

Concepto de invocar (descomprimir)

Dada la naturaleza del método empleado, no podemos decir o afirmar que estemos descomprimiendo datos. Esto se debe a que los datos resultantes se derivan de un fractal o arreglo matemático. En realidad lo que ocurre es que la data inicial se arregló (organizar) y se ecualizó para posteriormente aplicarle una fórmula dinámica de correlación. [[Ver diagrama 4635](#)]

Como se mencionamos anteriormente y dado que en dicho proceso de (caracterización) utilizamos vectores o indicadores temporales (Vcore, CPU Speed, RTC, etc) aun cuando la estructura de datos sea la misma y se realicen de forma paralela dos o más procesos de compresión en un mismo dispositivo la (llave) o motor para descomprimir dicho arreglo o fractal diferirá una de otra.

Esto se debe a que al incorporar en el proceso los parámetros anteriores cualquier software o dispositivo que interactúe con el host que hospeda el motor de cifrado y compresión terminará alterando dichos parámetros y la resultante del computo de los mismos sera proporcional a dicha interacción. Lo anterior aplica para el caso de un dispositivo foráneo o mal intencionado, aun cuando tenga mas poder de procesamiento que el host su data resultante diferirá de la data o llave original. Para el caso de procesos malintencionados ejecutándose en el host, estos se enfrentan a la paradoja de que teóricamente deberían de ejecutarse más veloz que la velocidad máxima del procesador del propio host. Lo cual es imposible aun cuando nuestra app se ejecute en un entorno virtual, esto debido a que nuestros valores de referencia son base hardware.

Partiendo de lo anterior podemos concluir que cada motor o llave generado corresponde y solo produce datos coherentes y validos para su respectivo fractal o arreglo matemático. Esto es de gran utilidad ya que nos brinda la comodidad de disponer de grandes volúmenes de información en muy reducido espacio así como la seguridad e integridad de datos.

Entonces, al (invocar) una data en realidad lo que estamos realizando es obtener la misma a partir de una ecuación única. Esto es de gran utilidad para efectos de descargar o enviar volúmenes de información grandes como pueden ser contenido multimedia (streaming) o para desempacar y ejecutar aplicaciones o sistemas operativos de manera remota.

Para tales casos basta con (invocar) en el host remoto dicha llave y su respectivo fractal. Esto ultimo es relativamente fácil ya que la totalidad de los dispositivos de computo realizan operaciones lógicas y estas en su mayoría están traducidas a formato Hexadecimal. Mismo que utilizamos para nuestro proceso de computo, mismo que he publicado en un escrito titulado:

(Almacenamiento de datos en la red blockchain) @ <https://steemit.com/@azazel0101>
<http://io-exchange.com>

El primer paso de nuestro modelo es convertir la totalidad de un binario o texto plano a Hexadecimal y después traducir (asignar) valores numéricos a cada uno de los 16 elementos que componen el Hexadecimal.

Se denomina extemporánea ya que la data a calcular viene de una posición futura (desconocida) y por lo tanto cualquier intento fuera del cuerpo principal de nuestro programa genera variación en la resultante final. En dicha ecuación incorporamos un modulo WachtDog para garantizar y prevenir posibles depuración de procesos, sin importar si los mismos se realizan en un entorno virtual asociado a nuestro hardware o dispositivo externo. Ver documento adjunto [[fórmula extemporánea](#)]

Nivel (Proc)

Como comprenderán al adjuntar y relacionar diferentes neuronas estas se comportan como compuertas lógicas, mismas que están definidas en el cuerpo principal del programa. El nivel (proc) es nuestra estructura de programa, misma que no es accesible exteriormente ya que a diferencia del método de compresión en el cual se genera solo una llave. En este nivel generamos dos, una la utiliza el propio programa en si para indexar y expandir su data (muy interesante para AI autónoma) y la otra la retiene el programador o propietario. Una vez sellado y en funcionamiento nuestro programa, para poder corregir o modificar la estructura del mismo se requiere de ambas llaves. Esto es así ya que dichas neuronas se declaran en estado perpetuo [PPT]. Esto es análogo al modelo de encadenado (chaining) utilizado actualmente en aplicaciones de cifrado. Por lo tanto el propio programa en si evita ser modificado de manera externa ya que para realizar dicho proceso se debe de (romper) la estructura del programa y esto equivale a detenerlo (principio de conservación).

En próximas publicaciones se estará ahondando en dicho concepto y demás detalles así como las distintas funciones del mismo (operandos, mnemonicos, pseudomnemonicos, sintaxis, etc).

Estos dos conceptos (muy claros) se incluyen en esta publicación para efectos aclaratorios en lo referente a propiedad intelectual.