МАТЕМАТИЧКА ГИМНАЗИЈА

МАТУРСКИ РАД

из предмета ПРОГРАМИРАЊЕ И ПРОГРАМСКИ ЈЕЗИЦИ

Криптографија

Ученик
Лука Королија, $IV_Б$

Ментор
Милош Арсић

Београд, мај 2024.

Садржај

1. Увод.. 2
2. Основе криптографије.. 4
3. Класична криптографија.. 9
4. Модерна криптографија.. 13
    4.1 RSA.. 13
    4.2 DES... 15
5. Закључак.. 19
Литература.. 20

1. Увод

Од како су људи научили да пишу, један од најважнијих проблема им је био како да заштитe то што записују од људи којима ти текстови нису намењени. Из ове потребе настала је криптографија, која је постала још важнија у модерној ери, а данас је неопходна за многе системе које свакодневно користимо. Од имејла до онлајн куповине, криптографија је саставни део многих модерних система.

Криптографија је наука која се бави методама трансформисања података у циљу да се прикрију информације које садрже. У криптографију спадају и методе прикривања и откривања прикривених или шифрованих информација. Сама реч „криптографија” потиче из грчких речи kryptós, што значи „скривено”, и gráfein, што значи „писати”.

За даље разумевање техника криптографије потребно је упознати се са најважнијим појмовима, наиме појмовима отвореног текста, шифровања, шифрата и дешифровања, што је најлакше учинити кроз пример. Нека особа А има тајну поруку коју жели да пошаље особи Б, и особа А жели да буде сигурна да ће само особа Б моћи да прочита ту поруку чак и у случају да је нађе непожељно треће лице, особаВ. Пре него што комуникација између особа А и Б почне, они морају да се договоре око **шифре**, тј. криптографског алгоритма и **кључа** које ће да користе. Појам кључа ће бити објашњен даље у раду, јер за овај пример није неопходан. Порука коју особа А жели да пошаље назива се отворени текст, јер је потпуно читљива свакоме ко је има. Да та порука не би била читљива, неопходно је шифровати је помоћу неке шифре, тј. алгоритма за шифровање. Текст који настаје применом шифре на отвореном тексту назива се **шифрат**. Особа А шаље шифрат особи Б, и у случају да особаВ затекне тај шифрат, у идеалном случају они не би били у могућности да га прочитају без знања које само особе А и Б треба да поседују. Кад шифрат стигне до особе Б, он и даље није разумљив, зато особа Б мора да га дешифрује помоћу алгоритма за дешифровање да би добила полазну поруку. Конкретнији примери ће бити дати даље у раду.

Вештина и наука која се бави шифровањем порука назива се криптографија, али постоји и вештина откривања шифрата која се назива криптоанализа. Криптографија и криптоанализа су две гране опширније математичке области која се назива криптологија. Иако су у прошлости шифре биле једноставне и једино што их је чувало од откривања је било само знање о функционисању шифре, данас су много компликованије и претежно се ослањају на математику, зато су већина данашњих криптолога математичари. Због комплексности модерних шифара, оне се већином примењују помоћу рачунара, а исто тако криптоаналитичари претежно користе рачунаре за откривање шифара. Због доминантности рачунара у процесу шифровања и дешифровања порука, шифра се обично сматра сигурнијом што је њено откривање помоћу рачунара мање ефикасно.

Овај рад ће се бавити представљањем основних концепата криптографије као и пар изабраних алгоритама класичне и модерне криптографије. У првом делу рада ће бити представљене основе криптографије и пример криптоанализе у сврхе илустровања како шифре могу бити дешифроване, јер криптографија не би имала сврху без опасности дешифровања шифри. У другом делу рада ће бити представљени неки концепти и алгоритми класичне криптографије као увод у даље, компликованије алгоритме. У трећем делу рада радиће се о модерној криптографији и DES и RSA алгоритмима.

2. Основе криптографије

Историјски, криптографија се састојала од једноставног шифровања и дешифровања података, које би користила два лица у циљу очувања тајности информација у порукама. Као што је већ поменуто, у прошлости је довољно било да алгоритам буде тајна да би порука била нечитљива. Како напредује криптоанализа, тако се развија и криптографија, зато велики број модерних алгоритама користи кључеве. Концепт кључа је настао да би се умањио утицај знања функционисања алгоритма за шифровање на дешифровање шифрата, јер кроз употребу кључа тајност престаје да зависи искључиво од тајности самог алгоритма за шифровање. Кључ је обично број или текст који служи као параметар при шифровању и дешифровању поруке, чиме се додаје додатни ниво заштите. У зависности од тога да ли су кључ за шифровање и дешифровање зависни или независни, алгоритми се редом називају симетрични и асиметрични, где су асиметрични сигурнији.

Симетрични алгоритми су алгоритми чији кључеви су зависни, тј. један кључ се може израчунати полазећи од другог. Најчешће код симетричних алгоритама су кључеви исти, али је довољно да се кључ за дешифровање може израчунати полазећи од кључа за шифровање да би алгоритам био симетричан. Ови алгоритми се такође називају алгоритми са тајним кључем, јер особа која зна кључ и коришћену шифру може лако да дешифрује шифрат, због чега кључ мора да остане тајна како би порука остала тајна, што је само један корак сигурније од шифри без кључа код којих је довољно знати само алгоритам. Симетрични алгоритми могу даље да се поделе на ланчане шифре и на блоковске шифре. Ланчане шифре шифрују отворени текст бит по бит, док блоковске шифре шифрују по груповима битова који се зову блокови.

Асиметрични алгоритми су алгоритми чији кључеви су независни и знати кључ којим је шифрат шифрован није довољно да се дешифрује шифрат направљен оваквим алгоритмом. Асиметрични алгоритми се такође називају и алгоритми са јавним кључем. Јавни кључ, што је кључ који се користи за шифровање, се тако назива јер, као што је пре речено, знати тај кључ није довољно да се дешифрује шифрат, па се може доставити свима јавно и свако може да шифрује било коју поруку помоћу тог кључа. Да би се порука дешифровала потребно је знати тајни кључ, што је кључ за дешифровање, и то је једини кључ помоћу ког се шифрат може дешифровати. У неким случајевима уз шифрат се прикључује и такозвани дигитални потпис. Дигитални потписи су поруке шифроване тајним кључем, које се дешифрују јавним, да се утврди идентитет особе која поруку шаље.

