McEliece titkosítás:

- kevés kriptográfiai ismeret;
- hibakorlátozó kódok alapos ismerete;
- jártasság a véges testekben.

Titkosítás ↔ hibakorlátozó kódolás:

- redundancia - titkosításnál megszüntetése, de legalábbis csökkentése (tömörítés az információszivárgás mérséklésére), kódolásnál szándékos növelése (terjengősség, hogy a zajban is felismerhető legyen az üzenet);
- visszaállítás - titkosításnál speciális ismeret nélkül legyen nehéz (lehetetlen), kódolásnál a minél könnyebb dekódolás.

Kódalapú rejtjelezés:

- adott a könnyen dekódolható, $[n, k, d]_q$-paraméterű C kód a $k \times n$-méretű $\mathbf{G}$ generátormátrixszal;
- választunk egy n-edrendű $\mathbf{P}$ permutációs és egy k-adrendű, reguláris $\mathbf{M}$ mátrixot;
- a nyilvános kulcs a $(\mathbf{G}', t)$ pár, ahol $\mathbf{G}' = \mathbf{MGP}$ és $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, a titkos kulcs $\mathbf{P}$, $\mathbf{M}$ és $\mathbf{G}$;
- a nyílt üzenet $\mathbf{u}$, a hozzá tartozó rejtjelezett üzenet $\mathbf{v}$, ahol $\mathbf{v}^T = \mathbf{u}^T\mathbf{G}' + \mathbf{e}^T$ és $\mathbf{e}$ egy n-méretű, t-súlyú szó;
- a visszafejtés: dekódoljuk $\mathbf{Pv}$-t, és ha ez $\mathbf{u}'$, akkor $\mathbf{u} = \mathbf{M}^{T^{-1}}\mathbf{u}'$.

$$(\mathbf{Pv})^T = \mathbf{v}^T\mathbf{P}^T = (\mathbf{u}^T\mathbf{G}' + \mathbf{e}^T)\mathbf{P}^T = \mathbf{u}^T\mathbf{MGPP}^T + \mathbf{e}^T\mathbf{P}^T = (\mathbf{u}^T\mathbf{M})\mathbf{G} + \mathbf{e}^T\mathbf{P}^T = \mathbf{u}'^T\mathbf{G} + \mathbf{e}'^T,$$

így dekódolás után $\mathbf{u}'^T = \mathbf{u}^T\mathbf{M}$-et kapunk, és ebből valóban $\mathbf{u} = \mathbf{M}^{T^{-1}}\mathbf{u}'$.
($\mathbf{c}$ oszlopvektort jelöl, ^{T} a transzponálás jele, így $\mathbf{c}^T$ a $\mathbf{c}$-nek megfelelő sorvektor; permutációs mátrix inverze a mátrix transzponáltja.)

A feladat egy könnyen dekódolható C kód konstruálása, majd ennek „álcázása” az $\mathbf{M}$ és $\mathbf{P}$ mátrixokkal úgy, hogy az így kapott $\mathbf{G}'$ mátrixból nehéz legyen megállapítani a dekódolási szabályokat.

Hibakorlátozó kódok:

- hibajelző kódok;
- hibajavító kódok.

A hibakorlátozó kódok fix szóhosszúságúak. Ha a kódábécé A, és $|A| = q$, továbbá a szóhosszúság n, akkor $C \subseteq A^n$. A kódolás egy $U \to C$ leképezés, ahol U az üzenetek halmaza. A hibakorlátozó kódoknál különböző üzenet kódja különböző, így $|U| \leq |C|$.

Hibajelzés: $\mathbf{v}'$ a vett szó, és $\mathbf{v}' \notin C$.

Hibajavítás: $\mathbf{v}'$ a vett szó, döntünk, hogy mi lehetett az elküldött $\mathbf{v}$ szó; ehhez **döntési függvény**. **Döntési hiba**: az eredeti szó $\mathbf{v}$, de a döntési függvény eredménye $D(\mathbf{v}') \neq \mathbf{v}$. Cél a döntési hiba minimálása.

A leggyakoribb döntési függvény a **minimális távolságú dekódolás**: a vett szóhoz legközelebbi, attól a legkevesebb helyen különböző kódszóra döntünk (probléma, ha több azonos, minimális távolságra lévő kódszó van). Bizonyos feltételek mellett a minimális távolságú dekódolásnál minimális a döntési hiba.

Távolság: két szó távolsága az eltérő komponensek száma – **Hamming-távolság**. **Kód távolsága** a páronként különböző kódszavak távolságainak minimuma.

Szó súlya a nullától különböző komponensek száma (a nullvektortól való távolsága).

Ha d a kód távolsága, és $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, akkor a kód

- jelez minden legfeljebb $d - 1$ hibát, de van olyan d hiba, amelyet nem jelez: a kód (pontosan) $d - 1$-hiba jelző;
- minimális távolságú dekódolással javít minden legfeljebb t hibát, de van olyan $t + 1$ hiba, amelyet nem javít vagy rosszul „javít”: a kód (pontosan) t-hiba javító a minimális távolságú dekódolással.

