https://doi.org/10.2352/ISSN.2470-1173.2021.4.MWSF-345

A Close Look at Robust Hash Flip Positions

Martin Steinebach; Fraunhofer SIT, Darmstadt, Germany

Abstract

Images can be recognized by cryptographic or robust hashes during forensic investigation or content filtering. Cryptographic methods tend to be too fragile, robust methods may leak information about the hashed images. Combining robust and cryptographic methods can solve both problems, but requires a good prediction of hash bit positions most likely to break. Previous research shows the potential of this approach, but evaluation results still have rather high error rates, especially many false negatives. In this work we have a detailed look at the behavior of robust hashes under attacks and the potential of prediction derived from distance from median and learning from attacks.

Motivation

Digital images have become an important subject of forensic investigation. Depending on the case, the content shown in these images is manifold. Some contents like child pornography are illegal. Other contents only become illegal if they are used in an specific manner, like for example blackmailing, leakage of hacked content or revenge porn. These images can become evidence or at least potential evidence. Most often searching for these images in a larger set of images is necessary. This can be done off-line during the investigation of storage media, or on-line with crawling and filtering methods.

In both cases, privacy can become an issue: if the content of a photo is of private nature and there is only the suspicion of an illegal distribution, the person shown on the photo will want to ensure that as little compromising information is shared with the rest of the world, including forensic investigators or filtering agents. Here a technical challenge arises: images will not be recognized by direct comparison, but by matching features or hashes. These hashes can either be very well suited to ensure privacy in the cases of cryptographic hashes but will fail to re-identify an images after even the slightest changes to the image file. Or, in the cases of robust hashes or feature vectors, they will tolerate changes but also cause some leakage of image structure and potentially content.

Recently concepts have been presented that combine robust and cryptographic hashes: a robust hash is hashed by a cryptographic hash function [1] or used as an input to a bloom filter [2]. This concepts have one flaw or drawback: Hashes are used fundamentally different by robust and cryptographic hash functions. The former compute similarity, the latter require identity. If used together with a cryptographic hash, the resulting hash becomes significantly less robust as the change of a single bit of the robust hash will lead to a different distributed hash with no chance of re-identification.

To counter this, assumptions are made in [1] to predict which hashes are likely to change and to 'neutralize' these bits. In our work we analyze how well these assumptions can be made and if a detailed analysis of the hash behavior can provide an improvement to the robustness of the proposed protocols.

Block Hash and Privacy

Several robust or perceptual hashes for various media types are known, which provide different levels of robustness. For example, Roover et al. [3] provide an image hash algorithm which is robust against geometrical operations like scaling and rotation; the hash draws its robustness from the use of the Radon transform. Friedrich and Goljan propose an approach based on random noise similarity in [4]. As there are too many algorithms to mention here, we recommend surveys like the one by Haouzia et al. [5] or Neemila and Singh [6]. There are also methods for audio and video streams as well as text data. Hash evaluation in the literature provides information about the total performance of the hash. No individual hash bits are discussed, but only the overall hash detection rate. This means, there is no data comparable to our results in this paper.

In 2006, Bian Yang et al. proposed a block mean value based perceptual image hash function [7]. Four slightly different methods are proposed. The latter two additionally incorporate an image rotation operation to enhance robustness against rotation attacks. This increases the computational complexity of the latter two methods. Here we focus on the simplest method:

- Convert the image to grey scale and normalize the original image into a preset size.
- Let n denote the bit length (usually 256 bit) of the final hash value. Divide the pixels of the image I into non-overlapped blocks $I_1, I_2, ..., I_n$.
- Calculate the mean of the pixel values of each block. That is, calculate the mean value sequence $M_1, M_2, ..., M_n$ from the corresponding block sequence.
- Calculate the median value $\tilde{M}$ of the mean value sequence M.
- Normalize the mean value sequence into a binary form and obtain the hash value h(i) as in equation 1.

$$h_i = \begin{cases} 0 & for\, M_i < \tilde{M} \\ 1 & for\, M_i \geq \tilde{M} \end{cases} \quad (1)$$

In 2012, we suggested a number of improvements to this [8]. One is based on the observation that the individual hash bits are not equally robust or stable. Depending on their distance to the median value, they are more likely to skip. Hash bits with a small difference between M_x and $\tilde{M}$ are supposed to be less robust than those with a large difference. The reliability hr_i of each hash bit is calculated as in equation 2. As the distance from media differs from image to image depending on texture, contrast and dynamics, hr is normalized as in equation 3

$$hr_i = |M_i - \tilde{M}| \quad (2)$$

$$nhr_i = \frac{hr_i}{max(hr)} \tag{3}$$

From this behavior a second distance besides the hamming distance is introduced which takes into account the hamming distance as well as the distance from the median. This distance significantly decreases the error rates of the hash algorithm.

