Висока Школа Електротехнике и Рачунарства Струковних Студија

У Београду



Дипломски рад

Сигурнос Информационих Система

Анђелковић Милош НРТ-35/13 др.Зоран Бањац

Београд 2016.

Садржај

[1 Увод 1](#_Toc460085196)

[2 Криптографија 3](#_Toc460085197)

[2.1 Класична криптографија 5](#_Toc460085198)

[2.2 Симетрични крипотсистеми 8](#_Toc460085199)

[2.2.1 Секвенцијалне шифре 8](#_Toc460085200)

[2.2.2 Блоковске шифре 10](#_Toc460085201)

[2.3 Асиметрични криптосистеми 15](#_Toc460085202)

[2.4 TLS и SSL 16](#_Toc460085203)

# Увод

*“Не постоји знање које није моћ.”*

* Ралф Валдо Емерсон

Информације су у савремено доба, поготово од популаризације интернета и паметних уређаја, постале врло одступне скоро свима. Довољно је да погледате у свој телефон и добијете инфромације о времену, а уз само неколико клика и о временској прогнози, реду вожње или најновијим вестима. Такође је подједнако лако да се контактира нека удаљена особа или да се дође до информациј где је и шта ради. Ово нарвно доноси велике предности од којих је најочигледнија, већ наведена, доступност скоро увек и скоро свуда, као и брзо преношењ и ширење информација. Наравно све ово има и своје мане којих просечан корисник информационог система није баш увек свестан, чак што више технички необразовани људи врло често нису.

Најочигледнија мана јесте непровереност информација на које наилазимо. Наиме веома често људи верују ономе шта прочитају и виде, чак иако је те очигледно нетачно. Још већи проблем предстаља чињеница да се информације, па тако и лажне, невероватном брзином шире по мрежи. Путем друштвених мрежа људи су константо „повезани“ једни са другима и двољно је да једна особа окачи неку погрешну информацију да би је десетине ако не и стотне других види. Неки од њих ће сматрати ту информацију занимљивом и вредном дељења са осталима, и тако врло брзо број људи који ће знати, можда чак и користити, погршну информацију експоненцијално расте. 1993. године Лиса Холст, да би доказала колико брзо и лако се дезинформације шире преко интернета и колико их људи лако прихватају, сатавила је лисут апсурдних „чињеница“, међи којима је можда најпознатија статисика да просечан човек за годину дана у просеку поједе осам паука у току сна. Те „чињенице“ су се наравно прошириле невероватном брзином, а у статистичку „чињеницу“ везану за паукове неки и дан данас верују иако је више пута објашњено да је у потпуности нетача. Међутим даљим истраживањем није могуће пронаћи више информација о Лиси Холст или магазину „PC Professional“, за који је она наводно писала чланке, што даље доводи до питања да ли је просто немогуће пронаћи те информација? Да ли је цела прича о Лиси Холст измишљена да би се проверило колико заиста људи верују у скоро све? Да ли то значи да је статисика о пауцима у ствари тачна, а цела прича о Лиси измишљена да људи не би поверовали у њу? У целој тој причи људи су се у главном концентрисали на то да ли је или није статистика тачна, да се скоро нико није ни запитао ко је и шта Лиса Холст. Могуће је да је једна нетачна или тачна информација побијена неком измишљеном причом. Увек постоји могућност да је неко случајно поделио неки нетачан податак или је то учино само да би се нашалио, али постоје и они који их намерно шире, било да би нашкодили некоме или једноставно зато што воле да праве неред.

При прикупљању и коришћењу информација са интернета увек је добро да се користи више извора. У колико се подациј о истом догађају, ствари или особи превише разликују на два извора онда неко ту греши или шири дезинформације. Треба користити и извор „од поверења“ чије би информације требало да су тачне, сигурне и проверене, али чак ни у том случају ништа не треба прихватати такво какво је, већ увек имати отворен ум и бити свестан могућности да није све онако како је нама приказано. Не мисли се наравно на то да човек треба да буде превише скептичан или параноичан и сматра да сви покушавају да га слажу или преваре, али исто тако не треба баш без и мало размишљања прихватити све шта нам се сервира.

Следећи велики проблем на који се наилази јесте количина личних информација које људи свесно или несвесно остављају на интернету. Свака информација је корисна, можда не за сваког али постоји сигурно неко ко може да је употреби или злоупотреби. У колико је прикупљено довољно наизглед небитних и бескорисних података и добро обрађено и изфилтрирано, могуће је доћи до веома конкретних и битних инфорамција. Популаризацијом друштвених мрежа људи су почели све више и више својих приватних информација и слика да деле са људима које чак ни не знају. Чак и људима са којима нису пријатељу преко друштвене мреже је омогућен приступ великој количини информација. Све чешће људи осећају потребу да са свима поделе то где су, с' ким и шта раде, што и није баш најпаметније. Поред прогонитељ који преко мреже могу да прате скоро сваки корак, позната је чињеница да велико компаније чије услуге користимо попут „Facebook“ и „Google“ без нашег знања прикупљају велике количине информација о нама. У Кини постоје веома стриктна правила везана за интернет. Конторлише се тачно чему корисници имају притуп и велика количина саобраћаја се филтрира. Они немају приступ неким од најпознатијих сајтова попут „Google“, „Facebook“, „Youtube“ и многи други, па уместо њих користе њихове алтернативе. Постоји апликација која се зове „WeChat“ која би требала да мења све апликације за друшвене мреже, али ради и много више од тога омогућујући људима да преко ње наручују храну, одћеу и слично, па чак и да плаћају. Са стране корисника то је одличан ствар јер коришћењем само једне апликације могу да обаве све шта им је потребно, нема потреба за констаним улажењм и излажењм из различитих апликација као ни за укључивањам више њих чиме се успорава рад уређаја, већ је све на једном месту. Велика мана тога јесте у томе што је све на једном месту, односно све информације су на једном месут и влада Кине, која се не устручава у томе, може да прикупи абнормалне количине информација о појединицима међу којима су ко је, одакле, шта ради, где излази, с' ким се виђа и још многе друге.

Постоји наравно добар разлог зашто се толике инфромацие прикупљају о просечном човеку који не би требало да је вредан толике пажње а то је безбедност. Свако од нас је потенцијални криминалац или терориста и у случају да се испостави да неко заиста то јесте могуће је коришћењм ових информација сазнати нешто више о том пеојединцу, што би помогло у његовом хватању, као и његових евентуалних сарадника, али и превенцији сличних ситуација у будућности. Чињеница је да има превише људи и да службе нису заинтересоване за сваког просечног човека понаособ, али све те информације могу да буду веома корисне и као такве их је неопходно чувати.

Претопстављамо, наравно, да компанија која поседује све те информације о нама нема намеру да их на било који начин злоупотреби. Али шта ако неко од запослених који им има приступ није морално чврст или има баш жељу да неком науди. Нарвано да се врше веома строге проверe када се запошљавају људи који у треба да имају приступ огромним базама података, али могућност увек постоји. Држимо се претопставке да нико из компаније ипак неће да злоупотреби наше информације, шта ако неко са стране некако успе да дође до њих. Колико год да је систем сигуран увек постоји могућност да постоји неки порпуст који неко ко је довољно вешт може да искористи. На крају крајева постоји могућност упада и крађе самих хард дискова на којима су сви ти подаци смештени. То су наравно све могућности на које ми никако не можемо да утичемо али могу веома да утичу на нас.

Даље постоји прблем комуникације, како преко мобилне мреже тако и преко интернета. Најочигледнији проблем јесте прислушкивање коминикације. Пресретањем порука или прислушкивањем разговора долази се до прилично личних информација, које у кралњем случају могу да буду и тајне. Највећи проблем јесте у томе што корисици могу да буду потпуно несвесни ове врсте напада јер што се њих тиче комуникације се одвија нормално. У случају пресретања порука могуће су њене измене и прослеђивање лажних порука. Могуће је приметити мало кашњење у коминикацији али је увек могуће да је разлог томе преоптерећеност мреже. Даље се поставља питање идентитета, односно да ли је онај с' ким комуницирамо онај којим се представља. Код разговора могуће је приметити разлике у гласу али шта је са порукама. Могуће је да је неко на неки начин успео да добије приступ профилу или телефону нашег саговорника, лажни профили на друштвеним мрежама су такође опште позната ствар. Како ми у том случају можемо да будемо сигурни да је наша комуникација безбедна?

