International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Comparison of Performance of Boneh-Shaw Finger Printing Codes with Tardos Under Randomized Bits Collusion Attacks

Alok Tripathi[1],Rajiv Pandey[2],Amarjeet Singh[3]

Submitted: 10/10/2022 **Revised:** 25/12/2022 **Accepted:** 20/01/2023

Abstract: Fingerprinting codes are needed for the protection of copyright issues of digital media. Before distributing digital media these codes are embedded in it. These codes are unique for each digital media. When illegal copies of these digital media are made they are identified by unique fingerprinting code inserted in them. In this way, culprits making illegal copies are identified. But these fingerprinting codes are subject to collusion attacks. In Collusion attacks, culprit users conspire together to instigate attacks against fingerprint codes. The culprit users cartelise and does the modification in the fingerprinting code of the digital media . The fingerprinting codes are changed in such a way so that an innocent user is found the culprit. Thus, these attacks are bottlenecks in protecting the digital rights of digital media. This research compares results of Boneh-Shaw fingerprinting codes with tardos code while launching Randomized bits Collusion attacks on both the codes.

Keywords: *Boneh-Shaw fingerprinting codes, Tardos-Codes, Randomised Bits Collusion Attacks,Piracy Protection.*

1. Introduction

1.1 Digital Water Marking

In digital watermarking, a marker is secretly embedded in an audio, video, or image file. This marker is used to identify the owner of the audio, video, or image file. "This marker does not have any relation to the original data of the audio, video, or image file. Thus simply we can say that these watermarks are used to verify the authenticity and integrity of the digital media. Thus they are used to check the piracy of digital media.

[1],*Amity Institute of Information Technology, Lucknow, Amity University, Uttar Pradesh, India, gkp.alok@gmail.com*

[2]*Amity Institute of Information Technology, Lucknow, Amity University, Uttar Pradesh, India, rpandey@lko.amity.edu*

Sri Ram Institute of Management and Technology Kashipur, Uttrakhand, India, singhamarjeet.9@gmail.com

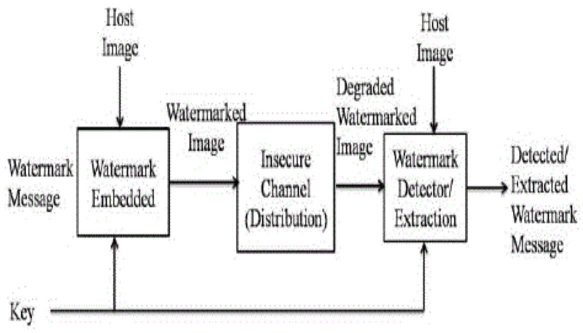
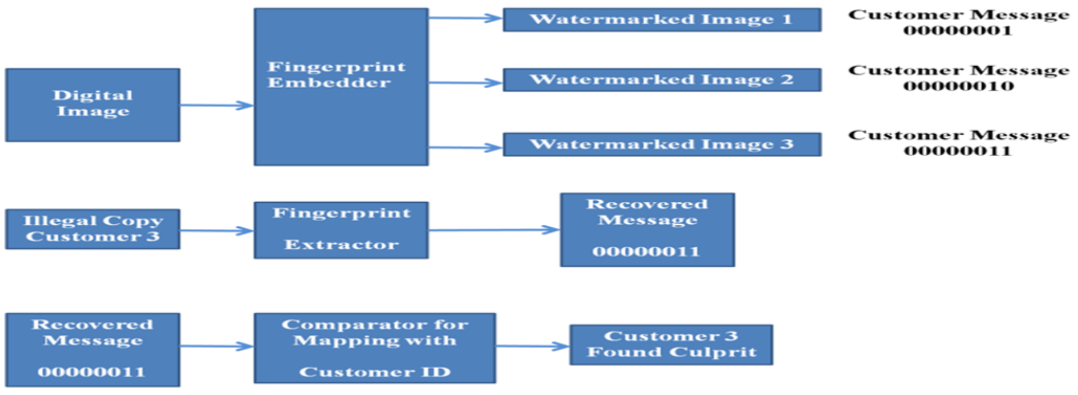

Host Image
Watermarked Image
Degraded Watermarked Image
Host Image
Watermark Message
Watermark Embedded
Insecure Channel (Distribution)
Watermark Detector/ Extraction
Detected/ Extracted Watermark Message
Key

Fig. 1. Digital Water Marking

The above figure exhibits the whole process.

