RSA feltörése Shor algoritmusával

Heltai Kilián, Funk Gábor

2026. április

Kivonat

Bevezetés a kvantuminformatikába és -kommunikációba tárgy házi feladat *nyilvános kulcsú titkosítás feltörése* témának kidolgozása. RSA titkosítás feltörése kvantumalgoritmussal, avagy klasszikus probléma kvantumosítása. A dokumentum bemutatja a klasszikus problémát, a kvantum Fourier-transzformációt és Shor algoritmusát.

Tartalomjegyzék

1. Bevezetés

Ezen dokumentum egy részt ismerteti a klasszikus Fourier-transzformációt és RSA algoritmust, annak pedig feltörésének komplexitását. Ez után bemutatásra kerülnek a kvantum megoldáshoz szükséges elméleti elemek, majd pedig az algoritmus. A végén pedig az IBM Quantum Cloud Composer platformján, illetve Qiskittel végzett szimulációk kerülnek bemutatásra.
Miért nem történt meg még a Q-Day, és mi lesz, ha megtörténik?

2. Klasszikus RSA alapja [1]

Az RSA az egyik legfontosabb aszimmetrikus titkosítási rendszer. Asszimmetrikus oly módon, hogy az algoritmus két kulcs alapján működik: egy nyilvános (publikus) kulcs, amely mindenki számára ismert és egy privát kulcs, amit titokban kell tartani.
Fontos elméleti alapot ad az Euler-féle φ függvény, amely adott $n = pq$ prímszám-szorzatra alkalmazva $\varphi(n) = (p-1)(q-1)$ eredményt ad. Illetve másik alapja az Euler-Fermat tétel, amely szerint, ha valamilyen a egész és n relatív prímek, akkor $a^{\varphi(n)} \equiv 1 \mod n$.
A kulcsgenerálás lépései:

1. Két nagy prímszám generálása: p és q.
2. $n = pq$ meghatározása.
3. $\varphi(n)$ kiszámítása.
4. e egész választása úgy, hogy $1 < e < \varphi(n)$ és $(e, \varphi(n)) = 1$.
5. d kiszámítása az Euklideszi algoritmussal: $d \cdot e \equiv 1 \mod \varphi(n)$.

A lépések után a nyilvános kulcs: (e, n), és a privát kulcs pedig (d, n).
A feltörás nehézsége azon alapszik, hogy nagyon nehéz az egészprím-faktorizáció, mert magas komplexitású feladat n-t felbontani p és q szorzatára. A leggyorsabb klasszikus faktorizáló algoritmus is csak szubexponenciális, nem polinomiális. [2] Maga a probléma viszont csak NP (és coNP), hiszen egy adott megoldás helyességének igazolása eléggé egyszerű.
A később taglalt feltöréshez meg kell említsünk egy fontos elméleti következményt. Ha találunk egy a-t és egy r rendet (a legkisebb pozitív egész, amelyre $a^r \equiv 1 \mod n$), és teljesülnek bizonyos feltételek, akkor: $(a^{\frac{r}{2}} \pm 1, n)$ *nagy valószínűséggel* nemtriviális faktort ad, vagyis nem 1-et és nem n-t. Ennek igazsága az Euler-Fermat tételből levezethető. A két feltétel, ami kell:

1. r legyen páros (különben $\frac{r}{2}$ nem egész, és értelmetlen eredményt kapnánk).
2. $a^{\frac{r}{2}} \not\equiv -1 \mod n$ (ha ez teljesülne, akkor csak triviális faktort kapnánk).

A feltörést biztosító kvantumos megoldás ezt a "trükköt" veszi alapul.

3. Klasszikus Fourier-transzformáció röviden [3]

A későbbiekben bemutatott Shor algoritmus egy részét képezi a kvantum Fourier-transzformáció (QFT). De mielőtt abba belemennénk, érdemes megemlíteni a klasszikus FT-t.
Szemléletesen: a legegyszerűbb a Fourier-transzformációra úgy tekinteni, hogy egy olyan függvény, ami egy jelről megmondja, milyen frekvenciákra bomlik le. (Vagy milyen frekvenciákból lehet összetenni.)