Постоје наравно још многи други проблеми као што су вируси и други нежељени програми који некако успеју да уђу на наш рачунара и праве проблеме, који могу бити од неких безазлених шала, до озбиљнијих проблема као што је брисање битних информација или пројеката, као и крађа истих. Затим чување података на такозваним „облацима“, за које се показало да могу да буду компромитовани, мада да ли су наши рачунари нешто безбеднији. Листа се наставаља са лажним сајтовима и мејловима који у себи крију вирусе и иде до плаћања преко интернета, на шта се посебно обраћа пажња, јер ипак новац је у питању.

Чињеница је да је немогуће направити потпуно сигуран систем и да ће злонамерни људи увек наћи начин да оборе систем или дођу до неких информација до којих не би требало. Цела сигурност се заснива на одговарању на питања типа: „Шта ако ...?“ и у једном тренутку се дође до тога да на неко питање не може да се одговори или једноставно систем не би могао да функционише нормалоно или уопште када би могао да одговори на свако питање. Потпуно сигуран систем је идеал којим се тежи али је неопходно правити компромисе.

Идеја овог пројекта јесте да се направи систем који омогућује корисницима колико-толико безбедну комуникацију преко мреже. Постоји сервер на који се корисници логују, чиме се обезбеђује аутентификација, односно корисници који нису пријављени на систем не могу да комуницирају са пријављеним корисницима. Све поруке се шифрују, чиме се обезбеђује интегритет, тако да нападачи и ако пресретну поруке не могу да дођу до њиховог садржаја, такође било која измена се детектује немогућствром дешифровања или деобијањем бесмисленог текста. За сваку комуникацију корисници добијају свој кључ за шифровање и дешифровање (биће у даљем тексту објашњено) који се не чува нигде, чак ни на серверу. Систем, наравно, има и своје мане које ће даље бити детаљније разматране.

# Криптографија

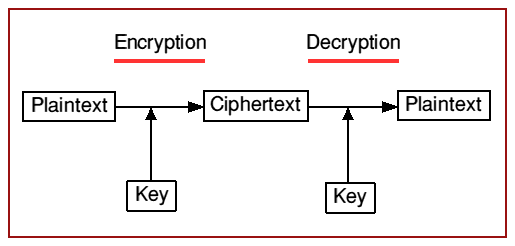
Криптографија је наука која се бави трансформацијом података тако да је њихово значење доступно само овлашћеним странам. Информације које се пренесе комуникационим каналом, који у општем случају није безбедан, могу услед прислушкивања да дођу до лица која не би требала да имају приступ тим информацијама. У колико су у питању финансијске, војне или државне тајне ово може да представља велики проблем, мада и злоупотреба приватних информација како би се нанела штета неком појединцу није занемарљив проблем. Употребом криптографије и одговарајућих криптографских система и алгоритама овај проблем је могуће превазићи.

Појавом писма појавила се и потреба да се сачува тајност садржаја неких писама, тада је и настала криптографија. Од самог почетка шифровање података користило се у војне сврхе. Међу првима је криптографију почео да користи Јулије Цезар, једна од најпознатијијх шифара приосте замене, Цезарова шифра, добила је име баш по њему. Он је све поруке које је слао својим војсковођама шифровао тако што је свако слово поруке замењивао, „померао“, словом које је 3 или више места после тог слова у абецеди. Тако шифровану поруку могли су да дешифрују и разумеју само они који су знали за колико места су слова померена, односно они којима је порука била намењена. Прву расправу о криптографији написао је Леоне Батиста Алберти 1467. Он је такође творац шифарског круга и неких других решења двоструког прикривања тексат која су прихватили усавршили немачки, енглески и француски бирои. Пола века након тога објављено дело Јоханеса Тритхемуса, прва књига о криптофгафији. У 16. веку значајан допринос дају милански доктор Гироламо Кардано, математичар Бартисто Порта и француски дипломата Блаисе де Вигенер.

Тек за време Другог светског рата појавила се машина која је шифровала поруке на до тада невиђен начин. Машину су направили немци и звала се Енигма. Међутим колико год да је она била револуционарана савезници су ипак успели да дешифрују поруке шифроване Енигмом. Оно шта је много више утицало на развој криптографије јесте употреба рачунара. Како су временом рачунари постајали све бржи и моћнији, број операција у секунди је скочио са неколико стотина на неколико милиона па чак и милијарди операција у секунди, 20. јуна. 2016. Sunway TaihuLight поставио је рекорд од 93 petaflops-a, што је 93\*1015 операција у секунди. Ово је потпуно променило начин шифровања и дешифровања порука, ове брзине омогућавају много брже разбијање шифрата што је довело да развијања нових сигурнијих и компликованијих алгоритама за шифровање.

Основни термини који се користе у криптографији су:

* Отворени текст (plaintext)- порука коју је потребно заштити.
* Шифровање (encryption)- операција којом се отворени текст мења тако да неовлашћеним странама буде потпуно неразумљив, односно да не могу извући никакве информације.
* Шифрат (ciphertext)- резултат шифровања, излазана вредост трансфорамације отвореног текста.
* Алгоритам шифровања- скуп правила који се користи за шифроавање отвореног текста.
* Кључ шифровања (key)- од њега зависе операције алгоритма шифровања, улазна вредност као и отворени текст.
* Дешифровање (decryption)- операција којом се из шифрата добија отворени текст
* Алгоритам дешифровања- скуп правила који се користи за дешифровање шифрата.



***Слика 2.1.:*** *Процес шифровања и дешифроавања*

Дужина кључа, односно број симбола којима је представљен, зависи од конкретне имплементације алгоритма и представља један од параметара сигурности система. Сигурност криптографског система почива искључиво на тајности кључа. Сам алгоритам је јаван, односно опште познат. Овакав принцип се користи јер у колико сигурност ситема зависи од алгоритам постоји могућност да неко украде или на неки други начин дође до алгоритам, што се у пракси показало као вероватно. У том случају потребно је мењани комплетан систем, док у колико сигурност система зависи искључиво од кључа у колико неко на неки начин дође до кључа довољно је само променити кључ и систем може да настави да се користи без икаквих већих промена. Такође јавни алгоритми су доступни великом броју криптографа који могу да открију и прије слабости.

Криптографија мора да обезбеди следеће:

* Интегритет односно ведостојност података који који се шифрују, односи се на то да не дође до било какве неовлашћене измене или брисање податакака. У колико ипак некако дође до неовлашћене промене података неопходно је то детектовати, односно мора да постоји начин да се провери да ли су подаци мењани. Ову врсту напада могу да изврше и легални корисници који случајно или намерно покушавају да прекораче своја ограничења.
* Тајност, односи се на то да је садрђај података доступан само овалшћеним лицим односно онима који поседују кључ.
* Провера идентитета, оноси се на то да корисници пре приступа систему морају прво да се пријаве на систем.
* Немогућност избегавања одговорности, односи се на то корисник не може да порекне нешто за шта се зна да је баш он урадио. Врло битно код савремених система где велики број новчаних трансакција обавља преко интернета.

