Criptografía y Computación

Curso 2018-19

Práctica 2. Cifrados clásicos

Autores[1]: Marcel Kemp, marcelkemp@correo.ugr.es

Víctor García, victorgarcia@correo.ugr.es

Síntesis.

Esta presente memoria recoge los pasos seguidos en un entorno de ejecución Linux para realizar la segunda práctica de la asignatura de Criptografía y Computación, donde realizamos un criptoanálisis de una serie de ficheros de texto cifrados con diversos métodos clásicos (sustitución monoalfabética, polialfabética o transposición) con el objetivo de descifrarlos.

Proceso de análisis.

Contamos con una serie de funciones para analizar diversas características de los textos cifrados, tales como la frecuencia de las letras, los n-gramas o cadenas de tamaño n más frecuentes, el numero de veces que aparece una cadena en el texto... De esta forma probamos a romper los cifrados clásicos, obsoletos hoy día aunque de importancia vital en la historia de la criptografía.

Hay que tener en cuenta que los textos cifrados están codificados como letras mayúsculas del abecedario español de 27 caracteres [A-Z] incluyendo la Ñ.

Puesto que a priori no sabemos qué tipo de cifrado se ha aplicado sobre cada texto, los resultados mostrados a continuación son resultado de probar los 4 diferentes tipos de análisis de descifrado sobre los 4 textos hasta dar con el tipo de cifrado de cada texto. A continuación se detallan los procesos de descifrado que se han llevado a cabo con cada texto y el resultado obtenido. Los textos descifrados al completo se encuentran en el archivo *textos.txt*

1 Como autor declaro que los contenidos del presente documento son originales y elaborados por mi. De no cumplir con este compromiso, soy consciente de que, de acuerdo con la "Normativa de evaluación y de calificaciones de los estudiantes de la Universidad de Granada" esto "conllevará la calificación numérica de cero … independientemente del resto de calificaciones que el estudiante hubiera obtenido ..."

Texto 1: Cifrado de Vigenère

Este primer texto sigue un esquema de cifrado polialfabético basado en el criptosistema de Vigenère. Este tipo de cifrado se basa en elegir una palabra clave *k* de longitud *m* y repetir la clave tantas veces como sea necesario hasta obtener un texto del mismo tamaño que el mensaje, llamemosle *textoclave*. Cada letra del mensaje o texto plano se cifra siguiendo el esquema de cifrado Cesar con la letra que le corresponde por posición en *textoclave*, es decir, sumando las letras módulo 27 (tamaño de nuestro abecedario) para *trasladar* la letra del mensaje en claro tantas posiciones como la letra de *textoclave*. De esta forma, cada letra se cifra de varias maneras, en media de *m* maneras diferentes, dependiendo de la letra de la clave *k* que le toque en *textoclave*. La función de descifrado es igual salvo que resta las letras de *k* en vez de sumarlas.

Este método de cifrado polialfabético resulta prácticamente indescifrable si la clave resulta ser de un tamaño similar al del mensaje a cifrar, algo realizable en el siglo XVI, época en la que se desarrolló este método, lo cual resulta algo sorprendente (fue apodado como el código indescifrable, *le chiffre indéchiffrable*). Se puede coger como clave un libro (su texto), y cifrar se limita a poner el texto debajo del mensaje en claro y sumar las letras una a una en correspondencia.

Para romper este método empleamos el método de Kasiski, que consiste en encontrar n-gramas o cadenas que se repiten en el texto. *Si el número de repeticiones es mayor que la longitud de la clave*, entonces esas cadenas han sido cifradas varias veces de la misma manera (ya se comentó previamente que cada letra puede cifrarse de *m* maneras diferentes, siendo *m* la longitud de la clave). Encontrando estas cadenas repetidas, suponiendo que provienen de la misma cadena en claro, significa que se han cifrado con la misma parte de la clave, por lo que la distancia que las separa en el texto debe ser múltiplo de la clave. Esto se ha llevado a cabo viendo los trigramas más frecuentes, encontrando el trigrama IJG. Al observar sus apariciones en el texto y el número de letras entre ellas, vemos que todas las distancias son múltiplos de 12. Esto nos lleva a concluir que la clave *k* es de longitud $m = 12$. Una vez conocemos el tamaño de la clave, dividimos el texto cifrado en bloques de tamaño $m = 12$, por lo que el descifrado se limita a romper el esquema Cesar de los bloques. Basta con un análisis de frecuencias, una explicación más detallada se encuentra en el texto 3.

La clave de cifrado k = ROCAMBOLESCO

Texto descifrado:

