Criptografia Pós-Quântica

tutorial ilustrado com canditaturas à 2ª fase do consurso NIST PQC[1]

José Manuel E. Valença

Dep. Informática & HASLab, Universidade do Minho

12 de Abril de 2019

[1] situação a 30/01/2019

Shor, Gover, ...

Nos últimos 20 anos apareceram algoritmos Q-PPT para

(1) resolver os problemas ´hard' da PKC clássica (Shor)

(2) resolver problemas de procura não estruturada (Gover)

(3) detectar colisões em funções de *hash*,

(4) resolver o *boolean circuit evaluation problem*

(1) é frequentemente citado; o impacto na PKC clássica é bem reconhecido;

(2-4) o impacto nas primitivas simétricas (AES, SHAKE) [2], é menos espetacular e quase ignorado;

[2] tipicamente assumidas como quantum-imunes!

Shor, Gover, ...

Nos últimos 20 anos apareceram algoritmos Q-PPT para

(1) resolver os problemas ´hard' da PKC clássica (Shor)

(2) resolver problemas de procura não estruturada (Gover)

(3) detectar colisões em funções de *hash*,

(4) resolver o *boolean circuit evaluation problem*

(1) é frequentemente citado; o impacto na PKC clássica é bem reconhecido;

PQC fornece soluções alternativas.

(2-4) o impacto nas primitivas simétricas (AES, SHAKE) [2], é menos espetacular e quase ignorado;

[2] tipicamente assumidas como quantum-imunes!

Shor, Gover, ...

Nos últimos 20 anos apareceram algoritmos Q-PPT para

(1) resolver os problemas ´hard' da PKC clássica (Shor)

(2) resolver problemas de procura não estruturada (Gover)

(3) detectar colisões em funções de *hash*,

(4) resolver o *boolean circuit evaluation problem*

(1) é frequentemente citado; o impacto na PKC clássica é bem reconhecido;

PQC fornece soluções alternativas.

(2-4) o impacto nas primitivas simétricas (AES, SHAKE) [2], é menos espetacular e quase ignorado;

potencialmente é o mais disruptivo;
não há soluções para além do aumento no tamanho de chaves.

[2] tipicamente assumidas como quantum-imunes!

Um pouco da história recente da NIST

Em Maio/2015 é registado o standard FIPS 202 para funções de "hash": SHA-3 e suas variantes SHAKE. Abre caminho à PQC.

Em Abril/2016 é publicado o relatório NISTIR 8015, alegando

> ··· *Equally clear is the urgency, implied by these investments, of the need for standardizing new post-quantum public key cryptography.*
> ··· *The most important uses of public key cryptography today are for digital signatures and key establishment.*

30/Nov/2017 é a data de fecho para candidaturas ao concurso NIST PQC; são aceites 58 candidaturas.

30/Jan/2019 é publicada a lista das candidaturas admitidas à 2ª fase; são aceites 26 candidaturas das quais 17 são esquemas KEx/PKE/KEM e os 9 restantes são de assinatura digital.

Segurança + Eficiência = Primitiva PKC

Segurança são crenças e crenças são sempre culturais.

1ª Crença: PKC seguro desenvolve-se em torno de "one-way trapdoor functions" (OWTF). Isto requer: uma estrutura algébrica de domínio finito, um "hard problem" nessa estrutura e um mecanismo "trapdoor": i.e. um algorimo PPT que, acedendo a um "oráculo privado", resolve o "hard problem". Adicionalmente, eficiência requer estruturas sofisticadas.

2ª Crença: Estruturas ricas em propriedades algébricas tornam visíveis estratégias não programadas para resolver "hard problems". Ao invés, PKC seguro e eficiente desenvolve-se em torno de primitivas simples, de baixa complexidade e com propriedades de segurança bem estudadas: por exemplo, hash's e circuitos booleanos de baixa densidade multiplicativa[3].

[3] um dos temas favoritos da novíssima (pós-moderna?!) criptografia.

Candidaturas na 2ª fase – em números e tipo

baseadas em ..	PKE/KEM	Signature
Lattice (NTRU)	3	1
Lattice (RLWE/RLWR)	6	2
Lattice (Codes)	7	
MQP/Factorization		3
EC isogenies	1	
MQP-ZKP-FST		1
Hash functions		1
MPC-ZKP-FST		1

MQP = multivariate quadratic problem
MPC = multi-party computation
RLWE/R = ring learning with errors/rounding
ZKP-FST = zero-knowledge proofs - Fiat-Shamir transform

Sejam $q < 2^\ell, k, n$ inteiros positivos; define-se $\mathbb{Z}_q$ como o anel $\mathbb{Z}/q\mathbb{Z}$ e o intervalo $\mathbb{I}_q \equiv [-\lfloor q/2 \rfloor, \lceil q/2 \rceil) \subset \mathbb{Z}$.