Крипто-анализа је процес којим се из шифрата долази до информација о отвореном тексту без познавање кључа, односно само на основу шифрата. У ширем смислу обухвата и проучава слабости крипотграфских елемеата, као што су, на пример, хеш функције или протоколи аутентификације. Технике криптоанализе називају се напади. Постоје два типа напада потпуна претрага кључева и скраћени напади. Потпуна претрага кључева представља испобавање сваке могуће вредности кључа из птостора кључева. Простор кључева је скуп свих могућих вредности кључева. Скраћени напади захтевају мање времена и обраде од потпуне претраге кључева. Неки од скраћених напада су:

* Напада на основу познавање шифтаза- у колико алогритам шифровања има слабости могуће је доћи до поруке само анализом шифрата.
* Напада на основу познатог дела отвореног текста- поруке које се шаљу најчешће имају унапред познате делове, као што су заглавља или слично. То може да се искористи за дешифроваењ остатка шифрата.
* Напад на основу изабраног отвореног текста- нападач себи шаље одређену поруку док корисник није за рачунаром(lunchbreak attack). Како нападач зна отворени текст и шифрат може да дође до кључа и дешифрује остале шифрате.
* Напда на основу адаптивно одабраног отвореног текста- нападач бира отворени текст, уради анализу, затим изабере нови текст на основу резулата. У колико добије очекиване резултате значи да је на добром путу.

Шифарски систем је сигуран у колико је најефикасније напад потпуна претрага кључева. Простор кључева мора да буде довољно велики.

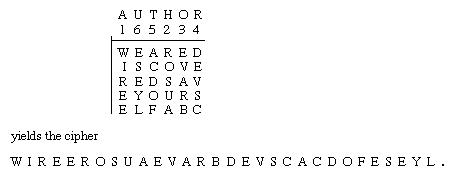
Постоје две врсте сигурности: рачунска и безусловна. Систем је рачунски сигуран у колико је цена разбијања шифрата већа од вредности шифроване информације или ако је време разбијања дуже од времена које информација треба да буде тајна. Систем је безусловно сигуран у колико не може да буде разбијен ни уз неограничене ресурсе и неограничено време.

Безусловно сигурна шифра је шифра код које се ни потпуном претрагом кључева не може доћи до отворене поруке.

## Класична криптографија

Класичне шифре које су коришћене у далекој историји су: шифре транспозиције, шифре супституције, One-time pad и кодне књиге. Оне су коришћене само за шифровање порука док се савремене шифре користе и за друге ствари, између осталог и проверу интегритета поруке, проверу идентитета пошиљаоца или примаоца и слично.

Шифре транспозиције заснивају се на замени редоследа слова у тексту. Први пут је коришћена у Спарти и звала се Скитала. Текст поруке је писан на кожној траци обмотаној око дрвеног штапа. Кад је порука написан штап се одмота и добије се текст у којем је редослед слова испремештен. Такву поруку је могао да дешифрује или прочита само онај који има штап исте дебљине или зна дебљину штапа. У данашње време се за ову врсту шифровања котисте матрице и посте транспозиција колона и редова. Код транспозиције колона текст се пише редом у матрицу одређених димензија, у колико нема довољно слова да се попуне сви редови додају се док матрица није скорз попуњена. Затим се текст чита из колона, да би се проширио простор кључева додаје се тајна реч која која одређује којим редоследом се читају колоне. Код двоструке транспозиције се користи исти принцип с' тим што се премутују и редови. Међутим ова врста шифровања је веома слаба. Кључ представљају димензије матрице, а димензије морају бити такве да њихов производ треба да буде једнак или већи дужини поруке. Шифрат садржи само симболе из отвореног текста и исте је дужине, а не би требао да садржи никакве информације из отвореног текст јер оне олакшавају крипто аналитичарима претрагу кључева. Крипто-анализ овако шифрованих порука се заснива на принципу „подели па владај“. Могуће је из шифрата сазнати тајну реч, што одмах даје једу од две димензије матрица, након чега је лако доћи до отвореног текста јер се лако могу одредити димензије матрице и зна се редослед којим се читају колоне. Чак и без тога уз довољан број покушаја могу да се сазнају димензије матрице и након тога тајана реч па и отворен текст. Чак и само коришћењем оловке и папира уз довољно времена и труда је могуће разбити овакву шифру, док је употребом рачунара то занемарљиво лако.



***Слика 2.1.1.:*** *Пример шифре транспозиције*

Код шифре замене слова или групе слова се систематски мењају другим словима или групама слова. Постоје моноалфабетске шифре, код њих се слова мењају увек истим словима, хомофоне шифре, код њих се уводи елемнт случајности, полиграмске шифре, код којих се замена врши над већом групом слова, и полиалфабетске, код којих свако слово може да се мења већим бројем слова. Најпознатија шифра просте замене је, већ поменута, Цезарова шифра у којој се свако слово поруке мења словом које је од њега удаљено одређени број места у алвабету или азбуци, с' тим да последња слова долазе на места првих. Алгоритам шифровања је:

**pi** i-то слово отовореног текста

**ci** i-то слово шифрата

**KE** кључ, број помераја

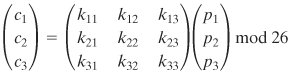
С' обзиром да је могуће померити за највише 26 места ова шифра има простор кључева од 26, што је врло мало, па је врло лако пробати све могуће вредности и доћи до отвореног текста. Због тога се дошло на идеју да кључ може бити било која премутација слова, што простор кључа повећава на 26! што је приближно 4\*1026. Међутим и даље је могуће разбити ову шифру коришћењем фрекфенцијске анализе. Наиме постоје на итернету јаве статитике фреквенције појављивања слова у неком произвољном тексту. Могуће је одрадити фреквенцијску анализу над шифратом и упоредити са неком другом анализом и на основу тога доћи до закључка којом логиком односно по ком правилу су слова мењана.

Због тога код хомофоне шифре шифат може да садржи више од 26 знакова а слова могу да се мењају већим бројем слова. У главном се слова са највећом фекфенцијом, односно са највећим бројем појављивања, предсатављају на више начина чиме се шири статистика шифрата. За правилан рад битно је пронаћи слова са највећом фрекфенцијом а затим одредити правило када доделити једну од две вредности за исто слово отовореног текста. С'обзитом на то да је шифрат треба да прикрије статистичке особине отовореног текста, он би требао да има особине случајног низа а постизање случајности. Шифрат је такође слаб на нападе познатог дела текста.

Код полиграмских шифара групе солва се шифрују заједно, а представници су плејфер шифра и Хилова шифра. Плејфер шифра шифрује по два слова заједно и код ње је могућа фрекфенцијска анализа али у дужим порукама. Хилова шифра је настала 1929. и осмислио ју је Лестер С. Хилл, користи блокове од три или више слова и заснована је на линеарним једначинама. Због оптужби да је његов систем због превелике компликованости неупотребљив, Хил је направио машину која шифрује користећи његов алгоритам али никада нију успео да је прода. Највећи његов допринос је уоптреба математике у осмишљању и анализирању криптографских система. Пример шифровања:

* Слова алфабета се представљају бројевим од 0 до 25.
* Текст се представи бројевима и подели на блокове дужине n. Сваки блок се обележи као р1,р2,р3,...рi где је i редни број блока, и запише као вектор димензија nх1.
* Шифрат блока се добије множењем по модулу 26 инвертибилне матрице А димензије nхn са вектором рi. A je кључ и њену вредност треба изабрати случајно из скупа инвертибилних матрица димензија nхn. Такође се мора водити рачуна да је могуће изтачунати инверзну матрицу, односно да детрминанта не сме да буде 0.
* Дешифровање се врши множењем блока шифрата инверзном матрицом матрице А (А-1) по модулу 26.

Због линеарности је рањива на нападе познатог дела текста. У колико је су позанте димензије кључа А и довољан број блокова отвореног текста и шифрата онда:



***Слика 2.1.2.:*** *Принцип шифровања коришћењем Хилове шифре*

Шифре полиалфабетске замене корите више алфабета за шифровање, уопштено свако слово може да има своје правило за шифровање. Представник је Вижнерова шифра код које се свако слово помера за m, за које важи 0≤m≤25, места у десно.

Шифровање:

Дешифровање:

Отпорнија је на статистичку анализу али је и она могућа ако се погоди дужина кључа. У том случају се шифрат уписује у матрицу дужине m и над сваком колоном засебно се врши статистичка анализа.