DEJAMOSENELANTERIORCAPITULOALVALEROSOVIZCAINOYALFAMOSODONQUIJOTECONLASESPADASALTASYDESNUDASENGUISADED
ESCARGARDOSFURIBUNDOSFENDIENTESTALESQUESIENLLENOSEACERTABANPORLOMENOSSEDIVIDIRIANYHENDERIANDEARRIBAABA
JOYABRIRIANCOMOUNAGRANADAYQUEENAQUELPUNTOTANDUDOSOPAROYQUEDODESTRONCADATANSABROSAHISTORIASINQUENO
SDIESENOTICIASUAUTORDONDESEPODRIAHALLARLOQUEDEELLAFALTABACAUSOMEESTOMUCHAPESADUMBREPORQUEELGUSTODEH
ABERLEIDOTANPOCOSEVOLVIAENDISGUSTOSDEPENSARELMALCAMINOQUESEOFRECIAPARAHALLARLOMUCHOQUEAMIPARECERFALT
ABADETANSABROSOCUENTOPARECIOMECOSAIMPOSIBLEYFUERADETODABUENACOSTUMBREQUEATANBUENCABALLEROLEHUBIESEF
ALTADOALGUNSABIOQUETOMARAACARGOENESCRIBIRSUSNUNCAVISTASHAZAÑASCOSAQUENOFALTOANINGUNODELOSCABALLEROSA
NDANTESDELOSQUEDICENLASGENTESQUEVANASUSAVENTURASPORQUECADAUNODEELLOSTENIAUNOODOSSABIOSCOMODEMOLDE
QUENOSOLAMENTEESCRIBIANSUSHECHOSSINOQUEPINTABANSUSMASMINIMOSPENSAMIENTOSYNIÑERIASPORMASESCONDIDASQUE
FUESENYNOHABIADESERTANDESDICHADOTANBUENCABALLEROQUELEFALTASEAELLOQUESOBROAPLATIRYAOTROSSEMEJANTESYASI
NOPODIAINCLINARMEACREERQUETANGALLARDAHISTORIAHUBIESEQUEDADOMANCAYESTROPEADAYECHADALACULPAALAMALIGNID
ADDELTIEMPODEVORADORYCONSUMIDORDETODASLASCOSASELCUALOLATENIAOCULTAOCONSUMIDAPOROTRAPARTEMEPARECIAQ
UEPUESENTRESUSLIBROSSEHABIANHALLADOTANMODERNOSCOMODESENGAÑODECELOSYNINFASYPASTORESDEHENARESQUETAMBI
ENSUHISTORIADEBIADESERMODERNAYQUEYAQUENOESTUVIESEESCRITAESTARIAENLAMEMORIADELAGENTEDESUALDEAYDELASAEL
LASCIRCUNVECINASESTAIMAGINACIONMETRAIACONFUSOYDESEOSODESABERREALYVERDADERAMENTETODALAVIDA...

Texto 2: Cifrado por sustitución

El siguiente texto ha sido sometido a un análisis de cifrado por sustitución monoalfabética, donde cada letra permuta con otra distinta del abecedario, de manera que cada letra es cifrada con otra diferente, a elección arbitraria (sin repetir letras). Por ende, la clave de cifrado es una permutación de las letras de nuestro abecedario, de la forma A:B, B:O, C:L ... donde la letra A se codifica como una B, la letra B como una O etc. Para descifrar se invierte la permutación, de forma que la permutación para descifrar el ejemplo anterior es de la forma B:A, O:B, L:C...

Este tipo de cifrado monoalfabético es vulnerable al análisis de frecuencias.

Un n-grama es una "palabra" de longitud n (secuencia de n letras), siendo los 1-gramas las letras, ejemplos de 2-gramas son DE, EN, ES, ejemplo de 3-grama QUE... Podemos obtener los n-gramas más frecuentes del texto cifrado y conociendo los n-gramas más frecuentes del lenguaje español podemos así obtener la sustitución o permutación de esas palabras (en concreto, de cada una de sus letras).

Comenzamos con los 1-gramas más frecuentes, que son las letras más frecuentes. Los obtenidos en orden descendente son M,W,T,U,Ñ que se corresponderían siguiendo la distribución estandar de un texto en español con las letras E,A,O,S,R/N(muy parecidas en frecuencia) respectivamente. Con el análisis de los siguientes n-gramas se refina este resultado a M=A, W=E, T=O, U=N, Ñ = S.

Repetimos el proceso con los 2-gramas. En este caso, los obtenidos en orden descendente fueron YW, WS, SM, TÑ, WÑ que, con el conocimiento de las letras del análisis anterior, y junto con los 2-gramas más frecuentes del español, podemos asociarlos con las palabras DE, EL, LA, OS, ES obteniendo Y=D, S=L.

Los 3-gramas más frecuentes obtenidos son PZW, ETU, YWS, WSM, MYW que igual que en el apartado anterior asociamos con las palabras QUE, CON, DEL, ELA, ADE obteniendo P=Q, Z=U, E=C.

Los 4-gramas más frecuentes obtenidos son YWSM, MUYT, WÑYW, LMJI, EITU que se corresponden con las palabras DELA, ANDO, ESDE, HABI, CION.

Es importante mencionar un problema del análisis de frecuencias como lo es que, al analizar n-gramas de mayor tamaño, es posible que los n-gramas más frecuentes sean aquellos compuestos por varios n-gramas de menor tamaño que son muy frecuentes. Esto se ejemplifica con algunos de los 4-gramas más frecuentes obtenidos como WÑYW = ESDE y YWSM = DELA, que en vez de ser algunas de las palabras de longitud 4 más frecuentes del español son composiciones de 2 palabras muy frecuentes del español como ES, DE o LA. Esto ocurre porque en el texto cifrado no hay espacios. Podríamos optimizar el análisis de n-gramas para descartar estos casos.

Continuando con este proceso, analizando otros n-gramas y no solo los 5 más frecuentes, cuadrando un poco las frecuencias de nuestro texto cifrado con las que debería seguir un texto en español y, probando a permutar entre las últimas letras sin descifrar (recordemos que, al no poder repetir letras, cuando faltan por ejemplo 3 letras, solo hay 3!=6 posibles combinaciones), obtenemos la permutación de todas las letras, resultando en el siguiente texto descifrado.

