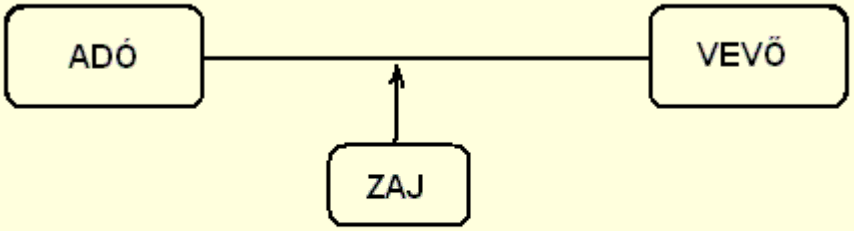
2. tétel: Kommunikácós csatorna modellje.

- Kommunikácós csatorna modellje.
- Információ mérése. Példák.
- Redundancia, tömörítés
- Titkosítás.
- Digitális aláírás.
- Jogi alapfogalmak: netikett, freeware, shareware, adatvédelem jogi szabályozásai
- Hogyan hatott a cenzúrára az Internet?
- Internethasználattal kapcsolatos bűncselekmények
- Melyek azok a problémák, amelyek a szoftver és adatmásoláskor (zene, film) felmerülnek?



Egy információs csatorna áll egy adóból, agy adatcsatornából meg egy vevőből.

Az adó egy véges jelkészletet (ABC-t) használva kódolja az üzenetet egy véges karaktersorozattá, amelyet az adatcsatornán keresztül küld el.

A vevő dekódolja a kapott jelsorozatot, és amennyiben lehetséges, értelmezhető üzenetté állítja vissza azt.

A jelsorozat a kommunikációs csatornán áthaladva a külső behatások miatt módosulhat. Ezt a külső hatást nevezzük zajnak.

Az információ mérése.

Az információ mérésére először Claude Shannon amerikai távközlési mérnök tett javaslatot. Ő vezette be az ún. Shannon entrópiát.

Szintén ő javasolta az információ mértékegységére a bitet

Jelöljük a kódolásra használt a jelkészletet $\{a_1,a_2,a_3,.....a_N\}$ szimbólumokkal. Ezek a szimbólumok valamilyen gyakorisággal fordulnak elő az üzenetekben. Vezessük be egy jel gyakoriságára a következő jelölést:

Jel	Gyakoriság:
a_1	p_1
a_2	p_2
a_3	p_3
....	.....
a_N	p_N

A Shannon entrópia (H) az egy karakterre jutó átlagos információmennyiség:

$$H=\sum_{i=1}^{N} p_i \mathrm{Log}_2 \frac{1}{p_i}=p_1 \mathrm{Log}_2 \frac{1}{p_1}+p_2 \mathrm{Log}_2 \frac{1}{p_2}+\ldots p_N \mathrm{Log}_2 \frac{1}{p_N}$$

A Shannon entrópia mértékegysége a **bit/szimbólum** vagy **bit/jel.**

Példák

Bináris kód

Két szimbólum van, az a_1=0 illetve az a_2=1. Mindkettő azonos valószínűséggel fordul elő egíy tetszőleges bináris kódban, ezért a p_1=0,5 illetve a p_2=0,5.

A Shannon entrópia ekkor:

$$H = p_1 \mathrm{Log}_2 \frac{1}{p_1} + p_2 \mathrm{Log}_2 \frac{1}{p_2} = \frac{1}{2}\mathrm{Log}_2 \frac{1}{\frac{1}{2}} + \frac{1}{2}\mathrm{Log}_2 \frac{1}{\frac{1}{2}} = \mathrm{Log}_2 2 = 1 \ {}^{\mathrm{bit}}/_{\mathrm{jel}}$$

Ez azt jelenti, hogy egy binárisan kódolt üzenet annyi bites, ahány bináris számjegyet tartalmaz: pl. az "100110" üzenet 6 bites.

A Morse ábécé

Három jelből áll:

-	p_1=0,36
ˇ	p_2=0,34
betűköz	p_3=0,30

A valószínűség értékei általam becsült értékek.