1.2 Digital Finger Printing

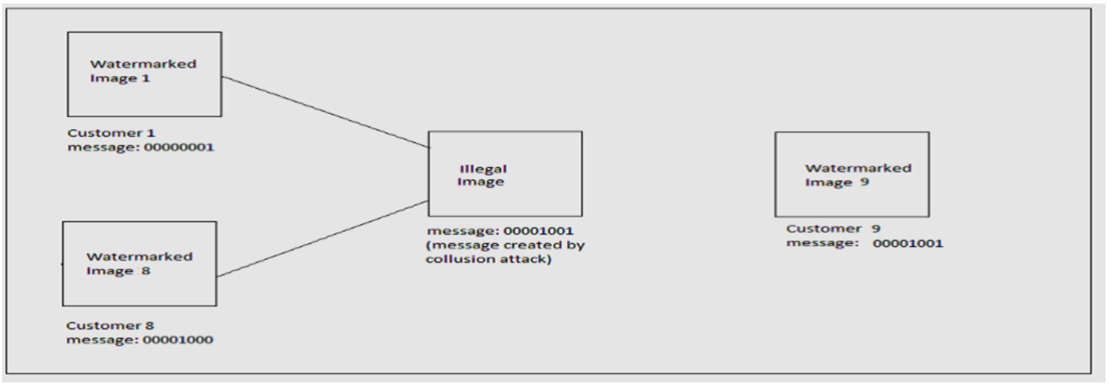
In Digital Fingerprinting unique codes are constructed using various algorithms . These unique codes are then embedded into the audio, video, or image file. Now one to one mapping of the codes embedded is done with the digital media file corresponding to that code and a separate database is maintained.. If a copyright violation is found after the distribution of these files the comparison of databse maintained with the fingerprinting code that is unique is done and the piracy of digital media is detected. Thus it is easy to catch the guilty usesr doing the piracy. Thus these markers are used to check the piracy of digital media. The following figure exhibits the whole piracy detection process.



Digital Image
Fingerprint Embedder
Watermarked Image 1
Watermarked Image 2
Watermarked Image 3
Customer Message 00000001
Customer Message 00000010
Customer Message 00000011
Illegal Copy Customer 3
Fingerprint Extractor
Recovered Message 00000011
Recovered Message 00000011
Comparator for Mapping with Customer ID
Customer 3 Found Culprit

Fig. 2. Digital Finger Printing

1.3 Collusion Attack on Digital Fingerprinting Codes



Watermarked Image 1
Customer 1 message: 00000001
Watermarked Image 8
Customer 8 message: 00001000
Illegal Image
message: 00001001 (message created by collusion attack)
Watermarked Image 9
Customer 9 message: 00001001

Fig. 3. Collusion Attack

The copyright infringement detection process illustrated above is subject to attacks that are categorized as collusion attacks. In these attacks, a collection of users who are doing the copy-right infringement may collude together to change the distinctive fingerprinting code. These colluders collude together to create a distinctive fingerprinting code and embed it inside pirated digital media and sometimes this unique code embedded by colluders matches with the unique code of an innocent user and he is found guilty. These collusion attacks are launched by matching the data of their digital media file with another user media file and where a difference is found the data is manipulated. The diff command of Linux operating system is useful here. This cartelization results in generating the copy of digital media that violates the copy right by manipulating the digital finger printing code. The illustarion below exhibits the whole .

