New Attacks on Sari Image Authentication System*

Jinhai Wu[1], *Bin B. Zhu*[2], *Shipeng Li*[2], *Fuzong Lin*[1]

[1]State Key Lab of Intelligent Technology and Systems, Beijing, 100084, P. R. China
[2]Microsoft Research Asia, Beijing, 100080, P. R. China

ABSTRACT

The image authentication system SARI proposed by Lin and Chang passes JPEG compression and rejects other malicious manipulations. Some vulnerabilities of the system have been reported recently. In this paper, we propose two new attacks that can compromise the SARI system. The first attack is called a histogram attack which modifies DCT coefficients yet maintains the same relationship between any two DCT coefficients and the same mean values of DCT coefficients. Such a modified image can pass the SARI authentication system. The second attack is an oracle attack which uses an oracle to efficiently find the secret pairs used by SARI in its signature generation. A single image plus an oracle is needed to launch the oracle attack. Fixes to thwart the proposed attacks are also proposed in this paper.

Keywords: image authentication, digital signature, SARI, histogram attack, oracle attack, integrity verification

1. INTRODUCTION

Modern image processing tools and widely available powerful computers have made image manipulations an easy task. Checking an image's authenticity becomes more and more important. Image authentication has been actively studied in recent years. One of key differences between image authentication and the classical authentication well studied in cryptography is that some modifications such as near-transparent compression do not produce visible distortion and should be permitted in image authentication. Image authentication provides integrity verification for images by detecting malicious manipulations while passing incidental operations. Classification of these two types of manipulations depends on applications. An admissible manipulation for one application can be a malicious one for another. Comprehensive reviews on image and multimedia authentications can be found in [1][2].

One of widely studied image authentication schemes is the Self-Authentication and Recovery Image (SARI) system proposed by Lin and Chang [3][4]. The SARI system exploits the fact that the same quantization table is applied to all the Discrete Cosine Transform (DCT) blocks of an image in the Joint Photographic Experts Group (JPEG) compression, and is able to pass JPEG compression yet reject other manipulations. More specifically, SARI uses a key-based secret mapping function W to partition the set P of all DCT blocks of an image I to be authenticated into two, typically non-overlapping as shown next, sets $P_p = \{p_1, p_2, ..., p_{N/2}\}$ and $P_q = \{q_1, q_2, ..., q_{N/2}\}$ such that $P_q = W(P_p)$, $P_p \cap P_q = \phi$, and $P_p \cup P_q = P$, where N is the number of blocks in P. For each pair of blocks, say p and q, a number of frequency bins are selected to produce feature bits used to authenticate the image. The difference for each selected pair, for example, $\Delta F_{p,q}(v)$ for the block pair p and q at the DCT frequency bin v, is compared against a threshold k to generate a feature bit. If $\Delta F_{p,q}(v) < k$, the feature bit is set to 0, otherwise 1. A set of different threshold values can be used. The generated feature bits are encrypted by the secret private key of an asymmetric encryption to form a digital signature for the image. SARI also records in the signature the mean value of DCT coefficients in each selected frequency bin for all blocks. The signature can be either appended to the image as a tag [3][4] or embedded into the image by a quantization based semi-fragile watermarking scheme [5][6]. This paper will mainly focus on the former case, but the proposed attacks and fixes can be easily extended to the latter case.

---

*This work was done when Jinhai Wu was an intern at Microsoft Research Asia.

To check authenticity, a challenged image undergoes the same steps as above to obtain the difference $\Delta\hat{F}_{p,q}(v)$ for each selected pair of DCT coefficients and compare with the quantized threshold $\hat{k}$ which is related to the original threshold k as the following:

$$\hat{k} = \begin{cases} \tilde{k}_v \cdot Q(v), & \text{if } \dfrac{k}{Q(v)} = \text{integer} \\ \{\tilde{k}_v + (-1)^{Z_n(v)}\} \cdot Q(v), & \text{otherwise} \end{cases}$$

where $Q(v)$ is the JPEG quantization step for the frequency bin v under discussion, $\tilde{k}_v = \textit{integer round}\{k / Q(v)\}$, and $Z_n(v)$ is the corresponding feature bit decrypted from the image's signature. If $Z_n(v) = 0$ but $\Delta\hat{F}_{p,q}(v) - \hat{k} > \tau$, or if $Z_n(v) = 1$ but $\Delta\hat{F}_{p,q}(v) - \hat{k} < -\tau$, SARI concludes that either DCT coefficient or both DCT coefficients have been maliciously manipulated. A small tolerance bound τ is introduced in the comparison to avoid small difference caused by rounding pixel values to integers, different DCT implementation, etc. If all results agree with their feature bits and the calculated means are close enough to the recorded means in the signature, the image is claimed to be authentic; otherwise manipulated. We note here that the JPEG quantization table is needed for image authentication unless the aforementioned threshold $k = 0$. This may put a strong limitation on the use of non-zero thresholds in some applications. For example, the feature codes generated by comparing with non-zero thresholds cannot be used if the image to be authenticated changes to a non-JPEG format. In addition, if an image undergoes JPEG compression twice and if the first JPEG compression has larger quantization steps, SARI claims the image inauthentic. For watermarking based SARI described in [5][6], it is typical to use only the threshold $k = 0$ since watermarking schemes have limited data embedding capacities.