Шифрат мора да испуњава три ставке да би успешно и тајно пренео поруку. Прва ставка је тајност, која се добија шифровањем отвореног текста сигурним алгоритмима. Друга ставка је интегритет поруке, што значи да порука не губи своје значење након шифровања, и може се проверити помоћу хеш функција и дигиталних потписа. Трећа ставка је провера аутентичности лица која комуницирају, и ту су важни дигитални потписи и инфраструктура јавног кључа. Сматра се да је шифра квалитетнија и сигурнија што је ефикаснија (тј. брза и захтева мало ресурса, као што су меморија или капацитет процесора), док њено дешифровање захтева напор који је много већи, чинећи криптоанализу толико неефикасном и непрактичном да је ефективно немогућа.

Један од најпознатијих и најранијих примера појаве криптографије налази се у шифрованим порукама Јулија Цезара, који је користио симетричну шифру касније названу по њему за размену порука са својим генералима. Као најједноставнији начин шифровања поруке, Цезарова шифра подразумева замену сваког слова у поруци словом N слова удаљеном у азбуци. Ако узмемо као пример да је N=3 и ако посматрамо слова иза слова у отвореном тексту (померамо свако слово за 3 улево). У случају да нема слова за 3 улево од слова у отвореном тексту, враћамо се на крај азбуке и настављамо да бројимо. Ова трансформација би се могла представити таблицом са две азбуке, где је једна померена за 3 улево, што би изгледало овако:

Отворени текст	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Шифровани текст	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

Табела 1. Приказ отвореног и шифрованоог текста за N=3

Ако узмемо отворени текст „TEKST”, применом Цезарове шифре добићемо следеће (користимо енглески алфабет јер је тај најприсутнији на рачунарима, те је лакши за коришћење у програмима):

Отворени текст	T	E	K	S	T
Шифровани текст	Q	B	H	P	Q

Табела 2. Шифровање текста „TEKST”

Цезарова шифра је такође пример једне врсте шифара које се називају шифре замене. Шифре замене замењују делове текста другим текстом, најчешће слова другим словима, за разлику од транспозиционих шифара које само „мешају” делове текста, али их не замењују ни са чим.

Иако је ово вероватно најједноставнији криптографски алгоритам, такође је један од најкоришћенијих, поготово заједно са другим алгоритмима, и зато ћемо помоћу њега демонстрирати два могућа криптоаналитичка напада на неку шифру.

Два најлакша начина да се дешифрује Цезарова шифра су брут-форс метода (енгл. чиста сила, нагађање кључа, или у овом случају помераја) и анализа учесталости. Једна од слабости Цезарове шифре је њен мали простор кључева, јер број могућих вредности за N је (број слова у азбуци) - 1, пошто ако је N једнако броју слова у азбуци коју користимо, шифрат се неће разликовати од отвореног текста. Друга слабост Цезарове шифре потиче из тога што је она проста шифра замене, и та слабост ће служити као пример за други напад. Пајтон програм за овај део рада је у прилогу рада и примењен је у следећем примеру.

Рецимо да имамо следећи шифрат (отворени текст је на енглеском језику, јер је фреквенција појављивања слова у енглеском језику најпознатија, што ће бити потребно за други напад):

Xqwlo prghuq wlphv, fubswrjudskb uhihuuhg doprvw hafoxvlyhob wr „hqfubswlrq", zklfk lv wkh surfhvv ri frqyhuwlqj ruglqdub lqirupdwlrq (fdoohg sodlqwhaw) lqwr dq xqlqwhooljleoh irup (fdoohg flskhuwhaw). Ghfubswlrq lv wkh uhyhuvh, lq rwkhu zrugv, prylqj iurp wkh xqlqwhooljleoh flskhuwhaw edfn wr sodlqwhaw. D flskhu (ru fbskhu) lv d sdlu ri dojrulwkpv wkdw fduub rxw wkh hqfubswlrq dqg wkh uhyhuvlqj ghfubswlrq. Wkh ghwdlohg rshudwlrq ri d flskhu lv frqwuroohg erwk eb wkh dojrulwkp dqg, lq hdfk lqvwdqfh, eb d „nhb". Wkh nhb lv d vhfuhw (lghdoob nqrzq rqob wr wkh frppxqlfdqwv), xvxdoob d vwulqj ri fkdudfwhuv (lghdoob vkruw vr lw fdq eh uhphpehuhg eb wkh xvhu), zklfk lv qhhghg wr ghfubsw wkh flskhuwhaw. Lq irupdo pdwkhpdwlfdo whupv, d „fubswrvbvwhp" lv wkh rughuhg olvw ri hohphqwv ri ilqlwh srvvleoh sodlqwhawv, ilqlwh srvvleoh fbskhuwhawv, ilqlwh srvvleoh nhbv, dqg wkh hqfubswlrq dqg ghfubswlrq dojrulwkpv wkdw fruuhvsrqg wr hdfk nhb. Nhbv duh lpsruwdqw erwk irupdoob dqg lq dfwxdo sudfwlfh, dv flskhuv zlwkrxw yduldeoh nhbv fdq eh wulyldoob eurnhq zlwk rqob wkh nqrzohgjh ri wkh flskhu xvhg dqg duh wkhuhiruh xvhohvv (ru hyhq frxqwhu-surgxfwlyh) iru prvw sxusrvhv. Klvwrulfdoob, flskhuv zhuh riwhq xvhg gluhfwob iru hqfubswlrq ru ghfubswlrq zlwkrxw dgglwlrqdo surfhgxuhv vxfk dv dxwkhqwlfdwlrq ru lqwhjulwb fkhfnv.

Први напад који ћемо покушати је брут-форс. Ова метода је једноставна и не захтева много објашњења. Користећи Цезарову шифру на шифрату док мењамо N да узима све вредности од 1 до 25 (јер енглески алфабет има 26 слова), наићићемо на отворени текст и приметићемо да је N при шифровању било 3 и да је померај био ка десно.