1.4 Boneh & Shaw Fingerprinting codes[1]

The problem of piracy of digital media due to collusion was solved by Boneh & Shaw.Boneh&Shaw derived a finger printing code that is secure aginast these collusion attacks. The length of this code is $O(n^3\log(n/\varepsilon))$ where ε is used to reperesent the error rate.

1.4.1 Code Construction

In Boneh & Shaw code let us denote the total number of users as n and the rate of error as Ɛ. Now we will generate a matrix where the number of rows in the matrix will be equal to the total number of users and the number of columns in the matrix will be equal to the length of the fingerprinting code. Now the methodogy illustrated below will be used to genearate the matrix.

$$Y(n,d)=\underbrace{(Y1Y1Y1Y1)}_{d\text{ times}}\underbrace{(Y2Y2Y2)}_{d\text{ times}}\ldots\underbrace{(Yn\text{-}1Yn\text{-}1\ldots\ldots\ldots\ldots Yn\text{-}1)}_{d\text{ times}}$$

Y(4,3) for users A,B,C,D is defined by

$$Y(4,3) = \begin{array}{ccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & A \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & B \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & C \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D \end{array}$$

The random permutation π will be applied to the columns present in Y(n,d) .The random permutation π is considered for those rows which equal those of users..Here Bm is considered to be the position's set of the columns Ym mapped by π,|Bm|=d. Consider if π=(π1,π2…………..πdn-1) then

Bm={π1|(m-1)d+1 ≤1 ≤md}

Here it should be noted that

{1,2,………d(n-1)}=B1υB2υ……………….υBn-1

Here the coloumns of Y(n,d) are permuted which is derived by partitioning {1,2,……….,dn-1} in to B1,B2,………………..Bn-1 due to the repetition of column . Hence

$$\binom{d(n-1}{d, d, --d} = (d(n-1)^{\lrcorner}$$

are the real different permutations of Y(n,d) for $2 \leq s \leq n-1$ which define Rs=Bs-1U Bs

Let us assume that the following permutation is considered for Y(4,3)

π = (7, 3, 2, 4, 9, 5, 1, 6, 8) then

$$\pi(Y(4,3)) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

B1={2,3,7} B2={4,5,9} B3={1,6,8}

R2={2,3,4,5,7,9}

R3={1,4,5,6,8,9}

1.4.2 Colluders detecting Algorithm for Boneh & Shaw Code

Let us asumme we are provided with $x \varepsilon \{0,1\}^{d(n-1)}$ then we have to derive a subset of collusion that produced x .

1. Each bit is put 0.
2. User 1 is concluded as guilty if $w(X|B_1)>0$.
3. User n is concluded as guilty if $w(X|B_{n-1}) <d$
4. For users S=2, 3,……n-1

The user S is concluded as guilty if

It is assumed that $k=W(X|R_s)$

If $W(X|B_{s-1}) < k/2-\sqrt{\binom{k}{2}\log n/\varepsilon}$

1.5 Tardos Fingerprinting codes

Tardos also soved this problem of collusion by deriving a code whose length is $m=100c^2k$ where c signifies the collusion size here k is taken as log(1/epsilon) and epsilon is taken as error rate.

1.5.1Generation of code

Tardos represented the finger printing code as a binary matrix of size n by m. Here n represents the total number of users and m represents the length of the fingerprinting code. The size of the fingerprinting code is decided by c the collusion size, the total number of users n and the error rate epsilon. The length of fingerprinting codes m is taken as m=100c2k. Then an empty two-dimensional Boolean matrix is taken to store the fingerprint codes.The elements of the finger print code matrix are assigned 0 or 1 value based on its probability value Each element in the fingerprint code matrix is assigned as 0 or 1 based on its probability value p,0<p<1. The length of the finger printing code will deteremine the probability values.The number of probability values will be m if the length of the finger printing code is m.These probability values will be distributed randomly and identically between t to 1-t, for t=1/300c. Let us compute these probablity values.Let us take t=sin2t' .To compute t'(0 < t' <π/4), we will generate a random value r in the range from t' to π/2 – t'.Then the p=sin2r will be used to calculate the final probability value. The distributor reserves this probability array for accusation.

