Kód alapú posztkvantum kriptográfia

tudománynépszerűsítő előadás

Nagy Gábor Péter

Szegedi Tudományegyetem
Bolyai Intézet

2022. május 9.

Tagolás

1. Bonyolultságelmélet
   - Számításos feladatok
   - Algoritmus bonyolultsága
2. Kriptográfia
   - Elméleti alapelvek
   - Napjaink kriptorendszerei
   - A kriptográfia matematikai fogalmai
3. Hibajavító kódok
   - Alapfogalmak
   - Dekódolási algoritmusok
   - Résztest részkódok
4. Kód alapú kriptográfia
   - Klasszikus McEliece
   - Kriptoanalízis

Tagolás

Mi a probléma?

Az alábbiakban a *számítás* fogalmát általánosan értelmezzük.

A Feladat

Adott egy számítási feladat, aminek egy X inputhoz egy Y outputot kell hozzárendelnie.

- A feladat matematikai értelemben világosan van meghatározva.
- Az input és az output állhatnak több részből.
- Az input és az output véges méretűek.

Példák:

- Két egész szám szorzata.
- Hamilton-kör keresése.
- Lineáris egyenletrendszer megoldása.
- Prímtényezős felbontás.
- Két egész szám legnagyobb közös osztója.

Mi a probléma?

Az alábbiakban a *számítás* fogalmát általánosan értelmezzük.

A Feladat

Adott egy számítási feladat, aminek egy X inputhoz egy Y outputot kell hozzárendelnie.

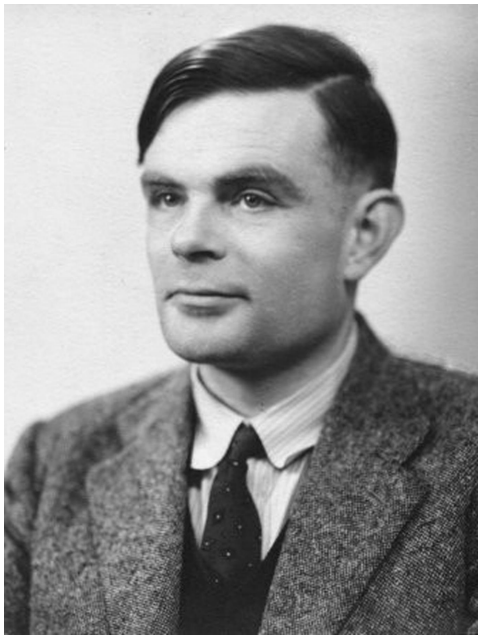
- A feladat matematikai értelemben világosan van meghatározva.
- Az input és az output állhatnak több részből.
- Az input és az output véges méretűek.

Példák:

- Két egész szám szorzata.
- Hamilton-kör keresése.
- Lineáris egyenletrendszer megoldása.
- Prímtényezős felbontás.
- Két egész szám legnagyobb közös osztója.

Mi a megoldás?

- Az ilyen típusú feladatokra a megoldás általában egy eljárást, egy algoritmust jelent.
- Az **algoritmus** fogalmának precíz meghatározása a 20. század elején vált szükségessé.



Alan Turing
brit matematikus
1912-1954

Neumann János
magyar-amerikai matematikus
1905-1957

Számító gépek

- Alan Turing 1936-ban publikálta a *Computable numbers* c. dolgozatát.
- Ebben egy elméleti modell alkotott egy univerzális gépre, amit my *Turing-gépnek hívunk.*
- Ennek segítségével lehet definiálni az **algoritmus,** a **kiszámíthatóság,** és a **bonyolultság** fogalmát.
- A gép a gyakorlatban is megvalósítható, de érdemi matematikai számításokra csak korlátozottan alkalmas.
- Sok ötletet alkalmaztak a brit *Blechley Park* kódtörői a német titkosítási eljárások feltöréséhez.

- A Neumann János által 1945-ben javasolt számítógép-architektúra a mai napig meghatározó.
- Ez a **CPU,** a **memória,** és az **input/output eszközök** elhelyezkedését és kapcsolatait írja le a *digitális* számítógépben.

Számító gépek

- Alan Turing 1936-ban publikálta a *Computable numbers* c. dolgozatát.
- Ebben egy elméleti modell alkotott egy univerzális gépre, amit my *Turing-gépnek hívunk.*
- Ennek segítségével lehet definiálni az **algoritmus,** a **kiszámíthatóság,** és a **bonyolultság** fogalmát.
- A gép a gyakorlatban is megvalósítható, de érdemi matematikai számításokra csak korlátozottan alkalmas.
- Sok ötletet alkalmaztak a brit *Blechley Park* kódtörői a német titkosítási eljárások feltöréséhez.

- A Neumann János által 1945-ben javasolt számítógép-architektúra a mai napig meghatározó.
- Ez a **CPU,** a **memória,** és az **input/output eszközök** elhelyezkedését és kapcsolatait írja le a *digitális* számítógépben.

Függvény nagyságrendje

Függvény nagyságrendje

Azt mondjuk, hogy a két $f, g : \mathbb{R} \to \mathbb{R}$ függvény azonos nagyságrendű, ha léteznek $A, B \in \mathbb{R}$ konstansok úgy, hogy

$$Ag(x) \leq f(x) \leq Bg(x) \qquad \text{és} \qquad Af(x) \leq g(x) \leq Bf(x)$$

teljesül minden $x \in \mathbb{R}$ esetén.

Jelölés: $f \approx g$.

Példák (majdnem...)

$$f(x) = \frac{x(x-1)}{2} \qquad g(x) = 100x^2 - 1 \tag{1}$$

$$f(x) = \log_2(x) \qquad g(x) = \log_{10}(x+1) \tag{2}$$

$$f(x) = 2^x \qquad g(x) = 2^{x+3} - x^{100} \tag{3}$$

Algoritmus bonyolultsága

Algoritmus bonyolultsága *(time complexity)*

- Tekintsünk egy **F** számítási feladatot.
- Legyen $\mathcal{A}$ az **F** egy megoldási algoritmusa.

Azt mondjuk, hogy az $\mathcal{A}$ algoritmus bonyolultsága $f(n)$, ha

- az *általános* n méretű input esetén
- az $\mathcal{A}$ végrehajtása nagyságrendileg $f(n)$ lépésben történik.

Példa: Egész számok szorzása

- A megoldási algoritmus az alsó tagozatban tanult „szorzás papíron”.
- Az input mérete a tényezők számjegyinek a száma.
- Az input általános, ha a számok nem speciális alakúak, pl. egyik sem 10-hatvány.
- Az algoritmus bonyolultsága n^2.

Algoritmus bonyolultsága

Algoritmus bonyolultsága *(time complexity)*

- Tekintsünk egy **F** számítási feladatot.
- Legyen $\mathcal{A}$ az **F** egy megoldási algoritmusa.

Azt mondjuk, hogy az $\mathcal{A}$ algoritmus bonyolultsága $f(n)$, ha

- az *általános* n méretű input esetén
- az $\mathcal{A}$ végrehajtása nagyságrendileg $f(n)$ lépésben történik.

Példa: Egész számok szorzása

- A megoldási algoritmus az alsó tagozatban tanult „szorzás papíron”.
- Az input mérete a tényezők számjegyinek a száma.
- Az input általános, ha a számok nem speciális alakúak, pl. egyik sem 10-hatvány.
- Az algoritmus bonyolultsága n^2.

Könnyű vagy nehéz?

Polinomiális, exponenciális bonyolultság

Legyen $\mathcal{A}$ az **F** feladat egy megoldási algoritmusa.

- Azt mondjuk, hogy az $\mathcal{A}$ polinomiális, ha bonyolultsága n^α ($\alpha > 0$).
- Azt mondjuk, hogy az $\mathcal{A}$ exponenciális, ha bonyolultsága β^n ($\beta > 1$).

- Az **F** feladatot **könnyűnek** nevezzük, ha ismert hozzá polinomiális megoldó algoritmus.
- Az **F** feladatot **nehéznek** nevezzük, ha ismert hozzá exponenciális megoldó algoritmus, de nem ismert polinomiális.

Példa: Faktorizáció és legnagyobb közös osztó

- Az egész számok prímtényezős felbontása **nehéz.**
- A legnagyobb közös osztó kiszámítása **könnyű,** mert az *euklideszi algoritmus* polinomiális (lineáris) bonyolultságú megoldási algoritmus.

Könnyű vagy nehéz?

Polinomiális, exponenciális bonyolultság

Legyen $\mathcal{A}$ az **F** feladat egy megoldási algoritmusa.

- Azt mondjuk, hogy az $\mathcal{A}$ polinomiális, ha bonyolultsága n^α ($\alpha > 0$).
- Azt mondjuk, hogy az $\mathcal{A}$ exponenciális, ha bonyolultsága β^n ($\beta > 1$).

- Az **F** feladatot **könnyűnek** nevezzük, ha ismert hozzá polinomiális megoldó algoritmus.
- Az **F** feladatot **nehéznek** nevezzük, ha ismert hozzá exponenciális megoldó algoritmus, de nem ismert polinomiális.

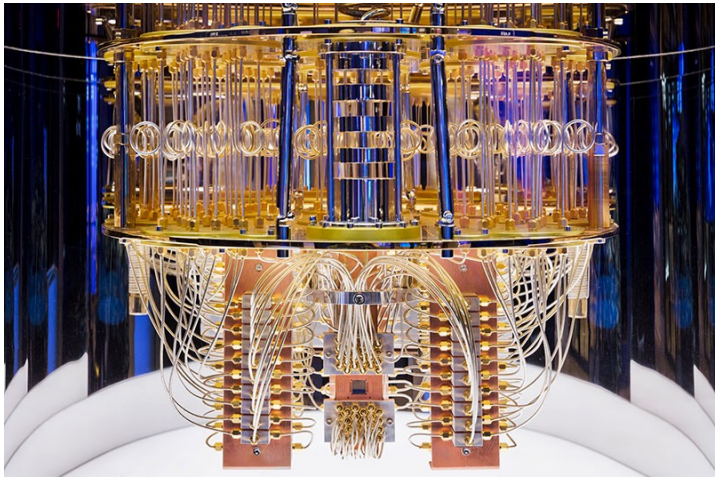
Példa: Faktorizáció és legnagyobb közös osztó

- Az egész számok prímtényezős felbontása **nehéz.**
- A legnagyobb közös osztó kiszámítása **könnyű,** mert az *euklideszi algoritmus* polinomiális (lineáris) bonyolultságú megoldási algoritmus.

A faktorizáció kvantum-könnyű

- Kvantumkomputerek a Turing-Neumann modelltől eltérő elven működnek.
- A kvantumalgoritmusok bonyolultságát a kvantum-biteken *(qubit)* végzett műveletek számával adjuk meg.
- Peter Shor 1994-ben polinomiális megoldó (kvantum)algoritmust adott a prímtényezős felbontásra.

Peter Shor (MIT)
2017-ben



IBM 100-qubit kvantumszámítógépe
2021-ben

Tagolás

Titkosírások tudományos alapja

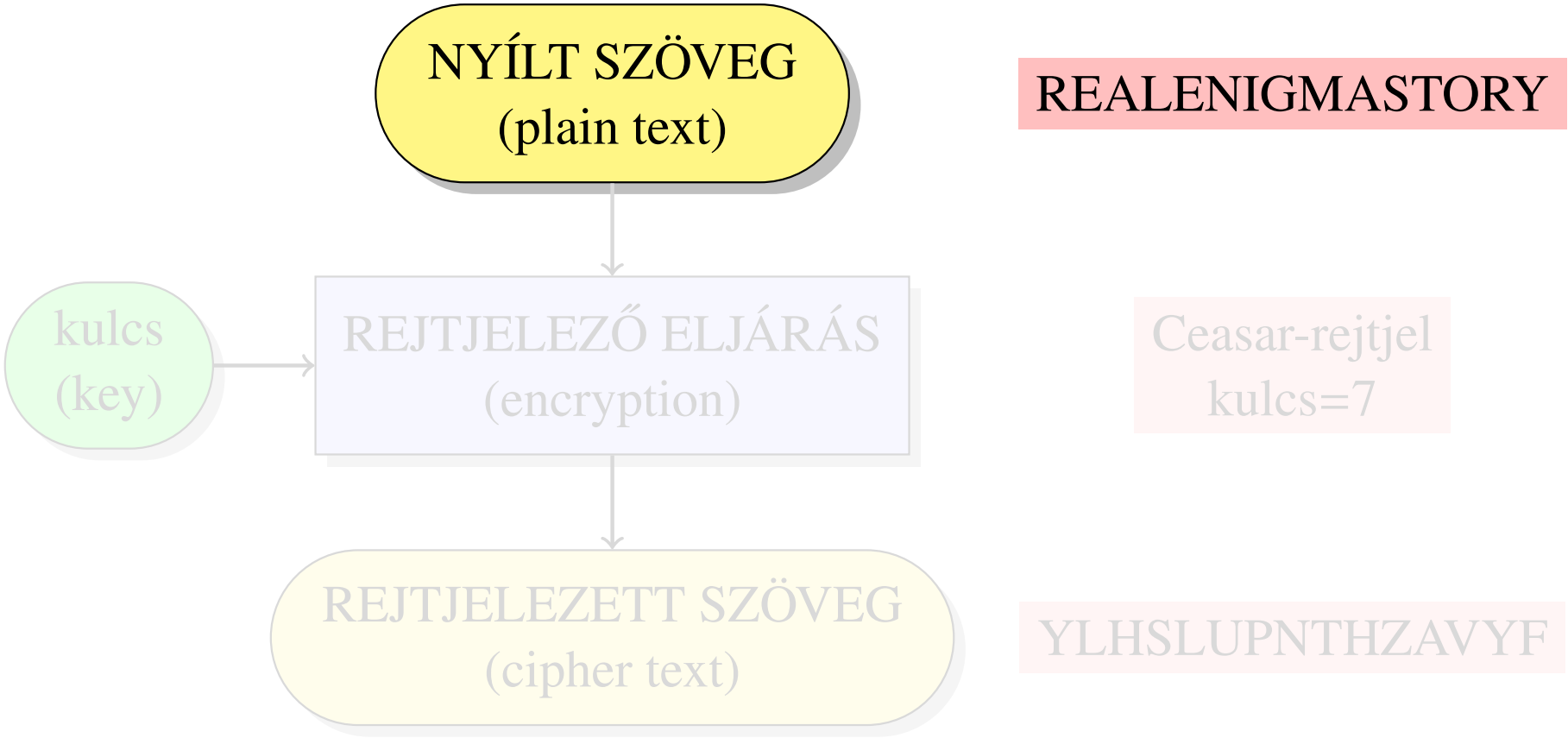
A történeti háttér:

- A titkosírások modern módszerei a 19. század elejétől terjedtek el.
- Ezeket a mai napig elsősorban katonai vagy üzleti célokra használják.
- Ekkortól próbálják a titkosítási technikákat tudományos alapokra helyezni

A KRIPTOANALÍZIS tudományos módszerei:

- Hagyományos: Nyelvészet, matematikai statisztika
- Modern: Absztrakt algebra, bonyolultságelmélet, számítástudomány, elliptikus görbék, pszeudo-véletlen sorozatok, kvantumalgoritmusok

