Eötvös Loránd Tudományegyetem
Természettudományi Kar

Poszt-kvantum kriptográfia

Kinorányi Dóra

Témavezető:
Szabó István

Matematika BSc - Szakdolgozat
Matematikai elemző szakirány

Budapest
2020

NYILATKOZAT

Név: Kinoránvi Dóra

ELTE Természettudományi Kar, szak: Matematika BSc

NEPTUN azonosító: UC9WXR

Szakdolgozat címe:
Poszt-kvantum kriptográfia

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2020.12.14.

a hallgató aláírása

Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek, Szabó Istvánnak, aki először adatvédelem tárgyon keltette fel az érdeklődésemet a kriptográfiával kapcsolatban. Ezután is további tárgyakat, illetve anyagokat ajánlott, és a szakdolgozat írása során körültekintő válaszaival és útmutatásával támogatta annak elkészültét.

Hálás vagyok továbbá a családomnak és Szabinak, akik az egyetemi képzés során segítettek engem, és mellettem voltak.

Tartalomjegyzék

1. Bevezetés **1**
1.1. Miért szükséges a poszt-kvantum kriptográfia? 1
1.2. Poszt-kvantum kriptográfiai kilátások 2
1.3. A dolgozat felépítése 2

2. Kvantumszámítási háttér **3**
2.1. Fogalmak 3
2.2. Kvantumbitek 5
2.3. Hadamard bázis 7
2.4. A kvantumszámítógépek modellezése 9

3. Kriptográfia és a kvantumszámítás rá gyakorolt hatása **12**
3.1. A kvantumszámítás hatásai 12
3.2. Kriptográfiai fogalmak 12
3.3. Szimmetrikus és aszimmetrikus kulcsú kriptográfia 13

4. Prímfaktorizáció kvantumszámítógépen **14**
4.1. Az RSA biztonsága 14
4.2. A prímfaktorizáció számelméleti háttere 16
4.3. Shor algoritmus 17

5. McEliece kriptorendszer **19**
5.1. Véges testek és polinomgyűrűk faktorizálása 19
5.2. Kódelméleti fogalmak 22
5.3. A McEliece kriptorendszer 23
5.4. Goppa kódok 26

6. NTRU **29**
6.1. Jelölések 29
6.2. NTRUEncrypt 30
6.3. Rácselméleti háttér 31
6.4. LLL algoritmus 32
6.5. Az NTRU mint SVP 34
6.6. Példa az NTRU algoritmus számolására 37

Függelékek **40**

A. a 6.4.9 példabeli számolás részletei **40**

B. 6.6 példa: f_p visszaszámolása **41**

Hivatkozások **42**

1. Bevezetés

1.1. Miért szükséges a poszt-kvantum kriptográfia?

1.1.1. Megjegyzés. Ezen alfejezet a NISTIR 8105: Report on Post-Quantum Cryptography [10] 2016-os cikk alapján íródott.

Az elmúlt évtizedek során a nyilvános kulcsú kriptográfia elhanyagolhatatlan részévé vált a globális kommunikációnak. Ennek segítségével működik biztonságosan például a gazdaság, az elektronikus kereskedelem, a mobiltelefonjaink, a közösségi hálózatok, és a felhő alapú számítástechnika.

A legfontosabb kommunikációs protokollok alapjait többnyire három kriptográfiai eljárás adja: a nyilvános kulcsú titkosítás, a digitális aláírás, és a kulcscsere. A gyakorlatban ezek implementálásához jellemzően a Diffie-Hellmann kulcscserét, az RSA titkosítórendszert, és az elliptikus görbéken alapuló titkosítást használják. Ezek biztonságát bizonyos számelméleti problémák bonyolultsága adja, úgymint a prímfaktorizáció, vagy a diszkrét logaritmus probléma.

1994-ben Peter Shor (Bell Laboratories) belátta, hogy az erős kvantumszámítógépek jelentős számítási előnyre tesznek szert, és ezen problémákat hatékonyan meg tudják oldani. Ezzel a modern kommunikáció számos formáját veszélybe fogják sodorni.

Shor felfedezése óta a kvantumalgoritmusok elmélete sokat fejlődött. Az alábbi táblázatban látható, hogy a nagyméretű kvantumszámítógépek hogyan hatnak az egyes klasszikus kriptográfiai eljárásokra.

Algoritmus	Típus	Alkalmazás	Kvantumszámítógép hatása
AES	Szimmetrikus kulcsú	Titkosítás	Nagyobb kulcsméret kell
SHA-2, SHA-3	—	Hashelés	Nagyobb kimeneti méret kell
RSA	Nyilvános Kulcsú	Aláírás, kulcsok létrehozása	Nem biztonságos többé
ECDSA, ECDH	Nyilvános Kulcsú	Aláírás, kulcscsere	Nem biztonságos többé
DSA	Nyilvános Kulcsú	Aláírás	Nem biztonságos többé

1.1.2. Megjegyzés. ECDSA, ECDH: elliptikus görbéken alapuló kriptográfia, DSA: Digital Signature Algorithm.

Ugyan a kvantumszámítógépek fizikai megvalósíthatóságára korábban kevés rálátás volt, manapság sok tudós gondolja úgy, hogy a következő 20 év során hatékony kvantumszámítógép lesz építhető, és ezzel szinte az összes jelenleg használt nyilvános kulcsú rejtjelezés törhetővé válik. A kvantumbiztos kommunikációra való átállás várhatóan jelentős erőfeszítést fog igényelni, így attól függetlenül, hogy nem becsülhető pontosan a kvantumkorszak eljövetele, érdemes már most elkezdeni ellenállóvá fejleszteni a biztonsági rendszereket. A tudomány ezen területe jelenleg is aktív kutatás alatt van, ezt nevezik poszt-kvantum kriptográfiának.

1.2. Poszt-kvantum kriptográfiai kilátások

Az alfejezet a Nicolas Sendrier, Raphael Overbeck: Code-Based Cryptography [18] cikk alapján íródott.

A következő kriptográfiai területek egyelőre biztonságosnak reméltek a klasszikus- és kvantumtámadások ellen. Mindegyik esetén megemlítünk egy-egy nevezetes módszert.

- **Hash-alapú kritográfia** Merkle-féle hash-fa alapú nyilvános kulcsú aláírás (1979).
- **Kódalapú kriptográfia** McEliece-féle titkosító rendszer (1978).
- **Rácsalapú kriptográfia** NTRU (1998).
- **Többváltozós kvadratikus egyenleteken alapuló kriptográfia** HFE (1996).
- **Titkos kulcs alapú kriptográfia** AES (1998).

Egyelőre még nem találták módját, hogy hogyan lehetne alkalmazni a Shor algoritmust - mellyel törhető az RSA, DSA és a ECDSA - a példabeli titkosítások feltörésére. A Shor algoritmust részletesebben tárgyalni fogjuk a 4.3 alfejezetben. Egy másik kvantumalgoritmus, a Grover algoritmus ugyan hatékony néhány fenti algoritmussal szemben, azonban ez nem olyan gyors, mint a Shor algoritmus, és így könnyen lehet védekezni ellene nagyobb kulcsméret használatával.

1.3. A dolgozat felépítése

A dolgozat célja, hogy bemutasson poszt-kvantum kriptográfiai elméleteket és eljárásokat, továbbá, hogy azok működését példák segítségével tegye szemléletesebbé.

Ehhez a 2. fejezetben átnézzük a kvantumszámítási alapokat, úgymint a kvantumbitekkel való számolást és a kvantumszámítógépek modellezését Hilbert terek segítségével. Ezek lényegesek annak megértésében, hogy miért szükséges újragondolni a biztonságos üzenetküldéshez használt algoritmusokat.

A 3. fejezetben összefoglalunk néhány kriptográfiai fogalmat, melyeket használunk a dolgozat további részében.

A 4. fejezet fókuszában a prímfaktorizációs probléma, az RSA és a Shor algoritmus áll. Részletesen fogunk foglalkozni a Shor algoritmus alapötletével, ehhez egy számelméleti bizonyítást is leírunk.

Az utolsó 2 fejezetben egy-egy ígéretesnek ígérkező poszt-kvantum eljárásról adunk összefoglalást. Az 5. fejezetben a McEliece kriptorendszert elemezzük. Ehhez fontosak lesznek az alapvetőbb algebrai kódelméleti fogalmak, illetve a bináris Goppa kódok elméletének áttekintése.

A 6. fejezetben az NTRU rendszerhez szükséges rácselméleti háttér összefoglalásával kezdünk, majd folytatjuk az NTRU titkosító eljárás lépéseivel, illetve egy lehetséges támadási móddal. A fejezetet egy példával zárjuk.

2. Kvantumszámítási háttér

Ebben a fejezetben áttekintjük a fontosabb kvantumszámítási fogalmakat és jelöléseket, majd Hilbert terek segítségével modellezzük a kvantumszámítógépek működését. Ehhez többnyire Stephanie Wehner and Nelly Ng: edX Quantum Cryptography [3] kurzus 1. fejezetét, és Michael A. Nielsen & Isaac L. Chuang: Quantum Computation and Quantum Information könyv [1] 1.2 fejezetét dolgozzuk fel, az ettől eltérő részeket megjegyzésben jelölni fogjuk.

2.1. Fogalmak

2.1.1. Definíció (Komplex konjugált)**.** Egy $z = a + bi \in \mathbb{C}$ szám komplex konjugáltja $\overline{z} = a - bi$, ahol $a, b \in \mathbb{R}$ és $i = \sqrt{-1}$.

2.1.2. Definíció (Braket-jelölés)**.** A ket egy n-dimenziós oszlopvektor a $\mathbb{C}^n$ komplex vektortérben, melyet $|\cdot\rangle$ -vel jelölünk. A ket vektorhoz tartozó bra vektor egy n-dimenziós sorvektor, amely megegyezik a hozzá tartozó ket vektor adjungáltjával (konjugált transzponáltjával), melyet $\langle\cdot|$-vel jelölünk, tehát $\langle\cdot| = \left(\overline{|\cdot\rangle}\right)^T$.

2.1.3. Definíció (Abszolút érték)**.** Egy $z = a + bi \in \mathbb{C}$ komplex szám abszolút értéke $|z| = \sqrt{\overline{z} \cdot z} = \sqrt{a^2 + b^2}$.

2.1.4. Definíció (Skaláris szorzat)**.** Legyen E egy vektortér a $\mathbb{C}$ test felett. Ekkor egy $\langle\cdot|\cdot\rangle : E \times E \longrightarrow \mathbb{C}$ leképezést E feletti skaláris szorzatnak nevezünk, ha teljesülnek a következők:

1. $\forall\, \varphi \in E$ esetén $\langle\varphi|\cdot\rangle : E \longrightarrow \mathbb{C}$ függvény lineáris;
2. $\forall\, \varphi, \psi, \chi \in E$ esetén $\langle\varphi + \psi|\chi\rangle = \langle\varphi|\chi\rangle + \langle\psi|\chi\rangle$ és $\lambda \in \mathbb{C}$ esetén $\langle\lambda\varphi|\chi\rangle = \overline{\lambda}\langle\varphi|\chi\rangle$;
3. $\forall\, \psi, \varphi \in E$ esetén $\langle\psi|\varphi\rangle = \overline{\langle\varphi|\psi\rangle}$;
4. $\forall\, \psi \in E$ esetén $\langle\psi|\psi\rangle \in \mathbb{R}$ és $\langle\psi|\psi\rangle \geq 0$, továbbá $\langle\psi|\psi\rangle = 0 \iff \psi = 0$.

2.1.5. Állítás. Tekintsük a $\mathbb{C}^n$ vektorteret, és tetszőleges $\psi, \varphi \in \mathbb{C}^n$ ($n \in \mathbb{N}$) vektorokat. Ekkor az $\langle\psi|\varphi\rangle = \sum_{k=1}^{n} \overline{\psi_k}\varphi_k$ függvény skaláris szorzat.

Bizonyítás. Vegyünk egy tetszőleges $\chi \in \mathbb{C}^n$ vektort, és $\lambda \in \mathbb{C}$ számot, és ekkor érvényesek a következők:

$$\langle\psi|\varphi + \chi\rangle = \sum_{k=1}^{n} \overline{\psi_k}(\varphi_k + \chi_k) = \sum_{k=1}^{n} \overline{\psi_k}\varphi_k + \sum_{k=1}^{n} \overline{\psi_k}\chi_k = \langle\psi|\varphi\rangle + \langle\varphi|\chi\rangle$$

$$\langle\psi|\lambda\varphi\rangle = \sum_{k=1}^{n} \overline{\psi_k}\lambda\varphi_k = \lambda \sum_{k=1}^{n} \overline{\psi_k}\varphi_k = \lambda\langle\psi|\varphi\rangle$$

Tehát az állításban meghatározott függvény lineáris a második változójában.
Az első változóban érvényes:

$$\langle\varphi + \psi|\chi\rangle = \sum_{k=1}^{n} \overline{(\varphi_k + \psi_k)}\chi_k = \sum_{k=1}^{n} (\overline{\varphi_k} + \overline{\psi_k})\chi_k = \sum_{k=1}^{n} \overline{\varphi_k}\chi_k + \sum_{k=1}^{n} \overline{\psi_k}\chi_k = \langle\varphi|\chi\rangle + \langle\psi|\chi\rangle$$

Illetve:

$$\langle \lambda\varphi|\psi\rangle = \sum_{k=1}^{n} \overline{\lambda\varphi_k}\psi_k = \overline{\lambda}\sum_{k=1}^{n}\overline{\varphi_k}\psi_k = \overline{\lambda}\langle\varphi|\psi\rangle$$

Mivel a komplex konjugálás ismételt elvégzése az eredeti értéket adja vissza, azaz $\overline{\overline{\chi}} = \chi$ tetszőleges $\chi \in \mathbb{C}$ esetén, és mivel az összeg konjugáltja a konjugáltak összege, ezért

$$\overline{\langle\varphi|\psi\rangle} = \overline{\sum_{k=1}^{n}\overline{\varphi}\psi} = \sum_{k=1}^{n}\varphi\overline{\psi} = \langle\psi|\varphi\rangle$$

igazolja a harmadik feltételt, az utolsóhoz pedig megfigyelhető, hogy tetszőleges

$$\psi = a + bi \in \mathbb{C},\ a, b \in \mathbb{R} \text{ esetén } \overline{\psi}\psi = (a+bi)(a-bi) = a^2 + b^2 \geq 0,$$

hiszen négyzetszámok összegét tekintjük, és az összeg pontosan akkor minimális, vagyis 0, ha $a = b = 0$.

□

2.1.6. Megjegyzés. A következőkben a $\langle\psi|\varphi\rangle = \sum_{k=1}^{n}\overline{\psi_k}\varphi_k$ skaláris szorzattal fogunk számolni.

2.1.7. Megjegyzés. A skaláris szorzatot szokás belső szorzatnak is nevezni. $|\psi\rangle$, $|\varphi\rangle \in \mathbb{C}^n$ esetén skaláris szorzatuk komplex szám lesz.

2.1.8. Példa. $|\psi\rangle = \begin{pmatrix} 1+2i \\ 3 \end{pmatrix}$, $|\varphi\rangle = \begin{pmatrix} 2+2i \\ i \end{pmatrix}$, $\langle\psi|\varphi\rangle = ?$

$$\langle\psi| = \begin{pmatrix} \overline{1+2i} \\ \overline{3} \end{pmatrix}^T = \begin{pmatrix} 1-2i \\ 3 \end{pmatrix}^T = \begin{pmatrix} 1-2i, & 3 \end{pmatrix}$$

$$\langle\psi|\varphi\rangle = \begin{pmatrix} 1-2i, & 3 \end{pmatrix} \cdot \begin{pmatrix} 2+2i \\ i \end{pmatrix} = (1-2i)(2+2i) + 3i = 6 - 2i + 3i = 6 + i$$

2.1.9. Állítás. $|\langle\psi|\varphi\rangle|^2 = \langle\varphi|\psi\rangle\langle\psi|\varphi\rangle$.

Bizonyítás. Az abszolút érték definícióját és a $\overline{\langle\varphi|\psi\rangle} = \langle\psi|\varphi\rangle$ összefüggést felhasználva érvényes a következő:

$$|\langle\psi|\varphi\rangle|^2 = \left(\sqrt{\overline{(\langle\psi|\varphi\rangle)}\cdot\langle\psi|\varphi\rangle}\right)^2 = \overline{(\langle\psi|\varphi\rangle)}\langle\psi|\varphi\rangle = \langle\varphi|\psi\rangle\langle\psi|\varphi\rangle$$

□

2.1.10. Definíció (Ket vektor hossza)**.** Tekintsük a $|\psi\rangle = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_n \end{pmatrix}$ ket vektort. Ekkor a $|\psi\rangle$ ket vektor hosszát meghatározza a következő képlet.

$$||\,|\psi\rangle||_2 = \sqrt{\langle\psi|\psi\rangle} = \sqrt{\sum_{i=1}^{n}\overline{\psi_i}\psi_i} = \sqrt{\sum_{i=1}^{n}|\psi_i|^2}$$

2.1.11. Definíció (Normalizált vektor)**.** Ha $||\,|\psi\rangle||_2 = 1$, akkor $|\psi\rangle$ normalizált.

2.1.12. Példa. Tekintsük $|\psi\rangle = \frac{1}{2}\begin{pmatrix}1+i\\1-i\end{pmatrix} \in \mathbb{C}^2$ ket vektort.
Ekkor a hozzá tartozó bra vektor $\langle\psi| = \frac{1}{2}\begin{pmatrix}1-i, & 1+i\end{pmatrix}$ lesz, és $|\psi\rangle$ hossza:

$$||\,|\psi\rangle||_2 = \sqrt{\langle\psi|\psi\rangle} = \sqrt{\sum_{i=1}^{n}|\psi_i|^2} = \left(\frac{1}{2}\sqrt{1+1}\right)^2 + \left(\frac{1}{2}\sqrt{1+1}\right)^2 = 2\cdot\frac{1}{4}\cdot 2 = 1$$

Tehát $|\psi\rangle$ normalizált.