A klasszikus Fourier-transzformáció egy $(x_0, x_1, \ldots, x_{N-1}) \in \mathbb{C}^N$ vektorra hat, és azt a $(y_0, y_1, \ldots, y_{N-1}) \in \mathbb{C}^N$ vektorra képezi le az alábbi módon:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \, \omega_N^{-jk}, \quad k = 0, 1, 2, \ldots, N-1,$$

ahol

$$\omega_N = e^{\frac{2\pi i}{N}}$$

az egység egy N-edik gyöke.

Ezt a függvényt úgy lehet értelmesen felfogni, hogy végig megy és *megnézi*, hogy egy frekvencia mennyire van a jelen. Az y_k pontosan ezt fogja megmondani.

Különösen fontos tulajdonsága, hogy ha x periodikus (azaz $x_{j+r} = x_j$ valamely r), akkor y_k csak azon k értékeknél nem nulla, amelyek $\frac{N}{r}$ többszörösei, vagyis a transzformált *kiszúrja* a periódust.

4. Kvantum-Fourier transzformáció [4]

A kvantum-Fourier transzformáció (QFT) lényege, hogy a periódikus struktúrát tartalmazó amplitúdókat olyan állapottá alakítja, ahol a periódus információja kiolvasható.

A QFT lényegében a qubitek egy lineáris transzformációja. (A kvantumos megfelelője a diszkrét Fourier-transzformációnak). A kvantum Fourier-transzformáció a klasszikus diszkrét Fourier-transzformáció, amelyet egy kvantumállapot amplitúdó-vektorára alkalmaznak, és amelynek hosszúsága van. Ha n qubites regiszterre alkalmazzuk akkor: $N = 2^n$

A QFT egy kvantumállapoton működik és kvantumállapotot állít elő.

$|x\rangle$ inputra és $|y\rangle$ outputra:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \, \omega_N^{jk}, \quad k = 0, 1, 2, \ldots, N-1,$$

Ha $|x\rangle$ egy bázisállapot, akkor a QFT egy ilyen leképezésként is le lehet írni:

$$\mathrm{QFT} : |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} \, |k\rangle.$$

A QFT kapu egy unitér kapu, ami kvantum állapotvektorokon dolgozik, s aminek a felső egyenlettel ekvivalens mátrixa: (ahol F_N-t a diszkrét Fourier-transzformáció mátrixának nevezzük)

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

ahol

$$\omega = \omega_N.$$

A kapu unitaritása nagyon fontos, mivel ez azt jelenti, hogy invertálható (létezik QFT^{-1}). Ennek pedig fontos szerepe van a Shor algoritmus működésének szempontjából.

A klasszikus FT $O(N \log N) = O(2^n \cdot n)$ lépést igényel, míg a QFT $O(n^2)$ kvantumkaput. Ez exponenciális gyorsítást jelent.

5. Diádikus racionális fáziskapu

A diadikus racionális fáziskapu egy unitér kvantumkapu, melynek mátrixa:

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{2\pi}{2^k}} \end{pmatrix}$$

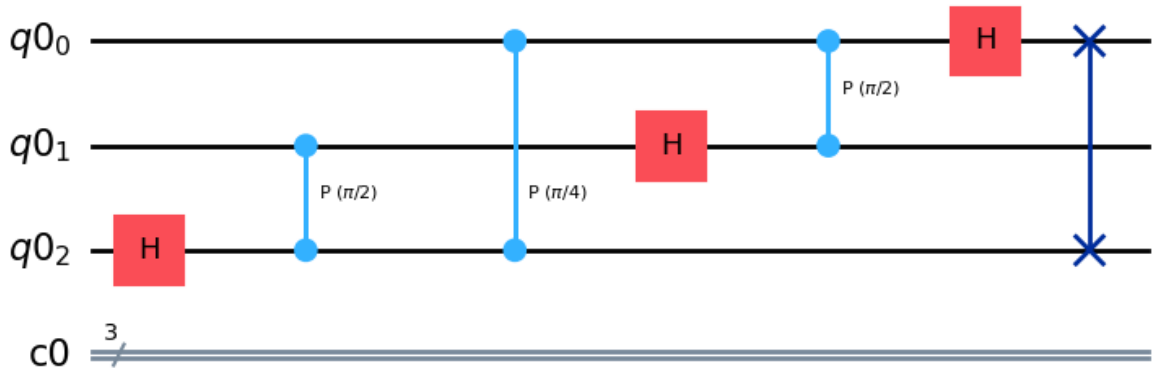
, mely egy k paramétertől függ.

