Kvantum kriptográfiai alkalmazások - rácselméleti protokollok

Csatári Jakab

2021

Poszt-kvantum kriptográfia

Napjainkban a legelterjedtebben alkalmazott kirptográfiai módszerek többsége alapszik azon a tényen, hogy a prímfaktorizációnak nem ismert gyors megoldása a klasszikus számítási modellünkben. Ilyenek például az RSA alapú titkosítások, vagy a Diffie-Hellman kulcscsere.

A kvantumszámítógépek olyan számítógépek, melyek alapvető számítási egységei az úgynevezett qubitek. Egy qubitnek az értéke lehet 0, vagy 1, amennyiben megmérjük; máskülönben lehet szuperpozícióban, amit egy $(a, b) \in \mathbb{C} \times \mathbb{C}$ (ahol $|a|^2 + |b|^2 = 1$) számmal írhatunk le. Ha szuperpozícióban lévő qubitot megmérünk $|a|^2$ valószínűséggel 0, $|b|^2$ valószínűséggel 1 értékű lesz. Egy kvantumszámítógép bizonyos problémákban azt tudja kihasználni, hogy míg a klasszikus számítógépen n bit értéke mindig meghatározott, úgy n qubit 2^n különböző állapotot reprezentál, mindegyiket valamilyen valószínűséggel.

1994-ben Peter Shor adott polinomiális futásidejű kvantumalgoritmust a prímfaktorizációra - ezzel például a diszkrét logaritmus probléma is megoldható polinom időben kvantumgépen. Ennek ellenére ma azért használhatóak még mindig az RSA alapú protokollok, mert kellően nagy teljesítményű kvantumgépek még nem léteznek. Viszont ezek gyártása az utóbbi évtizedben jelentősen felgyorsult, egyre erősebb kvantumgépeket készítenek - tudomásom szerint az eddigi legerősebb a Zuchongzhi, egy 66 qubites gép az USTC kutatóitól.

Emiatt a NIST (National Institute for Standards and Technology) 2016-ban meghírdette, hogy különböző poszt-kvantum kriptográfiai eljárások jelentkezését várják. Ezen protokolloktól elvárjuk, hogy biztonságosak legyenek nagy teljesítményű kvantumszámítógépek létezése mellett is. Közülük több fordulón keresztül választják ki a legalkalmasabbnak tűnőket különböző szempontok alapján.

Rácselméleti problémák

A PQC standardizáció harmadik fordulójának eredményeképp 2021 júliusában 4 kulcscsere protokollt és 3 digitális aláírást választottak döntősnek. Illetve néhány alternatív jelöltet is meghagytak ezeken kívül mindkét kategóriában. Az előbbi 4-ből 3 rácselméleti alapú, utóbbi 3-ból pedig 2. A féléves projekt során ezek közül azokat a protokollokat tekintettem át, melyek az LWE probléma nehézségére alapoznak. (Ezen kívül még NTRU alapú rácselméleti protokollok vannak, melyek az SVP nehézségét használják.)

Rács: $L := \left\{ \sum_{i=1}^{n} \mu_i e_i \;\middle|\; \forall i : \mu_i \in \mathbb{Z} \right\}$ -et a $B = (e_1, e_2, ..., e_n)$ bázishoz tartozó rácsnak nevezzük. $(\forall i : e_i \in \mathbb{R}^n)$

SVP: A legrövidebb vektor probléma. Adott L rács, találjuk meg a legrövidebb (nem nulla) vektort ezen a rácson:
$\lambda(L) := \inf \left\{ ||v|| \;\middle|\; v \in L \backslash \{0\} \right\}$

CVP: A legközelebbi vektor probléma. Adott $L \subset \mathbb{R}^n$ rács és $v \in \mathbb{R}^n$ vektor. Találjuk meg a v-hez legközelebbi L-beli vektort:
$dist(v, L) := \inf \left\{ ||u - v|| \;\middle|\; u \in L \right\}$

SIVP: A legrövidebb független vektorok problémája. Adott L, egy n dimenziós rács, találjunk lineárisan független $\{v_1, ..., v_n\} \subset L$ vektorokat, hogy $\max_{i=1..n} \{||v_i||\} \leq \max_{B} \{||e_i||\}$

Ezen problémák γ-approximációját SVP_γ, CVP_γ, SIVP_γ -val jelöljük.