In 2019, the idea of hash bit reliability was utilized in a concept for privacy and robust hashes[1]. To prevent the leakage of information about the content of an image by the robust hash, a method was introduced to combine robust and cryptographic hashes. Either all hash bits of reliability nhr_i below a threshold $thr(nhr)$ are set to neutral or the n bits with the lowest values nhr_i are set to neutral. Their hash values then do not influence the calculation of the cryptographic hash of the hash sequence.

$$h_i = \begin{cases} h_i & for\, nhr_i > \text{thr(nhr)} \\ neutral & for\, nhr_i \leq \text{thr(nhr)} \end{cases} \tag{4}$$

Evaluation Data

In this section we briefly describe the data generated for the robust hash behavior analysis in the following sections.

Data Set

We used 500 photos randomly picked from a smart-phone camera memory card and all downscaled to 1,000 pixel at the longest edge. This includes photos which are out of focus, contain motion blur and bad light conditions. This aims to simulate real-world scenarios where the actual usage of robust hashing and privacy protection could be of relevance. Downscaling to 1,000 pixel was done to keep the storage size of the testset small and the evaluation process lightweight. As the robust hash algorithm first executes a downsizing to 16x16 pixel, we assume the starting size of 1,000 x 750 pixel to be sufficient.

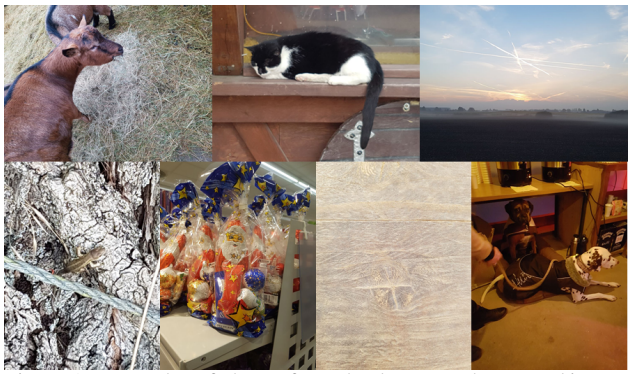


Figure 1: Examples of photos from the data set. The second image in the first row and the first in the second row are out of focus. The fourth in the second row suffers under bad light conditions.

Evaluation Run

Each photo was compressed with JPEG quality factors 100 to 1 with step size 1, resulting in 100 versions of each photo. While it is obvious that in real-world applications extremely low quality factors will not be applied, we still analyzed these as they could potentially provide a maximum set of flipped hash bits allowing the prediction of flipping at a better quality. It was also scaled by factors between 1.5 and 0.5 with step size 0.01, also resulting in another 100 versions of the photos.

For each photo the robust hash was computed and stored. Also the distance from the median as described in the ForBild [8] hash was stored. This resulted in 256 hash bit and 256 distance values for each version of the photo.

This data set is the foundation of the following analysis of the robust hash behavior. It allows a detailed allocation of hash bits flipped due to transformations and the relationship between bit flipping and distance from median.

Observations

In this section we discuss the various findings from the analysis of the data generated as described in the section before. We first describe the general behavior with respect to JPEG compression and scaling. Then we have a look at the role of the hash bit's distance from the median.

Hamming Distance

We first have a look at the hamming distances of the test data described in the previous section.

As JPEG compression can be seen as a continuous reduction of image quality with falling quality factor, the results are not surprising: with falling quality, the hamming distance between the robust hashes of the reference and the test image rises.

As average and median tend to hide extreme values, we also provide the maximum hamming distance for each quality factor from the 500 examples in table 1. Here one can see that while average and median stay low even at quality factor 10, maximum values are significantly higher: already at quality factor 70, a hamming distance of 104 occurred. The first time a hamming distance above 64 (or 25%) was observed was with quality factor 75 for image 215. The max column also shows a noteworthy behavior: At quality factor 90 the max hamming distance is 18, but drops to 9 at 80, rises to 104 at 70 and falls back to 22 at 60.