Texto descifrado:

ELMATRIMONIOESTUVOAPUNTODEACABARSEALOSDOSMESESPORQUEAURELIANOSEGUNDOTRATANDODEDESAGRAVIARAPETRACOT
ESLEHIZOTOMARUNRETRATOVESTIDADEREINADEMADAGASCARCUANDOFERNANDALOSUPOVOLVIOAHACERSUSBAULESDERECIENCA
SADAYSEMARCHODEMACONDOSINDESPEDIRSEAURELIANOSEGUNDOLAALCANZOENELCAMINODELACIENAGAALCABODEMUCHASSUP
LICASYPROPOSITOSDEENMIENDALOGROLLEVARLADEREGRESOALACASAYABANDONOALACONCUBINAPETRACOTESCONSCIENTEDESU
FUERZANODIOMUESTRASDEPREOCUPACIONELLALOHABIAHECHOHOMBRESIENDOTODAVIAUNNIWOLOSACODELCUARTODEMELQUIA
DESCONLACABEZALLENADEIDEASFANTASTICASYSINNINGUNCONTACTOCONLAREALIDADYLEDIOUNLUGARENELMUNDOLANATURALE
ZALOHABIAHECHORESERVADOYESQUIVOCONTENDENCIASALAMEDITACIONSOLITARIAYELLALEHABIAMOLDEADOELCARACTEROPUES
TOVITALEXPANSIVODESABROCHADOYLEHABIAINFUNDIDOELJUBILODEVIVIRYELPLACERDELAPARRANDAYELDESPILFARROHASTACON
VERTIRLOPORDENTROYPORFUERAENELHOMBRECONQUEHABIASOWADOPARAELLADESDELAADOLESCENCIASEHABIACASADOPUESC
OMOTARDEOTEMPRANOSECASANLOSHIJOSELNOSEATREVIOAANTICIPARLELANOTICIAASUMIOUNAACTITUDTANINFANTILFRENTEALA
SITUACIONQUEFINGIAFALSOSRENCORESYRESENTIMIENTOSIMAGINARIOSBUSCANDOELMODODEQUEFUERAPETRACOTESQUIENPROV
OCARALARUPTURAUNDIAENQUEAURELIANOSEGUNDOLEHIZOUNREPROCHEINJUSTOELLAELUDIOLATRAMPAYPUSOLASCOSASENSUPU
ESTOLOQUEPASADIJOESQUETEQUIERESCASARCONLAREINAAURELIANOSEGUNDOAVERGONZADOFINGIOUNCOLAPSODECOLERASEDE
CLAROINCOMPRENDIDOYULTRAJADOYNOVOLVIOAVISITARLAPETRACOTESSINPERDERUNSOLOINSTANTESUMAGNIFICODOMINIODEFI
ERAENREPOSOOYOLAMUSICAYLOSCOHETESDELABODAELALOCADOBULLICIODELAPARRANDAPUBLICACOMOSITODOESONOFUERAMA
SQUEUNANUEVATRAVESURADEAURELIANOSEGUNDOAQUIENESSECOMPADECIERONDESUSUERTELOSTRANQUILIZOCONUNASONRISAN
OSEPREOCUPENLESDIJOAMILASREINASMEHACENLOSMANDADOSAUNAVECINAQUELELLEVOVELASCOMPUESTASPARAQUEALUMBRAR
ACONELLASELRETRATODELAMANTEPERDIDOLEDIJOCONUNASEGURIDADENIGMATICALAUNICAVELAQUELOHARAVENIRESTASIEMPRE
ENCENDIDATALCOMOELLALOHABIAPREVISTOAURELIANOSEGUNDOVOLVIOASUCASATANPRONTOCOMOPASOLALUNADEMIELLLEVOAS
USAMIGOTESDESIEMPREUNFOTOGRAFOAMBULANTEYELTRAJEYLACAPADEARMIWOSUCIADESANGREQUEFERNANDAHABIAUSADOEN
ELCARNAVALALCALORDELAPARRANDAQUESEPRENDIOESATARDEHIZOVESTIRDEREINAAPETRACOTESLACORONOSOBERANAABSOLUT
AYVITALICIADEMADAGASCARYREPARTIOCOPIASDELRETRATOENTRESUSAMIGOSELLANOSOLOSEPRESTOALJUEGOSINOQUESECOMPA
DECIOINTIMAMENTEDEELPENSANDOQUEDEBIAESTARMUYASUSTADOCUANDOCONCIBIOAQUELEXTRAVAGANTERECURSODERECONCI
LIACIONALASSIETEDELANOCHETODAVIAVESTIDADEREINALORECIBIOENLACAMATENIAAPENASDOSMESESDECASADOPEROELLASEDIO
CUENTAENSEGUIDADEQUELASCOSASNOANDABANBIENENELLECHONUPCIALYEXPERIMENTOELDELICIOSOPLACERDELAVENGANZACON
SUMADADOSDIASDESPUESSINEMBARGOCUANDOELNOSEATREVIOAVOLVERSINOQUEMANDOUNINTERMEDIARIOPARAQUEARREGLARA
LOSTERMINOSDELASEPARACIONELLACOMPRENDIOQUEIBAANECESITARMASPACIENCIADELAPREVISTAPORQUEELPARECIADISPUESTO
ASACRIFICARSEPORLASAPARIENCIASTAMPOCOENTONCESSEALTEROVOLVIOAFACILITARLASCOSASCONUNASUMISIONQUECONFIRMOL
ACREENCIAGENERALIZADADEQUEERAUNAPOBREMUJERYELUNICORECUERDOQUECONSERVODEAURELIANOSEGUNDOFUEUNPARDEB
OTINESDECHAROLQUESEGUNELMISMOHABIADICHOERANLOSQUEQUERIALLEVARPUESTOSENELATAUDLOSGUARDOENVUELTOSENTR
APOSENELFONDODEUNBAULYSEPREPAROPARAAPACENTARUNAESPERASINDESESPERACIONTARDEOTEMPRANOTIENEQUEVENIRSEDI
JOAUNQUESOLOSEAAPONERSEESTOSBOTINESNOTUVOQUEESPERARTANTOCOMOSUPONIAENREALIDADAURELIANOSEGUNDOCOMPR
ENDIODESDELANOCHEDEBODASQUEVOLVERIAACASADEPETRACOTESMUCHOANTESDEQUETUVIERANECESIDADDEPONERSELOSBOTI
NESDECHAROLFERNANDAERAUNAMUJERPERDIDAPARAELMUNDOHABIANACIDOYCRECIDOAMILKILOMETROSDELMARENUNACIUDAD
LUGUBREPORCUYASCALLEJUELASDEPIEDRATRAQUETEABANTODAVIAENNOCHESDEESPANTOSLASCARROZASDELOSVIRREYESTREINT
AYDOSCAMPANARIOS...

