Decodificación de Texto en Español Utilizando Frecuencias de Palabras Mediante Cómputo Paralelo

Barbara Emma Sanchez Rinza, Jesús García-Ramírez, Mario Rossainz
Facultad de Ciencias de la Computación,
Benemérita Universidad Autónoma de Puebla,
14 Sur y Avenida San Claudio
Puebla México,

brinza@hotmail.com,

Alberto Jaramillo Nuñez
Inaoe
jaramil@inaoep.mx

Resumen.-La frecuentecias del español mediante un enfoque paralelo. La decodificación de texto es un área de investigación que ha tenido un gran auge debido a que, cada día los sistemas de seguridad informática deben ser más robustos, para ello hay que detectar las debilidades que tienen dichos sistemas para que éstos tengan mayor seguridad. En éste artículo se muestra un algoritmo para la decodificación de texto con codificación Cesar utilizando las palabras másfrecuentes

1 INTRODUCCION

La criptografía se puede dividir en dos partes: **la encriptación** que se encarga de codificar mensajes para que solo sean recibidos por una persona y el **criptoanálisis** que se encarga de decodificar estos mensajes sin tener indicios de cómo se codifico o tener alguna clave para hacerlo, sin embargo, por lo regular solo se encuentra una parte del mensaje, por lo cual no se decodifica por completo [1].

En los últimos años los sistemas informáticos necesitan detectar las vulnerabilidades en cuanto a seguridad informática y la protección de datos, debido a que dichos sistemas deben ser robustos ante ataques informáticos, es importante implementar algoritmos de decodificación a los archivos encriptados para detectar las vulnerabilidades en los sistemas de información como se muestra en [3, 4], dichos ataques son más comunes hoy en día ya que los recursos computacionales que se pueden obtener son más accesibles, por el bajo costo de éstos.

Dentro de la criptografía se pueden encontrar dos tipos de encriptación, la encriptación por llave simétrica, en la que se usa la misma clave para codificar y decodificar el mensaje, mientras que la encriptación por llave anti simétrica utiliza una llave diferente para la encriptación y la decodificación [2].

La computación paralela ha tenido un gran auge en los últimos años, éste tipo de enfoques sirve para reducir el tiempo de ejecución en una aplicación informática. Dentro de las principales herramientas que existen para aplicaciones paralelas están Message Passing Interface (MPI) que son librerías para implementar programación paralela en una red de computadoras, los enfoques multicore que utilizan la totalidad de los núcleos contenidos en un procesador convencional, entre otros.

En este artículo se propone un algoritmo paralelo para decodificar texto encriptado en Cesar utilizando MPI utilizando el lenguaje de programación Python con base en frecuencias de palabras y una lista de las comunes en el idioma español [3], el texto elegido fue un conjunto de libros los cuales fueron añadidos en un archivo de texto y posteriormente fueron codificados mediante el algoritmo Cesar.

La organización del artículo es la siguiente en la sección 2 se presentan los trabajos relacionados, en la tercera sección la metodología propuesta es presentada y finalmente las conclusiones y trabajos futuros son presentados.

2 METODOLOGÍA PROPUESTA

En el presente trabajo se presenta un algoritmo para decodificar texto cifrado mediante el algoritmo Cesar, donde se utiliza un corrimiento en cuanto a la posición de las letras con respecto al orden que tienen en el abecedario, este texto se toma como la entrada del algoritmo propuesto, el diagrama de flujo

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

del programa se presenta en la Fig.1. El archivo codificado elegido es la unión de distintos libros en formato txt, dando como resultado el archivo de texto con peso de 120 MB.

Una vez que se tiene el archivo que se va a decodificar, se realizó una búsqueda en internet para determinar las palabras más frecuentes en el español, dicha lista se utilizó para realizar la decodificación del texto, sin embargo, algunas palabras fueron excluidas, tales como nombres propios y algunos calificativos, dejando solo palabras comunes.

