Sistema de almacenamiento de datos en la red Blockchain.

Este método esta entendido para almacenar información en la red blockchain basado en las transacciones (Tx). Para tales efectos se utiliza la parte decimal de la unidad o Token. El modelo propuesto es ascendente, comenzando con el primer carácter del mensaje cifrado y desplazando este al ultimo lugar del stack o pila de transacciones.

El ejemplo descrito a continuación consta de (n) lineas y 18 columnas siendo este ultimo valor proporcional al decimal del token.

Para un mensaje de entrada: “0123456789abcdefghijklmnopqrstuvwxyz”

la salida para el mismo después de cifrado con método:

AES CBC SHA256
Iv: automático
password: “abc123”

produce la cadena:

“5Krrq0UICVAKsk/VoFsRiEm+DzKkEzSQmBJQoaBXbWUIiJh173hFK0ZX1NShC
N9VIN9FbRudUcYksNEVmVygOybS64Zdrv97fUv4EDpCyTye1IPsHbOSiOj/XZq7
kOwcvdYyMGTQasvgwkH+0cQn1NNi3rZTx86Vea5yoa0XUNBi06qsN0W6G/Kbn
kctRNHV/v2cild5Az0XPl9+TiU3QO6XKPx4STJMPkgVgqVROrJpWlMgqy2MXwd4
RteL1uxV+oe7Ef1QHKApeMEiZIcwCFGjITHURbqyV43lIlQuNEdoIB3d8dVYoL9x
G3jdazXw0oeJum4aBt7Hlsu2/MmlbQ”

Esta cadena posee una longitud de 342 caracteres, el método consiste en sustituir cada carácter por su valor expresado en forma decimal.

Ejemplo: “5Krrq0UICVAKsk/VoF”

conversión a Dec: 53 75 114 114 113 30 85 73 67 86 65 75 115 107 47 86 111 70

formato de indexación:

indexar los caracteres de forma ascendente de derecha a izquierda en filas de 18 caracteres (18 decimales) en bloques de 3 filas:

0 1 0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0
7 1 8 4 0 1 7 6 8 6 7 8 3 1 1 1 7 5
0 1 6 7 7 5 5 5 6 7 3 5 0 3 4 4 5 3

Este ejemplo ocupa una longitud de fila de 18 decimales, por lo tanto para transferir o almacenar los datos del ejemplo anterior utilizamos 3 transacciones de:

0.010011000000011100
0.718401768678311175
0.016775556735034453

Token’s

Para almacenar el mensaje cifrado del ejemplo inicial dividimos la longitud de mensaje (M) por el decimal del token asignado al contrato inteligente.

Ejemplo:

342/18 = 19 * 3 = 57

siendo 3 el numero de filas por bloque de datos a indexar. Para el caso del ejemplo anterior ocuparíamos 57 Txn o transacciones para almacenar dicho mensaje. Se puede aumentar la capacidad de almacenamiento en función del decimal del token o utilizando solamente valores de entrada hexadecimales como los producidos por el Hash de mensaje.

Ejemplo:

sha512
“6ca13d52ca70c883e0f0bb101e425a89e8624de51db2d2392593af6a84118090”

Este formato solo acepta 16 caracteres únicos, siendo letras en minúsculas y números enteros sin signos ni puntuaciones. Para este ejemplo utilizamos decimal 18 y solo dos filas por bloque de almacenamiento:

“6ca13d52ca70c883e0”

para este tipo de codificación asignamos un valor numérico a cada carácter siendo:

1=01, 2=02, 3=03, 4=04, 5=05, 6=06, 7=07, 8=08
9=09, 0=10, a=11, b=12, c=13, d=14, e=15, f=16

por sustitución el ejemplo anterior queda:

06 13 11 01 03 14 05 02 13 11 07 00 13 08 08 03 15 00

indexado en la transacción queda:

0 1 0 0 0 1 1 0 1 1 0 0 1 0 0 1 1 0
0 5 3 8 8 3 0 7 1 3 2 5 4 3 1 1 3 6

Este ejemplo ocupa una longitud de fila de 18 decimales, por lo tanto para transferir o almacenar los datos del ejemplo anterior utilizamos 2 transacciones de:

0.010001101100100110
0.053883071325431136

Token’spara almacenar tola la cadena Hash (64 caracteres) utilizamos 7.11 transacciones. Siendo la parte decimal no utilizable llenado con ceros (00).

Nota:

El software de implementación de lo anterior con método de cifrado propietario enfocado en un sistema de mensajería segura para dispositivos móviles estará disponible en los próximos días una vez quede revisado.

Puedes revisar el contenido y progreso en:

http://io-exchange.com

J.C. Gómez. (c)