Texto 3: Cifrado de Cesar

Para descifrar este texto hemos aplicado un análisis de descifrado por sustitución monoalfabética, siguiendo el esquema del criptosistema de Cesar. Se trata de un cifrado por traslación donde cada una de las m letras del abecedario se codifica como un número entre 0 y m-1 (nuestro alfabeto es de tamaño m=27, Z_{27}). El cifrado consiste en establecer una clave k entre 0 y m-1 y sumar este valor a cara carácter del mensaje. Esto resulta en la función de cifrado $E_k[m] = m + k$. Resulta absurdo probar a cifrar con una clave $k' >= m$ pues causaría el mismo texto cifrado que con la clave $k = k' \% m$.

La función de descifrado es $D_k[c] = c - k$. Puesto que los posibles valores que puede tomar la clave son ínfimos (problema del cifrado monoalfabético), podemos aplicar un ataque por fuerza bruta, probando los 27 posibles valores de la clave y aplicando la función de descifrado, observando los supuestos textos descifrados hasta dar con aquel que produce un texto legíble. En este caso, nuestro texto descifrado ha sido producido con un valor de clave k = 10.

Cabe destacar que este tipo de cifrado es vulnerable al análisis de frecuencias. La letra con mayor frecuencia en un texto en español es la "E", codificada como 4. Si obtenemos la lista de apariciones de cada letra en el texto cifrado observamos que la que más aparece es la “Ñ” codificada como 14. Luego, por análisis de frecuencias, parece ser que la “E” se ha codificado como la “Ñ”, lo que implica un valor de clave de:

k = “Ñ”-“E” = 14-4 = 10

Texto descifrado:

CADADOCEDEFEBREROSECONMEMORAELDIAINTERNACIONALCONTRAELUSODENIÑOSSOLDADOACTUALMENTESECALCULAQUEHAYU
NOSTRESCIENTOSMILNIÑOSYNIÑASSOLDADOENLOSCONFLICTOSARMADOSENTODOELMUNDONIÑOSYNIÑASQUESEVENABOCADOSAV
IVIRLAGUERRADEVERDADCONVIRTIENDOSEENCOMBATIENTESINVOLUNTARIOSMUCHOSDEESTOSNIÑOSESTANDIRECTAMENTEENLAL
INEADECOMBATEYOTROSSONOBLIGADOSAEJERCERCOMOCOCINEROSMENSAJEROSESCLAVASSEXUALESPARAREALIZARATAQUESSUIC
IDASDURANTEELTIEMPOENELQUEESTOSNIÑOSESTANVINCULADOSALASFUERZASYGRUPOSARMADOSSONTESTIGOSYVICTIMASDETE
RRIBLESACTOSDEVIOLENCIAEINCLUSOSONOBLIGADOSAEJERCERLALOSTRAUMASEMOCIONALESQUEESTOLESPUEDEPROVOCARSOND
IFICILESDESUPERARALGUNOSSONSECUESTRADOSAOTROSLAPOBREZALOSMALOSTRATOSLAPRESIONDELASOCIEDADOELDESEODEVE
NGARSEDELAVIOLENCIACONTRAELLOSOSUSFAMILIASLESLLEVANAUNIRSEAGRUPOSARMADOSYEMPUÑARUNARMASONVICTIMASINOC
ENTESDELASATROCIDADESDELAGUERRAPARAELLOSELREGRESOASUVIDAYLARECUPERACIONDESUINFANCIAESTANDIFICILQUEPUEDE
PARECERCASIIMPOSIBLEENLOSULTIMOSAÑOSLASGUERRASCADAVEZSONMASBRUTALESYMASLARGASALGUNASESTANENLOSMEDIOS
DECOMUNICACIONDEFORMAMASOMENOSESTABLECOMOSIRIAUNCONFLICTOQUESEPROLONGADESDEHACECASISIETEAÑOSPEROOTR
ASSONINVISIBLESPARALAMAYORIADENOSOTROSCOMOYEMENSUDANDELSURREPUBLICACENTROAFRICANANIGERIAYMUCHASOTRAS
CONTANSOLOAÑOSJOHNNOMBREFICTICIOPARAPROTEGERSUIDENTIDADHASUFRIDOSITUACIONESQUENADIEDEBERIAVIVIRNUNCACO
MOPERDERASUSPADRESPORLAGUERRAOVERCOMOMATABANASUSAMIGOSDELANTESUYOSINMEDIOSPARASUBSISTIRUNGRUPOARMA
DODESUDANDELSURLEOFRECIOCOMIDAYUNLUGARDONDEDORMIRNOQUIEROVOLVERASERUNNIÑOSOLDADOPORQUEALLINOHAYCOM
IDANIESCUELANOHAYNADARECIBIUNDISPAROENLAPIERNAYNADIEMEAYUDABANOSCUENTAJOHNLASPEORESCONSECUENCIASDELUS
ODENIÑOSSOLDADOSECUELASFISICASPUEDENSERCAUSADASPORLAPROPIABATALLAOSERFRUTODELASTORTURASYABUSOSPORPART
EDESUSJEFESMUCHOSNIÑOSSONMUTILADOSSUFRENDESNUTRICIONOINCLUSOENFERMEDADESDETRANSMISIONSEXUALENELCASOD
ELASNIÑASMUCHASSEQUEDANEMBARAZADASPORABUSOSSEXUALESTRAUMASEMOCIONALESELHECHODEHABERPRESENCIADOACTO
SDEVIOLENCIATERRIBLESOTENERQUECOMETERLOSDIRECTAMENTELESPUEDEATORMENTARSINOSELESDAAPOYOPSICOLOGICOMUC
HASVECESELPRIMERACTOQUELESOBLIGANACOMETERESMATARASUSPROPIOSPADRESPARAROMPERELVINCULOFAMILIARLADIFICULT
ADDESALIRDELAESPIRALDEVIOLENCIAYVOLVERACASAPORQUEPASANENELGRUPOOFUERZAARMADALOSAÑOSENLOSQUEDESARROLL
ANSUPERSONALIDADYAPRENDENACONVIVIRENUNENTORNOJERARQUICOYDEVIOLENCIAPORQUENOSABENDONDEESTASUFAMILIAYC
OMUNIDADYCUANDOPORFINSEENCUENTRANAVECESLAFAMILIALOSRECHAZAPORSUPASADOYAQUETIENENMIEDOAQUELOSATAQUEN
ONOACEPTANALASNIÑASCUANDOVUELVENCONHIJOSQUEHANTENIDODURANTESUAUSENCIAPORQUENOHANPODIDOIRALAESCUELAY
ESTOHACEQUESUSOPORTUNIDADESDEUNFUTUROMEJORSEREDUZCANENORMEMENTE

