A mérések végzéséhez az alkalmazásomból elmentett adatokat fogom tesztadatként használni. Az adatokat struktúrált xml dokumentum formátumban kerülnek tárolásra. Három különböző méretű fájlt vizsgálunk. Reális méretű fájlok használatára törekedtem (~500KB, ~2MB ~4MB). Már a 4 MB nagyság (nem titkosított) elérését a program rendeltetésszerű használata esetén is elég valószínűtlen tartom, mivel nagyon sok random adatot kellett bemásoljak ahhoz, hogy elérjem ezt a fájlméretet.

Feltehető a mérési eredmények arányos növekedése nagyobb fájlméret esetén.

A mérések összeállítása előtti próbálgatásaimból megtudtam, hogy a titkosítási és visszafejtési idő (a program kód alapján, a System.currentTimeMillis() parancs használatával) nem mindig egyezik meg ugyanazon fájl többszöri titkosítása esetén. Ebből következik, hogy egy adott fájl titkosítását mindegyik algoritmus használatakor többször is el kell végeznem, hogy egy korrekt közelítő értéket kapjak.

A titkosítási algoritmusok alkalmazására javax.crypto kriptográfiai könyvtárat használtam.

Korábban bemutatott szimmetrikus titkosítási algoritmusok összehasonlítása:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AES | DES | Triple DES | Blowfish |
| Lehetséges kulcs méret. | 128/192/256 | 64 (56-ot használ) | Összesen 168 bit | Mérete 32 bittől 448 bit-ig terjedhet |
| Block vagy stream cipher? | Block | Block | Block | Block |
| Ha block, mekkora egy block mérete? | 128 bit | 64 bit | 64 bit | 64 bit |
| Matematikai algoritmus neve. | Rijndael cipher | Feistel cipher | Feistel cipher | Feistel cipher |
| Köröket végez az algoritmus? | Igen | Igen | Igen | Igen |
| Ha igen, akkor mennyit? | Kulcs mérettől függ  128 bit – 10 kör  192 bit – 12 kör  256 bit – 14 kör | 16 kör | 3 x 16 kör | 16 kör |
| Ha igen, külön kulcsot használ-e minden körben? | Igen, mindegyik kör más kulcsot használ. | Igen, különböző 48 bites kulcsokat. | Három különböző DES kulcs. | Igen, 32 bit nagyságúakat. |
| Sikerült-e már feltörni? | Nem. | Igen, úgynevezett brute force támadással. | Vegyes válaszokat találtam, de mivel már nem igazán használják csak régebbi szoftverek, és mivel a DES is fel lett törve, ezért valószínűleg **igen**. | Nem. |

**AES** (ms – milliszekundum)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **500KB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 49 ms | 44 ms | 46 ms | 43 ms | 48 ms | **46 ms** |
| Visszafejtés | 7 ms | 7 ms | 6 ms | 7 ms | 7 ms | **6,8 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 52 ms | 58 ms | 51 ms | 56 ms | 51 ms | **53,6 ms** |
| Visszafejtés | 18 ms | 17 ms | 17 ms | 17 ms | 17 ms | **17,2 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **4MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 66 ms | 63 ms | 60 ms | 56 ms | 58 ms | **60,6 ms** |
| Visszafejtés | 23 ms | 23 ms | 22 ms | 23 ms | 22 ms | **22,6 ms** |

**DES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **500KB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 51 ms | 50 ms | 50 ms | 50 ms | 56 ms | **51,4 ms** |
| Visszafejtés | 17 ms | 21 ms | 17 ms | 16 ms | 17 ms | **17,6 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 86 ms | 85 ms | 90 ms | 86 ms | 85 ms | **86,4 ms** |
| Visszafejtés | 48 ms | 47 ms | 48 ms | 48 ms | 48 ms | **47,8 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **4MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 135 ms | 115 ms | 137 ms | 135 ms | 131 ms | **130,6 ms** |
| Visszafejtés | 84 ms | 85 ms | 83 ms | 83 ms | 83 ms | **83,6 ms** |