A better analysis of the distribution of hamming distances for the 500 image for different quality factors is provided by table 2. In contrast to table 1 here we can see the number of images falling into hamming distance categories between 0 and 128, calculated by rounding the log2 of the respective hamming distances. The table shows how well the robust hash stays at or under a desired hamming distance of 8 as already observed in [8].

Table 1: JPEG Hamming Distances

Quality Factor	Hamming Distance		
	Average	Median	Max
100	0.152	0	18
90	0.208	0	18
80	0.088	0	9
70	0.886	0	104
60	0.79	0	22
50	1.076	0	102
40	1.37	1	111
30	1.404	1	110
20	2.128	1	126
10	3.236	2	136
1	12.202	8	151

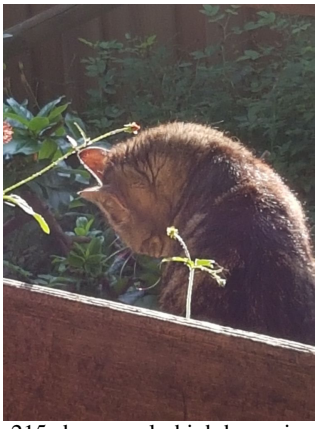


Figure 2: Image 215 shows early high hamming distances to the reference under JPEG compression.

The impact of scaling on the hamming distance was less linear than with JPEG quality. It could be assumed that with larger distance from the original size (increase as well as decrease) the hamming distance would grow. With size decreasing from the original size, the hamming distance increases, but significant local differences can be observed. As these are values averaging 500 images, quality loss due to different scale factors seems to be an overall challenge here.

We again provide a more detailed look at the values in table 3. One can see that hamming distance maxima are above 100 even for small size changes. Average and median values are nonetheless satisfying for most scale factors. A brief look at image 215 shows that here again it reacts rather strong on changes: its average hamming distance for all size changes is 82.

In the following section we will concentrate on the JPEG quality factor results as their more linear character make it more easy to explain and analyze the behavior of robust hash bits.

Table 3: Size Change Hamming Distances

Scale Factor	Hamming Distance		
	Average	Median	Max
1.49	3.71	3	102
1.40	3.434	2	109
1.30	6.954	4	152
1.20	5.674	4	124
1.10	6.058	4	110
0	0	0	0
-1.10	7.566	5	123
-1.20	6.152	4	112
-1.30	7.416	5	110
-1.40	8.95	6	123
-1.50	10.306	7	126

Table 2: Occurences of hamming distances of 500 image for the given JPEG quality factors

	Hamming Distance Category								
QF	**0**	**1**	**2**	**4**	**8**	**16**	**32**	**64**	**128**
90	444	43	6	3	3	1	0	0	0
70	324	117	29	19	8	2	0	0	1
30	222	142	80	42	11	1	0	1	1
10	82	93	107	159	49	7	1	0	2

Robust Hash Bit Persistence

If the robust hash is to be used in combination with a cryptographic hash as described in [1], it is important that not only the robust hash is truly robust and only few hash bits flip but also the positions of the hash bits within the robust hash bit vector remain the same: if a robust hash would always remain at a hamming distance of 8 from the reference image no mater what happens to the image (in the range of what is acceptable for a robust hash) but the positions of these 8 bits change randomly for each modification of the original images like in a cryptographic hash, combining cryptographic and robust hashes would be impossible.

Therefore we analyze the flipping positions of the individual images. Image 3 shows the robust hashes as a black and white pattern for 100 JPEG quality factors. One can see that while image 215 features more changes, the positions where the hash bits change is rather persistent for both images. Image 1 behaves as one would expect: with decreasing JPEG quality more and more positions flip, and also the overall number of flipped hash bits increases. Image 215 seems to react more strongly on some quality factors than to others. Also the number of flipped bits is much higher than image 1, which was already learned from the average hamming distances.