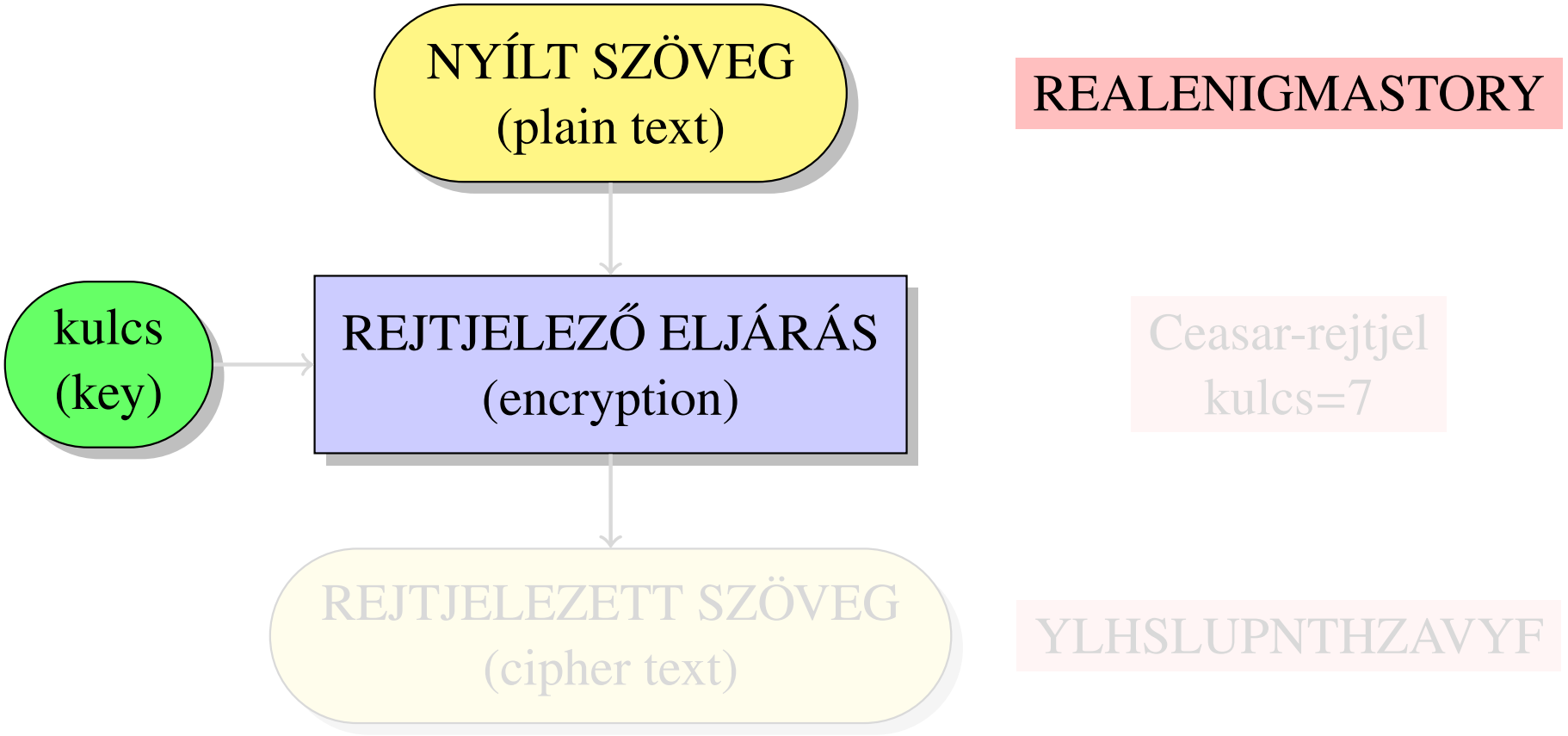
A kriptorendszer alapfogalmai: Rejtjelezés



A kriptorendszer kulcsterének fogalma

A kriptorendszer olyan kulcsainak halmaza, amik lényegesen különböző rejtjelezett szövegeket eredményeznek.

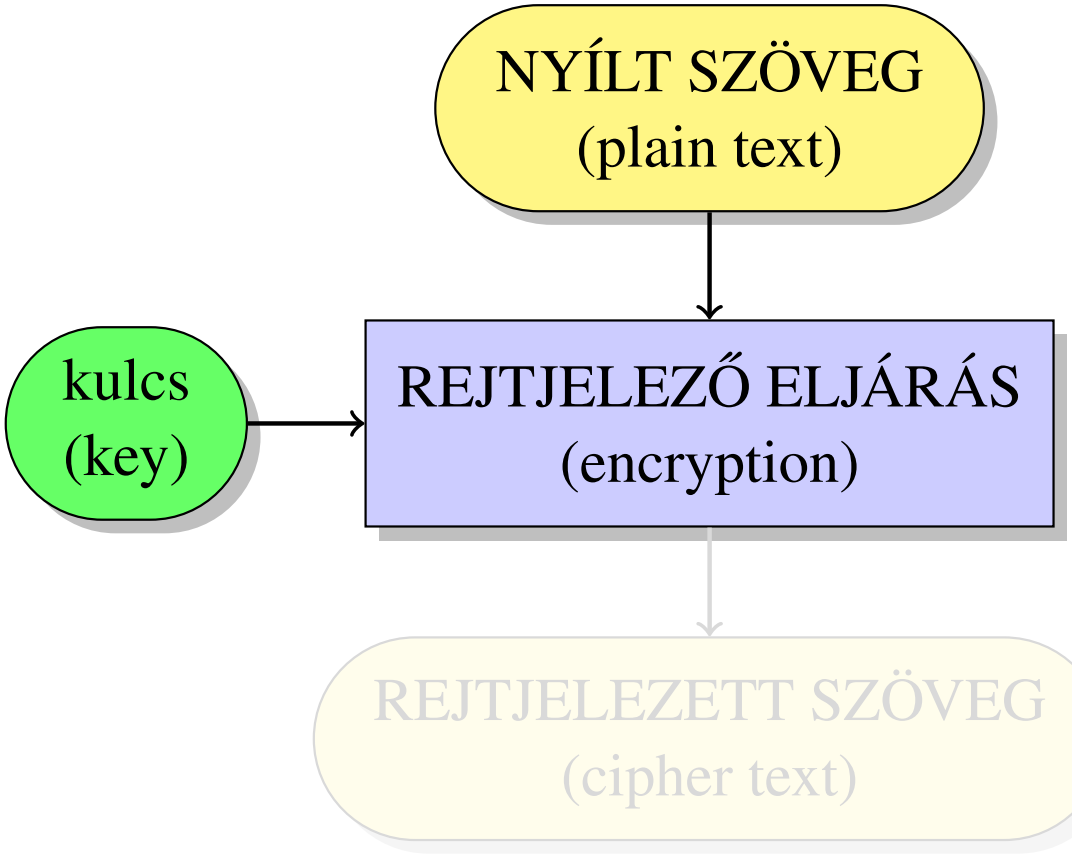
A kriptorendszer alapfogalmai: Rejtjelezés



A kriptorendszer kulcsterének fogalma

A kriptorendszer olyan kulcsainak halmaza, amik lényegesen különböző rejtjelezett szövegeket eredményeznek.

A kriptorendszer alapfogalmai: Rejtjelezés



REALENIGMASTORY

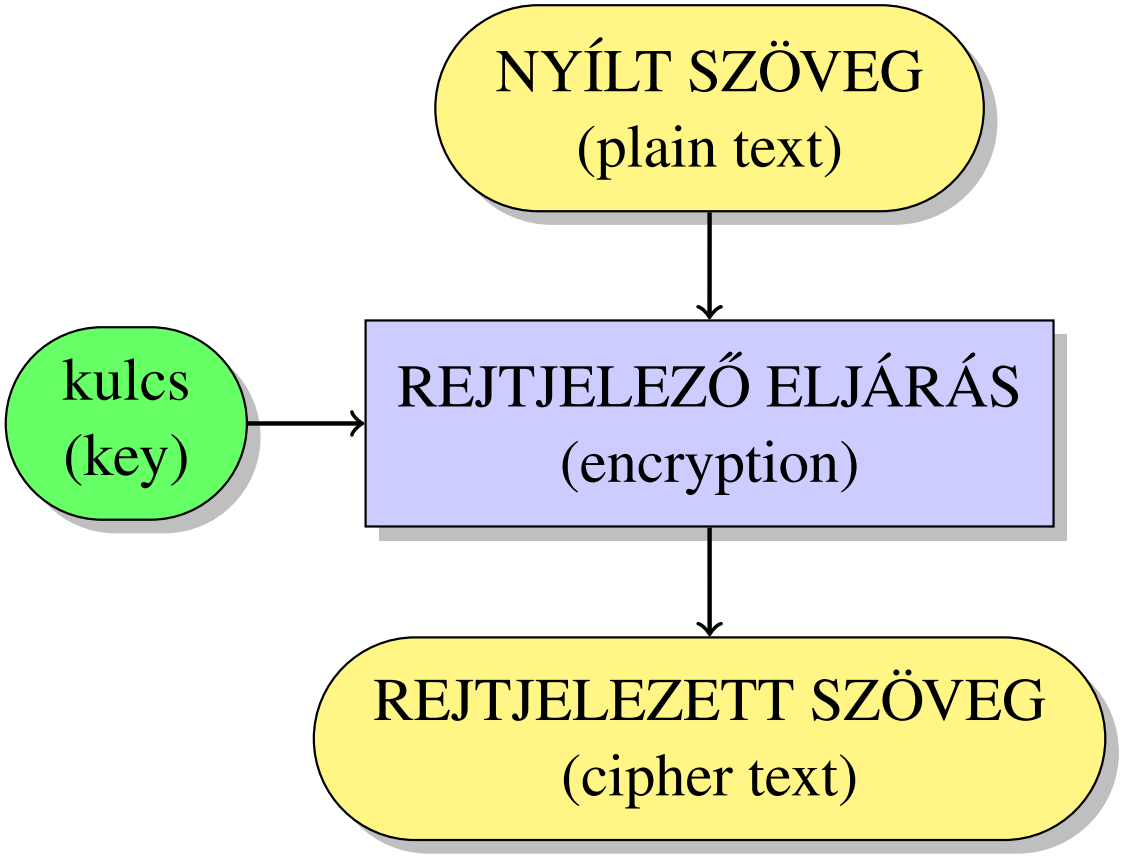
Ceasar-rejtjel
kulcs=7

YLHSLUPNTHZAVYF

A kriptorendszer kulcsterének fogalma

A kriptorendszer olyan kulcsainak halmaza, amik lényegesen különböző rejtjelezett szövegeket eredményeznek.

A kriptorendszer alapfogalmai: Rejtjelezés



REALENIGMASTORY

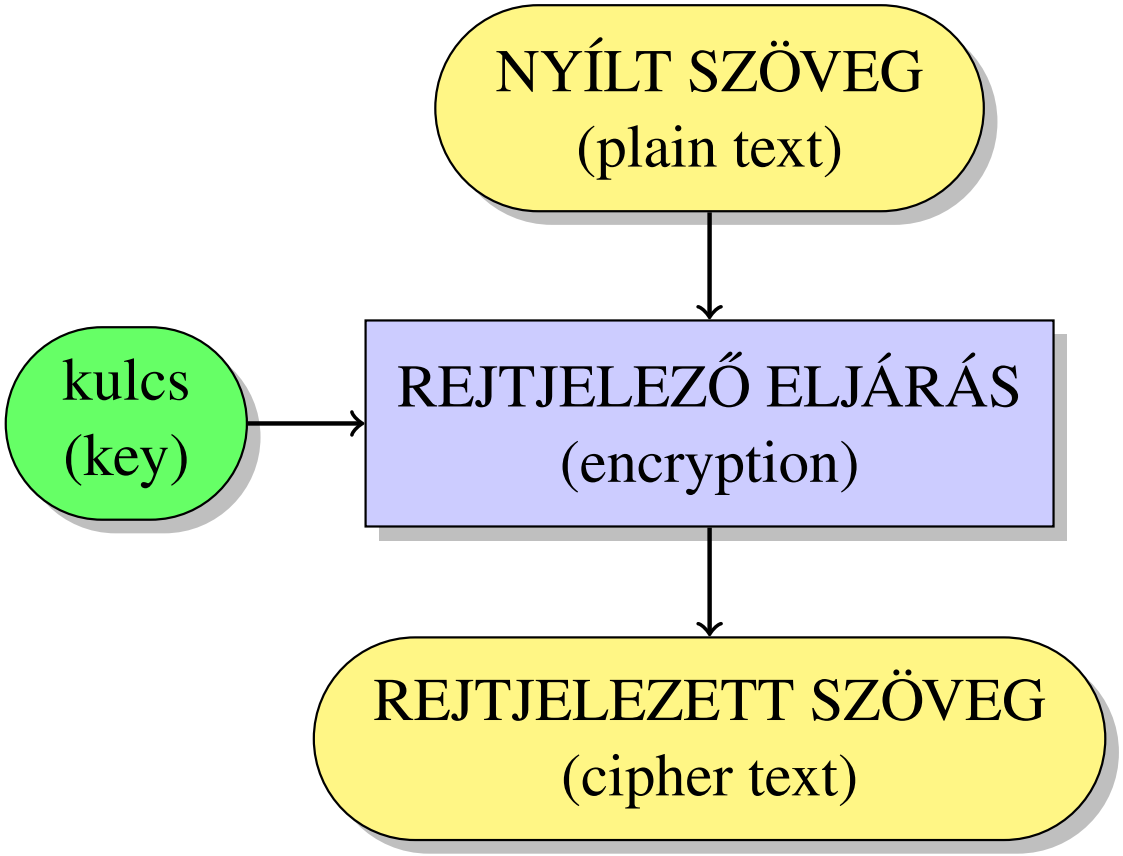
Ceasar-rejtjel
kulcs=7

YLHSLUPNTHZAVYF

A kriptorendszer kulcsterének fogalma

A kriptorendszer olyan kulcsainak halmaza, amik lényegesen különböző rejtjelezett szövegeket eredményeznek.