**Triple DES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **500KB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 73 ms | 69 ms | 70 ms | 74 ms | 71 ms | **71,4 ms** |
| Visszafejtés | 28 ms | 28 ms | 29 ms | 35 ms | 28 ms | **29,6 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 149 ms | 148 ms | 157 ms | 149 ms | 145 ms | **149,6 ms** |
| Visszafejtés | 109 ms | 110 ms | 111 ms | 110 ms | 110 ms | **110 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **4MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 258 ms | 254 ms | 258 ms | 255 ms | 303 ms | **265,6 ms** |
| Visszafejtés | 212 ms | 213 ms | 213 ms | 213 ms | 262 ms | **222,6 ms** |

**Blowfish**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **500KB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 48 ms | 44 ms | 49 ms | 49 ms | 46 ms | **47,2 ms** |
| Visszafejtés | 14 ms | 13 ms | 12 ms | 13 ms | 13 ms | **13 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 71 ms | 68 ms | 65 ms | 66 ms | 71 ms | **68,2 ms** |
| Visszafejtés | 32 ms | 33 ms | 32 ms | 32 ms | 32 ms | **32,2 ms** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **4MB** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| Titkosítás | 93 ms | 95 ms | 96 ms | 93 ms | 94 ms | **94,2 ms** |
| Visszafejtés | 51 ms | 53 ms | 52 ms | 50 ms | 40 ms | **49,2 ms** |

A fájl mérete titkosítás előtt és után minden esetben megegyezett, kivéve pár bájt különbséggel (500KB – 1bájt, 2MB – 4bájt, 4MB – 8bájt).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Titkosítás** | AES | DES | Triple DES | Blowfish |
| 500KB | 46 ms | 51,4 ms | 71,4 ms | 47,2 ms |
| 2MB | 53,6 ms | 86,4 ms | 149,6 ms | 68,2 ms |
| 4MB | 60,6 ms | 130,6 ms | 265,6 ms | 94,2 ms |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Visszafejtés** | AES | DES | Triple DES | Blowfish |
| 500KB | 6,8 ms | 17,6 ms | 29,6 ms | 13 ms |
| 2MB | 17,2 ms | 47,8 ms | 110 ms | 32,2 ms |
| 4MB | 22,6 ms | 83,6 ms | 222,6 ms | 49,2 ms |

A táblázat a mért átlagértékeket tartalmazza.

Volt néhány esetben viszonylag nagy eltérés az értékek között, például a Triple DES esetében, a 4MB méret 5.mérése ~50 milliszekundummal nagyobb mint a másik 4 mérés. Pontosabb átlagértékeket kaphatnánk, ha mondjuk az egyik módszerrel X méretű fájlt mondjuk százszor titkosítanánk, nem pedig csak ötször, és az így kapott eredményeket átlagolnánk.

Ettől függetlenül az így kapott értékekről lehet következtetéseket vonni.

Mindkét téren (titkosítás és visszafejtés) az AES használata bizonyult a leggyorsabbnak. Ahogy a fájl mérete kettőről négy megabájtra nőtt, úgy az AES titkosítási ideje nem nőtt olyan látványosan, mint a többi algoritmus esetében. Visszafejtési időre szintén igaz ez az állítás.

Gyorsaságot tekintve legközelebb az AES-hez a Blowfish állt, mind visszafejtés és titkosítás terén is.

A DES és a Triple DES közötti egyre inkább növekedő különbség várható volt, hiszen a Triple DES háromszor végzi el azt, amit a DES csak egyszer. Ennek ellenére titkosítási időben nem érte el a Triple DES az elődje háromszorosát, habár nagyobb fájlméret esetén a növekedésüket nézve valószínűleg el fogja, sőt ezt a tendenciát követve valószínűleg a különbség több, mint a háromszorosára is nőhet. Visszafejtési időben sokkal inkább látszik ez a különbség, 4 MB-os fájl titkosítása kétszer addig tart a Triple DES-nek, mint a simának, visszafejtést nézve 4 MB esetén ez az érték majdnem a háromszorosára nőtt.

Aszimmetrikus titkosítási algoritmusok:

**RSA**

Szerettem volna a szimmetrikus algoritmusokkal összehasonlítani, de nem lenne korrekt, mivel az adatmennyiség felsőhatára, amit a javax.crypto könyvtár használatával titkosítani tudtam, az 245 byte volt. Ha nagyobb mérettel próbálkoztam ugyanazt a hibaüzenetet kaptam „Data must not be longer than 245 bytes”.

Ezután az interneten jobban utánaolvastam a problémának és megtudtam, hogy az RSA algoritmus maximum akkora mennyiségű adatot képes titkosítani, mint az RSA kulcs mérete (mínusz az ún. header data, azaz fejlécadat).