2.1.13. Definíció (Vektorok tenzorszorzata)**.** Legyenek $|\psi\rangle \in \mathbb{C}^n$, $|\varphi\rangle \in \mathbb{C}^m$ vektorok, ekkor ezek tenzorszorzata:

$$|\psi\rangle \otimes |\varphi\rangle = \begin{pmatrix}\psi_1\\\vdots\\\psi_n\end{pmatrix} \otimes \begin{pmatrix}\varphi_1\\\vdots\\\varphi_m\end{pmatrix} = \begin{pmatrix}\psi_1|\varphi\rangle\\\vdots\\\psi_n|\varphi\rangle\end{pmatrix} = \begin{pmatrix}\psi_1\varphi_1\\\vdots\\\psi_1\varphi_m\\\psi_2\varphi_1\\\vdots\\\psi_n\varphi_m\end{pmatrix} \in \mathbb{C}^{n\cdot m}$$

2.1.14. Állítás (Tulajdonságok)**.** A tenzorszorzatra teljesülnek a következő tulajdonságok:

1. Disztributív:
$$|\psi\rangle \otimes (|\varphi\rangle + |\chi\rangle) = |\psi\rangle \otimes |\varphi\rangle + |\psi\rangle \otimes |\chi\rangle, \text{ és}$$
$$(|\psi\rangle + |\varphi\rangle) \otimes |\chi\rangle = |\psi\rangle \otimes |\chi\rangle + |\varphi\rangle \otimes |\chi\rangle$$
2. Asszociatív:
$$(|\psi\rangle \otimes |\varphi\rangle) \otimes |\chi\rangle = |\psi\rangle \otimes (|\varphi\rangle \otimes |\chi\rangle)$$
3. Nem kommutatív:
$$|\psi\rangle \otimes |\varphi\rangle \neq |\varphi\rangle \otimes |\psi\rangle, \text{ ha } |\psi\rangle \neq |\varphi\rangle$$

2.1.15. Megjegyzés. A tulajdonságok ket és bra vektorok esetén is teljesülnek.

2.2. Kvantumbitek

Mik a kvantumbitek?

A klasszikus számítástudomány egyik legalapvetőbb fogalma a bit. A kvantumszámítás elmélete is hasonló egységen nyugszik, a kvantumvilágban ezt kvantumbitnek, vagy röviden kubitnak nevezzük. Ebben a részben áttekintjük a kubitek tulajdonságait, és azok hasonlóságait, illetve különbözőségeit a klasszikus bitekkel szemben.

A szakdolgozat során a kvantumbitekre többnyire absztrakt matematikai objektumokként fogunk tekinteni. A klasszikus számításaink során egy bit kétféle értéket tudott tárolni, a 0-t, és az 1-et. Ehhez hasonlóan egy kvantumbiten is tudunk információt tárolni, ezt a tárolt információt fogjuk a kvantumbit állapotának nevezni. A klasszikus 0 és 1 bitek megfeleltethetők a $0 \to |0\rangle$ és az $1 \to |1\rangle$ állapotokkal. A $|\cdot\rangle$ jelölést Dirac-jelölésnek nevezzük. Ez az állapotok standard jelölése a kvantummechanikában.

2.2.1. Definíció (Standard bázis vagy számítási bázis)**.** Tekintsük a kétdimenziós komplex $\mathbb{C}^2$ vektorteret. Ennek a $\{|0\rangle, |1\rangle\}$ egy ortonormált bázisa, ahol $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ és $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$. Ezt számítási bázisnak, vagy más néven standard bázisnak nevezzük.

Az egyik legjelentősebb különbség a klasszikus és a kvantumbitek között, hogy a szuperpozícióban lévő kvantumbitek bizonyos valószínűséggel a $|0\rangle$ és a $|1\rangle$ értéket is felveheti. Ennek megfelelően lehetséges, hogy a kubit az említett állapotok affin kombinációjában legyen.

2.2.2. Definíció (Szuperpozícióban lévő kubit állapota)**.** A $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ felírás a kvantumbit állapota, ha $|\alpha|^2 + |\beta|^2 = 1$, ahol $\alpha, \beta \in \mathbb{C}$.

2.2.3. Definíció (Amplitúdó)**.** Az α és β együtthatókat a kvantumállapot amplitúdóinak nevezzük.

2.2.4. Megjegyzés. Egy kvantumállapotot, vagy egy több kubitból álló rendszer állapotát többféle alakban is fel lehet írni, ezek a megválasztott bázistól függnek. A fogalmak bevezetéséhez mi a számítási bázist használjuk. A bázisok közül a Hadamard bázist bővebben elemezzük a 2.3 szakaszban.

Kettő kubitból álló rendszer állapota

A több kubitból álló rendszerekről szóló szakaszok Nagy Benedek: Új számítási paradigmák [2] című könyve, és Stephanie Wehner and Nelly Ng: edX Quantum Cryptography [3] előadásjegyzetei alapján íródtak.

Ha van két klasszikus bitünk, 4 lehetséges kombinációjuk fordulhat elő. Ezeknek szintén lehet kvantumállapotokat megfeleltetni: $00 \longrightarrow |00\rangle$, $01 \longrightarrow |01\rangle$, $10 \longrightarrow |10\rangle$, $11 \longrightarrow |11\rangle$.

2.2.5. Definíció (Két kubitból álló rendszerek számítási bázisa)**.** A kétkubitos rendszer számítási bázisa $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, ahol a bázis elemei

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \quad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \quad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

Ezek segítségével fel tudjuk írni a rendszer állapotát:

2.2.6. Definíció (Két kubitbólból álló szuperpozícióban lévő rendszer állapota)**.** A

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

alakú felírás a rendszer állapota, ahol $\sum_{x\in\{0,1\}^2} |\alpha_x|^2 = 1$, vagyis az állapot normalizált. Az $\alpha_x, x \in \{0,1\}^2$ együtthatókat amplitúdóknak is nevezzük.

2.2.7. Megjegyzés. A hozzárendelt vektorok a $\mathbb{C}^4$ vektortérben vannak, ahol a dimenziószám megegyezik a lehetséges klasszikus bitsorozatok számával, vagyis $n = 2^2 = 4$.

2.2.8. Példa (Példa kétbites rendszer állapotára)**.**

$$|\psi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle = \frac{1}{2}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \frac{1}{2}\begin{pmatrix}0\\1\\0\\0\end{pmatrix} + \frac{1}{2}\begin{pmatrix}0\\0\\1\\0\end{pmatrix} + \frac{1}{2}\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Ez valóban lehetséges kvantumállapot, hiszen az amplitúdók négyzetösszege $4 \cdot \left(\frac{1}{2}\right)^2 = 1$.

2.2.9. Definíció (Bell-állapot vagy EPR-pár)**.** Az alábbi képlettel definiáljuk egy kétbites rendszer Bell-állapotát:

$$|EPR\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}\left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\right) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

2.2.10. Megjegyzés. A Bell-állapotot szokás harang állapotnak is nevezni.

n kubitból álló rendszer állapota

2.2.11. Definíció (n kubitból álló rendszerek számítási bázisa)**.** A $\mathbb{C}^{2^n}$ vektortér számítási bázisa $\{|x_1x_2\dots x_n\rangle : x_i \in \{0,1\}, i \in \{1,\dots,n\}\}$, ahol

$$|x_1x_2\dots x_n\rangle = \begin{pmatrix}0\\\vdots\\1\\\vdots\\0\end{pmatrix} \longrightarrow \text{s. koordináta,}$$

melyben s a $x_1x_2\dots x_n$ bináris szám tízes számrendszerbeli értéke, és a vektor egyetlen nemnulla értéke az s. koordinátában van.

2.2.12. Definíció (n kvantumbitből álló szuperpozícióban lévő rendszer állapota)**.** Egy $|\psi\rangle \in \mathbb{C}^{2^n}$ vektor felírható a számítási bázis kvantumállapotainak affin kombinációjaként a következőképpen:

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle, \text{ ahol minden } x \in \{0,1\}^n\text{-re } \alpha_x \in \mathbb{C}, \text{ és } \sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$$

A $|\psi\rangle$ vektort az n bitből álló rendszer kvantumállapotának nevezzük.

2.3. Hadamard bázis

A kvantumállapot fogalmának bevezetése során a számítási bázist használtuk. A $\mathbb{C}^2$ vektortének azonban további bázisai is vannak. A kvantumszámítások során gyakran előkerül a Hadamard bázis, ebben a fejezetben ezt fogjuk megvizsgálni.

2.3.1. Megjegyzés. A Hadamard bázisról szóló alfejezet a Stephanie Wehner and Nelly Ng: edX Quantum Cryptography [3] előadásjegyzetei alapján íródott.

2.3.2. Definíció (Hadamard bázis)**.** A $\mathbb{C}^2$ vektortér Hadamard bázisát $H = \{|+\rangle, |-\rangle\}$-vel jelöljük, melynek két eleme:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$
$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

2.3.3. Példa. Egy kubit állapotára példa a következő:

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.$$

Ha ennek állapotát megmérjük, akkor $|\alpha|^2 = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$ valószínűséggel $|0\rangle$-t, $|\beta|^2 = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$ valószínűséggel $|1\rangle$-et kapunk eredményül.

2.3.4. Állítás. A Hadamard bázis ortonormált bázis a $\mathbb{C}^2$ vektortérben.

Bizonyítás. Először ellenőrizzük, hogy $|+\rangle$ és $|-\rangle$ vektorok normalizáltak.

$$\sqrt{\langle -|-\rangle} = \sqrt{\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}^T \cdot \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}} = \sqrt{\frac{1}{2}\cdot 2} = 1$$

$$\sqrt{\langle +|+\rangle} = \sqrt{\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}^T \cdot \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}} = \sqrt{\frac{1}{2}\cdot 2} = 1$$

Szükséges továbbá, hogy a két vektor merőleges legyen egymásra, vagyis, hogy skaláris szorzatuk 0 legyen.

$$\sqrt{\langle +|-\rangle} = \sqrt{\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}^T \cdot \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}} = \sqrt{\frac{1}{2}\cdot (1-1)} = 0$$

□

2.3.5. Feladat. Írjuk fel $|1\rangle$-t a Hadamard bázisban! Ehhez keressük meg a megfelelő $\alpha, \beta \in \mathbb{C}$ együtthatókat, melyre fennáll: $|1\rangle = \alpha|+\rangle + \beta|-\rangle$, vagyis

$$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix} = \alpha|+\rangle + \beta|-\rangle = \alpha\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} + \beta\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

A vektorokra vonatkozó egynelőség meghatároz egy két egyenletből álló egyenletrendszert, melyben két ismeretlen van.

$$\begin{cases} 0 = \frac{\alpha}{\sqrt{2}} + \frac{\beta}{\sqrt{2}} \\ 1 = \frac{\alpha}{\sqrt{2}} - \frac{\beta}{\sqrt{2}} \end{cases}$$

Ennek megoldásai: $\alpha = \frac{1}{\sqrt{2}}$, $\beta = -\frac{1}{\sqrt{2}}$. Tehát a megoldás:

$$|1\rangle = \frac{1}{\sqrt{2}}|+\rangle - \frac{1}{\sqrt{2}}|-\rangle.$$

2.3.6. Állítás. A számítási bázisban felírt $\alpha|0\rangle + \beta|1\rangle$ állapot Hadamard bázisban felírva $\frac{\alpha+\beta}{\sqrt{2}} \cdot |+\rangle + \frac{\alpha-\beta}{\sqrt{2}} \cdot |-\rangle$ lesz.

Bizonyítás. Fejezzük ki a standard bázisállapotokat $|+\rangle$ és $|-\rangle$ segítségével.

$$|0\rangle = \frac{1}{2} \cdot 2 \cdot |0\rangle = \frac{1}{2}(|0\rangle + |0\rangle) = \frac{1}{2}(|0\rangle + |1\rangle + |0\rangle - |1\rangle) = \frac{1}{2}((|0\rangle + |1\rangle) + (|0\rangle - |1\rangle)) =$$

$$= \frac{1}{2}\left(\sqrt{2}|+\rangle + \sqrt{2}|-\rangle\right) = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$$

$$|1\rangle = \frac{1}{2} \cdot 2 \cdot |1\rangle = \frac{1}{2}(|1\rangle + |1\rangle) = \frac{1}{2}(|0\rangle + |1\rangle - |0\rangle + |1\rangle) = \frac{1}{2}((|0\rangle + |1\rangle) - (|0\rangle - |1\rangle)) =$$

$$= \frac{1}{2}\left(\sqrt{2}|+\rangle - \sqrt{2}|-\rangle\right) = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)$$

Ezeket behelyettesítve kapjuk:

$$\alpha|0\rangle + \beta|1\rangle = \alpha\left(\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)\right) + \beta\left(\frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)\right) =$$

$$= \frac{1}{\sqrt{2}}(\alpha(|+\rangle + |-\rangle) + \beta(|+\rangle - |-\rangle)) = \frac{\alpha+\beta}{\sqrt{2}} \cdot |+\rangle + \frac{\alpha-\beta}{\sqrt{2}} \cdot |-\rangle$$

□

2.3.7. Megjegyzés. Azaz annak a valószínűsége, hogy mérés eredményeként $|+\rangle$-t kapunk $\frac{|\alpha+\beta|^2}{2}$, és annak a valószínűsége, hogy $|-\rangle$-t kapunk $\frac{|\alpha-\beta|^2}{2}$ lesz.

2.4. A kvantumszámítógépek modellezése

Ezen alfejezetet Nicolas Sendrier, Raphael Overbeck: Code-Based Cryptography [18] publikációja alapján dolgoztuk ki.

2.4.1. Definíció (Prehilbert tér)**.** Legyen E vektortér, és $\langle\cdot|\cdot\rangle$ E feletti skaláris szorzat. Ekkor az $(E, \langle\cdot|\cdot\rangle)$ párt prehilbert térnek nevezzük.

2.4.2. Definíció (Metrikus tér)**.** Legyen X egy nemüres halmaz, és $\rho : X \times X \longrightarrow \mathbb{R}_0^+$ függvény, melyre teljesülnek a következők tetszőleges $x, y, z \in X$ esetén:

1. $\rho(x, y) = 0 \iff x = y$
2. $\rho(x, y) = \rho(y, x)$
3. $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$.

Ekkor az (X, ρ) párt metrikus térnek nevezzük.

2.4.3. Megjegyzés. A prehilbert téren a skalárszorzat természetes módon indukál egy metrikát a következőképpen: $\rho(x,y) := \sqrt{\langle x-y|x-y\rangle}$. Ennek ellenőrzése egyszerű számolás. Innen könnyű látni, hogy minden prehilbert tér egyben metrikus tér is.

2.4.4. Definíció (Hilbert tér)**.** Egy $(E, \langle\cdot|\cdot\rangle)$ prehilbert teret Hilbert-térnek nevezünk, ha minden E-beli Cauchy-sorozat konvergens a skalárszorzat által indukált metrika szerint (azaz teljes metrikus tér).

2.4.5. Definíció (Sűrű részhalmaz)**.** Legyen (X,ρ) egy metrikus tér. Ebben egy $Y \subset X$ részhalmazt sűrűnek nevezünk, ha minden $x \in X$ és minden $\varepsilon > 0$ esetén létezik $y \in Y$, hogy $\rho(x,y) < \varepsilon$.

2.4.6. Definíció (Szeparábilis metrikus tér)**.** Egy metrikus teret szeparábilisnek nevezünk, ha van olyan sűrű részhalmaza, mely megszámlálható.

2.4.7. Posztulátum (A kvantummechanika első posztulátuma [4])**.** Minden fizikai rendszerhez hozzárendelhető egy komplex, szeparábilis Hilbert tér. Ezt a teret a fizikai térhez tartozó állapottérnek nevezzük. Ha egy kvantummechanikai rendszer $|\psi\rangle$ állapotban van, akkor annak a valószínűsége, hogy a mérésünk eredményeként $|\varphi\rangle$ állapotot kapunk $P(|\psi\rangle \longrightarrow |\varphi\rangle) = |\langle\psi|\varphi\rangle|^2$. Ezt átváltozási valószínűségnek nevezzük.

A legegyszerűbb módja egy kvantumszámítógép fizikai állapotának modellezésére a véges dimenziós Hilbert terek használata. A számítógép pontos, fizikai vizsgálata nem célja a dolgozatnak, egyedül az adatot tároló regiszterek lesznek fontosak a továbbiakban számunkra. A kvantumszámítógép memóriája lehetővé teszi a 0 és 1 állapotok szuperpozíciójában lévő kubit állapotának tárolását. Tehát egy egy kubitet tároló regiszter kettő dimenziós, és megfeleltethető a $\mathcal{H} = \mathbb{C} \oplus \mathbb{C}$ Hilbert térnek.

Általános esetben az n bites regiszterek 2^n dimenziósak, és $\mathcal{H}_n = \mathcal{H} \otimes \cdots \otimes \mathcal{H}$-val modellezhető. Az $i_1, i_2, \cdots, i_n$ biteknek megfeleltetjük a $|i_1, i_2, \cdots, i_n\rangle = |i_1\rangle \otimes \cdots \otimes |i_n\rangle$ vektort.

Miután a kvantumszámítógép elvégezte egy algoritmus számításait, szükséges számunkra, hogy visszaalakítsuk a gép állapotát bitek sorozatára, mely az algoritmus kimenetének felel meg. Ezt a folyamatot mérésnek nevezzük.