1.5.2 Accusation

The Tardos code algorithm to determine the accusation is highly dependent on probability array and the fingerprinting code matrix. A matrix U has been defined by Tardos code where n is the number of rows in the matrix and m is the number of coulumns in the matrix.Here n is total number of users and m is the length of the finger printing code. The enteries of this matrix are defined by Tardos as follows(The pi symbol used here is taken as each element in the probability matrix and Xji symbol used here is each element of the fingerprint code matrix)

$$U_{ji} = \sqrt{\frac{1-p_i}{p_i}} \quad \text{if} \quad Xji = 1$$

$$U_{ji} = -\sqrt{\frac{p_i}{1-p_i}} \quad \text{if } Xji = 0$$

Here the Tardos logic catches a user j where the traitor copy $y=\{1,1\}^m$ is taken as input if:

$$\sum_{i=1}^{m} y_i U_{ji} > Z$$

1.6 Randomized Value Collusion Attacks

Here in this attack attacker compares bit by bit of the digital media files. If the bits match the same bit is copied at the position of that bit in the illegal copy but where their difference is observed while comparing then it is assigned a 1 or 0 Randomly at that bit position

2. Related work

1. The mentioned paper discusses an algorithm for the construction of a digital finger printing code known as Boneh & Shaw. This paper theortically discusses this code and claims it to be secure against collusion attacks.This paper also discusses a traitor detection algorithm derived for this type of code.The distribution methods for this code is also discussed in this paper.

2. This paper improves the Boneh & Shaw Codes. In this paper Boneh & Shaw code has been considered as primitive.This paper discusses about the construction of this code and acuusation algorithm for this Code.In this paper the code has been explained theoretically without the experimental results. Our research report analyzes the performance of Boneh & Shaw code and Tardos Code by attacking them with Randomized Bits Collusion attack.The performance of both the codes has been evaluated under Randomized Bits Collusion attack and then a performance analysis has been prsesnted for both the codes.In our research report Java simulator has been used for performance analysis of Boneh-Shaw Code whereas C++ is used for performance analysis of Tardos Code.The performance analysis of both the codes have been derived experimentally and then a detailed comparison has been done.

3. Proposed Work

For the analysis of Boneh-Shaw code following work is done and could be summarized as follows:

- For the construction of Boneh&Shaw Code simulator developed in Java has been used.
- For Collusion detection in Boneh & Shaw Code Java simulator has been developed.
- For analysis of Boneh & Shaw Code under Randomised Bits Collusion Attack a simulator developed in Java has been used to launch the Randomised Bits Collusion Attack.
- To detect the traitor for collusion the collusion detection algorithm for the Boneh&Shaw Code has been simulated in Java and this simulator has been used to detect the traitor.

For the analysis of Tardos code following work is done and could be summarized as follows:

- For the construction of Tardos Code simulator developed in C++ has been used.
- For Collusion detection in Tardos Code C++ simulator has been developed.
- For analysis of Tardos Code under Randomised Bits Collusion Attack a simulator developed in C++ has been used to launch the Randomised Bits Collusion Attack.
- To detect the traitor for collusion the collusion detection algorithm for the Tardos Code has been simulated in C++ and this simulator has been used to detect the traitor.

4. Experimental Results

The comprehensive experimentation has been done and the results of the experiment are exhibited here on the basis of which deep analysis of performance of both the codes has been done.The result below is for 40 users.