A Shannon entrópia:

$$H = p_1 \mathrm{Log}_2 \frac{1}{p_1} + p_2 \mathrm{Log}_2 \frac{1}{p_2} + p_3 \mathrm{Log}_2 \frac{1}{p_3}$$

Behelyettesítve:

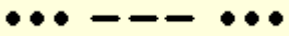
$$H = 0{,}36 \cdot \mathrm{Log}_2 \frac{1}{0{,}36} + 0{,}34 \cdot \mathrm{Log}_2 \frac{1}{0{,}34} + 0{,}30 \cdot \mathrm{Log}_2 \frac{1}{0{,}30}$$

Kiszámítjuk a logaritmusokat, majd a műveleteket elvégezve:

$$H = 0{,}36 \cdot 1{,}4739 + 0{,}34 \cdot 1{,}5564 + 0{,}30 \cdot 1{,}7369 = 1{,}58 \,\mathrm{bit/jel}$$

Tehát a Morse ABC egy jelének Shannon entrópiája 1,58 bit/jel

Így például az SOS üzenet (ti-ti-ti-betűköz-tá-tá-tá-betűköz-ti-ti-ti-betűköz):



12 jelet tartalmaz, az üzenet így 12*1,58 = 18,96 bites.

A magyar ábécé

Jel	p_i	$Log_2(1/p_i)$	$p_iLog_2(1/p_i)$	Jel	p_i	$Log_2(1/p_i)$	$p_iLog_2(1/p_i)$
szóköz:	0.11719	3.093078627	0.362477884	b:	0.01887	5.7277618	0.108082865
e:	0.10189	3.29491563	0.335718954	ö:	0.0183	5.7720125	0.10562783
a:	0.09278	3.430042344	0.318239329	h:	0.01808	5.7894615	0.104673464
t:	0.09213	3.440185177	0.31694426	j:	0.01179	6.4062925	0.075530188
l:	0.06523	3.938320561	0.25689665	f:	0.01098	6.5089781	0.07146858
n:	0.05671	4.140253033	0.23479375	u:	0.01043	6.583117	0.068661911
s:	0.05212	4.262019106	0.222136436	p:	0.00934	6.7423617	0.062973659
k:	0.04586	4.446619835	0.203921986	ó:	0.00871	6.8431116	0.059603502
o:	0.04364	4.518205088	0.19717447	c:	0.00849	6.8800197	0.058411368
r:	0.04346	4.524168015	0.196620342	bekezdésvég	0.00677	7.2066285	0.048788875
m:	0.03819	4.71066127	0.179900154	ü:	0.00589	7.4075167	0.043630273
z:	0.03721	4.748165799	0.176679249	í:	0.00439	7.8315633	0.034380563
i:	0.03667	4.769255924	0.174888615	ú:	0.00343	8.1875757	0.028083385
g:	0.03527	4.825414614	0.170192373	ű:	0.00177	9.1420349	0.016181402
á:	0.03159	4.984388253	0.157456825	ő:	0.00092	10.086079	0.009279192
é:	0.02979	5.069028066	0.151006346	w:	0.00047	11.055052	0.005195874
y:	0.02305	5.439089439	0.125371012	x:	0.00027	11.854753	0.003200783
d:	0.02284	5.452293539	0.124530384	q:	0.00003	15.024678	0.00045074

Az egyes valószínűségek 4 500 000 karakter statisztikai vizsgálata alapján lettek megállapítva.

Magyar nyelvű szövegekre a Shannon entrópia értékére **4,91875 bit/jel** adódik. Ha azonban figyelembe vesszük azt, hogy bizonyos betűk előtt vagy után milyen valószínűséggel szerepelnek más betűk, vagyis a betűkorrelációt, akkor erre az értékre **H=1,5 bit/jel** adódik! A betűkorreláció azt jelenti, hogy pl. az y előtt valószínűleg g, n, t, l áll, (gy, ny, ty, ly) de pl. a szóköz vagy egy másik y nem valószínű.