A kriptorendszer alapfogalmai: Rejtjelezés



REALENIGMASTORY

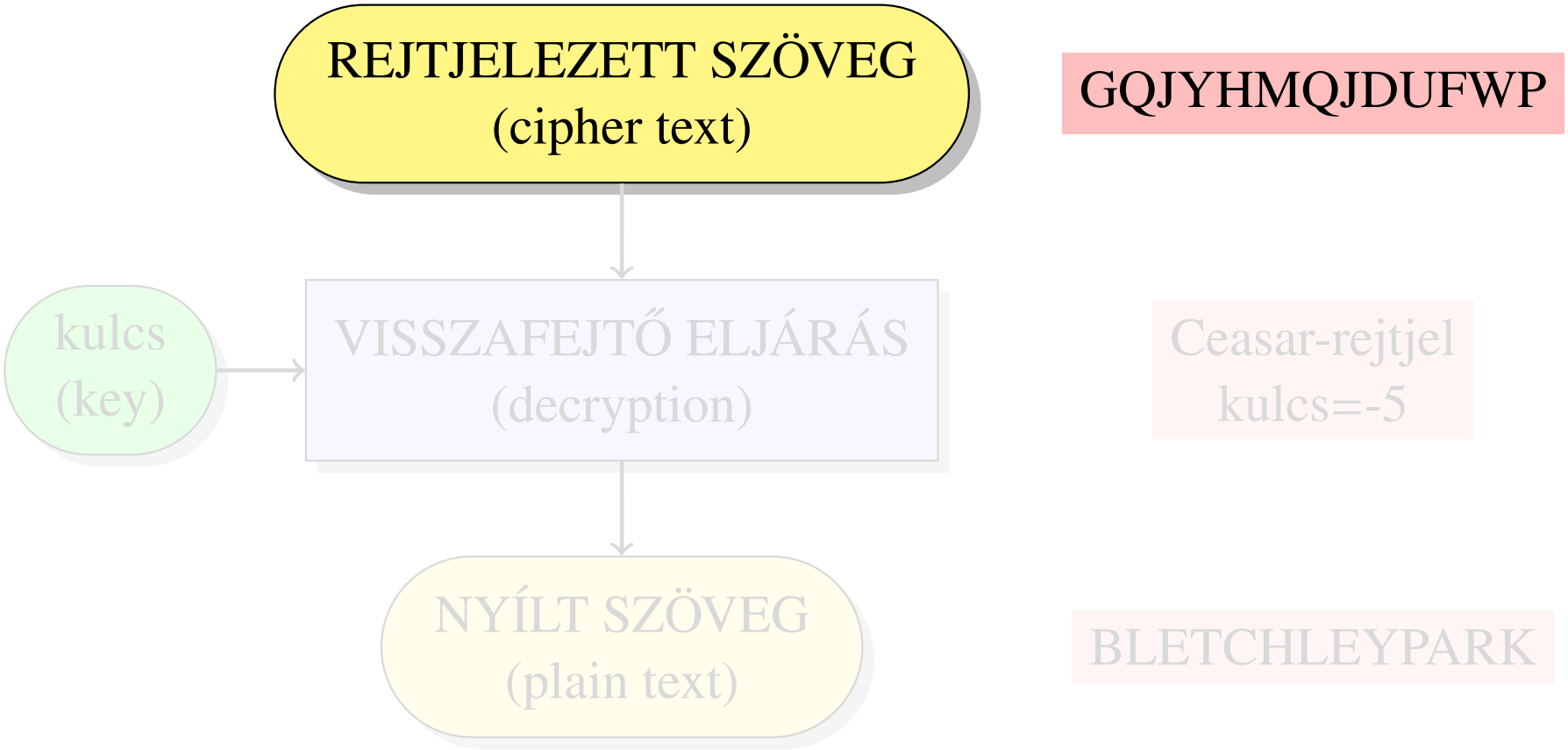
Ceasar-rejtjel
kulcs=7

YLHSLUPNTHZAVYF

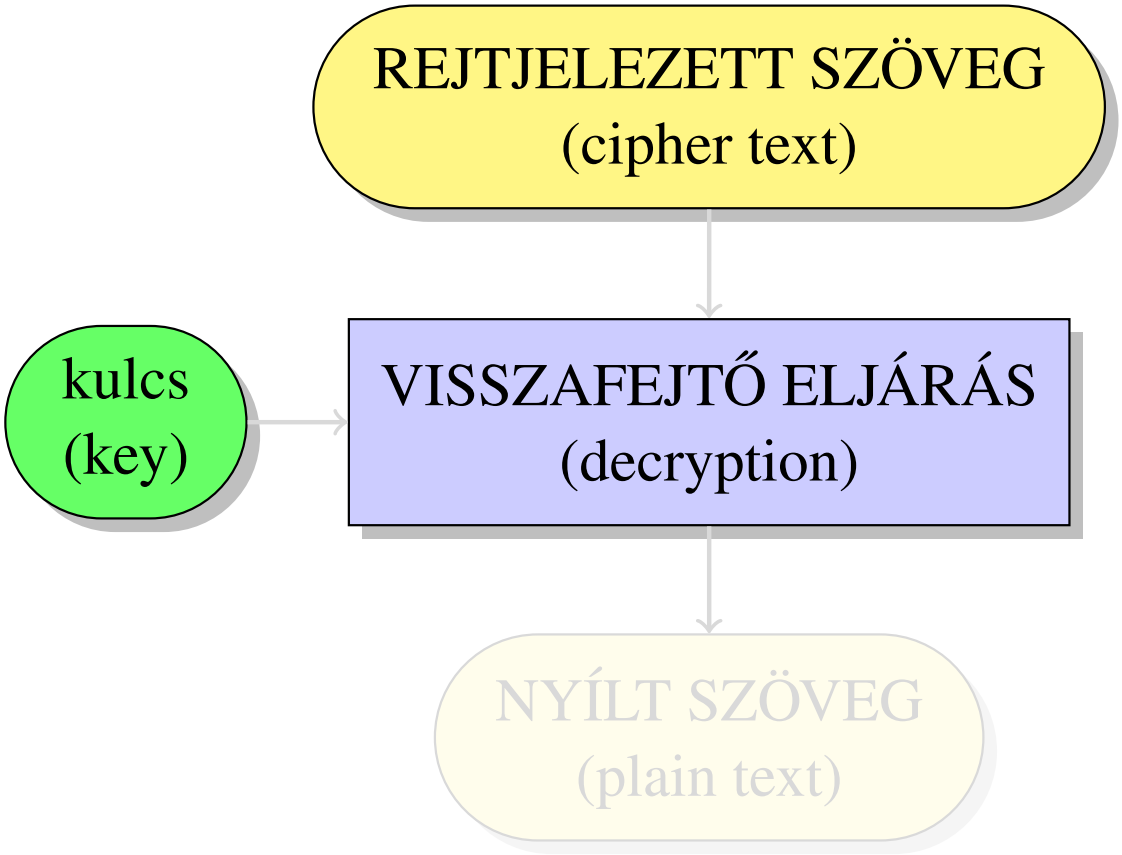
A kriptorendszer kulcsterének fogalma

A kriptorendszer olyan kulcsainak halmaza, amik lényegesen különböző rejtjelezett szövegeket eredményeznek.

A kriptorendszer alapfogalmai: Visszafejtés



A kriptorendszer alapfogalmai: Visszafejtés

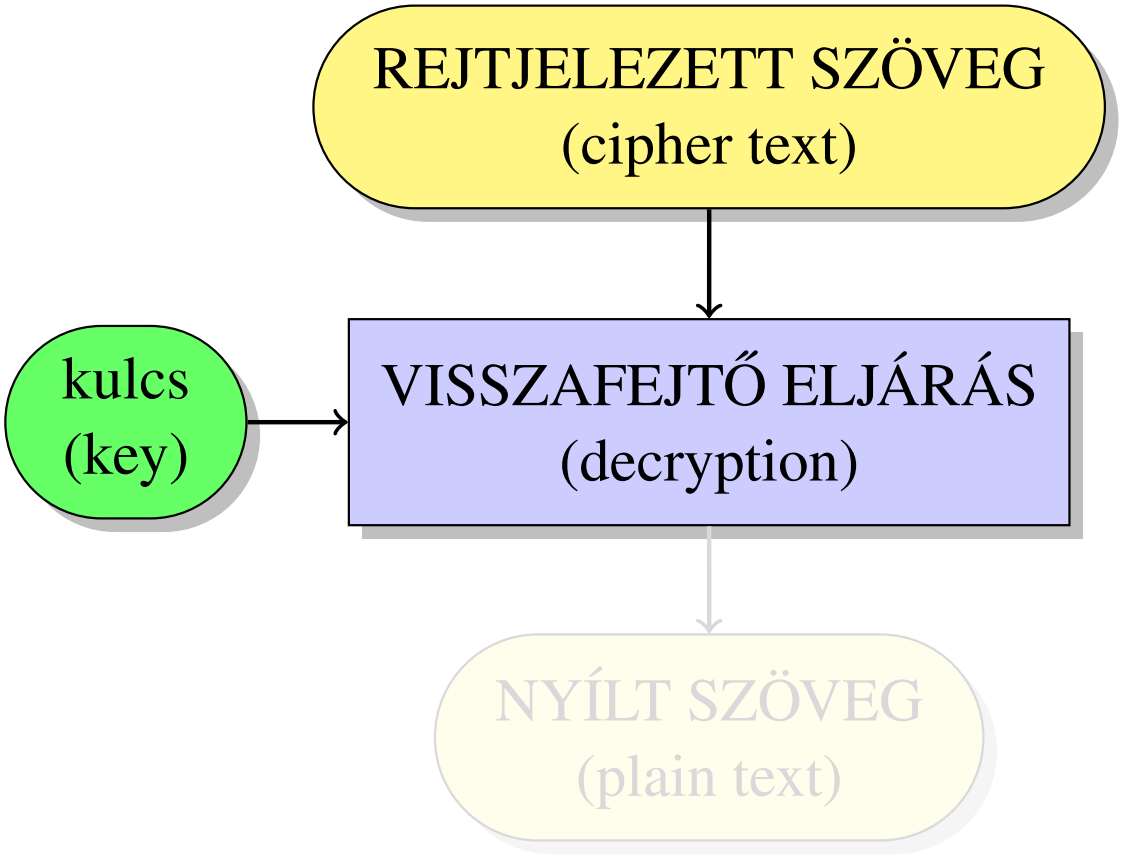


GQJYHMQJDUFWP

Ceasar-rejtjel
kulcs=-5

BLETCHLEYPARK

A kriptorendszer alapfogalmai: Visszafejtés

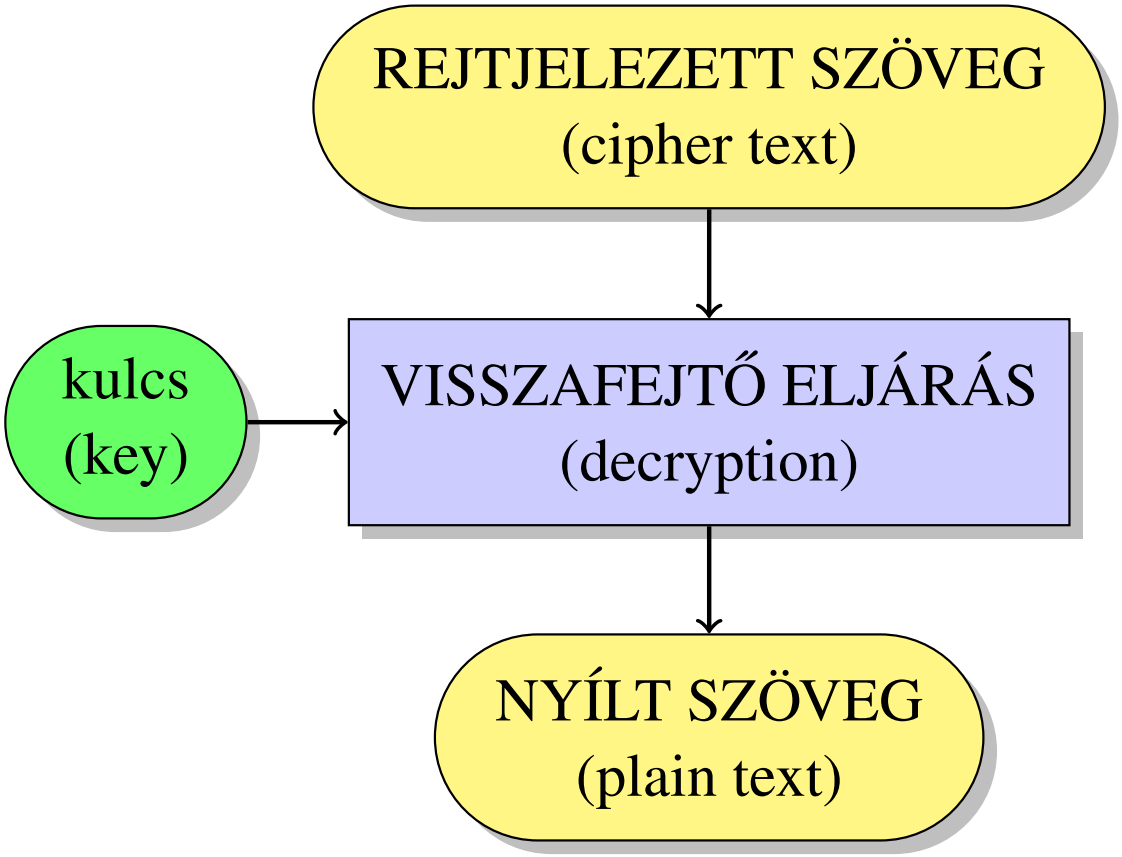


GQJYHMQJDUFWP

Ceasar-rejtjel
kulcs=-5

BLETCHLEYPARK

A kriptorendszer alapfogalmai: Visszafejtés



GQJYHMQJDUFWP

Ceasar-rejtjel
kulcs=-5

BLETCHLEYPARK

A Ceasar-féle titkosírás rendszer

- Tegyük fel, hogy a 26 betűs angol ábécét használjuk.
- Kulcsként válasszunk egy számot 1 és 25 között.
- Ebben a példában legyen a kulcs a 7-es.
- A nyílt szöveg minden betűjét helyettesítsük azzal a betűvel, ami az ábécében 7-el utána van: $A \mapsto H$, $B \mapsto I$, stb.

1	2	3	4	5	6	7	8	9	10	$\cdots$	24	25	26
A	B	C	D	E	F	G	H	I	J	$\cdots$	X	Y	Z
H	I	J	K	L	M	N	O	P	Q	$\cdots$	E	F	G

Sebezhetőségek

- A kulcstér csak 25 elemű.
- Ha ismerjük egyetlen betű megfejtését, akkor ismerjük a kulcsot.

A Ceasar-féle titkosírás rendszer

- Tegyük fel, hogy a 26 betűs angol ábécét használjuk.
- Kulcsként válasszunk egy számot 1 és 25 között.
- Ebben a példában legyen a kulcs a 7-es.
- A nyílt szöveg minden betűjét helyettesítsük azzal a betűvel, ami az ábécében 7-el utána van: $A \mapsto H$, $B \mapsto I$, stb.

1	2	3	4	5	6	7	8	9	10	$\cdots$	24	25	26
A	B	C	D	E	F	G	H	I	J	$\cdots$	X	Y	Z
H	I	J	K	L	M	N	O	P	Q	$\cdots$	E	F	G

Sebezhetőségek

- A kulcstér csak 25 elemű.
- Ha ismerjük egyetlen betű megfejtését, akkor ismerjük a kulcsot.

A Ceasar-rendszer „javításai”

A klasszikus monoalfabetikus rendszer

- Az ábécé ciklikus eltolásai helyett a betűk tetszőleges összecserélése.
- A kapott kulcstér mérete angol ábécé esetén

$$26! = 403\,291\,461\,126\,605\,635\,584\,000\,000 \approx 4 \cdot 10^{26}$$

- A ma legjobb szuperszámítógép kb. 370 év alatt végez ennyi műveletet.
- **Sebezhetőség:** A természetes nyelvekben a betűk a gyakoriságuk alapján beazonosíthatók.

Vigenère-rejtjel: A klasszikus polialfabetikus rendszer

- A Ceasar-kulcs értéke függ a betű helyétől.
- **Sebezhetőség:** Fejlett *statisztikai módszerek*, illetve *ismert szövegrészek*.

Ajánlott irodalom

- Edgar Allan Poe: Az aranybogár (1843)
- Verne Gyula: Nyolcszáz mérföld az Amazonason (1881)

A Kerckhoff-féle alapelvek és a „katasztrófa-forgatókönyv”

Auguste Kerckhoffs, *La Cryptographie Militaire*, 1883

1. A rendszernek gyakorlatilag, sőt lehetőleg matematikailag is **visszafejthetetlennek** kell lennie. A **rendszer maga nem lehet titkos,** nem jelenthet problémát, ha azt ismeri az ellenség.
2. A **kulcsnak rövidnek és könnyen továbbíthatónak** kell lenni, írott jegyzetek használata nélkül is.
3. A rendszer legyen használható a **Morse-távírós** kommunikációban.
4. A rendszernek **hordozhatónak** kell lennie, egy személy is tudja üzemelni.

A modern „katasztrófa-forgatókönyv” feltételezései

1. Az ellenfél teljesen ismeri a kriptorendszerünket.
2. Az ellenfél el tudja olvasni az összes rejtjelezett szövegünket.
3. Az ellenfél ismeri jelentős mennyiségű rejtjelezett szövegünkhöz tartozó nyílt szövegünket.

