BÁNKI KÖZLEMÉNYEK --- BÁNKI REPORTS
VII. ÉVFOLYAM 2. SZÁM --- VOLUME VII NO. 2

Többképkockás videószteganográfia

Multiframe Video Steganography

Kovács Gábor, Simon-Nagy Gábor
Óbudai Egyetem, Budapest, Magyarország
kovacs.gabor@nik.uni-obuda.hu, nagy.gabor@nik.uni-obuda.hu

***Összefoglalás* — Az innovatív titkosítási rendszerkétlépcsős megközelítést alkalmaz az adatok maximális védelme érdekében. Az első lépésben egy modern, megbízható titkosítási algoritmus kódolja az érzékeny információt, biztosítva annak dekódolhatatlanságát még felfedezés esetén is. A második lépésben az így titkosított adat láthatatlanul beágyazódik hétköznapi videók képkockáiba, elkerülve a hagyományos titkosításra utaló nyomokat. A rendszer pixelenkénti beágyazást és képkockák közötti különbségeket használ, így az információ emberi szem számára láthatatlan marad, tovább növelve a biztonságot és visszafejthetetlenséget.**

***Kulcsszavak*:** Szteganográfia, Titkosítás, Képkocka-különbség, LSB technika, XOR művelet

***Abstract* — The innovative encryption system employs a two-step approach to ensure maximum data protection. In the first step, a modern, reliable encryption algorithm encodes sensitive data, ensuring that it remains indecipherable even if discovered. In the second step, the encrypted data is inperceptibly embedded into everyday video frames, avoiding any traces typically associated with conventional encryption methods. The system utilizes pixel-level embedding and inter-frame difference techniques, making the hidden information invisible to the human eye, further enhancing security and resistance to unauthorized decryption.**

***Keywords*:** Steganography, Encryption, Frame Difference, Least Significant Bit Technique, XOR Operation

1 BEVEZETÉS

A szteganográfia az információ elrejtésének művészete, amelynek célja, hogy a titkos üzenet létezése ne legyen észlelhető. A módszer története több ezer évre nyúlik vissza, és számos kreatív technikát alkalmaztak az idők során.

1.1 Ókori eredetek

A szteganográfia első ismert alkalmazásai az ókori Görögországban jelentek meg. Hérodotosz, az ókori görög történetíró, egy különleges esetet jegyzett fel: Histiaeus, egy görög uralkodó, titkos üzenetet küldött szövetségeseinek úgy, hogy azt egy rabszolga fejbőrére tetoválták. Miután a rabszolga haja visszanőtt, elindult a címzetthez, ahol ismét leborotválták a fejét, így az üzenet ismét láthatóvá vált.

Egy másik ókori görög módszer a viaszbevonatos táblák használata volt. A titkos üzenetet először a fatáblára írták, majd viasszal bevonták, így a felületen látszólag üresnek tűnt. A címzett a viaszt lekaparva hozzáférhetett az elrejtett szöveghez.

1.2 Középkori és reneszánsz technikák

A középkorban és a reneszánsz idején a szteganográfia tovább fejlődött. Johannes Trithemius, egy német szerzetes és tudós, 1499-ben írta meg *Steganographia* című művét, amely a titkos kommunikáció módszereit tárgyalta. Bár a könyv elsőre okkult szövegnek tűnt, valójában kriptográfiai és szteganográfiai technikákat tartalmazott.

A láthatatlan tinta használata szintén elterjedt ebben az időszakban. A citromlével, ecettel vagy más vegyi anyagokkal írt üzenetek csak hő hatására váltak láthatóvá, így a titkos információ könnyen elrejthető volt.

1.3 Napjainkban

Az információbiztonság napjainkra kiemelt jelentőségűvé vált, hiszen az adatok sértetlensége és bizalmas kezelése alapvető követelmény a digitális világban. Az internetes kommunikáció és adattárolás rohamos fejlődésével párhuzamosan egyre kifinomultabb módszerekre van szükség az illetéktelen hozzáférés és adatlopás megakadályozására. A hagyományos titkosítási eljárások hatékonyak az adatok védelmében, azonban gyakran árulkodó jeleket hagynak maguk után, amelyek felhívhatják a figyelmet a titkosított tartalom jelenlétére. A szteganográfia ezen problémát küszöböli ki azáltal, hogy az információt olyan módon rejti el egy hétköznapi fájlban, hogy annak létezése észrevétlen marad.