Клод Шенон је 1949. увео појам „перфектна шифра“, који представља шифарски ситем код које шифрат не даје никакве информације о отвореном тексту или кључу. У простору отворених поруке *Р,* отворени текст *х* појављује се са вероватноћом *р(х),* у том случају постоји кључ који пресликава сваки отворени текст у сваки шифрат са подједнаком вероватноћом. Енторпија се користи као мера за количину информација садржаних у поруци. Количина информација зависи од вероватноће, што је нека порука вероватнија то је количина информација мања и обрнуто. Информациони садржај неке поруке *х* рачуна се по формули:

Пошто се користи логаритам са основом два, бинарни логаритам, количина информација поруке представља број бита који је потребан за опис поруке. У колико имамо поруку *Р* која садржи *хi* независних симбола са вероватноћом *рi*, i=0,…,n-1, ентропија предсавља просечну количину информација која је садржана у појединачним порукама и рачуна се формулом:

One-time pad су увели Гилбер Вернам и Џозеф Мауборгн 1917. и представља доказано сигурну шифру. Принцип рада шифре је крајње једноставан. Над отовреним текстом се изврши операција „ексклузивно или“ (XOR) кључем К, сви подаци су у бинарном формату. Поступак дешифровања је идентичан.

Шифровање:

Дешифровање:

Сигурност се заснива на случајности кључа као и на његовој једнократној употреби. У пракси је веом тешко генерисати чисто случајне низове и са становишта ктиптографије захтевано основно свјоство кључа се одређује ентропијом. Пжељна вредност ентропије је и она важи за вероватноћу појављивања нула и јединица , под условом да постоју униформа дистрибуција унутар бинарне секвенце. ОТР је оказано безбедна ако се користи по дефинисаним правилима:

* Шифрат не даје никакву информацију о отвореном тексту.
* Сваки отворени текст исте дужине подједнако је вероватан.
* Кључ мора бити „случајан“ и може се користити само једном.
* Кључ је познат само примаоцу и пошиљаоцу.
* Кључ је исте дужине као и порука.
* Механизам интегритета не постоји.

Због тешкоће генерисања и дистрибуције кључа ова шифра се не користи.

Кодна књига је скуп речи или фраза коју могу да се користе и доделе им се кодови. Постоје две једн за шифровање и друга за дешифровање, а кључ је сама кодна књига. Сигурност се заснива на самој физичкој сигурности књига. Цимерманов телграм је једна од најпознатијих кодних књига у историји а његово дешифровање било је повод за улазак САД у први сетски рат. Кодне књиге су подложне статистичкој анализи. Ради веће безбедности користе се и адитивне кодне књиге које садрже много случајних бројева који се сабирају са кодним речима и дају коначан шифрат. Почетно место из адитивне кодне књиге се договара док почетно место из кодне књиге одређује пошиљалац. Почетно место се назива индикатор порука или иницијализациони вектор.

## Симетрични крипотсистеми

Крајем Другог светског рата почела је употреба рачунара у криптоанализи што је даље довело до употребе рачунара и за шифровање и дешифровање. Код модерне криптографије отворени текст се преставља низом битова и алгоритми за шифровање и дешифровање раде над њима. И даље се користе комбинације транспозиције и супстуције. Две највеће предности модерене криптографије у односу на класичну, односно две највеће предности које доноси употреба рачунара, јесу могућнос реализације много сложенијих алгоритама за шифровање и дешифоровање и много већа брзина шифровања и дешифровања.

Симетрични крипто системи користе исти кључ за шифровање и дешифровање, тако да обе стране које комуницирају морају пре почетка комуникације на неки сигуран и тајан начин да размене кључ. Постоје две врсте симетричних шифара а то су секвенцијалне и блоковске.

### Секвенцијалне шифре

Као што је већ споменуто ОТР представља беузсловно сигурну шифру, код које чак и потпуном претрагом кључева није могуће разбити шифрат, односно могуће је како резултат за различите кључеве добити различите отворене текстове и проблем је утврдити који је прави. Међитим због својих недостатака се не користи. Један од највећих проблема у коришћењу ОТР шифре јесте генерисање случајног низа, односно кључа. Један он начина јесте мерење неких природних појава као што су атмосферски шум, термички шум или остале ектерне електромагнетне и квантне појаве, на пример космичко позадинско зрачење или радиоктивни распад представљају добар извор случајности. Проблем код овог начина генерисања кључа јесте време које је потребно да се сакупи довољно мерења да би се добио кључ довољно велике дучине, јер кључ треба да буде исте дужине као и порука.

Због тога се у крпто системима као кључ у главном корисете псеудо случајни низови. Псеудо случајни низови су по својим статичким особинама исти као и прави случајни низови. Једина разлика јесте у томе што су преиодични, односно у неком моменту ће генератор псеудо случајних низова почети да генерише исте вредности. За иницијализацију генератора псеудослучајних низова се користи кључ К који је прави случајни низ, и који је много краћи од радног кључа, односно генерисаног низа.

Процес самог шифровања се своди на XOR операцију над отвореним текстом и радним кључем, а шифорвање потпуно исти процес само што се врши над шифратом и радним кључем. Функција секвенцијалне шифре је:

Где је К кључ, а S генерисани псеудослучајни низ.

За псеудослучајни низ , отоврени текст генерисање шифрата и затим дешифровање шифрата у отворени текст изгледа овако:

Сам алгоритам шифровања и дешифорвања сам по се би није у опште компликован. Снага ове шифре лежи у кључу. Постоји више решења за генераторе псеудослучајних бројева од којх је једао примена линеарних померачких регистара са повратном спрегом (ЛПР).

Линеарни померачки регистри обављају две функције. Прва је функција померачког регистра дужине n који сваки бит помера за једно место у десно. Друга функција је функција повратне спреге која на основу одређених бита у регисту одређује који ће се бит уписати на прво место. Излаз је најчешће један бит, најчешће крајњи десни, који је одређен почетним стањем и повратном спрегом. Периода померачког регистра је број бита излазне секвенце пре него што почне да се понавља. Померачки регистар дужине n може да има могућих почетних стања, не могу да буду све нуле, и правилним одабиром повратне спреге може се постићи пероида од бита. Непожељне особине линеарног померачког регистра су те што су првих n битаво јенднаки почетном стању. Такође је на основу бита излаза могуће реконструисати генератор.

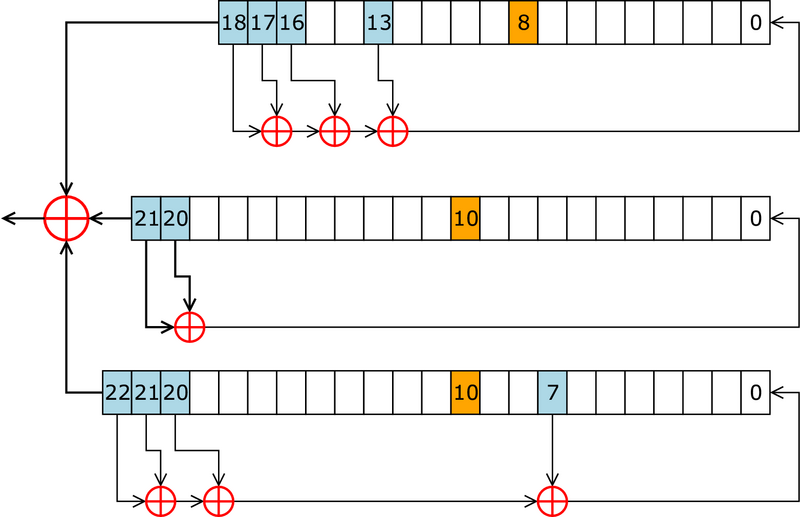
#### A5/1

Коришћен је у мобилној комуникацији за крипографску заштиту података код GSM заштитног протокола, тачније за безбедну комуникацију између телефона и базне сатанице. Алгоритам је чуван у тајности али је цурењем и реверзним инжењерингом доспео у јавност и показало се да алгоритам има озбиљних слабости.