A titokmegosztásnál: **törlődéses hiba** olyan, hogy ismerjük a helyét, csak a hiba értéke ismeretlen; a d-távolságú kód minimális távolságú dekódolással javít $2e + r < d$ hibát, ha r a törlődéses hibák és e a többi hiba száma.

A hiba javítására például két módszer:

- minden szóhoz tároljuk a döntési függvény eredményét (a szóval indexelve elegendő a függvény értékét tárolni);
- tároljuk a kódszavakat, és a beérkezett szót mindegyikkel összehasonlítva kiválasztjuk a (kiválasztunk egy) minimális távolságra lévő kódszót.

Memória - futási idő trade off. A dekódolás a kódszavak hosszának nem polinomiális függvénye.

Ha ismerjük a hiba helyét és a hiba értékét, akkor a hiba javítható. Bináris kód esetén a hiba helyének ismerete kell csak.

A hatékonyság javítása: struktúra a rendszerben; **lineáris kód**ok.

Lineáris kód: a szimbólumhalmaz egy véges test, $\mathbb{F}_q$ (gyakran $GF(q)$), ahol q a test elemeinek száma. $\mathbb{F}_q^n$, az n-hosszúságú szavak összessége, lineáris tér $\mathbb{F}_q$ fölött. A C kód ennek a térnek egy k-dimenziós altere. Ha a kód távolsága d, akkor C egy $[n, k, d]_q$-paraméterű kód, és a kód mérete (elemeinek száma) $M = |C| = q^k$.

A kódoláshoz elegendő tárolni a kód egy bázisát. Az ezekből mint sorvektorokból álló $k \times n$-méretű $\mathbf{G}$ mátrix a kód **generátormátrix**a.

$\mathbf{G}$ elemei $\mathbb{F}_q$ elemei, és a mátrix rangja k. Az üzenetek az $\mathbb{F}_q$ fölötti k-hosszúságú szavak, azaz $\mathbb{F}_q^k$ elemei, és az $\mathbf{u} \in \mathbb{F}_q^k$ üzenet kódja $\mathbf{v} \in \mathbb{F}_q^n$, ahol $\mathbf{v}^T = \mathbf{u}^T\mathbf{G}$.

Az $\mathbf{u}^T \mapsto \mathbf{u}^T\mathbf{G}$ leképezés a **kódolás**.

A C egy másik bázisát, és így egy másik generátormátrixot választva ugyanazt a kódot kapjuk, de más lesz a kódolás, lesz olyan üzenet, amelynek a kódja más lesz.

$C^{\perp} = \{\mathbf{w} \in \mathbb{F}_q^n | \forall(\mathbf{v} \in \mathbb{F}_q^n): (\mathbf{v}, \mathbf{w}) = 0\}$, ahol $(\mathbf{v}, \mathbf{w}) = \sum_{i=0}^{n-1} v_i w_i$ a $\mathbf{v}$ és $\mathbf{w}$ **skalárszorzat**a, a $C \leq \mathbb{F}_q^n$-hez tartozó **ortogonális altér**. $C^{\perp} \leq \mathbb{F}_q^n$ $n-k$-dimenziós, és egy bázisából álló $\mathbf{H}$ mátrix a kód **ellenőrző mátrix**a. $\mathbf{H}$ egy $\mathbb{F}_q$ fölötti, $(n-k) \times n$-méretű, $n-k$-rangú mátrix.

A generátormátrixhoz hasonlóan, az ortogonális altér bármely bázisa ellenőrző mátrixot ad.

$\mathbf{H}$ rangja $n-k$, így van $n-k$ lineárisan független oszlopa, és így a kódnak van olyan $\mathbf{H}$ ellenőrző mátrixa, amely részmátrixként tartalmaz egy $n-k$-rendű egységmátrixot. Ekkor alkalmas $\mathbf{P}$ permutációs mátrixszal $\mathbf{HP} = \mathbf{H}' = (\mathbf{I}^{(n-k)} \quad \mathbf{E})$ alakú, és,$\mathbf{G}' = (-\mathbf{E}^T \quad \mathbf{I}^{(k)})$-vel $\mathbf{G} = \mathbf{G}'\mathbf{P}^T$ a kód egy generátormátrixa.

$\mathbf{v} \in \mathbb{F}_q^n$ akkor és csak akkor eleme a C kódnak, ha $\mathbf{Hv} = \mathbf{0}$. De $\mathbf{H}$ „ennél többet tud”.

$\mathbf{s} = \mathbf{Hv}$ a **szindróma**, a $\mathbf{v}$ szindrómája. $\mathbf{s} \in \mathbb{F}_q^{n-k}$.