La herramienta para procesar en paralelo que se eligió es MPI, en la cual se utilizó un nodo principal que es el que realiza un conteo de líneas de texto para que se realice un balaceo de la carga en cada nodo dependiendo del identificador de cada proceso. La arquitectura en la que se implemento la decodificación se muestra en la Fig. 2.

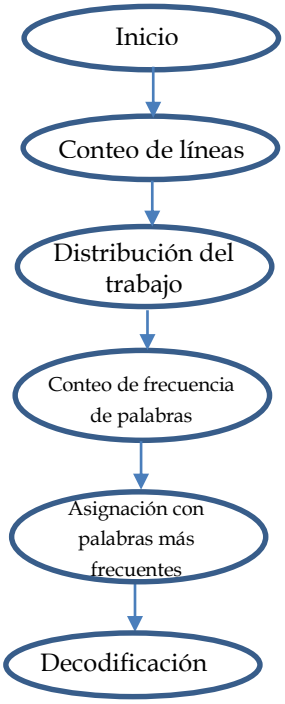

Fig. 1 Diagrama de flujo del algiritmo complete.

El primer paso que se realiza después de tener los dos archivos, es un pre-procesamiento de los archivos necesarios para decodificar el texto. El pre-procesamiento realizado es realizar un conteo del número de líneas en el nodo principal, este dato es enviado a cada nodo de procesamiento que realiza el balanceo de carga tomando en cuenta su identificador de proceso con la siguiente manera primero se encuentra la parte que le corresponde procesar a cada nodo, esto se realiza mediante una división del número total de líneas entre los nodos de procesamiento. A continuación, se realiza un cálculo de los límites de líneas que se van a procesar en cada nodo Fig. 2.

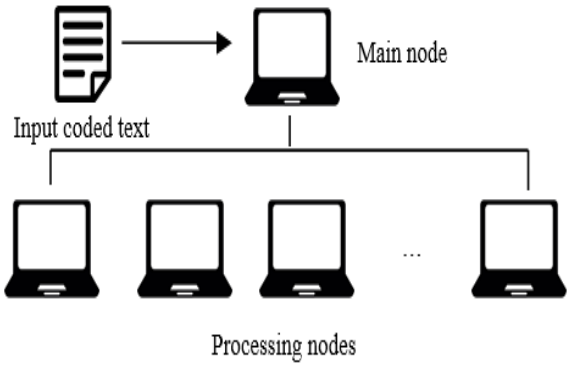

Fig. 2. Arquitectura de la implementación paralela.

Una vez que se calcularon los límites que tienen que ver con el procesamiento del archivo codificado, se comienza a realizar un conteo de frecuencias de palabras, en la que un diccionario *D* es utilizado y se guarda la frecuencia junto con la palabra, este proceso se realiza en paralelo en cada nodo de procesamiento. Cuando se cuenta con la frecuencia de las palabras que se encontraron en cada nodo, éstas son enviadas al nodo principal para realizar la unión de todos los diccionarios en uno solo para su posterior procesamiento.

Posterior a esto se ordenan los datos del diccionario con base en la longitud de la palabra y en la frecuencia de aparición en el texto encriptado, el siguiente paso es asociar las palabras del diccionario con la lista de palabras más frecuentes del español *LF*, ya que se procesó esta parte comienza la etapa de decodificación del texto.

Se comienza por analizar las palabras con frecuencia 1 y se asocian a las palabras de la misma frecuencia en un nuevo diccionario *L* donde se asocian las letras de cada palabra, para las palabras con frecuencia mayor a uno se realiza un recorrido por las palabras contenidas en *LF* de igual longitud y se cambian las letras de estas palabras de las cuales ya se tiene conocimiento previo y están contenidas en *L* una vez que se cambiaron las letras, se asocian las letras de las palabras de igual longitud, haciendo un recorrido por cada palabra, de tal manera que las letras que todavía no hayan sido asociadas se integren al diccionario *L*.

