International Journal of Computer Science and Software Engineering (IJCSSE), Volume 5, Issue 7, July 2016
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 142-147

A Three-Layer Visual Hash Function Using Adler-32

Andysah Putera Utama Siahaan

Faculty of Computer Science, Universitas Pembangunan Panca Budi, Jl. Jend. Gatot Subroto Km. 4,5 Sei Sikambing, 20122, Medan, Sumatera Utara, Indonesia

andiesiahaan@gmail.com

ABSTRACT

The visual integrity needs to be implemented in sending a picture. There is various image received have no originality. The small change of the pixels does not make the picture content detected by the eye. The integrity validation is very important to be applied. The picture captured by a camera has two dimensions. It is described in pixels such as Width and Length. This study is to validate all the pixels data or the color intensity of both dimensions. If there are a modification in the pixel, this method will give the wrong hash data. The validator will analyze the pixels in every layer such as red, green and blue to ensure the data transmitted is correct. Once there is a slight change in the pixels, the calculation gives the wrong value. It is very useful to compare the image before and after transmission.

Keywords: *Hash Function, Adler-32, Security.*

1. INTRODUCTION

Integrity is an aspect that ensures that the data must not be changed without the permission of the authorized competent [3]. For the application of digital imaging, integrity aspect is paramount. It contains the confidential information [2]. Access to data is often sought after by intruders [6][9]. The picture that has been submitted cannot be changed by the unauthorized parties. Violation of this would result in malfunctioning of the validation. It is especially in the fields of education, medicine, military, etc. It needs to prove the originality of the content. Some of them are used as the evidence of a fact. The integrity validation is not only saving someone's life, but it can be implemented on a security side. The highly use of imaging system leads to the data exchange over the air while attaching to the international network. While communication it is imperative to verify the message so that intruder cannot replace with the fake information [5].

For example, when the computer sends the picture, the third party can intercept it in the air, modify the content of the picture and send it to its destination. We should send them with the verification or the message digest. When the receiver checks the hash of the image, they can compare it to the hash send simultaneously. Sometimes, we do not understand what they are. The information retrieved is used directly without verification. Once we start it, it might run the script consists of some trojan or virus lines. Certain methods of checksums provide the way to verification to ensure the visual integrity [6]. Adler-32 presents to be a practical approach to help manage the originality.

2. THEORIES

2.1 Data Integrity

Security goals cover three points such as availability, confidentiality, and integrity [7]. A Understanding data integrity broadly refers to the confidence of resource system. Data integrity is paramount because it can ensure the data accuracy, consistency, accessibilities, and the high quality. Following the integrity rules is important. Data with integrity is identical on hold during any operation such as business transfer, storage or retrieval. In simple computer terms, data integrity is the assurance that data is consistent, certified and referenced. Data integrity means that the accuracy and correctness. Data integrity in a database system must be maintained to keep the truth of the stored data. Figure 1 illustrate the scheme of data integrity.

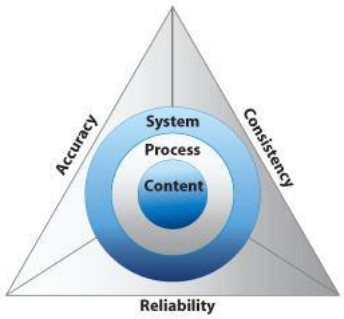

Fig. 1. Data integrity scheme.

An Example of integrity is the relationship between parents and children. The relationship is noted in a record according to genealogy. If the parent record has one or more of the child records related to all, the database itself will take care of the referential integrity. It automatically guarantees the accuracy, consistensi and reliabily of the data, so there is no record of children raised without parents, and no parent will lose their child's records. It also ensures that no one can remove the parent record while the parents have a record of each child.

2.2 Hash Function

A way to test the integrity of the data is to provide a checksum or a sign that data is not changed. The easiest way to do is to calculate the existing characters so that if there is a change, the result will be different. A hash function is a one-way function that produces a "checksum" or "fingerprint" of the data. A message that passes to the hash function will produce output called Message Authenticated Code (MAC). Hash function mapped out a set of data into a limited smaller size. The calculation algorithms uses matrix to map the byte array [1][4]. Let us take a simple example, the mathematical modulus function. The result of the modular expression is the remainder of integer division. For example, "12 mod 5" produces a value of 2, because 12 divided by 5 to produce a value of 2 and the remainder is 2. Every day we use modulo operation to express the hours where the modulo is 12.