$\mathbf{Hv}^{(1)} = \mathbf{Hv}^{(2)}$ akkor és csak akkor, ha $\mathbf{H}\left(\mathbf{v}^{(1)} - \mathbf{v}^{(2)}\right) = \mathbf{0}$, azaz ha $\mathbf{v}^{(1)} - \mathbf{v}^{(2)} \in C$, vagyis akkor és csak akkor, ha $\mathbf{v}^{(2)} = \mathbf{u} + \mathbf{v}^{(1)}$, ahol $\mathbf{u} \in C$.

Következmény: minden $\mathbf{s} \in \mathbb{F}_q^{n-k}$-hez q^k olyan $\mathbf{e} \in \mathbb{F}_q^n$, amelynek a szindrómája $\mathbf{s}$. Ezek közül pontosan egy a **javítható hibaminta**, az úgynevezett **mellékosztály-vezető**.

Ha a beérkezett szó $\mathbf{v}'$, ennek szindrómája $\mathbf{s}$, és $\mathbf{e}$ az $\mathbf{s}$-hez tartozó mellékosztály-vezető $\mathbf{e}$, akkor a döntési függvény $\mathbf{v} = \mathbf{v}' - \mathbf{e}$. Ez a **szindróma-dekódolás**.

A szindróma-dekódolás akkor minimális távolságú, ha minden $\mathbf{s}$ szindrómához a legkisebb (egy legkisebb) súlyú olyan szót választunk, amelynek szindrómája $\mathbf{s}$. Most a feladat az egyes osztályokhoz meghatározni a legkisebb súlyú (egy legkisebb súlyú) szót.

Legfeljebb t-súlyú hibák különböző osztályokban vannak, és ezek minimális súlyúak az osztályukban, így ezek mindig javítható hibaminták (ahogy annak lennie is kell minimális távolságú dekódolásnál).

Általános esetben a mellékosztály-vezetők megtalálása NP-teljes feladat [2]: meghatározzuk az összes 0-, 1-, ...,t-súlyú szóhoz tartozó szindrómát, így ezekhez a szindrómákhoz megvannak a mellékosztály-vezetők, majd sorban egymás után az ennél nagyobb súlyú olyan szavakhoz, amelyek szindrómája még nem szerepel, addig, míg megtaláltuk minden szindrómához a legkisebb súlyú (egy legkisebb súlyú) szót.

Vannak lineáris kódok, amelyeknél alkalmas ellenőrző mátrix választásával a szindrómából könnyű algoritmussal meghatározható a hiba, feltéve, hogy a hibák száma legfeljebb t. Kódolásra gyakorlatilag ezek a kódok használhatóak. Ugyanakkor ugyanezen kód egy másik ellenőrző mátrixából általában már nehéz a kód dekódolása.

Kód alapú rejtjelezéshez ilyen kódot használunk. Ilyen kód például a **Goppa-kód**.

Goppa-kód

- p prímszám, $s \in \mathbb{N}^+$, $q = p^s$, $\mathbb{F}_q$ a q-elemű test;
- $m \in \mathbb{N}^+$, $r \in \mathbb{N}^+$;
- $q^m \geq n \in \mathbb{N}^+$;
- $\boldsymbol{\alpha} = \{\alpha_i \in \mathbb{F}_{q^m} | n > i \in \mathbb{N}\}$, $|\boldsymbol{\alpha}| = n$;
- $g \in \mathbb{F}_{q^m}[x]$: $\deg g = r$, $\forall(\alpha_i \in \boldsymbol{\alpha})$: $\hat{g}(\alpha_i) \neq 0$;

$$\hat{g}(\alpha_i) \neq 0 \Rightarrow (g, x - \alpha_i) = e$$
$$\Rightarrow \exists\left(g^{(i)} \in \mathbb{F}_{q^m}[x]\right)\exists\left(t^{(i)} \in \mathbb{F}_{q^m}[x]\right): e = (x - \alpha_i)g^{(i)} + gt^{(i)} \wedge \deg g^{(i)} < \deg g\,;$$

- $C = \left\{\mathbf{c} \in \mathbb{F}_q^n \middle| g \left| \sum_{i=0}^{n-1} c_i g^{(i)} \right.\right\}$.

$g^{(i)} = -\left(\hat{g}(\alpha_i)\right)^{-1} \frac{g - \hat{g}(\alpha_i)}{x - \alpha_i} = h_i \frac{g - \hat{g}(\alpha_i)}{x - \alpha_i}$, ahol $h_i = -\left(\hat{g}(\alpha_i)\right)^{-1}$.

A fenti Goppa-kódot $\Gamma(q, m, g, \boldsymbol{\alpha})$, vagy röviden $\Gamma(g, \boldsymbol{\alpha})$ jelöli. g egyben megadja r-et is.

A dekódoláshoz egy $r \times n$-méretű $\mathbf{H}$ mátrixot használunk, ahol $H_{i,j} = h_j \alpha_j^i$.