Szimmetrikus kulcsú kriptográfia

SZIMMETRIA

Ugyanaz a kulcs szolgál rejtjelezésre és visszafejtésre.

- 1976: **DES** (Data Encryption Standard)
- 2001: **AES** (Advanced Encryption Standard)
- AES demó: `http://www.formaestudio.com/rijndaelinspector/`
- Előny: Gyorsaság, kicsi memóriaigény, matematikai módszerekkel jól ellenőrizhető biztonság („keverési” tulajdonságok)
- Hátrány: **A kommunikáló feleknek előre meg kell állapodni a közös titkos kulcsban** (*key management*)

Nyilvános (aszimmetrikus) kulcsú kriptográfia

ASSZIMETRIA

Két különböző kulcsot használunk a rejtjelezésre illetve a visszafejtésre: egy **nyilvános** és egy **privát** kulcsot

- Alice nyílt szövege Bobnak + Bob nyilvános kulcsa = **rejtjelezett szöveg Bobnak**
- rejtjelezett szöveg Bobnak + Bob privát kulcsa = **Alice nyílt szövege Bobnak**
- A nyilvános kulcsból a privát kulcs kiszámítása **elméletben lehetséges,** de a **gyakorlatban nem**
- 1978: **RSA** (Ronald Rivest, Adi Shamir és Len Adleman)
- Az RSA biztonsága azon múlik, hogy tudunk-e több száz számjegyből álló számokat prímtényezőkre bontani.

A matematikai kriptográfia alapfogalmai

Az üzenet:

- 0/1 sorozat
- rögzített hosszúságú 0/1 sorozat
- nemnegatív egész szám $0, 1, \ldots, 2^n - 1$
- nemnegatív egész szám $0, 1, \ldots, N - 1$ tetszőleges számrendszerben

Ugyanez mondható el a titkosító kulcsról.

Jelölés

- m nyílt üzenet *(plain text message)*
- k titkosító kulcs *(encryption key)*
- c titkosított üzenet *(ciphertext)*
- $m, k, c \in \{0, 1, \ldots, N - 1\}$, ahol N nagy pozitív egész

Titkosítás *(encryption)* és visszafejtés *(decryption)*

Emlékeztető:

Az ENC titkosító eljárás *(encryption)* egy kulcsból és egy nyílt üzenetből készít egy titkosított üzenetet:

$$(k, m) \overset{\text{ENC}}{\longmapsto} c$$

A DEC visszafejtő eljárás *(decryption)* egy kulcsból és egy titkosított üzenetből készíti el a nyílt üzenetet:

$$(k, c) \overset{\text{DEC}}{\longmapsto} m$$

Matematikai jelölés:

- $c = \text{ENC}(k, m) = \text{ENC}_k(m)$
- $m = \text{DEC}(k, c) = \text{DEC}_k(c)$ függvények
- Művelettel: $c = k * m$.

Kulcs kereső támadás *(key recovery attack)*

c, m ismert számok, $*$ ismert művelet, határozzuk meg x-et:

$$c = x * m$$

- A kulcstér végessége miatt **próbálgatással** megoldható
- Az eljárás biztonságosságának mértékegysége: **DOLLARDAY**
- Szivárgás (flaw): 1 bit információ a kulcsról megfelezi a kulcsteret
- Gyakori kulcscsere
- A kulcsot random választjuk a kulcstérből

Üzenet kereső támadás *(message recovery attack)*

c ismert szám, $*$ ismert művelet, határozzuk meg x, y-t:

$$c = x * y$$

- Nem reménytelen, különösen ha k-nak vagy m-nek **erős a struktúrája**
- Pl. $0 \leq y \leq 999$ és $x * y = 1000x + y$.
- Másik példa a monoalfabetikus rendszer feltörése *gyakoriság-analízissel.*

A $*$ művelet felépítése

- $m, k, c \in \{0, 1, \ldots, N-1\}$
- modulo n jelentése: egy egész szám n-el vett **osztási maradéka**
- **Alapműveletek** modulo N:

$$x \oplus y = x + y \pmod{N}$$
$$x \ominus y = x - y \pmod{N}$$
$$x \otimes y = x \cdot y \pmod{N}$$
$$x \oslash y = x/y \pmod{N}$$

- Melyik a kakukktojás??
- Algebrai test: **halmaz** ÉS **négy művelet** ÉS **számolási szabályok**
- Pl. $\{0, 1\}$, műveletek modulo 2: $1 \oplus 1 = 0$.

RSA

- **RSA** titkosítás:

$$c = \mathrm{ENC}_k(m) = m^k \pmod{N}.$$

- **RSA** visszafejtés: k-dik gyök modulo N.

RSA biztonsága

- A feladat: adott x, k, N, keressük $\sqrt[k]{x} \pmod{N}$.
- **Könnyű,** ha ismert N prímtényezős felbontása.
- **Nehéz** egyébként.
- **Kvantum-könnyű!** *(Kvantum-támadás)*

Az USA poszt-kvantum felhívása

`https://csrc.nist.gov/Projects/post-quantum-cryptography`

NIST Post-Quantum Challange 2017

- NIST = National Institute of Standard and Technology
- Az USA szabványügyi hivatala
- Nemzetközi felhívás kvantum-biztos kriptográfiai eljárások javaslatára
- Első három kör: 2017, 2019, 2021.
- Eredményhirdetés: 2022.

Valószínűleg el fogják fogadni:

- Kód alapú: **klasszikus McEliece**
- Rács alapú: **SABER, CRYSTAL**
- Izogénia alapú: **SIKE**

Az USA poszt-kvantum felhívása

`https://csrc.nist.gov/Projects/post-quantum-cryptography`

NIST Post-Quantum Challange 2017

- NIST = National Institute of Standard and Technology
- Az USA szabványügyi hivatala
- Nemzetközi felhívás kvantum-biztos kriptográfiai eljárások javaslatára
- Első három kör: 2017, 2019, 2021.
- Eredményhirdetés: 2022.

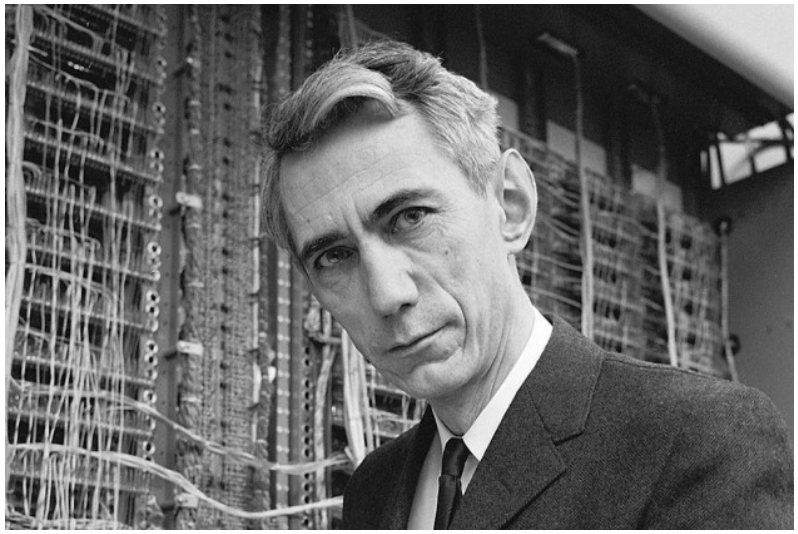
Valószínűleg el fogják fogadni:

- Kód alapú: **klasszikus McEliece**
- Rács alapú: **SABER, CRYSTAL**
- Izogénia alapú: **SIKE**

Tagolás

A hibajavító kódok elméletének megalapozása

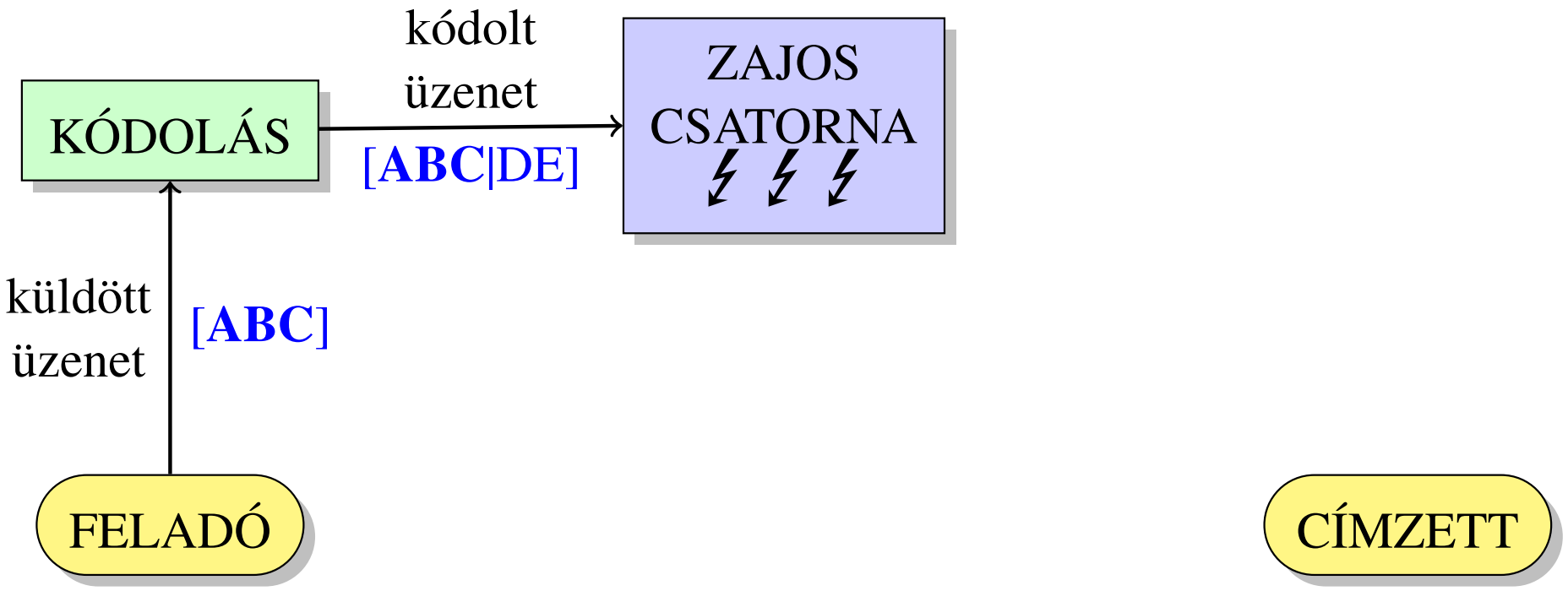
Richard Hamming
(1915-1998)
amerikai matematikus



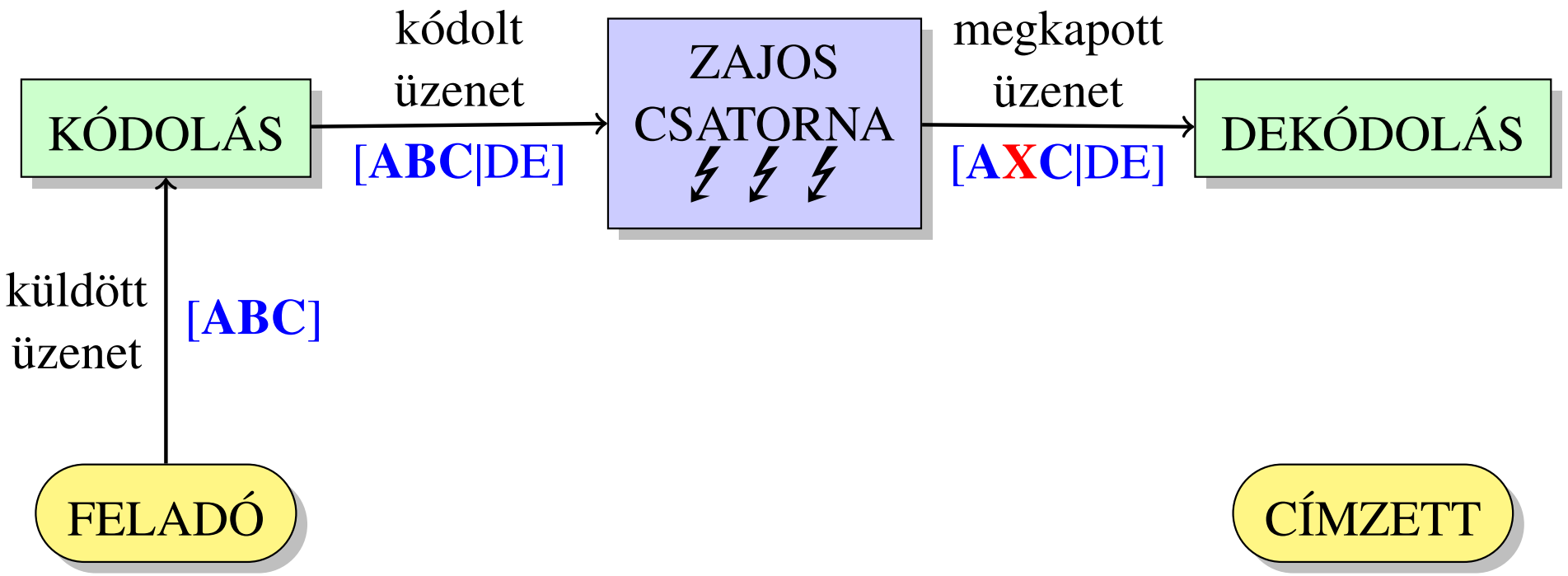
Claude Shannon
(1916-2001)
amerikai matematikus

Hibajavítás zajos kommunikációs csatornán

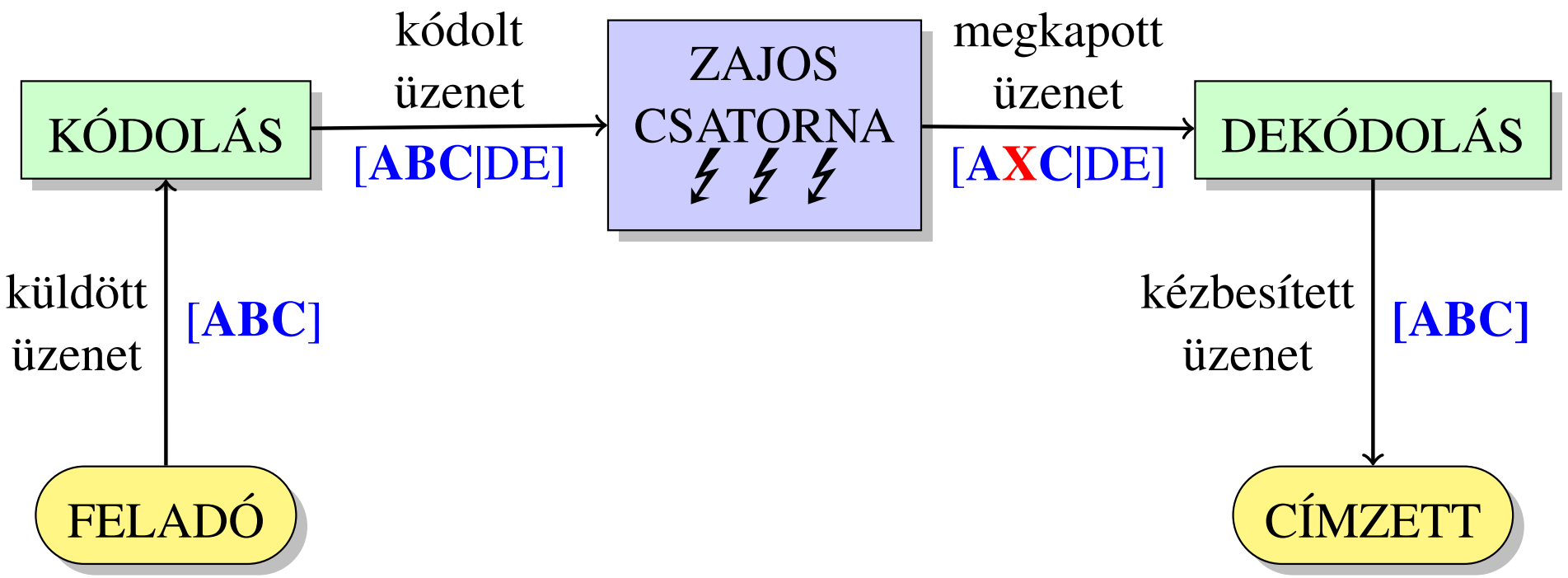
Hibajavítás zajos kommunikációs csatornán