A digitális szteganográfia több formában alkalmazható, például képekben, hangfájlokban vagy videókban történő rejtett adattárolás révén. A videószteganográfia különösen érdekes terület, mivel a mozgókép sajátosságai lehetőséget teremtenek nagy mennyiségű adat észrevétlen elhelyezésére. A képkockák természetes változásai miatt a vizuális módosítások nehezen észlelhetők, így az ilyen eljárások különösen hatékonyak lehetnek az adatok diszkrét tárolására.

Jelen kutatásban egy kétlépcsős videószteganográfiai módszert mutatunk be, amely ötvözi a modern kriptográfiai titkosítást és a *Least Significant Bit* (LSB) beágyazási technikát. Az eljárás első lépésében a rejtendő üzenetet ElGamal vagy RSA titkosítással kódoljuk, biztosítva ezzel az adatok erős matematikai védelmét. Ez a titkosítási forma garantálja, hogy az üzenet akkor is visszafejthetetlen maradjon, ha valaki észleli annak jelenlétét, mivel az aszimmetrikus titkosítás nyilvános és privát kulcsokat használ, ezáltal a dekódolás kizárólag az arra jogosult felek számára lehetséges.

A második lépésben a már titkosított üzenetet videószteganográfiai technikával rejtem el két egymást

követő képkocka különbségén alapuló LSB eljárás segítségével. Ez a technika lehetővé teszi az adatok diszkrét beágyazását az egyes képpontok legkisebb bitjeinek módosításával, míg a képkockák közötti eltérés használata további vizuális álcázást biztosít. Az emberi szem számára szinte észrevehetetlen módosítások révén az adatelrejtés hatékony és biztonságos módon valósul meg, mivel a képkockák természetes zaját kihasználva az információ még kevésbé detektálható.

Az alkalmazott kétlépcsős rendszer jelentős előnyt jelent a hagyományos szteganográfiai módszerekhez képest, hiszen nemcsak az információ rejtett tárolását biztosítja, hanem annak visszafejthetetlenségét is garantálja az előzetes titkosítás révén. Ez különösen fontos olyan környezetekben, ahol az információ védelme kiemelt szerepet játszik, például kormányzati, katonai vagy pénzügyi alkalmazások esetében, valamint olyan személyes adattárolás során, ahol a diszkréció kulcsfontosságú.

A cikkben részletesen ismertetjük a kétlépcsős videószteganográfia működését, technikai aspektusait és gyakorlati alkalmazási lehetőségeit, elemezzük az eljárás hatékonyságát különböző videótípusok esetén, valamint megvizsgáljuk a szteganográfiai támadásokkal szembeni védelmi stratégiákat. Különös figyelmet fordítunk a módszer robusztusságára, illetve arra, hogy miként csökkenthető az észlelhetőség kockázata modern szteganalízis eszközökkel szemben.

A kutatás célja, hogy új perspektívát kínáljon az információbiztonság területén, ötvözve a kriptográfiai titkosítás erejét a hatékony észlelhetetlenségi módszerekkel, ezzel hozzájárulva a szteganográfiai megoldások fejlődéséhez a jövőben.

2 Problémafelvetés

A szteganográfia olyan terület, amely az információ rejtett módon történő tárolásával foglalkozik. Az elmúlt évtizedekben a digitális világban egyre inkább előtérbe került, és számos alkalmazási területen fordul elő. Ebben a fejezetben megvizsgáljuk a szteganográfia kihívásait és lehetőségeit.

2.1 Titkos üzenetek rejtése

A szteganográfia célja, hogy olyan adatokat rejtsünk el más adatokban (például képekben, videókban vagy hangfájlokban), amelyek észrevétlenek a külső szemlélő számára.

A kihívás abban rejlik, hogy a rejtett üzeneteket hatékonyan kell elhelyezni a hordozófájlokban, hogy azok rejtve maradjanak.

2.2 Minőség és biztonság egyensúlya

A szteganográfia során a hordozófájlok minősége és az elrejtés biztonsága között egyensúlyt kell találnunk. Minél nagyobb mennyiségű információt rejtünk el, annál inkább csökken a hordozófájl minősége és a titkosítás biztonsága. Ugyanakkor a magas minőség és a biztonság sem érhető el anélkül, hogy a hordozófájl kapacitását ne áldoznánk fel.