$\mathbf{H}$-ból kapjuk, hogy a kód távolsága $d \geq r + 1$.

Az előbbi $\mathbf{H}$ mátrixból számoljuk a szindrómát, ám a kódnak nem ez az ellenőrző mátrixa: a kód $\mathbb{F}_q$ fölötti, de $\mathbf{H}$ elemei $\mathbb{F}_{q^m}$-beliek. A kód dimenziójának valamint generátormátrixának meghatározásához kell a $\tilde{\mathbf{H}}$ ellenőrző mátrix.

Legyen $\left\{\mathbf{b}^{(i)}\middle| m > i \in \mathbb{N}\right\}$ az $\mathbb{F}_{q^m}$ egy $\mathbb{F}_q$ fölötti bázisa. A $\mathbf{H}$ minden eleme felírható ebben a bázisban: $h_{i,j} = \sum_{k=0}^{m-1} a_k^{(i,j)} \mathbf{b}^{(k)}$. $\mathbf{H}$-ban írjuk $h_{i,j}$ helyére az $a_k^{(i,j)}$-kat tartalmazó oszlopvektort. Az így kapott $\mathbf{H}'$ mátrix egy $mr \times n$-méretű, $\mathbb{F}_q$ fölötti mátrix. $\mathbf{H}$ sorai lineárisan függetlenek, de $\mathbf{H}'$ sorai már nem feltétlenül azok. Legyen $\widetilde{\mathbf{H}}$ olyan mátrix, amely $\mathbf{H}'$ maximális számú lineárisan független sorából áll. A sorok száma most maximum mr, és minimum r, mert $\mathbf{H}$ r sora lineárisan független, így van r lineárisan független oszlopa, és a megfelelő oszlopok $\mathbf{H}'$-ben is lineárisan függetlenek.

A fentiek alapján a kód dimenziója $n - mr \leq k \leq n - r$.

$\widetilde{\mathbf{H}}$-ból meghatározható a kód generátormátrixa, $\mathbf{G}$.

A Goppa-kód dekódolása például euklideszi algoritmussal történhet (más módszer például Patterson algoritmusa, Berlekamp-algoritmus).

A dekódolás:

- C egy $\Gamma(q, m, g, \boldsymbol{\alpha})$ kód, ahol $\boldsymbol{\alpha}$ egyetlen komponense sem 0;
- $t_0 = \left\lfloor \frac{r}{2} \right\rfloor$, ekkor C t_0 hibát biztosan javít;
- $\boldsymbol{\varepsilon}$ a hibavektor, $1 \leq w(\boldsymbol{\varepsilon}) = t \leq t_0$, ahol $w(\boldsymbol{\varepsilon})$ a hibavektor súlya, vagyis a hibahelyek száma;
- $J = \{j_i \in \mathbb{N} | t > i \in \mathbb{N} \wedge n > j_i\}$ a hibás pozíciók indexeinek halmaza;
- $t > i \in \mathbb{N}$-re $X_i = \alpha_{j_i}$, $Y_i = \varepsilon_{j_i}$;

Ha ismerjük az X_i-ket és Y_i-ket, akkor ismerjük a hibás pozíciókat, és a hibás helyeken a hiba értékét, így a javítás már elvégezhető. Kérdés, hogy hogyan tudjuk ezeket az értékeket meghatározni.

Legyen $\mathbf{v}$ a vett szó az $\boldsymbol{\varepsilon}$ hibával, és $\mathbf{s} = \mathbf{Hv} = \mathbf{H\varepsilon} \neq \mathbf{0}$ a szindróma.

- $0 \leq i < r$-re $s_i = (\mathbf{H\varepsilon})_i = \sum_{j=0}^{n-1} H_{i,j}\varepsilon_j = \sum_{j \in J} H_{i,j}\varepsilon_j = \sum_{l=0}^{t-1} H_{i,j_l}\varepsilon_{j_l} = \sum_{l=0}^{t-1} h_{j_l} X_l^i Y_l$;
- $\sigma = \prod_{i=0}^{t-1}(e - X_i x)$; $\hat{\sigma}(0) = e$, $\deg(\sigma) = t \leq \frac{r}{2}$, és $\hat{\sigma}(u) = 0 \Leftrightarrow \exists (t > l \in \mathbb{N})\colon u = X_l^{-1}$;
- $t > i \in \mathbb{N}$-re $\sigma^{(i)} = \prod_{\substack{l=0 \\ l \neq i}}^{t-1}(e - X_l x)$, és $\omega = \sum_{i=0}^{t-1} h_{j_i} Y_i \sigma^{(i)}$; $\hat{\sigma}^{(j)}\left(X_i^{-1}\right) = 0 \Leftrightarrow i \neq j$, $h_{j_i} \neq 0$,

így $\hat{\omega}\left(X_i^{-1}\right) = \sum_{l=0}^{t-1} h_{j_l} Y_l \hat{\sigma}^{(l)}\left(X_i^{-1}\right) = h_{j_i} Y_i \hat{\sigma}^{(i)}\left(X_i^{-1}\right)$, és ebből $Y_i = \frac{\hat{\omega}\left(X_i^{-1}\right)}{h_{j_i}\hat{\sigma}^{(i)}\left(X_i^{-1}\right)}$;

- $\omega \neq 0$, $\deg(\omega) \leq t - 1$, $(\sigma, \omega) = e$;
- $S = \sum_{i=0}^{r-1} s_i x^i$, $\deg(S) \leq r - 1$.