Several vulnerabilities of the SARI system have been reported in the literature. Radhakrishnan and Memon [7][8] proposed an attack to find out the secret mapping function W if multiple images are available whose feature codes are known and generated with the same mapping function. They found out that the mapping function W could be deduced on average by roughly 28 images of 512 by 512 pixels. The attack is ineffective if each image uses different mapping function or the feature codes are encrypted such as proposed in [3][4] and kept away from attackers. Uehara and Safavi-Naini [9] proposed an oracle attack to use one authenticated image to find out the secret block pairs, i.e., the mapping function W, and then to use this information to produce visually undetectable manipulations to pass the SARI authentication system. Their scheme to find the unknown block q which forms a block pair with a known block p is described as follows:

Loop until the block q is found,
1. Choose a block r such that $r \neq p$,
2. Modify p and r by the same amount m,
3. Send the modified image to the oracle and observe its output.
4. If it is accepted, then $q = r$ is found.

This scheme, however, needs to know the thresholds $\{k_i\}$ used in generating the signature. Otherwise the parameter m in the second step above may be too small that two blocks of different pairs may pass the authentication test and the scheme incorrectly concludes that the two blocks are a pair used by SARI. If the thresholds $\{k_i\}$ are known, m has to satisfy the following condition to avoid the false alarm just mentioned: suppose k'_{low} and k'_{upper} are the lower and upper bounds of the possible values for a difference of DCT coefficients at the frequency bin under discussion, respectively, and suppose the possible values of the difference are partitioned by the thresholds into the n segments $k'_0 \equiv k'_{low} < k'_1 < \cdots < k'_{n-1} < k'_n \equiv k'_{upper}$, then the amount $m > \max\{k'_i - k'_{i-1}\} + \tau,\ i = 1, \cdots, n$. This condition guarantees that any two blocks, if they are not a pair, fail the authentication test in the third step above. For a small set of thresholds, m is very large. A large m means a large modification in the second step, and may result in out-of-range pixel values in the spatial domain, which in turn reduces the effective value of m. A large m may also change the mean of DCT coefficients at the frequency bin large enough that triggers the alarm of the SARI authentication even if the two blocks under test are actually a pair. Recall that SARI records the means of DCT coefficients at selected frequency bins

for all blocks in the digital signature. In conclusion, the above oracle attack may result in wrong pairs or may never find actual pairs.

In this paper, we propose two new attacks to the SARI authentication system. The first attack is called a histogram attack which works for the case when only the threshold $k = 0$ is used for image authentication. The attack modifies the histogram of DCT coefficients at a selected frequency bin yet maintains the mean value of the DCT coefficients of all blocks and the same relationships for any pair of blocks. A fix is to add the lower and upper bounds of DCT coefficients of selected frequency bins to the digital signature. The second attack is an oracle attack which uses the output of an oracle to find out the secret mapping function W. This attack does not need the knowledge of the thresholds used in generating the signature, and maintain the mean values of DCT coefficients within a very small range of the recorded mean values in the signature so it does not trigger the SARI system. The attack consists of two stages. The first stage finds the first pair of blocks, and the second stage finds the remaining pairs by using one or more pairs previously found to balance the modifications so the mean value of DCT coefficients remains in the small allowed range. Once all the secret pairs and the difference threshold for each pair are deduced in these two stages, it is possible to produce visually undetectable manipulations to change the the image content without triggering the SARI system. Unlike the attack proposed in [7][8], only one authenticated image is needed to launch the proposed attacks.

The rest of this paper is organized as follows. In Section 2, we first describe in detail our proposed histogram attack, and then propose a fix to thwart the attack. The proposed oracle attack is described in Section 3. We conclude the paper in Section 4.

2. HISTOGRAM ATTACK

As we described in the last section, when only the threshold $k = 0$ is used, the SARI signature records only the relationship for each pair of DCT coefficients. This relationship is invariant for the lossy JPEG compression and is therefore used in SARI to distinguish the JPEG compression from other manipulations. We may also carefully design a manipulation that distorts an image yet maintains the original relationships and the mean values of the selected DCT coefficients without knowing the actual pairs used in generating the image's signature. Given the limitations in using non-zero thresholds as mentioned in Section 1, the condition for the proposed histogram attack is not very strong. Most SARI authentication systems used in real applications should satisfy this condition.

