# DEBRECENI EGYETEM INFORMATIKAI KAR

## SZAKDOLGOZAT Kriptográfiai algoritmusok

Témavezető: Készítette:

Dr. Herendi Tamás Papp László

Egyetemi docens Programtervező informatikus BSc

DEBRECEN

2025

## Bevezető

A digitális korszakban az adatok biztonságos kezelése és védelme kiemelkedően fontos szerepet tölt be a magánélet, a gazdasági és állami intézmények működése során. Ennek alapját a kriptográfiai algoritmusok képezik, amelyek lehetővé teszik az adatok titkosítását, hitelesítését, illetve az adatok integritásának megőrzését. A kriptográfia napjainkra összetett tudományággá vált, mely magában foglalja a klasszikus, szimmetrikus és aszimmetrikus titkosítási módszereket, valamint a különféle biztonsági protokollokat.

A dolgozat célja, hogy átfogó képet nyújtson a kriptográfiai algoritmusok elméleti alapjairól, azok gyakorlati megvalósításáról és biztonsági értékeléséről. A kutatás során első lépésként különböző klasszikus és modern algoritmusokat fogok megvizsgálni, mint például a Caesar- és Vigenère-titkosításokat, az Advanced Encryption Standard (AES), az RSA és a Data Encryption Standard (DES) algoritmusokat. Ezek implementációját Python nyelven fogom végrehajtani, a tanulási folyamat során pedig megismerem a kapcsolódó matematikai fogalmakat, mint a kongruencia, az Euklideszi algoritmus, a Fermat-tétel, a kínai maradéktétel és a gyorshatványozás. Ezek az elméleti alapok nélkülözhetetlenek a kriptográfiai algoritmusok megértéséhez és elemzéséhez.

A továbbiakban a dolgozatban részletesen foglalkozom az algoritmusok biztonsági aspektusaival, összehasonlítva azok erősségeit és gyengeségeit különböző szempontok szerint. Ezen belül elemzem az algoritmusok aszimptotikus viselkedését, valamint alkalmazom a prímteszteket, például a Miller-Rabin és Fermat-próbákat, amelyek segítségével ellenőrizhető a használt számok prímségessége. A dolgozat második felében a modern blokk titkosítók, kriptográfiai hash függvények, digitális aláírások kerülnek bemutatásra, továbbá gyakorlati példákon keresztül szemléltetem azok alkalmazásait.

A szakdolgozat végső célja, hogy összegző képet adjon a kriptográfiai algoritmusok elméleti és gyakorlati oldaláról, hozzájárulva a témában való mélyebb megértéshez és a biztonságos információkezeléshez. Ezen keresztül remélem, hogy nemcsak az elméleti tudásomat bővítem, hanem gyakorlati készségeimet is fejlesztem, amelyek hasznosak lehetnek a jövőbeni szakmai tevékenységeim során.

## A krioptográfiai algoritmusok matematikai alapjai

#### Legnagyobb közös osztó

#### Legkisebb közös többszörös

#### Euklideszi algoritmus

#### Kongruencia, lineáris kongruenciák

a ≡ b (mod m)

a kongruens b-vel modulo m maradékosztályban, ha m | a - b

#### Szimultán kongruenciarendszer

#### Kínai maradéktétel

#### Euler-Fermat tétel

Euler-féle φ függvény: φ(n) = n π

#### Kis Fermat tétel

#### Gyorshatványozás

#### Affin Transzformáció

## Kriptográfiai algoritmusok

A kriptográfiai algoritmusokat három csoportba sorolhatjuk: Titkos kulcsú (szimmetrikus), nyilvános kulcsú (aszimmetrikus) és kulcs nélküli algoritmusok.

A szimmetrikus kriptográfia esetében a titkosításhoz és a visszafejtéshez ugyanazt a kulcsot használjuk, például az AES és a DES titkosítás esetén.

A szimmetrikus kriptográfia tárgyalásakor a Caesar és a Vigenére titkosítások meglehetősen egyszerűek, és a napjainkban rendelkezésre álló módszerekkel könnyen feltörhetők, szerepük inkább a kriptográfiai algoritmusok evolúciójának szemléltetésében van.

Az aszimmetrikus kriptográfiában a titkosítás egy publikus, a visszafejtés pedig egy privát kulccsal történik. Itt a nyilvános kulcsot bárki megkaphatja, a privát kulcsot viszont csak a jogosult fél ismeri. Erre a legismertebb példa az RSA titkosítás.

A kulcs nélküli algoritmusok közé tartoznak a hash függvények és a véletlenszám-generátorok.

### Szimmetrikus kriptográfia

#### Caesar titkosítás

A Caesar titkosítás egy egyszerű karakteralapú titkosítás, ahol minden karaktert eltolunk egy fix számú pozícióval.

#### Vigenére titkosítás

A Caesar titkosítás kiterjesztése, ahol a kulcs hosszabb és változó eltolásokat alkalmaz, így összetettebb szimmetrikus módszer.

#### AES titkosítás

https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

https://hu.wikipedia.org/wiki/Advanced\_Encryption\_Standard

https://www.reddit.com/r/cryptography/comments/1mx4qrz/why\_does\_aes\_not\_give\_multiple\_valid\_decryption/

Az Advanced Encryption Srandard (AES) az amerikai U.S. National Institute of Standards and Technology (NIST, Amerikai Nemzeti Szabványügyi és Technológiai Intézet) publikálta 2001-ben, egy általuk kiírt pályázatra küldte be a Rijmen - Daemen belga kriptológus páros. A módszer eredeti neve Rijndael algoritmus, AES-nek csak azt a változatot tekintjük, ahol a blokkméret 128 bit. 2002-ben az USA kormányának elsődleges titkosítási algoritmusává vált.

Az AES4 x4-es mátrixokat használ a titkosítás során, de a Rijndael algoritmus különböző blokkmérei esetén más a mátrixok mérete is.

A kulcsméret meghatározza, hogy hány átalakítási ciklus éri a bemeneti információt, míg eléri a végleges titkosított méretet.

* 128 bites kulcs esetén 10 ciklus
* 192 bites kulcs esetén 12 ciklus
* 256 bites kulcs esetén 14 ciklus

Az algoritmus lépései

Az inputot felosztjuk 16 darab 8 bites blokkra (16\*8=128 bit), ezt a 16 blokkot (16 bájtot) fogjuk a 4x4 -es mátrixban tárolni.

1. A kulcsütemezés (Key scheduling)

[Kidolgozandó kérdések]  
hogyan nem ad az AES különbözö visszafejtési eredményeket

- Claude shannon - unicity distance https://en.wikipedia.org/wiki/Unicity\_distance  
- láthatatlan szalamandra https://eprint.iacr.org/2019/016.pdf

#### DES titkosítás

### Aszimmetrikus kriptográfia

#### RSA algoritmus

Az RSA algoritmus egy széles körben használt eljárás az adatok és üzenetek titkosítására. A nevét az algoritmust megalkotó matematikusokról kapta: Ron **R**ivest, Adi **S**hamir és Leonard **A**dleman. A három matematikus 1977-ben publikálta az RSA-t.

Az RSA a nagy számok faktorizációjának nehézségén alapul, emiatt kellően biztonságosnak bizonyult.

#### Blokk Titkosítók

#### HASH Függvények

https://csrc.nist.gov/projects/hash-functions

#### Digitális aláírások