Ez egy, a QFT kapu implementálásához szükséges elemi fázisforgató kapu. Továbbá nem részletezzük.

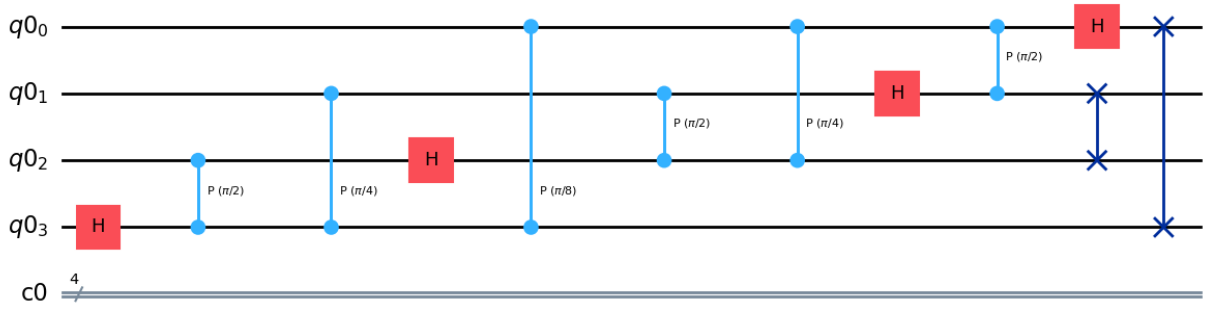
6. QFT kapu implementációja elemi kapukkal [5]

Az elemi kapuk közül kettőre lesz szükség: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ és $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix}$ (Hadamard és a diádikus racionális).

Ezen felül a SWAP művelet sem elhanyagolható, mert a QFT kapunál a bitek fel-le cserélése standard. Az n bemenetes QFT kapu az R_k kapu irányított verzióját alkalmazza:



1. ábra. A QFT kapu 3 bemenetre.



2. ábra. A QFT kapu 4 bemenetre.

A python programot, melyel n bemenetes QFT kaput lehet generálni magunk készítettük. Az következő érhető el: https://afghangoat.hu/imports/src/qiskit_excercises2/QFT_gate_visualizer.py.

7. Shor algoritmusának működése

Shor algoritmusa egy klasszikus problémát (faktorizálás) visszavezet egy perióduskeresési problémára, amelyet egy kvantumszámítógép hatékonyan meg tud oldani a kvantum Fourier-transzformáció segítségével.

Ezt az algoritmust Peter Shor fejlesztette ki 1994-ben. Az algoritmus egész számok faktorizálására lett kitalálva, mely polinomiális időben működik, ellentétben a klasszikus algoritmusokkal, melyek exponenciális időben tudják ezt a feladatot elvégezni.

