Titkosítások – elemző esszé

(ezzel lesz feltoltve a weblap hogy ne csak egy login legyen rajta)

<https://www.webmaster442.hu/alapveto-titkositasi-modszerek-es-algoritmusok/>

*A titkosítás története egészen az ókorig nyúlik vissza. Katonai célokra az idők kezdete óta alkalmaztak titkosítási algoritmusokat. Ebben a cikkben pár, mára már alapnak mondható titkosítási algoritmust mutatok be, és egy kicsit lesz szó a titkosítási alapfogalmakról is.*

## *Alapfogalmak*

*A titkosítási rendszerek tárgyalása kapcsán találkozhatunk néhány alapfogalommal. A fontosabbak:*

* *Plaintext  
  Egyszerű szöveg, ami tikosításra kerül.*
* *Ciphertext  
  Titkosított szöveg, amit algoritmus állít elő a bemeneti szövegből.*
* *Encryption, titkosítás  
  A folyamat, amely létrehozza az egyszerű szövegből a titkosított szöveget.*
* *Titkosító algoritmus  
  A titkosítás műveletsora. Mindig két bemenete van: titkosítandó szöveg és titkosító kulcs.*
* *Visszafejtés, Deciphering, Decryption  
  Folyamat, amely során titkosított szövegből egyszerű szöveg lesz.*
* *Visszafejtő algoritmus  
  Visszafejtés műveletsora. Két bemenete: titkosított szöveg és titkosítást feloldó kulcs.*
* *Titkos kulcs  
  Titkosító és titkosítást feloldó kulcs elnevezése, ha megegyeznek. Szokás szimmetrikus kulcsnak is nevezni.*
* *Kriptográfia, Cryptography  
  Titkosításokkal foglalkozó tudomány.*

*Titkosítási rendszerek esetén beszélhetünk szimmetrikus és aszimmetrikus rendszerekről. Az aszimmetrikus titkosítás csak az elmúlt század második felében vált ismertté, így az összes klasszikus algoritmus szimmetrikus felépítésű volt.*

*A bemeneti adat feldolgozásának szempontjából beszélhetünk stream alapú titkosításokról és blokk alapú titkosításokról. A stream alapú titkosítások bitenként, vagy papír alap esetén karakterenként dolgozzák fel a bemeneti információt. A blokk alapú titkosítási rendszerek pedig a bemeneti adatot fix méretű blokkokként kezelik és a blokk egészén végeznek műveleteket. A legtöbb ma használt algoritmus blokkokban dolgozik sebességbeli megfontolások miatt. Az, hogy blokkban vagy steam-ben dolgozik az algoritmus, ideális esetben nem kellene, hogy hatással legyen a biztonságra.*

## *Törési módszerek*

*Többféleképpen nekiállhatunk egy kód feltörésének. A legegyszerűbb és legjobban erőforrás pazarló módszer a Brute Force. Ez azt jelenti, hogy az összes lehetséges kombinációt ki kell próbálni. Általában az összes lehetséges kombinációt nem kell kipróbálni a sikeres töréshez, legjobb esetben elég egyszer próbálkozni. Ennek azonban a matematikai esélye ugyan megvan, de nem éppen életszagú eset.*

*Ha egy kicsit gondolkodunk és megvizsgáljuk a visszafejtendő kódot, akkor előállhatunk rafináltabb módszerekkel, amelyek segítségével könnyebben vissza tudjuk fejteni az eredeti információt. Ideális esetben a titkosító algoritmus nem rendelkezik hibákkal és csak Brute Force módszerrel törhető fel.*

*A Brute Force töréshez szükséges idő erősen függ a kódolás bonyolultságától, valamit a töréshez használt gép sebességétől. Éppen ezért fontos kérdés, hogy belátható időn belül feltörhető-e a rendszer vagy nem. A belátható idő fogalma erősen relatív. Éppen ezért jobb kifejezés egy módszer biztonságosságának megítélésére az, hogy megnézzük, számításilag biztonságosnak tekinthető-e.*

*Egy algoritmus számításilag biztonságosnak tekintett, ha a feltöréséhez annyi erőforrás és idő szükséges (pl. 1000 év), ami nem áll senki rendelkezésére.*

## *Caesar titkosítás*

*A legöregebb ismert titkosítási algoritmus. Állítólag Julius Caesar találta fel, azonban ennek a bizonyítására nem sok információ áll rendelkezésre. Az biztos, hogy az ő idejében már ismert volt a módszer. Ez egy betűcserén alapú algoritmus.*

*A cserélő algoritmusok lényege, hogy adott betűt egy másikkal helyettesítünk, így kapunk egy értelmetlen szöveget, ami már titkos tartalmú. A Caesar titkosítás esetén a csere szabály az, hogy minden betűhöz az ABC szerint elfoglalt pozíciója alapján a három hellyel arrébb található karaktert rendeljük: A -> D, B -> E, C -> F … Körbeérés után pedig az X, Y, Z betűkhöz az A, B, C betűket rendeljük.*