4.1 Illustration of Results Boneh-Shaw Code

No of Users	Size of Collusion	Error rate %	Length of Finger Printing Codes	% of False Positive	% of False Negative
40	2	5	920712	0	0
40	3	5	920712	0	0
40	5	5	920712	0	0
40	10	5	920712	0	0

Table-1. Deduced results at 5% error (Boneh-Shaw Code)

From the table it is clear that at 5% error rate with increase of collusion size the code length is constant at 920712 and % false positive and %false negative are nil

No of Users	Size of Collusion	Error rate %	Length of Finger Printing Codes	% of False Positive	% of False Negative
40	2	10	834210	0	0
40	3	10	834210	0	0
40	5	10	834210	0	0
40	10	10	834210	0	0

Table-2. Deduced results at 10% error (Boneh-Shaw Code)

From the table it is clear that at 10% error rate with increase of collusion size the code length is constant at 834210 and % false positive and %false negative are nil

No of Users	Size of Collusion	Error rate %	Length of Finger Printing Codes	% of False Positive	% of False Negative
40	2	15	783627	0	0
40	3	15	783627	0	0
40	5	15	783627	0	0
40	10	15	783627	0	0

Table-3. Deduced results at 15% error (Boneh-Shaw Code)

From the table it is clear that at 15% error rate with increase of collusion size the code length is constant at 783627 and % false positive and %false negative are nil

No of Users	Size of Collusion	Error rate %	Length of Finger Printing Codes	% of False Positive	% of False Negative
40	2	20	747708	0	0
40	3	20	747708	0	0
40	5	20	747708	0	0
40	10	20	747708	0	0

Table-4. Deduced results at 20% error (Boneh-Shaw Code)

From the table it is clear that at 20% error rate with increase of collusion size the code length is constant at 747708 and % false positive and %false negative are nil

No of Users	Size of Collusion	Error rate %	Length of Finger Printing Codes	% of False Positive	% of False Negative
40	2	30	697125	0	0
40	3	30	697125	0	0
40	5	30	697125	0	0
40	10	30	697125	0	0

Table-5. Deduced results at 30% error (Boneh-Shaw Code)

From the table it is clear that at 30% error rate with increase of collusion size the code length is constant at 697125 and % false positive and %false negative are nil

No of Users	Size of Collusion	Error rate %	Length of Finger Printing Codes	% of False Positive	% of False Negative
40	2	40	661230	0	0
40	3	40	661230	0	0
40	5	40	661230	0	0
40	10	40	661230	0	0

Table-6. Deduced results at 40% error (Boneh-Shaw Code)

From the table it is clear that at 40% error rate with increase of collusion size the code length is constant at 661230 and % false positive and %false negative are nil

No of Users	Size of Collusion	Error rate %	Length of Finger Printing Codes	% of False Positive	% of False Negative
40	2	50	633360	0	0
40	3	50	633360	0	0
40	5	50	633360	0	0
40	10	50	63360	0	0

Table-7. Deduced results at 50% error (Boneh-Shaw Code)

From the table it is clear that at 50% error rate with increase of collusion size the code length is constant at 63360 and % false positive and %false negative are nil

4.2 Illustration of Results-Tardos Code

Total Users	Collusion Size	Percentage of the rate of error	Code's length	Avg No. Of False Positive	Avg No. Of False Negative
40	2	5	1198	35.5	0
40	3	5	2696	21.55	0.75
40	5	5	7489	33.15	0.2
40	10	5	19172	18.2	3.15

Table-1. Deduced results at 5% error rate(Tardos Code)

From the table it is clear that at 5% error rate with increase of collusion size the code length increases and Average No of false positive are significant and Average false negative are always less than Avearge False Positive

Total Users	Collusion on size	Percentage of the rate of error	Code's length	Avg No. Of False Positive	Avg No. Of False Negative
40	2	10	921	33.2	0
40	3	10	2072	28.93	0.3
40	5	10	5756	31.0	0.25
40	8	10	14736	17.25	4.2
40	10	10	23025	3.8	4.05

Table-2. Deduced results at 10% error rate(Tardos Code)

From the table it is clear that at 10% error rate with increase of collusion size the code length increases and Average No of false positive are significant and Average false negative are always less then Avearge False Positive except at collusion size 10 where Average False negative is slightly higher than Average False Postive

Total Users	Collusion Size	Percentage of the rate of error	Code's length	Avg No. Of False Positive	Avg No. Of False Negative
40	2	15	758	33.2	0
40	3	15	1707	28.93	0.3
40	5	15	4742	31.0	0.25
40	8	15	12141	17.25	4.2
40	10	15	18971	3.8	4.05