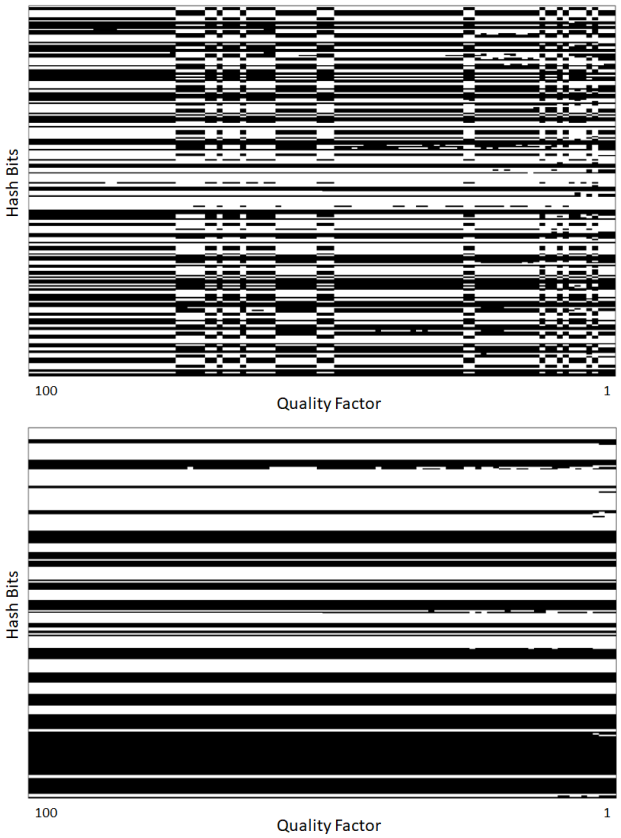


Figure 3: Top: Image 215 Robust Hashes for JPEG compression. Changes at multiple hash bit positions (y axis) with decreasing quality factor (x axis) can be observed. Bottom: Image 1 Robust Hashes for JPEG compression. Compared to image 215, much fewer hash bit positions flip during quality factor reduction.

Hamming Distance and Cryptographic Hash Image Detection

Until now we only discussed robustness without the need for actually choosing bit to be neutralized as in equation 4. We now have a look at the performance of such an approach for 4, 8 and 16 neutralized robust hash bits. Figure 4 shows how many images could not be detected as the number of flipped bit positions was higher than the number of positions neutralized. With only 4 allowed changes, at quality factor 64 already 10% of the images were not recognized. With 8 changes this level is only reached at quality factor 26 and with 16 changes at quality factor 8.

Again a more detailed look is provided by table 4. The column with the average changed positions shows that the average number of changed positions increases with falling quality and is still below 2 at JPEG quality 70. This means that on the average, only 2 bit positions of the robust hash of an image did change for all passes from quality 100 to 70. One must note the difference between hamming distance and changed positions: the latter can be seen as an OR operation on the hamming distance vectors of all quality steps for one image. Therefore the number of changed positions will always be equal or greater than the hamming distance for a given number of quality factors.

Table 4: Position changes in robust hashes and caused detection fails.

Quality Factor	Av. Changed Pos.	Detection Fails		
		hd4	hd8	hd16
100	0.152	4	2	1
90	0.376	8	4	2
80	0.628	13	5	2
70	1.908	30	11	6
60	2.92	69	21	8
50	3.298	86	25	8
40	3.754	112	29	8
30	4.398	147	38	8
20	5.878	211	64	18
10	8.99	322	150	36
1	23.546	474	407	250

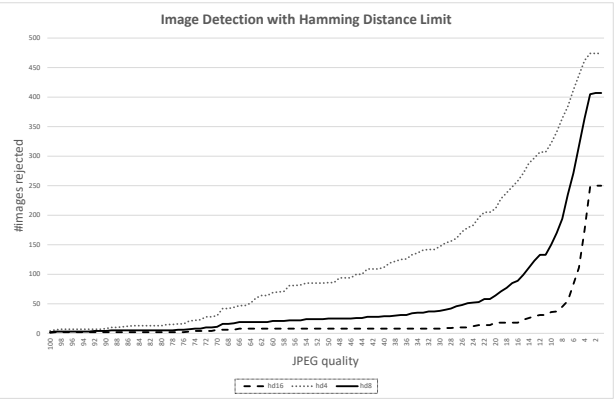


Figure 4: Simulation of 4, 8 and 16 neutralized hash bits.

Distance from Median

As the choice of neutralized hash bits is based on the distance from median as shown in equation 4, we now analyze the relationship between a robust hash bit's likelihood to flip and its distance from median. First we look at image 215 again. The tables 6 and 5 show a 16x16 matrix with the occurrences of bit flips for each hash bit (with a theoretical maximum of 100 as there were 100 quality factors) and its distance from median.