*Példa bemenet: a titkositas jo  
Kimenet: D WLWNRVLWDV MR*

*Ez első ránézésre megfejthetetlennek tűnik, főleg, ha nem három karakter eltolást alkalmazunk, hanem többet, vagy más elven cseréljük a betűket. Ha nem tudjuk a betűcsere algoritmusát, akkor 26!, az-az 403 291 461 126 605 635 584 000 000  számú próbálkozás kell a feltöréshez, legrosszabb esetben. Ha másodpercenként 10 millió lehetőséget ki tudunk próbálni, akkor nagyjából “csak” 1 278 828 834 115,31 évre van szükségünk a kódolás visszafejtéséhez.*

*Ha viszont jobban szemügyre vesszük a kódolást, akkor azt vesszük észre, hogy a tikosítás nem változtat semmit a betűk ismétlődési gyakoriságán. A példában a kódolatlan szövegben 2db A betű található, a kimeneten ez 2db D betűként jelentkezik. Tehát a feltöréshez kell egy ismétlődési statisztikát készíteni a szövegben található betűkről. Ezt pedig össze kell hasonlítani egy olyan statisztikával, amely az adott nyelvre vonatkozik. Ennek az elkészítése igen nehézkes tud lenni, ha papír alapon csináljuk, mivel az összes létező szó esetén figyelembe kell venni az ismétlődési valószínűségeket. Viszont ha már van statisztikánk, akkor onnantól kezdve nagyon könnyen visszafejthető a kódolás.*

*Probléma akkor van, ha nem ismerjük a nyelvet, amin az eredeti szöveg íródott. A II. világháborúban az Amerikai Egyesült Államok kormánya ezt a taktikát vetette be titkosítás terén. Egy Indián törzs tagjait toborozták be titkosításra. A törzs tagjai olyan anyanyelvvel rendelkezek, amelyet csak egy tucat ember beszélt a legjobb esetben. A módszer igen hatékonynak bizonyult, mivel nem tudták visszafejteni a kódot az ellenfeleik.*

## *XOR alapú titkosítás*

*Ezen titkosítási algoritmus a Boole algebra XOR műveletének következő azonosságain alapul:*

*A XOR B = C  
C XOR B = A*

*Titkosításra ez úgy használható fel, hogy az A változó jelenti az adat egy titkosítandó bitjét, B a hozzá tartozó kulcs egy bitjét, C pedig a kimenetként kapott titkosított bitet.*

*Az algoritmus rém egyszerű, és megfelelő használattal megfelel a feltétlen biztonság tételének. Ez azt jelenti, hogy nem számít, hogy a támadónak mennyi ideje és erőforrása van, sosem fogja tudni kinyerni a kulcsból vagy a titkosított adatból az eredeti üzenetet.*

*A feltétel csupán annyi, hogy a kulcs ugyan olyan hosszú legyen, mint a titkosítandó adat, és véletlenszerűen generált legyen. Ha a kulcs rövidebb a titkosítandó adatnál, akkor a kimeneti bitsorozatban a kulcs ismétlődni kezd, ami némi erőfeszítéssel igen könnyen kinyerhető. Ebben az esetben a titkosítás egyáltalán nem biztonságos. Ugyanez a helyzet áll fent, ha a kulcs nem véletlenszerűen választott.*

*Hátránya a módszernek, hogy 10GB adat titkosítása ezzel a módszerrel 10GB kulcsot és 10GB titkosított adatot generál, amit célszerű külön helyeken tárolni.*

## *Playfair titkosítás*

*A Caesar titkosítás és általánosított formái esetén a relatíve nagy kulcstér sem nyújt elég védelmet. Ezt nagyjából az 1800-as években már tudták. Viszont akkor még nem voltak számítógépek, így az e fajta titkosítások ugyan megtörhetőek voltak, de jelentős időt vettek igénybe.*

*A problémát Charles Weastone oldotta meg 1854-ben, de a barátja, Playfair báró után kapta nevét az általa kifejlesztett algoritmus.  Ez a bemeneti adatot nem betűnként kezelte, hanem blokkokban. Több lépéses algoritmus, amelynek a működését egy példán keresztül lehet a legjobban szemléltetni:*

1. *5×5-ös mátrix rajzolása az ABC számára*
2. *ABC beírása, ez lesz a kulcs*
3. *Maradék hely kitöltése egyéb betűkkel*
4. *A példa kulcs:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P | A | L | M | E |
| R | S | T | O | N |
| B | C | D | F | G |
| H | I | K | Q | U |
| V | W | X | Y | Z |

