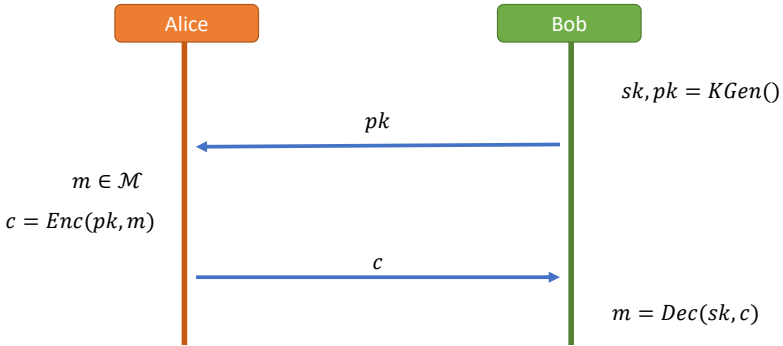
- Bob publica una clave pública que Alice usa para cifrar sus mensajes.

- [DH76] propone la primera solución al problema del establecimiento de claves (KEX).
- Introducen la noción de criptografía asimétrica, o de clave publica (PKC).



- Bob publica una clave pública que Alice usa para cifrar sus mensajes.

- [DH76] propone la primera solución al problema del establecimiento de claves (KEX).
- Introducen la noción de criptografía asimétrica, o de clave publica (PKC).



- Bob publica una clave pública que Alice usa para cifrar sus mensajes.

- Hoy en día el uso de PKC es omnipresente (e.g. pagos y comunicaciones electrónicas).

- Hoy en día el uso de PKC es omnipresente (e.g. pagos y comunicaciones electrónicas).
- La mayor parte de las soluciones PKC basan su seguridad en tres hipótesis de dificultad computacional.

- Hoy en día el uso de PKC es omnipresente (e.g. pagos y comunicaciones electrónicas).
- La mayor parte de las soluciones PKC basan su seguridad en tres hipótesis de dificultad computacional.

RSA (Rivest–Shamir–Adleman) [RSA78]

Dados

- p, q primos distintos, $\log p \approx \log q$,
- $N = p \cdot q$, $e \in (\mathbb{Z}_N)^\times$
- $m \leftarrow U(\mathbb{Z}_N)$
- $c := m^e \mod N$

recupera m.

- Hoy en día el uso de PKC es omnipresente (e.g. pagos y comunicaciones electrónicas).
- La mayor parte de las soluciones PKC basan su seguridad en tres hipótesis de dificultad computacional.

RSA (Rivest–Shamir–Adleman) [RSA78]

Dados

- p, q primos distintos, $\log p \approx \log q$,
- $N = p \cdot q$, $e \in (\mathbb{Z}_N)^\times$
- $m \leftarrow U(\mathbb{Z}_N)$
- $c := m^e \mod N$

recupera m.

Logaritmo discreto (DLOG)

Dados

- $G = \langle g \rangle$
- $x \leftarrow U(\{1, \ldots, |G| - 1\})$
- $h := g^x$

recupera x.

Posiblemente $G = (\mathbb{Z}_p)^\times$ o $E(\mathbb{F}_q)$.

- Hoy en día el uso de PKC es omnipresente (e.g. pagos y comunicaciones electrónicas).
- La mayor parte de las soluciones PKC basan su seguridad en tres hipótesis de dificultad computacional.

RSA (Rivest–Shamir–Adleman) [RSA78]

Dados

- p, q primos distintos, $\log p \approx \log q$,
- $N = p \cdot q$, $e \in (\mathbb{Z}_N)^\times$
- $m \leftarrow U(\mathbb{Z}_N)$
- $c := m^e \mod N$

recupera m.

Logaritmo discreto (DLOG)

Dados

- $G = \langle g \rangle$
- $x \leftarrow U(\{1, \ldots, |G| - 1\})$
- $h := g^x$

recupera x.

Posiblemente $G = (\mathbb{Z}_p)^\times$ o $E(\mathbb{F}_q)$.

Han sido extensivamente estudiadas y desplegadas en forma de código y de hardware.

Que queremos decir con “hipótesis de dificultad computacional”?

Que queremos decir con “hipótesis de dificultad computacional”?

- Es una familia de problemas computacionales, la cual dificultad depende de algunos parámetros (e.g. en RSA, depende de $\log N \approx 2 \log p$).

Que queremos decir con "hipótesis de dificultad computacional"?

- Es una familia de problemas computacionales, la cual dificultad depende de algunos parámetros (e.g. en RSA, depende de $\log N \approx 2 \log p$).
- Para decidir si el problema es difícil en la practica, investigamos algoritmos para resolverlo (criptanálisis) y elegimos parámetros de manera que el costos de estos sea alto (e.g. costo $\geq 2^{128}$ "operaciones").

Que queremos decir con "hipótesis de dificultad computacional"?

- Es una familia de problemas computacionales, la cual dificultad depende de algunos parámetros (e.g. en RSA, depende de $\log N \approx 2 \log p$).
- Para decidir si el problema es difícil en la practica, investigamos algoritmos para resolverlo (criptanálisis) y elegimos parámetros de manera que el costos de estos sea alto (e.g. costo $\geq 2^{128}$ "operaciones").
- También, investigamos relaciones entre problemas diferentes (o a si mismo), en la forma de reducciones de complejidad.

Que queremos decir con "hipótesis de dificultad computacional"?

- Es una familia de problemas computacionales, la cual dificultad depende de algunos parámetros (e.g. en RSA, depende de $\log N \approx 2 \log p$).
- Para decidir si el problema es difícil en la practica, investigamos algoritmos para resolverlo (criptanálisis) y elegimos parámetros de manera que el costos de estos sea alto (e.g. costo $\geq 2^{128}$ "operaciones").
- También, investigamos relaciones entre problemas diferentes (o a si mismo), en la forma de reducciones de complejidad.
- No podemos tener total certeza de que el problema sea difícil en absoluto (e.g. podría ser que $P = NP$).