GapSVP_γ: Adott L rács, eldöntendő, hogy $\lambda(L) \leq 1$, vagy $\lambda(L) > \gamma$.

SIS_γ: Rövid egész megoldás problémája. Adott $A \in \mathbb{Z}_q^{n \times m}$, keresendő rövid megoldása a homogén egyenletrendszernek: $x \in \mathbf{Z}^m$, ahol $||x|| \leq \gamma$ és $Ax = 0$.

LWE: Legyen R_q egy q-adrendű gyűrű. Adott egy fix $s \in R_q^n$ vektor és egy χ valószínűségi eloszlás. Legyen $a \overset{R}{\in} R_q^n$, $e \xleftarrow{\chi} R_q$. Tetszőlegesen sok $\left(a\ , \langle a, s \rangle + e\right)$ párból megállapítandó s.

Ezt a probléma keresőváltozatának hívjuk, létezik eldöntési változat is: DLWE. A DLWE-ben az ilyen párokat kell tudjuk megkülönböztetni véletlen (egyenletes eloszlással) generált (a, u) pároktól. A két probléma ekvivalens: LWE = DLWE.

Az LWE probléma nagyon hasznos eszköznek bizonyult PQC protokollok generálására. A fenti problémákat azért fontos megemlíteni, mert az LWE nehézsége épül rájuk:

- $\gamma = \sqrt{n}$ esetén a $\text{GapSVP}_\gamma \subset \text{NP} \cap \text{Co-NP}$. Peikert-Brakerski megmutatták, hogy klasszikus értelemben visszavezethető GapSVP_γ az LWE problémára.
- Bármely $\gamma \leq O(1)$ esetén SIVP_γ NP-nehéz. Regev mutatott kvantum visszavezetést $\text{SIVP}_\gamma \rightarrow$ LWE.

Tekintsünk m db LWE mintát: $\left(a_i\ , \langle a_i, s \rangle + e_i\right)$ $(i = 1..m)$. Ez felírható $(A, As + e)$ alakban, ahol $A \in R_q^{n \times m}$. Jelöljük A i-edik oszlopvektorát $A^{[i]}$-vel. Ekkor az As érték meghatároz egy rácspontot az $(A^{[1]}, A^{[2]}, ..., A^{[m]})$ bázis által megadott rácsban, és e egy kis "zaj-vektor", amivel el van tolva.

Az LWE alapú asszimetrikus kulcsú eljárások s-et (és e-t is) használják titkos kulcsnak, míg az A, $As + e$ értékek publikusak. Így a $g_A(s, e) := As + e \pmod q$ egyirányú függvény - feltéve a probléma nehézségét. (Ez utóbbi alatt azt értjük, hogy ha LWE $\notin$ P. Például, ha P$\neq$NP, akkor ez fennáll.)

Tehát van egy NP-nehéz problémánk, melyből egyirányú függvény és így asszimetrikus kulcsú kriptográfiai protokollok készíthetőek. NP-nehéz problémából ismerünk sokat, miért tűnnek a rácsalapúak mégis alkalmasabbnak erre a feladatra? Egy probléma nehézségét feltételezhetjük legrosszabb és átlagos esetben (worst-case, average-case hardness):

- worst-case: $\exists$ valahány input, melyre nehéz
- average-case: a legtöbb inputra nehéz

Általában az NP-nehéz problémáról csak worst-case nehézséget feltételezünk, azonban a rácselméleti problémákra létezik worst-case $\rightarrow$ average-case redukció (Ajtai [10]).

Kulcscsere protokollok

A PQC standardizációban úgynevezett KEM protokollok jelentkezését várták, melyek IND-CCA2 biztonságúak. Ezt minden általam vizsgált protokollban úgy oldották meg, hogy IND-CPA biztonságú PKE-re alkalmaznak Fujisaki-Okamoto transzformációt. A KYBER és SABER döntős protokollok, illetve a FrodoKEM egy alternatív KEM jelölt, mindhárom az LWE-t használja.