The mod operator cannot be used as a good hash function without integrated to the other formula. There are a few requirements to be used practically. For example, the range of the result of the hash function should be enough so that the probability of two different messages will generate the same hash function output. It should be emphasized the word "probability", because there will be two pieces of data that can generate the same hash function output. This is due to the range of hash functions is smaller than the space of the input. To make two messages are intelligible and have the same hash function output is not easy. Another requirement of a good hash function is the change of the character or single bits in the data must produce different output. This property is called avalanche effect.

2.3 Adler-32

Mark Adler invented the Adler-32 hash function. He created in 1995 and modified the Fletcher checksum. The length is same as CRC. It offers the speed of validation process. He claimed that Adler-32 is more reliable than Fletcher-16 and slightly less reliable than Fletcher-32. It is obtained by calculating two 16-bit checksums A and B and concatenating their bits into a 32-bit integer. It runs on the hexadecimal platform. A is the sum of all bytes in the stream plus one, and B is the sum of the individual values of A from each step. At the beginning of an Adler-32 run, A is initialized to 1, B to 0. The sums are done modulo 6552. The bytes are stored in network order, B occupying the two most significant bytes [8].

The function may be expressed as

$$A = 1 + D_1 + D_2 + \ldots + D_n \pmod{65521}$$

$$B = 1 + D_1) + (1 + D_1 + D_2) + \ldots + 1 + D_1 + D_2 + \ldots + D_n) \pmod{65521}$$

$$= n \times D_1 + (n-1) \times D_2 + (n-2) \times D_3 + \ldots + D_n + n \pmod{65521}$$

$$Adler\text{-}32(D) = B \times 65536 + A$$

2.4 Image Preprocessing

The most important thing in visual hash function is the grayscale process. It is mostly done in the picture processing is changing the color image into the grayscale image, it is used to simplify the model image. The color image consists of three layers, red, green and blue. The grayscale process is to mix the layers and produce a single color layer. When there is a calculation performed using a three-layer, it will be changed by grouping the third layer becomes grayscale and the result is a grayscale image. In this image, there is no color, only the gradation of black and white. There are three ways to get the grayscale intensity.

$$Grayscale = \frac{\max(R,G,B)+\min(R,G,B)}{2} \quad (1)$$

$$Grayscale = \frac{R+G+B}{3} \quad (2)$$

$$Grayscale = (0.21 * R) + (0.72 * G) + (0.07 * B) \quad (3)$$

The formulas above describe how to get the grayscale intensity. Formula 1 concerns to the lightness, Formula 2 concerns to the average and Formula 3 concers to the luminosity. Formula 1 is to find the highest and lowest values of the value of R, G, B, then the highest and lowest values are summed and then multiplied by 0.5. Formula 2 is adding up all the value of R, G, B, then divided by 3, to obtain an average value of R, G, and B. Formula 3 is to multiply each value of R, G, B with a certain constant predefined value, then the result of multiplying the entire value of R, G, B add up to one another.

3. PROPOSED WORK

In this research, we plan to calculate the color intensities of every layer. The color image has three layers of color

intensities. The red, green and blue layer must be combined and averaged. The average of its color is stored in a grayscale section. We do not build three checksums. However, we combine the three layers into a single converted layer and calculate the pixels. The first step is to split the colors and build the new intensity.

RED				
R^{11}	R^{12}	R^{13}	R^{14}	R^{15}
R^{21}	R^{22}	R^{23}	R^{24}	R^{25}
R^{31}	R^{32}	R^{33}	R^{34}	R^{35}
R^{41}	R^{42}	R^{43}	R^{44}	R^{45}
R^{51}	R^{52}	R^{53}	R^{54}	R^{55}

GREEN				
G^{11}	G^{12}	G^{13}	G^{14}	G^{15}
G^{21}	G^{22}	G^{23}	G^{24}	G^{25}
G^{31}	G^{32}	G^{33}	G^{34}	G^{35}
G^{41}	G^{42}	G^{43}	G^{44}	G^{45}
G^{51}	G^{52}	G^{53}	G^{54}	G^{55}