Como resultado del proceso anterior se obtiene el diccionario *L* en el que están asociadas las letras, finalmente se realiza la decodificación, se realiza un nuevo recorrido del archivo encriptado y se cambian las letras asociadas en cada línea, teniendo como resultado un archivo con el texto decodificado. El procedimiento del algoritmo puede ser visto gráficamente en la Fig. 3.

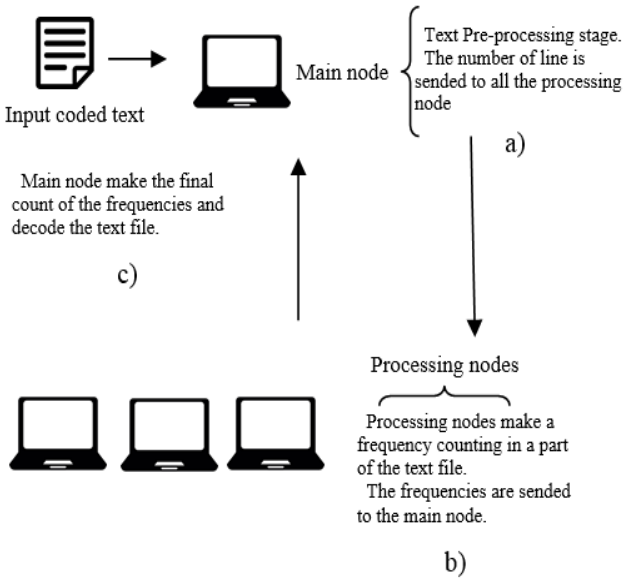

Figure 3. Procesamiento del algoritmo a) preprocesamiento en el nodo principal, b) procesamiento en los nodos de procesamiento, c) conteo final y decodificación

3 Pruebas y resultados

Para esta fase del proyecto se realizaron pruebas locales con un archivo de 123 MB en el cual están contenidos libros en formato txt encontrados en la red, dicho archivo fue encriptado mediante el método de cifrado Cesar.

La decodificación se realizó localmente en una computadora virtual con sistema operativo Linux Ubuntu 12.04, con 2 GB de memoria RAM y un núcleo con una velocidad de reloj de 3.3 Ghz.

Estás pruebas locales se realizaron para determinar que el porcentaje de decodificación del algoritmo el cual fue de 99.902559% con toda la lista de palabras, dependiendo de la longitud de palabras, se muestra en la Fig. 4 el porcentaje de decodificación, de esta forma se puede observar que el algoritmo fue contundente en cuanto la decodificación de éste archivo, hay que tomar en cuenta que la lista de palabras frecuentes puede cambiar para otro archivo.

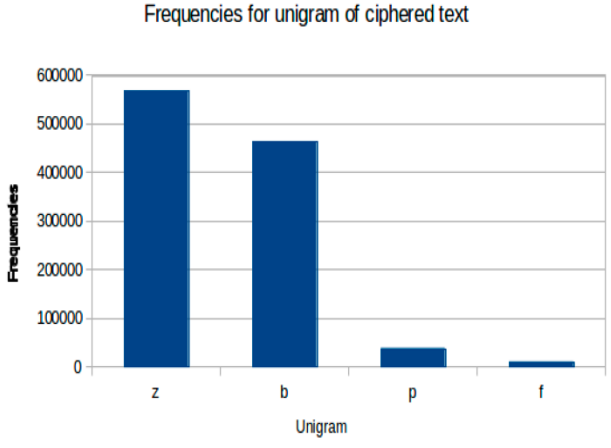

Figure 4. Frecuencia de palabras en el texto encriptado con longitud de uno.