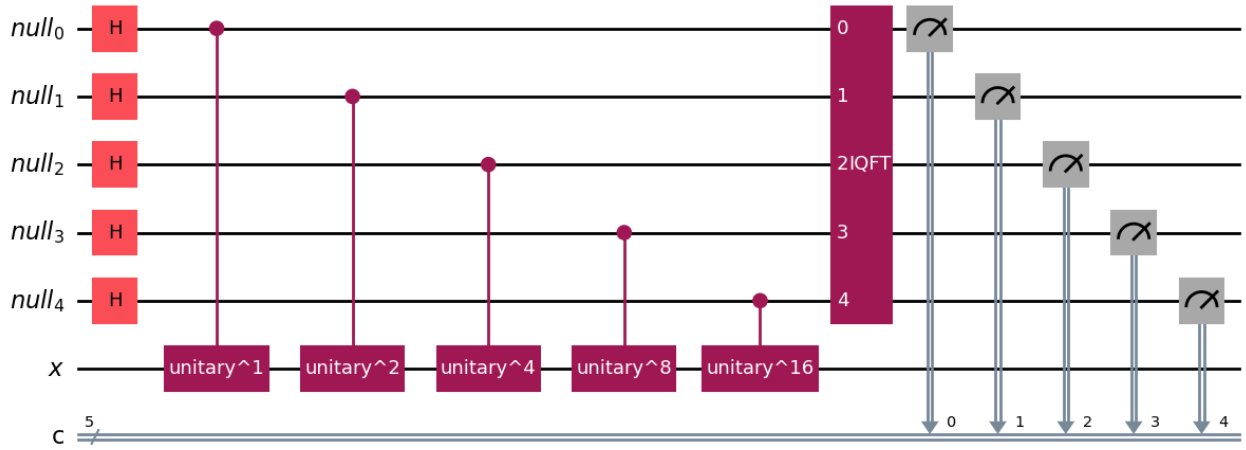
7.1. Fázisbecslési probléma

A fázisbecslés algoritmus segítségével az M_a operátor sajátértékeiből kinyerhető a periódus r.

A probléma arról szól, hogy adott egy $|x\rangle$ kvantumállapot, mely n qubitből áll és egy unitér kvantumrendszer amely n qubiten dolgozik. Azt várjuk, hogy $|x\rangle$ egy sajátvektora az U unitér mátrixnak, mely a kvantumrendszert leírja.

A feladat az, hogy meghatározzuk (megbecsüljük) azt a sajátértéket: $\lambda = e^{2\pi i\theta}$ amihez a $|x\rangle$ kvantumállapot tartozik. Egy olyan θ érték kell, ami a $[0, 1]$ tartományba esik és kielégíti az alábbi egyenletet: $U|x\rangle = e^{2\pi i\theta}|x\rangle$. Ennek a fázisbecslő rendszernek az a célja, hogy m qubiten megbecsülje a θ értékét.

Matematikailag így is felírható: Egy olyan y értéket kell keresni amelyre igaz, hogy $\theta \approx y/2^m$ ahol $y \in 0, 1, 2, 3, ...2^{m-1}$.



3. ábra. x jelöli a x-et és a $null_0, null_1 \ldots, null_m$ pedig az m bitnyi ket nullokat. Itt $m = 5$.

Kód a fentebbi áramkörhöz: https://afghangoat.hu/imports/src/qiskit_excercises2/Phase_estimation.py.

Ez a megvalósítás H kapukkal először csinál teljesen egyenletes szuperpozíciójú kubiteket, majd a $|x\rangle$-el alkalmaz egy irányított unitér kapukat, pontosabban azoknak a 2 hatványai mennyiségűt egymás után (U^{2^j}. (U sajátvektorát $|x\rangle$ tárolja.

7.2. Order finding problem

Legyen egy N szám, amit faktorizálni szeretnénk. Ehhez van egy a szám, ami relatív prím N-hez.

1. Definíció. *A rend az a legkisebb, pozitív r egész szám, amire igaz, hogy* $a^r \equiv 1 \mod N$

Ebből az következik, hogy ha valaki r birtokában van, nagy eséllyel már tudja faktorizálni N-t. Példa:

$$N = 15$$

$$a = 2$$

$$r = 4$$

Ezt a tulajdonságot kell kihasználni:

$$\gcd\left(a^{r/2} \pm 1, N\right)$$

$$a^{r/2} = 2^2 = 4$$

$$\gcd(4 - 1, 15) = 3$$

$$\gcd(4 + 1, 15) = 5$$

Viszont az r meghatározása klasszikus számítógépekkel nagyon nehéz. A kvantumszámítógép ezt a problémát perioduskeresésként oldja meg. Ennek pontosabb leírása megtalálható az IBM Shor

algoritmusánál található leírásban. [6] De a kulcsötlet az, hogy egy olyan függvény mint a következő: $f(x) = a^x \mod N$, periodikus és periódusa pedig pont az r.