Table-3. Deduced results at 15% error rate(Tardos Code)

From the table it is clear that at 15% error rate with increase of collusion size the code length increases and Average No of false positive are significant and Average false negative are always less then Average False Positive except at collusion size 10 where Average False negative is slightly higher than Average False Positive

Total Users	Collusion Size	Percentage of rate of error	Code's length	Avg No. Of False Positive	Avg No. Of False Negative
40	2	20	643	33.1	0
40	3	20	1448	31.4	.0.3
40	5	20	4023	32.1	0.2
40	8	20	10300	21.2	2.3
40	10	20	16094	3.9	3.7

Table-4. Deduced results at 20% error rate(Tardos Code)

From the table it is clear that at 20% error rate with increase of collusion size the code length increases and Average No of false positive are significant and Average false negative are always less then Average False Positive except at collusion size 10 where Average False negative is slightly higher than Average False Positive

Total Users	Collusion size	Percentage of the rate of error	Code's length	Avg No. Of False Positive	Avg No. Of False Negative
40	2	30	481	28.55	.05
40	3	30	1083	25.45	.65
40	5	30	3009	26.85	0.9
40	8	30	7705	24.75	1.9
40	10	30	12039	4.95	4.8

Table-5. Deduced results at 30% error rate(Tardos Code)

From the table it is clear that at 30% error rate with increase of collusion size the code length increases and Average No of false positive are significant and Average false negative are always less than Average False Positive.

Total Users	Collusion size	Percentage of the rate of error	Code's length	Avg No. Of False Positive	Avg No. Of False Negative
40	2	40	366	36.1	0
40	3	40	824	26	0.7
40	5	40	2290	32.15	0.3
40	8	40	5864	24.4	2.15
40	10	40	9162	26.4	0.95

Table-6. Deduced results at 40% error rate(Tardos Code)

From the table it is clear that at 40% error rate with increase of collusion size the code length increases and Average No of false positive are significant and Average false negative are always less than Average False Positive.

Table-7. Deduced results at 50% error rate(Tardos Code)

Total User s	Collusion Size	Percentage e of the rate of error	Code's length	Avg No. Of False Positive	Avg No. Of False Negative
40	2	50	277	34.15	0
40	3	50	623	21.95	0.6
40	5	50	1732	31.45	0.4
40	8	50	4436	15.15	4.45
40	10	50	6931	29.45	0.5

From the table it is clear that at 50% error rate with increase of collusion size the code length increases and Average No of false positive are significant and Average false negative are always less than Average False Positive

4.3 Analysis

4.3.1 Boneh-Shaw Code

i. If Analysis is considered for the length of the Boneh-Shaw code then it has been observed the length of code shortens as the error increases .Thus length of the Boneh-Shaw Code is higher at low error rates.

ii. For the analysis of length of code with respect to collusion size it has been observed that length is constant for all collusion sizes.

iii. If we consider analysis for false positives and false negatives from above experimentaional results it has been observed that under the Randomised Bits Collusion Attacks Boneh-Shaw Code shows no false positives and negatives.

So from the above analysis following assumptions could be drawn:

I. If we have to use a lower length Boneh-Shaw code then we heve to accommodate higer error rate.

II. We can accommodate any error rate with no impact on False Positives or False negatives.

III. It can be concluded that Boneh-Shaw Code sustains against Randomized Bits Collusion attack as Average False Positives and Average False Negatives are Nil at all Collusion Sizes and error rates.