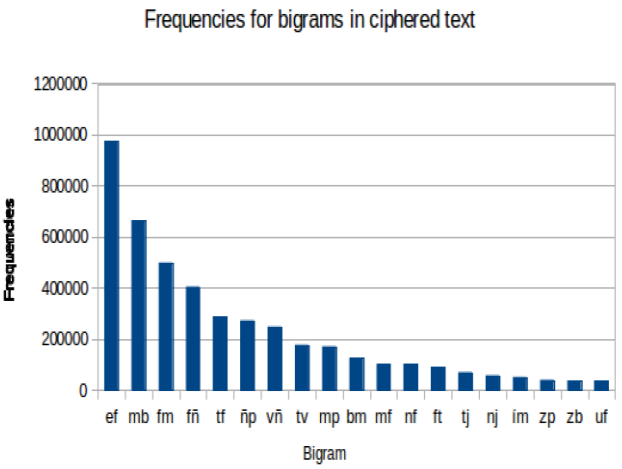

Figure 5. Frecuencia de palabras en el texto encriptado con longitud de dos.

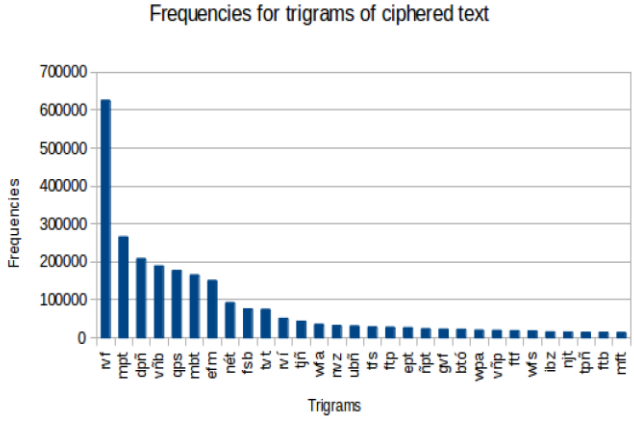

Figure 6. Frecuencia de palabras en el texto encriptado con longitud de tres.

Se puede observar que las frecuencias encontradas, con respecto a la longitud de las palabras se muestra en las Fig. 5 a la Fig 8. en la que se muestran las graficas de frecuencia de palabras en el texto encriptado. En la Fig. 4 se muestran las frecuencias de palabras con longitud uno, las cuales fueron asociadas con las palabras de longitud uno del texto prueba, las cuales fueron “y” con “z”, “a” con “b”, “o” con “p” y “e” con “f”, siendo la primera palabra la del texto encriptado y la segunda el resultado del texto prueba.

Para las palabras de longitud dos se asociaron nuevamente las palabras, como por ejemplo “de” con “ef”, “la” con “mb”, etc y se cambiaron las letras que se encontraron cuando se analizaron las palabras de longitud 1, por lo que se realiza una retroalimentación en cada paso. Para las palabras de longitud tres se realizaron los mismos pasos, sin embargo, por la retroalimentación de las etapas anteriores se realizó un cambio en algunas palabras decifrandolas y otras tomaron en cuenta para la decodificacion.

Para las palabras de frecuencia cuatro y cinco no fueron tomadas en cuenta todas las palabras del texto encriptado y del texto prueba, ya que muchas no aportan información para la decodificacion, de este modo solo las primeras treinta se tomaron en cuenta para dicha operación.

Para observar el rendimiento con la implementación paralela se realizaron pruebas en el Laboratorio Nacional de Sureste de México en el que se hicieron pruebas con varios nodos, los resultados son mostrados en la Fig. 9.

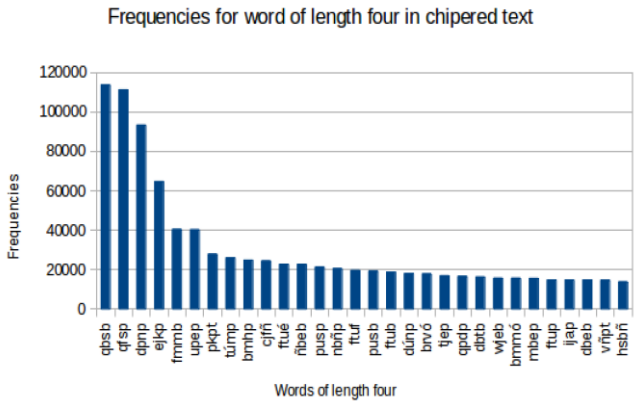

Fig 7. Frecuencia de palabras en el texto encriptado con longitud de cuatro.