BLUE				
B^{11}	B^{12}	B^{13}	B^{14}	B^{15}
B^{21}	B^{22}	B^{23}	B^{24}	B^{25}
B^{31}	B^{32}	B^{33}	B^{34}	B^{35}
B^{41}	B^{42}	B^{43}	B^{44}	B^{45}
B^{51}	B^{52}	B^{53}	B^{54}	B^{55}

Fig. 2. Red, Green and Blue color intensities

The previous figure describes the extracted pixel of a 5 x 5 image length. It splits into three parts. R11 to R55 represents the red color, G11 to G55 represent to the green color while B11 to B55 represent to the blue color. Then the grayscale evaluation converts the values into a single value. We can choose one of the formulas exists. The following equation shows how it performs.

$$I = \frac{R + G + B}{3} \quad (4)$$

Where:

I : New Intesity
R : Red Color Intensity
G : Green Color Intensity
B : Blue Color Intensity

GRAYSCALE				
I^{11}	I^{12}	I^{13}	I^{14}	I^{15}
I^{21}	I^{22}	I^{23}	I^{24}	I^{25}
I^{31}	I^{32}	I^{33}	I^{34}	I^{35}
I^{41}	I^{42}	I^{43}	I^{44}	I^{45}
I^{51}	I^{52}	I^{53}	I^{54}	I^{55}

Figure 3. New Intensity

As we can see in Figure 3, the values inserted into the cells are obtained from the above formula. It aims to reduce the amount of hash function. If we do not combine the color intensities, we have to make the separated hash evaluation and of course, it makes the computer performance slower.

There are three formulas that calculate the Adler-32 Hash to generate the integrity value.

$$A = 1 + \sum_{i=0}^{N} I \quad (5)$$

$$B = \sum_{i=0}^{N} A \quad (6)$$

$$D = B.65536 + A \quad (7)$$

Where:

I : Grayscale Intensity
A : The sum of all bytes
B : The sum of the individual values of A
D : Adler-32

The Adler-32 is obtained by multiplying B to 65536. The value 65536 is derived from a 16-bit hexadecimal maximum value. It happens since Adler-32 consists of 16-bit (A) and 16-bit (B) sections. The 32 comes from 16 + 16.

4. PROPOSED WORK

This test runs a 15 x 10 pixels color image. We try to analyze the Alder-32 hash value if there is a change in color intensity. Figur 4 below shows the original image.

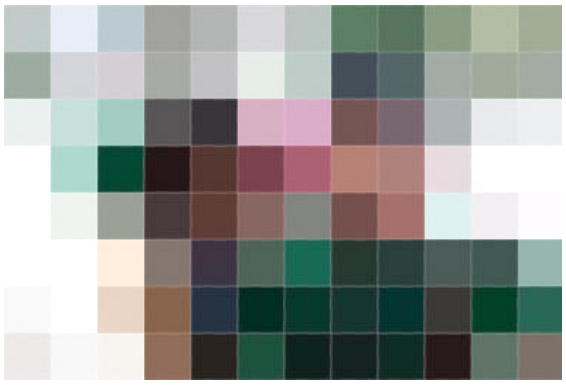

Fig. 4. A 10 x 15 image

Let's take an example, pixel 1 located at cell number 1 (Column = 1, Row = 1) consists of R = 192, G = 203 and B = 202. The Grayscale is $\frac{192+203+202}{3}$ = 200. This calculation continues until reach the end of the pixel or reach pixel 150 where the grayscale of the last pixel is $\frac{124+114+105}{3}$ = 114.

Table 1: Grayscale Intensites

GRAYSCALE INTENSITIES											
200	238	199	161	181	216	192	107	99	142	177	161
162	213	211	168	194	235	197	79	97	166	162	165
239	214	186	89	57	197	198	96	111	178	234	239
254	199	40	28	62	90	126	144	144	223	255	254
255	240	155	64	70	114	131	91	128	235	242	253
254	254	237	122	60	90	69	48	54	87	77	169
247	254	214	103	53	28	36	43	35	56	37	76
235	248	243	115	35	59	26	28	31	29	104	114

Tabel 1 shows the complete grayscale calculation of the previous image. This table will be the further data to find the Alder-32 value. The value is obtained by applied the earlier Alder-32 formula to these pixels value. Table 2 describes the overall process of Alder-32.