The histogram attack described below is such a kind of attack. Without loss of generality, we shall describe the histogram attack for any frequency bin $v \in [1, \mathrm{L}, 64\}$ selected in the signature generation.

2.1. Histogram Attack

The basic idea in the proposed histogram attack is that when a histogram of DCT coefficients is stretched or shrunken, the relationship between any two DCT coefficients does not change. The first step for the histogram attack is to collect all the DCT coefficients at a selected frequency bin v and calculate the histogram of these DCT coefficients. For a JPEG compressed image, the histogram can be the JPEG quantized integers of the DCT coefficients since any change will be quantized by the same JPEG quantization step. It is clear that as long as a manipulation keeps the mean value and the relative positions in the histogram unchanged, it will pass the SARI system. There are many manipulations that satisfy these two requirements. For example, the following manipulation satisfies these requirements and therefore passes the SARI system: suppose that the maximum and minimum values of the quantized DCT coefficients are $V_{\max}$ and $V_{\min}$, and there are $N_{\max}$ and $N_{\min}$ DCT coefficients at these values, respectively. Let the least common multiple of $N_{\max}$ and $N_{\min}$ is N_{com}. Let us add $m \cdot N_{com} / N_{\max}$ to all the maximum quantized DCT values and subtract $m \cdot N_{com} / N_{\min}$ from all the minimum quantized DCT values, where m is an arbitrary positive integer, then the mean and the relationship of any pair do not change. The same procedure can be applied to other DCT values as long as there are no changes to the relative positions in the histogram.

2.2. Discussions

In the above description of the histogram attack, we have not considered the possibility that the modification in the frequency values may result in out-of-range pixel values in the spatial domain for the manipulated image, which can be used by SARI to reject the histogram attack. Care has to be taken to avoid such a problem in the histogram attack. One way to do it is to modify the image in an iterative way: modify DCT values as described above, then transform to the

spatial domain and find out which part has the largest out-of-range values and corresponding DCT values, modify these DCT values to reduce the effect and check in the spatial domain again. This procedure is repeated until it produces a satisfactory result. Fig. 1 shows the original image on the left and the manipulated image on the right produced with the aforementioned histogram attack and the iterative method. One DCT frequency bin is used in the attack.

Fig. 1: Left: original image Lena. Right: image manipulated by the histogram attack.

In most cases, it might be difficult to use the proposed histogram attack to manipulate an image to achieve a desired content modification such as adding or removing an object, or changing the semantic meaning of an image. But such an attack would generate doubts on the credibility of such an authentication system. If a user Alice sees an image authentication system passes an image that is obviously perceptually distorted image, she would not trust the system since she may not be able to tell whether an image verified by such a system is really authentic or not, if she has no access to the original.

2.3. A Fix to Thwart the Histogram Attack

The histogram of the DCT coefficients at a frequency bin is typically densely distributed in a range. In other words, the distribution of the histogram typically does not have an obvious gap. In this case, a fix to thwart the proposed histogram attack is to simply add the maximum and minimum DCT coefficients of each selected frequency bin to the digital signature for an image. This increases slightly the size of the signature. If the distribution of the histogram has obvious gaps, say a gap that separates the histogram into two densely populated aggregations, the maximum and minimum values for each aggregation need to be added to the digital signature. Once DCT values are tightly bounded in densely populated ranges, it is much more difficult to modify the histogram without changing any relationship of DCT coefficients and with the mean values of DCT values close enough to the mean values recorded in the digital signature. We conclude that such a simple modification can effectively thwart the proposed histogram attack.

3. ORACLE ATTACK

With the above modification, the SARI system may still suffer an oracle attack which is designed to find out the secret mapping function W, i.e., the secret block pairs, used in generating the signature of an image. This information can then be used to construct a fraudulent image with desired content modification. Unlike the histogram attack, the proposed oracle attack still works even if nonzero thresholds $\{k_i\}$ are used in the authentication process. In this oracle attack, we assume that an attacker Bob has access to an authenticated image and an oracle so he can test if a modified image passes the authentication system or not. We would also assume that the DCT blocks are partitioned into non-overlapping pairs. We note that the assumption of an oracle available is not very restrictive since even in applications where access to an oracle is controlled, say three trials per users for a certain period of time, the needed number of tests can be easily achieved with a cooperation of Internet users, forgery of different users, waiting for enough time, etc.