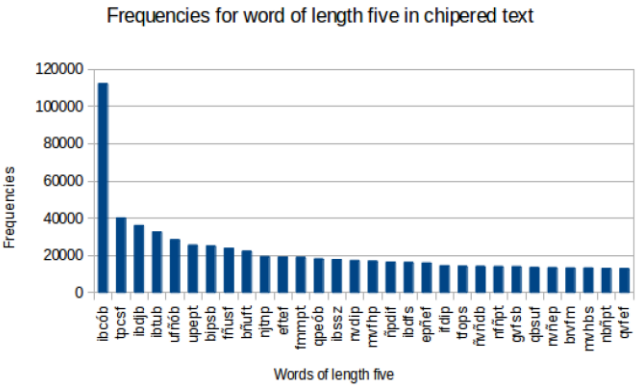

Figura 8. Frecuencia de palabras en el texto encriptado con longitud de cinco.

Percentage of decoding	Length
99.90255	<5
99.80445	5
99.57103	4
96.12853	3
81.32762	2
48.93695	1

Figure 9. Tiempos de ejecución del algoritmo de decodificación.

Como se puede observar a medida que se va aumentando el número de procesos el tiempo de ejecución disminuye debido a que la carga de trabajo se va reduciendo, a medida que se aumenta el número de procesos, tal como se muestra en la Figura 10.

Para observar como se va decodificando se muestra en la Fig 11 el texto encriptado, en las siguientes figuras (11-17) se muestra como se va decodificando una parte del mensaje codificado. Como se puede observar el mensaje se decodifica más siempre que se va aumentando la cantidad de palabras que se toma en cuenta, debido a que se aumenta el número de letras, ya que se tiene un mayor conocimiento sobre las letras que representan los caracteres en el texto decodificado.

Se puede concluir que siempre que se tomen en cuanta más palabra y de mayor longitud, hasta llegar a un porcentaje de decodificación de casi 100 por ciento.