Egy klasszikus bit megvizsgálásával egyértelműen eldönthető, hogy az azon tárolt információ értéke 0 vagy 1. A számítógépek például ilyen módon tudnak tartalmat előhívni a memóriájukból. Ezzel szemben, egy kvantumbit állapotát nem tudjuk megmérni a hagyományos értelemben vett módon, vagyis α és β értéke nem határozható meg pontosan. A mérés eredménye $|\alpha|^2$ valószínűséggel $|0\rangle$, $|\beta|^2$ valószínűséggel $|1\rangle$ lesz. Természetesen $|\alpha|^2 + |\beta|^2 = 1$. Geometriailag ez a feltétel megfelel a kvantumállapotot reprezentáló vektor normalizáltságának. Eszerint, egy általános kvantumállapot tekinthető egy egységvektornak a kétdimenziós komplex vektortérben.

Vegyük $\mathcal{I}_n$-t egy n kubitból álló rendszer bázisának, és legyen adott egy kvantumszámítógép végső állapota:

$$v = \sum_{I \in \mathcal{I}_n} \alpha_I |I\rangle.$$

A mérés a v-től függő P_v valószínűségi eloszlás szerint eredményez bit sorozatot. Tetszőleges $I \in \mathcal{I}_n$ esetén annak a valószínűsége, hogy I-t kapunk vissza a mérés eredményeként

$$P_v(I) = |\alpha_I|^2 / \sum_{J \in \mathcal{I}_n} |\alpha_J|^2.$$

Eszerint a kvantumalgoritmusunknak olyan állapotokat érdemes eredményként visszaadnia, melyek α_I amplitúdója a kívánt I-ben nagy a többi bázisvektorhoz tartozó amplitúdóhoz képest. Hacsak nem tudjuk a többi bázisvektorokhoz tartozó amplitúdót 0-ra csökkenteni, akkor tudnunk kell valamilyen módon ellenőriznünk az algoritmus eredményét, különben valamennyi bizonytalanság lesz a helyességet illetően.

A kvantumrendszerek változása unitér transzformációk által megy végbe. Tehát azon kapukat, melyeken áthalad az információ a kvantumszámítógépben, unitér műveletekként szükséges megadnunk a $\mathcal{H}_n$ állapottérben.

A kvantumbiteken végzett műveletek esetében szükséges, hogy megőrizzék a vektorok hosszát, tehát unitér transzformációkat tudunk elvégezni a kubiteken.

2.4.8. Definíció (Unitér mátrix)**.** $U \in \mathbb{C}^{n\times n}$ komplex, négyzetes mátrix unitér, ha transzponált konjugáltja egyben az inverze is, vagyis $U^{-1} = U^*$, $U^*U = UU^* = I_n$.

2.4.9. Definíció (Pauli mátrix)**.** A Pauli mátrixok 2×2-es unitér mátrixok, melyeket X, Y, Z-vel jelölünk, és a következőképpen definiálunk:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad Y = iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

2.4.10. Állítás. A Pauli-X transzformáció felcseréli a standard bázis elemeit.

Bizonyítás. A $0 \longrightarrow |0\rangle$, $1 \longrightarrow |1\rangle$ hozzárendeléssel X transzformáció megfelel a klasszikus NOT műveletnek.

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

$$X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

□

A kvantumalgoritmusok alapköveit $\mathcal{H}_n$ feletti műveletek fogják adni, melyeket a $\mathcal{H}_1$ és $\mathcal{H}_2$ feletti műveletekből fogunk eredeztetni. Ezt a következőképpen tudjuk megtenni: Legyen adott H művelet a $\mathcal{H}_2$ felett, ekkor

$$\tilde{H} = \mathcal{H}_n \longrightarrow \mathcal{H}_n : v_1 \otimes v_2 \otimes \cdots \otimes v_n \longrightarrow H(v_1 \otimes v_2) \otimes v_3 \otimes \cdots \otimes v_n.$$

2.4.11. Megjegyzés. H művelet nincsen az első két pozícióhoz kötve, bármely két egymást követő kubiton hathat.

3. Kriptográfia és a kvantumszámítás rá gyakorolt hatása

Az alábbi fejezet Stephanie Wehner and Nelly Ng: edX Quantum Cryptography [3] kurzus 1. fejezete, és Vasileios Mavroeidis, Kamer Vishi, Mateusz D. Zych, Audun Jøsang: The Impact of Quantum Computing on Present Cryptography [22] cikk alapján íródott.

A kriptográfia szó eredete a görög elrejtés és írás szavakból ered, az információ továbbításának, és tárolásának folyamata úgy, hogy ne tudjon hozzáférni egy ellenséges harmadik fél. Kétféle kriptorendszert különböztetünk meg egymástól, a szimmetrikus, és az aszimmetrikus rendszereket.

3.1. A kvantumszámítás hatásai

A kvantumszámítás elméletét először Richard Feynman vezette be, azóta részletesen kutatott területté vált. Általánosságában, a kvantumszámítás bizonyos felhasználásával veszélybe kerülhet a napjainkban használt, modern aszimmetrikus kriptográfiai módszerek biztonsága. Néhány kvantumalgoritmus bizonyos szimmetrikus titkosításokra is hatással lehet, de ezek ellen védekezhetünk nagyobb kulcsválasztással. Ismeretesek algoritmusok, melyekkel törhetőek azon rendszerek, melyek biztonsága a nagy számok prímfaktorizációjának nehézségében rejlik, sőt, az elliptikus görbéken alapuló kriptográfia - ami manapság az egyik legbiztonságosabb módszerként használt - is gyengének tűnik a kvantumszámítógépek ellen. Ezek alapján megnőtt az igény kvantumtámadásoknak is ellenálló kriptográfiai algoritmusokra.

A kvantum mechanika a mikroszkópikus fizikai jelenségekkel, és azok működésével foglalkozik. Egy kvantumszámítógép ennek hatásait használja ki annak érdekében, hogy növelje a számítási kapacitását. A szuperpozícióban lévő kubiteken végzett műveletek a kubit amplitúdóin egyidőben módosít például.

Egy másik fontos jelenség a kvantumösszefonódás. Két összefonódott kvantumbit állapota már nem írható egymástól külön, csak egy külön egységként, melyet leírhatunk 4 állapot segítségével. Továbbá, ha a két összefonódott kubit közül az egyik állapota megváltozik, a vele összefonódott kubit állapota is megváltozik, függetlenül a közöttük lévő távolságtól. Ennek köszönhetően exponenciális módon megnő az egy művelettel kiszámolható értékek száma. Eszerint egy n-kubites kvantumszámítógép párhuzamosan 2^n műveletet tud végrehajtani.

3.2. Kriptográfiai fogalmak

Alice és Bob szeretnének egymással kommunikálni, anélkül, hogy az őket lehallgató Eve megtudná, hogy mit üzennek egymásnak. Tegyük fel, hogy Alice és Bob rendelkezik egy titkos k kulccsal, amiről Eve semmilyen információt nem ismer, tehát $p(k) = \frac{1}{|K|}$, ahol $|K|$ a lehetséges kulcsok száma. A titkosító és a dekódoló személyek előre egyeztetik egymással a k kulcsot. A titkosító $\mathrm{Enc}(k, m) = e$ függvény paraméterei a k kulcs és az m üzenet, melyekhez az e titkosított üzenetet rendeli hozzá. A dekódoló függvényt $\mathrm{Dec}(k, e) = m$-mel jelöljük.

3.2.1. Definíció (Biztonságos titkosítás)**.** Egy $(\mathrm{Enc}, \mathrm{Dec})$ titkosítást biztonságosnak nevezünk, akkor és csak akkor, ha tetszőleges $p(m)$ a priori eloszlás, és m üzenet esetén $p(m) = p(m|e)$, ahol $e = \mathrm{Enc}(k, m)$.

Más szavakkal, e ismeretében Eve nem tud meg többet m üzenetről, mint nélküle. Például triviális biztonságos titkosítás, ha Alice random e üzenetet küld, hiszen ekkor Eve nem tudja visszafejteni az üzenetet, azonban Bob sem, így kommunikációra ez nem megfelelő.

3.2.2. Definíció (Lehetséges üzenet)**.** Egy m üzenet lehetséges, ha $p(m) > 0$.

3.2.3. Definíció (Helyes titkosítás)**.** Egy $(\mathrm{Enc}, \mathrm{Dec})$ titkosítást helyesnek nevezünk pontosan akkor, ha tetszőleges lehetséges m üzenet és lehetséges k kulcs esetén $m = \mathrm{Dec}\,(k, \mathrm{Enc}(k, m))$.

Erre triviális példa, hogyha Alice $e = m$-et küldi el, ez viszont nem védi az üzenetet Eve elől. A kriptográfia célja, hogy egyszerre biztonságos és helyes titkosítást hozzunk létre.

3.2.4. Lemma. Egy $(\mathrm{Enc}, \mathrm{Dec})$ titkosítás akkor lehet biztonságos és helyes egyszerre, ha a lehetséges kulcsok $|K|$ száma legalább annyi, mint a lehetséges üzenetek $|M|$ száma, azaz $|K| \geq |M|$.

Bizonyítás. Indirekt tegyük fel, hogy létezik biztonságos és helyes titkosításunk, melyre $|K| < |M|$. Belátjuk, hogy a titkosításunk mégsem lesz biztonságos. Eve a megfigyelt e titkosított üzenet alapján kiszámolja $I = \{\hat{m} \,|\, \exists k, \hat{m} = \mathrm{Dec}(k, e)\}$ halmazt. Mivel minden kulcs esetén legfeljebb 1 darab $\hat{m}$-et kaphatunk, ezért $|I| \leq |K|$, és a feltétel szerint $|K| < |M|$, tehát $|I| < |M|$. Eszerint létezik legalább 1 darab m, melyre $m \notin I$, vagyis $p(m|e) = 0$. Ez alapján a lehallgató személy tudni fogja, hogy az üzenet nem lehetett m. Mivel egy üzenet csak akkor lehetséges, ha $p(m) > 0$, ezért kapjuk, hogy $0 = p(m|e) \neq p(m) > 0$, amivel ellentmondásba kerültünk a biztonsági feltételben. □

3.3. Szimmetrikus és aszimmetrikus kulcsú kriptográfia

A szimmetrikus kulcsú titkosító rendszerek esetében az üzenet küldője és fogadója is ugyanazt a kulcsot, és ugyanazt az algoritmust használja az üzenet titkosítására és dekódolására is. A kulcsnak muszáj titkosnak maradnia, csak a küldő és a fogadó fél ismerheti, mely megnöveli az igényt arra, hogy titokban tudjunk kulcsot cserélni nyilvános hálózatokon. Az aszimmetrikus kriptográfiát a szimmetrikus kriptográfiában szükségessé vált kulcscsere megoldására vezették be. A népszerűbb szimmetrikus kucsú algoritmusokban alkalmazzák az AES-t, és a 3DES-t.

Az aszimmetrikus kulcsú kriptográfiát más névvel nyilvános kulcsú kriptográfiának (Public Key Cryptography = "PKC") is nevezik. A kommunikációban van nyilvános és titkos kulcs is használva, viszont az üzenetet visszafejteni csak a titkos kulcs birtokában lehet.

A nyilvános kulcsú titkosítást szokás használni digitális aláírásokhoz is. A PKC eljárások olyan számítási problémákon nyugszanak, mint például a nagy számok faktorizációja, vagy a diszkrét logaritmus probléma. Az ilyen jellegű algoritmusokat egyirányú függvényeknek nevezzük, mivel az egyik irányba könnyen elvégezhetőek a számítások, viszon azok visszafejtése már nehéz. A prímfaktorizációra később kitérünk a 4. fejezetben.

Diszkrét logaritmus probléma (DLP)

A Diffie-Hellman kriptorendszer, és az elliptikus görbéken alapuló (Elliptic Curve Cryptography = "ECC") aszimmetrikus kulcsú kriptográfia alapját a DLP adja. Ennek nehézsége a következő problémán nyugszik: keressük azon r egész számot, melyre $g^r \equiv x \pmod{p}$. Nagy paraméterválasztás mellett a DLP számítása nehéz probléma.

A Diffie-Hellman aszimmetrikus eljárás ezt a számítási nehézséget használja ki a biztonságos kulcscsere érdekében. Jelenleg a 2048 bit méretű, vagy annál nagyobb kulcsok minősülnek biztonságosnak.

Az ECC-t is széles körben használják. Hasonló mértékben biztonságos, mint az RSA, csak kisebb kulcsok kellenek hozzá, így könnyebb ezt használni kis számítási kapacitás mellett.

4. Prímfaktorizáció kvantumszámítógépen

4.1. Az RSA biztonsága

Az RSA-ról szóló alfejezet Buttyán Levente, Vajda István: Kriptográfia és alkalmazásai [7] könyv, és Varga Zsolt: Modern Prímfaktorizáció [8] című szakdolgozat alapján készült.

4.1.1. Definíció (Prímfaktorizációs probléma)**.** Adott egy pozitív összetett egész N szám, ezt szeretnénk prímek szorzatára bontani.

A prímfaktorizációs probléma egyszerűsíthető a következő módon anélkül, hogy lényegében változtatnánk a problémán. Tekintsük egy lépésnek, ha adott N összetett számot felbontjuk $N = N_1 \cdot N_2$ szorzatra, ahol N_1, N_2 nemtriviális (nem egy és nem N) egészek. Ezen lépés lineárisan sokszoros elvégzése után megkapjuk a prímfelbontását N-nek. A legelemibb eset tehát, ha feltesszük, hogy $N = pq$, ahol p, q prímek.

4.1.2. Tétel (A „kis" Fermat-tétel)**.** Ha p prím, és $(a, p) = 1$, akkor $a^{(p-1)} \equiv 1 \pmod{p}$.

Klasszikus számítógépeken az RSA algoritmus az egyik leggyakrabban használt nyilvános kulcsú rejtjelező módszer. Biztonsága az egész számok faktorizációjának nehézségében rejlik. Hogyha létezne erre hatékony módszer, akkor az RSA törhető lenne, mivel az $N = pq$ felbontásból a dekódoló kulcs könnyen számolható.

4.1.3. Algoritmus (RSA)**.** Szükségünk van publikus, és privát kulcsok előállításához.

Kulcsgenerálás

1. Válasszunk két nagy prímet $\longrightarrow p, q$
2. Számoljuk ki $n = pq$-t
3. Számoljuk ki $\varphi(n) = (p-1)(q-1)$-et
4. Válasszunk d-t, melyre teljesül: $(d, \varphi(n)) = 1$
5. Válasszunk e-t, melyre teljesül, hogy $ed \equiv 1 \pmod{\varphi(n)}$
6. Az (e, n) párt publikus kulcsnak, a (d, n) párt privát kulcsnak nevezzük.

Titkosítás

1. Reprezentáljuk az M üzenetet k darab számmal $\longrightarrow M_1 M_2 \dots M_k$
2. $\forall M_i$, $i \in \{1, \dots, k\}$ esetén számoljuk ki $C_i = M_i^e \pmod{n}$-t
3. A titkosított üzenet $C = C_1 \dots C_k$

Dekódolás

1. $\forall C_i, i \in \{1, \dots, k\}$ esetén számoljuk ki $M_i = C_i^d \pmod{n}$-et
2. A dekódolt üzenet $M = M_1 \dots M_k$

4.1.4. Állítás (RSA helyessége)**.** Tegyük fel, hogy az algoritmus lépései során leírtaknak megfelelően választottuk ki a szükséges e és d számokat. Ekkor tetszőleges M üzenet esetén azt kódolva, majd dekódolva visszakapjuk M-et, vagyis $(M^e)^d = M \pmod{n}$.

Bizonyítás. e és d megválasztásakor a következő volt a feltétel:

$$ed \equiv 1 \pmod{\varphi(n)} \Longrightarrow ed = k \cdot \varphi(n) + 1,\ k \in \mathbb{N}.$$

Ezt behelyettesítve kapjuk, hogy $(M^e)^d = M^{ed} = M^{k\cdot\varphi(n)+1}$.

A „kis" Fermat-tételből következik, hogy minden p prímre és M-re, ahol $p \nmid M$ teljesül, hogy $M^{p-1} \equiv 1 \pmod{p}$. Ezt átalakítva érvényesek lesznek a következők:

$$M^{k\cdot(p-1)} \equiv 1 \pmod{p}, \text{ illetve}$$
$$M^{k\cdot(p-1)+1} \equiv M \pmod{p}.$$

Ezeket alkalmazva p, q prímekre megkapjuk, hogy

$$M^{k\cdot(p-1)+1} \equiv M \pmod{p}, \text{ és}$$
$$M^{k\cdot(q-1)+1} \equiv M \pmod{q}$$

Ezekből következik, hogy
$M^{k\cdot(p-1)(q-1)+1} \equiv M \pmod{pq} \Longrightarrow M^{k\cdot\varphi(n)+1} \equiv M \pmod{n} \Longrightarrow M^{ed} \equiv M \pmod{n}$. □

4.1.5. Példa (Példa az RSA működésére)**.** A számolás egyszerűsítése végett kis, egymástól különböző prímeket választunk. Legyenek a választott prímek $p = 13$ és $q = 17$ számok.

Kulcsgenerálás

Ekkor $n = pq = 221$, és $\varphi(n) = (p-1)(q-1) = 12 \cdot 16 = 192$. Mivel $192 = 2^6 \cdot 3$, válasszuk $d = 5$-öt, hiszen $(5, 192) = 1$ megfelel, és $e = 77$-et, mivel $77 \cdot 5 \equiv 1 \pmod{192}$. Ekkor a nyilvános kulcs $(77, 221)$, a privát kulcs $(5, 221)$.

Titkosítás

Válasszunk egy n-nél kisebb M üzenetet, legyen ez például $M = 4$. Megfigyelve a 4 hatványait modulo 221 a következőt látjuk:

$$4, 16, 64, 35, 140, 118, 30, 120, 38, 152, 166, 1, 4, \ldots$$

Tehát azt kaptuk, hogy a 4 modulo 221 multiplikatív rendje 11. Vagyis elég a 77-nek a 12-vel vett maradékára emelni az üzenetet. $77 \pmod{12} = 5$, tehát $M^e \pmod{221} \equiv 4^5 \pmod{221} \equiv 140 = C$. Ez a titkosított üzenet.

