# ABSTRACT:

Year after Year, the exponentially increasing crime occurs in the tech world either through direct funds theft like bitcoin/NFT/digital Art theft or through Spam DDoS-Attacks on sensitive sites from Bots.

This report shows a method to convert a picture into a hash code using a hashing sequence similar to SHA256 where the image is converted to black and white then to binary then to a small unique hash value according to the algorithm designed by us.

This is the foundational algorithm and mechanism to how blockchain works

Image hashing is done for two reasons:

1. Digital Art/NFT verification using image hashing
2. New Captcha method to prevent Malicious Auto-Bots using picture check

# Index

Pg.no.

1. Abstract 2
2. Example 4
3. Flowchart 5
4. Explanation 6
5. Application 11
6. References 14

# EXAMPLE

Upload the image in the Anvil Web App

Graphical user interface, text

Description automatically generated with medium confidence

Output:

Graphical user interface, website

Description automatically generated

**FLOWCHART**

**Text

Description automatically generated with medium confidence**

**EXPLANATION**

**Step 1:** Image gets converted to greyscale and then to binary values of 0s & 1s where

0 🡪 black

1 🡪 white

**Chart, box and whisker chart

Description automatically generated**

**Step 2:**

The array of 0s and 1s is reshaped to form a long string

**A picture containing text

Description automatically generated**

**Step 3:** The 0s & 1s are converted to the corresponding letter using the following logic

111111111111 🡪 12 x 1s 🡪 12th capital letter 🡪 L

0000000 🡪 7 x 0s 🡪 7th Small letter 🡪 g

This is done by:

count the number of 1s and convert it to corresponding uppercase letter

count the number of 0s and convert it to corresponding lowercase letter

**Text

Description automatically generated**

Corresponding output:

**Text

Description automatically generated**

**Step 4:** Create a Hashing algorithm to convert this string into a hash value

1. split the long string into N blocks of 8 characters

Text

Description automatically generated with low confidence

1. Send the blocks for further hashing by a HashMap
2. Convert each character to its corresponding ascii value

Corresponding output:

**Text

Description automatically generated with medium confidence**

1. add all N blocks of ascii values to get one number

Corresponding output:

****

1. convert the sum → letters according to a Hash map A-I

example:

**Chart, histogram

Description automatically generated**

**Text

Description automatically generated**

1. even values become capital letters
2. odd values become small letters

**Step 5:** show final hash value

**Graphical user interface, text, application

Description automatically generated**

**APPLICATION**

**Digital Art/NFT verification using image hashing:**

Digital art is an artistic work or practice that uses digital technology as a part of the creative or presentation process

NFTs can really be anything digital such as drawings, digital art which uses hashing and blockchain technology to keep track of all the transactions for each trade.

****

**Unique Hash value: GcEhDhCcBdIcAdFiCbFaD**

When a buyer buys this unique digital art, it can be confirmed by the buyer that it is the authentic and real art piece by cross checking its hash value and confirming that it is the original digital art.

This method can be applied while trading/ selling NFTs (Non-Fungible tokens) to preserve the authenticity by comparing the hash values throughout all transactions

**New Captcha method to prevent Malicious Auto-Bots using picture check:**

reCAPTCHA protects the sensitive websites from spam and abuse.

Current captchas:

**Graphical user interface, text, application

Description automatically generated**

**Timeline

Description automatically generated with low confidence**

This method of prevention is obsolete, and the newer advanced bots can bypass this.

This new method of captcha can make a full proof way of preventing these bots

------------------------------------------------------------------------------------------------------

**Timeline

Description automatically generated with medium confidence**

Current Hash value : EfFiJhBfAhEcCfDdHfCdIa

Final hash value : FfCdDgHfDfAgHfEaGeAdBa

State : not solved

------------------------------------------------------------------------------------------------------**Timeline

Description automatically generated**

Current Hash value : DfGhJbJgGdFgJfFhJfCdFi

Final hash value : FfCdDgHfDfAgHfEaGeAdBa

State : not solved

------------------------------------------------------------------------------------------------------A living room with a couch and a table

Description automatically generated with low confidence

Current Hash value : FfCdDgHfDfAgHfEaGeAdBa

Final hash value : FfCdDgHfDfAgHfEaGeAdBa

State : solved

------------------------------------------------------------------------------------------------------

We are comparing the image hash value with its final hash value after which if same, access to a site will be granted

This will be useful as only humans can do this and cannot be easily solved by Bots thus protecting sensitive sites and servers

**REFERENCES**

1. S. Bhattacharjee and M. Kutter,” Compression tolerant image authentication,” Proc. ICIP-’98, vol. 1, pp. 435-439
2. ROBUST IMAGE HASHING R. Venkatesan1, S.-M. Koon2, M. H. Jakubowski1, and P. Moulin Cryptography Group, Microsoft Research
3. R. Venkatesan and S.-M. Koon,” Robust image hashing into binary strings,” manuscript, 1999