Варненски свободен университет

"Черноризец Храбър"

Дипломна работа

На тема:

Криптиране на wav файлове с псевдослучайна последователност

За придобиване на образователно-квалификационна степен "Бакалавър"

ДИПЛОМАНТ:
Петър Иванов

НАУЧЕН РЪКОВОДИТЕЛ:
Проф. Борислав Стоянов

November 3, 2021

"*Малко математика може да постигне това, което всички оръжия и бодлива тел не могат: малко математика може да пази тайна.*"

Edward Snowden

Благодарности

Искам да благодаря на Проф. Борислав Стоянов за всичкото време, което инвестира в мен за да подготвим тази изключително интересна дипломна работа. Благодарение на него успях да науча много нови неща за криптографията и това ми събуди интереса към нея. Не съм и очаквал, че писането на дипломна работа ще бъде изключително интересно за мен и ще допринесе толкова много за моите знания.

Съдържание

Фигури

Таблици

Глава 1

Увод

В резултат на днешните бързо развиващи се технологии, необходимостта от компютърни мрежи се увеличи драстично. Повишението на използване на интернет комуникации се дължи на лесният достъп от всяка точка на света.След това размерът и честотата на данните, предавани чрез компютърни мрежи, се увеличават всеки ден. Сигурността е една от значимите нужди на мрежовите структури. Необходимо е да се защити информацията, която е изпратена и получена по време на предаването на данни, от външни фактори, които се опитват да прекъснат системата. Днес отделни лица и компании разпределят големи бюджети за поддържане на сигурността на данните. Всеки ден има все повече и повече злонамерени атаки на данните, а загубите в сигурността на данните непрекъснато се увеличават.[8]

Развитието на компютърно управляваните комуникационни мрежи е безпроблемен и евтин контакт между хора и компютри от противоположни страни на света, заменяйки повечето поща с телекомуникации. За много приложения тези контакти трябва да бъдат защитени срещу подслушване и инжектиране на нелегитимни съобщения. В момента обаче решаването на проблеми със сигурността изостават много от другите области на съответни технолопгии за комуникация. За съжаление съвременната криптография не

е в състояние да отговори на изискванията, при това нейното използване би наложило такива сериозни неудобства на системата от потребители, за да елиминират много от предимствата на телепроцесирането.[22]

Най-известният криптографски проблем е този на поверителността: предотвратяване на неоторизираното извличане на данни от комуникации по несигурен канал. За да се използва криптография за осигуряване на поверителност, обаче е нужно понастоящем да се сподели ключ страни. Това става чрез предварително изпращане на ключа по някой защитен канал, като напр. частен куриер или препоръчана поща. Частен разговор между двама души без предварително познаване-това е често срещано явление в бизнеса и е нереалистично да се очаква дългосрочно отлагане на първоначални бизнес контакти достатъчно, за да се предават ключове по някакъв друг физически начин. Разходите и забавянето, наложени от това ключово разпределение, установявя че проблемът е основна бариера за прехвърлянето на бизнес комуникации с големи мрежи за телеобработка.

Краткият Оксфордски речник (2006) определя криптографията като изкуството на писане или решаване на кодове. Това определение може да е исторически точно, но то не успява да хване същността на съвременната криптография. Първо, той се фокусира единствено по проблема за тайната комуникация[14].

Това се доказва от факта, че дефиницията определя „кодове“, другаде дефинирана като „система от предварително организирани сигнали, особено използвани за гарантиране на секретност при предаване на съобщения ”. Второ, дефиницията се отнася до криптографията като форма на изкуство. Наистина,

до 20-ти век (и вероятно до края на този век) криптографията е изкуство. Изграждане на добри кодове или счупването на съществуващи , разчита на креативността и лично умение. Имаше много малко теория, на която можеше да се разчита и нямаше дори добре дефинирана представа за това какво представлява добър код.

В криптографията криптирането и декриптирането се определя от един или повече ключове. В зависимост от типа криптиране се ползват един или повече ключове Като типовете са следните[42]:

а) Симетрични -

- Използваме се само един единствен ключ, който служи за криптиране и декриптиране

- Ключът на симетричнен алгоритъм превръща обикновен текст с "таен ключ" за да създаде криптирани данни наречени шифров текст

- Симетричният ключ е уязвим при предаването на ключа

- Той е изключително бърз и служи за прехвърляне на голям брой данни

- Пример за това са: RC2,DES,3DES

б) Асиметрични -

- Разрешава проблема, който има симетричният ключ като не предава ключа

- Разчита на двойка ключове личен(private) и публичен(public)

- Публичният ключ се разпространява свободно, тъй като не може да помогне за декриптиране

- Съобщения криптирани с публичен ключ не могат да бъдат декриптирани с публичен а само с личният ключ.

- Всеки може да криптира, но само един човек може да декриптира

- Не толкова бързи колкото симетричният, но за сметка на това много по сигурни

- Пример за това са : HTTPS и SSL

Друг начин за осигуряване на Сигурността на системата е използването на цифров подпис. Подписът се прилага към целия документ, така че ако подписът е променен, документът става нечетлив. Така нареченият Hashing.

Такива системи могат драстично да намалят времето, необходимо за изчисляване на криптографски ключ. В резултат на това за сигурността при системите трябва да намерим нови техники, които да предават данните сигурно без да разчитаме на съществуващоти чисто математически методи, [4]. За това използваме понятия за алтернативна сигурност [5]. Основните алгоритми, които се приемат като алтернативни за сигурността са елиптични, вокални, квантови и алгоритми за криптиране на ДНК. Елиптичените алгоритмити се използват за преносими устройства които имат ограничена процесорна мощност, използвайки проста алгебра и сравнително малки шифри.Тези системи могат драстично намалеят времето, необходимо за изчисляване на криптографският ключ. В резултат на това за сигурността на системите трябва да се намерят нови техники за предаване данните сигурно без разчитайки на съществуващи

чисто математически методи, [4]. Затова използваме понятия за алтернативна сигурност [5]. Основните алгоритми , които се приемат като алтернативни за сигурността са елиптичните, вокалните, квантовите и алгоритмичните при криптиране на ДНК.

Криптографията е „наука за разработване на криптосистеми, които да шифроват и дешифроват данни, наред с други неща "[9]. Данните се променят посредством прилагане на шифър към обикновения текст, за да го направи безполезен за злонамерени потребители . Ние шифроваме съобщения с намерението да запазим нежелани лица от достъп до информацията. Шифроването е много по -сигурно отколкото първоначалното заместване на цели думи, това е така, защото при криптиране цифрови медии сме в състояние да разбием съобщението до битово ниво и шифроване на всеки отделен бит. Един от различните подходи за да гарантираме на сигурността на данните също използва динамични свойства на хаотичните системи. Проучванията за хаотично криптиране са получили повече внимание от стандартните методи за шифроване.

Има различни етапи от процеса на шифроване, през които един файл трябва да премине, преди да е напълно криптиран. Обикновеният текст са данните, които се съдържат в оригинален некриптиран файл.
Този Обикновен текст не се отнася конкретно до текст вместо това се отнася до непроменени данни, налични, когато оригиналният файл е отворен. Някои алтернативни типове файлове са аудио, видео или изображения. Обикновеният текст е двоично представяне на цифрови носители, които съдържат чувствителна информация. Шифровият текст е резултат от прилагането на криптирането

алгоритъм към оригиналния файл. Ключът за декриптиране е набор от стойности, които когато бъдат разкодирани (XORed) с шифровия текст ще върне оригиналния обикновен текст.

Хаотични системи изключително много зависят от началните условия и параметри. Хаотичните системи станаха все по-популярни, те могат успешно да поддържат объркване и разпространение на основните от тях компоненти на криптирането. Ето защо много алгоритми за криптиране, които използват свойствата на объркване и дифузия на хаотични знаци, се разработват всеки ден [27]. Извършени са множество проучвания относно сигурността на компютърната мрежа.

Има някои често използвани съвременни методи за криптиране като AES , DES, RSA [29], RC5, RC6[43] и Blowfish [23], предназначени да поддържат сигурността на данните. Подходът за криптиране, базиран на хаос, е важен избор, предпочитан за повишаване на сигурността на компютърни мрежи. В литературата има много изпълнения на криптиране, които се извършват с хаотични системи. Ще погледнем един от тях малко по-късно

За последователности, генерирани от детерминирани алгоритми, така че да симулират наистина случайни последователности, се казва, че са псевдослучайни (PR). Псевдослучайната последователност в единичният интервал [0, 1) се нарича последователност от псевдослучайни числа (PRN) . [40]

Компонентът Pseudo Random Sequence използва LFSR за да генерира псевдослучайна последователност, която да изведе псевдо случаен битов поток. LFSR е от формата на Галуа и използва предоставената максимална дължина на кода или точка. Компонентът PRS работи непрекъснато след стартиране, стига

входът за активиране да е задържан високо. Генераторът на номера на PRS може да се стартира с всяка валидна начална стойност, различна от 0[10].