2.3 Két különböző megközelítés

Az egyik technikai megközelítés a hatékonyságon alapuló technikák, amely módszerek a titkos üzenetet úgy rejtik el, hogy minimalizálják a hordozófájl torzulását. Erre példa a hatékony beágyazás a képkockákba.

A másik kategóriába a statisztikai észlelhetőségen alapuló technikák tartoznak, melyek sorána titkos üzenetet olyan részekre rejtjük, amelyeket egy előre definiált torzítási függvény határoz meg, például textúrált vagy zajos területekre.

3 Technológiák

3.1 LSB

Az LSB szteganográfia egy gyakran alkalmazott technika, amely digitális képekbe vagy hangfájlokba ágyaz be üzeneteket. Az eljárás lényege, hogy az üzenetet a kép vagy hang legkisebb helyiértékű bitjeibe (LSB – *Least Significant Bit*) rejti. Az LSB módszer előnye, hogy az üzenet észrevétlenül marad a hordozóban. Ha megvan az eredeti kép, akkor könnyen észrevehető, de akkor is csak szoftveres elemzés segítségével válik bizonyossá.

3.1.1 Működése

Képek eseténaz üzenetet a kép pixeleinek legkevésbé jelentős bitjeibe rejtjük. Például, ha egy kép egy adott pixelének vörös komponense 10101010, akkor az LSB módszerrel az utolsó bitet megváltoztathatjuk anélkül, hogy a kép láthatóan megváltozna. A táblázatban az utolsó 2 bit kerül eltávolításra, ezek helyére lehet az információt elrejteni.

1. táblázat: Maszkolás

Bitérték	128	64	32	16	8	4	2	1
Eredeti	1	0	1	0	1	0	1	0
Maszk	1	1	1	1	1	1	0	0
Eredmény	1	0	1	0	1	0	0	0

3.1.2 Hátrányok és korlátok

Az LSB módszer meglehetősen ismert, így már nem alkalmas biztonságos adatrejtésre. Emellett napjainkban inkább a tömörített formátumok (pl. JPG) örvendenek népszerűségnek, de itt is van lehetőség az adat elrejtésére.

Nagy adatmennyiség elrejtéséhez jelentős felbontású kép szükséges, hiszen bájtonként csak 1 vagy 2 bitet használhatunk adatrejtésre. Videók estében ez kisebb probléma, hiszen 24, 30 vagy akár 60 képkocka is rendelkezésre áll egy másodpercben, így ebben az esetben jobban kihasználható.

3.2 XOR-alapú LSB

Egy megadott videó fájlba úgy is elrejthető a titkos adat, hogy nem külön-külön használjuk fel a képkockákat, hanem azokat párba állítjuk, és a párok segítségével rejtjük el az üzenetet.

3.2.1 Működése videófájl esetén

Miután a videót beolvastuk, az algoritmus első lépése a lépkockák különbségének meghatározása. Ennek során a program először egy-egy képkockát párosít. Az első és a második képkockán XOR- műveletet hajtunk végre minden pixelértéken. Az XOR- művelet (exkluzív „vagy” művelet) két bináris adat között végzett logikai művelet, amelynek eredménye 1, ha a két bit különböző, és 0, ha azonos értékűek.

Az így kapott XOR-eredményen LSB-módszer segítségével elrejtjük az üzenetet. Az LSB-művelet során az üzenet bitjeit a képkockák legkisebb helyiértékű bitjeibe ágyazzuk be. Ez a módszer biztosítja, hogy a beágyazott

üzenet nem észlelhető szabad szemmel, és a képkockák látszólag változatlanok maradnak.

A második képkockát úgy kapjuk meg, hogy az első képkockán és az XOR-eredményen újabb XOR-műveletet hajtunk végre. Ezáltal a második képkocka pontosan rekonstruálható az első képkocka és a beágyazott üzenet segítségével, ami biztosítja az eljárás veszteségmentességét.

Az elrejtett üzenetet a második és az első képkocka XOR-műveletével lehet visszanyerni. Az eredményül kapott adatból az LSB-művelet segítségével kinyerhetjük az eredetileg elrejtett üzenetet. Ez a folyamat szintén veszteségmentes, mivel az XOR-műveletek és az LSB-módszer garantálják, hogy a teljes információ visszanyerhető.