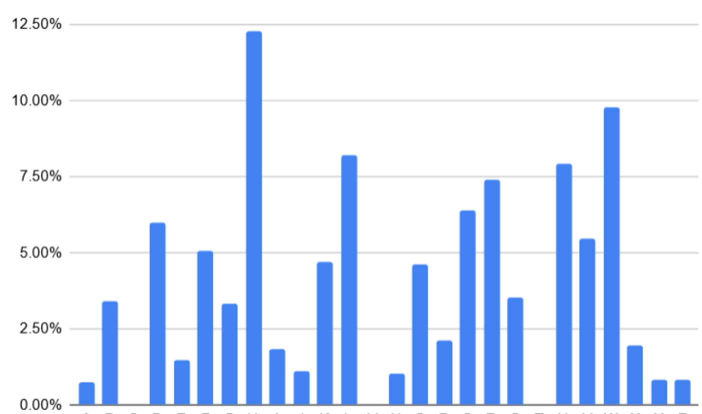
Други напад који ћемо покушати је анализа учесталости. Овај напад ћемо такође поједноставити, јер се рад првенствено бави криптографијом. Бројећи колико се пута свако слово појављује у шифрату можемо направити таблику учесталости:

A	B	C	D	E	F	G	H	I	J	K	L	M
8	37	0	65	16	55	36	133	20	12	51	89	0
0.74%	3.42%	0.00%	6.00%	1.48%	5.08%	3.32%	12.28%	1.85%	1.11%	4.71%	8.22%	0.00%

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
11	50	23	69	80	38	0	86	59	106	21	9	9
1.02%	4.62%	2.12%	6.37%	7.39%	3.51%	0.00%	7.94%	5.45%	9.79%	1.94%	0.83%	0.83%

Табела 3. Учесталост сваког слова у шифрату

У таблици, и на графику десно може се приметити да се у овом тексту највише појављује слово „H”, а након њега је најучесталије слово „W”. Да бисмо могли да направимо претпоставке о томе која слова ова представљају, морамо прво да знамо која слова су генерално најучесталија у језику у ком је отворени текст. Табела учесталости слова у енглеском језику је следећа.

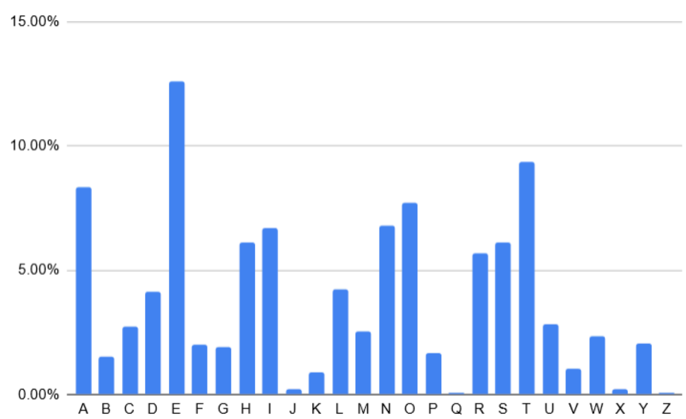


Слика 1. Учесталост сваког слова у шифрату

A	B	C	D	E	F	G	H	I	J	K	L	M
8.34%	1.54%	2.73%	4.14%	12.60%	2.03%	1.92%	6.11%	6.71%	0.23%	0.87%	4.24%	2.53%

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6.80%	7.70%	1.66%	0.09%	5.68%	6.11%	9.37%	2.85%	1.06%	2.34%	0.20%	2.04%	0.06%

Табела 4. Учесталост слова у просечном тексту Енглеског језика [3]



Слика 2. Учесталост сваког слова у просечном тексту Енглеског језика

Како су у енглеском језику најучесталија слова редом „E” и „T”, што видимо из табеле 4. и слике 2, можемо прво да претпоставимо да слова „H” и „W” у шифрату одговарају наведеним словима. Ово можемо да претпоставимо јер Цезарова шифра, пошто је шифра замене, свако исто слово у отвореном тексту мења истим одређеним словом, те се учесталост одговарајућих слова не мења у шифрату. Када заменимо ова слова, у првој реченици примећујемо да уместо „wr” добијамо „tr”. Следећа претпоставка коју ћемо направити заснива се на томе да је „to” најучесталија двословна реч која почиње на „t”. Тиме ћемо заменити слово „R” словом „O”. Док замењујемо слова у шифрату, морамо да пазимо на то која слова смо већ заменили, да не бисмо слово које претпостављамо да је део отвореног текста заменили у некој од следећих претпоставки, ово се лако може учинити на пример бојењем замењених слова. Пошто знамо да је за шифровање овог шифрата коришћена Цезарова шифра (или ако не знамо можемо да претпоставимо по чињеници да се свако слово до сада налазило 3 слова удесно од претпостављеног слова отвореног текста), можемо да померимо свако слово у супротном смеру за исти број којим је шифрат шифрован, што је у овом случају 3. Заиста, померањем сваког слова за 3 улево ћемо моћи да прочитамо отворени текст, који је изостављен због обима рада. Овај случај је намерно изабран за брзо решење, у реалним случајевима често је потребно много више рада да би се дешифровао шифрат.

Као што се види из примера, класичне шифре су рачунарским алатима веома лаке за дешифровање. Иако су класичне шифре слабе, веома су важне како и историјски тако и за модерну криптографију, јер се на њима заснивају или су од њих настале многе модерне шифре. У следећем поглављу биће представљено још пар класичних алгоритама уз примере и програме.

3. Класична криптографија

Класична криптографија обухвата старе историјске методе шифровања које су се користиле пре настанка модерних рачунара. Ове методе су започеле основе по којима су настале савремене, модерне криптографске технике. Иако класични алгоритми често подразумевају једноставне начине шифровања који се могу израчунати ручно (како су се и морали рачунати у време када су настали), и даље су корисни и данас. Представићемо на примерима једну шифру замене, исте врсте које је и Цезарова шифра коју смо већ представили, и две транспозиционе шифре, као и њихове предности и мане.