Table 5: Position changes for image 215

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	54	54	54	54	54	54	0	54	54	0	1	0	54	0	0
2	7	54	54	5	54	54	0	0	0	3	54	1	54	54	0	0
3	3	47	41	52	49	54	0	0	51	0	54	54	54	52	0	0
4	55	1	51	0	50	0	0	54	54	0	56	54	54	0	0	0
5	55	59	1	0	53	0	4	54	54	54	54	54	54	0	0	0
6	55	0	0	0	0	0	54	54	54	0	1	54	56	0	0	0
7	35	54	34	1	0	0	51	54	0	0	55	0	2	0	0	0
8	0	7	4	1	0	0	0	0	0	0	59	0	1	0	0	0
9	0	1	5	2	0	0	0	0	0	0	25	0	0	0	0	0
10	53	55	60	0	0	0	54	54	0	0	50	0	5	0	0	0
11	54	1	0	0	0	0	54	54	54	0	0	54	54	0	0	0
12	54	49	0	0	54	0	2	54	54	54	54	54	54	0	0	0
13	54	4	56	0	54	0	0	54	54	1	53	54	54	0	0	0
14	14	55	57	51	54	54	0	0	54	0	54	54	54	56	0	0
15	15	54	54	3	54	54	0	0	0	0	54	3	54	54	0	0
16	2	50	54	54	54	54	54	0	54	54	0	0	0	54	0	0

Table 6: Average distance from median for image 215

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	12	14	31	17	112	100	9	110	1	5	99	2	6	16	18	25
2	9	34	41	31	51	22	22	53	50	3	59	13	1	13	21	28
3	9	34	39	10	24	34	49	19	50	11	64	59	3	16	23	25
4	19	26	31	26	33	127	29	23	40	10	10	101	0	8	13	25
5	19	9	4	26	9	134	83	68	62	25	8	106	3	12	21	23
6	19	4	9	26	9	110	24	39	47	35	15	106	1	8	21	23
7	9	0	2	19	44	88	4	17	50	40	5	143	8	2	21	23
8	2	1	7	19	36	31	9	22	18	30	5	116	13	3	21	21
9	2	0	0	4	107	38	29	14	79	64	27	59	77	1	21	18
10	2	7	2	9	144	53	12	17	35	121	116	87	126	2	13	18
11	26	7	7	7	90	100	22	24	0	37	77	28	135	2	9	18
12	22	7	7	12	4	129	2	24	21	18	37	35	108	1	16	23
13	7	0	0	9	4	88	169	29	18	28	35	35	91	5	11	21
14	26	0	4	0	7	12	255	102	10	23	33	35	86	20	13	23
15	22	7	9	2	24	29	49	255	106	13	23	30	24	62	11	21
16	17	19	17	7	19	36	29	53	206	150	69	8	18	103	8	16

These data allows a detailed analysis of where changes were likely to occur. We address the individual cells by line and column, bit (14,7) is the bit from line 14 and column 7. Bit (14,3) flipped 57 times and has an average distance from the median of 0. Bit (15,8) never flipped and has the maximum distance of 255. This is the expected behavior: Weak bits flip, strong bits remain stable. But we also can spot contrasting behavior. Bit (16,9) flips 54 times but has a distance of 206 while bit (12,14) never flips but only has a distance of 1.

Figure 5 summarizes the results for all 500 images and JPEG quality factors, sorted by the average distance from median for all hash bits. The black line illustrates this overall mean value. The gray line is the average distance of all flipped bits of the individual images. Images with a small average distance from the median tend to behave more randomly with respect to the distance of the flipped hash bits. The average distance from the median for all bits is 55, for the flipped bits it is 48.

Prediction Potential

The previous section discussed various results regarding the behavior of the robust hash algorithm with respect to robustness and persistence. In this section we evaluate how well this information can be translated into a better prediction of which robust hash bits to neutralize in equation 4.