4.3.2 Tardos –Code

i. If Analysis is considered for the length of the Tardos code then it has been observered the length of code shortens as the error increases .Thus length of the Tardos Code is higher at low error rates.

ii. For the analysis of length of code with respect to collusion size it has been observed that as low collusion size the length of code is smaller but as coullusion size increases the length of the code increases.

iii. If we analyse for false positives and false negatives for Tardos Code we deduce that Average false naegatives are less than Avearage false postives.It has also been observed that average false negative is very insignificant.

iv. If we consider Collusion size then it is deduced that average false positive is higher with smaller collusion size but as the collusion size increases it becomes low.

v. If we consider with respect to error rates it is deduced that average false positive tends to be low at lower error rates but it tends to be high at higher error rates

vi. Thus it can be concluded that Tardos code is sustainable to Randomized Bits Collusion Attacks if we consider from false negative point of view but if we consider from false positive point of view average false positive decreases at higher collusion sizes but it shows significant average false positive at higer error rates hence finally it could be assumed that Tardos code sustains Randomized Bits collusion attacks in case of big collusion sizes and lower error rates.

4.3.3 Summary Thus the above analysis could be summarized as follows:

- From the sustainibility point of view considering the performance of Tardos code Under Randomized Bits collusion attack it is assumed that at higher error rates and small collusion sizes small length of Tardos code could be used

- The error rate should be low and collusion size should be more than the expected collusion size to reduce the average false positive.

From the above Analysis following inference could be drawn for the optimization of the two codes under Randomize Bits Collusion attacks.

1. Boneh-Shaw code outperforms the Tardos Code under Randomised Bits Collusion Attacks if false positives and false negatives are taken into consideration.
2. Tardos Code optimizes to a smaller code length in comparison to Boneh-Shaw Code under Randomised Bits Collusion Attacks.

5. Conclusion

Thus In this work the Boneh-Shaw code and Tardos code have been analysed in detailed from the experimental point of view under Randomized bits' collusion attack and detailed analysis is done of the simulated results and it has been concluded that from the analysis and optimization point of view Boneh-Shaw code seems to outperform Tardos Code if Average false positives and false negatives are considered but from optimization point of view in which code length is criteria Tardos code is better.

References

[1] Dan Boneh and James Shaw .Collusion-Secure ingerprinting for Digital Data. IEEE Transactions on Information Theory, Vol. 44, N0. 5, September 1998.

[2] G.Tardos(2003) ,Optimal Probablistic Fingerprint Codes,Proceedings of the 35th Annual ACM Symposium on Theory of Computing 2003

[3] Tripathi Alok,Pandey Rajiv , Analysis of Boneh-Shaw Finger Printing Codes under Majority Value Collusion Attacks International Journal of Computer Applications (0975 – 8887) Volume 167 – No.3, June 2017.

[4] Analysis of Boneh - Shaw Finger Printing Codes under randomized bits collusion attack. International Journals of Computer Science and Information Security Pittsburgh. PA-USA (ISSN1947500) May 2017 Edition.

[5] Analysis of Boneh - Shaw Finger Printing Codes under majority value collusion attack. International Journals of Computer Applications IJCA June 2017 Edition, Foundation of Computer Science New York USA. ISSN-0975-8887.

[6] Evaluating Performance of Boneh-Shaw Finger Printing Codes under Minority Value Collusion Attacks. The 12th International Conference on Computational Intelligence and Communication Networks (CICN 2020)- IEEE-Conference Bhimtal-25-26 Sep 2020.

[7] Simulating Tardos Finger Printing Codes under Randomized Bits Collusion Attacks. ICCCIS-2021, International Conference on Computing,Communication, and Intelligent Systems 19th-20th February, 2021 Greater Noida, India

[8] Alaria, S. K. "A.. Raj, V. Sharma, and V. Kumar.“Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language Using CNN”." *International Journal on Recent and Innovation Trends in Computing and Communication* 10, no. 4 (2022): 10-14.

[9] S. K. A. “An Improved Algorithm for Faster Multi Keyword Search in Structured Organization”, *International Journal on Future Revolution in Computer Science & Communication Engineering* Vol-5, issue- 5 (2019), 19–23.

[10] S. K. A. “Improving the Performance of Heterogeneous Hadoop Clusters Using Map Reduce”. *IJRITCC* 2019, *7*, 11-17.