Table 2: Overall process of Alder-32

No.	A	B
1	1	0
2	201	201
3	439	640
4	638	1278
5	799	2077
6	980	3057
7	1196	4253
8	1388	5641
9	1495	7136
10	1594	8730
11	1736	10466
12	1913	12379
13	2074	14453
14	2236	16689
15	2449	19138
16	2660	21798
17	2828	24626
18	3022	27648
19	3257	30905
20	3454	34359
21	3533	37892
22	3630	41522
23	3796	45318
24	3958	49276
25	4123	53399
26	4362	57761
27	4576	62337
28	4762	67099
29	4851	71950
30	4908	76858
31	5105	81963
32	5303	87266
33	5399	92665
34	5510	98175
35	5688	103863
36	5922	109785
37	6161	115946
38	6415	122361
39	6614	128975
40	6654	135629
41	6682	142311
42	6744	149055
43	6834	155889
44	6960	162849
45	7104	169953
46	7248	177201
47	7471	184672
48	7726	192398
49	7980	200378
50	8235	208613
51	8475	217088
52	8630	225718
53	8694	234412
54	8764	243176
55	8878	252054
56	9009	261063
57	9100	270163
58	9228	279391
59	9463	288854
60	9705	298559
61	9958	308517
62	10212	318729
63	10466	329195
64	10703	339898
65	10825	350723
66	10885	361608
67	10975	372583
68	11044	383627
69	11092	394719
70	11146	405865
71	11233	417098
72	11310	428408
73	11479	439887
74	11726	451613
75	11980	463593
76	12194	475787
77	12297	488084
78	12350	500434
79	12378	512812
80	12414	525226
81	12457	537683
82	12492	550175
83	12548	562723
84	12585	575308
85	12661	587969
86	12896	600865
87	13144	614009
88	13387	627396
89	13502	640898
90	13537	654435
91	13596	668031
92	13622	681653
93	13650	695303
94	13681	708984
95	13710	722694
96	13814	736508
97	13928	750436

There are 96 (12 x 8 pixels) + 1 calculations. No. 1, A = 1 and B = 0 is the inital state. The last A shows 13928 and the last B shows 750436. The calculation is not ended.

A = A % MOD_ADLER
= A % 65521
= 13928 % 65521
= 13928

B = B % MOD_ADLER
= B % 65521

= 750436 % 65521
= 29705

AD = B . 65536 + A
= 29705 . 65536 + 13928
= 1946760808 (decimal)
= 74093668 (hexadecimal)

The Adler-32 value showed above is still in decimal format. Alder-32 runs in hexadecimal. The value in hexadecimal is 74093668. It is a combination of two 16-bit value. The first section is 7409 and the last is 3668. When sending this picture to the receiver, the sender must send this Adler-32 value (74093668) simultaneously. Afterward, the recipients synchrony their hash with the sender. Once the value is different, there must be an error or interception while transmitting over the air.

What about if the content has been modified or there are a small undetected object has been inserted into the picture. It is time to prove the hash function. Assume that we modify pixel number 1. The earlier values are R = 192, G = 203 and B = 202. The new values are R = 190, G = 203, and B = 201. It is a small change. It cannot be detected by naked eyes. We just modified the red and blue colors; the green keep similar. The originality can be detected only by using the computer program. That is why the sender always sends the integrity value with the picture; it is to protect the information inside.