Redundancia

Mikor lenne a Shannon entrópia a fenti 36 jelből álló ABC esetén maximális? Ez akkor következne be, ha minden karakter (jel) egyforma valószínűséggel fordulna elő. Ez azt jelenti, hogy minden jel p=1/36 valószínűséggel fordulna elő.. Ekkor a Shannon entrópia:

$$H_{max} = \underbrace{\frac{1}{36}\cdot Log_2\frac{1}{\frac{1}{36}} + \frac{1}{36}\cdot Log_2\frac{1}{\frac{1}{36}} + \ldots + \frac{1}{36}\cdot Log_2\frac{1}{\frac{1}{36}}}_{36\ tag} = Log_2 36 = 5{,}167\ bit/jel$$

Egy üzenetnek akkor nagy a redundanciája, ha nagy a zajtűrőképessége. A zajtűrőképesség azt jelenti, hogy a zaj hatására eltorzult üzenet milyen mértékben állítható vissza. A zajtűrőképességet úgy növelhetjük meg, hogy a jeleket valamilyen módon többszörösen tároljuk.

-

A redundancia tehát végső soron az üzenetben fellelhető jelismétlések gyakoriságát jelenti

A redundancia matematikai képlettel pontosabban is megadható:

$$R = 1 - \frac{H}{H_{max}} = 1 - \frac{1{,}5}{5{,}167} = 0{,}71$$

Ez azt jelenti, hogy egy magyar szövegben a betűk (jelek) 71%-a felesleges, azaz jelismétlés. Ez általában minden beszélt nyelvre igaz, hogy kb 3 betűből 2 jelismétlés.

Mi ennek az oka? A válasz egyszerű: a beszélt nyelvek igen erős zajhatásnak vannak kitéve, továbbá az "adó" lehet beszédhibás, a vevő lehet, hogy nagyothalló, ezért az emberek úgy alakították ezeket a nyelveket, hogy ezeket a zajhatásokat eltűrjék.

A redundancia arról is informál minket, hogy az adott üzenet milyen mértékben tömöríthető.

Egy üzenet várható tömörítési aránya 1-R. Magyar szövegre pl. 1-0,71=0,29, vagyis az átlagosan elérhető tömörítési arány 29%.

Hibaérzékelő kódolás

Ha egy üzenetet úgy kódolunk, hogy az esetleges zaj okozta módosulásokat észlelni lehessen, akkor hibaérzékelő kódolást alkalmazunk.
Legyen pl. az üzenet a következő:

counterstrike

A zaj miatt növeljük a redundanciát úgy, hogy minden karaktert kétszer küldünk el:

ccoouunntteerrssttrriikkee

A zaj miatt a vevő ezt a következőképpen kapja meg:

ccoouunzzteerrssttrriikkee

A vevő a karaktereket párosával olvasva összehasonlítja a két beolvasott karaktert. Ha egyformák, akkor nem történt hibás vétel. Az n betűnél azonban két különböző karaktert olvasott be. Nyilván itt a zaj miatt valamelyik karakter a zaj hatására módosult, de a vevő nem tudja eldönteni, hogy melyik az eredeti karakter, és melyik a hibás: az **n** vagy a **z**? Ebben az esetben hibaérzékelő kódolással van dolgunk.

Ez a módszer nem százszázalékos, mert ha a zaj miatt a két egymásután küldött karakter módosul, ráadásul ugyanúgy, akkor a hiba észrevétlen marad. Ennek azonban elég kicsi a valószínűsége

Hibajavító kódolás

Ha egy üzenetet úgy kódolunk, hogy az esetleges zaj okozta módosulásokat ne csak észlelni lehessen, hanem nagy biztonsággal javítani is,
akkor hibajavító kódolásról beszélünk.

Lássunk erre is egy példát:

Küldjük el az előbbi üzenetet úgy hogy tovább fokozzuk a redundanciáját, mégpedig úgy, hogy minden karaktert 3-szor küldünk el:

cccooouuunnnttteeerrrsssttrrriiikkkeee

A zaj hatására itt is egy karakter módosulhat

cccooouuunnzttteeerrrsssttrrriiikkkeee

A vevő a karaktereket hármasával olvasva a vevő ellenőrzi, hogy a karakterek egyformák-e? Amennyiben nem, akkor "a többség győz" elve alapján javítunk: az **n**, **n**, **z** esetén megállapítható, hogy az **n** a helyes karakter, így a hiba javítható.