Kettő hatványait használtam kulcsméret megadására, először 2048 byte nagyságú kulccsal kezdtem. A következő mérések ezzel készültek:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **64 byte** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| **Titkosítás** | 28 ms | 28 ms | 25 ms | 23 ms | 24 ms | **26,5 ms** |
| **Visszafejtés** | 7 ms | 6 ms | 6 ms | 4 ms | 6 ms | **6,3 ms** |
|  | 6.mérés | 7.mérés | 8.mérés | 9.mérés | 10.mérés |  |
| **Titkosítás** | 26 ms | 27 ms | 30 ms | 29 ms | 25 ms |  |
| **Visszafejtés** | 7 ms | 5 ms | 7 ms | 8 ms | 7 ms |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **128 byte** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| **Titkosítás** | 24 ms | 24 ms | 26 ms | 26 ms | 25 ms | **25,1 ms** |
| **Visszafejtés** | 6 ms | 5 ms | 7 ms | 7 ms | 7 ms | **6,3 ms** |
|  | 6.mérés | 7.mérés | 8.mérés | 9.mérés | 10.mérés |  |
| **Titkosítás** | 21 ms | 23 ms | 28 ms | 26 ms | 28 ms |  |
| **Visszafejtés** | 4 ms | 7 ms | 7 ms | 5 ms | 8 ms |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **245 byte** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| **Titkosítás** | 28 ms | 30 ms | 30 ms | 28 ms | 28 ms | **25,9 ms** |
| **Visszafejtés** | 7 ms | 4 ms | 5 ms | 7 ms | 6 ms | **6 ms** |
|  | 6.mérés | 7.mérés | 8.mérés | 9.mérés | 10.mérés |  |
| **Titkosítás** | 25 ms | 22 ms | 22 ms | 21 ms | 25 ms |  |
| **Visszafejtés** | 4 ms | 9 ms | 5 ms | 7 ms | 6 ms |  |

Ezután megpróbáltam nagyobb fájlokat titkosítani, így növelnem kellett a kulcs nagyságát is.

A kulcs nagysága: 8192 byte. Titkosítandó fájl nagysága: 1013 byte (a fejlécadat nagysága miatt ez lett a limit, de a kettőt összeadva egyébként 2^10-en, azaz 1024 byte). Titkosított fájl mérete: 1024 byte.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1013 byte** | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| **Titkosítás** | 28 ms | 21 ms | 22 ms | 31 ms | 29 ms | 25,6 ms |
| **Visszafejtés** | 117 ms | 123 ms | 121 ms | 121 ms | 125 ms | 121,2 ms |
|  | 6.mérés | 7.mérés | 8.mérés | 9.mérés | 10.mérés |  |
| **Titkosítás** | 23 ms | 24 ms | 27 ms | 28 ms | 23 ms |  |
| **Visszafejtés** | 119 ms | 122 ms | 121 ms | 119 ms | 124 ms |  |

Annak ellenére, hogy ezek az értékek nem tűnnek nagynak, a számítógépemnek elég sokáig tartott minden mérés elvégzése, volt néhol 10 másodperc is, mire az eredményt megkaptam. Az említett ’System.currentTimeMillis()’ metódus használatát ajánlották az interneten kód futásidejének mérésére, így nem tudom mi lehetett a probléma, de mivel már az 1 kb-os fájl titkosítása eddig tartott, nagyobb fájlmérettel nem próbálkoztam.

----TEMPLATE-----

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| **Titkosítás** |  |  |  |  |  |  |
| **Visszafejtés** |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1.mérés | 2.mérés | 3.mérés | 4.mérés | 5.mérés | Átlag |
| **Titkosítás** |  |  |  |  |  |  |
| **Visszafejtés** |  |  |  |  |  |  |
|  | 6.mérés | 7.mérés | 8.mérés | 9.mérés | 10.mérés |  |
| **Titkosítás** |  |  |  |  |  |  |
| **Visszafejtés** |  |  |  |  |  |  |

/\*

\* Írd le, hogy a kulcs titkosítási algoritmusa és maga a szövegé ugyanaz kellett hogy legyen

\*egyetértessz-e vagy nem a dokumentumban leírtakkal? teljesítményi idő nagy, stb?

\*/