σ a **hibahelypolinom**, ω a **hibaérték-polinom**, S a **szindrómapolinom**.

A dekódolás alapja, hogy $x^r|\omega - \sigma S$. Ebből $\omega = \vartheta x^r + \sigma S$ egy valamilyen ϑ polinommal. Ez azt mutatja, hogy ω az x^r és S (ismert) polinomok lineáris kombinációja, és az S együtthatója σ.

A σ és ω meghatározása:

- kiterjesztett euklideszi algoritmus, leállás, amikor $\deg(r_{k-1}) \geq \frac{r}{2}$, $\deg(r_k) < \frac{r}{2}$ (r_i az osztási maradék);
- $\sigma = \left(\hat{b}_k(0)\right)^{-1} b_k$, $\omega = \left(\hat{b}_k(0)\right)^{-1} r_k$ ($r_i = a_i x^r + b_i S$ a kiterjesztett euklideszi algoritmusban).

A **kiterjesztett euklideszi algoritmus** test feletti polinomokra:
meghatározandó az f és g polinom legnagyobb közös osztója, d, továbbá olyan a és b polinom, hogy $d = af + bg$;
ha egyik polinom sem osztója a másiknak, akkor $\deg a < \deg g$, $\deg b < \deg f$;

Az alábbi algoritmusban $quo(f, g)$ az f g-vel való maradékos osztásánál a hányados, és $rem(f, g)$ a maradék.

euklidesz(f, g, d, a, b)
$a = e;$
$b = 0;$
$a0 = 0;$
$b0 = e;$
$u = f;$
$v = g;$
ciklus amíg $v \neq 0;$
$q = quo(u, v);$
$s = a - q * a0;$
$a = a0;$
$a0 = s;$
$s = b - q * b0;$
$b = b0;$
$b0 = s;$
$s = rem(u, v);$
$u = v;$
$v = s;$
ciklus vége
$d = u;$
euklidesz vége.

A dekódolásnál eltérés az algoritmusban, hogy a leállása feltétele $\deg v < \frac{r}{2}$, és ekkor a kimenet paraméterei v, $a0$ és $b0$:

Goppa_euklidesz$(f, g, v, a0, b0)$

$a = e;$

$b = 0;$

$a0 = 0;$

$b0 = e;$

$u = f;$

$v = g;$

ciklus amíg $\deg(v) \geq \frac{r}{2}$;

$q = quo(u, v);$

$s = a - q * a0;$

$a = a0;$

$a0 = s;$

$s = b - q * b0;$

$b = b0;$

$b0 = s;$

$s = rem(u, v);$

$u = v;$

$v = s;$

ciklus vége

Goppa_euklidesz vége.

- $\sigma = \left(\widehat{b0}(0)\right)^{-1} b0, \omega = \left(\widehat{b0}(0)\right)^{-1} v.$

Mellékeredmény: ha van hiba, de a hibák száma nem haladja meg $\frac{r}{2}$-t, akkor az egyébként legfeljebb $r-1$-edfokú S polinom foka legalább $\frac{r}{2}$.

$q = 2$, vagyis bináris Goppa-kód esetén: g négyzetmentes $\Rightarrow \Gamma(g, \boldsymbol{\alpha}) = \Gamma(g^2, \boldsymbol{\alpha}) \Rightarrow d \geq 2r + 1$.

A Goppa-kód „jó kód”.

Jó kód: kódok egy családjában létezik a kódok olyan sorozata, hogy a kódsebesség és a relatív távolság határértéke is pozitív.

Kódsebesség: q-elemű ábécével, azonos szóhosszúsággal kódolva egy M-elemű üzenethalmazt a minimális szóhosszúság $n_{min} = \lceil \log_q M \rceil$. Ha az üzeneteket n-hosszúságú szavakkal kódoljuk, akkor például soros átvitel esetén egy-egy üzenet átviteléhez $\frac{n}{n_{min}}$-szor több idő kell, vagyis a sebesség az $\frac{n_{min}}{n}$ arányban csökken. Ennek megfelelően a q-elemű ábécé feletti (n, M, d)-paraméterű kód kódsebessége (a kód nem feltétlenül lineáris, és M a kód mérete, azaz a kódszavak száma) $\mathcal{R} = \frac{1}{n} \log_q M$.