1. *A titkosítandó szöveg felbontása két karakteres párokra.*
2. *J betűk cserélése I betűkre*
3. *Ha a két karakteres pár mindkét karakterje azonos, akkor kitöltő karaktert kell alkalmazni, ami: x*
4. *Ha a szöveg végén kell még betű a két karakteres párhoz, akkor Z betűt kell alkalmazni*
5. *Ezután szöveg titkosítása a következő szabályok betartásával:*
   1. *Ha két karakter ugyanabban a sorban szerepel az ABC táblában, akkor a karakter mellett jobbra található karakter lesz használva (körkörösen!).*
   2. *Ha két karakter ugyanabban az oszlopban szerepel, akkor az egyel alatta lévő karakter lesz használva.(körkörösen!).*
   3. *Ha a fenti két eset nem áll fent, akkor kereszt cserét kell csinálni.*

*A kereszt cserét egy példán keresztül a legkönnyebb elmagyarázni. A kulcs alapján a BD karakterpár a KH párra fordul, mivel a B betűhöz a D betű alatt lévő karaktert kell rendelni, a D betűhöz pedig a B alattit.*

*Abban az esetben ha a sor vagy oszlop egyezik, akkor a példa kulcs alapján az AM pár LE-re fordul, ME pedig EP-re. RH pár BV-re, a HV pedig VP-re.*

*A fenti szabályok ismeretében a következő szöveget titkosítom: Lord Granville’s letter*

*A karakterpárok bontás után a szöveg így néz ki: lo rd gr an vi lx le sl et te rz*

*A titkosítás után a kész üzenet: MT TB BN ES WH TL MP TA LN NV*

*A titkosított üzenet visszafejtése a kulcs ismeretében és a szabályok fordított alkalmazásával lehetséges.*

*A titkosítás kulcs tere 26 x 26 karakter, vagyis 676 karakter egy üzenet esetén. Valaha feltörhetetlen kódnak hitték, de mint kiderült, fel lehet törni, mivel a módszer néhány betűismétlődést és összetett szót figyelmen kívül hagy. Széles körben használt volt az I. és II. világháború alatt az amerikai és brit hadsereg által.*

## *Polialfabetikus titkosítás*

*A polialfabetikus titkosítás több ABC variációt alkalmaz, mondhatni a betű cserés algoritmusok továbbfejlesztett változatai az ebbe a családba tartozó titkosítások. A legegyszerűbb ilyen algoritmus a Vigenère titkosítás.*

*Tételezzük fel, hogy adott az összes Caesar ABC variáció az abc összes betűjéhez. Ezeket jelöljük a következőképpen: { Ca, Cb, Cc, …, Cz }  
A választott kulcs legyen a biztonsag szó. Az ABC variációk és a kulcs ismeretében a titkosított szöveg a következő formában fog előállni: Cb, Ci, Cz, Ct, Co, Cn, Cs, Ca , Cg. Ha a szöveg hosszabb, mint a kulcs, akkor a kulcsot ismételni kell.*

*Ezen módszer azért jó, mert minden betűhöz több titkosított betű tartozik, így a betűk ismétlési statisztikái majdhogynem használhatatlanok, viszont ez nem jelenti azt, hogy a módszer feltörhetetlen.*

*A feltörés menete:*

1. *Ki kell találni a kulcs hosszúságát*
2. *Ha a kulcs hosszúsága N, akkor az algoritmus N Caesar titkosítást használ. K pozícióban lévő és N+k, 2N+k, valamint 3N+k betűk ugyanazzal a módszerrel vannak titkosítva.*
3. *A különálló Caesar kulcsok törése a Caesar algoritmusnál említett módon lehetséges.*

*Gyakorlatban ezt a titkosítást elektromechanikus titkosító gépekkel valósították meg. Ilyen volt a németek által a második világháborúban alkalmazott Enigma gép is. Ennek a hadi változata 10 elektromechanikus tárcsát alkalmazott, ami 10 karakteres kulcsot jelentett. Ebben az esetben egy üzenet esetén 26^10 kulcs kombináció lehetséges, ez másképpen leírva 141 167 095 653 376 kombináció.*

*[](https://www.webmaster442.hu/wp-content/uploads/2014/09/enigma.jpg)Az 1940-es években ez akkora kombinációnak számított, hogy az algoritmust feltörhetetlennek nevezték. Azonban sikerült feltörni, az említett feltörési módszerrel és némi gépi erővel. A gépi erőt nevezték Turing bombának, amely nevét Alan Turing matematikusról kapta. Többek között az ő nevéhez fűződik a Turing gép, ami a mai számítástechnika alapját képezi.*

## *Eredmény titkosítások*

*Az eredmény titkosítások több egyszerűbb titkosító algoritmust használnak fel a kimenet előállításához. Ebből adódóan jóval nehezebben törhetőek fel az ilyen titkosítási módszerek. Felfoghatóak hídként a modern kriptográfia felé.*