Dekódolás

Számoljuk ki $C^d \pmod{n}$-t. $C^d \pmod{n} \equiv 140^5 \pmod{221} = 4$, vagyis visszakaptuk az eredeti üzenetet.

4.1.6. Megjegyzés. Az RSA implementációja során számos szempotot figyelembe kell venni annak biztonságának érdekében. Erről bővebben olvashatunk Buttyán Levente, Vajda István: Kriptográfia és alkalmazásai [7] könyv 3.2 alfejezetében.

4.2. A prímfaktorizáció számelméleti háttere

A számelméleti háttérről szóló alfejezet Artur Ekert and Richard Jozsa: Quantum computation and Shor's factoring algorithm [6] mű alapján íródott.

4.2.1. Definíció (Periodicitási probléma)**.** Adott N pozitív összetett egész szám, random válasszunk egy N-hez relatív prím y-t. A feladat az $F_N(a) = y^a \pmod N$ függvény r periódusának megtalálása.

4.2.2. Állítás. A periodicitási probléma megoldásának felhasználásával a prímfaktorizáció elvégezhető.

Bizonyítás. Legyen adott $N \in \mathbb{N}$, és $y \in \mathbb{N}$ véletlenszerűen választott, melyre $(y, N) = 1$. Tekintsük az $x^2 \equiv 1 \pmod N$ alakú kongruenciát. Ennek $x \equiv \pm 1 \pmod N$ mindig megoldásai lesznek, ezeket hívjuk a triviális megoldásoknak. Ha N prím, akkor csak ezek a megoldások léteznek, mivel

$$x^2 \equiv 1 \pmod N \Longrightarrow (x+1)(x-1) = x^2 - 1 \equiv 0 \pmod N \Longrightarrow N|(x+1)(x-1),$$

és mivel N prím, ezért $N|(x+1)$ vagy $N|(x-1)$. Ha N összetett szám, vagyis $N = n_1 n_2$, ahol $n_1, n_2 \notin \{1, N\}$, akkor a kongruenciának léteznek $\pm a$, $a \neq 1$ alakú megoldásai is. Ennek igazolásához tekintsük a következő konguenciarendszereket.

- (a) $\begin{cases} x_1 \equiv 1 \pmod{n_1} \\ x_1 \equiv 1 \pmod{n_2} \end{cases}$
- (b) $\begin{cases} x_2 \equiv -1 \pmod{n_1} \\ x_2 \equiv -1 \pmod{n_2} \end{cases}$
- (c) $\begin{cases} x_3 \equiv 1 \pmod{n_1} \\ x_3 \equiv -1 \pmod{n_2} \end{cases}$
- (d) $\begin{cases} x_4 \equiv -1 \pmod{n_1} \\ x_4 \equiv 1 \pmod{n_2} \end{cases}$

A kínai maradéktétel szerint minden fenti kongruenciarendszernek van egyértelmű megoldása modulo N. Tetszőleges $i \in \{1, 2, 3, 4\}$ esetén $x_i^2 \equiv 1 \pmod{n_1}$ és $x_i^2 \equiv 1 \pmod{n_2}$, tehát $x_i^2 \equiv 1 \pmod N$ is teljesül. Az (a) és (b) rendszerek $x_1 \equiv 1 \pmod N$, $x_2 \equiv -1 \pmod N$ megoldásai adják $x^2 \equiv 1 \pmod N$ triviális megoldásait.
(c) és (d) rendszerek megoldásai nem lehetnek triviálisak, hiszen ha azok lennének, akkor például ha $x_i \equiv 1 \pmod N$, akkor $x_i \equiv 1 \pmod{n_1}$, és $x_i \equiv 1 \pmod{n_2}$, $i \in \{3, 4\}$, tehát $x_i, i \in \{3, 4\}$ a különböző modulusok szerint nem lehet eltérő értékekkel kongruens (-1-re ugyanez a gondolatmenet teljesül). Jelöljük tehát (c) és (d) rendszerek megoldásait a-val, és $-a$-val.

$$x^2 \equiv 1 \pmod N \Longrightarrow a^2 \equiv 1 \pmod N \Longrightarrow (a+1)(a-1) \equiv 0 \pmod N,$$

vagyis $N|(a+1)(a-1)$, és mivel $\pm a$ nemtriviális, ezért $a \pm 1 \not\equiv 0 \pmod N$, vagyis $N \nmid (a-1)$ és $N \nmid (a+1)$. Ezek alapján $\text{lnko}(N, a \pm 1) \neq N$, és $\text{lnko}(N, a \pm 1) \neq 1$, tehát $\text{lnko}(N, a \pm 1)$ nemtriviális osztója lesz N-nek.
Ismerjük a periodicitási probléma megoldását, azaz megkaptuk az $F_N(a) = y^a \pmod N$ függvény $r \in \mathbb{N}$ periódusát, melyre a definíció szerint érvényesül a következő:

$$\begin{aligned} F_N(a+r) \equiv y^{a+r} \equiv y^a \equiv F_N(a) \pmod N \qquad & / : y^a \ (N, y) = 1 \text{ miatt lehetséges} \\ y^r \equiv 1 \pmod N & \end{aligned}$$

Ha r páros, akkor válasszuk x-et $y^{r/2}$-nek. Ekkor $x^2 \equiv \left(y^{r/2}\right)^2 = y^r \equiv 1 \pmod{N}$. A kongruenciát átrendezve kapjuk, hogy

$$x^2 - 1 = (x+1)(x-1) \equiv 0 \pmod{N},$$

vagyis $N|(x+1)(x-1)$, és ha teljesül, hogy $x \pm 1 \not\equiv 0 \pmod{N}$, akkor $N \nmid x \pm 1$. Ekkor $\text{lnko}(N, x \pm 1)$ nemtriviális osztója N-nek. □

4.2.3. Megjegyzés. Az eljárás során akkor ütközhetünk problémába, ha r periódus páratlan, vagy ha $y^{r/2}$ triviális megoldása $x^2 \equiv 1 \pmod{N}$-nek. y véletlenszerű megválasztásával azonban ezek nem túl gyakran következnek be, erről szól a következő állítás.

4.2.4. Állítás. Legyen N páratlan szám, melynek prímfelbontása $N = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$. Tegyük fel, hogy y véletlenszerűen lett megválasztva, és teljesül rá, hogy $\text{lnko}(y, N) = 1$. Legyen továbbá y rendje r modulo N. Ekkor

$$\mathsf{P}(r \text{ páros és } y^{r/2} \not\equiv \pm 1 \pmod{N}) \geq 1 - \frac{1}{2^{k-1}}$$

Bizonyítás. A bizonyítás megtalálható a Artur Ekert and Richard Jozsa: Quantum computation and Shor's factoring algorithm [6] B függelékében. □

4.2.5. Példa (Bontsuk prímek szorzatára a 15-t)**.** $N = 15$, és random választanunk kell, egy ehhez relatív prím y-t. y értéke a $\{2, 4, 7, 8, 11, 13, 14\}$ számok közül kerülhet ki. Tegyük fel, hogy $y = 11$-et kaptuk, számoljuk ki $y = 11$ rendjét modulo 15. $a = 1, 2, 3, \dots$ hatványok esetén y^a értéke $11, 1, 11, \dots$, vagyis $y = 11$ rendje $r = 2$ modulo 15. Ezután x értéke legyen $y^{r/2} = 11^{2/2} = 11$, tehát lnko(11-1,15) = lnko(10,15) = 5 és lnko(11+1,15) = lnko(12,15) = 3 prímek, így az 5 és a 3 a 15 prímosztói.

4.3. Shor algoritmus

Shor kvantumalgoritmusa az r rendet határozza meg, amennyiben az létezik. Az algoritmus randomizált, mely $1 - \varepsilon$, $\varepsilon > 0$ valószínűséggel találja meg r-t, és futási ideje polinomiális. Ennek megfelelően a Shor algoritmussal tudunk prímfaktorizálni, tehát hatékony kvantumszámítógép használatával az RSA algoritmus törhető.

Az algoritmus lépéseit Koronka Gábor: Nagy számok faktorizációja Shor-algoritmussal [9] című szakdolgozat alapján foglaltuk össze, a QFT-ről szóló leírás Nicolas Sendrier, Raphael Overbeck: Code-Based Cryptography [18] alapján íródott.

4.3.1. Algoritmus (Shor prímfaktorizáció)**.** Az algoritmus inputja a faktorizálandó N szám.

1. Lefuttatunk egy prímtesztet N-re. Ha N prím, akkor készen vagyunk, és az algoritmus leáll.

2. Hogyha $2|N$, akkor ismerjük N egy nemtriviális felbontását. Ha páratlan, akkor folytatjuk a következő lépéssel.

3. Hogyha létezik olyan s, melyre $N = s^k$, $k \in \mathbb{N}$, akkor s^k megfelelő felbontása N-nek. Ha nem találtunk ilyen s-t, akkor nem prím, és nem is prímhatvány, tehát legalább két prím szorzata lehet, és ekkor folytatjuk a következő lépéssel.

4. Válasszunk egy random $m \in \mathbb{N}$ számot, és legyen $d := (N, m)$. Ha $d \neq 1$, akkor $d \cdot N/d$ nemtriviális felbontás. Ha $d = 1$, akkor folytatjuk az algoritmus következő lépésével.

5. Kiszámoljuk m szám rendjét, ezt jelöljük r-rel. Ehhez kvantumszámítógépet használunk, így kvantumpolinomiális időben találjuk meg a rendet.

6. Hogyha teljesül, hogy r páros, és $\left(N, m^{\frac{r}{2}} - 1\right) \neq 1$ és $\left(N, m^{\frac{r}{2}} + 1\right) \neq 1$, akkor $N = \left(N, m^{\frac{r}{2}} - 1\right) \cdot \left(N, m^{\frac{r}{2}} + 1\right)$ nemtriviális felbontás (4.2.2), és ekkor futassuk az algoritmust a 3. lépéstől.

4.3.2. Megjegyzés. A 3. lépés során ellenőrizzük, hogy N prímhatvány-e. Ezt a gyakorlatban úgy végezzük el, hogy $\forall\, 1 \leq k \leq \log N$-re megnézzük, hogy $N^{\frac{1}{k}}$ egész szám-e. Ehhez N k-adik gyökét olyan pontossággal közelítjük, hogy csak néhány egész szám fog a hibahatáron belül maradni. Ezeket a számokat k-adikra emelve ellenőrizzük, hogy valóban N-et kaptunk-e. Így megkapható polinomiális időben az összes olyan szám, melynek valamely hatványa N.

A rend kiszámításához kvantum Fourier transzformációt (Quantum Fourier Transformation = "QFT") szokás használni. A kvantum Fourier transzformáció a kvantum párhuzamosságot használja ki a gyors számolás érdekében. Ezt általában a rejtett részcsoport problémával (Hidden Subgroup Problem = "HSP") kapcsolatban szokták használni. Ennek legegyszerűbb formájában adott egy $\mathbb{Z}$ együtthatós periodikus függvény, és ennek keressük a periodusát. A kvantum Fourier transzformációt nem vizsgáljuk teljes részletességgel, bővebben olvashatunk róla a [18]-ben.

4.3.3. Definíció (QFT)**.** $N = 2^N$ hosszú intervallumon definiáljuk a kvantum Fourier transzformációt. Feleltessük meg i egésszel a $\mathcal{H}_k$-beli $|i\rangle$ vektort az i szám bináris felírása szerint. Ekkor QFT a következőképpen van definiálva:

$$QFT_k : \mathcal{H}_k \longrightarrow \mathcal{H}_k : |x\rangle \longrightarrow 2^{-\frac{N}{2}} \sum_{y=0}^{N-1} e^{2\pi i x y/N} |y\rangle.$$

5. McEliece kriptorendszer

2016-ban a NIST pályázatot hirdetett a legígéretesebb kvantum-biztos algoritmusok beküldésére. A cél a legjobb jelöltek további elemzése, majd szabványosítása. 2020. július 22-én a 3. fordulóba is bekerülő algoritmusok nyilvánosságra kerültek. A teljes lista megtalálható a NIST: Post-Quantum Cryptography, Round 3 Submissions [12] oldalon. Eszerint a McEliece kriptorendszer, és az NTRU az egyik legjobb jelöltek mint poszt-kvantum titkosító eljárások. Az NTRU-t bővebben elemezzük a 6. fejezetben.

Ezen fejezet Kiss Emil: Bevezetés az algebrába [23] könyv, Ashley Valentijn: Goppa Codes and Their Use in the McEliece Cryptosystems [19] cikk, és Michiel Marcus: White Paper on McEliece with Binary Goppa Codes [20] dolgozat alapján íródott.

5.1. Véges testek és polinomgyűrűk faktorizálása

5.1.1. Definíció (Test)**.** Az F halmazt testnek nevezzük, ha értelmezve vannak rajta a $(+, \cdot)$ műveletek úgy, hogy teljesülnek a következők:

1. $\exists a \in F$, melyre tetszőleges $x \in F$ esetén $x + a = x$ (nullelem),
2. $\exists b \in F$, melyre tetszőleges $x \in F$ esetén $x \cdot b = x$ (egységelem),
   továbbá tetszőleges $x, y, z \in F$ esetén teljesülnek a következők:
3. $x + y = y + x$,
4. $xy = yx$,
5. $(x + y) + z = x + (y + z)$,
6. $(xy)z = x(yz)$,
7. $x(y + z) = xy + xz$,
8. Tetszőleges $x \in F$ elemhez létezik $-x$, melyre $x + (-x) = a$,
9. Minden $x \neq a$ esetén létezik x^{-1}, melyre $xx^{-1} = b$.

5.1.2. Megjegyzés. Néha feltesszük, hogy $a \neq b$, különben egyelemű testről van szó. A továbbiakban eszerint számolunk. Mostantól az egységelem b-t 1-gyel, a nullelem a-t 0-val jelöljük.

5.1.3. Definíció (Véges testek)**.** Az olyan testeket, melyeknek elemszáma véges sok, véges testeknek, vagy Galois testeknek nevezzük, és $GF(q)$-val jelöljük, ahol q a test elemszáma.

5.1.4. Definíció (Irreducibilis polinom véges test fölött)**.** Egy polinom irreducibilis $GF(p^m)$ fölött, ha nem osztható semelyik másik kisebb fokú, $GF(p^m)$ feletti polinom által.

5.1.5. Példa. A $GF(2)$ véges test felett az (x^2+x^6) polinom nem irreducibilis, hiszen $x^2+x^6 = x^2(1 + x^4)$.

5.1.6. Definíció (Karakterisztika)**.** Legyen F egy test, ekkor, ha létezik egy olyan n, melyre $n := \underbrace{1+1+\cdots+1}_{n}$ 0-val egyenlő, akkor a legkisebb ilyen n-et F karakterisztikájának nevezzük, és $\mathrm{char}(F)$-fel jelöljük. Ha nem létezik ilyen $n \in \mathbb{Z}_0^+$, akkor azt mondjuk, hogy F karakterisztikája 0.

5.1.7. Tétel. Tetszőleges test karakterisztikája prím, vagy 0.

Bizonyítás. Jelölje n a karakterisztikáját F testnek. A karakterisztika nem lehet $n = 1$, hiszen $1 \neq 0$. Tegyük fel indirekt, hogy n összetett szám, vagyis $n = ab$, ahol $a, b \in F$, $1 < a, b < n$. Tudjuk, hogy minden test nullosztómentes, tehát ha

$$n = \underbrace{1+1+\cdots+1}_{n} = \underbrace{\underbrace{1+\cdots+1}_{a}+\cdots+\underbrace{1+\cdots+1}_{a}}_{b} = \underbrace{\underbrace{1+\cdots+1}_{b}+\cdots+\underbrace{1+\cdots+1}_{b}}_{a} = 0,$$

akkor vagy $a = 0$, vagy $b = 0$, tehát találtunk egy n-nél kisebb számot, ahányszor 1-et összeadva 0-t kapunk, tehát ellentmondásba kerültünk, vagyis egy test karakterisztikája vagy prím vagy 0. □

5.1.8. Tétel. Legyen p prím. Ekkor tetszőleges $p^m, m \in \mathbb{N}$ esetén egyértelműen létezik egy p^m elemszámú véges test. Minden véges test prímhatvány elemű.

Legyen F egy tetszőleges test, felette $F[x]$ az F-beli együtthatós polinomok gyűrűje. Legyen továbbá $f(x) \in F[x]$ egy polinom. Ekkor az $F[x]$ halmazon $\sim$ relációt értelmezni tudunk: $g_1(x) \sim g_2(x)$, ha $\exists h(x) \in F[x]$, hogy $g_1(x) - g_2(x) = f(x)h(x)$.

5.1.9. Állítás. A $\sim$ reláció ekvivalenciareláció.

Bizonyítás. A reflexivitás teljesül, hiszen $g(x) \sim g(x)$ a $h(x) \equiv 0$ mellett.

Szimmetriához adott legyen egy $h_1(x)$, mely esetén $g_1(x) - g_2(x) = f(x)h_1(x)$, és kell egy $h_2(x)$, melyre $g_2(x) - g_1(x) = f(x)h_2(x)$. Itt $h_2(x) = -h_1(x)$ jó, mert $g_1(x) - g_2(x) = f(x)h_1(x) \Longrightarrow g_2(x) - g_1(x) = f(x)(-h_1(x)) = f(x)h_2(x)$.