Шифре замене (супституционе шифре) су неке од најстаријих шифара икада коришћених. Ове шифре замењују сваки елемент отвореног текста другим елементом. Постоје три врсте шифри замене:

1. Шифре просте замене – шифре код којих свако слово отвореног текста увек одговара истом слову шифрата ако се користи исти кључ. Пример ове врсте је Цезарова шифра.
2. Хомофонске шифре – шифре код којих на основу неког правила једно слово отвореног текста може одговарати различитим знаковима шифрата. Разлог зашто је ова врста шифре замене развијена је један од криптоаналитичких метода које смо представили у прошлом поглављу, наиме метод анализе учесталости. Учесталијим словима се може дати више различитих знакова, а мање учесталим словима се може дати мање, и тиме се може уједначити учесталост сваког знака и смањити могућност напада анализом учесталости.
3. Полиалфабетске шифре – шифре које се формирају помоћу више цифара просте замене. Ове шифре имају више кључева, и сваки кључ служи да се шифрује једно слово. Кад се остане без кључева креће се поново од првог кључа. Број слова који се шифрује пре преласка назад на први кључ назива се периодом шифре. Генерално, поруке шифроване дужим периодом ће бити теже дешифровати.

Шифра коју ћемо сада разматрати је пример полиалфабетске шифре и назива се Вижнерова шифра. Основа Вижнерове шифре је примена различите Цезарове шифре у зависности до редног броја слова. Ово се постиже текстуалним кључем. Свако слово отвореног текста се шифрује Цезаровом шифром померањем удесно исто толико пута колики је редни број слова у кључу које по реду одговара слову отвореног текста (бројање креће од нуле). Ако је кључ краћи од отвореног текста продужава се тако што се понавља од почетка. Ова шифра је у своје време била револуционарна и било је потребно преко три века година да се шифра дешифрује. Нека за пример узмемо отворени текст „TEKST” и кључ „ABC”. Представићемо прво ова два текста у таблицама заједно са редним бројевима сваког слова кључа у алфабету. Програм који ово ради аутоматски је у прилогу рада.

Отворени текст	T	E	K	S	T
Кључ	A	B	C	A	B
Редни број слова	0	1	2	0	1

Табела 5. Помоћна табела за шифровање

Следећи корак је шифровање сваког слова појединачно Цезаровом шифром удесно за број који је у трећем реду ове таблице. Ово ћемо такође представити таблицом за лакше приказивање:

Отворени текст	T	E	K	S	T
N	0	1	2	0	1
Шифрат	T	F	M	S	U

Табела 6. Шифровање текста „TEKST"

Да бисмо дешифровали поруку, само треба да узмемо -N и применимо исту методу.

Постоји и други начин представљања Вижнерове шифре кроз таблице, а то је кроз tabulu rectu, што је таблица која почиње записивањем азбуке у првом реду, па затим у сваком следећем реду записивањем азбуке померене за једно слово улево. Ова метода је мање практична програмски па неће бити представљена.

Предности ове шифре су њена отпорност на брут-форс наспрам Цезарове шифре због коришћења шифре. Мана ове шифре је што је и даље подложна анализи учесталости ако је кључ превише кратак или је шифрат превише дугачак. У случају да је кључ потпуно насумичан и исте дужине као отворени текст, ова шифра постаје теоретски немогућа за откривање.

Шифре транспозиције не мењају ни једно слово отвореног текста, али им мењају редослед. Прве машине намењене за шифровање су користиле ову врсту шифри. Те машине су радиле помоћу механичких дискова који су се звали ротори и који су примењивали шифру транспозиције на азбуци. Најпознатија машина која је користила роторе је Енигма, која је била једно од најважнијих оружја коришћених у другом светском рату, и чије дешифровање је скратило рат годинама и спасило незамисливо много живота, као пример утицаја криптографије на свачији живот.

Прва шифра транспозиције коју ћемо представити је шифра ограде (зигзаг шифра). Назив „шифра ограде" добила је по начину на који се врши шифровање, наиме отворени текст при шифровању се нађе у позицији која подсећа на ограду са хоризонталним даскама. Ова шифра зависи само од једног броја и то је број „даски" које ћемо користити при шифровању. Ову шифру је најлакше приказати кроз пример па ћемо одмах прећи на један. Рецимо да имамо отворени текст „SIFRA TRANSPOZICIJE ZIGZAG". Започињемо тако што направимо N хоризонталних линија (или редова, ако радимо у таблици), у овом примеру 3 линије. У те линије (редове) напишемо наш отворени текст дијагонално прво ка доле па онда ка горе и то понављамо док не дођемо до краја поруке.

S				A				N				Z				J				G			
	I		R		T		A		S		O		I		I		E		I		Z		G
		F				R				P				C				Z				A	

Табела 7. Табела за шифровање шифром ограде

Након овога остаје само да прочитамо шифру ред по ред и запишемо шта смо добили, што је у овом примеру: „SANZJG IRTASOIIEIZG FRPCZA”, што се може и записати као „SANZJGIRTASOIIEIZGFRPCZA”. Дешифровање ове шифре је једноставно ако знамо број линија који је коришћен. У случају да не знамо који број линија је кориштен, можемо једноставно да искористимо брут-форс методу на ову шифру, јер је простор кључева веома мали. Ова шифра се због ове слабости сматра слабом.

Последња класична шифра о којој ће бити говорено у овом раду је шифра транспозиције по колонама. За ову шифру се поново мора писати отворени текст у таблицу по редовима, али овај пут величина таблице зависи од дужине речи коју изаберемо као кључ. Након што запишемо отворени текст у таблицу треба само да га прочитамо, али сада читамо по колонама уместо по редовима. Ред по ком читамо колоне зависи од редних бројева слова кључа у азбуци. Пример ове шифре ћемо такође представити преко таблице. Кључ ће нам бити „KLJUC”, а отворени текст ће бити „SIFRA TRANSPOZICIJE”.

Кључ	K	L	J	U	C
Редослед слова у азбуци	3	4	2	5	1
	S	I	F	R	A
		T	R	A	N
	S	P	O	Z	I
	C	I	J	E	

Табела 8. Табела за шифровање шифром транспозиције по колонама

Сада можемо да прочитамо колоне редом од најмањег броја до највећег и добијамо „ANI FROJS SCITPIRAZE”. Порука се може дешифровати тако што се запише у исту ову таблицу по колонама па прочита по редовима. Иако је ова шифра мало сигурнија, и даље се сматра слабом јер је подложна бројним нападима.