Predication by Distance from Median

The distance from median has been mentioned multiple times in the previous sections. In some cases it seems to have potential for correctly predicting which hash bits will flip. The alternative distance calculation in [8] uses the distance of median

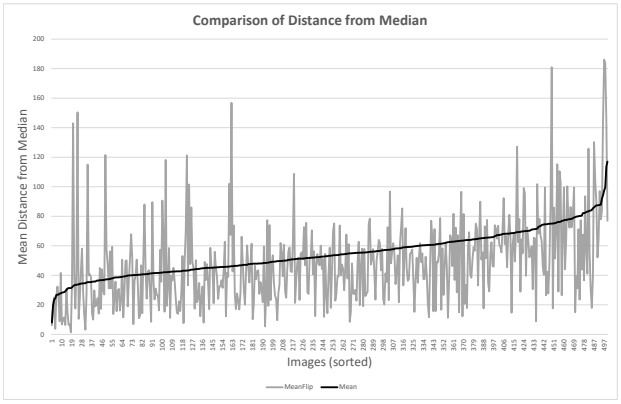


Figure 5: Distance from median

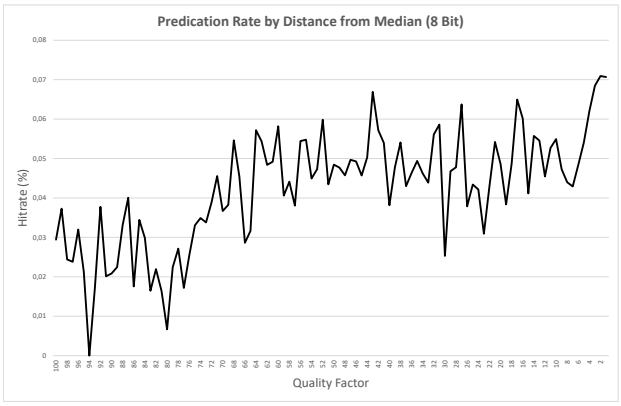


Figure 6: Prediction success rate by distance from median

of as a tool to improve error rates.

To analyze how well the distance from median correlates with flipping hash bits, we look at the gathered test data and sort the individual robust hashes by their increasing distance form median while remembering their hash bit numbers and whether the hash bit flipped.

Now if the assumption is true that a small distance from median results in a high likelihood that a hash bit will flip, a significant correlation between flipped bits and starting positions in the sorted list should occur. Figure 6 shows the results of this analysis for a 8-bit window and decreasing JPEG quality factors for the 500 test images. The "hitrate" shows how many percent of the flipped bits were among the first 8 positions in the sorted list. The numbers are small and increase with falling quality factor. This can be expected as the overall number of flipped bits increases and with it the likelihood that flipped bits are among the first 8 positions in the sorted list.

Prediction by Attack Example

As an alternative to the prediction by distance from median we have a look at the persistence of the flipping positions from one attack to another. In other words: Can we learn from the robust hash behavior of an image when applying JPEG compression which bits are likely to also flip when size changes occur.

Table 7 illustrates this for image 215. There are 4 potential states in the cells representing the 256 hash positions. "0" means no bit flipping occurred for this position under JPEG compression

Table 7: Image 215 Hash bit changes caused by JPEG (J) and Size (S)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
1	S	J	J	J	J	J	J	0	J	J	0	J	0	J	0	0
2	J	J	J	J	J	J	0	0	0	J	J	J	J	J	0	0
3	J	J	J	J	J	J	0	0	J	0	J	J	J	J	0	0
4	J	J	J	0	J	0	0	J	J	S	JS	J	JS	0	0	0
5	J	JS	J	0	J	0	J	J	J	J	JS	JS	JS	0	0	0
6	J	0	S	S	0	0	J	J	J	0	J	J	J	0	0	0
7	J	J	JS	J	0	0	J	J	0	0	J	0	J	0	0	0
8	0	J	J	J	0	0	0	0	0	0	J	0	J	0	0	0
9	0	J	J	J	0	0	0	0	0	0	J	0	S	0	0	0
10	J	J	J	0	0	0	J	J	0	0	J	0	J	0	0	0
11	JS	J	S	0	0	0	J	J	J	0	S	J	J	0	0	0
12	JS	J	0	0	J	0	J	J	J	J	J	J	J	0	0	0
13	JS	J	J	0	J	0	0	J	J	J	J	J	JS	0	0	0
14	J	J	J	J	J	J	0	0	J	0	J	J	J	J	0	0
15	J	J	J	J	J	J	0	0	0	S	J	J	J	J	0	0
16	J	J	J	J	J	J	J	0	J	J	0	0	0	J	0	0

or size change, "J" means flipping occurred only with JPEG, "S" only with size change and "JS" means flipping occurred both with JPEG and size changes. Ideally for prediction "0" or "JS" would dominate the results, but we find 107 "0", 130 "J", 8 "S" and 11 "JS" cells. In this case more than 50% of the flipped positions for size changes could be predicted by looking at the JPEG behavior. But only 11 of the 141 position changes caused by JPEG actually become position changes also caused by size changes.