Texto 4: Cifrado de transposición

El último texto a descifrar presenta un cifrado por transposición. En este tipo de cifrado se elige como clave un valor *n* y se recorre el texto de *n* en *n*, comenzando en la primera posición o letra. Al llegar al final del texto, comienza en la segunda posición y repite el proceso, así hasta su última repetición que comienza en la posición *n* -1.

Esquema:

1º Frecuencias para poder descartar vigenere (frecuencias muy igualadas), y el cifrado de Cesar (frecuencias=abecedario movidas x posiciones).

2º Ngramas repetidos de 3, para comprobar si puede ser sustitución. Improbable porque salen ngramas que no tienen sentido sustituirlos, como (['AAE', 'EEE', 'EAE', 'AES', 'OAA'], [19, 25, 19, 17, 15]).

3º Como podemos que puede ser transposición, probamos un bucle que nos saque las ocurrencias de QUE de n en n, donde n toma el valor del bucle (1, len(txt_cifrado)). Y dentro del bucle, ponemos la condición de si encuentra >=5 ocurrencias, imprima el valor de n.

4º Sacamos que n=31, ahora probamos len(txt_cifr)/n=217,... ; luego para ver si tiene sentido, dividimos cadena entre 217 (o 218), y vemos que tiene sentido gran parte del texto pero está desordenado.

5º Descifrar guardando bloques de n=31 con un bucle, y la función siguiente.

Texto descifrado:

TODASLASCOSASSERCRIADASAMANERADECONTIENDAOBATALLADICEAQUELGRANSABIOHERACLITOENESTEMODOOMNIASECUNDUM
LITEMFIUNTSENTENCIAAMIVERDIGNADEPERPETUAYRECORDABLEMEMORIAYCOMOSEACIERTOQUETODAPALABRADELHOMBREESCI
ENTEESTAPREÑADADEESTASEPUEDEDECIRQUEDEMUYHINCHADAYLLENAQUIERAREVENTARECHANDODESITANCRECIDOSRAMOSYHO
JASQUEDELMENORPIMPOLLOSESACARIAHARTOFRUTOENTREPERSONASDISCRETASPEROCOMOMIPOBRESABERNOBASTEAMASDERO
ERSUSSECASCORTEZASDELOSDICHOSDEAQUELLOSQUEPORCLARORDESUSINGENIOSMERECIERONSERAPROBADOSCONLOPOCOQUED
EALLIALCANZARESATISFAREALPROPOSITODEESTEPERBREVEPROLOGOHALLEESTASENTENCIACORROBORADAPORAQUELGRANORAD
ORYPOETALAUREADOFRANCISCOPETRARCADICIENDOSINELITEATQUEOFENSIONENIHILGENUITNATURAPARENSSINLIDYOFENSIONNIN
GUNACOSAENGENDROLANATURAMADREDETODODICEMASADELANTESICESTENIMETSICPROPEMODUMUNIVERSATESTANTURRAPIDO
STELLOBVIANTFIRMAMENTOCONTRARIAINUICEMELEMENTACONFLIGUNTTERRTREMUNTMARIAFLUCTUANTAERCUATITURCREPANT
FLAMMBELLUMIMMORTALEVENTIGERUNTTEMPORATEMPORIBUSCONCERTANTSECUMSINGULANOBISCUMOMNIAQUEQUIEREDECIRE
NVERDADASIESYASITODASLASCOSASDEESTODANTESTIMONIOLASESTRELLASSEENCUENTRANENELARREBATADOFIRMAMENTODELCI
ELOLOSADVERSOSELEMENTOSUNOSCONOTROSROMPENPELEATREMENLASTIERRASONDEANLOSMARESELAIRESESACUDESUENANLAS
LLAMASLOSVIENTOSENTRESITRAENPERPETUAGUERRALOSTIEMPOSCONTIEMPOSCONTIENDENYLITIGANENTRESIUNOAUNOYTODOS
CONTRANOSOTROSELVERANOVEMOSQUENOSAQUEJACONCALORDEMASIADOELINVIERNOCONFRIOYASPEREZAASIQUEESTONOSPARE
CEREVOLUCIONTEMPORALESTOCONQUENOSSOSTENEMOSESTOCONQUENOSCRIAMOSYVIVIMOSSICOMIENZAAENSOBERBECERSEMA
SDELOACOSTUMBRADONOESSINOGUERRAYCUANTOSEHADETEMERMANIFIESTASEPORLOSGRANDESTERREMOTOSYTORBELLINOSPO
RLOSNAUFRAGIOSYINCENDIOSASICELESTIALESCOMOTERRENALESPORLAFUERZADELOSAGUADUCHOSPORAQUELBRAMARDETRUENO
SPORAQUELTEMEROSOIMPETUDERAYOSAQUELLOSCURSOSYRECURSOSDELASNUBESDECUYOSABIERTOSMOVIMIENTOSPARASABERL
ASECRETACAUSADEQUEPROCEDENNOESMENORLADISENSIONDELOSFILOSOFOSENLASESCUELASQUEDELASONDASENLAMARPUESENT
RELOSANIMALESNINGUNGENEROCARECEDEGUERRAPECESFIERASAVESSERPIENTESDELOCUALTODOUNAESPECIEAOTRAPERSIGUEELL
EONALLOBOELLOBOLACABRAELPERROLALIEBREYSINOPARECIESECONSEJADETRASELFUEGOYOLLEGARIAMASALCABOESTACUENTAE
LELEFANTEANIMALTANPODEROSOYFUERTESEESPANTAYHUYEDELAVISTADEUNSUCIORATONYAUNDESOLOOIRLETOMAGRANTEMORE
NTRELASSERPIENTESELBASILISCOCRIOLANATURATANPONZOÑOSOYCONQUISTADORDETODASLASOTRASQUECONSUSILBOLASASOM
BRAYCONSUVENIDALASAHUYENTAYESPARCECONSUVISTALASMATA...