Кроз историју, ове класичне шифре су играле веома важне улоге у чувању тајни како личних тако и државних, али како су се времена мењала и наука напредовала, постале су недовољно сигурне. Иако су државни чиновници за време ренесансе сматрали да је Вижнерова шифра немогућа да се дешифрује, ми данас знамо да је дешифрујемо за неколико секунди. Упркос свему томе ове шифре граде основу данашње криптографије и довеле су до развића много компликованијих и сигурнијих шифри. У следећем поглављу ћемо две такве шифре да опишемо и представимо.

4. Модерна криптографија

Пре почетка 20. века, криптографија се углавном бавила прикривањем информација лексикографским путем. Једини циљ криптографије је био да се порука учини нечитљивом за људе. Од настанка првих рачунара је криптографија проширила обим и сада се у великој мери служи математичким поддисциплинама. Развој електронике и дигиталних рачунара, заједно са применом математике, омогућили су много сложеније шифре. Значајна разлика између класичне и модерне криптографије је да модерни алгоритми шифрују текст у бинарном формату, док класични алгоритми шифрују само писане текстове на језику. Ова чињеница је веома утицала на криптоанализу, која се до тада често ослањала на лингвистичке и лексикографске особине поруке.

Поменуто је да постоје две врсте симетричне шифре: ланчане шифре и блоковске шифре. Блоковске шифре, као што су DES и AES, шифрују поруку по блоковима фиксиране величине, где је блок група битова. Најчешће су блокови величине 64 или 128 битова, довољно велики да отежају криптоанализу, и довољно мали да шифровање не буде превише споро. Блоковске шифре су корисне кад се зна величина текста који се шаље и могуће је обрадити га по деловима исте величине. Блоковске шифре су сигурније од линијских, али не могу да се примењују на текстове чија величина није позната, на пример током комуникација у реалном времену. Линијске шифре, као што је RC4, шифрују поруку бит по бит, и због тога су идеалне у ситуацијама кад подаци непрекидно долазе и потребна је брза шифра. Линијске шифре генеришу линију кључа преко кључа, на коју се примењује XOR операција са отвореним текстом, чиме се добија шифрат.

4.1 RSA

RSA (Rivest-Shamir-Adleman, назван по својим ауторима) је најпознатији асиметрични криптографски алгоритам. RSA је откривен јавности 1977., исте године које је DES (следећи алгоритам који ћемо разматрати) постао међународни стандард за шифровање података. Пошто је асиметрични алгоритам, то значи да је шифра за дешифровање различита од шифре за шифровање. RSA налази честу употребу сигурном преносу података, дигиталним потписима и протоколима размењивања кључева. Сигурност RSA алгоритма заснива се на практичној тежини факторисања производа два велика проста броја. Не постоји ни један познати начин да се RSA дешифрује без кључа ако је довољно велик. RSA је релативно спор алгоритам, звог чега се ретко користи за директно шифровање података. Уместо директног шифровања, RSA се користи за сигуран пренос кључева који се користе при бржим, обично симетричним алгоритмима.

Овај алгоритам се састоји од четири корака:

1. Генерисање кључа – прво се изаберу два насумична велика проста броја (од 100 па чак до 200 цифара) p и q. Затим се израчуна $n = pq$, што је број који ће служити као модуло за оба кључа. Дужина броја n ће представљати дужину кључа. Овај број ће,

за разлику од p и q, који треба да буду тајни, бити део јавног кључа. Даље треба израчунати Ојлерову функцију од броја n, која је једнака броју бројева мањих од n, који су узајамно прости са n. Израчунавање тог броја је лако, јер је n производ два проста броја, чија је последица да је $\varphi(n) = (p - 1)(q - 1)$, што је лако проверити. Следећи корак је изабрати цео број e такав да важи $1 < e < \varphi(n)$, и да су e и $\varphi(n)$ узајамно прости. Најчешће изабран број у овом кораку је $2^{16} + 1 = 65537$, а најмањи могући је 3. Овај број је такође део јавног кључа. Последње што треба да се нађе је број d из једначине $d \equiv e^{-1} \pmod{\varphi(n)}$. Број d ће бити експонент тајног кључа.

Јавни кључ се састоји од броја n и експонента e. Тајни кључ се састоји од експонента d. Сем броја d, вредности p, q и $\varphi(n)$ такође морају остати тајне, јер се преко њих могу израчунати све остале вредности.

2. Дистрибуција кључева – Особа Б која прима поруку мора особи А послати свој јавни кључ да би особа А могла шифровати поруку. Кад особа Б добије поруку искористи свој тајни кључ (тајни експонент d) да је дешифрује, и не дели тај кључ ни са особом А.
3. Шифровање – Први корак шифровања је претворити поруку M (недопуњени отворени текст) у цео број m (допуњени отворени текст), такав да важи $0 \leq m < n$. Ово особа А уради унапред договореним инверзибилним протоколом који се зове шема допуњавања. Затим особа А израчуна шифрат помоћу јавног кључа особе Б решавањем једначине $c \equiv m^{\mathrm{e}} \pmod{n}$. Особа А сада може да пошаље c особи Б.
4. Дешифровање – Особа Б може да дешифрује m из c користећи њен тајни експонент d, тако што израчуна једначину $c^{\mathrm{d}} \equiv (m^{\mathrm{e}})^{\mathrm{d}} \equiv m \pmod{n}$. Кад особа Б нађе m може да добије M инверзном применом договорене шеме допуњавања.

Ако узмемо на пример да желимо да шифрујемо поруку $m = 12$, за вредности $p = 41$ и $q = 61$, процес би ишао овако: Прво нађемо $n = pq = 41*61 = 2501$, затим $\varphi(n) = (p - 1)(q - 1) = 40*60 = 2400$, затим било које e тако да важи $1 < e < \varphi(n)$, ако узмемо $e = 17$ остаје нам да израчунамо $d \equiv e^{-1} \pmod{\varphi(n)}$, $d = 1553$. Јавни кључ је ($n = 2501$, $e = 17$), а тајни кључ је ($n = 2501$, $d = 1553$). Да бисмо шифровали $m = 12$ морамо израчунати $c \equiv m^{\mathrm{e}} \pmod{n}$, $c = 12^{17} \pmod{2501} = 1059$. Да бисмо ово дешифровали треба да израчунамо $c^{\mathrm{d}} \equiv (m^{\mathrm{e}})^{\mathrm{d}} \equiv m \pmod{n}$, тј. $m = c^{\mathrm{d}} \pmod{n} = 1059^{1553} \pmod{2501} = 12$.