When looking at all 500 images and their hashes at the different attacks, 22% of the positions of flipped hash bits caused by size changes are also caused by JPEG compression and 14% vice versa.

We also analyzed the confusion values for using JPEG hash bit flipping as an predictor for flipping caused by scaling. For this, we used a threshold which controlled which bits of the robust hash vectors were used as predictors. As shown in equation 5 we find the maximum number $max(hflips)$ of occurred flips for quality factors 100 to 1 for an individual image in the hash vector $hflips$. Then we multiply this value with the threshold and predict all $hflips_i$ as potential flipping positions that are greater than the result. Thereby we only take into account the positions which flipped most often during JPEG compression.

The resulting vector of i "P" and "N" values is compared to the hash bit flipping vector of the same image under resizing. If a "P" at position i points to a value > 0, a TP is generated, a "P" at a 0 is a FP, a "N" at a 0 is a TN and finally a "N" at a value > 0 is FN.

Table 8 shows the results for thresholds between 0.1 and 0.9. The True Negative results are by far most common due to the robustness of the robust hash: at many positions both JPEG and size changes do not cause bits to flip. The threshold 0.1 shows a TP of 4%, a FP of 1% and a FN of 10%. With increasing threshold the FN rises and the TP falls.

$$hashbittoken_i = \begin{cases} N & for hflips_i \leq max(hflips) * threshold \\ P & for hflips_i > max(hflips) * threshold \end{cases} \quad (5)$$

Summary and Discussion

In this work, we executed a deep analysis of the behavior of the block hash algorithm with respect to stability of individual

Table 8: Confusion values for various thresholds

Threshold	TP	TN	FP	FN
0.1	0.04	0.85	0.01	0.10
0.25	0.02	0.86	0.00	0.12
0.5	0.01	0.86	0.00	0.13
0.75	0.01	0.86	0.00	0.13
0.9	0.00	0.86	0.00	0.14

hash bits. This is different to known research were the perspective is more coarse and only the hamming distance of the robust hashes is discussed. This means, only the sum of flipped hash bits are addressed for various attacks, but their positions are ignored. Our aim is to increase understanding of the robust hash behavior to improve the combination of robust and cryptographic hashes to enable privacy-preserving robust hashing.

Prediction of the hash bits that are most likely to flip is vital if they are to be neutralized as described in the background section. Choosing the positions randomly will not succeed as the number of potential positions to choose from is high (256 for the standard block hash). Even with a hamming distance median of only 1 as shown to be typical for JPEG in our observations, it is only a chance of 1 in 256. The chances decrease dramatically for higher hamming distances like after size changes.

Our observations show a hash bit behavior that is promising for predictions: The robustness of the block hash is high and the positions in the individual hashes which are likely to flip during the attacks are stable. For one image, its hash is more likely to change at the same bit position it once changed during one attack type than at another arbitrary position. This can be used to provide good predictors for individual attacks: For e.g. JPEG compression, we can identify hash bit positions most likely to flip by applying a low quality factor. When encoding that image with JPEG quality factor 70 and comparing its resulting hash to the original one featuring 8 neutralized bit positions identified before, 489 of 500 images would have identical hash bits for reference and test images at the remaining 248 positions.

Inter-attack strategies are less likely to succeed. In our section on predictions we show only few positions flip both due to JPEG and size changes. The overall results still show that the likelihood to flip for a bit is still higher when it flips after one attack to also flip in the other, true positive predictions are significantly higher than false positives.

We also analyze using the distance from the media as a predictor for hash bit flipping. Compared to random guessing selecting neutral bits bit their distance to the median is significantly more likely to succeed. Still, for 8 neutral bits the chances for correctly representing changed positions has only a chance of a few percent. This is still not sufficient for a reliable prediction and a satisfying combination of robust and cryptographic hashes.

To summarize: we see a detailed analysis for the robust hash bit behavior as vital for designing privacy-preserving protocols for image identification. Prediction by distance from median or by learning from attacks is of no sufficient precision so far. A deeper analysis why certain hash bits of an image are more likely to lip during attacks is therefore recommended as future work. Distribution of areas, image contrast and dynamics and position of edges could be good starting points for investigation.

Acknowledgment

This research work has been funded by BMBF and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

[1] Martin Steinebach, Sebastian Lutz, and Huajian Liu. Privacy and robust hashes. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–8, 2019.