Relatív távolság: az n-szóhosszúságú, d-távolságú kód relatív távolsága $\delta = \frac{d}{n}$. A relatív távolság jelzi, hogy milyen arányban tudunk minimális távolságú dekódolással hibát javítani.

A q-elemű ábécével felírt $n$szóhosszúságú szavak halmazában egy szótól legfeljebb t távolságra lévő szavak száma, azaz az adott szó mint középpont körüli t-sugarú gömb térfogata $V_q(n,t) = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$

Kódolási korlátok: $A_q(n,d)$ a q-elemű ábécével kódolt, n szóhosszúságú, d távolságú kódok méretének maximuma. Pontos érték csak triviális esetekre van, az egyéb esetekre felső és alsó határok léteznek.

Felső határ például

- a **Hamming-korlát** (**gömbkitöltési korlát**, **gömbpakolási korlát**): $A_q(n,d) \leq \frac{q^n}{V_q(n,t)}$, ahol $t = \left\lfloor \frac{d-1}{2} \right\rfloor$. Ha egy kódnál egyenlőség van, akkor a kód **tökéletes** vagy **perfekt**, ilyen például a **Hamming-kód**; lineáris kódokra a korlát alakja $V_q(n,t) \leq q^{n-k}$;
- a **Singleton-korlát**: $A_q(n,d) \leq q^{n-d+1}$, és lineáris kódokra $k \leq n-d+1$, és ebből a fontos korlát a távolságra $d \leq n-k+1$; ha itt egyenlőség van, akkor a kód **MDS-kód** (Maximum Distance Separable); példa a **Reed-Solomon kód**.

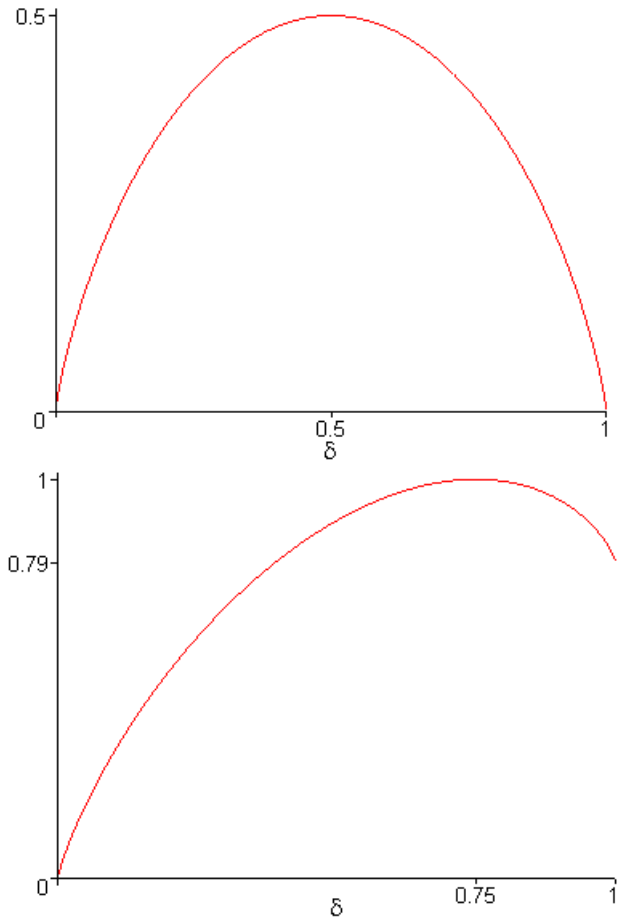
Alsó határ például

- **Varshamov-Gilbert korlát**: $A_q(n,d) \geq \frac{q^n}{V_q(n,d-1)}$, lineáris kódoknál $V_q(n,d-1) \geq q^{n-k}$ (tulajdonképpen ezek a **Gilbert-korlát**ok, a **Varshamov-korlát** egy picit erősebb, és lineáris kódokról szól);

Aszimptotikus korlát: $A_q(n,d) = A_q(n,\delta n)$, és $a_q(\delta) = \overline{\lim_{n\to\infty}} \frac{1}{n}\log_q A_q(n,\delta n)$. $\frac{1}{n}\log_q A_q(n,\delta n)$ a maximális méretű kód kódsebessége, így $a_q(\delta)$ a relatív távolság, azaz a hibajavító képesség függvényében adja a kódsebességet.