Това е само малка част от най-използваните начини за криптиране, но това са най-популярните, които се използват всеки ден както за по малки цели, като криптиране на парола за Wi-Fi така и за криптиране на критични данни например в Пентагона.

Глава 2

Анализ на текущото състояние на криптиране на wav файлове

2.1 Въведение

криптирането на данни е критично за запазването на реалната информация при опит за достъпване и фалшифициране, но и работи върху неговата невидимост във всичките му видове. Счита се за допринасящ фактор за добавяне на по-голяма защита срещу четене, слух, подправяне или унищожаване.

Техниките, използвани за кодиране на информация, са различни, също така техните методи се развиха, за да достигнат такива нива, каквито са в момента. Поради изключителното подобрение и електронното въстание се появи концепцията за биометрична информация, след което тя беше въведена в секретно криптиране на данни и се появиха съответните системи. Това ни доведе до разработване на концепцията за системи за био-криптиране. Криптографията ни дава способността да се съхраняват значителните данни от хакери, които се опитват да ги използва за забранена употреба.

Нещо повече, може да се характеризира като процедура, като дава гаранция, че се защитава кореспонденцията между двете точки [46] , [6]. Включват се главно две части: криптиране и декриптиране. криптирането управлява

смесена субстанция на безопасно съобщение, за да направи разединено или недешифрируемо за всеки неодобрен човек или програма.

Декриптирането е процес при преобразуване на шифрованото съобщение до идентичния му открит текст[28],[25] Пространството на вълни се увеличават изключително бързо. Wavelets са използвани адекватно като актив в множеството различни области като при обработката на сигнали, наблюдението на звезди и криптография [39] . Целочислена вълна на преобразуването е вид вълнообразно преобразуване, което се съпоставя с целочисления набор от данни с друг набор от цели числа, значимите свойства на целочислената wavelet трансформация и нейните коефициенти имат подобен динамичен диапазон. Това прави по-прости съображения за използване по отношение на размера на факторите, които трябва да се използват, и на техният обхват за да се побере в изчислението за кодиране [30].

Форматът на аудиофайла на формата на вълната(WAV) е аудио на Microsoft и IBM стандартен файлов формат за съхранение на аудио битов поток на персонални компютри. Това е приложение на файла за обмен на ресурси Формат (RIFF) метод за формат на битов поток за съхраняване на данни в "парчета" и по този начин също е близо до 8SVX и форматът AIFF,[26] използван на компютри Amiga и Macintosh. Това е основният използван формат на Windows системи за необработени и обикновено
некомпресирано аудио. Обичайното кодиране на битов поток е линейнеен (LPCM) формат [41]. Звукът е основно вълна под налягане или механична енергия с разлика в налягането в еластична среда. Дисперсията се разпространява като компресия и разреждането при което то компресира възниква, когато налягането е по-високо от околното налягане и неговото разреждане се появява, когато налягането на разпространяващата се вълна е по-малка от околното налягане [24],[7].

Точно по същия начин един WAV файл просто представлява дискретизирани

звукови вълни, които се намират над или под надморското налягане или налягането на околния въздух. В момента обменните данни между потребителите бързо нарастват, така че потребителите трябва да защитят своите системни данни (например видео, аудио, изображение и текст), за да има поверителност от нападателите. Тези системи за сигурност са широко разпространено използвани в областта на базата данни като интернет банкиране и аудио комуникационни канали.

Следователно системите за сигурност са важни аспекти в информационни системи, които са много изследователите продължават да се развиват най- вече в областта на аудио сигурността.

2.2 Видове WAV криптиране

Криптирането на аудио файлове работи на същият принцип както и в криптирането на някакъв различен тип данни. Имайки предвид силата на звука на аудио файловете в различни изследвания [1] се е обсъждало, че всички техники за криптиране могат да бъдат класифицирани в три широки категории като: **Пълно криптиране**, **селективно криптиране** и **комбиниран подход за криптиране на компресия**.

Пълният подход за криптиране е традиционният начин за постигане на поверителност на съдържанието, която криптира целият файл с помощта на традиционни шифри като DES, AES, 3DES, RC4 или RSA. Това води до висока обработка и изчислителна сложност. При подходът за селективното криптиране се криптират части на един мултимедиен файл, за да намалим изчислителните изисквания на клиентската страна в реално време при тези приложения. Основният проблем при този подход е да изберем важни данни, които трябва да бъдат криптирани. Комбинираният подход за криптиране на

компресия комбинира процесът на компресията и процеса на криптиране в една стъпка.

В своята работа изследователите Aathithan, N Radha и Venkatesulu [1] са взели информация при подходът за криптиране за мултимедийните данни в реално време като например изображения, аудио и видео комуникационни приложения.Така при пълното двоично дърво се извършва заместване и две размерният масив извършва линейната дифузия.

Експерименталните резултати доказват успеха на алгоритъма с някое забавяне при стартиране. Изследователите се опитват да приложат своята работа във вградени/мобилни приложения.

Пълният подход за криптиране е традиционният начин за постигане на поверителност на съдържанието, която криптира целият файл с помощта на традиционни шифри като DES, AES, 3DES, RC4 или RSA. Това води до висока обработка и изчислителна сложност. Този подход за селективно криптиране криптира частите на мултимедиен файл, за да намалим тези изчислителни изисквания на клиентската страна в реално време.

Основният проблем при този подход е да изберем важни данни, които трябва да бъдат криптирани. Комбинираният подход за криптиране на компресия комбинира процесът на компресия и процесът на криптиране в една стъпка.

2.2.1 Криптиране на WAV чрез генератор на псевдо-случайни числа

Генераторът на псевдо случайни числа е детерминиран алгоритъм, при който се генерира поредица от случайни числа с намерението, че всяко генерирано число е непредсказуемо.

Много алгоритми са разработени с идеята да се създадат наистина случайни последователности от числа, безкрайни низове от цифри, в които теоретично е напълно невъзможно да се предвиди следващата цифра в последователността въз основа на цифрите до дадена точка. Но самото съществуване на алгоритъма, колкото и сложен да е означава, че следващата цифра може да се предвиди! Това породи съответният термин псевдослучайност за такива машинно генерирани низове от цифри. Те са еквивалентни на последователности със случайни числа за повечето приложения, но не са наистина случайни според строгото определение.

Генератори на псевдо случайни числа (PRNG) играят важна роля в компютрите и комуникацията технология, независимо дали става въпрос за онлайн хазарт, намаляване на сблъсъци в Ethernet мрежи или осигуряване на данни чрез криптографски техники. Често приложение на PRNG е да се осигури комуникация на големи количества данни, чрез несигурни публични канали. Често срещан подход е да се обменят части на PRNG използвайки криптосистеми с публичен ключ и след това XOR потока от данни е с произведен поток от псевдо случайни числа от PRNG. Приемникът може да обърне операцията XOR като той също може да генерира същия поток от псевдослучайни числа.[33]

Съставен с хаотична окръжност и модифицирани уравнения на въртене. Схемата на нов псевдо-случаен генератор е представена и използвана като основа за хаотични пермутации и заместване на битово ниво, приложени към структурата на аудио файловете за успешно криптиране.

2.2.2 Криптиране на WAV чрез RSA

RSA(Rivest-Shamir-Adleman) е проектиран от Рон Ривест, Ади Шамир и Леонард Адлеман през 1978 г. Това са едни от най-известните публични ключове при криптосистеми за обмен на ключове или цифрови подписи или криптиране на

блокове данни. RSA използва променлив размер блок за криптиране и ключ с променлив размер. Той е асиметричен (публичен ключ) криптосистема, базирана на теорията на числата, която е в блокова система за шифроване. Той използва две прости числа за да генерира публичните и личните ключове. Използват се тези два различни ключа за криптиране и декриптиране. Подателят криптира съобщение с помощта на публичния ключ на приемника и когато съобщението получава предаване към приемника, тогава приемникът може да го дешифрира, като използва неговия собствен личен ключ [3] [47]. Операциите на RSA могат да бъдат разложени в три широки стъпки; генериране на ключове, криптиране и декриптиране. RSA имат много недостатъци в своя дизайн, следователно не са предпочитани за търговска употреба. Когато малките стойности на p и q са избрани за проектиране на ключ, след това процесът на криптиране става твърде слаб и човек може да успее да декриптира данни чрез използване на теория на случайните вероятности и канал за странични атаки. От друга страна, ако са избрани големи p и q дължини след това отнема повече време и производителността се влошава в сравнение с DES. Освен това алгоритъмът също изисква подобни дължини за p и q, на практика това са много трудни условия за задоволяване [13].