Szükséges még a tranzitivitás. Ehhez adottak:

$$\exists h_1(x) : g_1(x) - g_2(x) = f(x)h_1(x),$$

$$\exists h_2(x) : g_2(x) - g_3(x) = f(x)h_2(x).$$

Ekkor kell egy $h_3(x)$, melyre $g_1(x) - g_3(x) = f(x)h_3(x)$. Fejezzük ki a 2. sorból $g_2(x)$-et: $g_2(x) = f(x)h_2(x) + g_3(x)$, és helyettesítsük be az 1. sorba:

$$g_1(x) - g_2(x) = g_1(x) - f(x)h_2(x) - g_3(x) = f(x)h_1(x) \qquad / + f(x)h_2(x)$$

$$g_1(x) - g_3(x) = f(x)h_2(x) + f(x)h_1(x) = f(x)\left(h_1(x) + h_2(x)\right) = f(x)h_3(x)$$

Tehát $h_3(x) = h_1(x) + h_2(x)$ jó. □

5.1.10. Állítás. Ha $g_1(x) \sim g_2(x)$ és $g_3(x) \sim g_4(x)$, akkor a következők igazak:

1. $g_1(x) + g_3(x) \sim g_2(x) + g_4(x)$

2. $g_1(x) \cdot g_3(x) \sim g_2(x) \cdot g_4(x)$

Bizonyítás. Ismerjük a következőket:

$$g_1(x) \sim g_2(x) \Longrightarrow \exists h_1(x) : g_1(x) - g_2(x) = f(x)h_1(x)$$

$$g_3(x) \sim g_4(x) \Longrightarrow \exists h_2(x) : g_3(x) - g_4(x) = f(x)h_2(x)$$

Ezekből felírhatjuk az alábbiakat:

$$g_1(x) = g_2(x) + f(x)h_1(x)$$

$$g_3(x) = g_4(x) + f(x)h_2(x)$$

Melyeket felhasználva kapjuk:

$$g_1(x) + g_3(x) = g_2(x) + g_4(x) + f(x)h_1(x) + f(x)h_2(x) = g_2(x) + g_4(x) + f(x)(h_1(x) + h_2(x))$$

Tehát megkaptuk, hogy $g_1(x) + g_3(x) \sim g_2(x) + g_4(x)$.

Hasonlóan láthatjuk:

$$g_1(x) \cdot g_3(x) = (g_2(x) + f(x)h_1(x))\,(g_4(x) + f(x)h_2(x)) =$$

$$= g_2(x)g_4(x) + f(x)h_2(x)g_2(x) + f(x)h_1(x)g_4(x) + (f(x))^2 h_1(x)h_2(x) =$$

$$= g_2(x)g_4(x) + f(x)\,(h_2(x)g_2(x) + h_1(x)g_4(x) + f(x)h_1(x)h_2(x))\,,$$

vagyis $g_1(x) \cdot g_3(x) \sim g_2(x) \cdot g_4(x)$. □

Jelöljük $F[x]/f(x)$-szel a fenti ekvivalenciareláció által meghatározott ekvivalenciaosztályok halmazát, jelöljük $g(x)$ polinom ekvivalenciaosztályát $[g(x)]$-szel. A fenti halmazon definiálunk két műveletet:

1. $+ : [g_1(x)] + [g_2(x)] := [g_1(x) + g_2(x)]$
2. $\cdot : [g_1(x)] \cdot [g_2(x)] := [g_1(x) \cdot g_2(x)]$

A definiált műveletekkel a következő gond lehet: Ha veszünk két különböző ekvivalenciaosztályt, akkor azokból ha veszünk tetszőlegesen egy-egy elemet, azok összegéről/szorzatáról nem látszódik elsőre, hogy bármely választás esetén ugyanabban az ekvivalenciaosztályban van. Az 5.1.10 állítás azonban pont ezt biztosítja.

5.1.11. Tétel. Ha $f(x)$ polinom irreducibilis az F test fölött, akkor az $F[x]/f(x)$ halmaz a fenti két művelettel test lesz.

Bizonyítás. Kiss Emil: Bevezetés az algebrába [23] 5.2.9. állítás. □

5.1.12. Állítás. Legyen F test, $f(x) \in F[x]$ polinom, és tekintsük az $F[x]/f(x)$ ekvivalenciaosztályokat. Ekkor minden ekvivalenciaosztályban létezik pontosan egy $q(x)$ polinom, melyre $\deg(q(x)) < \deg(f(x))$, vagy $q = 0$.

Bizonyítás. Először belátjuk, hogy valóban létezik ilyen polinom. Vegyünk egy tetszőleges $p(x) \in F[x]$ polinomot, és tekintsük a $[p(x)]$ ekvivalenciaosztályt, majd válasszuk $g_1(x)$-et a $[p(x)]$-beli minimális fokú polinomok egyikének. Vezessük be a következő jelöléseket:

$$n := \deg(f), \qquad m := \deg(g_1).$$

A bizonyítást indirekt végezzük, vagyis feltesszük, hogy ebben az ekvivalenciaosztályban a legkisebb fokú $g_1(x)$ polinom foka is nagyobb, vagy egyenlő, mint $f(x)$ foka, tehát, hogy $m \geq n$. Abból szeretnénk ellentmondást kapni, hogy $g_1(x) \sim g_2(x)$, és $\deg(g_2) < \deg(g_1)$. A definíció szerint:

$$g_1(x) \sim g_2(x) : \exists h(x) \in F[x], \text{ amelyre } g_2(x) - g_1(x) = h(x)f(x)$$

Ezt átalakítva kapjuk: $g_2(x) = g_1(x) + h(x)f(x)$. Tehát keresünk egy olyan $h(x) \in F[x]$-et, melyre $\deg(g_1(x) + h(x)f(x)) < \deg(g_1)$. Legyenek $g_1(x)$ és $f(x)$ együtthatói az alábbiak:

$$g_1(x) = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0 \qquad \forall i : a_i \in F$$

$$f(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_0 \qquad \forall i : b_i \in F$$

$h(x) = -a_m b_n^{-1} x^{m-n}$ jó lesz, hiszen ilyen választás mellett:

$$h(x) \cdot f(x) = -a_m b_n^{-1} x^{m-n} b_n x^n + l(x) = -a_m x^m + l(x), \text{ ahol } \deg(l) < m.$$

Ekkor $g_1(x) + h(x)f(x)$ foka legfeljebb $m-1$ lehet, tehát találtunk egy $g_1(x)$-nél kisebb fokú polinomot, ami ellentmondás.

Szükséges még belátnunk, hogy pontosan egy ilyen polinom van. Indirekt feltesszük tehát, hogy létezik $[p(x)]$-ben két egymástól különböző polinom, melyek fokai kisebbek, mint $f(x)$-é. Jelöljük ezeket $g_3(x), g_4(x) \in [p(x)]$-szel. Mivel $g_3(x) \sim g_4(x)$, tehát egy ekvivalenciaosztályban vannak, a definíció szerint ez azt jelenti, hogy létezik egy $h(x) \in F[x]$, melyre $g_3(x) - g_4(x) = h(x)f(x)$. Ha $h(x) \neq 0$, akkor $h(x)f(x)$ foka legalább n lesz. $g_3(x) - g_4(x)$ foka viszont legfeljebb $\max\{\deg(g_3), \deg(g)\} < n$, hiszen mindkét polinom foka is kisebb, mint n, tehát azt kaptuk, hogy $h(x) = 0$, melyből következik, hogy $g_3(x) = g_4(x)$, ami ellentmondás. □

5.2. Kódelméleti fogalmak

A kódelméleti fogalmak Szőnyi Tamás Algebrai kódelmélet című kurzusának [25] anyaga alapján kerülnek bevezetésre.

Elképzelhető, hogy az üzenet küldése során hiba csúszik a folyamatba - például egy zajos csatornán kerül elküldésre az üzenet -, és így értelmetlen lesz a beérkező üzenet. A hibajavító kódok detektálják a hibát az elküldött üzenetben, és kijavítják azt. Ehhez az üzeneten felüli extra információ hozzáadása szükséges, mellyel könnyebben visszafejthető lesz a hibás üzenet. A következőkben összefoglalunk néhány algebrai kódelméleti fogalmat.

5.2.1. Definíció (Kód, kódszó)**.** Legyen Q egy q elemű halmaz, egy úgynevezett ábécé, és jelölje Q^n az összes olyan n hosszú betűsorozatból álló halmazt, melyek elemei Q-beliek. Ekkor C-t kódnak nevezzük, ha Q^n egy nemüres részhalmaza. A $c = (c_1, \cdots, c_n) \in C$ elemeket kódszavaknak nevezzük.

A továbbiakban csak azzal az esettel fogunk foglalkozni, mikor Q elemei a 0 és az 1, vagyis ahol az ábécé a $GF(2) = \mathbb{Z}_2 = \{0, 1\}$ véges test.

A kód rendelkezik egy generátormátrixszal, mely a k hosszú üzenetet n hosszúvá alakítja, és egy paritásellenőrző mátrixszal, mely képes n hosszú vektorok $n - k$ hosszú szindrómáivá alakítására.

5.2.2. Definíció. Azon kódokat, melyek a $GF(2)$ ábécét használják, bináris kódoknak nevezzük.

5.2.3. Definíció (Hamming-távolság)**.** Legyenek $c_i = (c_{i1}, \cdots, c_{in})$ és $c_j = (c_{j1}, \cdots, c_{jn})$ kódszavak, ekkor c_i és c_j Hamming-távolsága azon koordináták száma, melyekben eltérő értékük van.

5.2.4. Definíció (Minimális Hamming-távolság)**.** C kód minimális Hamming-távolságának nevezzük a C kódban előforduló két kódszó közötti legkisebb Hamming-távolság értéket, és $d(C)$-vel jelöljük.

5.2.5. Definíció (Hamming-súly)**.** Egy c kódszó Hamming-súlya azon koordináták száma, melyekben nemnulla érték van. Bináris kódszavak esetében ez megegyezik a kódszóban lévő 1-esek számával. $w(c)$-vel jelöljük.

5.2.6. Definíció (Hibajelző, hibajavító kódok)**.** Legyen $1 \leq t \leq n$ egy egész szám. Azt mondjuk, hogy egy $C \subseteq Q^n$ kód t-hibajavító, ha egy kódszó legfeljebb t koordinátájában az értéket megváltoztatva nem kaphatunk kódszót. Legyenek továbbá $u, v \in C \subseteq Q^n$ kódszavak úgy, hogy $u \neq v$. Azt mondjuk, hogy C t-hibajavító, ha u, illetve v legfeljebb t koordinátájában az értéket megváltoztatva nem kaphatjuk ugyanazt a C-beli kódszót.

5.2.7. Definíció (Lineáris kód)**.** Legyen $Q = GF(q)$ véges test, ekkor a $C \subseteq Q^n$ kódot lineárisnak nevezzük, ha C lineáris altere a $GF(q)$ feletti Q^n vektortérnek.

5.2.8. Definíció (Lineáris kód generátormátrixa)**.** Egy C lineáris kód generátormátrixa G, ha a G mátrix sorvektorai a C vektortér egy bázisát adja.

5.2.9. Definíció (Lineáris kód duálisa)**.** Egy $C \subseteq Q^n$ lineáris kód duálisa

$$C^{\perp} = \left\{ x \in Q^n \middle| \langle x|c \rangle = 0, \forall c \in C \right\}.$$

5.2.10. Definíció (Paritásellenőrző mátrix)**.** Egy C lineáris kód paritásellenőrző mátrixa H mátrix, ha H a $C^{\perp}$ duális kód generátormátrixa.

5.2.11. Állítás. Ha egy $[n, k]$-kód generátormátrixa $G = [I_k|P]$ alakú, akkor a $H = \left[-P^T|I_{n-k}\right]$ mátrix a kód egy paritásellenőrzőmátrixa.

5.2.12. Definíció (Szindróma)**.** Az x vektor szindrómája: $s(x) = x \cdot H^T$.

5.3. A McEliece kriptorendszer

Az alfejezet írása során felhasználtuk a D. Engelbert, R. Overbeck, and A. Schmidt: A Summary of McEliece-Type Cryptosystems and their Security [24] című cikket.

A McEliece kriptorendszer nyilvános-kulcsú titkosítás, mely lineáris, hibajavító kódokat használ fel. Általában bináris Goppa kódokkal szokták használni.

1978-ban Robert McEliece nyilvánosságra hozta a legelső kódeléleti alapokon nyugvó nyilvános kulcsú kriptorendszert. Ebben Goppa kódok használatát ajánlja a titkosításhoz, amely módszer megfelelő paramterválasztás mellett törhetetlen maradt. 1986-ban Harald Niederreiter egy eltérő eljárást javasolt, amely általános Reed-Solomon kódokat használt Goppa kódok helyett. Erről később Sidelnikov és Shestakov belátta 1992-ben, hogy nem biztonságos.

Később további változatok jöttek nyilvánosságra, melyek többsége a Goppa kódok helyett más kódokat használt, de a legtöbb vagy nem volt biztonságos, vagy nem volt annyira hatékony, mint az eredeti McEliece rendszer.

A McEliece eljárás legfontosabb változatai Kobara és Imai személyekhez köthetőek, melyeket 2001-ben publikáltak. Ezek CCA2-es biztonságúak, és bizonyíthatóan olyan biztonságosak, mint az eredeti McEliece kriptorendszer.

5.3.1. Definíció (Permutációmátrix)**.** Azt mondjuk, hogy P mátrix permutációmátrix, ha minden sorában és oszopában pontosan egy elem nemnulla, és ennek értéke 1.

5.3.2. Algoritmus (McEliece titkosítás)**.** Input: Egy (n, k) lineáris, $\leq t$ hibajavító kód.

Kulcscsere

1. Bob kiválasztja C egy G generátormátrixát.
2. Bob választ egy $k \times k$-as S invertálható mátrixot.
3. Bob választ egy $n \times n$-es P permutációmátrixot.
4. A privát kulcs (G, S, P).
5. A nyilvános kulcs $G' = SGP$.

Titkosítás

1. Alice véletlenszerűen választ egy k hosszú bináris e vektort, melyre $w(e) \leq t$.
2. $m = (m_1, \cdots, m_k)$ üzenet esetén a titkosított üzenet $y = mG' + e$.
3. Alice elküldi y-t.

Dekódolás

1. Bob kiszámolja $y' = yP^{-1}$-t.
2. $y' = yP^{-1} = (mG' + e)P^{-1} = mG'P^{-1} + eP^{-1} = mSGPP^{-1} + e' = mSG + e'$.
3. Bob kiszámolja a dekódoló algoritmussal mS-t.
4. Mivel S invertálható, ezért Bob ki tudja számolni S^{-1}-t.
5. Visszafejthető az eredeti üzenet: $m = (mS)S^{-1}$.

5.3.3. Példa (McEliece titkosítás Hamming kódolással)**.** A könnyebb számolás érdekében Goppa kód helyett a $(7, 4)$ Hamming kódot fogjuk használni.

Kulcsgenerálás
A $(7,4)$ Hamming kód egy generátormátrixa az alábbi:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Legyen a Bob által választott $k \times k$-as S mátrix, és a az $n \times n$-es P permutációmátrix:

$$S = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

majd Bob kiszámolja $G' = SGP$-t, és nyilvánossá teszi azt.

$$G' = SGP = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Titkosítás
Alice el szeretné küldeni az $x = \begin{pmatrix} 1, & 0, & 1, & 1 \end{pmatrix}$ üzenetet, ehhez választ egy 1-súlyú hibavektort, ez legyen:

$$e := \begin{pmatrix} 0, & 1, & 0, & 0, & 0, & 0, & 0 \end{pmatrix},$$

majd kiszámolja $y = xG' + e$-t:

$$y = xG' + e = \begin{pmatrix} 0, & 0, & 1, & 1, & 0, & 1, & 1 \end{pmatrix} + \begin{pmatrix} 0, & 1, & 0, & 0, & 0, & 0, & 0 \end{pmatrix} = \begin{pmatrix} 0, & 1, & 1, & 1, & 0, & 1, & 1 \end{pmatrix}$$

és y-t elküldi Bobnak.
Dekódolás

Bob kiszámítja P^{-1}-et. Ezt könnyű megtenni, mivel P permutációmátrix, így az inverze a transzponáltja lesz. Ennek segítségével kiszámolja $y' = yP^{-1}$-et:

$$y' = yP^{-1} = yP^T = \begin{pmatrix} 1, & 1, & 1, & 0, & 1, & 1, & 0 \end{pmatrix}.$$

A dekódolás során Bobnak ellenőriznie kell a hibákat. $G = \begin{pmatrix} I_4 | P \end{pmatrix}$ alak felhasználásával könnyen felírhatja az alábbi $H = \begin{pmatrix} P^T | I_3 \end{pmatrix}$ paritásellenőrző mátrixot:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Bob kiszámolja Hy'-t, majd megkeresi H azon oszlopát, melyben Hy' van. Ezen oszlop indexében van hiba.

$$Hy' = \begin{pmatrix} 1, & 1, & 0 \end{pmatrix}$$

Az 1. index értékét megváltoztatva y" $= \begin{pmatrix} 0, & 1, & 1, & 0, & 1, & 1, & 0 \end{pmatrix}$-et kapjuk. Mivel G-t $\begin{pmatrix} I_4 | P \end{pmatrix}$ alakban választottuk meg, így y" első 4 indexét véve megkapjuk xS-t. S mátrixot invertálhatónak választottuk, tehát jobbról S^{-1}-gyel szorozva megkapjuk az eredeti üzenetet:

$$\begin{pmatrix} 0, & 1, & 1, & 0 \end{pmatrix} S^{-1} = \begin{pmatrix} 0, & 1, & 1, & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1, & 0, & 1, & 1 \end{pmatrix}.$$

5.4. Goppa kódok

Ugyan tetszőleges lineáris kód, mely rendelkezik hatékony dekódoló algoritmussal használható a McEliece titkosításhoz, a Goppa kódok használata javasolt. A Goppa kódok Valery Goppa által lettek bevezetve 1970-ben. Ezek lineáris, hibajavító kódok, melyek használhatóak titkosításra és dekódolásra.

Az m paraméter meghatározza azt a $GF(2^m)$ véges testet, melyben a számításainkat fogjuk végezni, ez a véges test elemszámában a 2 kitevője.