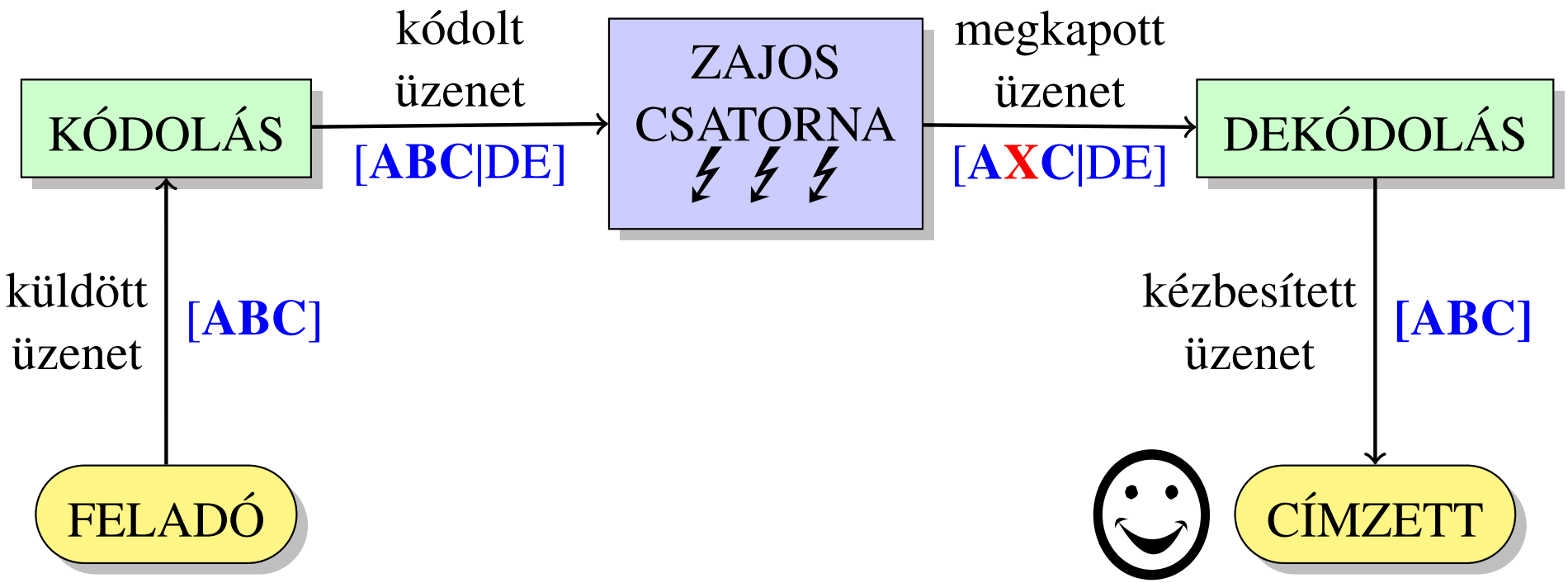
Hibajavítás zajos kommunikációs csatornán



Hibajavítás zajos kommunikációs csatornán



Hibajavítás zajos kommunikációs csatornán



Példa: Hibajavítás a QR-kódban

- A QR-kódokban az $\mathbb{F}_{256}$ véges test felett értelmezett Reed-Solomon kódokat használják.

Példa: Hibajavítás a QR-kódban

- A QR-kódokban az $\mathbb{F}_{256}$ véges test felett értelmezett Reed-Solomon kódokat használják.

Lineáris kódok fő paraméterei

A továbbiakban $\mathbb{F}_q$ egy $q = p^m$ rendű véges testet jelöl, ahol p prím.

Definíció: n hosszúságú lineáris kód

- A Q **ábécé** az $\mathbb{F}_q$ véges test.
- $C \leq \mathbb{F}_q^n$ egy lineáris altér.

A főbb paraméterek:

- **Hosszúság** n
- **Dimenzió** k
- **Információs ráta** $R = k/n$
- **Minimum távolság** d

Singleton-korlát

$k + d \leq n + 1$.

A Reed–Solomon-kódok

- Legyen $0 \leq k \leq n \leq q$. Legyenek $\alpha_1, \ldots, \alpha_n$ az $\mathbb{F}_q$ különböző elemei.
- $\mathbf{RS}_k = \{(f(\alpha_1), \ldots, f(\alpha_n) \mid f \in \mathbb{F}_q[X], \deg(f) < k\}$

Tétel

- A Reed–Solomon-kód minimum távolsága $d = n + 1 - k$.
- A **Peterson-algoritmus** ki tud javítani $\lfloor \frac{n-k}{2} \rfloor = \lfloor \frac{d-1}{2} \rfloor$ hibát.

Irving S. Reed (1923-2012)
Gustave Solomon (1930-1996)

W. Wesley Peterson
(1924-2009)

Random kódok

Nincs annál *könnyebb*, mint jó paraméterekkel rendelkező bináris lineáris kódokat készíteni:

Zajos csatornakódolási tétel (Shannon 1948)

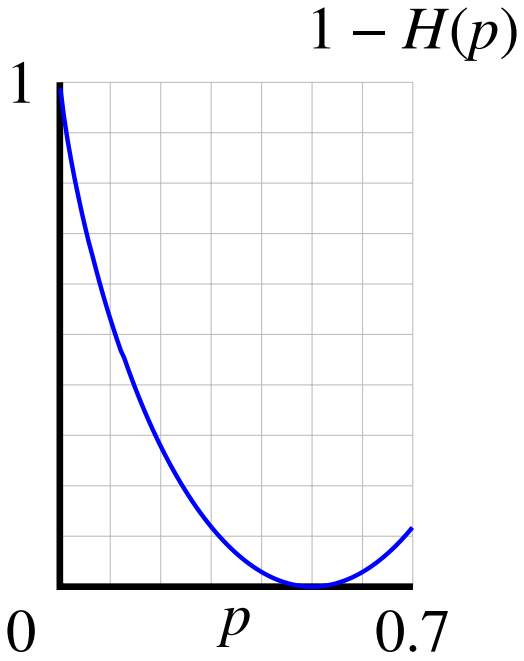
Értelmezzük a bináris entrópiafüggvényt:

$$H(p) = -p \log_2 p - (1-p) \log_2 (1-p).$$

Rögzítsük az $0 < R < 1$ rátát. Ekkor:

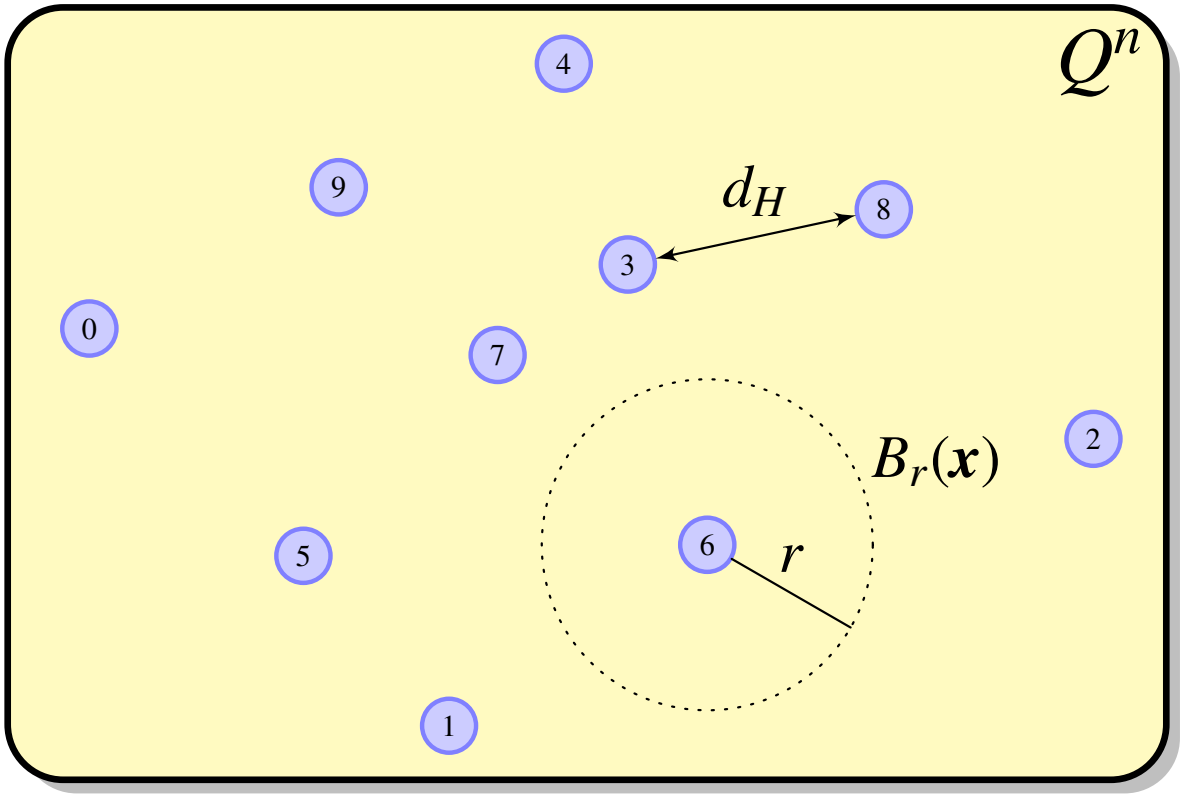
- kellően nagy n esetén
- az n hosszúságú és R rátájú
- **„random”** bináris lineáris kód
- minimum távolsága legalább

$$n \cdot H^{-1}(1-R).$$



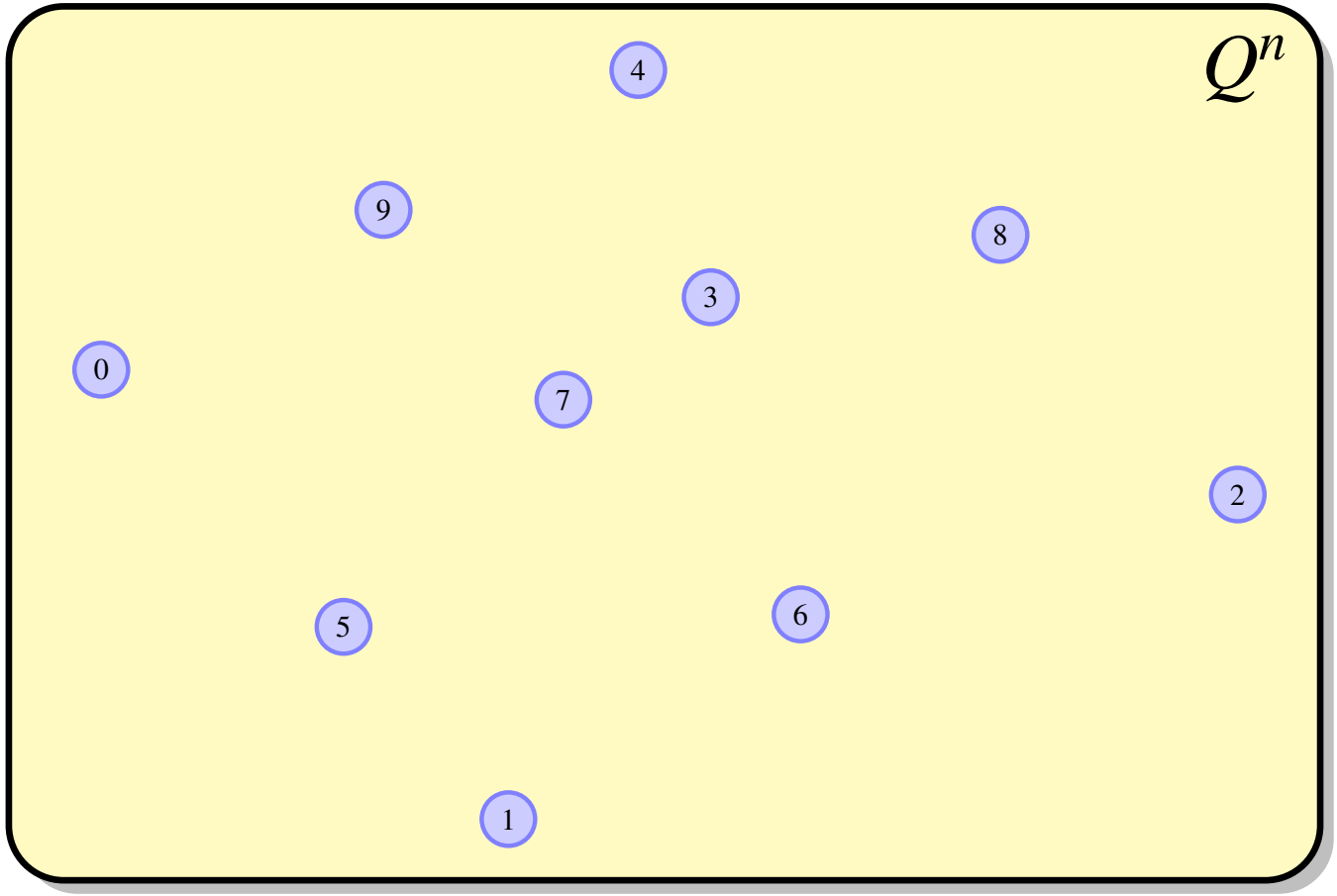
A hibajavító kódolás

- **Véges ábécé** $Q = \{0, 1, \ldots\}$; általában bináris: $Q = \{0, 1\}$
- Egy n **hosszúságú kód** a $C \subseteq Q^n$ halmaz részhalmaza