Ejemplo: seguridad de RSA

- En RSA, la clave publica es (N, e), donde $N = p \cdot q$ y $e \in (\mathbb{Z}_N)^\times$.
- Dada una instancia del problema (una clave publica), el mejor ataque conocido es a través de la factorización de N.

Ejemplo: seguridad de RSA

- En RSA, la clave publica es (N, e), donde $N = p \cdot q$ y $e \in (\mathbb{Z}_N)^\times$.
- Dada una instancia del problema (una clave publica), el mejor ataque conocido es a través de la factorización de N.
- Trivialmente, esta puede tardar $2^{\frac{\log N}{2}} \approx 2^{\lceil \log p \rceil}$ intentos de división (adivinar p).

Ejemplo: seguridad de RSA

- En RSA, la clave publica es (N, e), donde $N = p \cdot q$ y $e \in (\mathbb{Z}_N)^\times$.
- Dada una instancia del problema (una clave publica), el mejor ataque conocido es a través de la factorización de N.
- Trivialmente, esta puede tardar $2^{\frac{\log N}{2}} \approx 2^{\lceil \log p \rceil}$ intentos de división (adivinar p).
- Pero existen ataques mucho mas rápidos, como la *criba general del cuerpo de números* (general number field sieve, GNFS) que tarda
$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln N)^{\frac{1}{3}}(\ln \ln N)^{\frac{2}{3}}\right) \text{ operaciones.}$$
- Eligiendo de manera apropiada $\log N$, el GNFS es suficientemente complejo de ejecutar.

Ejemplo: seguridad de RSA

- En RSA, la clave publica es (N, e), donde $N = p \cdot q$ y $e \in (\mathbb{Z}_N)^\times$.
- Dada una instancia del problema (una clave publica), el mejor ataque conocido es a través de la factorización de N.
- Trivialmente, esta puede tardar $2^{\frac{\log N}{2}} \approx 2^{\lceil \log p \rceil}$ intentos de división (adivinar p).
- Pero existen ataques mucho mas rápidos, como la *criba general del cuerpo de números* (general number field sieve, GNFS) que tarda

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

- Eligiendo de manera apropiada $\log N$, el GNFS es suficientemente complejo de ejecutar.

Sabemos que no hay mejores estrategias? No! La única opción es hacer el mejor esfuerzo en estudiar el problema computacional y los posibles ataques.

Computadoras cuánticas

- La complejidad de RSA parecería ser subexponencial usando computadoras clásicas.
- Sin embargo, en 1994 Peter Shor desarrolla un algoritmo de ataque [Sho97] en

$$O\left((\log N)^2(\log\log N)(\log\log\log N)\right) \text{ operaciones}$$

usando una *computadora cuántica*.

Computadoras cuánticas

- La complejidad de RSA parecería ser subexponencial usando computadoras clásicas.
- Sin embargo, en 1994 Peter Shor desarrolla un algoritmo de ataque [Sho97] en
$$O\left((\log N)^2(\log\log N)(\log\log\log N)\right) \text{ operaciones}$$
usando una *computadora cuántica*.

Computadoras cuánticas (QC)

Computadoras (o circuitos) que pueden operar sobre registros cargados con datos en "superposición cuántica".

Computadoras cuánticas

- La complejidad de RSA parecería ser subexponencial usando computadoras clásicas.
- Sin embargo, en 1994 Peter Shor desarrolla un algoritmo de ataque [Sho97] en

$$O\left((\log N)^2(\log\log N)(\log\log\log N)\right) \text{ operaciones}$$

usando una *computadora cuántica*.

Computadoras cuánticas (QC)

Computadoras (o circuitos) que pueden operar sobre registros cargados con datos en "superposición cuántica".

$$U_f \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} |x, 0^q\rangle = \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} |x, f(x)\rangle$$

Computadoras cuánticas

- La complejidad de RSA parecería ser subexponencial usando computadoras clásicas.
- Sin embargo, en 1994 Peter Shor desarrolla un algoritmo de ataque [Sho97] en
$$O\left((\log N)^2(\log\log N)(\log\log\log N)\right) \text{ operaciones}$$
usando una *computadora cuántica*.

Computadoras cuánticas (QC)

Computadoras (o circuitos) que pueden operar sobre registros cargados con datos en "superposición cuántica".

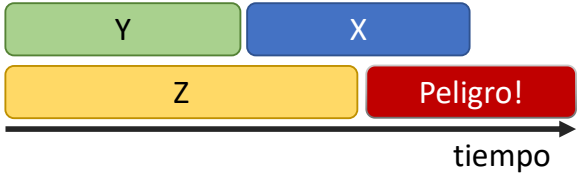
$$U_f \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} |x, 0^q\rangle = \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} |x, f(x)\rangle$$

De paso, el algoritmo de Shor resulta también en ataques poly-log contra DLOG.

- En academia, este resultado ha motivado mucha investigación sobre diferentes hipótesis de complejidad para construir PKC que resista ataques cuánticos, o sea criptografía *post-cuántica* (PQC).
- Algunas de estas hipótesis ya existían anteriormente a [Sho97], pero daban lugar a construcciones menos eficientes a las basadas en RSA y DLOG.

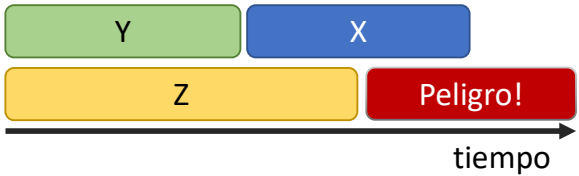
- En academia, este resultado ha motivado mucha investigación sobre diferentes hipótesis de complejidad para construir PKC que resista ataques cuánticos, o sea criptografía *post-cuántica* (PQC).
- Algunas de estas hipótesis ya existían anteriormente a [Sho97], pero daban lugar a construcciones menos eficientes a las basadas en RSA y DLOG.
- A partir de los '80, las QCs representaban un riesgo muy especulativo.
- Sin embargo, en los últimos años tecnologías cuánticas vieron mucha inversión industrial [MQT18, MN18, AAB+19, Gib19, WFG21].