Erre pedig pont jó a fentebb taglalt fáziselemzési probléma. Felhasználásának lépései és egy megfelelő algoritmus kiépítése:

1. A quantum fázisbecslés segítségével szuperpozícióba rak sok x kubitet.
2. Egyszerre ki kell számolni az $f(x)$-eket (kvantumpárhuzamosság felhasználásával egyszerű).
3. A periódust ki lehet nyerni a kimenetből (utófeldolgozás).

Ehhez definiáljunk egy M_a operátort, amire igaz, hogy $M_a|x\rangle = |ax \mod N\rangle$ Az M_a operátor egy unitér operátor, mivel permutációként viselkedik a bázisállapotokon.

Ezekhez tartozik egy fázis, $\omega_r^j = e^{2\pi ij/r}$

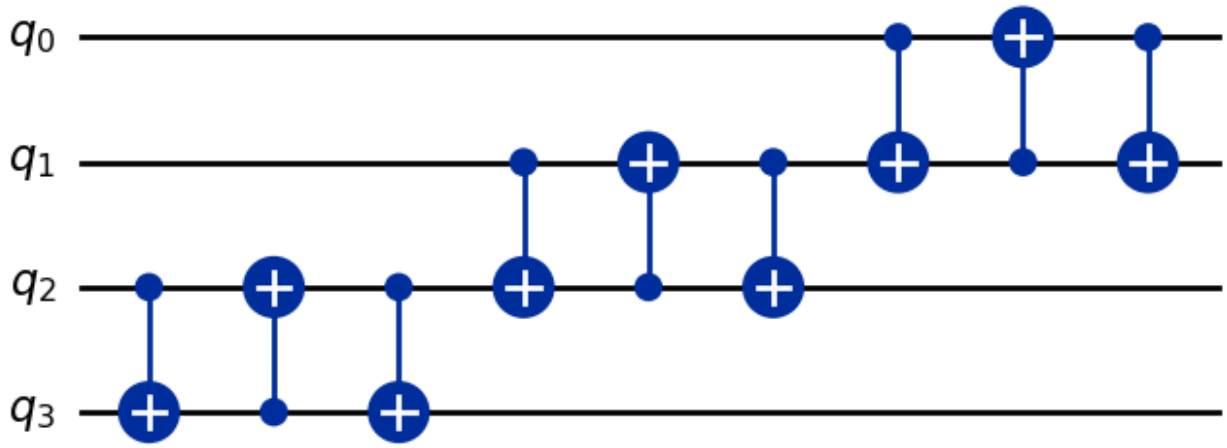
Így a rendszer j/r-t adja vissza, amiből meg lehet kapni r-t.

7.3. A probléma kvantumrendszerbe való átalakítása

Az alábbi kvantumrendszer egy példa, amely $N = 15$ és $a = 2$-re fog működni. Ehhez egy moduláris hatványozó kaput kell létrehozni, melyet M_2^k fog jelölni. Ennek az alábbi kimeneteit várjuk el (például):

$M|0\rangle = |0\rangle$, $M|5\rangle = |10\rangle$, $M|10\rangle = |5\rangle$, ...

Ilyen viselkedést 4 qubiten SWAP kapukkal lehet megvalósítani:



4. ábra. A képen már a Qiskit "decompose"-olt állapot van.

Kód a fentebbi áramkörhöz: https://afghangoat.hu/imports/src/qiskit_excercises2/Modexp.py.

7.4. Shor algoritmus pszeudokódja

1. Egy véletlen $a < N$ szám kiválasztása.
2. $\gcd(a, N)$ kiszámolása:
   - Ha $\gcd(a, N) \neq 1$, akkor nem triviális faktort találtunk, és az algoritmus véget ér.