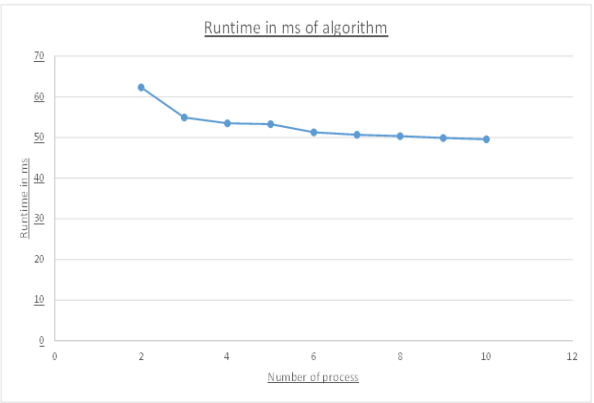

Fig 10. Tiempos de ejecución con diferente número de nodos.

```
mjcsp ef bvejp fejubep qps 3 vwft epcmft qvñup
mff fnf qf usft qvñup dpn..
fm gjmútpgp bvupejebdub
bcfñupgbjm (1110-1185)
gjmútpgp z níejdp ijtqbñpésbcf.
ñbdjú fñ hvbejy (hsbñbeb) ibdjb fm bop 1110,
gvf níejdp efm tvmuéñ bmnpibef bcü zb'rvc zvtvg
z pdvqú vñ qvftup ef jñgmvfñdjb fñ fm rvf mf
tvdfejú fm hsbñ gjmútpgp bwfsspft. fñ fm ñádmfp
ef tvt jefbt gjmptúgjdbt tf fñdvfñusb fm
qspcmfnb ef mb vñjúñ efm fñufñejnjfñup ivnbñp
dpñ ejpt. tv pcsb qsjñdjqbm gvf dpñpdjeb fñ
pddjefñuf dpñ fm uóuvmp ef fm gjmútpgp
bvupejebdup. fñ fmmb, bcfñupgbjm ftuvejb dúnp
ft qptjcmf rvf fm ipncsf fñ dpnqmfub tpmfebe
qvfeb bmdbñabs mb vñjúñ dpñ ejpt nfejbñuf fm
fñufñejnjfñup. usbt bñbmjabs mbt pqjñjpñft nét
jnqpsubñuft ef mpt gjmútpgpt bñufsjpsft b ím (
bwfnqbdf, bmhbafm, bwjdfñb, bmgbsbcó),
bcfñupgbjm fyqpñf mpt eftdvcsjnjfñupt rvf
sfbmjab fm qspubhpñjtub ef tv pcsb ibtub
bmdbñabs mb vñjúñ dpñ ejpt.
qsúmphp efm bvups
ifñ fm ñpncsf ef ejpt, dmfnfñuf z
njtfsjdpsejptp! cfñejhb ejpt b ñvftusp tfops
nbipnb z b tv gbnjmjb z dpnqbofspt, z efmft mb
qba.
npujwp pdbtjpñbm ef ftuf mjcsp: fm íyubtjt
nf qfejtuf, ifsnbñp tjñdfsp (ejpt uf eí mb
jñnpsubmjebe fufsñb z uf ibhb hpabs mb qfsqfuvb
qfmjdjebe), rvf uf dpnvñjdbtf brvfmmpt
```

Fig. 11 Texto cifrado

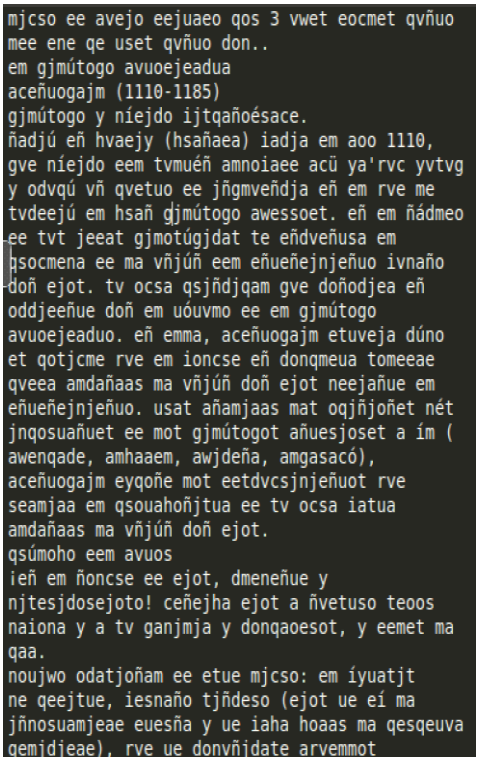

Fig. 12 Texto decodificado con letras extraidas de palabras con longitud uno.

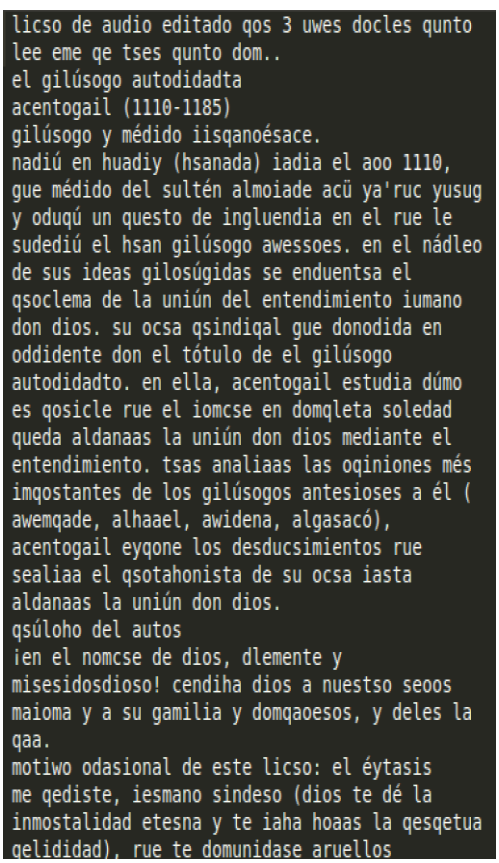

Fig 13. Texto decodificado con letras extraidas de palabras con longitud dos.