Mivel a két képkockát nem csak adott távolságra vehetjük fel, így a két képkocka közti különbséget is eltárolhatjuk, mint egy kulcsot, ami megmutatja, hogy egy adott képkockát melyik másikkal kell párba állítani. Ezt a folyamatot nevezzük kulcsgenerálásnak.

Egy lehetséges továbbfejlesztési irány, hogy egy vágáson belül több képkocka párt is meghatározunk, ebben az esetben viszont figyelembe kell venni, hogy lehetnek ütközések is. Emiatt a végrehajtási sorrend ebben az esetben megfordul, és az üzenetet nem elölről, hanem hátulról tudjuk visszafejteni.

4 Fájltípus

Az FFV1 (FF Video Codec 1) egy veszteségmentes videókodek, amelyet az FFmpeg projekt keretében fejlesztettek ki. Célja a digitális videók hatékony tömörítése, miközben megőrzi az eredeti képminőséget. Az FFV1 formátumot különösen az archívumi és digitális megőrzési környezetekben használják, ahol a minőség és a megbízhatóság kiemelt fontosságú.

4.1 Veszteségmentes tömörítés

Az FFV1 formátum egyik legfontosabb jellemzője a veszteségmentes tömörítés. Ez azt jelenti, hogy a tömörített videó dekódolása után az eredeti képminőség veszteség nélkül visszaállítható. Ezt a tulajdonságot különösen értékelik azok a szakemberek, akik digitális archívumokat kezelnek; ez ugyanis garantálja az eredeti anyag integritását.

4.2 Hatékony tömörítési algoritmus

Az FFV1 tömörítési algoritmusa hatékonyan csökkenti a videófájlok méretét, mindemellett megőrzi a veszteségmentességet. Az algoritmus többféle módszert alkalmaz, beleértve a prediktív és a kontextusalapú aritmetikai kódolást, amelyek segítségével optimalizálja a tömörítési folyamatot.

4.3 Rugalmasság és kompatibilitás

Az FFV1 formátumot úgy tervezték, hogy széles körben kompatibilis legyen különböző rendszerekkel és szoftverekkel. A kodek része a nyílt forráskódú FFmpeg projektnek, amely biztosítja a kiterjedt támogatást és egyszerű integrációt különböző alkalmazásokban.

4.4 Transzparens színkezelés

Az FFV1 formátum támogatja a többféle színteret és bitmélységet, beleértve a 8 és 16 bites színmélységeket. Ez lehetővé teszi a formátum használatát különböző színkezelési követelményekkel rendelkező projektekben, biztosítva a színek pontos és torzításmentes megőrzését.

5 Többszintű titkosítás

A szteganográfia az információ elrejtésére szolgál valamilyen médiumban, de nem biztosítja az adat titkosítását. Ez azt jelenti, hogy ha valaki rájön, hogy a médium rejtett információt tartalmaz, akkor hozzáférhet az üzenethez. Ezért célszerű először titkosítani az üzenetet, majd a titkosított üzenetet elrejteni a kiválasztott médiumban. Ezáltal még ha valaki rá is jön, hogy a médium rejtett információt tartalmaz, az üzenet továbbra is titkos marad.

5.1 ElGamal

Az ElGamal titkosítási rendszer a modern kriptográfia egyik alapvető eleme. Taher Elgamal1985-ben olyan rendszert hozott létre, amely a nyilvános kulcsú titkosítás és a digitális aláírások széles körben alkalmazott módszerévé vált. Az ElGamal titkosítás a Diffie–Hellman-kulcscsere protokollon alapul, és a diszkrét logaritmus probléma nehézségén alapuló biztonságát használja fel.

5.1.1 Működése

Kulcsgenerálás:

Generál egy ciklikus csoportot renddel, generátorral.

Választ egy egész számot véletlenszerűen.

Kiszámítja a -t. A nyilvános kulcs az értékekből áll. Alice közzéteszi ezt a nyilvános kulcsot, és megtartja a-t mint privát kulcsot, amit titokban kell tartani.

Titkosítás:

Leképezi az üzenetet egy elemre egy visszafordítható leképező funkció segítségével.

Választ egy egész számot véletlenszerűen.

Kiszámítja a -t. Ez a megosztott titok.

Kiszámítja a -t.

Kiszámítja a -t. Bob elküldi a titkosított üzenetet Alicenak.