А5/1 користи три линеарна померачка регистра са повратном спрегом, означићемо их као X,Y и Z. Регистар X се састоји од 19 битова, Y од 22 и Z од 23, и укупан збир свих битова је 64, колика је и дужина кључа К. Кључ је иницијална вредност ови регистара након чега регистри генеришу псеудослучајни низ, односно радни кључ који се користи за шифровање и дешифровање. За ту дужину кључа очекивани низ случајних битова је , мада од самог дизајна померачких регистара са повратном спрегом зависи дужина радног кључа и величина периоде.

Поступак генерисања кључа:

* , maj представља маријотетну вредност односно „два од три“.
* Ако је онда се рачуна  и помера Х а на место првог бита се уписује вредност .
* Ако је онда се рачуна и помера Y a на место провг бита се уписује вредност .
* Ако је онда се рачуна и помера Z а на место првог бита се уписује вредност .
* Бит радног кључа се рачуна као .



***Слика 2.2.1.1.1:*** *А5/1 регистри*

Алогритам се врло лако имплементира у хардвер и генерише битове кључа пропорционално брзини такта процесора. Број бита радног кључа који се могу генерисати из једног кључа од 64 бита је практично неограничен, али ће се сигурно у једном тренутку поновити.

#### RC4

RC4 је секвенцијална шифра која софтверски имплементира и генерише један бајт кључа у сваком такту. У основи овај алгоритам садржи једну табелу са свим пермутацијама једног бајта 0-255, и сваки пут када се генерише бајт радног кључа табела се мења тако да увек садржи пемутацију од . Иначе комплетан алгоритам ради над бајтовима.

Прва фаза је инцијализација табеле коришћењем кључа. Ако је табела означена као Ѕ а кључ као кеy, онда псеудокод за иницијализацију табле изгледа овако:

**for** i **from** 0 **to** 255

S[i] := i

**endfor**

j := 0

**for** i **from** 0 **to** 255

j := (j + S[i] + key[i [mod](https://en.wikipedia.org/wiki/Modulo_operation) keylength]) mod 256

swap values of S[i] and S[j]

**endfor**

Ѕ[i] и key[i] су бајтови. Кључ може бити дужине онд 1 до 256 бајтова, најчешће је изеђу 5 и 40, и користи се само за иницијализацију пермутација.

Након овога се генеришу бајтови радног кључ. Псеудокод за генерисањ радног кључа изгледа овако:

i := 0

j := 0

**while** GeneratingOutput:

i := (i + 1) mod 256

j := (j + S[i]) mod 256

[swap values](https://en.wikipedia.org/wiki/Swap_(computer_science)) of S[i] and S[j]

K := S[(S[i] + S[j]) mod 256]

output K

**endwhile**

У свакој итерацији i се инкрементира, узима се i-ти елемент из табле, S[i], и додаје ј. Затим се мењају вредности i-тог и ј-тог елемента, S[i] и S[j], и њихов збир по модулу 256 се користи као идекс за узимање елемента који у ствари представља бајт радног кључа, К. Тај бајт се XOR-ује са бајтом отвореног текста у колико се шифрује, односно шифрата у колико се ради о дешифровању.

За правилан рад ове шифре неопходно је одбацити првих 256 бајтова радног кључа јер је на основу њих могуће реконструисати радни кључ. Такође је врло битно да кључ буде случајан.

Коришћен је у многим апликацијама укључујући заштитне протоколе SSL и WEP. Међутим од 2015. постоје спекулације да неке државне агенције могу да разбију RC4 када се користи у TLS протоколу. IETF (International Engineering Task Force) је објавио RFC 7464 којим забрањује употребу RC4 у TLS, Mozilla и Microsoft су објавили сличне препоруке.

### Блоковске шифре

У општем случају отворени текст се дели на блокове одређене фиксне дужине и генеришу се блокови шифрата исте дужине. Над блоком отовреног текста се више пута понавља функција одређени број рунди. Улазни параметри сваке рунде су кључ и излаз из претходен рудне, и сама функција зависи од њих.

Блоковске шифре комбинују својаства дифузије, конфузије и комплетности. Дифузија значи да на основу познавања пара блока отвореног текста и шифрата не може да се одреди блок отвореног теста неког другог блок шифрата. Захтевано је да мале промене у блоку отвореног текста доведу до непредвидљивих промен у датом блоку шифрата. Својство конфузије подразумева су сви кључеви подједнако вероватни у случају напада потпуном претрагом кључева. Комплетност подразуева да је сваки бит шифрата функција сваког бита кључа.

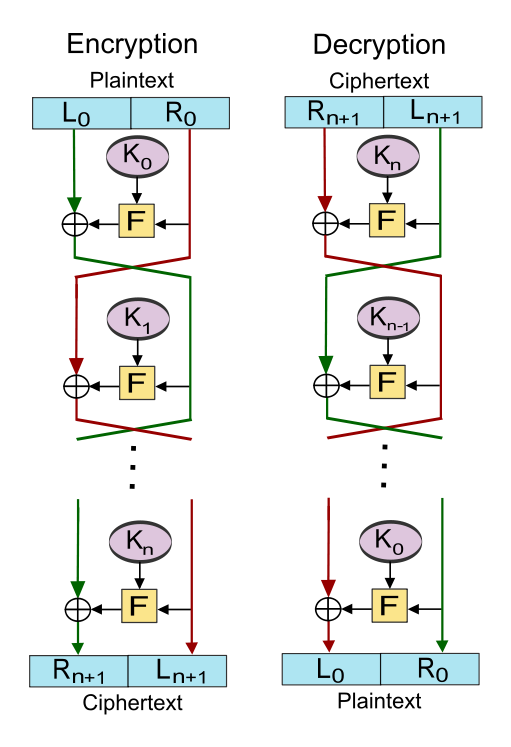
Блоковске шифре се најчешће реализују софтверски и треба да буду сигурне и ефикасне, што захтева висок ниво криптографског умећа.

#### Фејстел шифра

Названа је по Немачком криптографу Хорсту Фејстелу, који је радио за IBM и био пионир у развоју дизајна блоковских шифара. Фејстел шифра је дизајн блоковске шифре а не сама шифра. Његова иницијална истраживања су довел да развоја DES (Data Encryption Standard) алгоритма 1970.

Идеја и принцип рада Фејстел шифре су следећи:

* Отворени текст се дели на блокове одређене дужине (*P*).
* Сваки блок се дели на леви и десни део који су означени као *L* и *R*.
* У свакој рунди се рачуна , , где је *F* функција рунде а *Ki* подкључ који се добија комбиновањем битова кључа *K.*
* Шифрат је излаз последље рунде
* Дешифровање је инверзно
* *F* може да буде било која функциј али се збогбезбедности пажљиво бира.



***Слика 2.2.2.1.1:*** *Принцип рада Фејстел шифре*

#### DES

Развијен је „у мрачном добу компјутерсек ере“, односно седамдесетих, и базиран на IBM-овој Луцифер шифри.

Средином седеамдестих крипографија је била слабо позната ван војних и државних кругова, што је значило да корисници криптографије нису били упознати са стварном јачином зачтите криптографских производа, која је иначе била врло слаба. Али је било јасно да постоји потреба за заштитом дигиталних података, због све већег развоја и све веће употребе рачунара. У таквим условима је Национални биро за стандарде, или НБС, издао захтев за предлог нових шифара. Победник би постао стандард америчке владе и *de facto* индустријски стандард. Један он озбиљнијих кандидата на том конкурсу је била IBM-ова Луцифер шифра. Проблем је био у томе што у НБС-у није било стручњака за криптографију, па су морали да се окрену Националној сигурној агенцији, која се бави дизајнирањем и прављењем крипографских механизама за војску и владу, за помоћ.