- En academia, este resultado ha motivado mucha investigación sobre diferentes hipótesis de complejidad para construir PKC que resista ataques cuánticos, o sea criptografía *post-cuántica* (PQC).
- Algunas de estas hipótesis ya existían anteriormente a [Sho97], pero daban lugar a construcciones menos eficientes a las basadas en RSA y DLOG.
- A partir de los '80, las QCs representaban un riesgo muy especulativo.
- Sin embargo, en los últimos años tecnologías cuánticas vieron mucha inversión industrial [MQT18, MN18, AAB+19, Gib19, WFG21].
- Aunque QCs capases de ejecutar Shor no estén aun disponibles, es necesario empezar ahora la transición a PQC.



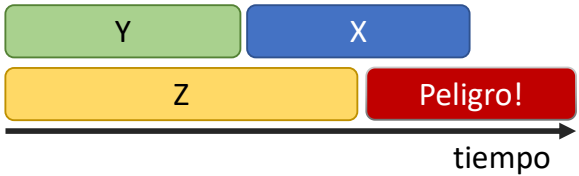
Desigualdad de Mosca [Mos15]

- $X =$ por cuanto tiempo tus claves publicas necesitan ser seguras.



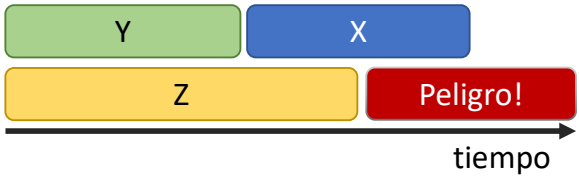
Desigualdad de Mosca [Mos15]

- X = por cuanto tiempo tus claves publicas necesitan ser seguras.
- Y = cuanto tiempo se tardara‘ en implementar PQC en amplia escala.



Desigualdad de Mosca [Mos15]

- X = por cuanto tiempo tus claves publicas necesitan ser seguras.
- Y = cuanto tiempo se tardara‘ en implementar PQC en amplia escala.
- Z = cuanto tiempo hasta que se construyan QC que ejecutan Shor.

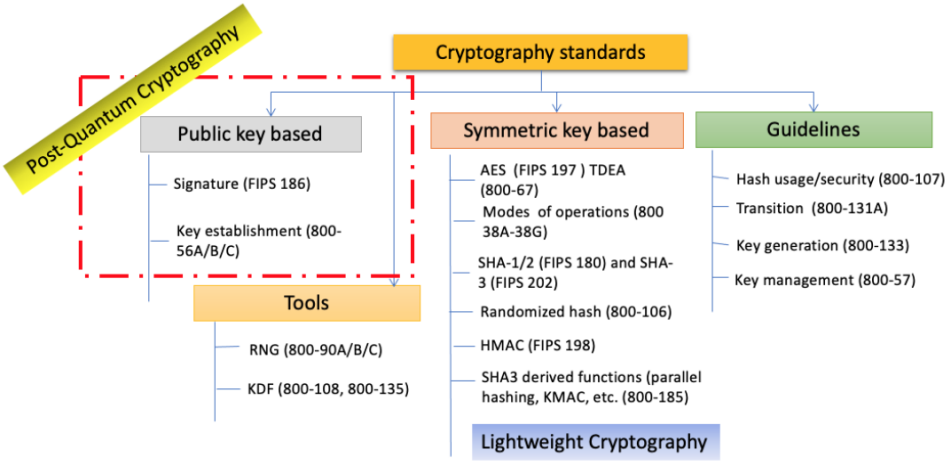


Desigualdad de Mosca [Mos15]

- $X =$ por cuanto tiempo tus claves publicas necesitan ser seguras.
- $Y =$ cuanto tiempo se tardara‘ en implementar PQC en amplia escala.
- $Z =$ cuanto tiempo hasta que se construyan QC que ejecutan Shor.

$\Longrightarrow$ “Teorema”: Si $X + Y > Z$, preocupate.
(Si necesitas cifrar datos en dìa $Y - 1$, no vas a tener PQC disponible.)

Estándares del Instituto Nacional de Estándares y Tecnología (NIST)



- EE.UU. considera posible la construcción de QC que un dia puedan ejecutar el algoritmo de Shor.
- Por ende, en 2016 NIST anunció un proceso publico para crear estándares PQC de cifrado (PKE) y firma digitales (SIG) [Nat16].

- EE.UU. considera posible la construcción de QC que un dia puedan ejecutar el algoritmo de Shor.
- Por ende, en 2016 NIST anunció un proceso publico para crear estándares PQC de cifrado (PKE) y firma digitales (SIG) [Nat16].

El proceso de diseño de tales propuestas tiene dos componente principales:

- La propuesta de una hipótesis de complejidad difícil de resolver usando computadoras clásicas y cuánticas.
- El diseño de PKE o SIG con seguridad basada en tal hipótesis (a través de extensiva criptanálisis y posiblemente reducciones).

Los proponentes presentaron propuestas que incluyen:

- Un documento de diseño que describe la construcción, la criptanálisis y los parámetros elegidos.
- Una implementación de referencia (en C99 portable) y posiblemente varias optimizadas (ASM/FPGA/etc).

El proceso se desarrolla públicamente, con 69 propuestas aceptadas a la primera ronda de estudio en 2017.

Dos siguientes rondas con discusión publica[1] a propósito de

- criptanálisis,

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List

Dos siguientes rondas con discusión publica[1] a propósito de

- criptanálisis,
- consideraciones sobre los diseños,

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List

Dos siguientes rondas con discusión publica[1] a propósito de

- criptanálisis,
- consideraciones sobre los diseños,
- benchmarking en software y hardware,

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List

Dos siguientes rondas con discusión publica[1] a propósito de

- criptanálisis,
- consideraciones sobre los diseños,
- benchmarking en software y hardware,
- discusiones sobre transición de protocolos,

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List

Dos siguientes rondas con discusión publica[1] a propósito de

- criptanálisis,
- consideraciones sobre los diseños,
- benchmarking en software y hardware,
- discusiones sobre transición de protocolos,
- y flame wars.

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List

Dos siguientes rondas con discusión publica[1] a propósito de

- criptanálisis,
- consideraciones sobre los diseños,
- benchmarking en software y hardware,
- discusiones sobre transición de protocolos,
- y flame wars.