Обрнутом применом јавног и тајног кључа се овај алгоритам може користити за дигиталне потписе. Тајни кључ се може користити да се дешифрује порука намењена само особи којој припада тај тајни кључ, или да се шифрује порука која је намењена да свако може да је дешифрује, али коју може шифровати само особа са тим тајним кључем.

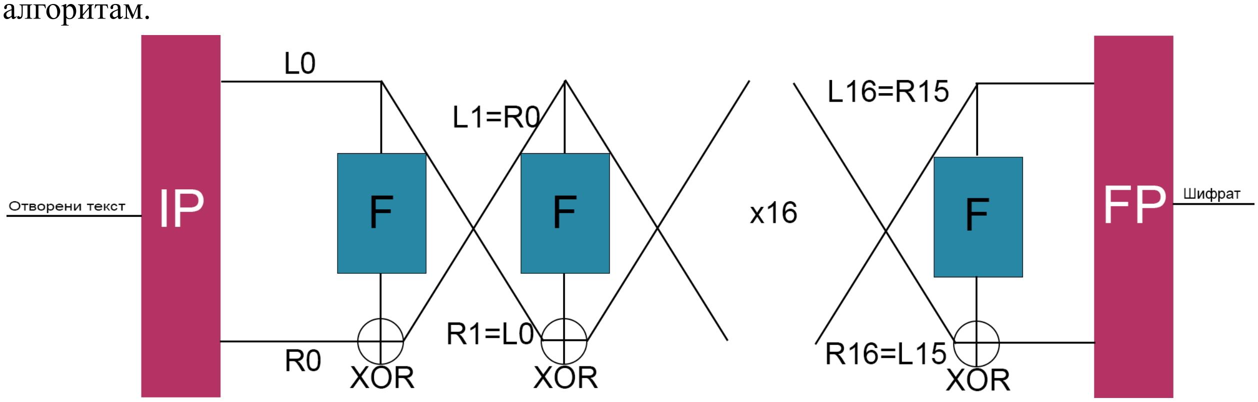
RSA је основа многих важних сигурносних протокола, као што су SSL, TLS или PGP. За разлику од овог алгоритма, који се и данас користи, следећи је већином избачен из употребе, али је важан развоју криптографије колико и овај, ако не и више.

4.2 DES

DES (Data Encryption Standard, енгл. стандард за шифровање података) је један од најкоришћенијих криптографских алгоритама на свету. Иако више није званични стандард за шифровање порука због коришћења кратке шифре (64 бита, од којих 8 служи за проверавање грешака и оставља само 56 битова за шифровање), што га чини несигурним за употребу данас због напредака у криптоанализи, имао је велик утицај на напредак криптографије. Постао је међународни стандард 1977. године, и држао је ту позицију до 2001. године кад га је заменила новија шифра звана AES (Advanced Encryption Standard, енгл. напредни стандард за шифровање).

DES је симетрична блоковска шифра. Ова шифра узима низ од 64 бита отвореног текста и трансформише га низом компликованих операција у други низ битова шифрата који је исте дужине. Дешифровање шифрата шифрованог помоћу DES се врши помоћу кључа који се такође састоји од 64 бита, али последњих 8 битова се одбацује јер они служе за проверавање грешака, те кључу ефективно остаје 56 битова. Кључ се преноси у форми 8 бајтова чији последњи бит служи за проверавање грешке. Дешифровање овим алгоритмом има исту структуру као и шифровање, тако да се исти хардвер и софтвер може користити и у шифровању и у дешифровању. Као и остале блоковске шифре, DES сам по себи није довољно сигуран за употребу, и зато се практично користи само као један део процеса шифровања поруке.

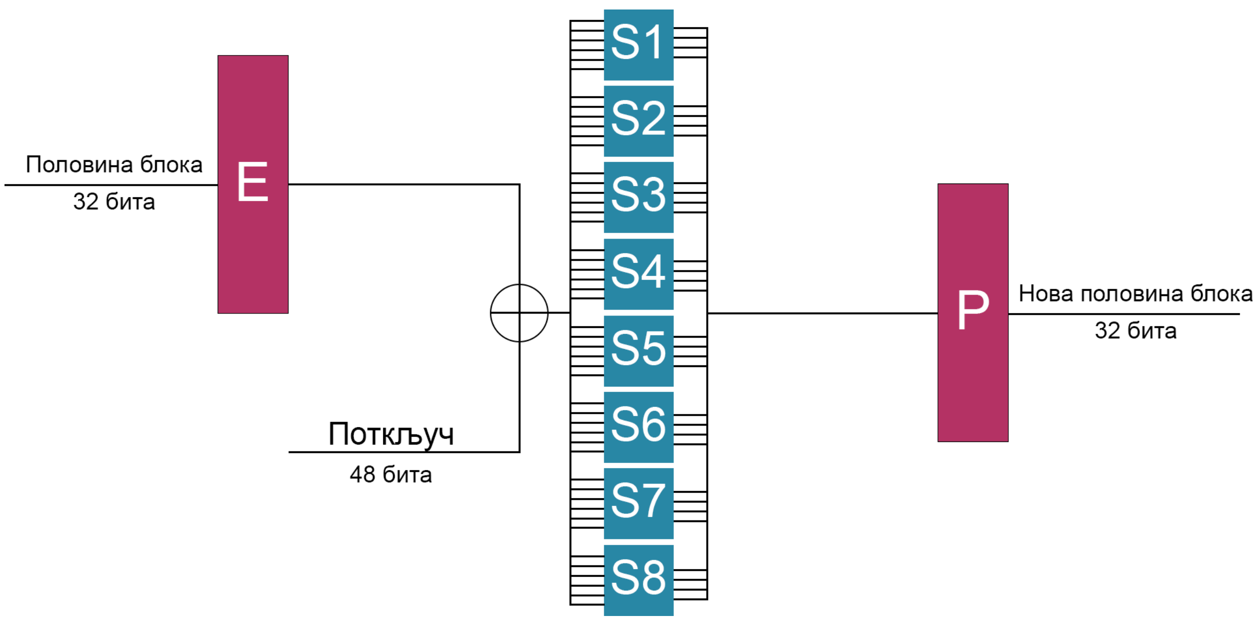
DES шифрује поруку на основу 16 идентичних фаза процесирања, које се називају рунде. Једна рунда се састоји од две трансформације текста, смене и пермутације које су одређене кључем. После прве фазе се блок раздваја на две половине од по 32 бита које се процесују независно једна од друге. Последња фаза је инверзна првој, што значи да се поништавају и само 16 фаза између њих су криптографски значајне. Алгоритам се заснива на две врсте трансформација, на „конфузију" и „дифузију", за које се сматрало пре још тридесет година пре настанка овог алгоритма да су неопходне за сигуран и практичан алгоритам.