Иако нису били одушевљени идејом да се мешају у ту причу ипак су се под притискуом умешали под условом да њихово учеће буде тајно. Када је информација о томе процурела појавиле се у се сумље да ће НСА оставити „задња врада“, backdoor, односно начин да разбију разбију шифру. Међутим за 30 година крипоанализе нису откривена ни једна „задња врата“ или нека друга врста пропуста у дизајну алгоритма.

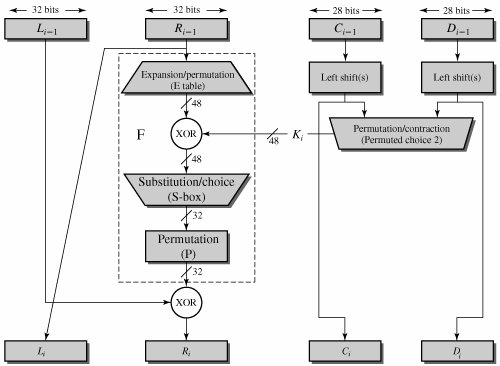
На крају Луцифер је постао DES уз неколико не баш суптилних промена, од којих су неке смањење кључа са 128 на 64 бита, од којих се 8 одбацује тако да је стварна дужина кључа 56 бита, и кутије замене или S-кутије.

Циљ је био да се уведе принцип конфузије, шифрат зависи од отвореног текста и кључа на сложен начин, дифузије, сваки бит шифрата је функција свих битова отвореног текста и свих битова кључа. Поред тога захтеван је и „лавински ефекат“, мале промене отвореног тексата доводе до великих промена шифрата, промена једног бита кључа или отвореног текста мења 50% битова шифрата.

DES је Фејстел шифра са следећим параметрима:

* Дужина блока отовреног текста је 64 бита.
* Дужина кључа је 56 бита.
* Има укупно 16 рунди пре генерисања једног блока шифрата.
* Дужина подкључа сваке рунде је 48 бита.

Функција *F* кодDES шифре се састоји од: експанзије (E-кутије), нелинеарних трансфорамације (S-кутије), пермутација (P-кутије), сабирању по модулу 2 и одабира бита кључа (К-кутија). Шифровање прати правила приказана у једначини:



***Слика 2.2.2.2.1.:****Шема DES алгоритма*

Експанзија (Е-кутија) пермутује и проширује свој улаз са 32 на 48 бита. Мења се редослед битова а неки се понављају ради постизања лавинског ефекта, тачно мапирање је одређено екпсанзионом табелом. Битови којие се добију као излаз се XOR-ују са 48 бита подкључа те рунде.

S-кутија је реализована као нелинеарна стриктура што их чини тешким за анализу и најважнијим елеменатом на којем почива безбедност DES алгоритма. Има их осам и улаза сваке Ѕ-кутије се деоводи 6 битова а на излазу се добије 4, чиме се величина враћа на 32 бита. Ѕ-кутија се посматра као матрица од 4 реда и 16 врста, са једном 4-битном вредности која може да се нађе на било којој од 64 позиција. Свака Ѕ-кутија има своју табелу са константим вредностима. Када се на улаз доведе 6 бита први и последњи се издвајају и конвертују у деццималну врендсот и они представљају индекс реда. Унутрашња четири бита се такође конвертују у децималну вреност и представљају индекс колоне. Пошто су довијени вредност колоне и реда узима се вредност из Ѕ-кутије, која се дели са 2 а остатак при дељењу се конверује у бинарну вредност. Када се добије излаз из свих Ѕ-кутија оне се пермутују на основу пермутационе матрице.

Р-кутија представља пермутацију излаза Ѕ-кутије по дефинисаном правилу.

К-кутије генеришу подкључеве , дужине 48 бита, за сваку рунду на основу кључа *К* дужине 56 бита. Генерисање се одвија у три фаза. У првој фази се прво одбацују крајње десни битови кључа. У следећем кораку кључ се представља пемутационом матрицом која се назива пермутациони избор 1. Битови се пермутују онако како је то задато у матрици иницијалне пермутације и као резулатат се добија скраћени и пермутовани кључ. У другој фази се генерише 16 подкључева, по један за сваку рунду. Кључ се дели на две половине леву и десну, дужине 28 бита, и свака се циклично помера одређени број пута. Тај нови кључ је улаз у последњу фазу. У последњној фази користи се изабрана пермутација 2, која садржи 48 позиција. За дате позиције се издвајају битови из кључа и као крајњи резулата се довија подкључ за рунду.

Како су линеране једначине лаке за решавање, сигурност DES алгоритма почива на једнином нелиеарном делу што су Ѕ-кутије, а унешто мањој мери и од распореда кључа. За 30 година крипо анализе није откривена ни један пропуст или backdoor, како се у почетку сумљало због умешасти НСА. Најевећа слабост овог алгоритма јесте у томе што је дужина кључа веома мала. Иако су развијени напади који теоријски захтевају мање времена и посла од потпуне претраге кључева, сви до сада направљени програми за разбијање DES-a користе потпуну претрагу кључева. 1993. Мајкл Винер је показао да је могуће наравити хардвер којим је могуће разбити DES нападом типа отоворени текст и то за 35 сати са буџетом од 100 000 долара, 3,5 сати са буџетом од милион долара и 21 минута са буџетом од од десет милона долара. 2006. тим са Универзитета Бохуму и Кил у Немачкој развио је COPACABANA, комерцијално доступну машину за разбијање DES-a. 2008. развијена је COPACABANA RIVYERA која је смањила време разбијања шифре на мање од јендог дана. Један оваква машина може да се произведе за 10 000 долара.

#### Троструки DES

Пров се одшло на идеју да се користи двоструки DES, односно да се порука два пута шифрује чиме се дуплира величина кључа а самим тим и порстор кључева. Међутим нападом типа „meet-in-the-middelе“ могуће је разбити двоуструки DES мање више слично DES-у. Напад је типа изабраног отвореног текста. У општем случају то изгледа овако:

Циљ је пронаћи оба кључа. За свако могуће К1 шифрује се Р и резултати се записују у табелу у облику парова. Затим се претпостави К2 и дешифру је С1 и на тај начин се добије неко Х. Ако Х постоји у табели пар кључева се проверава над новим отовореним текстом и шифратом, у колико се добије очекивани резултат онда је пронађен прави пар кључева. На овај начин се број неопходних покушаја да би се сигурно дошло до кључа смањује са 2112 на 256. Иако је величина потребне табеле огромна овај напад је изводљив.

Због несигурности двоструког DES-a појавио се троструки. И даље се користи два кључа, јер је дужина кључа од 112 бита довољна, али је прицнип рада другачији. Шифровање и дешифровањ код троструког DES-a су дефинисани на следећи начин:

Разлог због којег се шифрује једним кључем, затим дешифрује другим па поново шифрује првим је компатибилност, јер троструки DES код којег су оба кључа идентична, је у ствари обичан DES.

#### АЕЅ

Током деведестих дошло се до закључка да је DES превише несигуран јер је простор кључева од 56 бита једноставно превише мали. Неки посебно развијени програми су могли да открију кључ за неколико сати. Национални институт за стандарде и технологију издао је захтев за криптографски предлог алгоритма или AES (Advanced Encryption Standard). Било је много квалитетних предлога али је алгоритам под називом „Rijndael“ победио.

АЕЅ алгоритам има компликовану математичку структуру и није Фејстел шифра. Отпоран је на све познате нападе, веома је брз, могућ је паралелни дизајн као и имплементација на моногим процесорима и паметним картицама. Параметри АЕЅ-а су:

* Величина кључа је 128, 198 или 256 бита.
* Величина блока отовреног текста је 128, 198 или 256 бита (не зависи од величине кључа).
* Има 10 до 16 рунди, зависно од дужине кључа

Свака рунда се састоји од четири функције:

* ByteSub, нелинерани слој.
* ShiftRow, слој линеарног мешања.
* MixColumns, нелинеарни слој.
* AddRoundKey, додатни слој кључа.