Visszafejtés:

Kiszámítja a -t. Mivel, ezért ugyanaz a megosztott titok, amit Bob használt a titkosításkor.

Kiszámítja a -t, ami a csoportban. Ezt többféleképpen is ki lehet számolni. Ha egy multiplikatív csoport részcsoportja, ahol prím, akkor a moduláris multiplikatív inverzet kiszámítható az kiterjesztett euklideszi algoritmussal. Egy alternatíva a -t kiszámítása, ami az inverze a Lagrange-tétel miatt.

5.2 RSA-agoritmus

Az RSA- (Rivest–Shamir–Adleman-) algoritmus egy széles körben használt nyilvános kulcsú kriptográfiai algoritmus, amely biztonságos adatátvitelt tesz lehetővé. Az algoritmust Ronald Rivest, Adi Shamir és Leonard Adleman fejlesztette ki 1977-ben. Az RSA-algoritmus alapja a számelmélet és a nagy prímszámok faktorizációjának nehézsége.

5.2.1 Az algoritmus lépései

Prímszámok kiválasztása:

Két nagy prímszámot, p és q, választunk. Ezeket titokban kell tartani, mivel ezek a privát kulcs alapját képezik.

Szorzat kiszámítása:

Kiszámítjuk n-t, amely a két prímszám szorzata:

$$n = p \times q \quad (1)$$

Euler-féle ϕ-függvény:

Kiszámítjuk az Euler-féle ϕ-függvényt:

$$\phi(n) = (p-1) \times (q-1) \quad (2)$$

Nyilvános és privát kulcsok:

Választunk egy nyilvános kulcsot, e-t, amely egy olyan egész szám, amely 1 és $\phi(n)$ között van, és relatív prím $\phi(n)$-hez képest, vagyis legnagyobb közös osztójuk (lnko) 1. Képlettel:

$$\text{lnko}\big(e, \phi(n)\big) = 1 \quad (3)$$

Kiszámítjuk a privát kulcsot, d-t, amely az e modulo $\phi(n)$ multiplikatív inverze, vagyis úgy, hogy az $e \times d$ és 1 közötti kongruencia teljesüljön. Ez azt jelenti, hogy $\phi(n)$-nel osztva mindkét oldal maradéka megegyezik (számszerűleg 1):

$$e \times d \equiv 1 \ \big(\text{mod}\phi(n)\big) \quad (4)$$

Nyilvános és privát kulcspárok:

A nyilvános kulcs a (e, n) pár, míg a privát kulcs a (d, n) pár.

Titkosítás:

Az üzenetet M-ként értelmezzük, ahol M egy olyan egész szám, hogy

$$0 \leq \text{M} < \text{n}. \quad (5)$$

A titkosított üzenet, C, a következőképpen számítható ki (C az M^e egész szám n szerinti maradéka):

$$C = M^e \bmod n. \quad (6)$$

Visszafejtés:

A titkosított üzenet, C, visszafejtése a privát kulcs segítségével történik:

$$M = C^d \bmod n. \quad (7)$$

6 Feldolgozás

A videófájlok feldolgozása során különböző technikákat alkalmaztunk annak érdekében, hogy hatékonyan elemezzük, titkosítsuk és visszafejtsük az adatokat. Első lépésben megvizsgáltuk a videó metaadatait, például a képkockák számát, felbontását és FPS- (*Frame Per Second*, képkockasebesség) értékét, hogy pontosan meghatározhassuk a feldolgozási paramétereket. Ezután beolvastuk a videót képkockánként, amelyeket egy pufferben tároltunk, miközben a vágásokat detektáltuk. Ez a folyamat automatikusan azonosította a jelentős változásokat a képkockák között.

A képkockák feldolgozása során képpiramisokat használtunk, amelyek lehetővé tették a képkockák különböző méretű reprezentációinak létrehozását. A képpiramisok alkalmazása elősegítette a hatékony elemzést és feldolgozást, különösen a vágások detektálása során, mivel a piramisok segítségével könnyebben felismerhetők voltak a képkockák közötti különbségek alacsonyabb felbontásban. Ez nemcsak az elemzést gyorsította fel, hanem pontosabb detektálást is biztosított.

Az így esélyesnek tartott képkockákat ezután 4x4-es blokkokra bontotta az eljárás, majd össze is hasonlította a blokkok színvilágát, ezzel javítva az eredményt..