Ez a módszer sem százszázalékos, de a többszörös karakterismétléssel ennek a biztonsága fokozható.

A gyakorlatban a bemutatottnál hatékonyabb módszereket használunk, ahol a hibajavító információ csupán 8-10 %-a a valódi üzenetnek.

A hibajavító kódolást a távközlésben, adatrögzítésnél (CD, DVD) alkalmazzuk.

Tömörítés

Egy üzenet csak akkor tömöríthető, ha nagy a redundanciája. Ha tehát a z üzenet redundanciája kicsi, akkor ne próbálkozzunk tömörírtéssel.

A tömörítés hatására a redundancia nagymértékben lecsökken. Ezért a tömörített üzenetek igen érzékenyek a zajra. Ez azt jelenti, hogy ha egy tömörített üzenet megsérül, akkor nem, nagyon kicsi a valószínűsége annak, hogy vissza tudjuk állítani a tömörítetlen változatot

Kétféle tömörítés létezik:

- **Veszteségmentes:** Itt az üzenet teljes egészében visszaállítható Ilyen módszerrel tömörítjük a **ZIP, ARJ, RAR** kiterjesztésű állományokat.
- **Veszteséges**: Itt az üzenet csak bizonyos mértékben állítható vissza. Ilyen módszerrel tömörítjük a **JPG, MP3, WMA, MPEG** állományokat

Veszteségmentes tömörítést inkább ott alkalmazunk, ahol fontos az, hogy az állományunk bitről bitre, bájtról bájtra visszaállítható legyen. Ilyenek a programfájlok, futtatható állományok, dokumentumok, szövegek.

A veszteséges tömörítést a kép-és hangállományok tömörítésére használjuk, mivel pl. egy képnél nem tűnik fel, ha itt-ott egy pixel színe egy halvány árnyalattal eltér az eredetitől. A veszteséges tömörítések általában jóval nagyobb tömörítési arányt tesznek lehetővé, mint a veszteségmentes módszerek. Azonban a tömörítési arány növelése a minőség egyre nagyobb romlásához vezet

Rejtjelezés

Alapvetően két rejtjelezés-típust különböztethetünk meg:

1. A titkos kulcsú rejtjelezést - itt mindkét félnek ismerni kell a kulcsot, amit titokban kell tartani
2. A nyilvános kulcsú rejtjelezést - itt a kulcsot kettévágjuk, egyik részét nyilvánossá tesszük, a másikat titokban tartjuk. Természetesen a nyilvános kulcs ismeretében kiszámítható a titkos kulcs is, de ez igen nagyon-nagyon-nagyon sok időbe telne...

A titkos kulcsú rejtjelezés

Az üzleti élet, az elektronikus kereskedelem, elektronikus ügyintézés szükségessé teszi a a titkosítási módszerek alkalmazását
Az egyszerű titkosítási módszerek:

- A Caesar kód
- A Vigenére kód (olvasd: Vizséner)

A Caesar kód a legegyszerűbb, betűkeverésen alapuló módszer, amikor egy betűt egy másikkal helyettesítűnk.

Az eredeti, Julius Caesar által használt verziót mutatjuk be. Az eredeti latin ábécé (ami egyezik az angollal) három hellyel való eltolásával kapjuk a kódábécét, a következő képpen:

Nyílt ABC	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Kód ABC	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**	**A**	**B**	**C**

A "**MAMA**" nyílt szöveg kódolása "**PDPD**". Ebből már látható is a gyenge pontja ennek a kódnak. Egy adott betűt mindig ugyanazzal a másik betűvel kódolunk. Julius Caesar ezt a kódot megfejthetetlennek tartotta.

Visszafejtés:

A címzettnek ismerni kell az eltolás mértékét, a KULCSOT. A kulcs lehet a fenti táblázat is, amit most fordítva alkalmazunk: az alsó betűnek keressük meg a megfelelőjét a felső sorból.

A Caesar kód természetesen tetszőleges mértékű eltolással is alkalmazható. A betűhelyettesítésre más módszerek is alkalmazhatók, ez azonban a feltörést különösebben nem nehezíti meg.