2.2.3 Криптиране на WAV чрез DES

DES (Data Encryption Standard) е блоков шифър, базиран на незначителни вариации на Feistel структурата [21], . Издаден е през 1977 г. като FIPS PUB 46 (Федерални стандарти за обработка на информация) от NIST (Национален институт по стандарти и технологии) [44] и след това широко използван повече от две десетилетия. Откритият текст е обработени в 64-битови блокове. Ключът е с дължина 56 бита, което е разделен на 16 подключове за 16 кръга на

обработка; Всеки по един се използва за всеки кръг. Декриптирането е същото като криптирането където шифровият текст се използва като вход за DES и под-клавишите Ki се използват в обратен ред, т.е. от K16 в бърз кръг до K1 в последния кръгъл. С 56 битов ключ има 256 = 7,2 x 1016 възможни Налични са ключове, което прави атака с груба сила непрактично. Но DES окончателно се оказа несигурна през юли 1998 г., когато Фондацията за електронна граница (EFF98) обяви, че са кракнали DES криптиране с помощта на „DES Cracker“ със специално предназначение.

2.2.4 Криптиране на WAV чрез RC4

По представения начин алгоритъмът за криптиране RC4 се използва за криптиране на оригинален текст и същия ключ се използва за криптиране и декриптиране. Тук размерът на ключа варира от 1 до 256 бита. Резултатът на шифровят текст е вграден в обекта на корицата. Обектът на корицата, използван в предложената техника, е аудио файл. Прилага се процедура за вземане на проби към аудио и след това подходящият бит се променя с шифров бит [18]. Форматът на използваните аудио файлове е WAV файлове. WAV файловете са некомпресирани файлове. Когато нещо се компресира, губим разделителна му способност. В света на звука това означава, че има по-малко информация за работа с аудио обработка и по-малка възможност за коригиране или регулиране на звука. Следователно WAV файловете се вземат предвид при внедряването.В WAV файловете се съдържат заглавка и данни. Заглавката на WAV файла е дълга 44 байта.

2.2.5 Криптиране на WAV чрез AES

Така наречения разширен стандарт за шифроване (AES), известен също с неговото оригиналното име Rijndael, е спецификация за криптиране на електронни данни, създадени от Националния институт на Съединените Американски Щати на Стандартите и технологиите (NIST) през 2001 г. Той замества Standard Encyption Standard (DES), който е публикуван през 1977 г. По-популярното и широко прието симетрично криптиране алгоритъм, който вероятно ще се срещне в днешно време, е Advanced Стандарт за шифроване (AES). Открива се поне шест пъти по-бързо от тройния DES. Смята се за по-надежден от други алгоритми за криптиране при използване на голяма дължина на секретеният ключ. Беше необходима подмяна на DES, тъй като размерът на ключа също беше малък. С увеличаването на изчислителната мощност се считаха за уязвими срещу изчерпателна атака при търсене на ключове. Тройна DES е проектиран да преодолее този недостатък, но е намерен бавен. Алгоритъмът AES се състои от четири етапа, които съставят кръг, който се повтаря 10 пъти за ключ с 128-битова дължина, 12 пъти за 192-битов ключ и 14 пъти за 256-битов ключ. Това е ефективен алгоритъм. Обикновено е малко вероятно да пропуснете този алгоритъм.

2.3 Извод

При криптирането на WAV файлове вариантите определено са много, като всеки различен тип криптиране си има своите предимства и недостатъци . Ако се чудим кой тип криптиране е по-добър от другия, тогава няма да има ясен победител, тъй като както симетричното, така и асиметричното криптиране носят своите предимства на масата и не можем да изберем само едно за сметка на друго.

От гледна точка на сигурността, асиметричното криптиране криптиране

несъмнено е по-добро, тъй като гарантира удостоверяване и отхвърляне. Производителността обаче също е аспект, който не можем да си позволим да пренебрегнем и затова винаги ще е необходимо да предпочетем симетрично криптиране при криптирането на WAV файлове.

Глава 3

Моделиране на псевдослучаен генератор на базата на функцията triple Ikeda

3.1 Въведение

Във физиката и математиката картата на Икеда е динамична система с дискретно време, дадена от сложната карта. Оригиналът на карта е предложена първо от Кенсуке Икеда като модел на светлина, преминаваща през нелинеен оптичен резонатор (пръстенна кухина,съдържаща нелинейна диелектрична среда) в по-обща форма.Тя се свежда до опростената "нормална" форма от Ikeda, Daido и Akimoto означава електрическото поле вътре в резонатора на n-та стъпка на въртене в резонатора и са параметри, които показват лазерната светлина приложени отвън и линейна фаза съответно през резонатора.

Генераторите на случайните числа са физически източници, които могат да се връщат равномерно разпределени и с напълно непредсказуеми данни. Наистина случайни числа са приложими за вид задачи, като криптиране на данни, игри и експериментални дизайни. Генерирането на случайни числа е особено трудно. Генераторите на псевдослучайни алгоритми са софтуерни

алтернативи на наистина случайни генератори.[36] Те са алгоритми, които използват математически формули за производството на псевдослучайните последователности от различни числа. През последните петнадесет години хаотичността на функциите и схемите за самосвиване се използват активно в областта на псевдослучайно поколение. Рекурсивните хаотични функции се характеризират с „непредсказуемо“ поведение, което може да се използва при производството на псевдослучайни стойности. За разпознаване на псевдослучайни потоци има ключов пространствен анализ [2], корелационен анализ [19], анализ на скоростта и статистически тестовите за пакети ENT [45] и NIST [31].

3.2 Формула при Псевдослучаен генераторна базата на Triple Ikeda

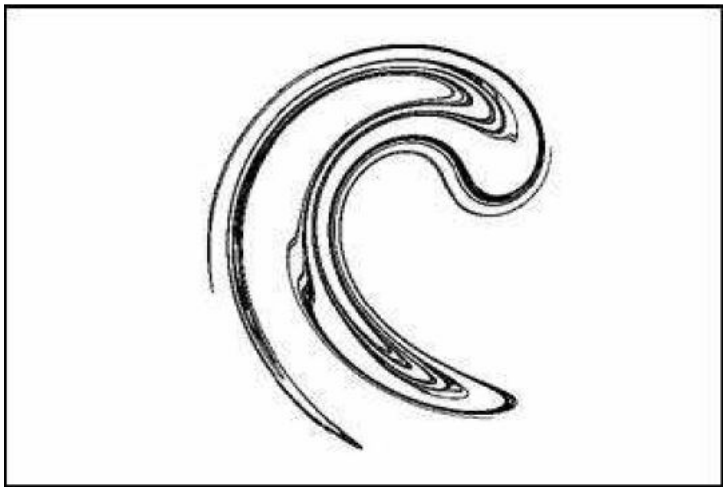
Един от начините за добавяне на нелинейност е да приемем, че ъгълът на въртене 0 е нелинейна функция, какъвто е случаят с известния атрактор Ikeda. Моделът се основава на система от две уравнения на въртене с параметър за превод a, добавен към първото уравнение, пространствен параметър b и нелинеен ъгъл на въртене от формата:

$$\theta_t = c - \frac{d}{1 + x_t^2 + y_t^2}$$

Системата за въртене в този случай има следната форма:

$$x_t + 1 = a + b(x_t cos(\theta_t) - y_t sin(\theta_t))$$

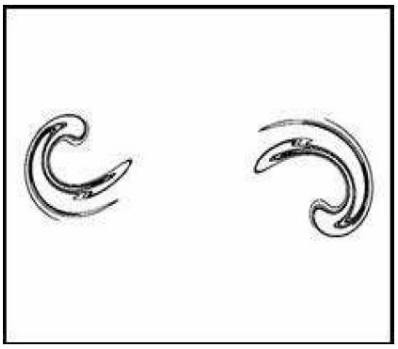
$$y_t + 1 = b(x_t sin(\theta_t) - y_t cos(\theta_t))$$



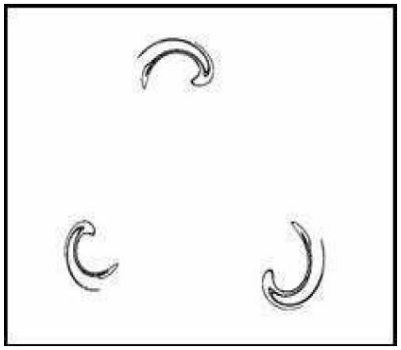
Фиг. 3.1: Икеда атрактор

Тази система е предложена за първи път от Ikeda (1979), за да обясни разпространението на светлината в пръстеновидна кухина. Стойностите на параметъра в този случай са a = 1, b = 0,83, c = 0,4 и d = 6 и те водят до много интересен атрактор, илюстриран на фигура 3.1.