Dado o coset $a \in \mathbb{Z}_q$, $\lceil a \rfloor_q$ é o único elemento de $a \cap \mathbb{I}_q$. Para $z \in \mathbb{Z}$, $[z]_q$ é o elemento de $\mathbb{Z}_q$ que contém z; define-se $\lceil z \rfloor_q \equiv \lceil [z]_q \rfloor_q$.

O *módulo* de $a \in \mathbb{Z}_q$ é $|a| \equiv |\lceil a \rfloor_q|$. Para $x \in \mathbb{Z}_q^n$ ou $x \in \mathbb{Z}^n$ tem-se $\|x\|_1$, $\|x\|_2$ e $\|x\|_\infty$ definidos da forma usual; as distâncias são euclidianas $d(x, y) \equiv \|x - y\|_2$.

"Short vectors"

Para um racional β, um vetor $z \in \mathbb{Z}^n$ diz-se β-**curto** quando $\|z\|_2 \leq \beta$. Para inteiro w, z é w-**curto** quando $\|z\|_\infty \leq 1$ e $\|z\|_1 \leq w$.

$S_{\beta,n}$ (resp. $S_{w,n}$) denota a variável aleatória, com distribuição uniforme, sobre os vectores β-curtos (resp. w-curtos) de $\mathbb{Z}^n$.

“Lattice”

Um **reticulado** é um $\mathbb{Z}$-módulo finitamente gerado. Um **reticulado**-$[q, k, n]$ é um sub-módulo $\mathcal{L} \subseteq \mathbb{Z}^n$ de dimensão k que verifica $q\mathcal{L} \equiv \mathcal{L}$.

$\mathcal{L}$ é um módulo q-periódico; este período assegura que $\mathcal{L}$ é descrito por uma bit-string de comprimento $|\mathcal{L}| \leq k \times n \times \ell$.

Dualidade

Dois reticulados $\mathcal{L}', \mathcal{L} \subset \mathbb{Z}^n$, de dimensões d', d, são **duais** quando $d' + d = n$ e $y'\, y^\top \equiv 0 \bmod q$ para todos $y' \in \mathcal{L}', y \in \mathcal{L}$.

A matriz *full rank* $A \in \mathbb{I}_q^{k \times n}$ define um par de q-lattices em $\mathbb{Z}^n$ *duais*, $\mathcal{L}(A)$ e $\mathcal{L}^*(A)$, de dimensões k e ℓ respetivamente.

$$\mathcal{L}(A) \equiv \{\, y \in \mathbb{Z}^n \mid x\, A \equiv y \bmod q \quad \textit{para algum } x \in \mathbb{Z}^k \,\}$$

$$\mathcal{L}^*(A) \equiv \{\, y \in \mathbb{Z}^n \mid y\, A^\top \equiv 0 \bmod q \,\}$$

Seja $\mathcal{L}$ um reticulado $[q, k, n]$ e $\lambda(\mathcal{L})$ a menor distância entre dois pontos distintos de $\mathcal{L}$.

γ-SIVP (“shortest independent vectors problem”)

Dado racional $\gamma \geq 1$ determinar todos $z \in \mathcal{L} \neq 0$, linearmente independentes e $(\gamma\lambda(\mathcal{L}))$-curtos.

β-SIS (“short integer solution”)

Dado o racional $\beta > 0$, determinar algum $z \in \mathcal{L} \neq 0$ que seja β-**curto**.

Seja χ_α uma variável aleatória sobre $\mathbb{I}_q$, com distribuição gaussiana discreta, normalizada a esse intervalo, com $\sigma = \alpha\, q$ e $\mu = 0$.

Para um segredo $s \in \mathbb{I}_q^n$, o oráculo $\mathrm{LWE}_\alpha(s)$ gera pares (a, b) com $a \leftarrow \mathbb{I}_q^n$, $e \leftarrow \chi_\alpha$ e $b \leftarrow \lceil s\, a^\top + e \rfloor_q$.

LWE “learning with errors”

Determinar s consultando $\mathrm{LWE}_\alpha(s)$ não mais do que $\mathrm{poly}(n)$ vezes.