Actualmente se están esperando en breve el anuncio de los primeros "ganadores" prontos para ser estandarizados.

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List

Dos siguientes rondas con discusión publica[1] a propósito de

- criptanálisis,
- consideraciones sobre los diseños,
- benchmarking en software y hardware,
- discusiones sobre transición de protocolos,
- y flame wars.

Actualmente se están esperando en breve el anuncio de los primeros "ganadores" prontos para ser estandarizados.

Aunque nuevos estándares estén por ser anunciados, quedan muchos desafíos para la migración y direcciones de investigación.

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/Email-List

Desafíos

Implementación

- Verificabilidad (CS)
- Puesta en seguro (EE)
- Diseño de coprocesadores HW (EE, M)

Desafíos

Implementación

- Verificabilidad (CS)
- Puesta en seguro (EE)
- Diseño de coprocesadores HW (EE, M)

Criptanálisis

- Criptanálisis (cuántica) de las hipótesis de complejidad (M, P)
- Diseño de circuitos que minimizan costos prácticos de los ataques (P, EE, M)
- Modelos de memoria cuántica (P)

Desafíos

Implementación

- Verificabilidad (CS)
- Puesta en seguro (EE)
- Diseño de coprocesadores HW (EE, M)

Criptanálisis

- Criptanálisis (cuántica) de las hipótesis de complejidad (M, P)
- Diseño de circuitos que minimizan costos prácticos de los ataques (P, EE, M)
- Modelos de memoria cuántica (P)

Despliegue

- Adaptación de protocolos a sintaxis no-[DH76] (CS)
- Demostraciones de seguridad para los nuevos protocolos (CS)
- Adaptación de código fuente heredado (CS)

- Naturalmente, el primer paso con nuevos estándares es implementarlos.
- Los diseñadores ya publicaron código de referencia, y algunas implementaciones optimizadas (SSE, AVX, Neon, etc).
- De todos modos, implementaciones son a menudo la fuente de varias vulnerabilidades en criptografía.

- Naturalmente, el primer paso con nuevos estándares es implementarlos.
- Los diseñadores ya publicaron código de referencia, y algunas implementaciones optimizadas (SSE, AVX, Neon, etc).
- De todos modos, implementaciones son a menudo la fuente de varias vulnerabilidades en criptografía.

Session 1D: Applied Cryptography and Cryptanalysis CCS '20, November 9–13, 2020, Virtual Event, USA

LadderLeak: Breaking ECDSA with Less than One Bit of Nonce Leakage

Diego F. Aranha
DIGIT, Aarhus University
Denmark
dfaranha@eng.au.dk

Felipe Rodrigues Novaes
University of Campinas
Brazil
ra135663@students.ic.unicamp.br

Akira Takahashi
DIGIT, Aarhus University
Denmark
takahashi@cs.au.dk

Mehdi Tibouchi
NTT Corporation
Japan

Yuval Yarom
University of Adelaide and Data61
Australia

- Naturalmente, el primer paso con nuevos estándares es implementarlos.
- Los diseñadores ya publicaron código de referencia, y algunas implementaciones optimizadas (SSE, AVX, Neon, etc).
- De todos modos, implementaciones son a menudo la fuente de varias vulnerabilidades en criptografía.

Session 1D: Applied Cryptography and Cryptanalysis CCS '20, November 9–13, 2020, Virtual Event, USA

***LadderLeak*: Breaking ECDSA**

870 IEEE TRANSACTIONS ON RELIABILITY, VOL. 67, NO. 3, SEPTEMBER 2018

Finding Bugs in Cryptographic Hash Function Implementations

Nicky Mouha, Mohammad S. Raunak, D. Richard Kuhn, *Fellow, IEEE*, and Raghu Kacker

Verificabilidad

Una dirección importante de investigación es la de producir implementaciones verificables (correctas y seguras).

Verificabilidad

Una dirección importante de investigación es la de producir implementaciones verificables (correctas y seguras).

- Las operaciones matemáticas necesarias pueden ser difíciles de implementar.
- *Corner cases* pueden ser difíciles de identificar, y a menudo imposibles de detectar con debugging tradicional cuando no resultan en crashes.

Verificabilidad

Una dirección importante de investigación es la de producir implementaciones verificables (correctas y seguras).

- Las operaciones matemáticas necesarias pueden ser difíciles de implementar.
- *Corner cases* pueden ser difíciles de identificar, y a menudo imposibles de detectar con debugging tradicional cuando no resultan en crashes.
- El uso de lenguajes que facilitan la demostración de que la implementación es correcta (Cryptol, F*, F#), así como ambientes de análisis de binario (angr) es muy útil.

Puesta en seguro

- El código criptográfico es normalmente ejecutado en entornos de ejecución no confiables.
- Por ende, las implementaciones tienen que prestar mucha atención al *leakage* de información por canales secundarios (side-channels).

Puesta en seguro

- El código criptográfico es normalmente ejecutado en entornos de ejecución no confiables.
- Por ende, las implementaciones tienen que prestar mucha atención al *leakage* de información por canales secundarios (side-channels).

Los side-channels pueden ser

- de tiempo
- de patrón de acceso a memoria
- de energía consumida

Leakage de estos tipos viene cubierto usando *masking*. Implementaciones eficientes con masking son una área muy activa de investigación.

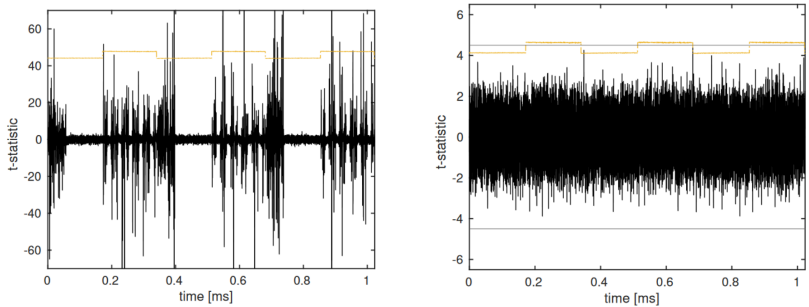


Figure: Test Vector Leakage Assessment de Keccak-f[1600] antes y después de aplicar masking, en [BDK+20].