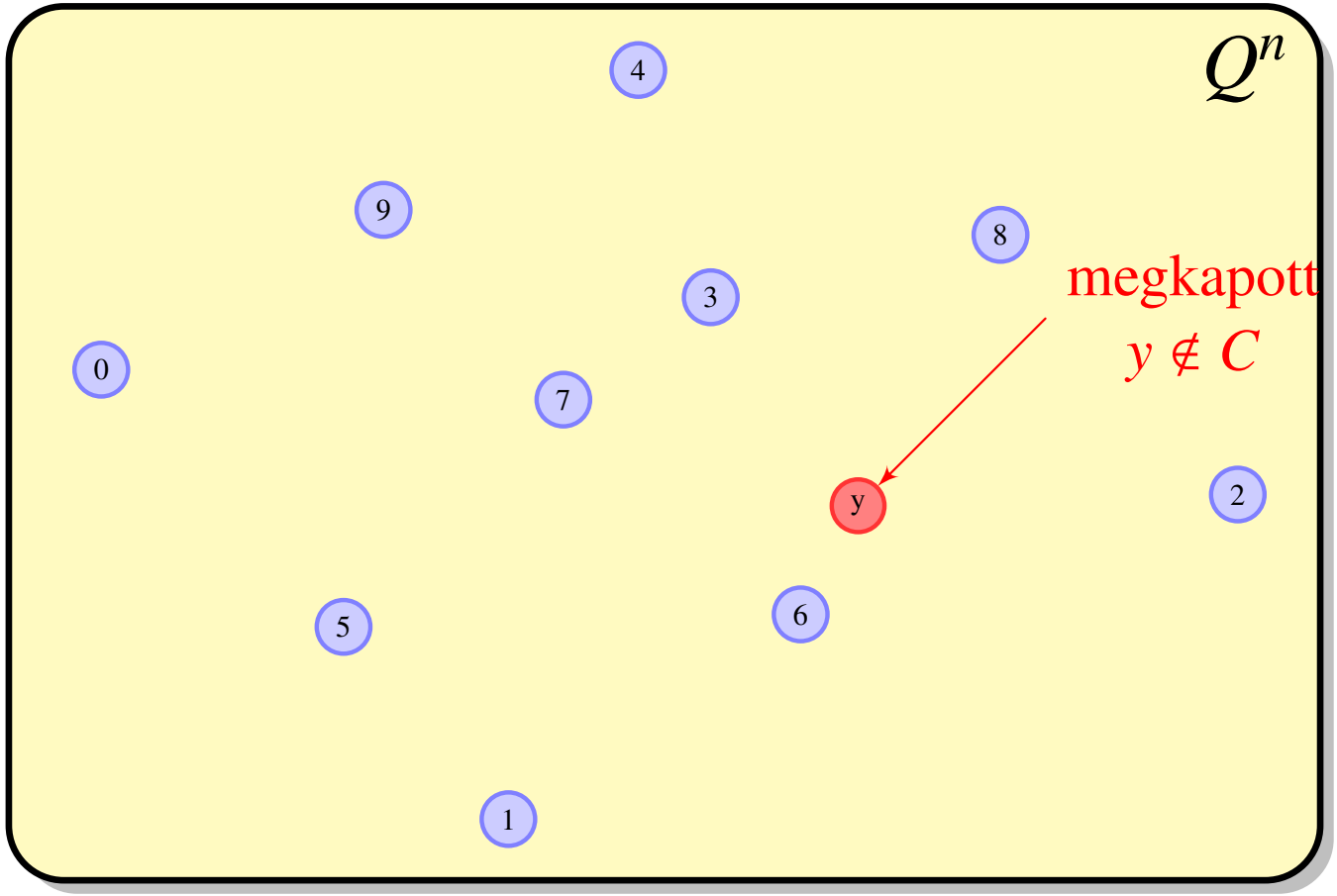
- Példa: **3-szoros ismétlő kód:** $0 \mapsto 0|00,\ 1 \mapsto 1|11.$
- $Q = \{0, 1\}$, $C = \{000, 111\} \subseteq \{0, 1\}^3$.

Legközelebbi szomszéd *(maximum likelihood)* dekódolás



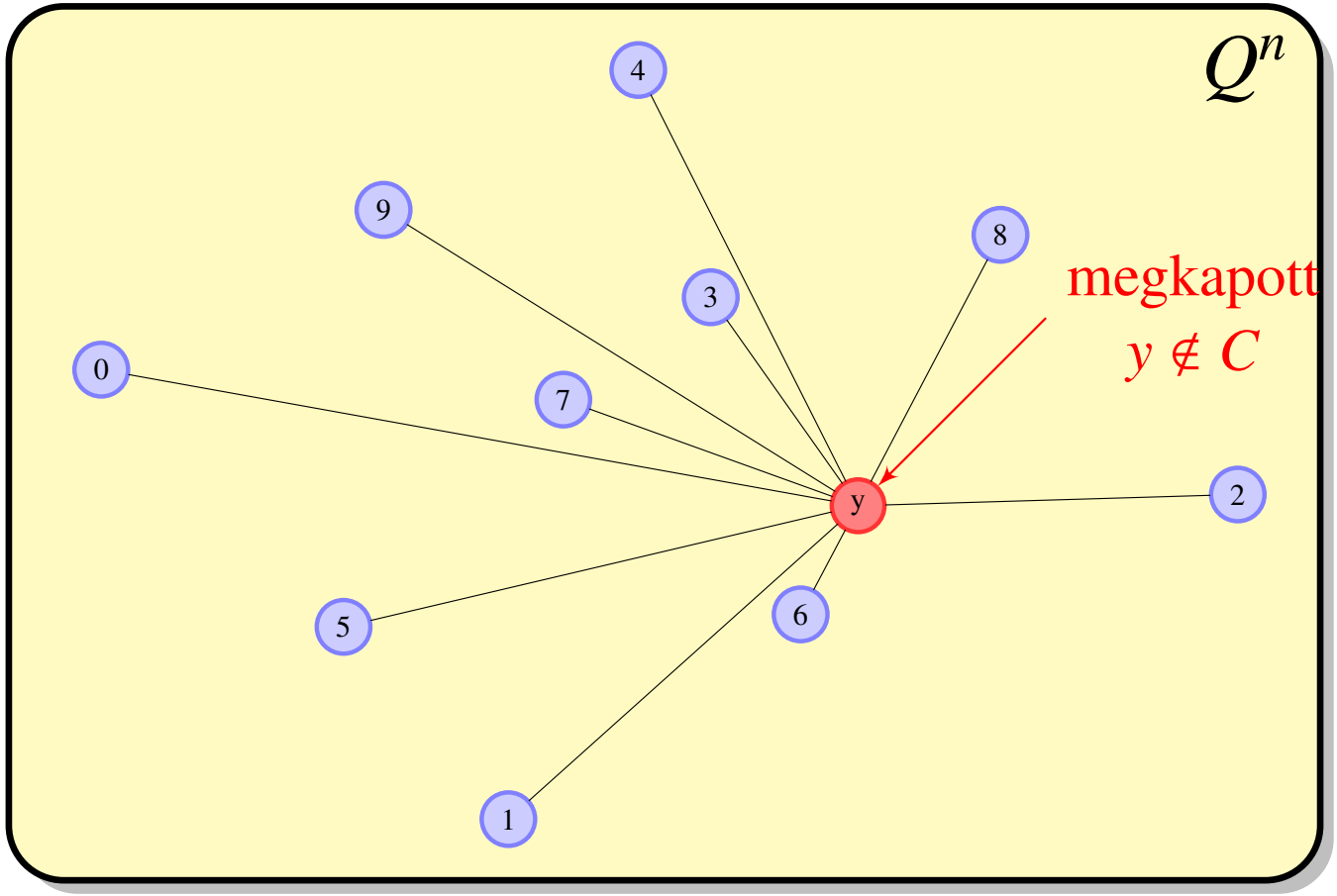
- $0|00, 1|00, 0|10, 0|01 \mapsto 0|00 \mapsto \mathbf{0}$
  $1|10, 1|01, 0|11, 1|11 \mapsto 1|11 \mapsto \mathbf{1}$.
- Prob(k hiba) > Prob(k-nál több hiba).

Legközelebbi szomszéd *(maximum likelihood)* dekódolás



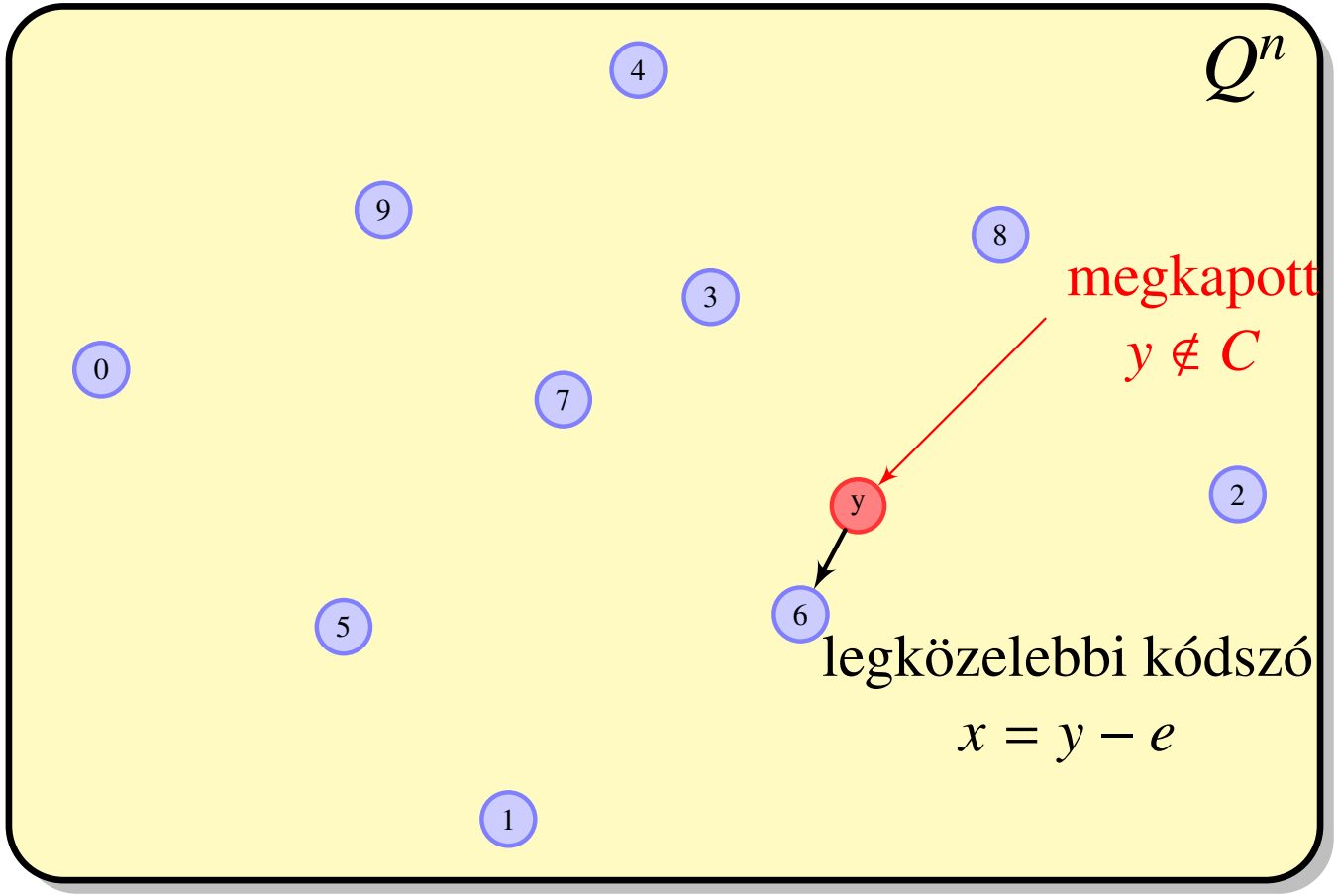
- $0|00, 1|00, 0|10, 0|01 \mapsto 0|00 \mapsto \mathbf{0}$
  $1|10, 1|01, 0|11, 1|11 \mapsto 1|11 \mapsto \mathbf{1}$.
- Prob(k hiba) > Prob(k-nál több hiba).

Legközelebbi szomszéd *(maximum likelihood)* dekódolás



- $0|00, 1|00, 0|10, 0|01 \mapsto 0|00 \mapsto \mathbf{0}$
  $1|10, 1|01, 0|11, 1|11 \mapsto 1|11 \mapsto \mathbf{1}$.
- Prob(k hiba) > Prob(k-nál több hiba).

Legközelebbi szomszéd *(maximum likelihood)* dekódolás



- $0|00, 1|00, 0|10, 0|01 \mapsto 0|00 \mapsto \mathbf{0}$
  $1|10, 1|01, 0|11, 1|11 \mapsto 1|11 \mapsto \mathbf{1}$.
- Prob(k hiba) > Prob(k-nál több hiba).

A dekódolási feladat

A legközelebbi szomszéd dekódolási feladat

Adott: a $C \subseteq Q^n$ kód és az $\boldsymbol{y} \in Q^n$ vektor.

Keresünk: olyan $\boldsymbol{x} \in C$ kódszót, amire $d_H(\boldsymbol{x}, \boldsymbol{y})$ minimális.

A küszöbértékes dekódolási feladat

Adott: a $C \subseteq Q^n$ kód, az $\boldsymbol{y} \in Q^n$ vektor és a t pozitív egész.

Keresünk: olyan $\boldsymbol{x} \in C$ kódszót, amire

$$d_H(\boldsymbol{x}, \boldsymbol{y}) \leq t.$$

Ha nincs ilyen, akkor az eredmény legyen **„NINCS”.**

- A **küszöbértékes dekódolás** lényegesen könnyebb a legközelebbi szomszéd dekódolásnál, ha $t < d(C)/2$.

A dekódolási feladat lineáris kódokra

Legyen C **lineáris kód** a szokásos paraméterekkel: q, n, k, d.

- C megadható egy $\mathbb{F}_q$ feletti $k \times n$ **generátormátrixszal.**
- A generátormátrix nk elemet tartalmaz, tehát a feladat mérete bitben számolva

$$(nk + n) \log_2(q).$$

- Mivel $k \leq n$, a feladat méretének nagyságrendjét tekinthetjük n^2-nek.

Állítás

- A **kimerítéses keresési eljárás** mindkét dekódolási feladatra exponenciális bonyolultságú megoldási algoritmust ad az n paraméterben.
- Rögzített q mellett a Reed–Solomon-kódok dekódolása *könnyű.*
- A Petersen-algoritmus bonyolultsága n-ben polinomiális.

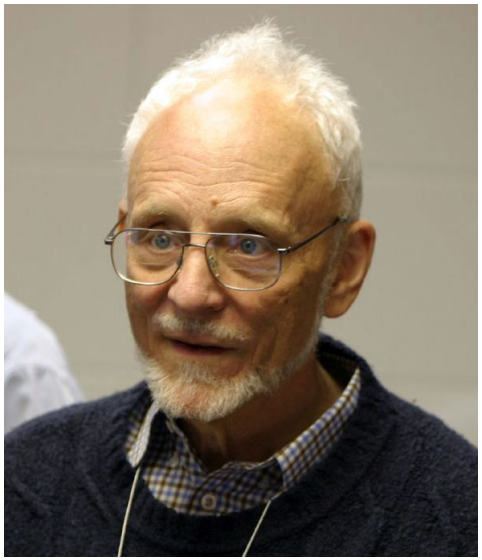
A dekódolás *nehéz* — még a bináris esetben is

Random bináris lineáris kódok *használhatatlanok* a gyakorlatban:

Tétel (Berlekamp, McEliece, van Tilborg, 1978)

Bináris lineáris kódok esetén a küszöbértékes dekódolási feladat ***nagyon nehéz*** (*"NP-teljes"*).

Robert McEliece
(1942-2019)



Elwyn Berlekamp
(1940-2019)

Henk van Tilborg
(1947-)

Az általánosított Reed–Solomon-kód

Definíció: Általánosított Reed–Solomon-kód

Legye q prímhatvány, $0 \leq k \leq n \leq q$. Legyenek $\alpha_1, \ldots, \alpha_n$ az $\mathbb{F}_q$ különböző elemei, $v_, \ldots, v_n$ az $\mathbb{F}_q$ nem nulla elemei.

$$\mathbf{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}) = \{(v_1 f(\alpha_1), \ldots, v_n f(\alpha_n)) \mid f \in \mathbb{F}_q[X],\ \deg(f) < k\}.$$

- A *Reed–Solomon* és az *általánosított Reed–Solomon-kódok* paraméterei megegyeznek.
- A Peterson-dekódolás az általános esetben is működik.