PKE séma: Publikus kulcsú elkódolás. Egy titkos (privát) kulcsot és egy publikus kulcsot használó kódolás, ahol egy üzenet könnyen elkódolható a publikus kulcs ismeretében, könnyen dekódolható a privát kulcs ismeretében, azonban a privát kulcs nélkül nehéz dekódolni. (A legjobb esély a brute-force megtippelése a privát kulcsnak.) Egy PKE séma 3 részből áll:

- KeyGen: $() \to (pk, sk)$
- Encrypt: $(pk, m) \to (c)$
- Decrypt: $(c, sk, pk) \to (m)$

A KeyGen egy kulcsgeneráló mechanizmus, előállítja a publikus kulcsot: (pk) és a privát kulcsot: (sk). Egy m üzenetet bárki el tud kódolni az Encrypt-tel egy c kódolt-üzenetté, melyet csak a privát kulcsot ismerő fél tud dekódolni Decrypt-tel.

KEM: Kulcs-beágyazó mechanizmus. Célja, hogy két fél közötti közös munkamenetkulcsot hozzon létre:

- KeyGen: $() \to (pk, sk)$
- Encaps: $(pk) \to (c, K)$
- Decaps: $(c, sk, pk) \to (K)$

A PKE-hez hasonló séma, itt K egy random generált munkamenetkulcs, melyet megőriz a generálója. c alapján viszont Decaps-szal sk ismeretében könnyen vissza lehet kapni.

IND-CCA2 biztonság: Adott A valószínűségi Turing gép egy orákulummal, mely az elkódolt üzeneteket dekódolja. Adott B Turing gép, ha A a következő protokollnak $\leq \frac{1}{2} \pm \epsilon$ valószínűséggel tud eleget tenni, akkor IND-CCA2 biztonságú a kódolás:

1. B generál (pk, sk) kulcsokat, pk-t elküldi A-nak.
2. A (előzetes számítások után) két különböző m_0, m_1 üzeneteket küld B-nek.
3. B $\frac{1}{2}$ valószínűséggel választva $b \in \{0, 1\}$-et, visszaküldi $c :=$Encrypt(pk, m_b)-t.
4. A számításokat végezve és az orákulumot néhányszor megkérdezve (c-t megtiltjuk, hogy közvetlen kérdezhesse tőle) megmondja b értékét.

Az **IND-CPA** biztonság gyengébb, a definíció ugyanez, csak A nem rendelkezik dekódoló orákulummal.

Fujisaki-Okamoto transzformáció: Ez a módszer IND-CPA biztos PKE-ből → IND-CCA2 biztos KEM-et csinál. Ahhoz, hogy egy PKE IND-CPA biztonságot adjon az Encrypt részben szükséges valamilyen random r-et használjon, most feltehető, hogy ezt inputként beadhatjuk neki: Encrypt: $(pk, m, r) \to (c)$. Jelölje "$a||b$", hogy a-t és b-t egymás után írjuk.

Ekkor KEM.Encaps(pk):

1. H és G hashfüggvények (G kétszer olyan hosszú inputot vár), m random érték. $k||r := G(m, H(pk))$
2. Hívjuk meg az elkódolást úgy, hogy a random biteket r, az üzenetet m adja: $c :=$ PKE.Encrypt(pk, m, r)
3. $K := f(k, c)$ (ahol f valamilyen publikus key-derived-function, jellemzően egy extended-output-function)

A KEM.Decaps pedig úgy néz ki, hogy PKE.Decrypt-tel vissza tudjuk fejteni m-et, mellyel ugyanazt az eljárást csinálhatjuk, mint KEM.Encaps esetén, hogy ugyanahhoz a K kulcshoz jussunk. Megjegyzés: amennyiben az input c nem egyezik azzal, amit kiszámolunk, akkor ettől az inputtól függő, de más K kulcsot adunk vissza.