3. Kvantumos rész (perióduskeresés):
   (a) Szuperpozíció létrehozása az első regiszteren.
   (b) Moduláris hatványozás alkalmazása:
   $$|x\rangle \mapsto |a^x \mod N\rangle$$
   (c) QFT alkalmazása.
4. Első regiszter megmérése, miután egy y értéket kapunk, amely közelíti:
   $$\frac{y}{2^m} \approx \frac{s}{r}$$
5. Klasszikus utófeldolgozás:
   - r meghatározása törtközelítéssel (pl.: lánctörtekkel [7]).
6. Ha r páros és $a^{r/2} \not\equiv -1 \mod N$, akkor:
   $$\gcd(a^{r/2} \pm 1, N)$$
   nem triviális faktort ad.
7. Ellenkező esetben újabb a-t kell választani, és az algoritmus futtatását megismételni.

Az algoritmus hatékonyságát a QFT biztosítja, amely lehetővé teszi a periódus polinomiális időben történő meghatározását.

8. RSA kód feltörése az algoritmussal

Shor algoritmusa igazából az RSA lényegi részének a gyors visszafejtését teszi lehetővé. Az algoritmus polinomiális időben tudja meghatározni az RSA kulcs csere N komponensének a p és q prímszám összetevőit (faktorizálás).

8.1. Egyszerű példa RSA feltörésére Shor algoritmusával

Tekintsünk egy egyszerű RSA példát, ahol a nyilvános kulcs része:

$$N = 15$$

Tudjuk, hogy:

$$N = p \cdot q$$

de p és q ismeretlenek.
Cél: meghatározni p-t és q-t Shor algoritmusával.

1. Válasszunk egy $a < N$ számot
Legyen:

$$a = 2$$

$$\gcd(2, 15) = 1$$

Tehát folytathatjuk az algoritmust.

2. Definiáljuk a függvényt

$$f(x) = 2^x \mod 15$$

Számoljuk ki néhány értékét:

$$f(0) = 1, \quad f(1) = 2, \quad f(2) = 4, \quad f(3) = 8, \quad f(4) = 1$$

Látható, hogy a függvény periódusa:

$$r = 4$$

3. Kvantumos rész

A kvantumszámítógép a kvantum Fourier-transzformáció segítségével meghatározza ezt a periódust ($r = 4$) hatékonyan.

4. Faktorizálás

Mivel r páros, kiszámoljuk:

$$a^{r/2} = 2^2 = 4$$

$$\gcd(4 - 1, 15) = \gcd(3, 15) = 3$$

$$\gcd(4 + 1, 15) = \gcd(5, 15) = 5$$

5. Eredmény

$$N = 15 = 3 \cdot 5$$

Tehát sikeresen faktorizáltuk N-t.

Megjegyzés: Valós RSA rendszerekben N több száz vagy ezer bites szám, ahol klasszikus módszerekkel a faktorizálás rendkívül nehéz, míg Shor algoritmusa ezt elméletben polinomiális időben képes elvégezni.

9. Továbblépési lehetőségek

A Shor-algoritmus egy kellően nagy kvantumszámítógépen képes lenne feltörni az RSA-t és hasonló kriptográfiai rendszereket is. Ez komoly kihívást jelent a modern információbiztonságra nézve, mert egy fajta, úgynevezett *harvest now, decrypt later* támadás már ma is releváns: egy támadó titkosított tartalmakat gyűjthet most, és visszafejtheti majd a jövőben egy kellően erős kvantumszámítógéppel. [8] Kettő módon lehet védekezni: *posztkvantum kriptográfiával*, amely klasszikus számítógépeken fut, de ilyen kvantumtámadásoknak is ellenáll, illetve *kvantumkulcs-elosztással* (QKD), amely magát a kvantummechanikát használja a biztonság garantálására.