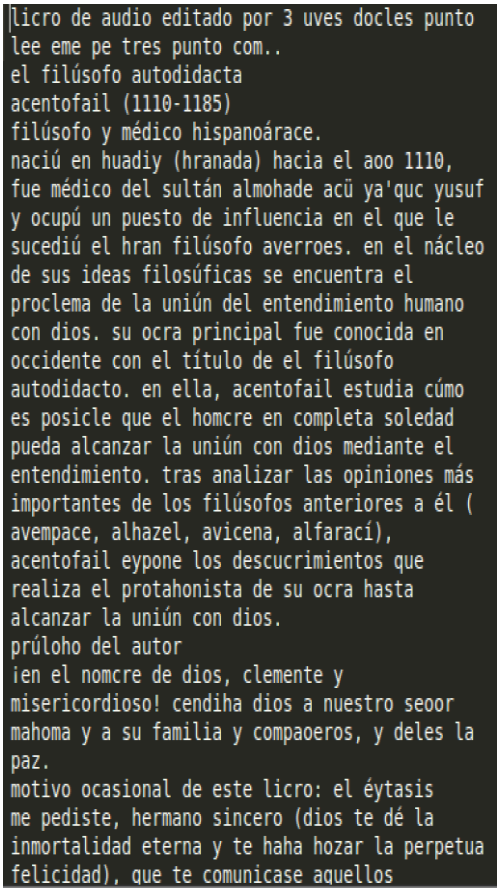

Fig. 14 Texto decodificado con letras extraidas de palabras con longitud tres.