A középkorra a kémek, hadvezérek és diplomaták körében a ismertté vált az a tény, hogy a Caesar kód könnyen feltörhető.

Ezért a XVI. században Blaise de la Vigenére (ejtsd bléz dö lá vizséner) francia kriptográfus (=rejtjelező) egy új módszert dolgozott ki, amely egy összetett Caesar kódolás, melynél minden betűhöz más eltolási értéket rendelünk. Az egymásutáni eltolási értékek alkotják a KULCSOT.

A kulcs tehát egy többjegyű szám, például 2361. Legyen a kódolandó szöveg: "NAPLEMENTE"

Alája írjuk a kulcsot, egymásután folytonosan:

N	A	P	L	E	M	E	N	T	E
2	3	6	1	2	3	6	1	2	3
P	D	V	M	G	P	K	O	V	H

majd az adott betűtől annyi karakterrel ugrunk jobbra az ABC-ben, amennyi a kulcsban lévő számjegy. Ezért az N-től 2-re levő karakter a P, az A-tól már 3 karaktert ugrunk, stb. Ez tehát egy összetett Caesar kód. Noha több E betű van a kódban, az egyes E betűk kódja mégsem egyforma. Ez lényegesen nehezíti a feltörést.

Visszafejtés

P	D	V	M	G	P	K	O	V	H
2	3	6	1	2	3	6	1	2	3
N	A	P	L	E	M	E	N	T	E

A kódolt üzenet alá írjuk az ismert kulcsot, amit a küldő és a címzett egyaránt ismer. Az első karakter a **P**. Ettől 2 karaktert visszaszámolunk, mivel a P alatt a **2** számjegy van- A visszafejtett karakter az **N** A második betű a **D**. Alatta a **3** számjegy van. A **D**-től balra 3 betűnyire az ABC-ben az **A** található, stb.

A Vigenére kód hosszú ideig tartotta magát, Először annak A Vigenére kódot 1854-ben törte fel az a Charles Babbage, aki az első programozható számítógép terveit is kidolgozta.

A titkos kulcsú rejtjelezést ma is alkalmazzák, ilyen a DES kódnevű módszer.

Nyilvános kulcsú rejtjelezés

A titkosítás gyengéje a XX. század második feléig az a tény volt, hogy a kulcsokat a küldőnek el kellett juttatni a címzetthez, ami kockázatos volt. Vagyis a küldőnek meg a címzettnek meg kellett egyeznie egy közös kulcsban, amivel rejtjeleztek illetve visszafejtettek

Az 1970-es években dolgozták ki az úgynevezett nyilvános kulcsú titkosítást.

Az RSA titkosítás

Legelterjedtebb ilyen típusú titkosítás az RSA, amely egy kulcspáron alapul: a K_{NY} kulccsal titkosítunk, a K_T kulccsal visszafejtünk. A K_{NY} kulcsot nyilvánossá tesszük, a K_T kulcsot titokban tartjuk. Hogyan függ össze egymással a két kulcs? A K_T kulcsot egy nagyon nagy, 5-600 jegyű prímszámból készítjük.

A K_{NY} nyilvános kulcs az előbbi prímszám és egy másik, szintén nagy prímszám szorzata alapján készül. Így egy olyan összetett szám a nyilvános kulcs, amely 1000 -1200 számjegyű lesz. Ezt a nyilvános kulcsot és a hozzá tartozó titkosító eljárást (programot) közzé tesszük, ezzel bárki, akinek ezt a kulcsot megadtuk titkosított üzenetet küldhet nekünk. Visszafejteni, elolvasni azonban csak az tudja, akinél a titkos kulcs van, vagyis csak mi, hiszen a K_{NY} prímtényezős felbontását csak mi ismerjük.

A feltörés nehézsége abban áll, hogy noha minden szám prímtényezőkre bontható, bizonyos nagy számok esetében a feladat igen nehéz, éspedig pont azoknál a számoknál, amelyek két nagyon nagy prímszám szorzatai!

Példa az illusztrálásra:

1. Legyen a felbontandó szám **391**. Ez csak egyféleképpen bontható fel: **391=17*23**. Más felbontása nincs... Különösebben nem nehéz a felbontása, de meg kell dolgozni érte...