5.4.1. Megjegyzés. Ha $f(z)$ irreducibilis polinom $GF(2)[z]$ felett, akkor $GF(2^m)$-nek megfelel $GF(2)[z]/f(z)$, ahol $deg(f) = m$. Ugyanis az 5.1.11 állítás alapján $GF(2)[z]/f(z)$ test, elemszáma pedig az 5.1.12 állítás alapján látható: m-nél kisebb fokú bináris polinomokból 2^m van, vagyis megkaptuk a 2^m elemszámú véges testet, a $GF(2^m)$-t.

Az 5.1.12 állítás alapján ennek elemeire gondolhatunk legfeljebb $m-1$ fokú, bináris együtthatós polinomokként is. Ezeket felírhatjuk az alábbi módon:

$$a(z) = \sum_{j=0}^{m-1} a_j z^j.$$

Legyen $n \leq 2^m$, ekkor $\sigma = \sigma_1, \sigma_2, \cdots, \sigma_n$, ahol minden $i = 1, \cdots, n$ esetén $\sigma_i \in GF(2^m)$, és ha $i \neq j$, akkor $\sigma_i \neq \sigma_j$. Ekkor a σ sorozatot tartónak nevezzük, ez a kód második paramétere. σ tetszőleges elemére gondolhatunk a következőképpen, polinom alakban:

$$\sigma_i(z) = \sum_{j=0}^{m-1} \sigma_{i,j} z^j,$$

ahol $\sigma_{i,j}$ a j. együtthatója a $\sigma_i(z)$ tartóelemnek.

A harmadik $t \in \mathbb{N}^+, t \leq n$ paraméter azon hibák száma, melyeket a kód javítani tud. Ehhez egy t-fokú $g(x) \in GF(2^m)[x]$ polinomot fogunk használni. Fontos, hogy a σ tartó egyik σ_i elemére se adjon $g(x)$ 0-t, vagyis

$$g(\sigma_i(z)) \neq 0 \qquad \forall \sigma_i \in \sigma.$$

5.4.2. Megjegyzés. A $g(\sigma_i(z)) \neq 0, \forall \sigma_i \in \sigma$ feltétel azt jelenti, hogy $\sigma_i(z)$ nem lehet gyöke a $g(x)$ polinomnak. Ha mégis gyöke lenne, akkor érvényes a következő szorzatra bontás: $g(x) = h(x)(x - \sigma_i(z))$, valamely $h(x) \in GF(2^m)[x]$ mellett. Ha $g(x)$ irreducibilis $G(2^m)$ felett, akkor nyilván a kiemelés nem tehető meg, tehát a továbbiakban $g(x)$-et irreducibilisnek választjuk $GF(2^m)$ felett.

5.4.3. Megjegyzés. Legyenek $k(x), l(x), g(x)$ ugyanazon test feletti polinomok. A

$$k(x) \equiv l(x) \pmod{g(x)}$$

jelölés a 5.1.9 állítás előtt definiált $g(x)$ szerinti relációval értendő.

5.4.4. Megjegyzés. Nézzük az $x - \sigma_i(z) \in GF(2^m)[x]$ polinomot. Mit jelent az

$$(x - \sigma_i(z))q(x) \equiv 1 \pmod{g(x)}$$

ekvivalencia? A definíció szerint valamely $s(x)$ polinom mellett teljesül:

$$1 = (x - \sigma_i(z))q(x) - s(x)g(x)$$

Amennyiben $g(x)$ és $x - \sigma_i(z)$ relatív prímek, akkor Kiss Emil: Bevezetés az algebrába [23] 3.2.7. tételének egy speciális eseteként látható, hogy mindig van ilyen $s(x)$ és $q(x)$ polinom. Mivel $g(x)$ polinomot irreducibilisnek választottuk, ezért csak akkor nem lehetnének relatív prímek $x - \sigma_i(z)$-vel, ha egyenlők volnának. Viszont akkor $g(x)$ $\sigma_i(z)$-ben 0-t venne fel, amit kikötöttünk, hogy nem lehet. Ennek megfelelően a továbbiakban a $\frac{1}{x-\sigma_i(z)} := q(x)$ jelölést használjuk.

5.4.5. Definíció (Bináris Goppa kód)**.** Legyen adott $GF(2^m)$ véges test, egy n hosszú σ tartó, és egy t-fokú $g(x)$ irreducibilis polinom $GF(2^m)$ felett. Ekkor a bináris Goppa kód definíciója:

$$C = \left\{ c \in \{0,1\}^n \middle| \sum_{i=0}^{n-1} \frac{c_i}{x - \sigma_i(z)} \equiv 0 \pmod{g(x)} \right\}$$

A Goppa kódok lineáris kódok, tehát használhatjuk a leírásukra az $[n, k, d]$ jelölést, ahol n a kód hossza, k a dimenziószám, és d a minimális Hamming-távolság.

5.4.6. Definíció (Goppa kód paritásellenőrző mátrixa)**.** Egy Goppa kód paritásellenőrző mátrixa egy olyan H mátrix, melyre $Hc^T = 0$ teljesül minden olyan $c \in GF(2^m)$ vektorra, mely eleget tesz a Goppa kódok feltételeinek.

5.4.7. Állítás. A kibővített Euklideszi algoritmus n-szeres elvégzésével megkapható a Goppa kód egy paritásellenőrző mátrixa.

Az euklideszi algoritmus n-szeres elvégzésénél van hatékonyabb módja is a paritásellenőrző mátrix számításának, erről szól a következő állítás.

5.4.8. Állítás. Tekintsük a következő mátrixokat.

$$X = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ g_{t-1} & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & g_3 & \dots & 1 \end{pmatrix}$$

$$Y = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \sigma_1(z) & \sigma_2(z) & \dots & \sigma_n(z) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1(z)^{t-1} & \sigma_2(z)^{t-1} & \dots & \sigma_n(z)^{t-1} \end{pmatrix}$$

$$Z = \begin{pmatrix} \frac{1}{g(\sigma_1(z))} & 0 & \dots & 0 \\ 0 & \frac{1}{g(\sigma_2(z))} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{g(\sigma_n(z))} \end{pmatrix}$$

$H = XYZ$ esetén H paritásellenőrző mátrixa a Goppa kódnak.

A bizonyítást itt nem írjuk le, azonban megtalálható a [20] cikkben.

A Goppa-kódokkal való titkosítás során szükségünk lesz az üzenet a Goppa-kód generátormátrixszával való szorzására.

5.4.9. Definíció (Bináris Goppa kód generátormátrixa)**.** Egy $C \subseteq GF(2^n)$ bináris Goppa kód generátormátrixa egy olyan $G \in GF(2^{k \times n})$-es mátrix, ha C azon Gu alakú vektorok halmaza, ahol u befutja $GF(2^n)$-t.

Az üzenet küldése előtt az m üzenetet k hosszú blokkokra kell bontani, majd utána minden blokkot külön kell megszorozni a generátormátrixszal. A kapott vektorok kódszavak egy halmaza lesz. Egy blokk titkosítása:

$$(m_1, \cdots, m_k) \cdot G = (c_1, \cdots, c_n)$$

5.4.10. Megjegyzés (Patterson algoritmus)**.** A Patterson algoritmus a beérkező kódszóban lévő hibákat javítja, vagyis meghatározza $y = c + e$-ben c, e-t. Az algoritmus Pattersontól származik 1975-ből. Ha a McEliece titkosítás során bináris Goppa kódokat használunk, a Patterson algoritmust használhatjuk a dekódoláshoz.

6. NTRU

Az 6. fejezet írása során Abderrahmane Nitaj: The Mathematics of the NTRU Public Key Cryptosystem [13] cikket, és Radnai András: Rácselmélet alkalmazása a számelméletben [14] szakdolgozatot használtam fel.

Az NTRU (Nth Degree Truncated Polynomial Ring Units) a legelső olyan nyilvános kulcsú titkosító eljárás, amely biztonsága nem a prímfaktorizációra vagy a diszkrét logaritmus problémára alapul, hanem rácselméleti háttere van. Hozzátartozik a titkosító eljárás (NTRUEncrypt) és a digitális aláíráshoz szükséges algoritmus (NTRUSign) is, és hatékonyabb, mint a jelenleg széles körben használt RSA és ECC eljárások. Tekintve, hogy megfelelően implementálva ellenáll a klasszikus-, és kvantumtámadásoknak, ígéretes jelölt mint nyilvános kulcsú titkosítás.

A következőkben megvizsgáljuk az algoritmus működését, a matematikai hátterét, és belátjuk, hogy nem megfelelő paraméterválasztás mellett a titkosítás törhető.

Az NTRU biztonsága a rácselméleti legrövidebb vektor (Shortest Vector Problem = "SVP") nehézségén alapul, melynek megoldására feltételezhetően nincsen polinomiális idejű algoritmus.

6.1. Jelölések

Legyen q egész szám, és jelöljük $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$-vel az egész számok gyűrűjét modulo q. Az NTRU során használt műveleteket a $\mathcal{P} = \mathbb{Z}_q[x]/(x^N - 1)$ faktorgyűrűben végezzük el. Ebben a gyűrűben egy f polinom együtthatóit felírhatjuk az $\{1, x, x^2, \ldots, x^{N-1}\}$ bázisban:

$$f = (f_0, f_1, \ldots, f_{N-1}) = f_0 + f_1 x + \cdots + f_{N-1} x^{N-1},$$

ahol $f_i \in \mathbb{Z}_q$, $i = 0, 1, \cdots, N-1$.

6.1.1. Definíció (Polinomok összege)**.** f és g polinomok összegét minden koordinátában a megfelelő fokhoz tartozó együtthatók összege adja modulo q, vagyis

$$f + g = (f_0 + g_0, f_1 + g_1, \ldots, f_{N-1} + g_{N-1})$$

6.1.2. Definíció (Polinomok szorzata)**.** f és g polinomok szorzatát a következő képlet adja:

$$f \cdot g = h = (h_0, h_1, \ldots, h_{N-1}), \text{ ahol } h_k = \sum_{i+j \equiv k \,(\mathrm{mod\ N})} f_i \cdot g_j, \qquad k = 1, \cdots, N-1$$

6.1.3. Definíció (Polinom inverze)**.** Azt mondjuk, hogy az $f \in \mathcal{P}$ polinom inverze $\mathcal{P}$-ben modulo q f_q, ha a szorzást modulo q tekintve $f \cdot f_q = 1$.

6.1.4. Definíció (Polinom hossza)**.** f polinom Euklideszi normája (hossza):

$$||f|| = \sqrt{\sum_{i=0}^{N-1} f_i^2}$$

6.1.5. Definíció (Bináris polinomok halmaza)**.** Legyen d pozitív egész szám, és tekintsük azon legfeljebb $N-1$-edfokú polinomok halmazát, melyeknek pontosan d darab együtthatója 1, és a többi 0. Ekkor $\mathcal{B}(d)$-vel jelöljük a bináris polinomok halmazát, amit így definiálunk:

$$\mathcal{B}(d) = \left\{ f(x) = \sum_{i=0}^{N-1} f_i x^i \in \mathcal{P} \,\middle|\, f_i \in \{0,1\}, \sum_{i=0}^{N-1} f_i = d \right\}$$

6.2. NTRUEncrypt

Az NTRUEncrypt-nek számos változata létezik, mi a 2005-ben megjelent verzióját fogjuk megvizsgálni.

6.2.1. Algoritmus (NTRUEncrypt)**.** Szükségesek N, p, q, d_f, d_g, d_r nyilvános egész számok, és $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_m, \mathcal{L}_r$ nyilvános halmazok, ahol:

- N prím, amely elég nagy a biztonságos működéshez
- p és q egymáshoz relatív prím egész számok, ahol q sokkal nagyobb, mint p
- $\mathcal{L}_f = \mathcal{B}(d_f)$ kis polinomok halmaza, ahonnan a privát kulcsokat fogjuk választani
- $\mathcal{L}_g = \mathcal{B}(d_g)$ az előzőhöz hasonló kis polinomok halmaza, ahonnan a többi privát kulcsokat fogjuk választani
- $\mathcal{L}_m = \mathbb{Z}[x]/(x^N - 1)$, ahol $\{m \mid m \in \mathcal{L}_m\}$ halmaz a titkosítható üzenetek halmaza
- $\mathcal{L}_r = \mathcal{B}(d_r)$ polinomhalmaz

Kulcsgenerálás

1. Véletlenszerűen kiválasztunk egy $f \in \mathcal{L}_f$ polinomot úgy, hogy f-nek létezik inverze $\mathcal{P}$-ben modulo p és modulo q.
2. Kiszámítjuk $f_p = f^{-1}$ modulo p-t, és $f_q = f^{-1}$ modulo q-t.
3. Véletlenszerűen kiválasztunk egy $g \in \mathcal{L}_g$-t.
4. Kiszámoljuk $h \equiv g \cdot f_q$ modulo q-t.
5. A nyilvános kulcs az $(N, h, p, q, \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_m, \mathcal{L}_r)$.
6. A privát kulcs (f, f_p).

Titkosítás

1. Reprezentáljuk az üzenetet $m \in \mathcal{L}_m$ polinomként.
2. Véletlenszerűen válasszunk egy $r \in \mathcal{L}_r$ polinomot.
3. Titkosítsuk m üzenetet a nyilvános kulcsot és a következő szabályt használva: $e \equiv p \cdot r \cdot h + m \pmod{q}$.

Dekódolás

1. A dekódoló kiszámolja a privát kulcs birtokában $a \equiv f \cdot e \pmod{q}$-t.
2. Transzformáljuk a-t egy olyan polinomba, melynek együtthatói a $\left[-\frac{q}{2}, \frac{q}{2}\right)$ intervallumba esnek.
3. Kiszámoljuk $m \equiv f_p \cdot a \pmod{p}$-t, ami az eredeti üzenet volt.

6.2.2. Állítás. Az NTRUEncrypt (6.2) titkosítás helyes.

Bizonyítás.

$$\begin{aligned}
a &= f \cdot e \pmod q = \\
&= f \cdot (p \cdot r \cdot h + m) \pmod q = && / \, h = g \cdot f_q \text{ -t behelyettesítve} \\
&= f \cdot p \cdot r \cdot g \cdot f_q + f \cdot m \pmod q = \\
&= p \cdot r \cdot g \cdot f \cdot f_q + f \cdot m \pmod q = && / \, f_q = f^{-1} \pmod q \\
&= p \cdot r \cdot g + f \cdot m \pmod q.
\end{aligned}$$

A titkosítás akkor helyes, hogyha teljesül az alábbi összefüggés (az egyenlőség jobb oldalán modulo q nélkül):

$$p \cdot r \cdot g + f \cdot m \,(\text{mod } q) = p \cdot r \cdot g + f \cdot m \in \mathbb{Z}[x]/(x^N - 1).$$

Tehát, ha $a = p \cdot r \cdot g + f \cdot m \in \mathbb{Z}[x]/(x^N - 1)$, akkor

$$a \cdot f_p \pmod p \equiv (p \cdot r \cdot g + f \cdot m) \cdot f_p \pmod p \equiv 0 + f \cdot f_p \cdot m \pmod p \equiv m \pmod p.$$

□

6.2.3. Megjegyzés. Ajánlott az NTRU használata során töltés (padding) használata, erre megfelelő a NAEP [11] padding.

6.3. Rácselméleti háttér

6.3.1. Definíció (Rács)**.** Legyen n és d pozitív egész szám, továbbá $b_1, \ldots, b_d \in \mathbb{R}^n$ d darab lineárisan független vektor. A $(b_1, \ldots, b_d)$ által generált $\mathcal{L}$ rács a következőképpen definiált vektorhalmaz:

$$\mathcal{L} = \sum_{i=1}^{d} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{d} x_i b_i \middle| x_i \in \mathbb{Z} \right\}$$

6.3.2. Definíció (Bázis)**.** Az előző definícióban szereplő $b_1, \ldots, b_d$-t az $\mathcal{L}$ rács bázisának nevezzük.

6.3.3. Definíció (Rács rangja és dimenziója)**.** A 6.3.1 definícióban szereplő n szám az $\mathcal{L}$ rács rangja, d az $\mathcal{L}$ rács dimenziója.

6.3.4. Megjegyzés. Egy $d \geq 2$ dimenziójú $\mathcal{L}$ rácsnak végtelen sok bázisa van. Tetszőleges két különböző bázisnak ugyanannyi eleme van, és a közöttük lévő kapcsolat leírható egy megfelelő unimoduláris mátrix segítségével.

6.3.5. Tétel. Legyen $\mathcal{L} \subset \mathbb{R}^n$ egy d-dimenziós rács, és $(b_1, \ldots, b_d)$, $(b'_1, \ldots, b'_d)$ az $\mathcal{L}$ két különböző bázisa. Ekkor létezik egy $d \times d$-s U mátrix, melynek elemei egész számok és amire $\det(U) = \pm 1$ úgy, hogy teljesíti $[b'_1, \ldots, b'_d]^T = U[b_1, \ldots, b_d]^T$ egyenlőséget.

6.3.6. Definíció (Rács determinánsa)**.** Legyen $\mathcal{L}$ a $(b_1, \ldots, b_d)$ bázisú rács. Ekkor $\mathcal{L}$ determinánsa $\det(\mathcal{L}) = \sqrt{\det(BB^T)}$, ahol B a bázisból álló sorvektorok alkotta $d \times n$-es mátrix. Erre a B mátrixra bázismátrix néven is fogunk hivatkozni.

6.3.7. Tétel. Legyen $\mathcal{L}$ egy d-dimenziós rács, ekkor $\mathcal{L}$ determinánsa független a bázis megválasztásától.

6.3.8. Definíció (Teljes rangú rács)**.** Ha egy rács dimenziója és rangja megegyezik, vagyis $n = d$, akkor teljes rangú rácsról beszélünk.