A vágások között véletlenszerűen kiválasztottunk két képkockát, amelyek további feldolgozásra kerültek. Ezt követően az LSB-szteganográfiai módszerrel elrejtettük az adatokat.

A visszafejtési folyamat során azonosítottuk a két képkockát a kulcs segítségével, majd ezeken elvégeztük a XOR-műveletet, az így megkapott képen hajtottuk végre az LSB-t visszafelé.

A teljes folyamat párhuzamosan futott több szálon, ahol a képkockák olvasását és az adatrejtést külön szálak kezelték, amelyeket biztonságosan szinkronizáltunk egy pufferrel és mutex segítségével.

A memóriahasználat optimalizálása miatt célszerű a fájlt kétszer beolvasni és csak darabokban vizsgálni, mivel az ilyen formátumú videófájlok nagy méretűek. Az első beolvasás során történik a vágások detektálása, majd a kulcs létrehozása, végül az ismételt beolvasáskor zajlik és azadatok rejtése az állományba.

6.1 GPU-alapú megközelítés

Ebben a megközelítésben az OpenCL (*Open Computing Language*) nevű nyílt, platformfüggetlen programozási

```
__kernel void xorLSBKernel(__global const uchar* frameA, __global uchar* frameB,
                           __global const uchar* data, int dataSize, int frameSize) {
    int id = get_global_id(0);

    if (id < frameSize) {
        uchar c = frameA[id] ^ frameB[id];

        if (id < dataSize) {
            c = (c & ~1) | (data[id] & 1);
        }

        frameB[id] = frameA[id] ^ c;
    }
}

__kernel void decodeXorLSBKernel(__global const uchar* frameA, __global const uchar* frameB,
                                 __global uchar* extractedData, int dataSize, int frameSize) {
    int id = get_global_id(0);

    if (id < frameSize) {
        uchar c = frameA[id] ^ frameB[id];

        if (id < dataSize) {
            extractedData[id] = c & 1;
```

1. ábra OpenCl kódrészlet

keretrendszer segítségével párhuzamosan futtatjuk a titkosítási és dekódolási (visszafejtési) műveleteket, kihasználva a modern GPU-k és számítási gyorsítók erejét. Az alkalmazott kód egy részletét az 1. ábra mutatja.

A titkosítás során két adatfolyamot kezelünk. Az egyik adatfolyam referenciaértékként szolgál, a másikat módosítjuk. Az XOR-művelet lehetővé teszi az információ elrejtését úgy, hogy az eredmény strukturálisan nem tér el jelentősen az eredetitől.

A folyamat során az adatok beágyazásához az LSB-technikát alkalmazzuk , azaz az adatok legkisebb helyiértékű bitjeit módosítjuk. Ez biztosítja, hogy az információ rejtve marad, miközben a vizuális vagy bináris adatok alig változnak. A visszafejtési művelet hasonló

logikával működik: a titkosított és az eredeti adatfolyamot XOR-művelettel kombinálva rekonstruálja az elrejtett információt, majd az LSB-k kiolvasásával visszanyeri az eredeti adatokat.

Az OpenCL által biztosított párhuzamos feldolgozás jelentősen gyorsítja ezt a műveletet, mivel egyszerre több szálon történik a számítás, így az XOR- és LSB-manipuláció hatékonyan és nagy méretű adathalmazokon is gyorsan végrehajtható.

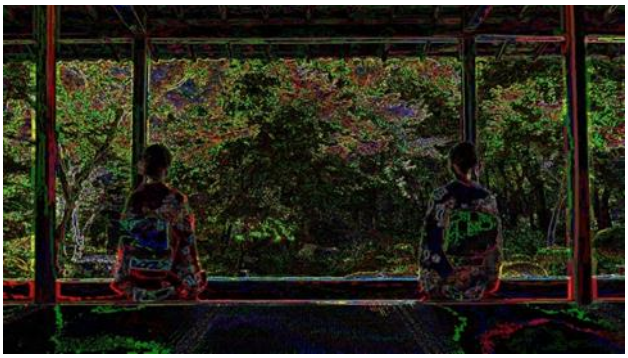
7 Eredmények

A képeken két képkockán végrehajtott műveletek figyelhetőek meg. A program kiválasztotta a két képkockát, amelyek a 2-es és a 3-as ábrán láthatóak. Ezt követően előállította a különbségképet (4. ábra), majd a második képet felülírta ennek segítségével, így állt elő az 5. ábra. Ez a módosított kép kerül vissza a videófájlba. Mivel a két kép minimálisan tér el, így a különbség is viszonylagosan kicsi, amin végrehajt a program egy LSB- műveletet. A különbség igencsak nehezen észrevehető, a visszafejtéshez már két képre van szükség.