FrodoKEM: Ez a protokoll csak alternatív jelölt maradt a harmadik forduló végén, aminek - úgy tudom - első sorban teljesítménybeli oka volt, a döntősöknél lassabb protokollnak bizonyult. A FrodoKEM egy az egyben használja fel az LWE problémát az $R_q := \mathbb{Z}_q$ gyűrűn.

- KeyGen:
$S_0, E_0 \in \mathbb{Z}_q^{n \times m}$, $A \in \mathbb{Z}_q^{n \times n}$
$sk := (S_0, E_0)$
$pk := (A, T) = (A, AS_0 + E_0)$
- Encrypt:
$S_1, E_1 \in \mathbb{Z}_q^{m \times n}$, $E_2 \in \mathbb{Z}_q^{m \times m}$
$c := (U, V) = (S_1A + E_1, S_1T + E_2 + encode(m))$
- Decrypt:
$m := decode(V - US_0)$

A FrodoKEM Decrypt valóban visszaadja m-et:
Legyen $m' := V - US_0$, ekkor:

$$m' = S_1T + E_2 + encode(m) - (S_1A + E_1)S_0 \quad (1)$$

$$m' = S_1AS_0 + S_1E_0 + E_2 + encode(m) - S_1AS_0 - E_1S_0 \quad (2)$$

$$m' = encode(m) + E^* \quad (3)$$

És $decode(encode(m) + E^*) = m$ ([7] - 2.18 lemma szerint)

A FrodoKEM előnyös abból a szempontból, hogy az LWE-re közvetlenül épít, így a biztonságát egy alaposan vizsgált problémára alapozza. Hátránya, hogy mátrixokkal számol, ami időigényes, ezért a gyorsasága elmarad a többi döntős jelölthöz képest.

KYBER: A KYBER a Frodohoz hasonló protokoll, azonban $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, így $A \in R_q^{n \times n}$ és $s_0, e_0 \in R_q^n$ vektorok; illetve q 2-hatvány. Ennek pedig az lesz az előnye, hogy használható az úgynevezett NTT (number theoretic transform).

NTT: Ha adott $A \in R_q^{n \times k}$ és $s \in R_q^n$, akkor As kiszámításához $n \cdot k$ polinomszorzást kell elvégezni. Azonban NTT domainben számolva ez lecsökkenthető n szorzásra:
$X^n + 1 = (X - r)(X - r^3)...(X - r^{2n-1})$
A kínai maradéktétel szerint így $\forall\ a \in \mathbb{Z}_q[X]/(X^n + 1)$ szám egyértelműen írható fel $\left(a(r), a(r^3), ..., a(r^{2n-1})\right)$ alakban, ezt hívjuk NTT domainnek. Két NTT domain-beli polinomot pedig koordinátánként lehet összeszorozni.

KYBER a FrodoKEM-hez képest jelentősen gyorsabb. Az ilyen polinomgyűrűre épülő verzióját az LWE-nek modul-LWE-nek (MLWE) hívják, és habár bizonyítottan legalább olyan nehéz, mint a rácsproblémák modulokra vonatkoztatva ([5]), nem kizárt hogy a specifikusabb struktúrát kihasználva olyan támadás veszélyes lehet rá, amely a standard LWE-re nem az.

SABER: KYBER-hez hasonlóan $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, továbbá adott $p : p|q$ és hasonlóan $R_p = \mathbb{Z}_p[X]/(X^n + 1)$.

Jelölés: ha $v \in R_q$, akkor $\lceil v \rfloor_{q \to p}$ egy R_p-beli elemet jelöl.
Legyen $\lceil v \rfloor_{q \to p} := \frac{p}{q} \cdot v \pmod p$. A SABER az LWR problémát használja, melynek az ötlete, hogy, ha e kicsi, akkor $\lceil \langle a, s \rangle + e \rfloor_{q \to p} = \lceil \langle a, s \rangle \rfloor_{q \to p}$. (Megjegyzés: a modulus váltásnál használhatunk egy $+c$ shiftelést.)