Diseño de coprocesadores

- Mucho código criptográfico viene ejecutado en coprocesadores hardware especializados (por ejemplo, en le caso de tarjetas con chip).
- En el caso de RSA y DLOG, estos chip traen multiplicadores para enteros de alta dimensión (2000+ bits), que pueden ser usados para PKE, KEX, SIG.

Diseño de coprocesadores

- Mucho código criptográfico viene ejecutado en coprocesadores hardware especializados (por ejemplo, en le caso de tarjetas con chip).
- En el caso de RSA y DLOG, estos chip traen multiplicadores para enteros de alta dimensión (2000+ bits), que pueden ser usados para PKE, KEX, SIG.
- Muchos de los diseños PQC propuestos no comparten el mismo tipo de operación fundamental, y necesitaran nuevos coprocesadores.

Diseño de coprocesadores

- Mucho código criptográfico viene ejecutado en coprocesadores hardware especializados (por ejemplo, en le caso de tarjetas con chip).
- En el caso de RSA y DLOG, estos chip traen multiplicadores para enteros de alta dimensión (2000+ bits), que pueden ser usados para PKE, KEX, SIG.
- Muchos de los diseños PQC propuestos no comparten el mismo tipo de operación fundamental, y necesitaran nuevos coprocesadores.
- Asimismo, un mismo tipo de operaciones puede invitar diferentes diseños de coprocesador, a según de la métrica de costo usada (superficie, consumo energético, profundidad de circuito).

Criptanálisis

- Aunque RSA y DLOG fueron propuestos en los ‘70 y estándares como PKCS #1 v1.1 ya existían en 1991, su criptanálisis no fue estable hasta mediados de los ‘90 [Len93].

Criptanálisis

- Aunque RSA y DLOG fueron propuestos en los ‘70 y estándares como PKCS #1 v1.1 ya existían en 1991, su criptanálisis no fue estable hasta mediados de los ‘90 [Len93].
- Asimismo, solo este año un nuevo ataque devastador fue encontrado en Rainbow, un finalista NIST publicado por primera ves en 2005 [Beu22].
- Mucho trabajo aun por hacer en criptanálisis de las hipótesis de complejidad propuestas.

Criptanálisis

- Aunque RSA y DLOG fueron propuestos en los '70 y estándares como PKCS #1 v1.1 ya existían en 1991, su criptanálisis no fue estable hasta mediados de los '90 [Len93].
- Asimismo, solo este año un nuevo ataque devastador fue encontrado en Rainbow, un finalista NIST publicado por primera ves en 2005 [Beu22].
- Mucho trabajo aun por hacer en criptanálisis de las hipótesis de complejidad propuestas.

Voy a dar una panorámica de la naturaleza matemática de tales hipótesis. Un buen recurso es [BBD09].

PQC usando códigos de corrección de errores

Sean

- $n, t \in \mathbb{Z}$ donde $t \ll n$,

PQC usando códigos de corrección de errores

Sean

- $n, t \in \mathbb{Z}$ donde $t \ll n$,
- G una matriz generadora de código de corrección de errores binario de dimensión k y distancia mínima $d \geq 2t + 1$ (por ejemplo, un código irreducible de Goppa),

PQC usando códigos de corrección de errores

Sean

- $n, t \in \mathbb{Z}$ donde $t \ll n$,
- G una matriz generadora de código de corrección de errores binario de dimensión k y distancia mínima $d \geq 2t + 1$ (por ejemplo, un código irreducible de Goppa),
- S una matriz $k \times k$ aleatoria y no singular y P una matriz permutación $n \times n$,

PQC usando códigos de corrección de errores

Sean

- $n, t \in \mathbb{Z}$ donde $t \ll n$,
- G una matriz generadora de código de corrección de errores binario de dimensión k y distancia mínima $d \geq 2t + 1$ (por ejemplo, un código irreducible de Goppa),
- S una matriz $k \times k$ aleatoria y no singular y P una matriz permutación $n \times n$,
- $\boldsymbol{m} \in \{0,1\}^k$ y $\boldsymbol{z} \in \{0,1\}^n$ de peso de Hamming t.

PQC usando códigos de corrección de errores

Sean

- $n, t \in \mathbb{Z}$ donde $t \ll n$,
- G una matriz generadora de código de corrección de errores binario de dimensión k y distancia mínima $d \geq 2t + 1$ (por ejemplo, un código irreducible de Goppa),
- S una matriz $k \times k$ aleatoria y no singular y P una matriz permutación $n \times n$,
- $\boldsymbol{m} \in \{0,1\}^k$ y $\boldsymbol{z} \in \{0,1\}^n$ de peso de Hamming t.

McEliece [McE78]

Dados t, $G^{\text{pub}} := SGP$ y $\boldsymbol{c} := \boldsymbol{m}G^{\text{pub}} \oplus \boldsymbol{z}$, recuperar $\boldsymbol{m}$.

PQC usando anillos polinomiales

Sean

- $n, q \in \mathbb{Z}$ y $\phi \in \mathbb{Z}[x]$ un polinomio mónico irreducible de grado n,

PQC usando anillos polinomiales

Sean

- $n, q \in \mathbb{Z}$ y $\phi \in \mathbb{Z}[x]$ un polinomio mónico irreducible de grado n,
- $\mathcal{R}_q \coloneqq \mathbb{Z}_q[x]/(\phi)$,

PQC usando anillos polinomiales

Sean

- $n, q \in \mathbb{Z}$ y $\phi \in \mathbb{Z}[x]$ un polinomio mónico irreducible de grado n,
- $\mathcal{R}_q \coloneqq \mathbb{Z}_q[x]/(\phi)$,
- $f \in \mathcal{R}_q^\times$ y $g \in \mathcal{R}_q$ polinomios con coeficientes pequeños (e.g. en $\{-1, 0, 1\}$).