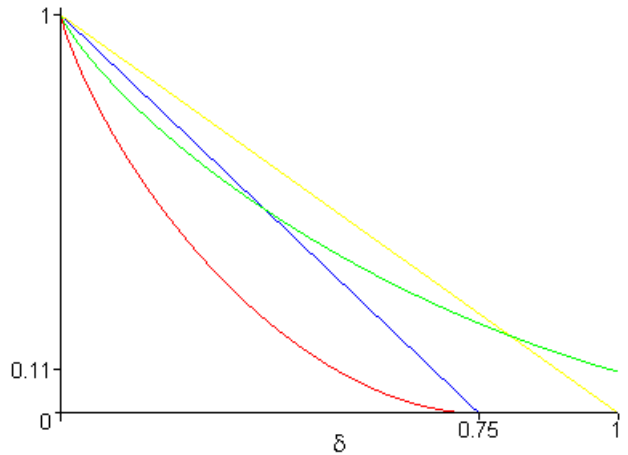
Aszimptotikus Varshamov-Gilbert korlát: $a_q(\delta) \geq 1 - H_q^*(\delta)$, ha $0 \leq \delta \leq 1 - \frac{1}{q}$. Az előbbi kifejezésben $H_q^*(\delta) = H_q(\delta) + \delta\log_q(q-1)$, és $H_q(\delta) = -\delta\log_q\delta - (1-\delta)\log_q(1-\delta)$ a **Shannon-féle entrópia**.

$q = 4$ esetén $H_q(\delta)$ és $H_q^*(\delta)$ ebben a sorrendben:



$H_q^*(\delta)$-nak $1 - \frac{1}{q}$-ban van a maximuma. Ha $q = 2$, akkor $H_q^*(\delta) = H_q(\delta)$.

Az aszimptotikus korlátok, ismét $q = 4$ esetén:



Az alsó görbe a Varshamov-Gilbert korlát, a felső a Hamming-korlát, és a felső egyenes a Singleton-korlát.

A McEliece rendszer pici **elemzése** (McEliece eredeti cikke ([1]) alapján):

Két lehetőség:

- valahogy meghatározni $\mathbf{G}'$-ből $\mathbf{G}$-t; ha ez sikerül, akkor feltörtük a rendszert;
- valahogy visszafejteni a konkrét rejtjelszövegből az eredeti üzenetet $\mathbf{G}$ ismerete nélkül.

Az első eset gyakorlatilag reménytelen a paraméterek megfelelő választása esetén. Ha $\mathbf{G}$ sorainak száma, azaz a kód dimenziója k, akkor az $\mathbb{F}_q$ fölöttii, k-adrendű reguláris mátrixok száma

$$\prod_{i=0}^{k-1}\left(q^k - q^i\right) \geq \prod_{i=0}^{k-1}\left(q^k - q^{k-1}\right) = \left(q^{k-1}(q-1)\right)^k,$$

a bináris esetben $\prod_{i=0}^{k-1}\left(2^k - 2^i\right) \geq 2^{k(k-1)}$, az n-edrendű permutációs mátrixok száma pedig $n!$. Ám $\mathbf{G}$ ismeretében még mindig nem ismerjük $\mathbf{H}$-t, ami a könnyű dekódoláshoz kellene.

A második módszernél egy lehetőség a nyers erő (brute force attack), ami azt jelenti, hogy meghatározzuk az összes kódszót, és ebből kiválasztjuk a rejtjelünkhöz legközelebbit. Valamivel kevesebb, de még mindig reménytelenül sok munka meghatározni a mellékosztály-vezetőket.

Reménykeltőnek tűnhetne az n koordinátából kiválasztani k-t, és reménykedni, hogy ezek nem tartalmaznak a hiba-vektorból nem nulla komponenst. Ekkor egy k egyenletből álló, k ismeretlent tartalmazó lineáris egyenletrendszert kell megoldani, ahol az együtthatómátrix reguláris. Annak a valószínűsége, hogy jól választottuk a koordinátákat, $\left(1-\frac{t}{n}\right)^k$. Egy-egy választásnál az egyenlet megoldásához nagyjából k^3 lépés kell, így a sikeres fejtéshez nagyjából $k^3\left(1-\frac{t}{n}\right)^{-k}$ műveletet kell végezni. Az eredeti cikkben például $n = 1024$, $t = 50$, $k \approx 524$, így $k^3\left(1-\frac{t}{n}\right)^{-k} \approx 10^{19}$.

Fontos megjegyezni, hogy ugyanazt az üzenetet nem szabad kétszer, egymástól függetlenül titkosítva, elküldeni. Ha ugyanis a két esetben a szándékosan generált hibavektor $\mathbf{e}^{(1)}$ és $\mathbf{e}^{(2)}$, akkor a két rejtjelszöveg különbsége csak a két hibavektor $\mathbf{e} = \mathbf{e}^{(1)} - \mathbf{e}^{(2)}$ különbsége. Ennek súlya maximum $2t = d - 1 \leq n - k$, és ebből a biztosan hibátlan helyek száma $n - 2t \geq k$. Ha $w(\mathbf{e})$ közel van $2t$-hez, akkor viszonylag könnyű k olyan pozíciót találni, ahol a rejtjel betűi azonosak az üzenet hibátlan kódjával, így a rejtett szöveg könnyen fejthető.