Слика 3. Приказ Фајснерове структуре у DES

Структура алгоритма је приказана на слици 3. Прва пермутација текста је IP, а последња FP. Како ове две пермутације нису криптографски значајне оне неће бити разматране. Након прве пермутације блок се раздваја на два дела исте величине (32 бита).

У првој рунди се процесује један део, након чега делови мењају места и процес се понавља. Ово наизменично процесирање назива се Фајселова шема. Фајселова шема омогућава да су процеси шифровања и дешифровања што сличнији. Једина разлика у дешифровању од шифровања је да се поткључеви примењују обрнутим редоследом (више о поткључевима ће бити ускоро). Ова шема веома поједностављује имплементацију алгоритма јер нема потребе за различите алгоритме за шифровање и дешифровање. ⊕ симбол означава XOR операцију. F функција обрађује део блока заједно са делом кључа, и резултат те функције се спаја са другим делом блока. Блокови затим мењају места пре следеће рунде. Након последње рунде, делови блока мењају места поново.



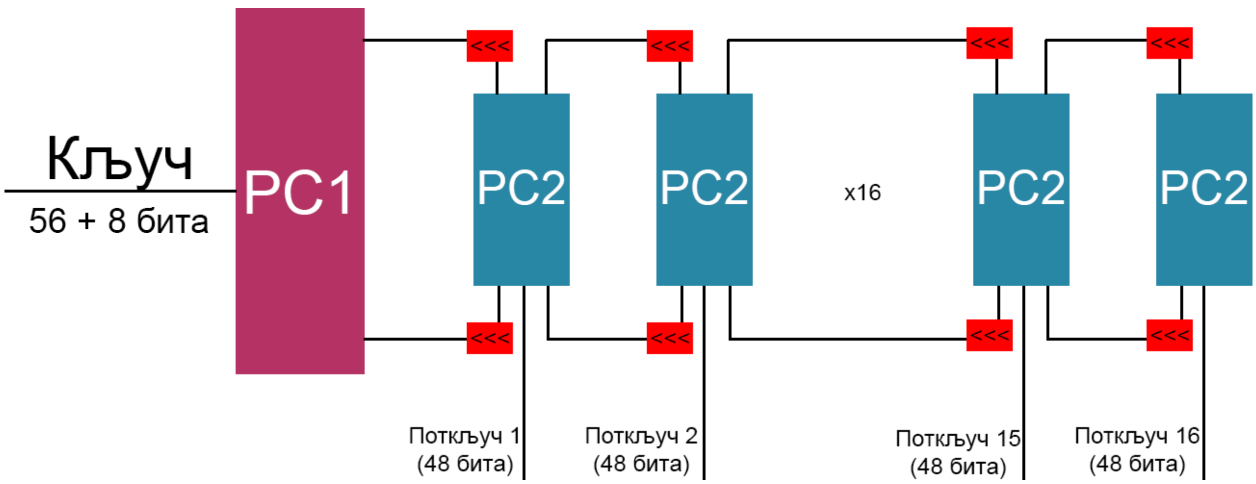
Слика 4. Приказ Фајснерове функције (F-функције)

F-функција у Фајснеровој шеми назива се такође и Фајснерова функција. Ова функција прима једну половину блока, која се састоји од 32 бита, и процесује га кроз 4 фазе.

1. Проширење (E-функција) – ова функција проширује део блока који улази у F-функцију са 32 бита на 48 бита дуплицирањем пола битова. Функција узима по 4 бита из блока (aaaa) и са обе стране додаје по копију бита који се налази најближе тој страни из суседних скупова од по 4 бита. Операција ове функције може се приказати овако: xxxa + bbbb + cxxx -> abbbbc. Од 4 улазна бита, ова функција враћа 6 битова. Укупно од 32 бита, 8 пута по 4 бита, се добије 48 бита, 8 пута по 6 битова.
2. Мешање кључа – На резултату E-функције се са једним од 16 поткључева (чије добијање ће бити објашњено даље у тексту), сваким за по једну рунду, врши XOR операција. Овај процес је приказан на слици симболом ⊕.

3. Замена – Након мешања кључа, блок се дели на 8 група од по 6 битова, свака од којих пролази крој једну S-кутију (substitution box, енгл. кутија замене). На свакој групи се изврши по нелинеарна трансформација (дата лукап табелом). Овај део процеса даје главни допринос сигурности овог алгоритма. Без овог дела процеса алгоритам би био линеаран и тривијалан за дешифровање.
4. Пермутација – Последњи корак F-функције узима сваки од 32 бита из S-кутија и пермутира их по фиксној пермутацији која се налази у P-кутији. Овај корак служи да излазни битови сваке појединачне S-кутије следећи пут уђу сви у различите S-кутије.

Наизменична проширења, пермутације и замене доводе до већ поменуте „конфузије” и „дифузије”.