Lineáris kódok résztest részkódjai

Definíció: Résztest részkód

Legyen $q = p^m$ prímhatvány és $C \leq \mathbb{F}_q^n$ lineáris kód $\mathbb{F}_q$ felett. Legyen

$$C|_{\mathbb{F}_q} = C \cap \mathbb{F}_p^n.$$

Ekkor $C|_{\mathbb{F}_q} \leq \mathbb{F}_p^n$ *lineáris kód* $\mathbb{F}_p$ felett.

- A résztest részkód minimum távolsága legalább akkora, mint az eredeti kódé.
- Az eredeti kód **dekódolási eljárásai alkalmazhatók** a résztest részkód esetén is
- Küszöbértékes dekódolás esetén a küszöbérték megőrződik.
- A résztest részkód dimenziója

$$\dim(C|_{\mathbb{F}_q}) \geq n - m(n-k).$$

- Nagy m-re pontatlan a becslés, de **nehéz** sokkal jobbat mondani.

Lineáris kódok résztest részkódjai

Definíció: Résztest részkód

Legyen $q = p^m$ prímhatvány és $C \leq \mathbb{F}_q^n$ lineáris kód $\mathbb{F}_q$ felett. Legyen

$$C|_{\mathbb{F}_q} = C \cap \mathbb{F}_p^n.$$

Ekkor $C|_{\mathbb{F}_q} \leq \mathbb{F}_p^n$ *lineáris kód* $\mathbb{F}_p$ felett.

- A résztest részkód minimum távolsága legalább akkora, mint az eredeti kódé.
- Az eredeti kód **dekódolási eljárásai alkalmazhatók** a résztest részkód esetén is
- Küszöbértékes dekódolás esetén a küszöbérték megőrződik.
- A résztest részkód dimenziója

$$\dim(C|_{\mathbb{F}_q}) \geq n - m(n-k).$$

- Nagy m-re pontatlan a becslés, de **nehéz** sokkal jobbat mondani.

Reed–Solomon-alapú bináris kódok

Definíció: Alternáns kód

Az általánosított Reed–Solomon-kódok résztest részkódjait alternáns kódoknak nevezzük.

- Az alternáns kód konstrukcióval bináris lineáris kódok egy széles osztályát kapjuk.
- Ezek paraméterei nem különösebben jók.
- Nagy előnyük, hogy ismert hozzájuk polinomiális dekódolási eljárás.
- Tehát megfelelő t küszöbértékkel a dekódolásuk *könnyű*.

Fontos alternáns kód altípusok:

- BCH *(jól skálázható)*
- bináris Goppa *(a várt küszöbérték duplájáig dekódolható)*
- Srivastava

Reed–Solomon-alapú bináris kódok

Definíció: Alternáns kód

Az általánosított Reed–Solomon-kódok résztest részkódjait alternáns kódoknak nevezzük.

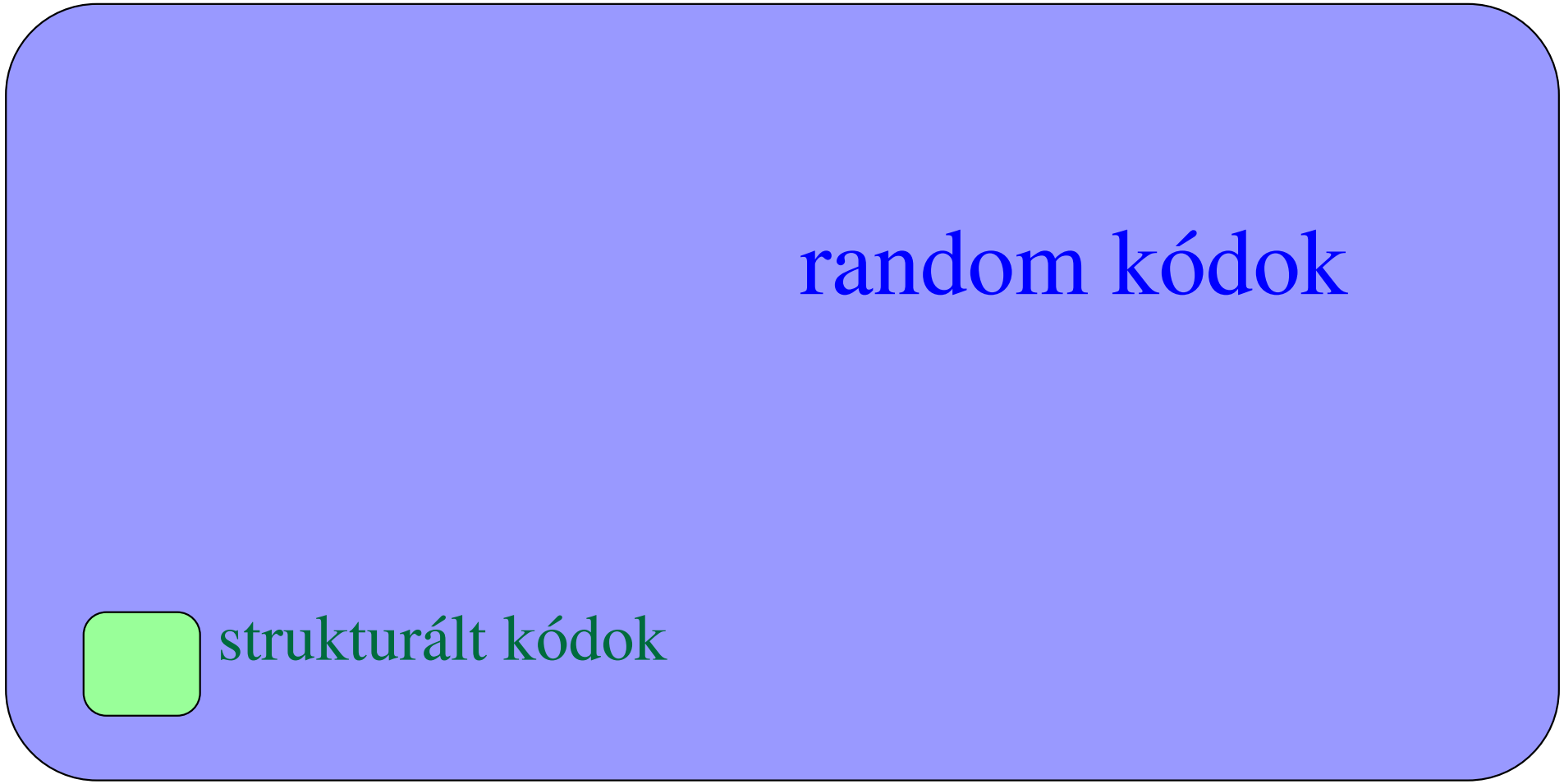
- Az alternáns kód konstrukcióval bináris lineáris kódok egy széles osztályát kapjuk.
- Ezek paraméterei nem különösebben jók.
- Nagy előnyük, hogy ismert hozzájuk polinomiális dekódolási eljárás.
- Tehát megfelelő t küszöbértékkel a dekódolásuk *könnyű*.

Fontos alternáns kód altípusok:

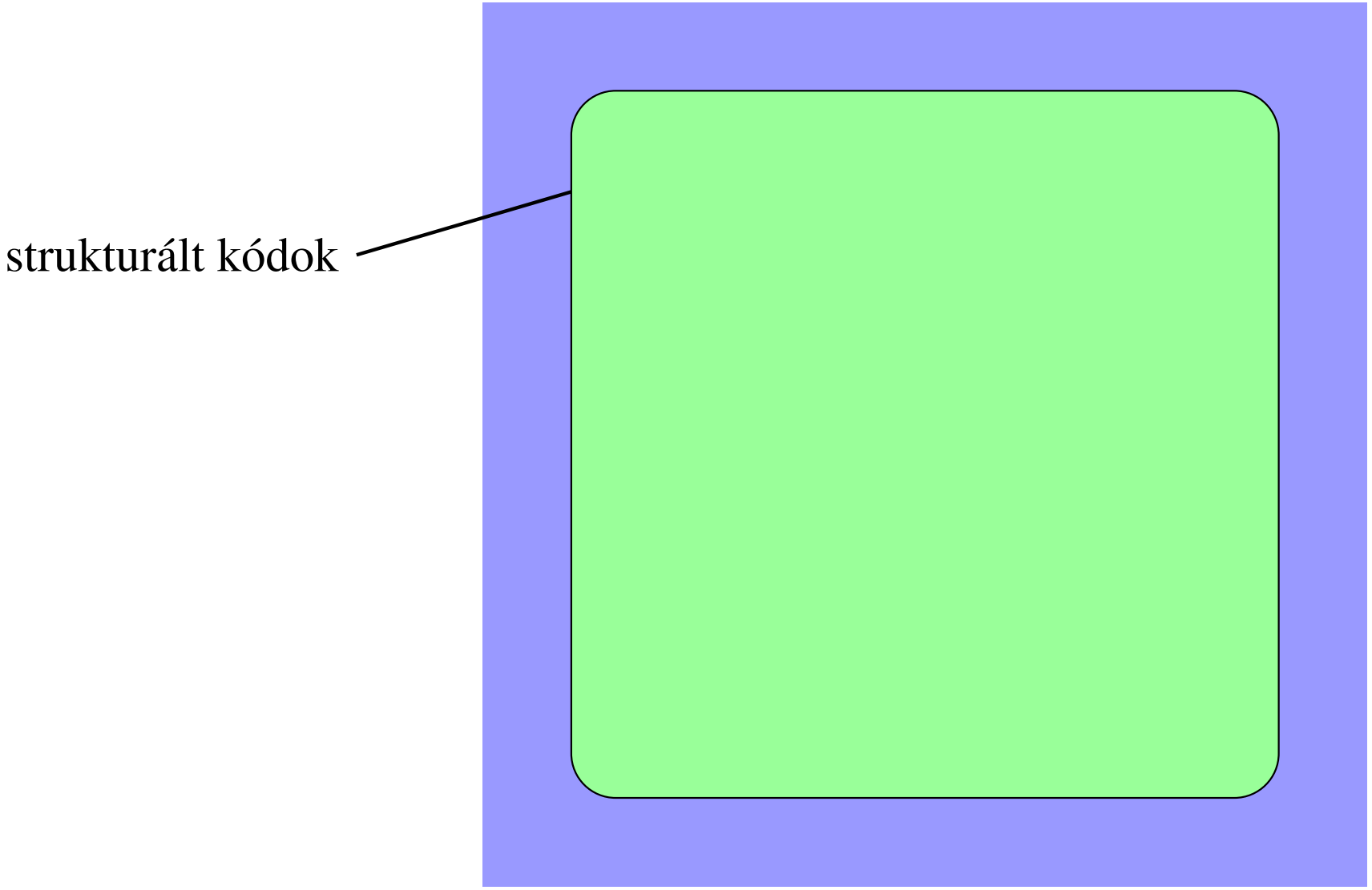
- BCH *(jól skálázható)*
- bináris Goppa *(a várt küszöbérték duplájáig dekódolható)*
- Srivastava

Az n, k, q paraméteres kódok tengere

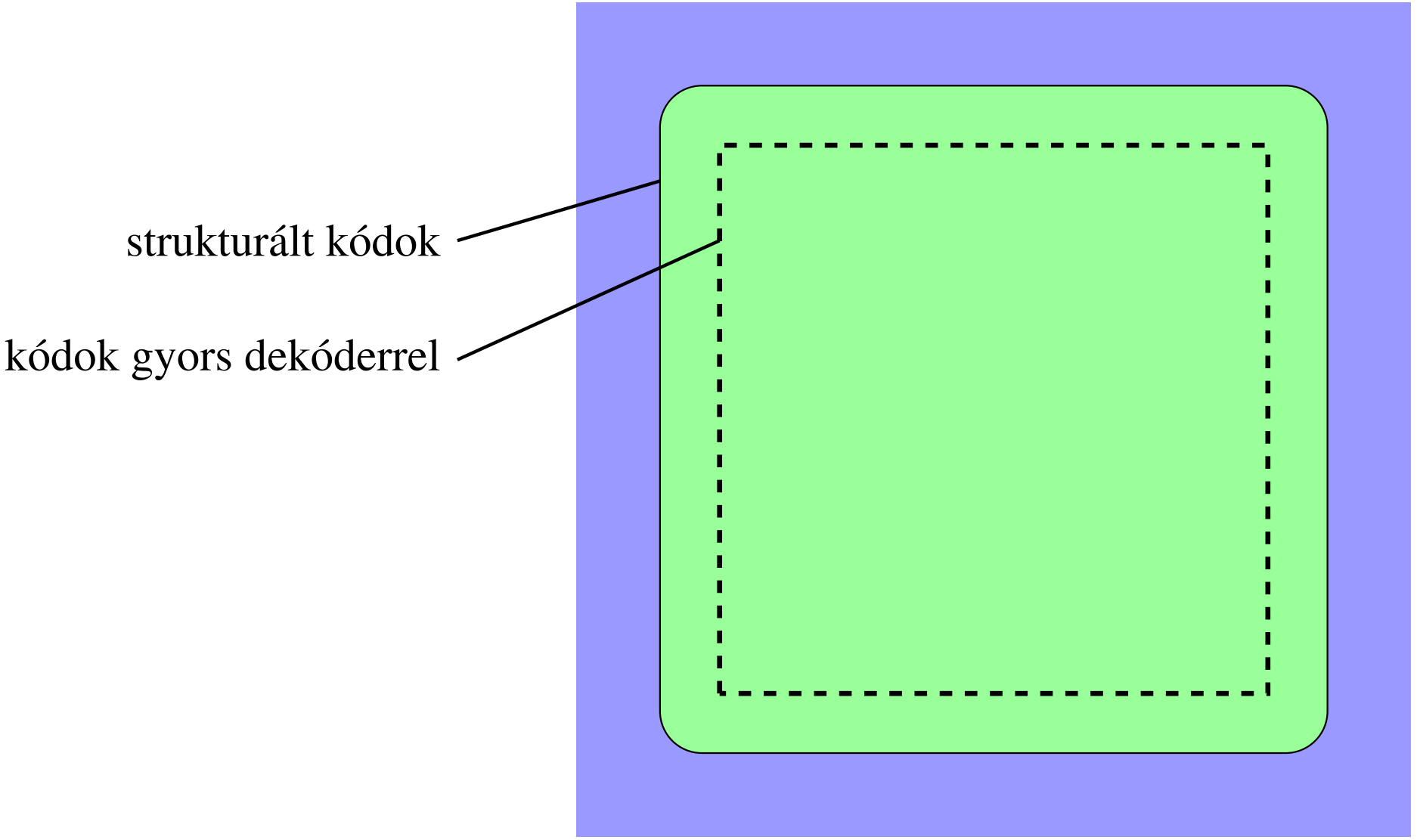
n hosszúságú, k dimenziós lineáris kódok $\mathbb{F}_q$ felett