Свойствата на хаотичните карти в тази категория са толкова богати и често неочаквани. Формата на атрактора Ikeda в предишната стандартна форма зависи от изборът на стойностите на параметрите. Като променим тези параметри, забелязваме че се появяват нови форми. Два от тези атрактори са илюстрирани на фигури 3.2 и 3.3. В Фигура 3.2 се появява двойка атрактори. Параметрите са a = 6, b = 0.9, c = 3.1 и d = 6. На фигура 3.3 се променя само параметърът c (c = 2.22). Сега три присъстват изображения на атракторите.[34]



Фиг. 3.2: Двоен Икеда атрактор



Фиг. 3.3: Троен Икеда атрактор

3.3 Експерименти

3.3.1 Изчисляване на XOR Троен Икеда атрактор с 100 000 итерации

Като първи опит реших да направя 100 000 калкулации при стойности a = 6, b = 0.9, c = 2.22 , d= 6, x = 0.1 и y = 0.1 .

Изчисленията направих с програмата Матлаб, като изплозвах формула за Псевдослучайната карта на Triple Ikeda.

След изчислението на 100 000 итерации получените резултати обърнах на цели числа. След като ги преобърнах двете цели числа ги обърнах по модул 2 и ги направих по XOR. Получените резултати изглеждат по следният начин:

Count of "1" = 55584

Count of "0" = 44416

С тези резултати разбираме, че изчисленията са ни правилни, тъй като получихме почти равен брой 1 и 0.

3.3.2 Изчисляване на Троен Икеда атрактор при 2 милиарда итерации и тестване с програма за тестване на псевдослучайни числа

При тези изчисления поради огромният брой итерации беше невъзможно за мен да направя изчисленията на Матлаб за това за целта използвах C++ .

Използвах същите стойности както при миналият опит, но промених x = 0.121135546 и y = 0.780000034 .

За програма за тестване на псевдослучайни числа използвахме ENT, която прилага различни тестове към поредици от байтове, съхранявани във файлове, и отчита резултатите от тези тестове. Програмата е полезна за оценка на генераторите на псевдослучайните числа при приложения за криптиране и статистическа извадка, алгоритми за компресиране и други приложения, където информационната плътност на файл представлява интерес.[37]

ENT извършва различни тестове на потока от байтове във infile (или стандартен вход, ако няма посочен infile) и произвежда изход, както следва на стандартния изходен поток.

Например :

Entropy = 7.980627 bits per character.

Optimum compression would reduce the size of this 51768 character file by 0 percent.

Chi square distribution for 51768 samples is 1542.26, and randomly would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 125.93 (127.5 = random).

Monte Carlo value for Pi is 3.169834647 (error 0.90 percent).

Serial correlation coefficient is 0.004249 (totally uncorrelated = 0.0).

Entropy

Информационната плътност на съдържанието на файла, изразена като брой битове на знак. Горните резултати, получени при обработка на файл с изображение, компресиран с JPEG, показват, че файлът е изключително плътен в информация - по същество случаен.

Следователно, компресирането на файла е малко вероятно да намали неговия размер. Обратно, изходният код на C на програмата има ентропия от около 4,9 бита на символ, което показва, че оптималното компресиране на файла би намалило размера му с 38%

Chi-square тест

Chi-square тест е най-често използваният тест за случайност на данните и е изключително чувствителен особено към грешки в генераторите на псевдослучайните последователности. Chi-square изчислява потока от байтове във файла и се изразява като абсолютно число и получава процент, който показва колко често една случайна последователност би надвишила тази изчислена стойност.

Ние интерпретираме процента като степента, до която се подозира, че тестваната последователност е неслучайна. Ако процентът е по -голям от 99% или по -малък от 1%, последователността почти със сигурност не е случайна.

Ако процентът е между 99% и 95% или между 1% и 5%, последователността е съмнителна. Процентите между 90% и 95% и 5% и 10% показват, че последователността е „почти подозрителна“.

Интересно е прилагането на този тест към изхода на различни типове генератори на псевдослучайни последователности. 8 бита от нисък ред, върнати от стандартната функция Unix rand (), например:

Chi-square разпределение за 500 000 проби е 0,01 и на случаен принцип би надвишило тази стойност повече от 99,99 процента от пъти.
Докато подобрен генератор съобщава:

Chi-square разпределение за 500 000 проби е 212,53 и на случаен принцип би надвишило тази стойност 97,53 процента от пъти.

По този начин стандартният Unix генератор (или поне байтовете от нисък ред) е неприемливо неслучаен, докато подобреният генератор е много по-добър от старият, но все пак достатъчно случаен, за да предизвика безпокойство за взискателни приложения.

Arithmetic Mean(Средноаритметично)

Това е просто резултат от сумиране на всички байтове (битове, ако е посочена опцията -b) във файла и разделяне на дължината на файла. Ако данните са близки до случайни, това трябва да бъде около 127,5 (0,5 за изход за опция -b). Ако средната стойност се отклонява от тази стойност, стойностите са постоянно високи или ниски.

Monte Carlo Value for Pi(Монте Карло Стойност за Pi)

Всяка последователност от шест байта се използва като 24-битови X и Y координати в квадрат. Ако разстоянието на произволно генерирана точка е по-малка от радиуса на окръжност, вписана в квадрата, шест-байтовата последователност се счита за „хит“. Процентът попадения може да се използва за изчисляване на стойността на Pi.

За много по-големи потоци (това приближение се сближава много бавно) стойността ще се доближи доста до правилната стойност на Pi, ако нейната последователност е близка до случайна. Файл от 500 000 байта, създаден чрез радиоактивен разпад, даде: Стойността на Монте Карло за Pi е 3,143580574 (грешка 0,06 процента).

Serial Correlation Coefficient(Коефициент серийна корелация)

Това количество измерва степента, в която всеки байт във файла зависи от предходния байт. За случайни последователности тази стойност (която може да бъде положителна или отрицателна), разбира се, ще бъде близка до нула. Неслучайният байтов поток като програма на C ще даде сериен коефициент на корелация от порядъка на 0,5. Изключително предсказуеми данни, като некомпресирани растерни карти, ще показват приближаване на серийни коефициенти на корелация.

След дълги изчисления получих изключително точен резултат с "ENT".

Резултатите бяха следните :

Entropy = 1.000000 bits per bit.

Optimum compression would reduce the size of this -1179869184 bit file by 0 percent.

Chi square distribution for -1179869184 samples is -32.85, and randomly would exceed this value more than than 99.99 percent of the times.

Arithmetic mean value of data bits is 0.5001 (0.5 = random).

Monte Carlo value for Pi is 3.141533967 (error 0.00 percent).

Serial correlation coefficient is 0.935734 (totally uncorrelated = 0.0).

3.4 Увод

Двата експеримента ни показват колко точно се работи при генериране на псевдослучайни числа на принципа на картата Тройна икеда. Това което забелязваме с програмата ENT, е че оценката на генераторите на псевдослучайните числа за криптиране и статистическа извадка е много висока и например Монте карло сойност за ни излезе с 0 процента грешка, което е забележително.

В тази глава използвахме нов източник на псевдослучайни байтове, базиран на Тройна Икеда. Размерът при началната стойност, линейна сложност и статистически резултати от тестове показват, че новата схема можем да гарантирам добър подобен на случайният принцип характер и негово допустимо ниво на сигурност.

Глава 4

Моделиране на алгоритъм за криптиране на wav файлове

4.1 Въведение

Криптографията като цяло е изкуство на тайно предаване на данни от подател до получател или група от получатели. Съвременните технологии промениха начина, по който информацията се изпраща, тъй като самата информация вече е цифрова под формата на битове (1 и 0), прехвърлени в компютърните мрежи по целия свят. Този фактор изисква стандартни криптографски алгоритми, използвани преди дигиталната ера, да бъдат приложени и/или модернизирани за работа с цифрова информация. В тази статия представяме нов метод за криптиране, предназначен за защита на аудио файлове, за да се съхранява и прехвърля безопасно този конкретен тип файлове. Другият важен аспект на криптологията е криптографският анализ с основната цел да разкрие шифрованите съобщения.

Криптографският анализ често използва различни видове емпирични експерименти, които могат да се използват за установяване дали някой от предложените алгоритми има необходимото ниво на сигурност или за доказване, че алгоритъмът не е достатъчно защитен. В тази работа ние предоставяме обширен криптографски анализ за потвърждаване нивото на сигурност на

предложената схема за криптиране на аудио.