A struktúra felépítése:

- n, m és r megválasztása (általában $n = 2^m$ vagy $n = 2^m - 1$);
- 2^m-elemű test generálása; ehhez egy, a kételemű test fölött irreducibilis m-edfokú polinom keresése;
- $\boldsymbol{\alpha}$ megválasztása ($n = 2^m$ vagy $n = 2^m - 1$ esetén ez a teljes test vagy annak nem nulla elemei);
- a 2^m-elemű test fölötti r-edfokú, négyzetmentes polinom keresése, amelynek $\boldsymbol{\alpha}$ elemei nem gyökei (például irreducibilis $\mathbb{F}_{2^m}$ fölött);
- $\mathbf{H}$, majd $\widetilde{\mathbf{H}}$ és ebből $\mathbf{G}$ meghatározása;
- $\mathbf{M}$ és $\mathbf{P}$ megválasztása;
- $\mathbf{G}' = \mathbf{MGP}$ és $t = \left\lfloor \frac{r}{2} \right\rfloor$ nyilvánossá tétele.

$\mathbf{M}$ sűrű mátrix kell, hogy legyen, azaz kevés 0-t tartalmazzon.

A McEliece rendszer nem alkalmas digitális aláírásra, mivel csak a kódszavaktól legfeljebb t távolságra lévő szavakra működne az ellenőrzés. Egy módosítással azonban aláírásra is alkalmas rendszer kapható.

Niederreiter titkosítás:

- választunk egy $[n, k]$-paraméterű bináris, legalább t hibát javító Goppa-kódot;
- meghatározzuk a kód $(n-k) \times n$-méretű $\mathbf{H}$ ellenőrző mátrixát;
- választunk egy n-edrendű $\mathbf{P}$ permutációs és egy $(n-k)$-rendű $\mathbf{M}$ reguláris mátrixot;
- kiszámítjuk a $\mathbf{H}' = \mathbf{MHP}$ mátrixot;
- a nyilvános kulcs a $(\mathbf{H}', t)$ pár, titkos $\mathbf{M}$, $\mathbf{H}$ és $\mathbf{P}$.

A titkosítás:

- a nyílt szöveg egy n-hosszúságú, maximum t-súlyú bináris vektor, $\mathbf{m}$;
- a rejtjelezett szöveg az $m-k$-hosszúságú $\mathbf{c} = \mathbf{H}'\mathbf{m}$ vektor.

A legális fejtés:

- $\mathbf{c}' = \mathbf{M}^{-1}\mathbf{c} (= \mathbf{HPm})$;
- $\mathbf{c}'$-ből $\mathbf{m}' = \mathbf{Pm}$ meghatározása;
- $\mathbf{m} = \mathbf{P}^T\mathbf{m}'$.

A McEliece és a Niederreiter a biztonság szempontjából azonos, de az utóbbi gyorsabb, és jobban alkalmas aláírásra, bár ennek is vannak problémái. A McEliece-nél a paramétereket megfelelően kell választani ahhoz, hogy alkalmas legyen aláírásra (lásd [3]).

A rendszerre újabban van viszonylag gyors fejtési módszer, amely a sebességét a párhuzamos, egymástól független számításoknak köszönheti. Az eredeti McEliece rendszer 200 géppel nagyjából 7 nap alatt volt törhető 2008-ban ([4]).

Irodalom:

1. R. J. McEliece: A Public-Key Cryptosystem Based On Algebraic Coding Theory; DSN Progress report 42-44, 1978 január és február; http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF
2. E. R. Berlekamp, R. J. McEliece, H. C. A. van Tilborg: On the Inherent Intractability of Certain Codiig Problems; IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-24, NO. 3, MAY 1978, pp. 384-386.; http://authors.library.caltech.edu/5607/1/BERieeetit78.pdf
3. N. Courtois, M. Finiasz, N. Sendrier: How to achieve a McEliece-based Digital Signature Scheme, Advances in Cryptology – ASIACRYPT 2001, Lecture Notes in Computer Science V. 2248, pp. 157-174; https://www.iacr.org/archive/asiacrypt2001/22480158.pdf
4. D. J. Bernstein, T. Lange, C. Peters: Attacking and defending the McEliece cryptosystem, Post-Quantum Cryptography in Lecture Notes in Computer Science V. 5299, pp. 31-46; https://cr.yp.to/codes/mceliece-20080807.pdf
5. V. C. Huffman, V. Pless: Fundamentals of Error-Correcting Codes, Cambridge University Press, 2003
6. J. H. van Lint: Introduction to Coding Theory, Springer-Verlag, 1982
7. F. J. MacWilliams, M. J. A. Sloane: The Theory of Error-Correcting Codes, North-Holland, 1977
8. Gonda, J.: Véges testek, 2011 http://www.inf.elte.hu/karunkrol/digitkonyv/Jegyzetek2011/GondaJanos-VegesTestek15.pdf
9. Gonda, J.: Hibakorlátozás; http://compalg.inf.elte.hu/material/DOWNLOAD/hibakor.pdf