PQC usando anillos polinomiales

Sean

- $n, q \in \mathbb{Z}$ y $\phi \in \mathbb{Z}[x]$ un polinomio mónico irreducible de grado n,
- $\mathcal{R}_q \coloneqq \mathbb{Z}_q[x]/(\phi)$,
- $f \in \mathcal{R}_q^\times$ y $g \in \mathcal{R}_q$ polinomios con coeficientes pequeños (e.g. en $\{-1, 0, 1\}$).

NTRU [HPS98]

Dado $h \coloneqq g \cdot f^{-1} \mod q$, recuperar g o f.

Mas PQC usando anillos polinomiales (o retículos)

Sean

- $k, q \in \mathbb{Z},\ n := 2^k$ y $\phi = x^n + 1$,

Mas PQC usando anillos polinomiales (o retículos)

Sean

- $k, q \in \mathbb{Z},\ n := 2^k$ y $\phi = x^n + 1$,
- $\mathcal{R}_q := \mathbb{Z}_q[x]/(\phi)$,

Mas PQC usando anillos polinomiales (o retículos)

Sean

- $k, q \in \mathbb{Z}$, $n := 2^k$ y $\phi = x^n + 1$,
- $\mathcal{R}_q := \mathbb{Z}_q[x]/(\phi)$,
- $a \leftarrow U(\mathcal{R}_q)$, $s, e \in \mathcal{R}_q$ con coeficientes seleccionados usando una distribución de Gauss redondeada sobre $[-q/2, q/2) \cap \mathbb{Z}$.

Mas PQC usando anillos polinomiales (o retículos)

Sean

- $k, q \in \mathbb{Z}$, $n := 2^k$ y $\phi = x^n + 1$,
- $\mathcal{R}_q := \mathbb{Z}_q[x]/(\phi)$,
- $a \leftarrow U(\mathcal{R}_q)$, $s, e \in \mathcal{R}_q$ con coeficientes seleccionados usando una distribución de Gauss redondeada sobre $[-q/2, q/2) \cap \mathbb{Z}$.

Ring Learning With Errors (RLWE) [Reg05, SSTX09, LPR10]

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

PQC usando sistemas polinomiales cuadráticos

Sean

- $q, n, m \in \mathbb{Z}$ y $\mathbb{F}_q$ el cuerpo numérico de q elementos,

PQC usando sistemas polinomiales cuadráticos

Sean

- $q, n, m \in \mathbb{Z}$ y $\mathbb{F}_q$ el cuerpo numérico de q elementos,
- $p_1(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ polinomios cuadráticos sobre $\boldsymbol{x} = (x_1, \ldots, x_n)$.

PQC usando sistemas polinomiales cuadráticos

Sean

- $q, n, m \in \mathbb{Z}$ y $\mathbb{F}_q$ el cuerpo numérico de q elementos,
- $p_1(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ polinomios cuadráticos sobre $\boldsymbol{x} = (x_1, \ldots, x_n)$.

Multivariate Quadratic (MQ)

Dados $p_1, \ldots, p_m$, encontrar una solución $\boldsymbol{x}$ al sistema $p_1(\boldsymbol{x}) = \cdots = p_m(\boldsymbol{x}) = \mathbf{0}$, si una existe.

PQC usando isogenias entre curvas elípticas

Sean

- $p = 2^{e_2}3^{e_3} - 1$ un primo con $2^{e_2} \approx 3^{e_3}$ y $e_3 \gg 1$,

PQC usando isogenias entre curvas elípticas

Sean

- $p = 2^{e_2}3^{e_3} - 1$ un primo con $2^{e_2} \approx 3^{e_3}$ y $e_3 \gg 1$,
- $\mathbb{F}_{p^2}$ el cuerpo numérico de p^2 elementos, y $\ell \in \{2, 3\}$,

PQC usando isogenias entre curvas elípticas

Sean

- $p = 2^{e_2}3^{e_3} - 1$ un primo con $2^{e_2} \approx 3^{e_3}$ y $e_3 \gg 1$,
- $\mathbb{F}_{p^2}$ el cuerpo numérico de p^2 elementos, y $\ell \in \{2, 3\}$,
- E y E' curvas elípticas súper-singulares sobre $\mathbb{F}_{q^2}$ tales que una isogenia separable $\phi : E \to E'$ de grado ℓ^{e_ℓ} exista.

PQC usando isogenias entre curvas elípticas

Sean

- $p = 2^{e_2}3^{e_3} - 1$ un primo con $2^{e_2} \approx 3^{e_3}$ y $e_3 \gg 1$,
- $\mathbb{F}_{p^2}$ el cuerpo numérico de p^2 elementos, y $\ell \in \{2, 3\}$,
- E y E' curvas elípticas súper-singulares sobre $\mathbb{F}_{q^2}$ tales que una isogenia separable $\phi : E \to E'$ de grado ℓ^{e_ℓ} exista.

Computational Supersingular Isogeny (CSSI) [JD11]

Dadas E y E', encontrar ϕ.

- Ataques cuánticos y clásicos deben ser investigados!
- Lamentablemente la comunidad de algoritmos cuánticos y la de criptanálisis no están en contacto lo suficiente :(

- Ataques cuánticos y clásicos deben ser investigados!
- Lamentablemente la comunidad de algoritmos cuánticos y la de criptanálisis no están en contacto lo suficiente :(
- En algunos casos el mejor algoritmo es clásico (e.g., CSSI [JS19]).

- Ataques cuánticos y clásicos deben ser investigados!
- Lamentablemente la comunidad de algoritmos cuánticos y la de criptanálisis no están en contacto lo suficiente :(
- En algunos casos el mejor algoritmo es clásico (e.g., CSSI [JS19]).
- En otros, el mejor algoritmo es "cuántico", pero tan solo es un algoritmo clásico que utiliza búsqueda de Grover como subrutina. A menudo la ventaja es solo asintótica y no practica, a causa de circuitos muy profundos o de modelos de memoria demasiado generosos hacia el atacante.