Предишни изследвания в тази област ни показаха, че използването на хаотични карти за изграждане на алгоритми за криптиране на аудио води до високи нива на сигурност. В [20] Liu, Kadir и Li ни предлагат схема за криптиране , използваща хаос и дифузия , базирана на добре познатата хаотична система с голям брой превъртания.

Хато и Шихаб извършват оценка при хаотичната система на Лоренц и Рослер за криптирането на речеви сигнали в [11]. По-ценни изследвания са представени в [32], където Сатиямурти и Рамакришнан използват хаотична карта на Бернули за изграждане на алгоритъм за криптиране. Тамими и Абдала предоставят още един подобен подход за алгоритъм за шифроване на аудио в [38].

4.2 Псевдо-случаен генератор, базиран на тройна Ikeda

Псевдо-случайният генератор са софтуерно проектирани инструменти и са предназначени да осигурят безкрайна последователност от случайни битове.Псевдо-случайният генератор най-често се използват като основен ресурс за симетрични криптографски алгоритми, като се използват като случайни битове за криптиране и декриптиране на цифрови файлове.
Хаотичните карти са доста широко предпочитани за конструирането на псевдослучайните генератори поради тяхното хаотично поведение [[15], [17], [35]] .

4.2.1 Уравнения за изчисляване на базата на тройна Ikeda

За извършването на анализите ще имаме нужда от уравнения за да можем да създадем псевдо-случайният генератор.

Уравненията, които ще ползваме за анализите са следните:

$$\theta_t = c - \frac{d}{1 + x_t^2 + y_t^2}$$

$$x_t + 1 = a + b(x_t cos(\theta_t) - y_t sin(\theta_t))$$

$$y_t + 1 = b(x_t sin(\theta_t) - y_t cos(\theta_t))$$

4.2.2 Първоначални Стойности

Непроменливите стойности на уравненията са следните:

$$\text{a} = 1$$

$$\text{b} = 0{,}83$$

$$\text{c} = 2{,}22$$

$$\text{d} = 6$$

Стойностите, които ще бъдат различни спрямо съответният аудио WAV файл са X_t и Y_t .

4.2.3 Начин на изчисления

За нашата цел имах нужда от програмен код и софтуер за да мога лесно да успея да направя изчисленията, да правя бързи и лесни промени по стойностите и да мога да създам файл, който лесно да изпълнява кодът за бързо генериране на

файловете. За тази цел използвах Програмнен код C++ и софтуерът на който работих е "Visual Studio 2019".

4.3 Стъпки по криптиране и декриптиране на аудио файл

4.3.1 Криптиране

1. Използвайки уравненията от 4.2.1 Подготвяме кода по тях.
2. Променяме X_t и Y_t на случаен принцип Например $X_t = 0.123355611$.
3. Генерираме .exe файла през "Visual Studio"
4. Поставяме файлът в една и съща папка заедно с WAV файлът.
5. Подкарваме .exe файлът
6. Програмата ни пита за име на сорс файлът и на дестинационният файл. Подаваме данните.
7. В зависимост от големината на аудио файла може да отнеме между няколко секунди и няколко минути.

4.3.2 Декриптиране

Декриптирането е много по-лесно разбира се, тъй като криптиращият ключ , който използваме е симетричен което означава, че същият ключ, който създадохме за криптиране може да бъде използван за декриптиране.

1. подкарваме същият .exe файл, с който криптирахме аудио файла.
2. Като сорд файл подаваме името на криптираният файл и избираме име на дестинационният файл и потвърждаваме.

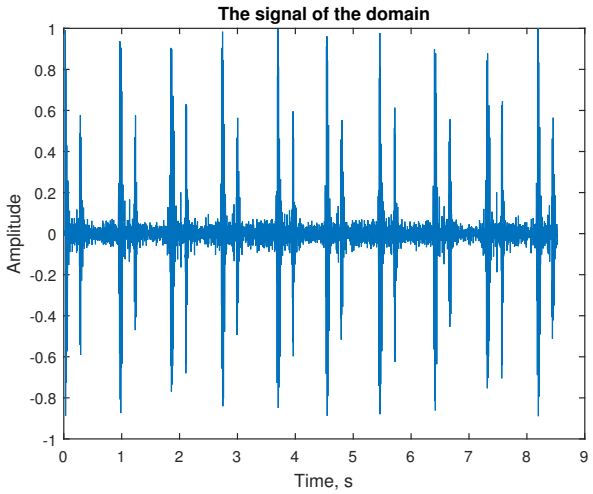
3. Времето за декриптиране е еднакво с времето за декриптиране.

4.4 Криптографичен Анализ

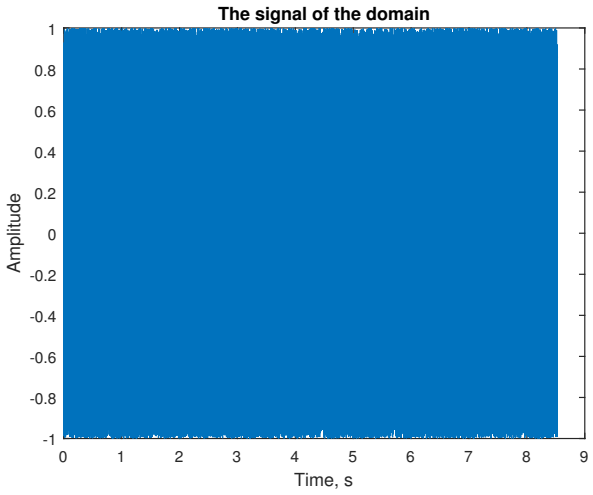
Основната цел на криптографския анализ е да възстановим обикновеното съобщение от криптираното съобщение. В този раздел, за да можем точно да докажем ефективността на аудио криптирането се налага да извършим емпиричните тестове, за да сравним обикновените файлове и съответните им криптирани файлове.

4.4.1 Вълнообразна форма на чертане

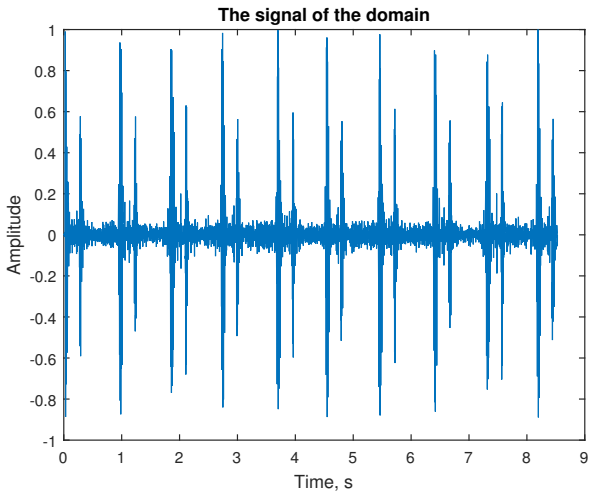
Един от най-често срещаните подходи, отнасящи се до анализ на аудио сигнал, е начертаването на формата на вълната за показване на амплитуда на аудио сигнала, разпределена във времето. За да сравним обикновените аудио файлове с шифрованите, представяме визуализацията на един от тестваните файлове.Фигура 4.1 ни представлява формата на вълната на файла преди криптирането, Фигура 4.2 ни представя промените във файла след криптиране, а Фигура 4.3 ни демонстрира възстановения файл след декриптиране.