АЕЅ улазне податке третира као матрице димезије 4х4, односно све операције шифровања и дешифровања се врше над матрицама. Улазни подаци се копирају у матрицу стања над којом се врше разне операције, а завршно стање те матрце, односно вредност матрице након извршених свих операција, се копира у излазни блок података, односно блок шифрата.

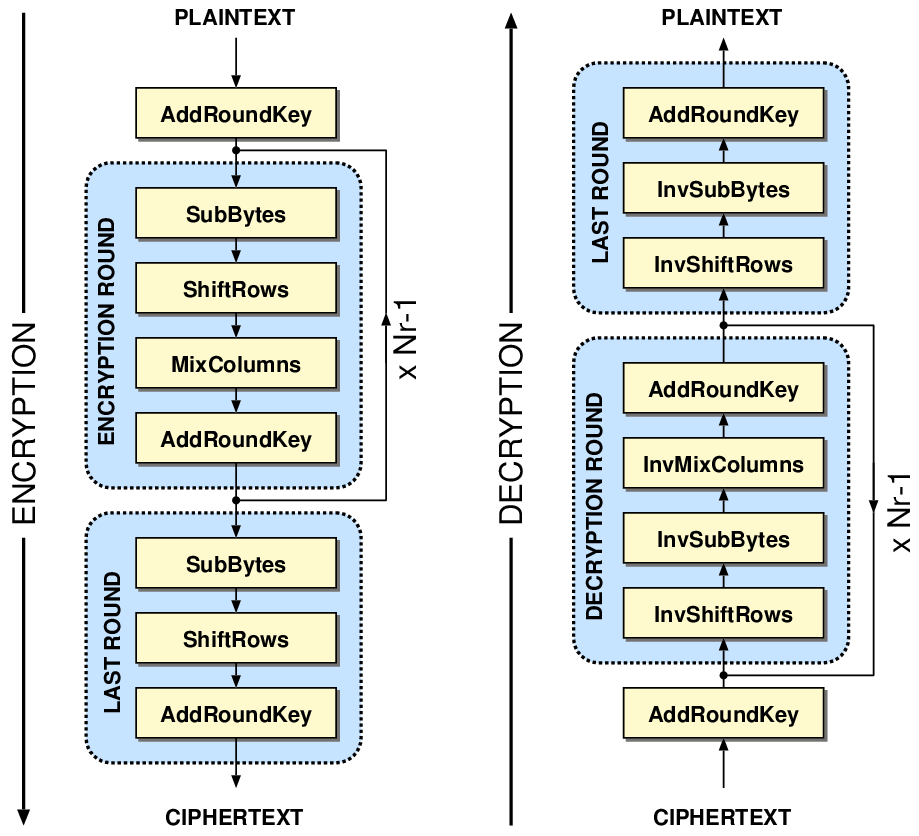
ByteSub је нелинерна функција и сличан је Ѕ-кутији код DES алгоритма. Може да се посматра као нелинеарна, инвертибилна композиција две математичке функције или као табела. Табела садржи 256 унапред дефинисаних вредности. Блок улазних података се записује у облику матрице а затим се нелинеарном функцијом добије друга матрица. На основу виша четири бита се адресира врста а на основу нижа четири колона табеле. Вредност бајта у коју ће се вредност из полазне матрице пресликати се добија из табеле.

ShiftRow је укратко циклично померање бајтова у последња три реда матрице. Сваки бајт друге колоне се циклично помера за једно место у лево, слично се бајтови друге и треће колоне циклично померају за два и три места у лево. Уопштено бајтови у реду n се циклично померају за n-1 места. Постоји мала разлика за блокове од 256 бита, први ред се и овде не помера али се други, трећи и четврти померају за 1,3 и 4 места. Међутим ово не важи за АЕЅ јер он не користи блокове те величине, али уопштено важи за Rijndael алгоритам. Битност овог корака је у томе да се избегне да колоне буду линеарно независне.

MixColumn је операција мешања колона матрице. Колоне се премештају по дефинисаним правилима и при томе се свака XOR-ује са одговарајућом матрицом. Цела операција је нелинеарна али инвертибилна. Мешање колона, MixColumn, и померање редова, ShiftRow, представљају основни извор дифузије.

AddRoundKey je XOR-овање добијене матрица са подкључем. Слично DES-у од целог кључа се бира подкључа за сваку рунду. Подкључ се пресликва у матрицу, која је исих димензија као и матрица која се добија као резулатат претходних операција, и која се затим XOR-ује са добијеном матрицом.

Све операције су инвертибилне и као резулата цео алгоритам је инвертибилан, због чега је могуће дешифровати исто као што се шифрује.



***Слика 2.2.2.4.1.:*** *Шема шфорвања и дешифровања АЕЅ алгоритмом*

#### Режим рада блоковских шифара

Режим рада блоковских шифара представља начин на који се блоковска шифра примењује на низ узастопних блокова. Наиме у колико се користи кључ *K* блоковкса шифра је исто што и кодна књига јер једнозначно преслликава блокове отвореног текста у блокове шифрата. Водећи се идејом кодне књиге дошло се до коришћења блоковских шифара у режиму „електронске кодне књиге“ или EBC. Шифровање и дешифровање изгледају овако:

Међутим овај режим се не користи јер има озбиљних сигурностних пропуста. Како се блокови отовреног текста једнозначно пресликавају у блкове шифрата, може се на основу идентичних блокова шифрата закључити да су њихови блокови отвореног текста такође идентични. У колико нападач зна део отвореног текста свако подударање са познатим блоком открива нови блок. Чак и ако не зна део отвореног текста шифрат не би требао да одаје никакве информације о отвореном тексту. Ова слабост се превазилази коришћењем CBC или CTR режимом рада.

Код CBC, или Chiper Block Chaining, суседни блокови се уланчавају. Сваки блок се XOR-ује са шифратом претходног блока, за први се генерише нека случајна вредност која се назива иницијални вектор, *IV*. Као резултат блокови истог отвореног текста дају различите шифрате. Формула за шифровање је:

Формула за дешифровање:

Међутим због уланчавања у колико се шифрат преноси неким комуникационим каналом и у току преноса дође до неке грешке, неки бит промени своју вредност, отоврени текст може да постане непоправљив. У колико само један бит промени вредност онда два блока отвореног текста постају неисправан, јер сваки блок зависи само од два узастопна блока шифрата.

Код CTR, или режима бројача, се такђе користи иницијални вектор *IV* али на другачији начин. Иницијални векотр се шифрује и XOR-ује са блоком отвореног текста. Затим се иницијални векотр увећа за 1 и поступак шифровња се и XOR-овања се понавља за следећи блок, и тако за све блокове. Формула за шифровање је:

Формула за дешифровање је:

## Асиметрични криптосистеми

Други тип криптосистема представљају асиметрични кирптосистеми односно ситеми са јавним и тајним кључем. Они неће бити детаљно разматрани као симетрични али ће бити објашњене неке основне особине и карактеристике због њихове употребе код SSL протокола.

За разлику од симетричниш криптосистема асиметрични користе два кључа, јавни и тајни. Јавни и тајни кључ се генеришу у пару, јер порука шифрована јавним кључем може да буде дешифрована једино његовим паром, односно тајним кључем који је парњак тог јавног кључа. Ови крипто системи су у главном засновани на математичким проблемима који немају ефикасно решење као што су једносмерне функције са замком. Кандидати за такве функције су дисктерни експоненет, његова инверзна функција је дикретни логаритам, производ целих бројева, његова инверзна функција је факторизација добијеног броја. Особине ових функција су те да се лако рачунају у једном смеру али им је тешко израчунати инверзну функцију осим ако се не зна тајна вредност.