9.1. Posztkvantum kriptográfia

A posztkvantum kriptográfia célja olyan kriptográfiai rendszerek tervezése, amelyek mind klasszikus, mind kvantumszámítógépekkel szemben biztonságosak maradnak. A biztonság alapja minden esetben egy olyan matematikai probléma, amelyre nem ismert hatékony kvantumalgoritmus. A NIST (National Institute of Standards and Technology) 2022-ben négy algoritmust fogadott el szabványként. Ezeknek három kategóriája van: [9] [10]

1. Lattice-based cryptography.
2. Hash-alapú aláírások.
3. Kódalapú kriptográfia.

A modern webes titkosítás éppen egy átmeneti állapotban van: a kapcsolat kulcscseréje egyszerre támaszkodik egy klasszikus és egy posztkvantum algoritmusra. [11] A bevezetés már zajlik: 2025 márciusára a Cloudflare hálózatán az emberi HTTPS-forgalom körülbelül 38%-a hibrid posztkvantum kézfogást használt. [12]

9.2. Kvantumkulcs-elosztás (QKD)

Az alapötlet a következő: két fél kvantumállapotokat (tipikusan fotonokat) küld egymásnak egy kvantumcsatornán. A kvantummechanika *No Cloning tétele* kimondja, hogy ismeretlen kvantumállapotot nem lehet lemásolni. Ebből következik, hogy ha egy harmadik fél megpróbálja lehallgatni a csatornát, szükségszerűen megzavarja a kvantumállapotokat, mely zavarás detektálható.

BB84 protokoll. Az első QKD-protokollt Bennett és Brassard írta le 1984-ben, ez a BB84 protokoll. Az egyik fél (Alice) véletlenszerű biteket küld fotonok polarizációjaként, két különböző bázist ($+$ és $\times$) véletlenszerűen váltogatva. A másik fél (Bob) szintén véletlenszerűen választ mérési bázist. Ezután egy nyilvános csatornán egyeztetik, hogy melyik mérésnél egyezett a bázisuk; ezek a bitek alkotják a közös kulcsot. Ha egy támadó (Eve) lehallgatta a csatornát, a báziseltérésekből adódó hibaarány megnő, és ez detektálható.
A biztonság tehát nem azon múlik, hogy Eve nem tud elég gyorsan számolni, hanem azon, hogy *fizikailag nem tudja megmérni az állapotot anélkül, hogy megzavarná.*

Hivatkozások

[1] Nyilvános kulcsú titkosítás és RSA algoritmus,
https://cs.bme.hu/bsz1/jegyzet/bsz1_jegyzet.pdf, 2026.

[2] NFS faktorizáló algoritmus,
https://en.wikipedia.org/wiki/General_number_field_sieve, 2026.

[3] Klasszikus Fourier-transzformáció,
https://math.bme.hu/~tasnadi/merninf_anal_2/Fourier-transzform%C3%A1ci%C3%B, 2026.

[4] Quantum fourier transform – Wikipedia,
https://en.wikipedia.org/wiki/Quantum_Fourier_transform, 2026.

[5] Implementing the QFT with Qiskit,
https://leftasexercise.com/2019/02/25/implementing-the-quantum-fourier-transform-with-qiskit/, 2026.

[6] IBM Qiskit Shor algoritmusa,
https://quantum.cloud.ibm.com/docs/en/tutorials/shors-algorithm, 2026.

[7] Lánctört,
https://hu.wikipedia.org/wiki/L%C3%A1nct%C3%B6rt#L%C3%A1nct%C3%B6rtbe_fejt%C3%A9s/, 2026.

[8] HNDL,
https://en.wikipedia.org/wiki/Harvest_now,_decrypt_later, 2026.

[9] Általános posztkvantum kriptográfiai megoldások,
https://youtu.be/_MoRcYLN-7U?si=GtikFz9c-TwN94-r, 2026.

[10] NIST PQC,
https://csrc.nist.gov/projects/post-quantum-cryptography, 2026.

[11] TLS1.3,
https://postquantum.com/post-quantum/infrastructure-challenges-pqc/,
2026.

[12] TLS1.3 metrika és ajánlások,
https://www.intelligentliving.co/quantum-hybrid-tls-ml-kem-browser/,
2026.