6.3.9. Tétel. Legyen $\mathcal{L}$ egy teljes rangú, n-dimenziós rács, a $(b_1, \ldots, b_n)$ egy bázisa $\mathcal{L}$-nek, és B a bázisból álló sorvektorok alkotta mátrix. Ekkor

$$\det(\mathcal{L}) = |\det(B)| .$$

6.3.10. Definíció (Rács minimális távolsága)**.** Legyen $\mathcal{L}$ egy rács. $\mathcal{L}$ rács minimális távolsága λ_1, mely a rácsban előforduló legrövidebb nemnulla vektor hossza, vagyis

$$\lambda_1 = \inf \left\{ ||v|| \,\middle|\, v \in \mathcal{L} \backslash 0 \right\} .$$

6.3.11. Definíció. Legyen $\mathcal{L}$ egy teljes rangú, n-dimenziós rács, ekkor $i = 1, \ldots, n$ esetén az i-edik szukcesszív minimuma $\mathcal{L}$-nek

$$\lambda_i = \min \left\{ \max \{||v_1||, \ldots, ||v_i||\} \,\middle|\, v_1, \ldots, v_i \in \mathcal{L} \text{ lineárisan független vektorok} \right\}$$

6.3.12. Definíció (Legrövidebb vektor problémája)**.** Legyen $\mathcal{L} \subset \mathbb{R}^n$ egy teljes rangú, n-dimenziós rács. A legrövidebb vektor problémája (Shortest Vector Problem = "SVP") a következő: $\mathcal{L}$ egy adott B bázismátrixa mellett keressük meg $v \in \mathcal{L}$ nemnulla vektort, melyre $v = \lambda_1(\mathcal{L})$.

6.3.13. Definíció (Legközelebbi vektor problémája)**.** Legyen $\mathcal{L} \subset \mathbb{Z}^n$ egy teljes rangú, n-dimenziós rács. A legközelebbi vektor problémája (Closest Vector Problem = "CVP") a következő: $\mathcal{L}$ egy adott B bázismátrixa és $v \in \mathcal{L}$ vektor mellett keressünk egy $u \in \mathcal{L}$ vektort, melyre $||v - u||$ minimális.

6.3.14. Megjegyzés. Az 6.3.12 és 6.3.13 pontokban leírt problémák NP-nehezek, és nem ismert rájuk hatékony algoritmus.

6.4. LLL algoritmus

Az LLL algoritmust eredetileg egész együtthatós polinomok faktorizációjára használták. Számos területen alkalmazzák, ilyenekre példák a diofantikus egyenletek megoldáskeresése, illetve a kriptoanalízis is. Leginkább $\mathbb{R}^n$-beli véges vektorhalmazok megfelelő bázisának megtalálására szokás használni.

Az LLL algoritmus az egyik legfontosabb eszköze a rácselméleti eljárásoknak. Az SVP problémára részben ad megoldást, hiszen polinomiális idejű algoritmusról van szó, ami megbecsüli egy n-dimenziós rács legrövidebb vektorát. Az LLL algoritmus jellemzően mind elméletben, mind gyakorlatban kiemelkedő eredményeket mutat, és hatékonyan használható bizonyos rácselméleti problémák megoldására. Az algoritmus felhasználja a Gram-Schmidt-féle ortogonalizációs eljárást. Röviden összefoglaljuk ennek lényegét.

6.4.1. Tétel (Gram-Schmidt)**.** Legyen V egy n-dimenziós vektortér, és $(b_1, \ldots, b_n)$ a V egy bázisa. Legyen továbbá $(b_1^*, \ldots, b_n^*)$ n darab vektor, melyekre teljesülnek:

1. $b_1^* = b_1$
2. $\mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$ minden $j < i$ esetén
3. $b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$

Ekkor $(b_1^*, \ldots, b_n^*)$ ortogonális bázisa V-nek.
Mátrixokkal felírva $(b_1, \ldots, b_n)$-re és $(b_1^*, \ldots, b_n^*)$-re teljesül:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ \mu_{2,1} & 1 & 0 & 0 & \cdots & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1,1} & \mu_{n-1,2} & \mu_{n-1,3} & \cdots & 1 & 0 \\ \mu_{n,1} & \mu_{n,2} & \mu_{n,3} & \cdots & \mu_{n,n-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} b_1^* \\ b_2^* \\ b_3^* \\ \vdots \\ b_{n-1}^* \\ b_n^* \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

6.4.2. Definíció (Gram-Schmidt együtthatók)**.** A tételben szereplő $\mu_{i,j}$ értékeket Gram-Schmidt együtthatóknak nevezzük.

6.4.3. Megjegyzés. Ha $(b_1, \ldots, b_n)$ bázisa egy $\mathcal{L}$ rácsnak, akkor általános esetben $(b_1^*, \ldots, b_n^*)$ nem bázisa $\mathcal{L}$-nek.

6.4.4. Algoritmus (Gram-Schmidt-féle ortogonalizációs eljárás)**.** Bemeneteként adott $V \subset \mathbb{R}^n$ vektortér, és annak $(b_1, \ldots, b_n)$ bázisa. A 6.4.1 tételben leírtak szerint számoljunk az alábbiakban:

1. $b_1^* := b_1$
2. $\forall i \in \{2, \ldots, n\}$ esetén:
   - $\forall j \in \{1, \ldots, i-1\}$ esetén kiszámoljuk $\mu_{i,j}$-t.
   - Kiszámoljuk b_i^*-t

A kimenet a $(b_1^*, \ldots, b_n^*)$ ortogonális bázisa V-nek.

6.4.5. Definíció (LLL-redukció, gyenge redukció)**.** Legyen $B = (b_1, \ldots, b_n)$ egy $\mathcal{L}$ rács bázisa. Azt mondjuk, hogy ez LLL-redukált, ha a Gram-Schmidt-féle ortogonalizációs eljárás segítségével kapott $\mu_{i,j}$-re és a $B^* = (b_1^*, \ldots, b_n^*)$-re teljesülnek:

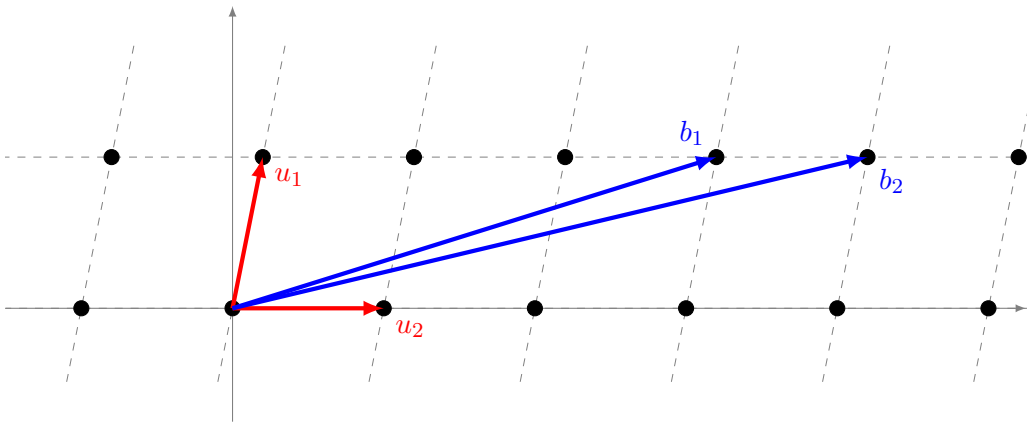
$$|\mu_{i,j}| \leq \frac{1}{2}, \qquad 1 \leq j < i \leq n \text{ esetén}$$

$$\frac{3}{4}||b_{i-1}^*||^2 \leq ||b_i^* + \mu_{i,i-1} b_{i-1}^*||^2, \qquad 1 < i \leq n \text{ esetén.}$$

Ha csak az első feltétel teljesül, akkor a bázist gyengén redukáltnak nevezzük. A második feltételt szokás Lovász-feltételnek is hívni.

Ha $\mu_{i,j} = 0$ minden i, j esetén, akkor a bázis ortogonális.

Tekintve, hogy végtelen sok bázis van, egyes bázisokat előnyösebbnek fogunk tartani a számításaink során. Intuíció alapján jó bázisoknak azoknak fogjuk tartani, melyek vektorai lehetőség szerint rövidek, és nagyjából merőlegesek egymásra. Következésképpen az LLL-redukált bázis ideális jelölt mint jó bázis.



1. ábra. Egy rács jó bázisa (u_1, u_2), és rossz bázisa (b_1, b_2).

6.4.6. Megjegyzés. Jelölje $\lfloor \mu_{i,l} \rceil$ a $\lfloor \mu_{i,l} \rfloor$ és $\lceil \mu_{i,l} \rceil$ közül a $\mu_{i,l}$-hez közelebbit.

6.4.7. Algoritmus (LLL algoritmus)**.** Az algoritmus bemenete az $\mathcal{L}$ rács egy $(b_1, \ldots, b_n)$ bázisa. Kimenete egy LLL-redukált bázisa $\mathcal{L}$-nek.

1. $i := 2$
2. Amíg $i \leq n$:
   (a) Kiszámoljuk $(b_1^*, \ldots, b_n^*)$-t a Gram-Schmidt-féle ortogonalizációs eljárással.
   (b) $b_i := b_i - \sum_{l=1}^{i-1} \lfloor \mu_{i,l} \rceil b_l$
   (c) Ha $||b_i^*||^2 > \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \cdot ||b_{i-1}^*||^2$, akkor $i := i + 1$,
   (d) különben cseréljük fel b_i-t, és b_{i-1}-et, és $i := \max\{2, i-1\}$.

6.4.8. Megjegyzés. Az LLL algoritmus során kapott bázis vektorai jelentősen rövidek lesznek, így az algoritmus használható a legrövidebb vektor becslésére.

6.4.9. Példa. Használjuk az LLL algoritmust a $b_1 = (48, 32), b_2 = (70, 33)$ bázisra. A számolás részletei megtalálhatóak az A függelékben. Az algoritmus eredménye:

$$v_1 = (22, 1) \qquad v_2 = (4, 30)$$

Ez valóban az eredeti rács egy bázisa, hiszen:

$$v_1 = b_2 - b_1, \quad \text{és} \quad v_2 = 3b_1 - 2b_2.$$

6.4.10. Tétel. Legyen $(b_1, \ldots, b_n)$ egy $\mathcal{L}$ rács bázisa, és használjuk a $B = \max_i ||b_i||$ jelölést. Ekkor az LLL algoritmus az LLL-redukált bázist $\mathcal{O}(n^4 \log^3 B)$ lépésben adja meg.

6.5. Az NTRU mint SVP

Ezen alfejezet a Joseph H. Silverman: An Introduction to the Theory of Lattices and Applications to Cryptography [17] jegyzet alapján íródott. A következőkben röviden áttekintjük,

hogy miképpen lehet az NTRU törésére egy legközelebbi vektor problémaként tekinteni. Az NTRU titkosítás nyilvános és privát kulcsa közötti kapcsolat felhasználásával meghatározhatunk egy rácsot, melyet NTRU rácsnak is hívunk. A rács bázisa a nyilvános kulcsból felírható, vagyis publikusan elérhető. A titkos kulcsra tekinthetünk ezen rács legrövidebb vektoraként. Következésképpen, az NTRU feltörésének megfeleltethető a rácsban előforduló legrövidebb vektor keresésének.

6.5.1. Definíció (Standard NTRU rács)**.** Standard NTRU rácsnak nevezzük, és $\mathcal{L}_h$-val jelöljük azt a $2N$-dimenziós rácsot, melyet a következő mátrix sorvektorai generálnak:

$$\mathcal{L}_h = \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right)$$

Belátjuk, hogy (f, g) benne van a rácsban. Ehhez bevezetünk egy új jelölést:

$$u = (u_0, u_1, \cdots, u_{N-1}) := \frac{-f \cdot h + g}{q} \in \mathcal{R}$$

Ekkor teljesül:

$$\left(\begin{array}{c} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \\ \hline u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{array}\right)^T \cdot \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right) = \left(\begin{array}{c} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \\ \hline g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{array}\right)^T$$

Valóban, a szorzat első N koordinátájában az f együtthatói fognak szerepelni. Nézzük meg, hogy mi lesz a szorzat egy tetszőleges $N+j+1$-edik ($j \in \{0, \cdots, N-1\}$) koordinátájában:

$$f_0 \cdot h_j + f_1 \cdot h_{j-1} + \cdots + f_{N-1} \cdot h_{j+1-N} + u_j \cdot q =$$

$$= (f \cdot h)_j - (f \cdot h)_j + g_j = g_j$$

Egy másik módja, hogy felírjuk ugyanezt a rácsot:

$$\mathcal{L}_h = \left\{(a, b) \in \mathcal{R} \times \mathcal{R} \middle| a \cdot h \equiv b \,(\mathrm{mod}\ q), \text{ ahol } \mathcal{R} = \mathbb{Z}[x]/(x^N - 1)\right\}.$$

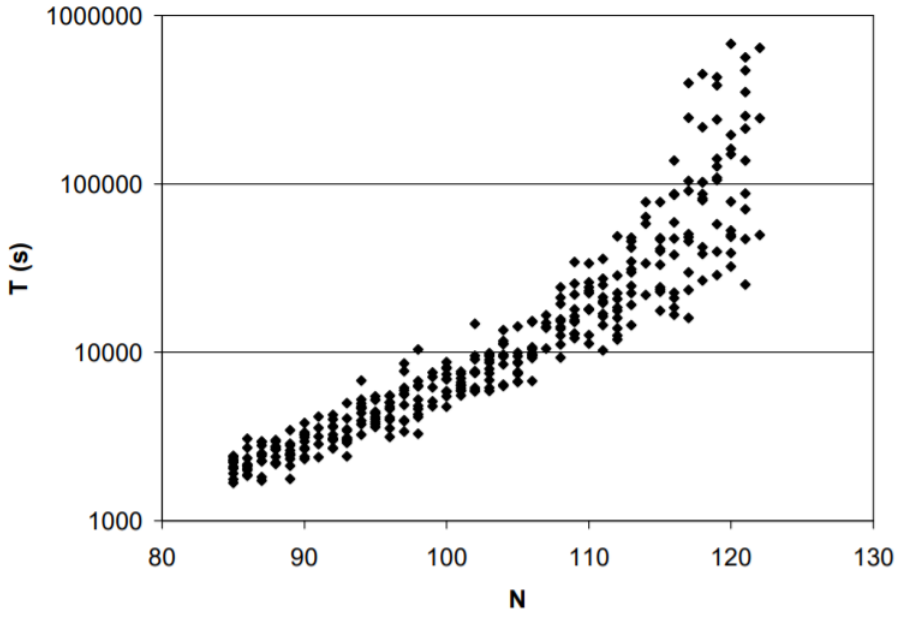
Tekintsük N, q, p számokat. A feltételek szerint az (N, h) nyilvános kulcsra teljesül, hogy $h \equiv g \cdot f_q \pmod q$. A két oldalt beszorozva f-fel is látható, hogy $f \cdot h \equiv g \pmod q$ teljesül. f-et és g-t kicsinek választottuk. Vagyis $\mathcal{L}_h$ rács tartalmazza a (f, g) rövid vektort. Tartalmazza

továbbá a hosszabb $(1, h)$, és $(0, q)$ vektorokat. Az (f, g) valószínűleg a legrövidebb vektor $\mathcal{L}_h$-ban, tehát majdnem biztosan megtalálható az $\mathcal{L}_h$-beli SVP megoldásával. Következésképpen, minden olyan algoritmus, amely megoldja a legrövidebb vektor problémáját, használható az NTRU feltörésére.

6.5.2. Megjegyzés. Az alábbi táblázat a paraméterek szerinti biztonság mértékét, és a kulcsméreteket tartalmazza. A táblázat forrása: NTRUEncrypt and Lattice Attacks: Security estimation [16]

Security level	N	p	q	d_f	d_g	d_r	$\lvert\boldsymbol{f}\rvert + \lvert\boldsymbol{f}_p^{-1}\rvert$	$\lvert\boldsymbol{h}\rvert$
Moderate	167	3	128	61	20	18	530	1169
Standard	263	3	128	50	24	16	834	1841
Highest	503	3	256	216	72	55	1595	3521

Az NTRU törésére az LLL algoritmus a gyakorlatban is használhatónak mutatkozik. Erről készítettek kísérletet, melynek eredményeit Jeffrey Hoffstein, Joseph H. Silverman és William Whyte publikálta NTRU Cryptosystems Technical Report, Estimated Breaking Times for NTRU Lattices [15] című cikkükben. Ebből származik az alábbi grafikon.



A kísérlet során 85 és 122 közötti N értékek esetén tudták sikeresen alkalmazni az LLL algoritmust. Az ábra jól mutatja, hogy N paraméter növelésével növekszik az NTRU töréséhez szükséges idő. A kísérlet pontos részletei megtalálhatóak az eredeti cikkben: [15].