Фиг. 4.1: Оригинален WAV файл



Фиг. 4.2: Шифриран WAV файл

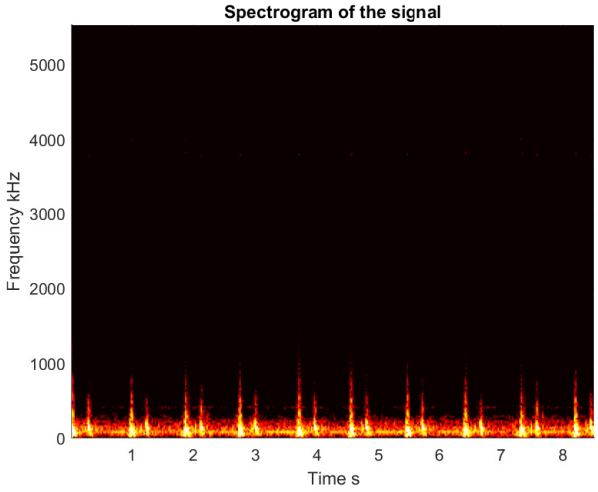


Фиг. 4.3: Дешифриран WAV файл

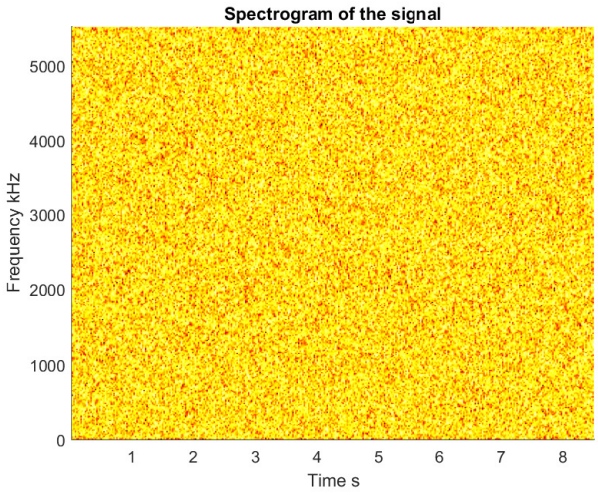
Разликата която наблюдаваме между обикновенният графичен файл и криптираният графичен файл е индикация за успешно криптиране. Освен това силната разлика означава също, че оригиналният файл не може да бъде възстановен дори частично.

4.4.2 Изграждане на спектрограма

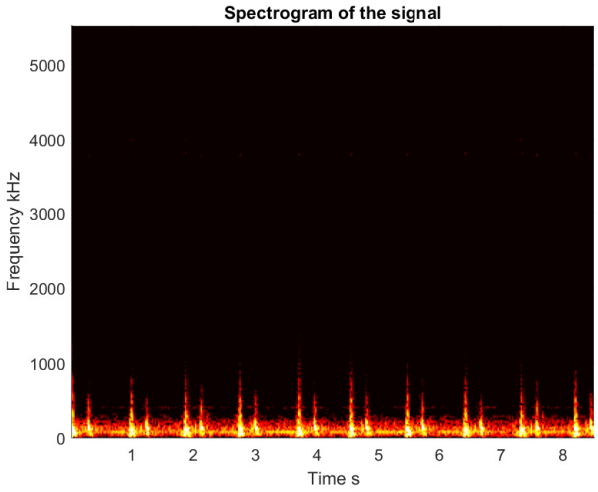
Начертаването на спектрограма е друг важен подход за анализ на аудио сигнали. В този случай основният фокус е честотата на звука спрямо времевата област. Сравняването на обикновени файлове с криптирани файлове ни позволява да видим разликата между файловете и да оценим предложения алгоритъм за криптиране на звука. Фигура 4.4 ни показва спектрограмата на обикновен файл, фигура 4.5 представя промените във файла след криптиране, а фигура 4.6 демонстрира възстановения файл след декриптиране. Този график на спектрограма на криптиран файл означава, че честотата на оригиналния сигнал в обикновения файл е изцяло унищожена. Този тест е друг показател за високите свойства на криптиране на предложения алгоритъм за криптиране на WAV файл.



Фиг. 4.4: Спектрограма на Оригинален WAV файл



Фиг. 4.5: Спектрограма на Шифриран WAV файл



Фиг. 4.6: Спектрограма на Дешифриран WAV файл

4.4.3 Корелационен анализ

Измервателният коефициент на корелацията между двата аудио файла изразява зависимостта между съответните им примерни стойности. Това е друга статистическа оценка за тестване на качествените алгоритми за криптиране. Изчисляването на коефициента на корелация определя нивото на корелация между два файла и коефициентът на корелация винаги е в диапазона -1,1. Стойности между 1–0,7 се счита за силна корелация (извадки от обикновените файлове са подобни на проби от криптиран файл) , корелация между 0.7–0.3 се счита за средна корелация и стойности между 0,3–0 се счита за слаба корелация.

Коефициентът на корелация може да се изчисли, както следва:

$$r_x y = \frac{cov(x,y)}{\sqrt{D_(x)}\sqrt{D_(y)}}$$

където

$$D_(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2$$

$$D_(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2$$

$$cov(x,y) = \sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})$$

N е общият брой на извадките, x_i и y_i са примерните стойности на криптираните файлове, $\overline{x}$ и $\overline{y}$ са средните стойности на пробите и накрая $cov_(x,y)$ е ковариация между двата файла. Таблица 4.1 показва получените

стойности на резултатите от нашите тестове.

	File	File Size	File Length (s)	Correlation Coefficient
	Forrest.wav	105 kb	4 s	-0.000503917
	HeartBeat.wav	183 kb	8 s	-0.000259124
	Knocking.wav	97.7 kb	2 s	-0.000563234
	R2D2.wav	21.8 kb	2 s	0.001639352
	Sheep.wav	28.8 kb	2 s	-0.000903734

Табл. 4.1: Корелация между обикновени и криптирани аудио файлове.

Резултатите в Таблица 4.1 показват стойности, близки до нула, което означава, че няма зависимост между двата файла. Резултатите означават и високо качество на криптирането.

4.4.4 Съотношение сигнал / шум

Съотношението сигнал / шум (SNR) е широко разпространено за да се определи качеството на сигналите [11] , [12] . Стойностите, по -големи от 0 dB, показват, че чистият сигнал е повече от шума. За този тест се нуждаем както от обикновени, така и от криптирани аудио файлове и SNR се изчислява, както следва:

$$SNR = 10log_{10}\frac{\sum_{i=1}^{N} x_i^2}{\sum_{i=1}^{N}[x_i - y_i]}$$

където x_i и y_i са съответстващи пробни стойности от аудио файлове, а N е броят на пробите. Резултатите от нашите тестове за SNR са показани в следващата таблица 4.2.

File	File Size	File Length (s)	SNR (dB)
Forrest.wav	105 kb	4 s	-23.31432437
HeartBeat.wav	183 kb	8 s	-7.546696633
Knocking.wav	97.7 kb	2 s	-22.64190246
R2D2.wav	21.8 kb	2 s	-2.166093787
Sheep.wav	28.8 kb	2 s	-6.447583639

Табл. 4.2: Съотношение сигнал / шум.

Всички получени стойности за SNR са отрицателни, което означава, че шифрованите файлове са много шумни и методът на криптиране напълно унищожава чистия сигнал от обикновени аудио файлове.

4.4.5 Пиково съотношение сигнал / шум

Съотношението връх сигнал към шум (PSNR) е различен подход с който можем да измерим силата на чистия сигнал спрямо силата на шума. PSNR е по-приложим за алгоритми за криптиране на изображения, но може да се използва за тестване на качеството на предложената схема за криптиране в тази статия. PSNR се изчислява, както следва:

$$PSNR = 10log_{10}\frac{MAX^2}{MSE}dB$$

където MAX е максималната възможна стойност на аудио потока (В нашия случай максималната стойност е 65,535), а MSE е средна квадратна грешка между обикновения и криптиран файл. MSE се определя като:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

където N е общият брой проби, xi и yi са съответните стойности на извадката на обикновените и криптирани файлове.

Таблица 4.3 съдържа резултатът от нашият тест.

File	File Size	File Length (s)	PSNR (dB)
Forrest.wav	105 kb	4 s	4.7659
HeartBeat.wav	183 kb	8 s	4.5308
Knocking.wav	97.7 kb	2 s	4.7607
R2D2.wav	21.8 kb	2 s	3.9877
Sheep.wav	28.8 kb	2 s	4.4857

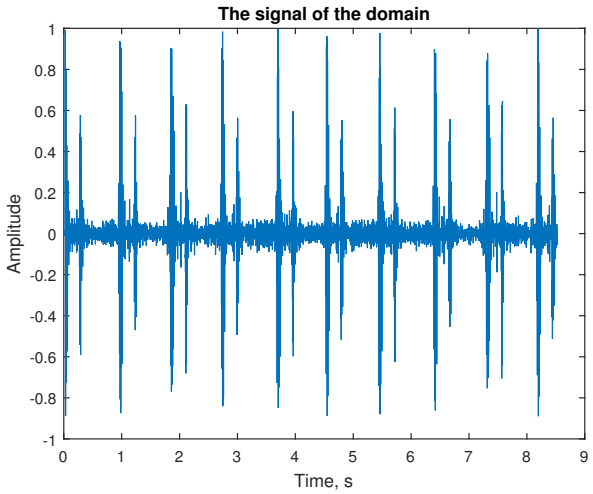
Табл. 4.3: Пиково съотношение сигнал / шум.

Всички получени стойности за PSNR са близки до нула (или по-ниска), което показва много високо ниво на шум в шифрованите аудио файлове.