2. Legyen a prímtényezőkre bontandó szám **1 506 467**. Tessék próbálkozni! Ennek egyedüli lehetséges felbontása: **997*1511**. Ha nem ismerjük a felbontást, végig kell próbálni az összes prímszámmal egészen **997**-ig, Először próbáljuk osztani 2-vel, aztán 3-mal, aztán 5-tel, 7-tel, 11-gyel, 13-mal, stb. és így tovább, amíg végre meg nem kapjuk azt a prímszámot, amely maradék nélkül osztja **1 506 467**-et! Ez összesen **168** osztást jelent, mert 997-ig pontosan ennyi prímszám van. Itt a nyilvános kulcs **K_{NY} =1 506 467** , a titkos kulcs **K_T =997** lenne.

3. Legyen most a szám **313 725 463 505 593**. Ez 15 jegyű. Ennek a felbontásához majdnem 1 000 000 osztást kell(ene) elvégezni.
   A felbontás: **313 725 463 505 593=15 395 203 * 20 378 131**
   A nyilvános kulcs: K_{NY} **=313 725 463 505 593**, a titkos kulcs: K_T **=15 395 203**.

A számjegyek növelésével a titkos kulcs kiszámítása a nyilvános kulcs ismeretében egyre reménytelenebb: 1000 jegyű nyilvános kulcs esetén és kb 500 jegyű titkos kulcs esetén kb. 10^{497} próbálkozás kellene!

A nyilvános kulcsú titkosítást az elektronikus levelezésben, elektronikus kereskedelemben, pénzügyi tranzakciók védelmére illetve elektronikus ügyintézésben használjuk.

Digitális aláírás

Az ügyintézéskor, elektronikus iratok kiadásakor, üzleti tranzakcióknál fontos azt tudni, hogy az üzenetet valóban a megfelelő személy küldte, továbbá azt is szeretnénk tudni, hogy az adott banki átutalást, üzleti rendelést, szerződést, stb. nem módosította utólag valaki.

Erre dolgoztak ki egy olyan módszert, amit digitális aláírásnak neveznek. Az aláírás két lépésben történik.

1. Az első lépésben az aláírandó dokumentum minden karakterét felhasználva kiszámítunk egy úgynevezett ellenőrző értéket, ez az üzenet lenyomata. Ez egy fix hosszúságú, általában128 bites szám.
   A lenyomat előállítása egy olyan különleges bit-keverési módszerrel történik, hogy amennyiben a legkisebb módosítást is végrehajtjuk az eredeti állományon, úgy az ellenőrző összeg, vagyis a lenyomat értéke jelentősen módosul.

2. Ezután a lenyomatot RSA módszerrel titkosítja a feladó, a saját titkos kulcsát használva.

Az ellenőrzés:

A címzett ismeri a feladó nyilvános kulcsát. Ezzel visszafejti a lenyomatot, majd elkészíti ő is a megkapott dokumentum lenyomatát, és összeveti a visszafejtett lenyomattal.

Amennyiben egyezik a kettő, úgy a címzett meggyőződik arról, hogy:

1. Az dokumentumot valóban a címzett küldte, mert ezzel a módszerrel csak ő tud titkosítani

2. Az dokumentum sértetlen, vagyis ő is pontosan azt a dokumentumot látja, amit a feladó digitálisan aláírt, hiszen az elküldött üzenet lenyomata és az általunk kapott üzenet lenyomata egyezik.

Rejtjelezés és tömörítés

A rejtjelezés egy adott üzenet Shannon entrópiáját megnöveli, és emiatt a redundancia is lecsökken.

Ennek fontos következményei:

- A rejtjelezett üzenetet nem, vagy csak alig lehet tömöríteni. Ezért ha tömöríteni is akarunk, akkor ezt a rejtjelezés előtt tegyük meg!
- A rejtjelezett üzenetek a lecsökkent redundancia miatt sokkal érzékenyebbek a zajra, ezért ilyen üzeneteket csak igen csekély zajú kommunikációs csatornákon küldözgessünk!