Texto 5 (Extra de telegram): Cifrado de Vigenère

Hemos seguido los mismos pasos que en el primer texto, y hemos obtenido que la clave era de longitud 5. Por lo que mediante el análisis de las frecuencias, hemos llegado a sacar que la clave usada para el cifrado era: k = CLAVE.

Texto descifrado:

QUERIDOSOYENTESBIENVENIDOSUNASEMANAMASAENTREVISTAALAVISTANUESTRAINVITADADEHOYESTAMARASAGASTAUNAMU
JERQUEASEGURAHABLARUSANDOUNICAMENTELAVOCALAESCORRECTOTAMARAAJABIENCOMENCEMOSCONUNASPREGUNTASSE
NCILLASPARACONOCERLAUNPOCOMEJORYCOMPROBARSIESCIERTOESOQUEDICELEPARECEALACARGACUALESSUNOMBRECOMPL
ETOANATAMARASAGASTACAÑADAESTADOCIVILCASADAPROFESIONAZAFATAENQUEPUEBLOOCIUDADVIVECASAFRANCANOLOCO
NOZCOAQUEPROVINCIAPERTENECEASALAMANCAMEDIRIAELNOMBREDELACALLESANTABARBARACUALESSUCOLORFAVORITONA
RANJASUPELICULAPREFERIDACASABLANCAYALGUNAMASMODERNACAZAFANTASMASESAESDELOSOCHENTAUNAMASRECIENTEP
ORFAVORAVATARVAYAPARECEQUEVIENEBIENPREPARADAPEROESTOYDISPUESTOALOGRARQUEDIGAALGUNAVOCALQUENOSEALA
AFRACASARASLEAPUESTOCINCUENTAEUROSAQUELOCONSIGOVAARRANCAVAMOSALLAPROGRAMADETELEVISIONFAVORITOPAS
APALABRAYALGUNOQUENOSOPORTELASCAMPANADASVAYAMAMARRACHADAQUEPAISESLEGUSTARIAVISITARMADAGASCARPANA
MACANADAMALTABAHAMASDONDEVERANEAMATALASCAÑASQUECASUALIDADYOTAMBIENCARAMBAYQUESUELEHACERALLIBAJ
ARALAPLAYACADAMAÑANAPARANADARABRAZAHASTAACABARCANSADATODOSLOSDIASARAJATABLA...