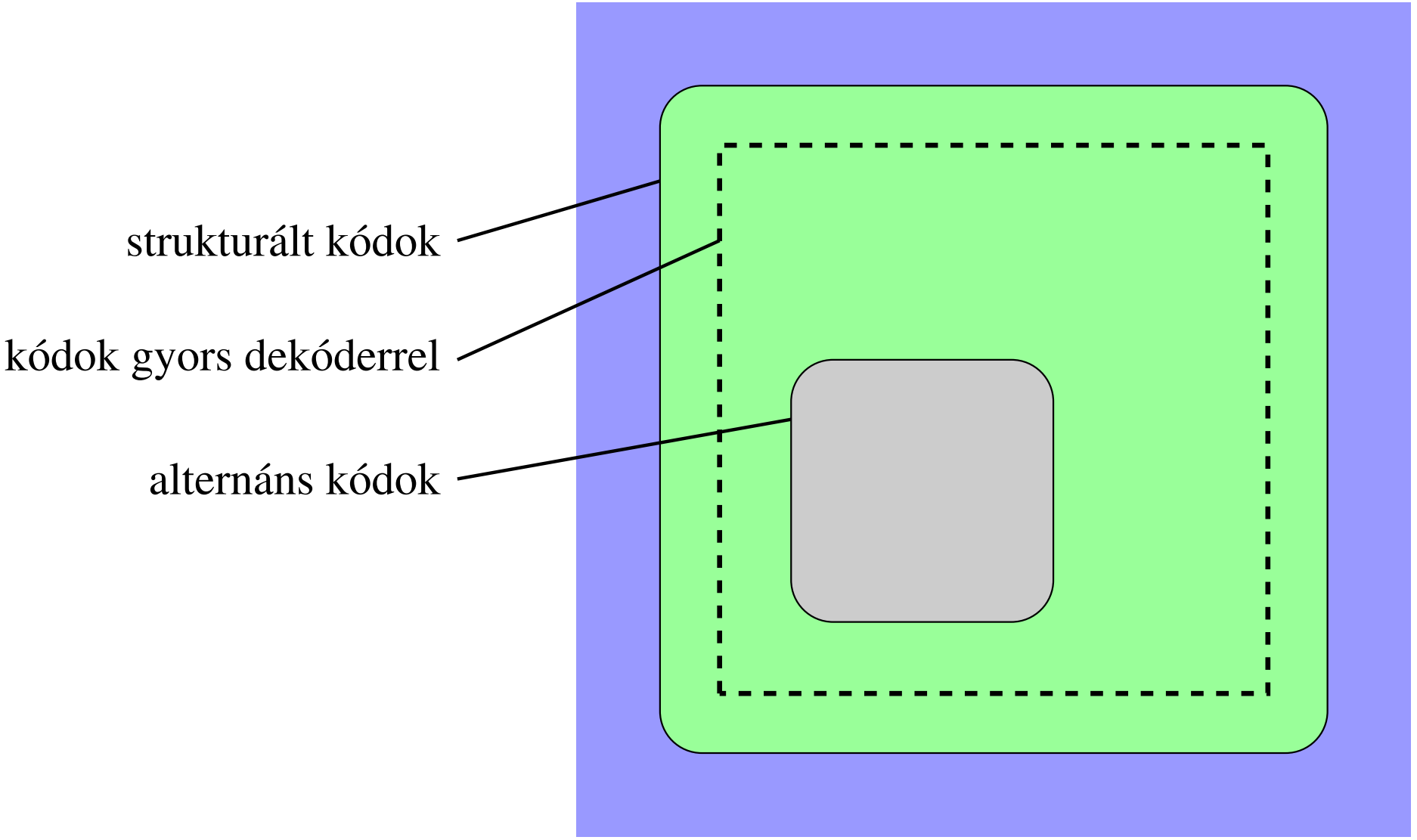
A strukturált kódok szigete



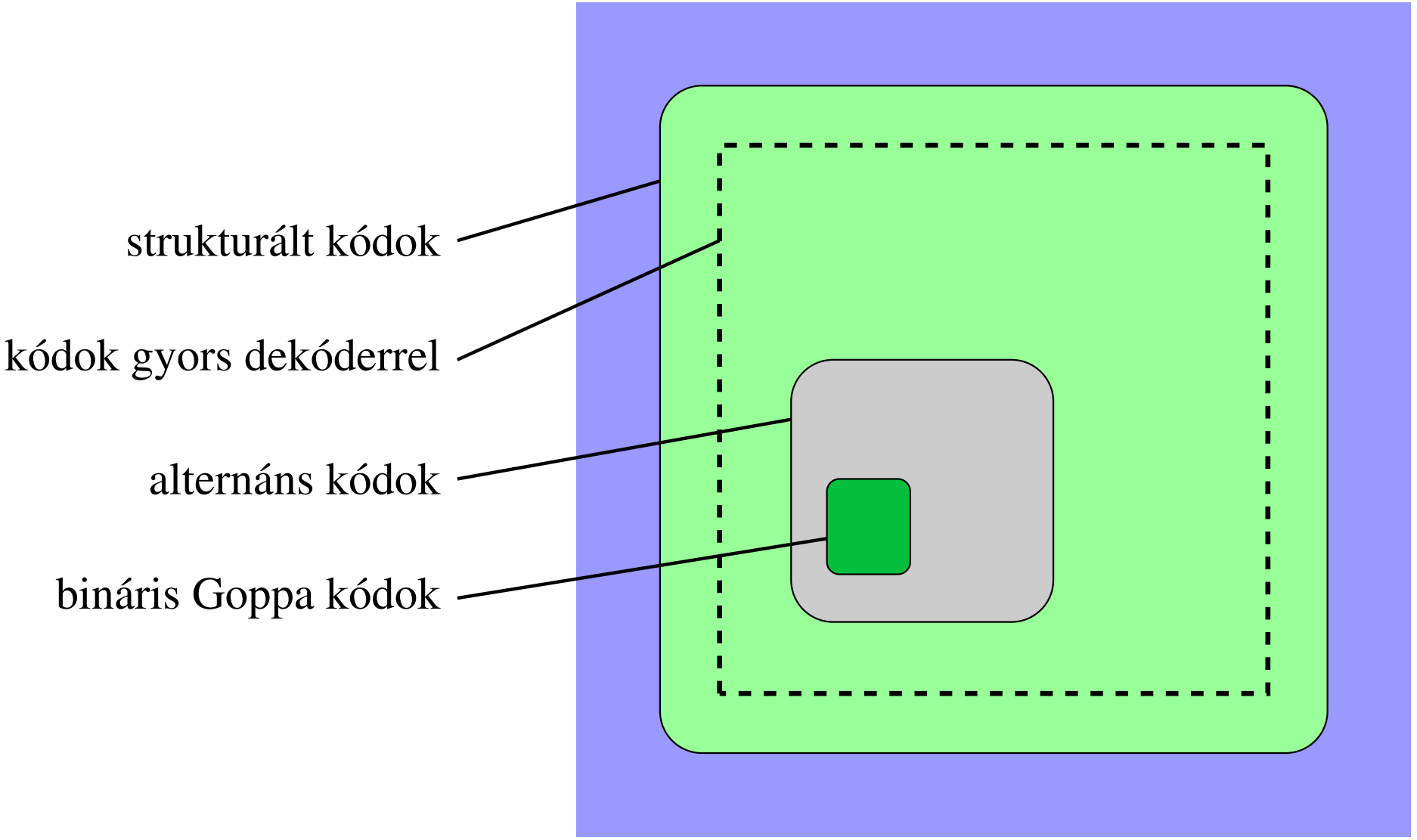
A strukturált kódok szigete



A strukturált kódok szigete



A strukturált kódok szigete



Tagolás

A McEliece-féle kriptoséma (1978)

Tegyük fel, hogy Alice egy k bites $\boldsymbol{m} \in \mathbb{F}_2^k$ titkos üzenetet akar Bobnak küldeni.

Kulcs generálás

1. Bob választ egy kellően nagy $q = 2^m$ 2-hatványt, $k < n \leq q$ egészt és t küszöbszámot úgy, hogy a megfelelő paraméterekkel rendelkező **alternáns kód** létezik.
2. Bob választ random $\alpha_1, \ldots, \alpha_n$ és $v_1, \ldots, v_n \in \mathbb{F}_q$ elemeket és megkonstruálja a
$$C = \mathbf{GRS}_k(\alpha, v) \cap \mathbb{F}_2^n$$
bináris alternáns kódot.
3. Bob képes t hibát javítani C-ben, mert ismeri az α_i, v_i értékeket. Ezek az értékek alkotják Bob **privát kulcsát.**
4. Legyen G a C egy *generátormátrixa.* Ez Bob **nyilvános kulcsa,** ezt elküldi Alicenak.

A McEliece-féle kriptoséma (1978)

Tegyük fel, hogy Alice egy k bites $\boldsymbol{m} \in \mathbb{F}_2^k$ titkos üzenetet akar Bobnak küldeni.

Kulcs generálás

1. Bob választ egy kellően nagy $q = 2^m$ 2-hatványt, $k < n \leq q$ egészt és t küszöbszámot úgy, hogy a megfelelő paraméterekkel rendelkező **alternáns kód** létezik.
2. Bob választ random $\alpha_1, \ldots, \alpha_n$ és $v_1, \ldots, v_n \in \mathbb{F}_q$ elemeket és megkonstruálja a
$$C = \mathbf{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}) \cap \mathbb{F}_2^n$$
bináris alternáns kódot.
3. Bob képes t hibát javítani C-ben, mert ismeri az α_i, v_i értékeket. Ezek az értékek alkotják Bob **privát kulcsát.**
4. Legyen G a C egy *generátormátrixa.* Ez Bob **nyilvános kulcsa,** ezt elküldi Alicenak.

A McEliece-féle kriptoséma (1978)

Tegyük fel, hogy Alice egy k bites $\boldsymbol{m} \in \mathbb{F}_2^k$ titkos üzenetet akar Bobnak küldeni.

Kulcs generálás

1. Bob választ egy kellően nagy $q = 2^m$ 2-hatványt, $k < n \leq q$ egészt és t küszöbszámot úgy, hogy a megfelelő paraméterekkel rendelkező **alternáns kód** létezik.
2. Bob választ random $\alpha_1, \ldots, \alpha_n$ és $v_1, \ldots, v_n \in \mathbb{F}_q$ elemeket és megkonstruálja a
$$C = \mathbf{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}) \cap \mathbb{F}_2^n$$
bináris alternáns kódot.
3. Bob képes t hibát javítani C-ben, mert ismeri az α_i, v_i értékeket. Ezek az értékek alkotják Bob **privát kulcsát.**
4. Legyen G a C egy *generátormátrixa.* Ez Bob **nyilvános kulcsa,** ezt elküldi Alicenak.

A McEliece-féle kriptoséma (1978)

Tegyük fel, hogy Alice egy k bites $\boldsymbol{m} \in \mathbb{F}_2^k$ titkos üzenetet akar Bobnak küldeni.

Kulcs generálás

1. Bob választ egy kellően nagy $q = 2^m$ 2-hatványt, $k < n \leq q$ egészt és t küszöbszámot úgy, hogy a megfelelő paraméterekkel rendelkező **alternáns kód** létezik.
2. Bob választ random $\alpha_1, \ldots, \alpha_n$ és $v_1, \ldots, v_n \in \mathbb{F}_q$ elemeket és megkonstruálja a
$$C = \mathbf{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}) \cap \mathbb{F}_2^n$$
bináris alternáns kódot.
3. Bob képes t hibát javítani C-ben, mert ismeri az α_i, v_i értékeket. Ezek az értékek alkotják Bob **privát kulcsát.**
4. Legyen G a C egy *generátormátrixa.* Ez Bob **nyilvános kulcsa,** ezt elküldi Alicenak.

Titkosítás és visszafejtés a McEliece-sémában

Titkosítás

1. Alice generál egy **random** t súlyú n hosszú $\boldsymbol{e} \in \mathbb{F}_2^n$ vektort.
2. Alice kiszámítja az
$$\boldsymbol{m}' = \boldsymbol{m}G + \boldsymbol{e}$$
vektort és elküldi Bobnak.

Visszafejtés

1. Bob *(polinomiális) dekódolási algoritmusa* meg tudja határozni azt az **egyetlen** $\boldsymbol{y} \in C$ kódszót, amire
$$d_H(\boldsymbol{m}', \boldsymbol{y}) \leq t.$$
2. Ez pontosan az $\boldsymbol{m}G$ kódszó.
3. Az
$$\boldsymbol{m}G = \boldsymbol{y}$$
lineáris egyenletrendszer megoldásával Bob meghatározza az $\boldsymbol{m}$ üzenetet.

Titkosítás és visszafejtés a McEliece-sémában

Titkosítás

1. Alice generál egy **random** t súlyú n hosszú $\boldsymbol{e} \in \mathbb{F}_2^n$ vektort.
2. Alice kiszámítja az
$$\boldsymbol{m}' = \boldsymbol{m}G + \boldsymbol{e}$$
vektort és elküldi Bobnak.

Visszafejtés

1. Bob *(polinomiális) dekódolási algoritmusa* meg tudja határozni azt az **egyetlen** $\boldsymbol{y} \in C$ kódszót, amire
$$d_H(\boldsymbol{m}', \boldsymbol{y}) \leq t.$$
2. Ez pontosan az $\boldsymbol{m}G$ kódszó.
3. Az
$$\boldsymbol{m}G = \boldsymbol{y}$$
lineáris egyenletrendszer megoldásával Bob meghatározza az $\boldsymbol{m}$ üzenetet.

Titkosítás és visszafejtés a McEliece-sémában

Titkosítás

1. Alice generál egy **random** t súlyú n hosszú $\boldsymbol{e} \in \mathbb{F}_2^n$ vektort.
2. Alice kiszámítja az
$$\boldsymbol{m}' = \boldsymbol{m}G + \boldsymbol{e}$$
vektort és elküldi Bobnak.

Visszafejtés

1. Bob *(polinomiális) dekódolási algoritmusa* meg tudja határozni azt az **egyetlen** $\boldsymbol{y} \in C$ kódszót, amire
$$d_H(\boldsymbol{m}', \boldsymbol{y}) \leq t.$$
2. Ez pontosan az $\boldsymbol{m}G$ kódszó.
3. Az
$$\boldsymbol{m}G = \boldsymbol{y}$$
lineáris egyenletrendszer megoldásával Bob meghatározza az $\boldsymbol{m}$ üzenetet.

A McEliece-séma biztonsági elemzése *(kriptoanalízis)*

- **Privát kulcs:** $\alpha_1, \ldots, \alpha_n, v_1, \ldots, v_n$. Ezekből konstruáljuk a C alternáns kódot.
- **Nyilvános kulcs:** Az alternáns kód G generátormátrixa.
- A séma biztonságos, ha a nyilvános kulcsból a privát kulcs meghatározása **nehéz feladat.**
- Ennél kicsit többet várunk el: G ne legyen megkülönböztethető egy hasonló méretű **random mátrixtól.**
- Ekkor a biztonságot garantálja, hogy random mátrix esetén a dekódolási feladat **nehéz** (Berlekamp–McEliece–van Tilborg-tétel).
- Ehhez az n, k, t értékeket **kellően nagyra** kell választani.

Klasszikus McEliece modern köntösben

- McEliece eredeti 1978-as javaslata a **bináris Goppa kódokra** épül.
- Ez a $v_1, \ldots, v_n$ *multiplikátoroknak* egy speciális megválasztását jelenti.
- Erre az osztályra a mai napig nem sikerült a privát kulcs visszafejtése.
- Az **eredeti javaslatban** szereplő paraméterek:

$$n = 1014, \quad k = 524, \quad t = 50.$$

- A **2020-as NIST javaslatban** a legmagasabb 256 bites biztonsági szinthez javasolt paraméterek:

$$n = 6686, \quad k = 128, \quad t = 13.$$

- **Túl nagy nyilvános kulcs:** 32, illetve 836 kilobyte.
- *Rengeteg ötlet és javaslat a nyilvános kulcs méretének lényeges csökkentésére – eddig sikertelenül...*

KÖSZÖNÖM A FIGYELMET!

Érdeklődőknek:

- szakdolgozati témák
- TDK dolgozati témák
- NSUCRYPTO verseny felkészítés

KÖSZÖNÖM A FIGYELMET!

Érdeklődőknek:

- szakdolgozati témák
- TDK dolgozati témák
- NSUCRYPTO verseny felkészítés