γ-BDD ("bounded distance decoding")

Dados $\gamma \leq 1/2$ e $t \in \mathbb{Z}^n$ tal que $\min_{z \in \mathcal{L}} \|z - t\| \leq \gamma\, \lambda(\mathcal{L})$, determinar $z \in \mathcal{L}$ que minimiza $\|z - t\|$.

"Sampling" $(A, c) \in \mathbb{I}_q^{k \times n} \times \mathbb{I}_q^k$

Determinar um vector curto $\mathbf{e}$ tal que $\mathbf{e}\, A^\top \equiv c \mod q$.

Um "sampling oracle" para $A \in \mathbb{I}_q^{k \times n}$, é um oráculo S_A que, sob input $c \in \mathbb{I}_q^k$, gera soluções do "sampling problem" (A, c).

"trapdoor sampler" para $A \in \mathbb{I}_q^{k \times n}$

Um "**trapdoor sampler**" $\mathsf{TS}_A(c, s)$ é um algoritmo PPT que, sob input $c \in \mathbb{I}_q^k$ e "trapdoor" $s \in \{0,1\}^t$ gera soluções do "sampling" (A, c).

O “trapdoor sampler” TS_A é “**leak-free**” quando, para todo $s \in \{0,1\}^t$, os oráculos $\{(c, \mathrm{S}_A(c)) \mid c \leftarrow \mathbb{I}_q^k\}$ e $\{(c, \mathrm{TS}_A(c,s)) \mid c \leftarrow \mathbb{I}_q^k\}$ são indistinguíveis.

“short base trapdoor decoder” para A

$\mathrm{TD}_A(t, B)$ é um algoritmo PPT que, sob input de um “target” $t \in \mathbb{Z}^n$ e de uma *matrix curta* B (a “trapdoor”) tal que $B\,A^\top \equiv 0 \bmod q$, gera $z \in \mathcal{L}^*(A)$ tal que $(z - t)$ é curto.

Existe uma relação próxima entre estes dois conceitos; é sempre possível construir um “trapdoor sampler” apartir de um “trapdoor decoder” (o inverso não é verdade); porém o TS assim construído não é necessariamente “leak-free”.

Params: $\alpha, q, n, m = \text{poly}(n)$

KeyGen: $\text{sk} \equiv \left\{ \mathbf{s} \leftarrow \mathbb{I}_q^n \right\} \quad ; \quad \text{pk} \equiv \left\{ (a_i, b_i) \leftarrow \text{LWE}_\alpha(\text{sk}) \right\}_{i=1..m}$

Encrypt(m): sejam $\mathbf{A} \equiv \{a_i\}$ e $\mathbf{b} \equiv \{b_i\}$

$$r\,,\, \varepsilon \leftarrow \chi_\alpha^m \quad ;\; u \leftarrow r\,\mathbf{A}^\top + \varepsilon \quad ;\; v \leftarrow r\,\mathbf{b}^\top + \textbf{enc}(m)$$

Decrypt:

$$m \leftarrow \textbf{dec}(v - u\,\mathbf{s}^\top)$$

Notas:
enc *codifica* m *em* $\mathbb{I}_q$*;* **dec** *descodifica removendo erros curtos.*

O "framework" é adaptável a outras primitivas: CCA2-PKE, KEM, KEx e Sign. Este (e qualquer outro) CPA-PKE converte-se em CCA2-PKE com a transfromação de Fujisaki-Okamoto.

KeyGen:
$\mathrm{sk} \equiv \text{short } \mathbf{B} \leftarrow \mathbb{I}_q^{k \times n} \quad ; \quad \mathrm{pk} \equiv \text{full-rank } \mathbf{A} \text{ s.t. } \mathbf{B}\,\mathbf{A}^\top \equiv 0 \bmod q$

McElice

KEM: $z \leftarrow \mathcal{L}^*(A) \quad ; \ \mathrm{key} \leftarrow \mathrm{Hash}(z) \quad ; \ e \leftarrow S_w \quad ; \ c \leftarrow z + e$

deKEM: $z \leftarrow \mathrm{TD}_{\mathbf{A}}(c, \mathbf{B}) \quad ; \quad \mathrm{key} \leftarrow \mathrm{Hash}(z)$

Niederreiter:

KEM: $e \leftarrow S_w \ ; \ \mathrm{key} \leftarrow \mathrm{Hash}(e) \ ; \ c \leftarrow e\,\mathbf{A}^\top$

deKEM: Resolver $e_0 = c\,\mathbf{A}^\top$ usando algebra linear

$$z \leftarrow \mathrm{TD}_{\mathbf{A}}(e_0, \mathbf{B}) \quad ; \quad \mathrm{key} \leftarrow \mathrm{Hash}(e_0 - z)$$

KeyGen:
$\mathsf{sk} \equiv \mathbf{s} \leftarrow \{0,1\}^t$, $\mathsf{pk} \equiv A \leftarrow \mathbb{I}_q^{k \times n}$ s.t. $\mathbf{s}$ é "trapdoor" de TS_A.