- Ataques cuánticos y clásicos deben ser investigados!
- Lamentablemente la comunidad de algoritmos cuánticos y la de criptanálisis no están en contacto lo suficiente :(
- En algunos casos el mejor algoritmo es clásico (e.g., CSSI [JS19]).
- En otros, el mejor algoritmo es "cuántico", pero tan solo es un algoritmo clásico que utiliza búsqueda de Grover como subrutina. A menudo la ventaja es solo asintótica y no practica, a causa de circuitos muy profundos o de modelos de memoria demasiado generosos hacia el atacante.
- Los pocos ataque cuánticos conocidos tocan solo problemas relacionados [DPW19].

Despliegue

Finalmente, los algoritmos implementados tiene que ser puestos en acción.

[2]CVE-2022-21449: Psychic Signatures in Java

Despliegue

Finalmente, los algoritmos implementados tiene que ser puestos en acción.

Habrá problemas de desarrollo...

[2] CVE-2022-21449: Psychic Signatures in Java

Despliegue

Finalmente, los algoritmos implementados tiene que ser puestos en acción.

Habrá problemas de desarrollo...

- La sintaxis de PKE y SIG es la misma entre pre-quantum y post-quantum, por lo cual "simplemente" remplazando el viejo código tendría que ser suficiente...

[2] CVE-2022-21449: Psychic Signatures in Java

Despliegue

Finalmente, los algoritmos implementados tiene que ser puestos en acción.

Habrá problemas de desarrollo...

- La sintaxis de PKE y SIG es la misma entre pre-quantum y post-quantum, por lo cual "simplemente" remplazando el viejo código tendría que ser suficiente...
- Excepto ser común que viejo código falte de flexibilidad, con valores *hardcoded* que no pueden ser fácilmente cambiados. (`unsigned char ciphertext[32]` es una pesadilla!)

[2] CVE-2022-21449: Psychic Signatures in Java

Despliegue

Finalmente, los algoritmos implementados tiene que ser puestos en acción.

Habrá problemas de desarrollo...

- La sintaxis de PKE y SIG es la misma entre pre-quantum y post-quantum, por lo cual "simplemente" remplazando el viejo código tendría que ser suficiente...
- Excepto ser común que viejo código falte de flexibilidad, con valores *hardcoded* que no pueden ser fácilmente cambiados. (`unsigned char ciphertext[32]` es una pesadilla!)
- Mucho código heredado va a necesitar actualizaciones y posiblemente ser remplazado, con todos los problemas que suelen surgir![2]

[2] CVE-2022-21449: Psychic Signatures in Java

Y habrá problemas de investigación.

Y habrá problemas de investigación.

- Muchos protocolos como TLS y Signal usan Diffie-Hellman (DH) KEX basada en DLOG.

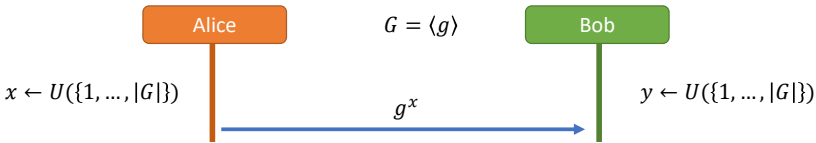
Y habrá problemas de investigación.

- Muchos protocolos como TLS y Signal usan Diffie-Hellman (DH) KEX basada en DLOG.
- Con DH, Alice y Bob pueden acordar una clave secreta en una sola ronda de comunicación.

Y habrá problemas de investigación.

- Muchos protocolos como TLS y Signal usan Diffie-Hellman (DH) KEX basada en DLOG.

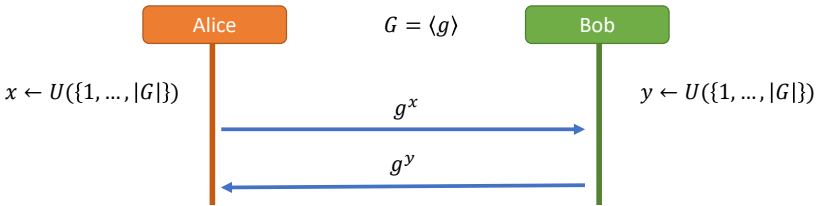
- Con DH, Alice y Bob pueden acordar una clave secreta en una sola ronda de comunicación.



Y habrá problemas de investigación.

- Muchos protocolos como TLS y Signal usan Diffie-Hellman (DH) KEX basada en DLOG.

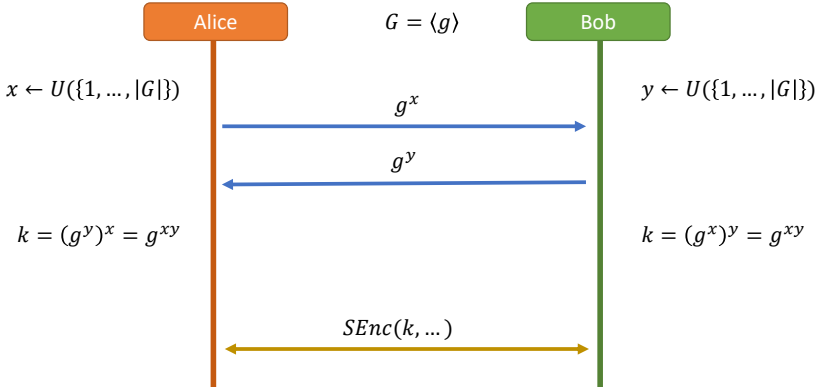
- Con DH, Alice y Bob pueden acordar una clave secreta en una sola ronda de comunicación.



Y habrá problemas de investigación.

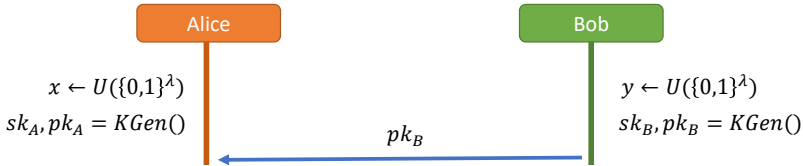
- Muchos protocolos como TLS y Signal usan Diffie-Hellman (DH) KEX basada en DLOG.

- Con DH, Alice y Bob pueden acordar una clave secreta en una sola ronda de comunicación.