Tabel 3: The modified process

No.	A	B
1	1	0
2	199	199
3	438	637
4	636	1273
5	796	2069
6	978	3047
7	1195	4242
8	1387	5629
9	1494	7123
10	1593	8716
11	1735	10451
12	1912	12363
13	2073	14436
14	2237	16673
15	2451	19124
16	2662	21786
17	2830	24616
18	3024	27640
19	3258	30898
20	3455	34353
21	3534	37887
22	3631	41518
23	3797	45315
24	3959	49274
25	4124	53398
26	4363	57761
27	4577	62338
28	4763	67101
29	4851	71952
30	4907	76859
31	5104	81963
32	5303	87266
33	5399	92665
34	5510	98175
35	5688	103863
36	5922	109785
37	6161	115946
38	6415	122361
39	6613	128974
40	6653	135627
41	6681	142308
42	6743	149051
43	6832	155883
44	6959	162842
45	7103	169945
46	7247	177192
47	7470	184662
48	7725	192387
49	7979	200366
50	8234	208600
51	8474	217074
52	8629	225703
53	8693	234396
54	8763	243159
55	8877	252036
56	9008	261044
57	9099	270143
58	9227	279370
59	9462	288832
60	9704	298536
61	9957	308493
62	10211	318704
63	10465	329169
64	10703	339872
65	10824	350696
66	10884	361580
67	10974	372554
68	11044	383598
69	11092	394690
70	11146	405836
71	11233	417069
72	11310	428379
73	11479	439858
74	11728	451586
75	11982	463568
76	12196	475764
77	12300	488064
78	12353	500417
79	12380	512797
80	12416	525213
81	12459	537672
82	12494	550166
83	12550	562716
84	12587	575303
85	12663	587966
86	12897	600863
87	13145	614008
88	13388	627396
89	13503	640899
90	13538	654437
91	13597	668034
92	13623	681657
93	13652	695309
94	13683	708992
95	13712	722704
96	13816	736520
97	13930	750450

Table 3 illustrates the process after we modify several color intensities. The modified A shows 13930 and the modified B shows 750450. Moreover, these values are completely different. The calculation is different from the earlier since there was a modification of the byte array.

A = A % MOD_ADLER
= A % 65521

= 13930 % 65521
= 13930

B = B % MOD_ADLER
= B % 65521
= 750450 % 65521
= 29719

AD = B . 65536 + A
= 29719. 65536 + 13930
= 1947678314 (decimal)
= 7417366A (hexadecimal)

The hexadecimal value is 7417366A. If we compare to the previous value (74093668) or although we change only 1 bit, the hash value is entirely different. This method is used to testing the level of image originality. Every byte in array is connected each other. If we modify one of them, it affects to the rest.

5. CONCLUSION

We wish to thank Mark Adler personally for the Alder-32 checksum algorithm. This algorithm runs fast for image processing. The Adler-32 value is obtained by concatenating two 16-bit A and B. It will be a 32-bit integer. Since it does not use the complex arithmetic expression, it can be applied to the bigger picture. Alder-32 can calculate the originality of what senders send to the recipients. This research does not provide the information hiding; it is only to ensure what the senders send are what the receivers get. The picture can prove anything in real life. So it should be original if used as evidence. His feedback on this research made the algorithm can work together with the image processing.

REFERENCES

[1] A. P. U. Siahaan, "Three-Pass Protocol Concept in Hill Cipher Encryption Technique," SNATI, Yogyakarta, 2016.
[2] A. P. U. Siahaan, "RC4 Technique in Visual Cryptography RGB Image Encryption," International Journal of Computer Science and Engineering, vol. 3, no. 7, 2016.
[3] B. Forouzan, Cryptography and Network Security, McGraw-Hill, 2006.
[4] H. Anton dan C. Rorres, Elementary Linear Algebra, 2011: John Wiley & Sons.
[5] R. Bhanot dan R. Hans, "A Review and Comparative Analysis of Various Encryption Algorithms," International Journal of Security and Its Applications, vol. 9, no. 4, pp. 289-306, 2015.
[6] S. K. Das, G. Sharma dan P. K. Kevat, "Integrity and Authentication using Elliptic Curve cryptography," Imperial Journal of Interdiscliplinary Research, vol. 2, no. 5, 2016.
[7] D. Shah, "Digital Security Using Cryptographic Message Digest Algorithm," International Journal of Advance Research in Computer Science and Management Studies, vol. 3, no. 10, pp. 215-219, 2015.
[8] M. Adler, "Wikipedia," Wikipedia, 22 3 2016. [Online]. Available: https://en.wikipedia.org/wiki/Adler-32. [Diakses 8 7 2016].
[9] A. P. U. Siahaan, "BPCS Steganography Noise-For Region Security Improvisation," International Journal of Science & Technoledge, vol. 4, no. 6, 2016.