Sign: msg. m

$$r \leftarrow \{0,1\}^\ell \; ; \; c \leftarrow \mathsf{Hash}(r \| m) \; ; \; \sigma \leftarrow \mathrm{TS}_A(c, \mathbf{s})$$

Verify: m , r , σ

$$\mathsf{Hash}(r \| m) \stackrel{?}{\equiv} \sigma A^\top \mod q$$

$q > n > \ell > 0$, $\mathsf{Z} \equiv \mathbb{Z}[x]$, $\phi \in \mathsf{Z}$ com $\deg(\phi) = n$
$\mathsf{R} \equiv \mathsf{Z}/\phi\mathsf{Z}$, $\mathsf{R}_q \equiv \mathsf{R}/q\mathsf{R}$.

$f \in \mathsf{Z}$ é **curto** sse o vector dos coeficientes $\{f_i\}$ é curto; tem-se

- $\lceil f \rfloor_q \in \mathbb{I}_q^n$ definido como $\{\lceil f_i \rfloor_q\}$
- ${}^*f \in \mathbb{I}_q^{n \times n}$ definido como $\{\lceil h_k \rfloor_q\}$ com $h_k \equiv x^k * f \mod q$.

Típicamente

i) $n = 2^\ell$, q é primo, $q \equiv 1 \mod 2n$, $\phi \equiv x^n + 1$

ii) $n + 1$ é primo, $q = 2^\ell$, $\phi \equiv (x^{n+1} - 1)/(x - 1)$

iii) n é primo, $q = 2^\ell$, $\phi \equiv x^n - 1$

ϕ, no caso i), tem n raízes módulo q em $\mathbb{I}_q$; em ii), ϕ é irredutível módulo q; ambos os casos facilitam o uso da NTT ("number theoretic transform"); em iii) todo ${}^*f$ é uma matriz circulante.

“ring (polynomial) lattice”

Seja $\mathcal{R}$ uma extensão polinomial finitamente gerada de $\mathbb{Z}$; um $\mathcal{R}$-**reticulado** é um sub-módulo de $\mathcal{R}^d$, $d > 0$.

NTRU “lattice”

Cada $h \in \mathrm{R}$ determina $\mathcal{L}_h \equiv \{ (f, g) \in \mathrm{R}^2 \mid f * h \equiv g \mod q \}$.

$\mathcal{L}_h$ tem uma **base pública**, $\mathcal{B}_p \equiv \{(1, h), (0, q)\}$ e **bases secretas** $\mathcal{B}_s \equiv \{(f, g), (F, G)\}$ formadas por um vetor curto $(f, g) \in \mathcal{L}_h$, e por um vetor (F, G) tal que $f * G - g * F = q$.

Como reticulados q-ários, a base pública e as bases secretas são descritas por $\mathbb{I}_q$-matrizes $2n \times 2n$:

$$G_p \equiv \left(\begin{array}{c|c} 1_n & {}^*h \\ \hline 0_n & q\,1_n \end{array}\right) \quad , \quad G_s \equiv \left(\begin{array}{c|c} {}^*f & {}^*g \\ \hline {}^*F & {}^*G \end{array}\right)$$

NTRU q, n, ϕ e um pequeno primo $p \ll \lfloor q/2 \rfloor$

KeyGen

$u, v \leftarrow S_w \; ; \; f \leftarrow 1 + p\,u \mid \text{existe } f^{-1} \; ; \; h \leftarrow f^{-1} * v \; ; \; \text{sk} \equiv f \; ; \; \text{pk} \equiv h$

Encrypt$[m \in \{0,1\}^t]$ $\quad r \leftarrow S_w \; ; \; \mu \leftarrow \textbf{encode}(m) \; ; \; c \leftarrow r * h + \mu$

Decrypt $\quad m \leftarrow \textbf{decode}(f * c \mod p)$

RLWE q, n, ϕ e o gerador gaussiano discreto χ_α

KeyGen $\quad a \leftarrow \mathrm{R}_q \; ; \; s, e \leftarrow \chi_\alpha^n \; ; \; b \leftarrow a * s + e \; ; \; \text{sk} \equiv s \; ; \; \text{pk} \equiv (a, b)$

Encrypt$[m \in \{0,1\}^t]$

$r, \varepsilon \leftarrow \chi_\alpha^n \; ; \; u \leftarrow a * r + \varepsilon \; ; \; v \leftarrow b * r + \textbf{encode}(m)$

Decrypt $\quad m \leftarrow \textbf{decode}(v - s * u)$

PQC baseada em códigos correctores de erros (ECC)

Esquemas PKE e KEM, derivados de ECC, derivam de "frameworks" de McEliece e Niederreiter onde o "short base trapdoor decoder" é implementado por um ECC.