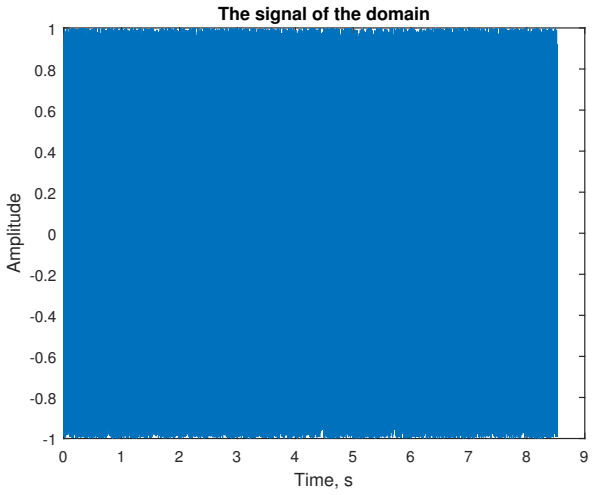
4.4.6 Чувствителност на ключовете за криптиране / декриптиране

Анализирайки общото поведение на метода за криптиране, използвахме много сходни ключове за криптиране и възстановяване на шифрован аудио файл. Ключът за декриптиране се получава чрез промяна на една цифра от една от променливите. В фигурите по-долу ще видим, че дори и при минимална промяна на един от ключовете аудио файлът не може да бъде декриптиран.

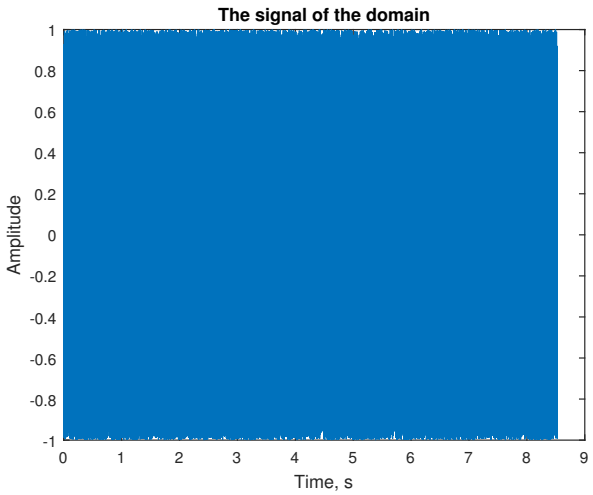
Както 4.9, така и 4.12 показват, че дешифрирането е неуспешно дори при много сходен секретен ключ. Промяната на една цифра на ключа води до неуспешно декриптиране. Експериментът е доказателство за висока чувствителност на ключа по отношение на предложения алгоритъм за криптиране на аудио.[16]



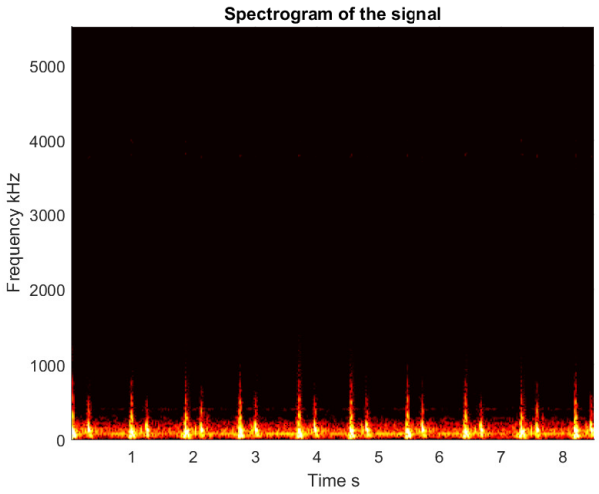
Фиг. 4.7: Оригинален WAV файл



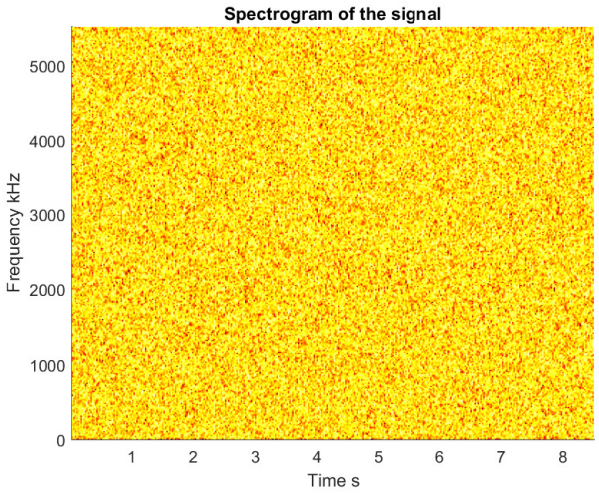
Фиг. 4.8: Шифриран WAV файл



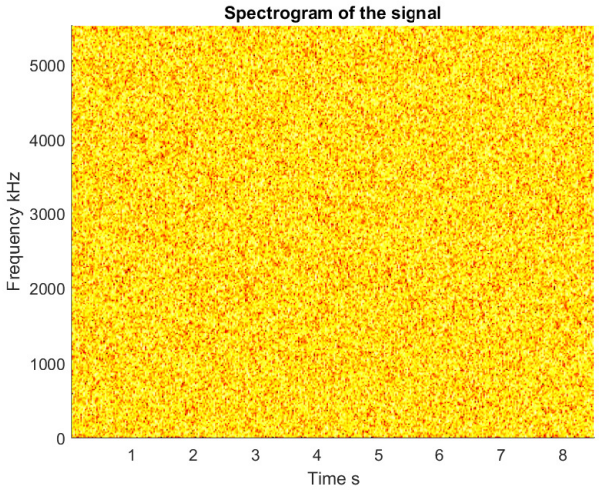
Фиг. 4.9: Дешифриран WAV файл с ралзичен ключ



Фиг. 4.10: Спектрограма на Оригинален WAV файл



Фиг. 4.11: Спектрограма на Шифриран WAV файл



Фиг. 4.12: Спектрограма на Дешифриран WAV файл с различен ключ

4.4.7 Скорост на изпълнение

За измерване на необходимото време за шифроване използвахме аудио файлове с различен размер с хардуерна конфигурация-2Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz 2.69 GHz, 12 GB RAM, Windows 10 20H2. 4.4 съдържа резултатите от нашите тестове.

File	File Size	Bytes per Sample	Encryption Time (s)
HeartBeat.wav	183 kb	2	0.238 s
Knocking.wav	97.7 kb	2	0.155 s
police.wav	68.7 kb	2	0.108 s
R2D2.wav	277 kb	2	0.554 s
Sheep.wav	87 kb	2	0.134 s
Forest.wav	9.95 mb	2	30.008 s

Табл. 4.4: Скорост на изпълнение за криптиране на файловете

4.4.8 Извод

Този експеримент тества и оценява нов дизайн за алгоритъм за криптиране на аудио файлове. Предложеният криптографски алгоритъм разчита на пермутационно-заместваща архитектура, реализирана чрез използване на генератор на псевдослучайни числа и функцията Тройна Икеда. Разширен криптографски анализ се извършва за тестване на предложения метод за сигурност. Графиците на формата на вълна и спектрограмите за тест на аудио файлове демонстрират промените в шифрованите файлове когато сравним с обикновените файлове. Измерените стойности на SNR и PSNR показват високи нива на шум в шифрованите файлове, което показва, че оригиналният сигнал е унищожен в процеса на шифроване. Анализът на ключовото пространство показва необходимото ниво на защита срещу атаки с груба сила, а анализът на чувствителността на ключа показва, че дори минимална промяна на секретния ключ води до неуспешно декриптиране. Като се имат предвид получените резултати по време на криптографския анализ, можем да заключим, че

предложеният алгоритъм има необходима криптографска защита за криптиране на аудио файлове.

Литература

[1] N Radha Aathithan and M Venkatesulu. “A complete binary tree structure block cipher for real-time multimedia”. In: *2013 Science and Information Conference*. IEEE. 2013, pp. 346–352.

[2] Gonzalo Alvarez and Shujun Li. “Some basic cryptographic requirements for chaos-based cryptosystems”. In: *International journal of bifurcation and chaos* 16.08 (2006), pp. 2129–2151.

[3] Kumar Aman, Jakhar Sudesh, and Makkar Sunil. “Comparative Analysis between DES and RSA Algorithm”. In: *International Journal of Advanced Research in Computer Science and Software Engineering* 2.7 (2012), pp. 386–391.

[4] Thomas Beth and Zong-Duo Dai. “On the Complexity of Pseudo-Random Sequences - or: If You Can Describe a Sequence It Can’t be Random”. In: *Advances in Cryptology — EUROCRYPT ’89*. Ed. by Jean-Jacques Quisquater and Joos Vandewalle. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 533–543. ISBN: 978-3-540-46885-1.