Слика 5. Приказ формирања поткључева

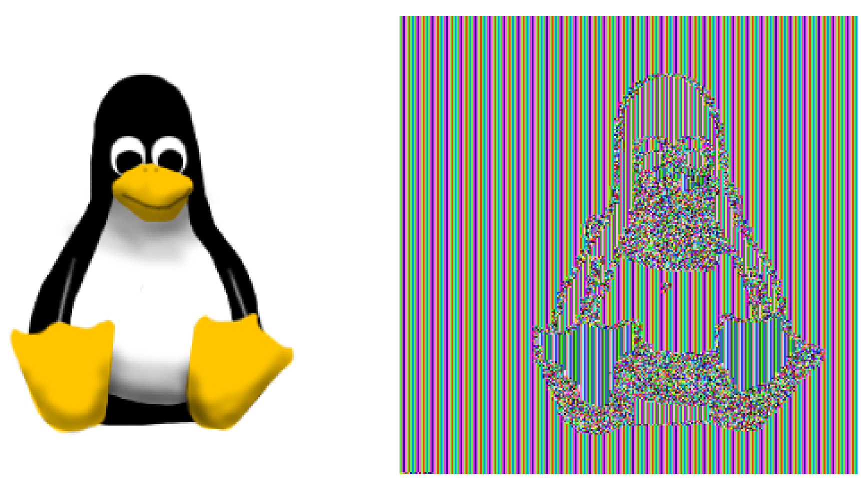
Поткључеви се формирају у делу алгоритма који се назива „кључни распоред”. PC1 (Permuted Choice 1, енгл. пермутирани избор 1) бира 56 битова од почетних 64 бита кључа, а осталих 8 одбацује или користи за проверу грешака. Ових 56 битова дели на два дела од по 28 битова с којима рукује раздвојено. У свакој рунди овог процеса обе половине се ротирају за 1 или 2 бита. Број битова за који ће ротирати у некој рунди је унапред одређен. Сваке рунде, након ротирања обе половине, PC2 бира 24 бита из једне половине и 24 бита из друге, укупно 48 бита. Пошто половине кључа ротирају у свакој рунди, сваки поткључ ће имати различите битове кључа. Овај процес се при дешифровању разликује само по обрнутом редоследу поткључева. Сваки поткључ одговара једној рунди у Фајснеровој структури, зато и „кључни распоред” има 16 рунди.

Пошто је DES био стандард шифровања преко две деценије, очекивано је да је на њега падало много пажње криптоаналитичара. Заиста, DES је најиспитиванија блоковска шифра у историју, али упркос томе најпрактичнији напад је и даље брут-форс. 1998. група програмера је уз 250000 долара дешифровала DES брут-форсом након мало више од 48 сати, чиме су демонстрирали потребу за квалитетнијим стандардом пифровања. Иако постоје теоретски бржи напади, они нису практични за употребу, а пошто је кључ који DES користи релативно мале величине брут-форс се сматра довољно практичним за његово дешифровање.

Постоје три напада која су теоретски бржа од брут-форса за дешифровање DES-а. Први је диференцијална криптоанализа. При дизајнирању DES-а, одлучено је да се алгоритам делимично измени од провобитног предлога да би био отпорнији на овај напад, иако је то смањило отпорност на брут-форс нападе, који за време настанка DES-а нису били велика претња због релативне слабости рачунара наспрам две деценије касније. За ову методу је потребно знати 2^{47} шифрата шифрираних истим кључем. Други напад је линеарна криптоанализа. За овај напад теоретски је потребно знати 2^{43} шифрата. Након открића овог могућег напада, неке теорије су показале да би можда било довољно знати само 2^{39} шифрата. Трећи напад је направљен специфично за дешифровање DES-а, за разлику од прва два која се могу користити и на осталим шифрама. Тај напад се зове побољшан Дејвисов напад, који је први предложио Доналд Дејвис још у 1980-им годинама. Најјача могућа форма напада има комплексност 2^{50}, потребно је знати 2^{50} шифрата и отворених текстова, и има 51% шансу успеха.

Као што је споменуто, DES се не користи сам за шифровање, зато су стандардизовани и његови начини коришћења кроз модове операције шифре. Најпознатији су ECB и CBC. Најједноставнији од свих је ECB (Electronic Cookbook, енгл. Електронска кодна књига). Шифровање ECB модом врши се тако што се отворени текст прво подели на блокове адекватне величине. Сваки блок се шифрује посебно на исти начин. Кад су сви блокови шифровани поново се споје у један шифрат, који је коначни резултат. ECB се не препоручује за употребу јер му фали дифузија, што доводи до тога да не може да прикрије понављања у отвореном тексту, јер сваки блок се шифрује на исти начин. Ова слабост у ECB шифровању се поготово примећује при шифровању bitmap слика које садрже велике количине исте боје. Пошто се сваки блок шифрује на исти начин, већи простори исте боје ће бити шифровани на потпуно исти начин, јер ће цели упасти у један блок. Програм који шифрује текст помоћу ECB стандарда и програм који је кориштен за шифровање ове слике као пример су достављени уз рад.

Слика 6. Отворена слика и иста слика шифрована ECB стандардом.



5. Закључак

У овом раду представили смо неке криптографске алгоритме и дали многе примере за њих да боље илуструјемо примену и значај ове науке како у историји тако и у свакодневном животу. Позабавили смо се по примером за сваку познатију врсту криптографских алгоритама и објаснили на ком принципу функционишу, па завршили на њиховим предностима и манама. Упркос теоријским објашњењима, главна идеја овог рада је била дати пример програма за сваки до наведених алгоритама. За понеке алгоритме смо такође и представили њихове слабости, да ли теоријским или програмским путем. У случају DES, због његовог утицаја и важности за савремену криптографију, дали смо два примера програма: један који шифрује текст и један који шифрује слику.

Криптографија од свог настанка никада није изгубила значаја, и тешко је замислити да икада хоће с обзиром на чињеницу да готово сваки савремени рачунарски систем користи криптографију у неком степену. Како се налазе сигурнији алгоритми, тако криптоанализа постаје све тежа наука, у неким случајевима чак немогућа, али као што предатор и плен један другог терају на адаптацију и тиме на еволуцију, у истом односу су криптоанализа и криптографија. Можда једног дана криптоанализа стигне да се дохвати криптографије и да дешифрује и најкомпликованије алгоритме, али по теоријским предвиђањима, тај дан највероватније неће стићи пре краја нашег универзума.

Литература

[1] М. Живковић, *Алгоритми,* Математични факултет, Београд

[2] R. Pass, A. Shelat, *a Course in Cryptography* (2010)

[3] https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html, приступљено 23.05.2024.

[4] https://datatracker.ietf.org/doc/html/rfc2828, приступљено 23.05.2024.

[5] https://en.wikipedia.org/wiki/RSA_(cryptosystem), приступљено 26.05.2024.

[6] https://en.wikipedia.org/wiki/Data_Encryption_Standard, приступљено 26.05.2024.

[7] https://github.com/tkeliris/ecb-penguin, приступљено 26.05.2024.