Још једна битан разлика у односу на симетричне криптосистеме је та да није потребан тајни и сигуран начин за размену кључа. Користи се Дафи-Хелманов алгоритам за размену кључева. Принцип рада је следећи:

* Нека је *р* велики прост број и *q* тако да се за свако може наћи *n* тако да .
* Корисници, зваћемо их Алиса и Боб ради лакшег објашњавања, али и нападач знају *p* и *q.*
* Алиса бира тајну вредност *а*, која је неки велики случајан цео број.
* Боб бира тајну вредност *b,* која је такође велико сличајан цео број.
* Алиса јавно шаље Бобу.
* Боб јавно шаље Алиси.
* Обоје могу да израчунају вредност што у ствари представља кључ.
* Нападач зна и им може једино да израчуна што није кључ.

RSA алгоритам који се данас корити функционише на следећи начин. Ако су *С* шифрат *М* отворени текст, *N* и *е* представљају јавни кључ а *d* тајни, шифровање изгеда овако:

A дешифровање:

Вредности *N*, *е* и *d* треба да буду такве да за свако *М<N.*

Јавни и тајни кључ се генеришу на следећи начин:

* Изаберу се два велика проста броја *p* и *q.*
* Формира се
* Израчунава се (Ојлерова функција).
* Бира се *е* тако да је мање од и узајамно просто са .
* Нађе се *d* тако да је , односно
* *e*, *N* представљају јавну кључ.
* *d* представља тајни.

Дигитални потпис се добија када се отоврени текст шифрује тајним кључем. Тада се дигитални потпис дешифрује јавним кључем чиме се потврђује идентитет пошиљаоца. У главном се за дигитални потпис и шифровање користе различити парови јавног и тајног кључа.

Пробел који се јавља јесте утврђивање идентитета, односно власника јавног кључа. У ву сврху се користе стране од поверења које треба да гарантује за све регистроване кориснике и да на захтев потврди њихов идентитет. Сертификациона страна је трећа страна од поверења. Корисници генеришу сертификат који сертификациона страна потписује.

С' обзиром на то да су системи са јавним кључем два до три пута спорији од симетричних систем, дужина кључа код RSA је 1024 бита а код AES-а 256, системи са јавним кључем се користе за размену симетричног кључа па се онда прелази на симетричну криптографију.

## TLS и SSL

TLS (Transport Layer Security) и његов претходник SSL (Secure Socket Layer) су криптографски протоколи који обезбеђују сигурну комуникацију преко мреже. Постоји више верзија оба, иако је TLS настао од SSL-a и у великоје мери га заменио и даље се често користи термин SSL када се мисли на TLS, и користе се у велоком броју апликација за претраживање интернета, електронску пошту, слање факса преко интернета, ћаскање преко интерента и VOIP (Voice-Over-IP), већи сајтови користе TLS за сигурну комуникацију између сервра и претраживача.

Развије је од стране Netscape Communicaions деведесетих, за сигуран пренос података између Netscape Navigator претраживача и сервера, како би обезбедили заштићеност осетљивих података као што су бројеви кедитних картица. Прва вертзија никада није јавно објављена а верзија 2.0 која је објављена фебруара 1995. је имала велики број сигурносних пропуста. Верзија 3.0 је захтевала потпуни редизајн и објаљена је 1996. Иако никада није био формално стандардизован постао је де факто стандард за сигурност у комуникацији преко интернета. Након што га је IETF (Internet Engineering Task Force) стандардизовала верзија 3.1 је објављена као TLS 1.0, име је промењено из правних разлога, и увдена су побољшања која су побољшала слабости. Многи од напада на SSL су засновани на самој имплементацији, али POODLE слабаост је позната мана саме 3.0 верзије која је омогућавала нападачу да дешифрује осетљиве податке као што су колачићи (cookies). TLS није слаб на ову врсту напада али има још предности које га чине сигурнијим од SSL-а. Најновија верзија је 1.2, а документација за 1.3 је објављена.

SSL се састоји од два под-протокола: протокол за „руковање“ и протокол записа који омогућују успостављање шифроване SSL конекције. У току „процеса иницијалног руковања“ клијент шаље листу алгоритама које може да користи и насумичну вредност *Rа* сервер, да би потврдио свој идентитет, клијенту шаље свој дигитални сертификат, насумичну вредност *Rb*и одабрани алгоритам. Сертификат је у Х.509 формату који дефинише Public-Key Cryptography Standards (PKCS). Аутентификација користи шифру са јавним кључем да валидира сертификат и потврди да је сервер тај којим се представља. Када је сервер аутентификован постављају се подешавања за шифровање и кључ који ће се користити за шифровање свих података које буду размењивали, чиме се обезбеђује поверљивост података и интегритет.

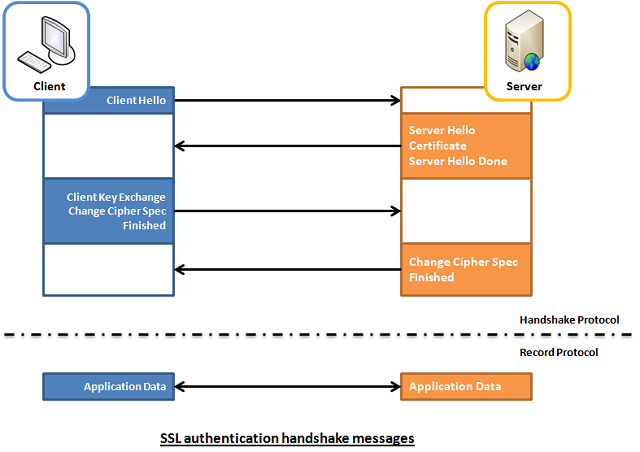
Када клијент код сертификационог тела потврди примљени сертификат он једино зна да има прави сертификат и јавни кључ, али још увек није сигуран да комуницира са правим сервером. Генерише тајну вредност S коју шифрује јавним кључем који је добио од сервера и кључ *К*:

*h* предстаља хеш функцију. На све претходно размењен поруке, које се шљау да би сервер био сигуран да су све порује успешно размењене, клијент дописује договорену константу *CLNT* и *К,* и на крају рачуна хеш вредност свега тога.

Добијена вредснот се шифрује договореним алгоритмом и кључем *К.* Клијент затим шаље шифровану вредност *Ѕ* и

Када сервер прими ове поруке може да израчуна, дешифрује, *Ѕ* и пошто зна *Rа*и *Rb*може да израчуна *К* и дешифрује . Затим формира поруку тако што на све већ размењене поруке допише константу *SRVR* и кључ *К* и клијенту шаљ вредност:

И у овом случају све послате поруке се шаљу ради сигурности да су све успешно размењене. Након овога корисник је сигуран да је комуницира са правим сервером и да су све поруке успешно размењене. Вредност *К* се користи за даљу комуникацију и из ње се рачуна барем шест кључева: два за шифровање, два за интегритет и две иницијалне вредности по једна за пријем и предају.



***Слика 2.3.1.:*** *Поступак аутентификације код SSL протокола*

Читав овај процес је невидљив за корисника и не захтева никакво ангажовање са његове стране. У случају претраживача, у колико је успостављена сигурна комуникација, URI се мења из HTTP у HTTPS. Уколико ипак дође до напада или сертификат није могуће аутентификовати корисник добија упозорење али и даље има могућност да настави комуникацију.

Конекција осигурана SSL протоколом има следећа својства:

* Конекција је приватна, сигурна, јер се симетрична криптографија користи за шифровање података који се шаљу. Кључ који се користи је једниствен за сваку конекцију и заснован на дељеној тајни која се договоара на почетку сесије (). Алгоритам који ће се користит за шифровање је такође договорен пре почетка слаља података. Договарање дељене тајне је такође сигурно, нападача не може да дође до ње чак и ако се нађе у сред конекције и комуникација за време преговарања не може да се промени а да то не буде детектовано.
* Обе стране у комуникацији могу да буду аутентификоване коришћењем криптографског система са јавним кључем. Ово може да буде опционо, али се у главном захтева да се барем једна страна, сервер, аутентификује.
* Конекција обезбеђује интегритет јер свака послата порука укључује и кода за аутетнтификацију поруке како би се спречио недетектовани губитак или измене података током преноса.