Anexo

Funciones.py

```
import time

import base64


#Dado un texto (string) solo con mayusculas lo transforma en una lista de números (A-0,...,Z-26). Por un espacio añade un -1

def cadenatolista(cadena):
    l = []
    for s in cadena:
        x = ord(s)
        if x == 32:
            l.append(-1)
        elif x < 79:
            l.append(x-65)
        elif x == 209:
            l.append(14)
        else:
            l.append(x-64)
    return l


#Inverso del anterior. Una lista de números (0--26) lo transforma en un string con mayúsculas

def listatocadena(l):
    s = ''
    for x in l:
        if x == -1:
            s = s + ' '
        elif x <= 13:
            s = s + chr(x+65)
        elif x == 14:
            s = s + 'Ñ'
        else:
            s = s + chr(x+64)
    return s
```

```
#Calcula las frecuencias de aparición de cada letra en una cadena de texto (en mayúsculas)
def frecuencias(texto):
    tabla = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
    lista = cadenatolista(texto)
    for x in lista:
        tabla[x] = tabla[x]+1
    return tabla


#Calcula el indice de coincidencia de un texto.
def indice_coincidencia(texto):
    tabla = frecuencias(texto)
    num_caracteres = 0
    aux = 0
    for x in tabla:
        num_caracteres +=x
        aux = aux + x*(x-1)
    ic = aux/(num_caracteres*(num_caracteres-1))
    return ic


#Divide un texto en n subtextos, recorriéndolo de n en n.
def divide_cadena(cadena,n):
     subcadenas = []
     for i in range(n):
         subcadenas.append('')
     j = 0
     for x in cadena:
         subcadenas[j] = subcadenas[j] + x
         j = (j+1)%n
     return subcadenas


#Dado un texto y una clave  (ambos un string con mayúsculas), lo cifra usando el cifrado de Vigenère
def cifra_vigenere(texto,clave):
     lista_texto = cadenatolista(texto)
```

```
    lista_clave = cadenatolista(clave)
    (n,m) = (len(lista_texto),len(lista_clave))
    for i in range(n):
        lista_texto[i] = (lista_texto[i] + lista_clave[i%m])%27
    texto_cifrado = listatocadena(lista_texto)
    return texto_cifrado

#Dado un texto y una clave  (ambos un string con mayúsculas), lo cifra usando el cifrado de Vigenère
def descifra_vigenere(texto,clave):
    lista_texto = cadenatolista(texto)
    lista_clave = cadenatolista(clave)
    (n,m) = (len(lista_texto),len(lista_clave))
    for i in range(n):
        lista_texto[i] = (lista_texto[i] - lista_clave[i%m])%27
    texto_cifrado = listatocadena(lista_texto)
    return texto_cifrado



"""Dado un texto y una permutación de las letras (diccionario) lo cifra aplicando la sustitución dada
La sustitución hay que darla como un diccionario. Por ejemplo:
sustitucion = {'A':'B', 'B':'C', 'C':'D', 'D':'E', 'E':'F', 'F':'G', 'G':'H', 'H':'I', 'I':'J', 'J':'K', 'K':'L', 'L':'M', 'M':'N', 'N':'Ñ', 'Ñ':'O', 'O':'P', 'P':'Q', 'Q':'R', 'R':'S', 'S':'T', 'T':'U', 'U':'V', 'V':'W', 'W':'X', 'X':'Y', 'Y':'Z', 'Z':'A'}
que sustituye cada carácter por el que le sigue en el alfabeto
"""
def cifra_sustitucion(texto,permutacion):
    texto_cifrado = ''
    for x in texto:
        y = permutacion.get(x)
        texto_cifrado = texto_cifrado + str(y)
    return texto_cifrado


"""permutación es una lista de dos strings (de igual longitud). Por ejemplo, ['AEG',fkl']. En este caso, se recorre el texto y cada vez que encuentre un carácter que coincida con un de los que hay en 'AEG' lo sustituye por el correspondiente carácter en 'fkl'"""
def descifra_sustitucion(texto,permutacion):
    texto_des = ''
    p0 = permutacion[0]
```

```
    p1 = permutacion[1]
    for x in texto:
        if x in p0:
            pos = p0.index(x)
            texto_des = texto_des + p1[pos]
        else:
            texto_des = texto_des + x
    return texto_des


#En un texto selecciona los m n-gramas que más se repiten (m=5 por defecto) y da la frecuencia de aparición de cada uno de ellos.
def ngramas_repetidos(texto,n,m=5):
    ngramas = []
    ngramasrep = []
    frecuencias = []
    for i in range(m):
        frecuencias.append(0)
        ngramasrep.append(texto[i:i+n])
    minimo = 0
    for i in range(len(texto)-n):
        aux = texto[i:i+n]
        if aux not in ngramas:
            f = 1
            ngramas.append(aux)
            for j in range(i+1,len(texto)-n):
                if aux == texto[j:j+n]:
                    f+=1
            if f > minimo:
                k = frecuencias.index(minimo)
                ngramasrep[k] = aux
                frecuencias[k] = f
                minimo = min(frecuencias)
    return (ngramasrep,frecuencias)
```

```
#Dada una cadena y un texto calcula las veces en que aparece la cadena, y la separación entre estas apariciones.
def apariciones(cadena,texto):
    m = len(cadena)
    n = len(texto)
    posicion = []
    for i in range(n-m):
        if cadena == texto[i:i+m]:
            posicion.append(i)
    return (posicion,len(posicion))



#Recorre el texto de n en n, comenzando por la primera posición. Al llegar al final, comienza por la segunda posición y así sucesivamente.
def cifra_transposicion(texto,n):
    m = len(texto)
    k = m%n
    texto_cif = ''
    for i in range(n):
        if i < k:
            aux = m//n+1
        else:
            aux = m//n
        for j in range(aux):
            texto_cif = texto_cif + texto[i+n*j]
    return texto_cif