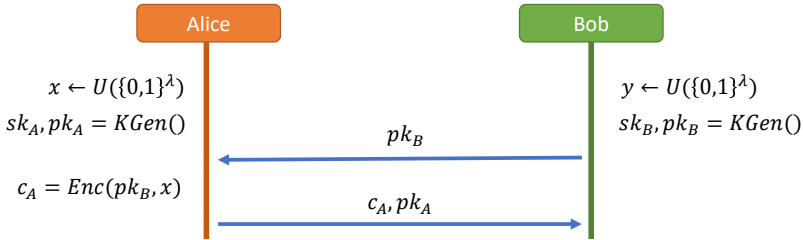


Usando PQC corriente, esto requiere 1.5 rondas.

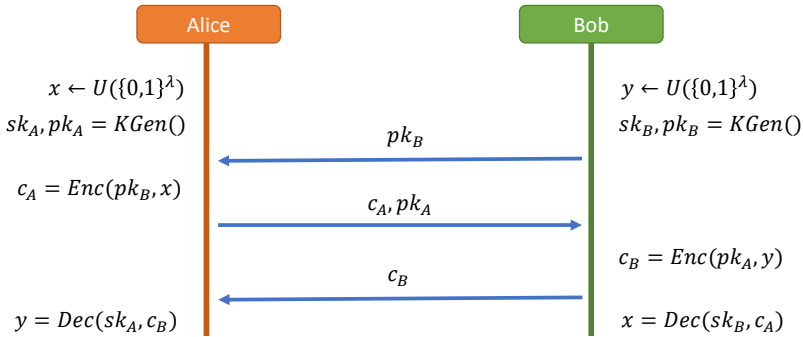
Usando PQC corriente, esto requiere 1.5 rondas.



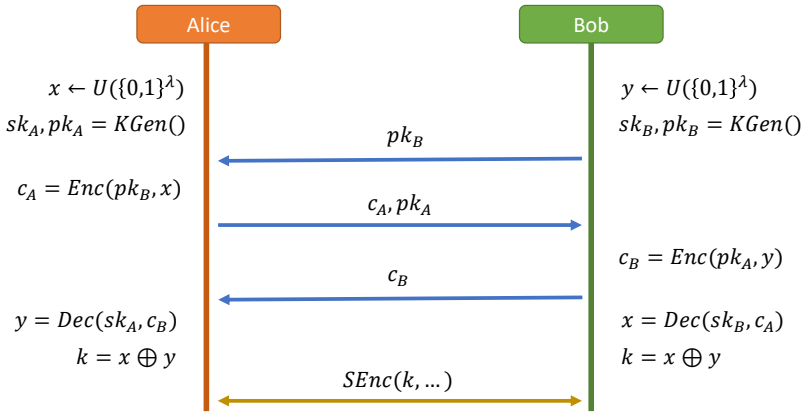
Usando PQC corriente, esto requiere 1.5 rondas.



Usando PQC corriente, esto requiere 1.5 rondas.



Usando PQC corriente, esto requiere 1.5 rondas.



- Protocolos como TLS y Signal tendrán que ser cambiados para suportar PQC (KEMTLS [SSW20] y PQ Signal [BFG+20] han sido propuestos).
- Probablemente muchos mas protocolos necesiten cambios parecidos, y así mismo nuevas demostraciones de seguridad.

En conclusión:

- La transición a PQC va a ser un proceso difícil y largo.

En conclusión:

- La transición a PQC va a ser un proceso difícil y largo.
- Será también inevitable, dado que institutos de estándar van a requerir cumplimiento de estándares PQ para obtener certificaciones como FIPS.

En conclusión:

- La transición a PQC va a ser un proceso difícil y largo.
- Será también inevitable, dado que institutos de estándar van a requerir cumplimiento de estándares PQ para obtener certificaciones como FIPS.
- Asimismo, será una buena oportunidad para producir mucha investigación y desarrollo interesantes en criptografía.

En conclusión:

- La transición a PQC va a ser un proceso difícil y largo.
- Será también inevitable, dado que institutos de estándar van a requerir cumplimiento de estándares PQ para obtener certificaciones como FIPS.
- Asimismo, será una buena oportunidad para producir mucha investigación y desarrollo interesantes en criptografía.

Gracias

Diapositivas @ 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.


Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors.
Post-Quantum Cryptography.
Springer Berlin Heidelberg, 2009.


Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede.
A side-channel resistant implementation of SABER.
Cryptology ePrint Archive, Report 2020/733, 2020.
https://eprint.iacr.org/2020/733.


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


Jacqueline Brendel, Marc Fischlin, Felix Günther, Christian Janson, and Douglas Stebila.
Towards post-quantum security for signal's x3dh handshake.
In *International Conference on Selected Areas in Cryptography*, pages 404–430. Springer, 2020.


Whitfield Diffie and Martin E. Hellman.
New directions in cryptography.

IEEE Transactions on Information Theory, 22(6):644–654, 1976.


Léo Ducas, Maxime Plançon, and Benjamin Wesolowski.
On the shortness of vectors to be found by the ideal-SVP quantum algorithm.
In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 322–351. Springer, Heidelberg, August 2019.


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.


David Jao and Luca De Feo.
Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies.
In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011.


Samuel Jaques and John M. Schanck.
Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE.
In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 32–61. Springer, Heidelberg, August 2019.

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.

Michele Mosca.
Cybersecurity in an era with quantum computers: Will we be ready?
Cryptology ePrint Archive, Report 2015/1075, 2015.
https://eprint.iacr.org/2015/1075.

Microsoft Quantum Team.

Developing a topological qubit.
Cloud Perspectives Blog, 2018.

National Institute of Standards and Technology.
Submission requirements and evaluation criteria for the Post-Quantum Cryptography standardization process.
http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf, December 2016.

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.

Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman.
A method for obtaining digital signatures and public-key cryptosystems.
Communications of the Association for Computing Machinery, 21(2):120–126, 1978.

Peter W. Shor.
Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.
SIAM J. Comput., 26(5):1484–1509, October 1997.

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.


Peter Schwabe, Douglas Stebila, and Thom Wiggers.
Post-quantum TLS without handshake signatures.
In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1461–1480. ACM Press, November 2020.


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