6.6. Példa az NTRU algoritmus számolására

Ebben a példában kiszámoljuk az NTRU lépéseit adott paraméterek alapján. Paraméterek:

$$\begin{aligned} N &= 11 \\ p &= 3 \\ q &= 61 \\ f &= -x^{10} - x^8 - x^6 + x^4 + x^2 + x + 1 \\ g &= -x^9 - x^7 - x^5 + x^4 + x^2 + 1 \\ m &= x^8 - x^4 + x^2 + x + 1 \\ r &= -x^7 + x^5 + x^2 + 1 \end{aligned}$$

Kulcsgenerálás

Szükségünk van f_p-re és f_q-ra. Euklideszi algoritmust használunk az inverz polinom kiszámítására. f_p esetén részletesen leírjuk a számolást, f_q esetén hasonlóképpen számolhatunk. f_p kiszámolása:

$$\begin{aligned} x^{11} - 1 &= (-x)(-x^{10} - x^8 - x^6 + x^4 + x^2 + x + 1) \\ &\quad + (-x^9 - x^7 + x^5 + x^3 + x^2 + x - 1) \\ -x^{10} - x^8 - x^6 + x^4 + x^2 + x + 1 &= x(-x^9 - x^7 + x^5 + x^3 + x^2 + x - 1) \\ &\quad + (x^6 - x^3 + 2x + 1) \\ -x^9 - x^7 + x^5 + x^3 + x^2 + x - 1 &= (2x^3 + 2x + 2)(x^6 + 2x^3 + 2x + 1) \\ &\quad + (x^5 + x^4 + x^3 + x) \\ x^6 + 2x^3 + 2x + 1 &= (x + 2)(x^5 + x^4 + x^3 + x) \\ &\quad + (2x^2 + 1) \\ x^5 + x^4 + x^3 + x &= (2x^3 + 2x^2 + x + 2)(2x^2 + 1) + 1 \end{aligned}$$

Ebből visszafejtve megkapjuk f_p-t. A számolás részleteit kifejtjük az B függelékben. Így kapjuk a következőket:

$$\begin{aligned} f_p &= x^9 + x^7 + x^5 + 2x^4 + 2x^3 + 2x^2 + x \\ f_q &= 45x^{10} + 49x^9 + 26x^8 + 40x^7 + 53x^6 + 47x^5 + 21x^4 + 24x^3 + 60x^2 + 32x + 31 \end{aligned}$$

A privát kulcs: $(f = -x^{10} - x^8 - x^6 + x^4 + x^2 + x + 1, f_p = x^9 + x^7 + x^5 + 2x^4 + 2x^3 + 2x^2 + x)$
A h polinom kiszámolása során szükséges polinomszorzások részeredményeit levezetjük, azonban a későbbi szorzásoknak csak a végeredményeit fogjuk leírni.

$$
\begin{aligned}
h &= (h_0, h_1, \ldots, h_{10}) = g \cdot f_q \\
h_0 &= g_0 \cdot f_{q0} + g_2 \cdot f_{q9} + g_4 \cdot f_{q7} + g_5 \cdot f_{q6} + g_7 \cdot f_{q4} + g_9 \cdot f_{q2} \\
&= 31 + 49 + 40 - 53 - 21 - 60 \pmod{61} = 47 \\
h_1 &= g_0 \cdot f_{q1} + g_2 \cdot f_{q10} + g_4 \cdot f_{q8} + g_5 \cdot f_{q7} + g_7 \cdot f_{q5} + g_9 \cdot f_{q3} \\
&= 32 + 45 + 26 - 40 - 47 - 24 \pmod{61} = 53 \\
h_2 &= g_0 \cdot f_{q2} + g_2 \cdot f_{q0} + g_4 \cdot f_{q9} + g_5 \cdot f_{q8} + g_7 \cdot f_{q6} + g_9 \cdot f_{q4} \\
&= 60 + 31 + 49 - 26 - 53 - 21 \pmod{61} = 40 \\
h_3 &= g_0 \cdot f_{q3} + +g_2 \cdot f_{q1} + g_4 \cdot f_{q10} + g_5 \cdot f_{q9} + g_7 \cdot f_{q7} + g_9 \cdot f_{q5} \\
&= 24 + 32 + 45 - 49 - 40 - 47 \pmod{61} = 26 \\
h_4 &= g_0 \cdot f_{q4} + g_2 \cdot f_{q2} + g_4 \cdot f_{q0} + g_5 \cdot f_{q10} + g_7 \cdot f_{q8} + g_9 \cdot f_{q6} \\
&= 20 + 60 + 31 - 45 - 26 - 53 \pmod{61} = 49 \\
h_5 &= g_0 \cdot f_{q5} + g_2 \cdot f_{q3} + g4 \cdot f_{q1} + g_5 \cdot f_{q0} + g_7 \cdot f_{q9} + g_9 \cdot f_{q7} \\
&= 47 + 24 + 32 - 31 - 49 - 40 \pmod{61} = 44
\end{aligned}
$$

$$
\begin{aligned}
h_6 &= g_0 \cdot f_{q6} + g_2 \cdot f_{q4} + g_4 \cdot f_{q2} + g_5 \cdot f_{q1} + g_7 \cdot f_{q10} + g_9 \cdot f_{q8} \\
&= 53 + 21 + 60 - 32 - 45 - 26 \pmod{61} = 31 \\
h_7 &= g_0 \cdot f_{q7} + g_2 \cdot f_{q5} + g_4 \cdot f_{q3} + g_5 \cdot f_{q2} + g_7 \cdot f_{q0} + g_9 \cdot f_{q9} \\
&= 40 + 47 + 24 - 60 - 31 - 49 \pmod{61} = 32 \\
h_8 &= g_0 \cdot f_{q8} + g_2 \cdot f_{q6} + g_4 \cdot f_{q4} + g_5 \cdot f_{q3} + g_7 \cdot f_{q1} + g_9 \cdot f_{q10} \\
&= 26 + 53 + 21 - 24 - 32 - 45 \pmod{61} = 60 \\
h_9 &= g_0 \cdot f_{q9} + g_2 \cdot f_{q7} + g_4 \cdot f_{q5} + g_5 \cdot f_{q4} + g_7 \cdot f_{q2} + g_9 \cdot f_{q0} \\
&= 49 + 40 + 47 - 21 - 60 - 31 \pmod{61} = 24 \\
h_{10} &= g_0 \cdot f_{q10} + g_2 \cdot f_{q8} + g_4 \cdot f_{q6} + g_5 \cdot f_{q5} + g_7 \cdot f_{q3} + g_9 \cdot f_{q1} \\
&= 45 + 26 + 53 - 47 - 24 - 32 \pmod{61} = 21
\end{aligned}
$$

Tehát a nyilvános kulcs:
$(N = 11, h = 21x^{10} + 24x^9 + 60x^8 + 32x^7 + 31x^6 + 44x^5 + 49x^4 + 26x^3 + 40x^2 + 53x + 47)$

Titkosítás

$$
\begin{aligned}
e &= p \cdot r \cdot h + m \pmod{61} \\
&= 3 \cdot \left(38x^{10} + 4x^9 + 3x^8 + 8x^7 + 51x^6 + 32x^5 + 50x^4 + 10x^3 + 55x^2 + x + 53\right) \\
&\quad + x^8 - x^4 + x^2 + x + 1 \pmod{61} \\
&= 53x^{10} + 12x^9 + 9x^8 + 24x^7 + 31x^6 + 35x^5 + 28x^4 + 30x^3 + 43x^2 + 3x + 37 \\
&\quad + x^8 - x^4 + x^2 + x + 1 \pmod{61} \\
&= 53x^{10} + 12x^9 + 10x^8 + 24x^7 + 31x^6 + 35x^5 + 27x^4 + 30x^3 + 44x^2 + 4x + 38
\end{aligned}
$$

Dekódolás

Az e titkosított üzenetet az f kulcs segítségével fejtjük vissza.

$$a = f \cdot e \pmod{61} = 58x^{10} + 55x^9 + 59x^8 + 53x^7 + 2x^6 + 2x^5 + 7x^4 + 2x^3 + 9x^2 + 3x + 58$$

Áttranszformáljuk a-t úgy, hogy együtthatói a $\left[-\frac{q}{2}, \frac{q}{2}\right) = [-30, 31)$ intervallumba essenek.

$$a = -3x^{10} - 6x^9 - 2x^8 - 8x^7 + 2x^6 + 2x^5 + 7x^4 + 2x^3 + 9x^2 + 3x - 3$$

Végül kiszámoljuk $f_p \cdot a \pmod 3$-t:

$$f_p \cdot a \pmod 3 = 0x^{10}+0x^9+1x^8+0x^7+0x^6+0x^5-x^4+0x^3+1x^2+1x+1 = x^8-x^4+x^2+x+1 = m$$

Ez valóban megegyezik az eredeti m üzenettel.

Függelékek

A. a 6.4.9 példabeli számolás részletei

$$b_1 = (48, 32) \qquad b_2 = (70, 33)$$

Legyen $i = 2$. Kiszámoljuk a Gram-Schmidt-féle ortogonalizációs eljárással b_1^*, b_2^*-t:

$$b_1^* = b_1 = (48, 32)$$

$$b_2^* = (70, 33) - \frac{(70, 33) \cdot (48, 32)}{(48, 32) \cdot (48, 32)} \cdot (48, 32) = \left(\frac{32}{13}, -\frac{123}{13}\right)$$

$$b_2 := (70, 33) - \left\lfloor \frac{(70, 33) \cdot (48, 32)}{(48, 32) \cdot (48, 32)} \right\rceil \cdot (48, 32) = (70, 33) - 1 \cdot (48, 32) = (22, 1)$$

$$||b_1^*||^2 = 48^2 + 32^2 = 3328$$

$$||b_2^*||^2 = \frac{32^2 + 123^2}{13^2} \approx 95,58$$

$$\mu_{2,1} = \frac{(22, 1)(48, 32)}{(48, 32)(48, 32)} = \frac{1088}{3328}, \qquad \frac{3}{4} - \mu_{2,1}^2 \approx 0,64$$

Ezeket felhasználva kapjuk:

$$95,58 \approx ||b_2^*||^2 \not\geq \left(\frac{3}{4} - \mu_{2,1}^2\right) ||b_1^*||^2 \approx 2140,31$$

Tehát felcseréljük a vektorainkat, majd elvégezzük a számolást az új vektorokkal. Eszerint az aktuális vektorok:

$$b_1 = (22, 1) \qquad b_2 = (48, 32)$$

$$b_1^* = (22, 1)$$

$$b_2^* = (48, 32) - \frac{(48, 32)(22, 1)}{(22, 1)(22, 1)}(22, 1) = \left(-\frac{656}{485}, \frac{14432}{485}\right)$$

$$b_2 := (48, 32) - \left\lfloor \frac{(48, 32)(22, 1)}{(22, 1)(22, 1)} \right\rceil (22, 1) = (48, 32) - 2(22, 1) = (4, 30)$$

$$||b_1^*||^2 = 22^2 + 1^2 = 485$$

$$||b_2^*||^2 = \frac{656^2 + 14432^2}{485^2} \approx 887,29$$

$$\mu_{2,1} = \frac{(4, 30)(22, 1)}{(22, 1)(22, 1)} = \frac{118}{485}, \qquad \frac{3}{4} - \mu_{2,1}^2 \approx 0,69$$

Ezeket felhasználva kapjuk:

$$887,29 \approx ||b_2^*||^2 \geq \left(\frac{3}{4} - \mu_{2,1}^2\right) ||b_1^*||^2 \approx 335,04$$

Tehát a a kapott LLL redukált bázis:

$$b_1 = (22, 1) \qquad b_2 = (4, 30).$$

B. 6.6 példa: f_p visszaszámolása

$$
\begin{aligned}
&f := -x^{10} - x^8 - x^6 + x^4 + x^2 + x + 1, \quad && h := x^{11} - 1,\\
&a := 2x^3 + 2x^2 + x + 2, && b := 2x^2 + 1,\\
&c := x^5 + x^4 + x^3 + x, && d := x^6 - x^3 + 2x + 1,\\
&e := -x^9 - x^7 + x^5 + x^3 + x^2 + x - 1, && g := 2x^3 + 2x + 2,\\
&i := x + 2
\end{aligned}
$$

$$
\begin{aligned}
1 &= c - ab = e - gd - ab = h + xf - gd - ab = h + xf - g(f - xe) - ab =\\
&= h + xf - gf + gxe - ab = h + xf - gf + gx(h + xf) - ab =\\
&= h + xf - gf + gxh + gx^2 f - ab =\\
&= h(1 + gx) + f(x - g + gx^2) - ab =\\
&= h(1 + gx) + f(x - g + gx^2) - a(d - ic) =\\
&= h(1 + gx) + f(x - g + gx^2) - ad + aic =\\
&= h(1 + gx) + f(x - g + gx^2) - a(f - xe) + aic =\\
&= h(1 + gx) + f(x - g + gx^2) - af + axe + aic =\\
&= h(1 + gx) + f(x - g + gx^2 - a) + axe + aic =\\
&= h(1 + gx) + f(x - g + gx^2 - a) + ax(h + xf) + aic =\\
&= h(1 + gx) + f(x - g + gx^2 - a) + axh + ax^2 f + aic =\\
&= h(1 + gx + ax) + f(x - g + gx^2 - a + ax^2) + aic =\\
&= h(1 + gx + ax) + f(x - g + gx^2 - a + ax^2) + ai(e - gd) =\\
&= h(1 + gx + ax) + f(x - g + gx^2 - a + ax^2) + aie - aigd =\\
&= h(1 + gx + ax) + f(x - g + gx^2 - a + ax^2) + ai(h + xf) - aigd =\\
&= h(1 + gx + ax) + f(x - g + gx^2 - a + ax^2) + aih + aixf - aigd =\\
&= h(1 + gx + ax + ai) + f(x - g + gx^2 - a + ax^2 + aix) - aigd =\\
&= h(1 + gx + ax + ai) + f(x - g + gx^2 - a + ax^2 + aix) - aig(f - xe) =\\
&= h(1 + gx + ax + ai) + f(x - g + gx^2 - a + ax^2 + aix) - aigf - aigxe =\\
&= h(1 + gx + ax + ai) + f(x - g + gx^2 - a + ax^2 + aix - aig) - aigxe =\\
&= h(1 + gx + ax + ai) + f(x - g + gx^2 - a + ax^2 + aix - aig) - aigx(h + xf) =\\
&= h(1 + gx + ax + ai) + f(x - g + gx^2 - a + ax^2 + aix - aig) - aigxh + aigx^2 f =\\
&= h(1 + gx + ax + ai - aigx) + f(\underbrace{x - g + gx^2 - a + ax^2 + aix - aig + aigx^2}_{f_p})
\end{aligned}
$$

$$f_p = x - g + gx^2 - a + ax^2 + aix - aig + aigx^2 \pmod 3 = x^9 + x^7 + x^5 + 2x^4 + 2x^3 + 2x^2 + x.$$

Hivatkozások

[1] Michael A. Nielsen & Isaac L. Chuang: Quantum Computation and Quantum Information, http://mmrc.amss.cas.cn/tlb/201702/W020170224608149940643.pdf

[2] Nagy Benedek: Új számítási paradigmák, https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0103_28_uj_szamitasi_paradigmak/index.html

[3] Stephanie Wehner and Nelly Ng: edX Quantum Cryptography, https://courses.edx.org/courses/course-v1:CaltechDelftX+QuCryptox+3T2018/pdfbook/0/

[4] Markus J. Pflaum: The Geometry of Classical and Quantum Fields http://www.libermath.org/GeometryClassicalAndQuantumFields/viewpdf.html

[5] Samuel J. Lomonaco, JR.: A lecture on Shor's quantum factoring algorithm Version 1.1 https://arxiv.org/pdf/quant-ph/0010034.pdf

[6] Artur Ekert and Richard Jozsa: Quantum computation and Shor's factoring algorithm http://www.cs.tau.ac.il/~amnon/Classes/2003-Class-Quantum/Papers/ekert-joza-on-shor.p733_1.pdf

[7] Buttyán Levente, Vajda István: Kriptográfia és alkalmazásai

[8] Varga Zsolt: Modern prímfaktorizáció https://web.cs.elte.hu/blobs/diplomamunkak/bsc_alkmat/2012/varga_zsolt.pdf

[9] Koronka Gábor: Nagy számok faktorizációja Shor-algoritmussal https://web.cs.elte.hu/blobs/diplomamunkak/bsc_alkmat/2012/koronka_gabor.pdf

[10] NISTIR 8105: Report on Post-Quantum Cryptography https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf

[11] NAEP: Provable security in the presence of decryption failures https://eprint.iacr.org/2003/172.pdf

[12] NIST: Post-Quantum Cryptography, Round 3 Submissions https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions

[13] Abderrahmane Nitaj: The Mathematics of the NTRU Public Key Cryptosystem https://nitaj.users.lmno.cnrs.fr/ntru3final.pdf

[14] Radnai András: Rácselmélet alkalmazása a számelméletben https://web.cs.elte.hu/blobs/diplomamunkak/bsc_mat/2010/radnai_andras.pdf

[15] Jeffrey Hoffstein, Joseph H. Silverman, William Whyte: NTRU Cryptosystems Technical Report, Estimated Breaking Times for NTRU Lattices https://ntru.org/f/tr/tr012v2.pdf

[16] NTRUEncrypt and Lattice Attacks: Security estimation
https://www.semanticscholar.org/paper/NTRUEncrypt-and-Lattice-Attacks-Rosenberg/67e7020ce5649947e2199bc0eb8bd62b9a31ca4e

[17] Joseph H. Silverman: An Introduction to the Theory of Lattices and Applications to Cryptography
https://www.math.brown.edu/~jhs/Presentations/WyomingLattices.pdf

[18] Nicolas Sendrier, Raphael Overbeck: Code-Based Cryptography
https://www.researchgate.net/profile/Nicolas_Sendrier/publication/226115302_Code-Based_Cryptography/links/540d62d50cf2df04e7549388/Code-Based-Cryptography.pdf

[19] Ashley Valentijn: Goppa Codes and Their Use in the McEliece Cryptosystems
https://surface.syr.edu/cgi/viewcontent.cgi?article=1846&context=honors_capstone

[20] Michiel Marcus: White Paper on McEliece with Binary Goppa Codes
https://www.hyperelliptic.org/tanja/students/m_marcus/whitepaper.pdf

[21] Harshdeep Singh: Code based Cryptography: Classic McEliece
https://arxiv.org/pdf/1907.12754.pdf

[22] Vasileios Mavroeidis, Kamer Vishi, Mateusz D. Zych, Audun Jøsang: The Impact of Quantum Computing on Present Cryptography
https://arxiv.org/pdf/1804.00200.pdf

[23] Kiss Emil: Bevezetés az algebrába
https://regi.tankonyvtar.hu/hu/tartalom/tamop425/2011-0001-526_kiss_emil/index.html

[24] D. Engelbert, R. Overbeck, and A. Schmidt: A Summary of McEliece-Type Cryptosystems and their Security
https://www.researchgate.net/publication/220336630_A_Summary_of_McEliece-Type_Cryptosystems_and_their_Security

[25] Szőnyi Tamás jegyzetei
https://szonyitamas.web.elte.hu/oktelozo.html