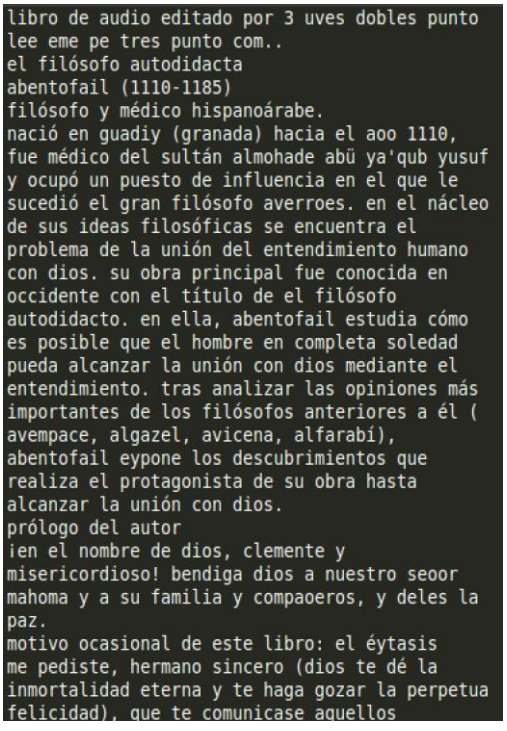

Fig. 15 Texto decodificado con letras extraidas de palabras con longitud cuatro

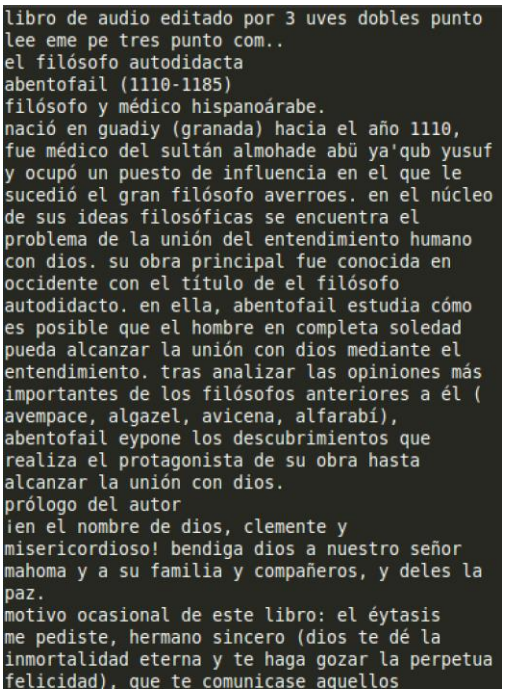

Fig. 16 Texto decodificado con letras extraidas de palabras con longitud cinco**.**

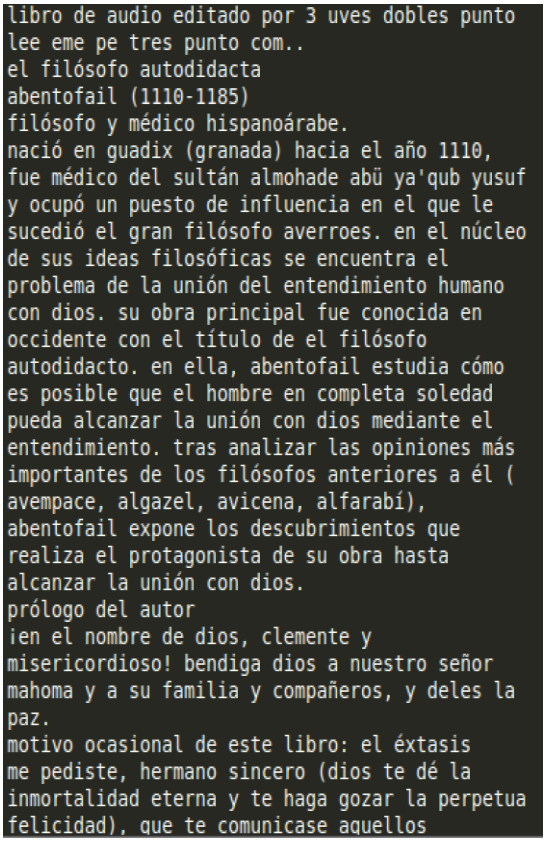

Fig. 16 Texto decodificado con letras extraidas de palabras con longitud mayor a cinco.

Conclusiones

En este trabajo se presenta un algoritmo paralelo implementado mediante un enfoque paralelo utilizando el lenguaje de programación Python y las librerías para programación paralela de MPI para la decodificación de texto encriptado basado en la frecuencia de palabras y una lista de palabras más frecuentes en el idioma español, el cual tuvo un buen rendimiento, se debe tomar en cuenta que la lista de palabras frecuentes debe ser cambiada si el texto es diferente.

Como trabajos futuros se pretende una forma de encontrar automáticamente la lista de palabras con base en otros archivos de prueba, además de implementar un algoritmo para que se realicen más procesos de manera paralela.

References

[1] Ashish Kumar, Himani Agrawal, “A Survey Report on various Cryptoanalysis Techniques”, Intenational Journal of Soft Computing and Engineering (IJSCE), vol. 3, no. 2, 2013

[2] Bo Yang, Kaijie Wu, Ramesh Karri, “Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard”, Proceedings of International Test Conference (2004).

[3] Bárbara Sánchez, María del Rocio Morales, Pablo León, “Decryption System of Thematic Text in spanich using Frequency Analysis Including Unigrams, Bigrams and Trigrams”, International Journal of Engineering and Innovate Technology (IJEIT), vol. 5, no. 6 (2015).

[4] Sánchez, B. Bigurra, Diana. et al. De-Encryption of a text in spanish using probability and statistics. 18th International Conference on Electronics, Communications and Computers: isbn 07695 3120 2 march 2008.