[5] Simon R. Blackburn, Tuvi Etzion, and Kenneth G. Paterson. “Permutation Polynomials, de Bruijn Sequences, and Linear Complexity”. In: *Journal of Combinatorial Theory, Series A* 76.1 (1996), pp. 55–82. ISSN: 0097-3165. DOI: https://doi.org/10.1006/jcta.1996.0088. URL: https://www.sciencedirect.com/science/article/pii/S0097316596900886.

[6] Aman Chadha et al. “Dual-layer video encryption using RSA algorithm”. In: *arXiv preprint arXiv:1509.04387* (2015).

[7] Sandipan Dey, Ajith Abraham, and Sugata Sanyal. "An LSB data hiding technique using natural number decomposition". In: *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*. Vol. 2. IEEE. 2007, pp. 473–476.

[8] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[9] Jürgen Eichenauer-Herrmann. "Statistical independence of a new class of inversive congruential pseudorandom numbers". In: *Mathematics of Computation* 60.201 (1993), pp. 375–384.

[10] Rafik Hamza. "A novel pseudo random sequence generator for image-cryptographic applications". In: *Journal of Information Security and Applications* 35 (2017), pp. 119–127.

[11] Eman Hato and Dalya Shihab. "Lorenz and rossler chaotic system for speech signal encryption". In: *International Journal of Computer Applications* 128.11 (2015), pp. 25–33.

[12] Hristo Kabakchiev et al. "Comparison of two algorithms for signal detection in pulsarbased FSR". In: *2018 19th International Radar Symposium (IRS)*. IEEE. 2018, pp. 1–9.

[13] Ajay Kakkar, ML Singh, and PK Bansal. "Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication". In: *in Multinode Network", International Journal of Engineering and Technology Volume*. Citeseer. 2012.

[14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. 2nd. Chapman and Hall,CRC, 2014. ISBN: 1466570261.

[15] KM Kordov. "Modified Chebyshev map based pseudo-random bit generator". In: *AIP conference proceedings*. Vol. 1629. 1. American Institute of Physics. 2014, pp. 432–436.

[16] Krasimir Kordov. "A Novel Audio Encryption Algorithm with Permutation-Substitution Architecture". In: *Electronics* 8.5 (2019). ISSN: 2079-9292. DOI: 10.3390/electronics8050530. URL: https://www.mdpi.com/2079-9292/8/5/530.

[17] Krasimir Kordov. "Signature attractor based pseudorandom generation algorithm". In: *Advanced Studies in Theoretical Physics* 9.6 (2015), pp. 287–293.

[18] Harish Kumar et al. "Enhanced LSB technique for audio steganography". In: *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*. IEEE. 2012, pp. 1–4.

[19] Dragan Lambić. "Cryptanalyzing a novel pseudorandom number generator based on pseudorandomly enhanced logistic map". In: *Nonlinear Dynamics* 89.3 (2017), pp. 2255–2257.

[20] Hongjun Liu, Abdurahman Kadir, and Yanling Li. "Audio encryption scheme by confusion and diffusion based on multi-scroll chaotic system and one-time keys". In: *Optik* 127.19 (2016), pp. 7431–7438.

[21] Akash Kumar Mandal, Chandra Parakash, and Archana Tiwari. "Performance evaluation of cryptographic algorithms: DES and AES". In: *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science*. IEEE. 2012, pp. 1–5.

[22] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001. URL: http://www.cacr.math.uwaterloo.ca/hac/.

[23] Tingyuan Nie and Teng Zhang. "A study of DES and Blowfish encryption algorithm". In: *Tencon 2009-2009 IEEE Region 10 Conference*. IEEE. 2009, pp. 1–4.

[24] Masoud Nosrati et al. "Taking a Brief look at steganography: Methods and Approaches". In: *Journal of American Science* 7.6 (2011).

[25] EO Osaghae. "Replication of ciphertext in cryptographic system". In: *Journal of Applied Sciences and Environmental Management* 22.8 (2018), pp. 1193–1197.

[26] Shital C Patil and RR Keole. "Cryptography, Steganography & Network Securities". In: *International Journal of Pure and Applied Research in Engineering and Technology* 1.8 (2012), pp. 9–15.

[27] Louis M. Pecora and Thomas L. Carroll. "Synchronization in chaotic systems". In: *Phys. Rev. Lett.* 64 (8 1990), pp. 821–824. DOI: 10.1103/PhysRevLett.64.821. URL: https://link.aps.org/doi/10.1103/PhysRevLett.64.821.

[28] AV Prabu et al. "Audio encryption in handsets". In: *International Journal of Computer Applications* 40.6 (2012), pp. 40–45.

[29] Priteshkumar Prajapati et al. "Comparative analysis of DES, AES, RSA encryption algorithms". In: *International Journal of Engineering and Management Research (IJEMR)* 4.1 (2014), pp. 132–134.

[30] R Punidha et al. "Integer wavelet transform based approach for high robustness of audio signal transmission". In: *International Journal of Pure and Applied Mathematics* 116.23 (2017), pp. 295–304.

[31] Andrew Rukhin et al. "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication 800-22 (revised May 15". In: (2002).

[32] P Sathiyamurthi and S Ramakrishnan. "Speech encryption using chaotic shift keying for secured speech communication". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2017.1 (2017), pp. 1–11.

[33] Maruti Satti and Subhash Kak. "Multilevel indexed quasigroup encryption for data and speech". In: *IEEE Transactions on Broadcasting* 55.2 (2009), pp. 270–281.

[34] Borislav Stoyanov. "Double Ikeda map as a source of pseudorandom numbers". In: *AIP Conference Proceedings*. Vol. 2333. 1. AIP Publishing LLC. 2021, p. 070004.

[35] Borislav Stoyanov. "Pseudo-random bit generation algorithm based on Chebyshev polynomial and Tinkerbell map". In: *Applied Mathematical Sciences* 8.125 (2014), pp. 6205–6210.

[36] Borislav Stoyanov. "SELF-SHRINKING CHAOS BASED PSEUDO-RANDOM ALGORITHM". In: ().

[37] Borislav Stoyanov and Krasimir Kordov. "Novel secure pseudo-random number generation scheme based on two tinkerbell maps". In: *Advanced Studies in Theoretical Physics* 9.9 (2015), pp. 411–421.

[38] Abdelfatah A Tamimi and Ayman M Abdalla. "An audio shuffle-encryption algorithm". In: *The world congress on engineering and computer science*. 2014.

[39] MF Tolba et al. "Using integer wavelet transforms in colored image steganography". In: *International Journal on Intelligent Cooperative Information Systems* 4.2 (2004), pp. 230–235.

[40] Alev Topuzoğlu and Arne Winterhof. "PSEUDORANDOM SEQUENCES". In: *Topics in Geometry, Coding Theory and Cryptography*. Ed. by Arnaldo Garcia and Henning Stichtenoth. Dordrecht: Springer Netherlands, 2007, pp. 135–166. ISBN: 978-1-4020-5334-4. DOI: 10.1007/1-4020-5334-4_4. URL: https://doi.org/10.1007/1-4020-5334-4_4.

[41] Animesh Kr Trivedi et al. “RISM–Reputation Based Intrusion Detection System for Mobile Ad hoc Networks”. In: *arXiv preprint arXiv:1307.7833* (2013).

[42] Mircea Vaida. “DNA Security using Symmetric and Asymmetric Cryptography”. In: *International Journal of New Computer Architectures and their Applications (IJNCAA)* 1 (Jan. 2011), pp. 34–51.

[43] Harsh Kumar Verma and Ravindra Kumar Singh. “Enhancement of RC6 block cipher algorithm and comparison with RC5 & RC6”. In: *2013 3rd IEEE International Advance Computing Conference (IACC)*. IEEE. 2013, pp. 556–561.

[44] Om Prakash Verma et al. “Notice of Violation of IEEE Publication Principles: Peformance analysis of data encryption algorithms”. In: *2011 3rd International Conference on Electronics Computer Technology*. Vol. 5. IEEE. 2011, pp. 399–403.

[45] John Walker. “ENT: A Pseudorandom Number Sequence Test Program, 2008”. In: *URL: http://www. fourmilab. ch/random* (2020).

[46] S. F. Yousif. *Encryption and Decryption of Audio Signal Based on RSA Algorithm,* vol. 5. 2018, pp. 57–64.

[47] Xin Zhou and Xiaofei Tang. “Research and implementation of RSA algorithm for encryption and decryption”. In: *Proceedings of 2011 6th international forum on strategic technology*. Vol. 2. IEEE. 2011, pp. 1118–1121.