2. ábra Eredeti első képkocka

3. ábra Eredeti második képkocka

4. ábra A két képkocka különbsége



5. ábra Módosított második képkocka

8 Összegzés

A bemutatott program egy innovatív, kétlépcsős titkosítási rendszerrel igyekszik garantálni az adatok maximális védelmét. Ez a rendszer a szteganográfia és az erős kriptográfia ötvözésével biztosítja az információ észrevétlen elrejtését és a visszafejtés elleni magas szintű védelmet.

Az első lépésben a rendszer nem csupán titkosítja, hanem láthatatlanná is teszi az érzékeny információkat, azáltal, hogy egy hétköznapi kép- vagy videófájlba rejti azokat. Míg a klasszikus titkosítási technológiák felismerhetők (például egy jelszóval védett fájl könnyen azonosítható), addig ez az eljárás nem kelt gyanút, hiszen a módosított videók a titkosítás után is hétköznapi módon megnyithatók. Csak akkor lehet megbizonyosodni, hogy változás történt, ha az eredeti médiummal össze lehet hasonlítani.

A második lépcsőben a rejtett információ egy napjainkban is elismert, megbízható titkosítási algoritmussal kerül titkosításra. Ez a kettős védelem biztosítja, hogy még ha valaki felfedezné is az adatrejtést, az információ dekódolása az erős titkosítás miatt továbbra sem lehetséges.

A rendszer többféle technikával képes az adatok integrálására:

- Pixel-alapú beágyazás, amely az egyes képpontok legkisebb helyiértékű bitjeit használja fel információ tárolására, biztosítva, hogy a módosítás vizuálisan nem érzékelhető marad.
- Több képkocka használata az adatrejtéshez, hogy a felfedezés után a másik képkockát is meg kelljen találni a feloldáshoz.

Ezen módszerek kombinációja lehetővé teszi, hogy az elrejtett adatok maximálisan védve legyenek, miközben az eredeti fájl működése és használhatósága változatlan marad. A program fejlesztése során kiemelt figyelmet kapott a lehető leggyorsabb végrehajtás és az üzenetek biztonságos elrejtése.

Ez az eljárás új szintre emeli az információbiztonságot. A hagyományos titkosítási technikák mellett egy nehezen észrevehető adatrejtési eljárást is alkalmaz, amely teljes diszkréciót tesz lehetővé. Legyen szó érzékeny személyes információkról vagy kritikus üzleti adatok titkosításáról, a rendszer lehetőséget kínál arra, hogy ezek észrevétlenül és biztonságosan továbbíthatók és tárolhatók legyenek.

IRODALOMJEGYZÉK

[1] „ffmpeg.org,” [Online]. Available: https://ffmpeg.org/~michael/ffv1.html. [Hozzáférés dátuma:. 2024.11.05.].

[2] Bamanga, M. A., Babando, A. K., Shehu, M. A. (2024). Recent Advances in Steganography. In: Mayer, J. (ed.). Steganography - The Art of Hiding Information. Intech Open. https://www.intechopen.com/chapters/1180100. [Hozzáférés dátuma: 2024.10.05.]

[3] Alanzy, M. et al. (2023). Image Steganography Using LSB and Hybrid Encryption Algorithms. Appl. Sci., 13(21), 11771. [Hozzáférés dátuma: 2024.10.30.]

[4] El Laz, M., Gregoire, B., Rezk, T. (2020). Security Analysis of ElGamal Implementations.. Proceedings of the 17th International Joint Conference on e-Business and Telecommunications - SECRYPT. SciTePress, pp. 310-321. [Hozzáférés dátuma: 2024.10.25.]

[5] R. Rivest, A. Shamir és L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems „people.csail.mit.edu,” 1997. [Online]. Available: https://people.csail.mit.edu/rivest/Rsapaper.pdf. [Hozzáférés dátuma: 25 10 2024].