[2] Uwe Breidenbach, Martin Steinebach, and Huajian Liu. Privacy-enhanced robust image hashing with bloom filters. In Melanie Volkamer and Christian Wressnegger, editors, *ARES 2020: The 15th International Conference on Availability, Reliability and Security, Virtual Event, Ireland, August 25-28, 2020*, pages 56:1–56:10. ACM, 2020.

[3] Cedric De Roover, Christophe De Vleeschouwer, Frédéric Lefebvre, and Benoit Macq. Robust video hashing based on radial projections of key frames. *IEEE Transactions on Signal processing*, 53(10):4020–4037, 2005.

[4] Jiri Fridrich and Miroslav Goljan. Robust hash functions for digital watermarking. In *Proceedings International Conference on Information Technology: Coding and Computing (Cat. No. PR00540)*, pages 178–183. IEEE, 2000.

[5] Adil Haouzia and Rita Noumeir. Methods for image authentication: a survey. *Multimedia tools and applications*, 39(1):1–46, 2008.

[6] Arambam Neelima and Kh Manglem Singh. A short survey on perceptual hash function. *ADBU Journal of Engineering technology*, 1, 2014.

[7] Bian Yang, Fan Gu, and Xiamu Niu. Block mean value based image perceptual hashing. In *2006 International Conference on Intelligent Information Hiding and Multimedia*, pages 167–172. IEEE, 2006.

[8] Martin Steinebach, Huajian Liu, and York Yannikos. Forbild: Efficient robust image hashing. In *Media Watermarking, Security, and Forensics 2012*, volume 8303, page 83030O. International Society for Optics and Photonics, 2012.

Author Biography

Prof. Dr. Martin Steinebach is the manager of the Media Security and IT Forensics division at Fraunhofer SIT. In 2003 he received his PhD at the Technical University of Darmstadt for this work on digital audio watermarking. In 2016 he became honorary professor at the TU Darmstadt.

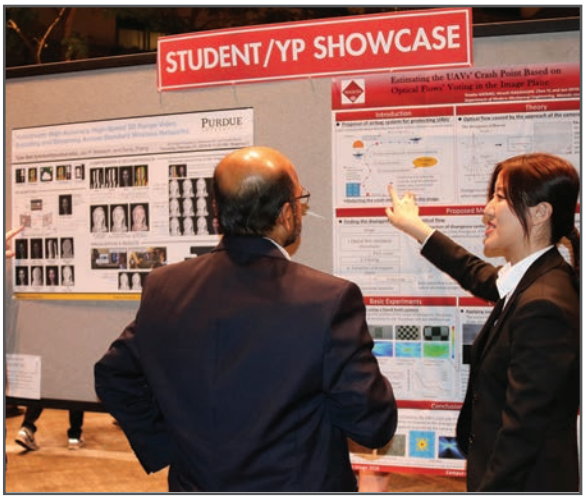
JOIN US AT THE NEXT EI!

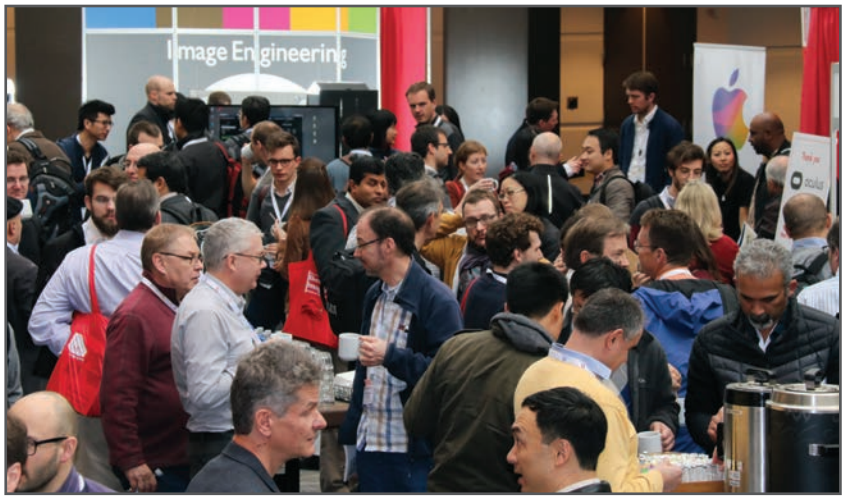
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org