Tehát a SABER protokoll az előzőekhez hasonlóan írható le, annyi különbséggel, hogy az As -ek helyett $\lceil As \rfloor_{q \to p}$ -t használunk. Az LWR legalább olyan nehéz, mint LWE.

Digitális aláírások

A digitális aláírás szintén fontos terület PQC szempontból. Egy digitális aláírásnak három fázisa van:

- KeyGen: $() \to (pk, sk)$
- Sign: $(sk, pk, m) \to (sign)$
- Verify: $(pk, m, sign) \to (\{\text{ELFOGAD, ELULTASÍT}\})$

DILITHIUM: LWE problémára épülő döntős az aláírások körében. Pontosabban az MLWE problémára, illetve az úgynevezett SelfTargetMSIS-re épül. Ez utóbbi a SIS-nek egy olyan változata, ahol adott A mátrixhoz keresünk y, c, M értékeket, hogy teljesüljön: $c = hash\left((I|A) \begin{pmatrix} y \\ c \end{pmatrix}, M \right)$.

A digitális aláírásoknál fontos szempont az aláírás és a hozzá tartozó kulcsok méretének minimalizálása. A DILITHIUM használ egy $HighBits(w)$ függvényt, melynek alapvetően tömörítés a célja. Ha w nagy szám z-hez képest, akkor $HighBits(w + z) = HighBits(w)$.

- KeyGen: $t := As_1 + s_2$
$pk := (A, t)$
$sk := (s_1, s_2)$
- Sign: y random
$c := hash(m||HighBits(Ay))$
$z := cs_1$
$sign := (c, z)$
- Verify:
$c' := hash(m||HighBits(Az - ct))$
ELFOGAD ha $c' = c$, különben ELUTASÍT

Sign közben vizsgáljuk, hogy z adott korláton belül van-e, amíg nincs újraszámoljuk másik y számból. Ez ahhoz kell, hogy a HighBitjei ne "zavarjanak be". Így Verify helyes, ugyanis:

$$Az - ct = A(y + cs_1) - c(As_1 + s2) = Ay - cs_2 \tag{4}$$

$$HighBits(Ay - cs_2) = HighBits(Ay) \Rightarrow c' = c \tag{5}$$

Felhasznált irodalom

1. M. A. Nielsen, I. L. Chuang. (2010) Quantum Computation and Quantum Information
`http://mmrc.amss.cas.cn/tlb/201702/W020170224608149940643.pdf`

2. Kinorányi D. (2020) Poszt-kvantum kriptográfia
`https://web.cs.elte.hu/blobs/diplomamunkak/bsc_matelem/2021/kinoranyi_dora.pdf`

3. ETSI (2021) CYBER; Quantum-Safe Public-Key Encryption and Key Encapsulation
`https://www.etsi.org/deliver/etsi_tr/103800_103899/103823/01.01.01_60/tr_103823v010101p.pdf`

4. R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehlé (2021) CRYSTALS-Kyber Algorithm Specifications And Supporting - Documentation
`https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf`

5. A. Langlois, D. Stehlé - Worst-Case to Average-Case Reductions for Module Lattices
`https://eprint.iacr.org/2012/090.pdf`

6. D. Bogdanov (2005) IND-CCA2 secure cryptosystems
`https://courses.cs.ut.ee/2005/crypto-seminar-fall/slides/S5.Bogdanov.indcca2.pdf`

7. E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, D. Stebila (2020) FrodoKEM - Learning With Errors Key Encapsulation - Algorithm Specifications And Supporting Documentation
`https://frodokem.org/files/FrodoKEM-specification-20200325.pdf`

8. ETSI (2021) CYBER; Quantum-Safe Signatures
`https://www.etsi.org/deliver/etsi_tr/103600_103699/103616/01.01.01_60/tr_103616v010101p.pdf`

9. S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé (2021) CRYSTALS-Dilithium - Algorithm Specifications and Supporting Documentation
`https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf`

10. Ajtai M. (1996) Generating hard instances of lattice problems Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. Philadelphia, Pennsylvania, United States: ACM. pp. 99–108.