Das 7 candidaturas NIST nesta família, quase todas usam códigos "quasi-cíclicos" onde, por questões de eficiência de espaço, a matriz geradora ou de paridade tem uma estrutura de blocos circulantes.

Nenhuma prova, e poucas discutem, se o "decoder" é "leak-free".

PQC de "baixa densidade algébrica"
MPC $\rightarrow$ HVZKP $\rightarrow$ NIZNP

A candidatura PICNIC ilustra um Sign-"framework" baseado na transformação sucessiva de uma cadeia de protocolos:

MPC *N*-"multi-party computation" num circuito booleano público $\mathcal{C}$ de baixa densidade multiplicativa[4]

> Um segredo x é fracionado entre N agentes; em cooperação os agentes avaliam $\mathcal{C}(x) = 1$ sem nenhum revelar a sua "fração".

HVZKP "honest-verifyer ZKP": protocolo interactivo, "prover"/"verifyer", obtido do MPC por transformação "MPC-in-the-Head".

NIZKP "non-interactive ZKP" obtido modificando o HVZKP num Σ-protocolo e aplicando-lhe a transformação Fiat-Shamir.

> as primitivas usadas são só funções de **hash** e "gates" **and** e **xor**

[4] Para $\mathcal{C}$, PICNIC propõe LowMC: um "substitution permutation network" com S-boxes 3×3 que usam só uma multiplicação por bit. Em alternativa propõe a primitiva da cifra ChaCha20.

PQC de "baixa densidade algébrica"

"Multivariate Quadratic Digital Signature Scheme"

A candidatura MQDSS baseia-se num "framework" idêntico ao PICNIC em que o MPC usa polinómios quadráticos multivariáveis.

funções MQ

MQ_n é o conjunto de todos $f \in \mathbb{F}_q[x_1, \cdots, x_n]$ de grau 2.
$F : \mathbb{F}_q^n \to \mathbb{F}_q^m \equiv \{f_1, \cdots, f_m\}$ é $MQ_{n,m}$ se todos $f_i \in MQ_n$

MQP

Dados $F \in MQ_{n,m}$ e $y \in \mathbb{F}_q^m$, encontrar $x \in \mathbb{F}_q^n$ tal que $F(x) = y$.

Quando $n \simeq m$ a complexidade do MQP tende para exponencial.

Protocolo MPC-MQP

Com chave pública (F, y), e chave privada x dividida pelos vários agentes, o MPC avalia $F(x) \stackrel{?}{=} y$ sem revelar qualquer fração.

O esquema SPHINCS+ deriva de uma longa sucessão de esquemas de assinaturas que usam só funções de “hash” ou primitivas análogas.
A primitiva sign básica é a OTS (“one time signature”) de Winternitz:

i) Dado um “hash” h, define-se funções $f^i(x,k)$ por $f^0(x,k) = k$ e $f^{i+1}(x,k) = h(x \| f^i(x,k))$. Escolhe-se $w \in \{4, 16, 256\}$.

ii) $\text{sk} \equiv \{s_1, \cdots, s_\ell\}$; $\text{pk} \equiv \{x, p_1, \cdots, p_\ell\}$ com $p_i = f^{w-1}(x, s_i)$.

iii) Uma mensagem é um tuplo de inteiros positivos $m \equiv \{m_i\}_{i=1}^{\ell}$, $m_i < w$. A sua assinatura é $\sigma \equiv \{f^{m_i}(x, s_i)\}$. A assinatura é válida quando $p_i \stackrel{?}{=} f^{w-1-m_i}(x, \sigma_i)$ é válido para todo i.

Como o nome indica um par de chaves só é seguro para uma única mensagem. Para assinar 2^h mensagens usa-se uma **árvore de Merkle** com h níveis para armazenar pares de chaves suficientes. A gestão das árvores é a essência do esquema apresentado.

Questões?