#Suponiendo que se ha recorrido de n en n un texto de tamaño m, nos dice en que posición estaría el carácter siguiente al que está en la posición x. Primero calculamos donde estaría el último, pues ese no tiene siguiente.
def siguiente(m,n,x):
    aux = m%n
    aux2 = m//n
    if aux == 0:
        ultimo = -1
    else:
        ultimo = aux * (aux2 + 1) - 1
```

```
    if ultimo == -1 and x == m-1:
        return -1
    elif x < ultimo:
        return x+1+aux2
    elif x >= m - aux2:
        return x + aux2 + 1 - m
    elif x > ultimo:
        return x+aux2
    else:
        return -1



#Cuenta cuantas veces se repite una cadena en el fichero texto suponiendo que éste se ha obtenido recorriendo un fichero de n en n.
def ocurrencias(cadena,texto,n):
    l = len(texto)
    k = len(cadena)
    ocur = 0
    for i in range(l):
        contador = i
        cadenab = ''
        for j in range(k):
            if contador == -1:
                cadenab = cadenab + ' '
                contador = 0
            else:
                cadenab = cadenab + texto[contador]
                contador = siguiente(l,n,contador)
        if cadena == cadenab:
            ocur +=1
    return ocur

f1 = open('VictorGarciaMarcelKemp1.txt', mode='r', encoding='utf-8')    #Resuelto
f2 = open('VictorGarciaMarcelKemp2.txt', mode='r', encoding='utf-8')    #Resuelto
f3 = open('VictorGarciaMarcelKemp3.txt', mode='r', encoding='utf-8')    #Resuelto
f4 = open('VictorGarciaMarcelKemp4.txt', mode='r', encoding='utf-8')    #Resuelto
```

```
#f5 = open('Textocifrado.txt', mode='r', encoding='utf-8')              #Resuelto

#-------------------------------------
""" #Algoritmo Cesar
abc="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"
#cad="ABCZ"
cad3=f3.read()

max = 0
frecuencias = frecuencias(cad3)
print(frecuencias)
# Obtenemos la letra del texto cifrado
for frecletra in frecuencias:
    if frecletra > max:
        max = frecletra

# La letra con mayor frecuencia en un texto en castellano es la "E", codificada como 4
# La letra con mayor frecuencia en frecuencias(cad3) será la "E" cifrada
posclave = frecuencias.index(max) - 4

print("Posible clave por análisis de frecuencias: K = " + str(posclave) )

descif=""
for k in range(1,27):
    for i in cad3:
        if i in abc:
            # Función de descifrado de Cesar con clave k
            descif += abc[(abc.index(i)-k)%(len(abc))]
        else:
            descif += i
    print(descif)
    descif=""
    time.sleep(2)
print("Texto descifrado con clave: K = 10")
k=10
for i in cad3:
```

```
        if i in abc:
            # Función de descifrado de Cesar con clave k
            descif += abc[(abc.index(i)-k)%(len(abc))]
        else:
            descif += i

print(descif) """

#----------------------------------------
"""#Algoritmo de sustitucion
sustitucion = {'A':'M','B':'X','C':'T','D':'G','E':'C','F':'K','G':'P','H':'J','I':'I','J':'B','K':'Z','L':'H',
               'M':'A', 'N':'R', 'Ñ':'S','O':'V','P':'Q','Q':'Y','R':'F','S':'L', 'T':'O', 'U':'N',
               'V':'Ñ', 'W':'E','X':'W', 'Y':'D','Z':'U'}
#perm=['','QUE']
cad=f2.read()
print(ngramas_repetidos(cad,1))
print(ngramas_repetidos(cad,2))
print(ngramas_repetidos(cad,3))
print(ngramas_repetidos(cad,4))
print(frecuencias(cad))
print(cifra_sustitucion(cad,sustitucion))"""
#----------------------------------------
#Algoritmo de transposicion
cad='QUE'
cad4=f4.read()
#print(frecuencias(cad4))

#print(divide_cadena(cad4,2))
#print(ngramas_repetidos(cad4,3))
""" for k in range(1,len(cad4)):
    if(ocurrencias(cad,cad4,k)>5):
        print(ocurrencias(cad,cad4,k), k)  """

""" for k in range(1,len(cad4)):
    if((k%31)==0):
        print(cifra_transposicion(cad4,k), k) #217  """
```

```
#print(ocurrencias(cad,cad4,31))



""" texto=divide_cadena(cad4,217)

for x in range(0,31):
    txt=""
    for i in range(0,217):
        for j in range(0,31):
            txt+=texto[i][(j+x)%31]

    print(txt)
    print(x)
    time.sleep(3) """

"""
num=0
txt=""

for y in range(0,len(cad4)):
    txt+=cad4[num]
    num=siguiente(6731,31,num)



print(txt) """



#print(divide_cadena(cad4,217))
#txt2=cifra_transposicion(cad4,217)
#print(ngramas_repetidos(txt,3))
#print(len(cad4))

#----------------------------------------------------------------
#Cifrado de Vigenere
```

```
#cad1=f1.read()

#print(ngramas_repetidos(cad1,3))
#(['RJO', 'IJG', 'FFI', 'SUP', 'QGF'], [15, 14, 11, 13, 12])
""" print(apariciones('IJG',cad1)) """
#RJO: 204, 408, 444, 1339, 156, 413, 564, 1608, 48, 228, 252, 960, 264, 1452 mcd=1
#IJG: 1284, 588, 204, 12, 324, 588, 1080, 192, 264, 528, 456, 144, 1500 mcd=12

""" txt=divide_cadena(cad1,12)
print(frecuencias(txt[11]))          #R-O-C-A-M-B-O-L-E-S-C-O
print(ngramas_repetidos(txt[11],1))  """

#print(descifra_vigenere(cad1,'ROCAMBOLESCO'))

#----------------------------------------------------------------------
#Texto optativo: Vigenere
#cad5=f5.read()
""" for i in range(1,101):
    txt=divide_cadena(cad5,i)
    print(indice_coincidencia(txt[0]), i) """

""" print(ngramas_repetidos(cad5,1))
print(ngramas_repetidos(cad5,2))
print(ngramas_repetidos(cad5,3))
print(ngramas_repetidos(cad5,4)) """

#print(apariciones('ENL',cad5))
#620, 230, 155, 115, 180, 15, 140, 470, 100, 70, 155, 350 mcd=5

""" txt=divide_cadena(cad5,5)
print(frecuencias(txt[4]))          #C-L-A-V-E
print(ngramas_repetidos(txt[4],1))   """

#print(descifra_vigenere(cad5,'CLAVE'))
```

```
f1.close()
f2.close()
f3.close()
f4.close()
f5.close()
```