In the SARI system, if two DCT coefficients from two blocks in the same pair at a selected frequency bin are modified by the same amount, the corresponding feature bit does not change. If the DCT coefficients are from two blocks in different pairs, feature codes may be changed in the process. This feature is exploited by the oracle attack proposed by Uehara and Safavi-Naini [9], denoted as the USN attack in the following, as well as our proposed oracle attack to find the secret pairs used in the SARI system. In the USN attack, a pair is claimed to be found if two DCT coefficients from two

blocks at the same frequency bin are modified by the same amount m and pass the oracle test. As we pointed out in Section 1, this scheme requires the knowledge the thresholds used in the signature generation, otherwise it may result in a wrong pair of blocks. Even with such knowledge, the USN attack may not be able to find right pairs due to out-of-range pixel values and its ignorance of the mean values of DCT coefficients of selected frequency bins for all blocks recorded in the digital signature in SARI. Our proposed oracle attack to be described in detail next, on the other hand, does not require the knowledge the thresholds used in generating the signature. Our attack is carefully designed to ensure that the mean values of DCT coefficients of selected frequency bins for all blocks do not vary much from the recoded values in the digital signature. In addition, our attack guarantees that all the secret pairs will be correctly deduced in a finite number of oracle tests. In the following subsections, we shall also analyze the required oracle tests in our oracle attack and propose a modified SARI that can thwart our oracle attack.

3.1. Our Oracle Attack

Without loss of generality, we use DCT values at a frequency bin v to launch the oracle attack. This frequency bin can be any of the frequency bins used in the SARI authentication system, such as the DC frequency or a low AC frequency. For convenience, we would assume that the image is in the JPEG format and all the DCT values mentioned below are the quantized DCT value. This means that adding 1 to a quantized DCT value is actually adding $Q(v)$ to the reconstructed DCT value, where $Q(v)$ is the JPEG quantization step for the DCT coefficient. It is equivalently to treat all $Q(v)$ as 1 in describing our oracle attack. This assumption does not actually put any restriction to our oracle attack since an image in a different format other than the JPEG format can be easily convert to the JPEG format which does not trigger the SARI system.

SARI records the mean value for each selected frequency bin, and small variations are allowed since the JPEG compression may cause some differences in these mean values. In the following description, we assume that a small variation of $2 \cdot Q(v) / N$ is allowed for the mean of the selected frequency bin v, where N is the total number of blocks. This means that for quantized DCT values, the sum of modified DCT values must be within the maximum distance of 2 from the original sum of the quantized values. In addition, we assume the frequency bins used in generating the signature is known but the thresholds are unknown. Since the allowed change to the mean value of DCT coefficients at the frequency bin v is so small, we have to make sure that modification in an oracle test does not cause modification to the mean value larger than the allowed value. To achieve this, our oracle attack consists of two stages: the first stage is to find the first pair of blocks. The second stage is to use the pairs previously found to balance the modification to testing blocks to ensure the sum of the mean value of DCT values after modification within the allowed range. In the process to find all the secret pairs, the allowed maximum distortion to a pair of DCT coefficients can also be found. The deduced information can be used to modify the content of the image without triggering the SARI system.

To simplify the description, we denote $Test(\bigcup_{i=1}^{n}(C_i + a_i))$ as the result of a procedure that adds a_i to the quantized DCT coefficient C_i for $i = 1, \ldots, n$, sends the resulting modified image to the oracle, and observes the output of the oracle, where n is the number of the coefficients to be modified simultaneously. The two stages of the proposed oracle attack can be described as follows:

Stage 1: Finding the first pair

1. For all the quantized DCT coefficients, find the coefficient A which has the largest value that can be added to it without resulting in out-of-range pixel values in spatial domain. Do the same thing to find another coefficient B which has the largest value that can be subtracted from it. Note that A and B are from different blocks so they have no interference in carrying out these procedures.
2. Find the amount m ($m \geq 0$) such that $Test(\{A+m, B-m\})$ is authentic but $Test(\{A+m+1, B-m-1\})$ is inauthentic.
3. Let $r_1 = Test(\{A+m+1, B-m\})$ and $r_2 = Test(\{A+m, B-m-1\})$.

   If r_1 and r_2 are both authentic, then (A, B) is a pair and exit Stage 1.

   If r_1 is inauthentic and r_2 is authentic, loop Step 3.1 until the first pair is found or all DCT values are tested.

   3.1. For each untested quantized DCT coefficient C, if $Test(\{A+m+1, B-m-1, C+1\})$ is authentic, then

(A, C) is a pair and exit Stage 1.

If r_1 is authentic and r_2 is inauthentic, loop Step 3.2 until the first pair is found or all DCT values are tested.

3.2. For each untested quantized DCT coefficient D, if $Test(\{A+m+1, B-m-1, D-1\})$ is authentic, then (B, D) is a pair and exit Stage 1.

If r_1 and r_2 are both inauthentic, let $r_3 = Test(\{A+m+1, B-m+1\})$. If r_3 is authentic, (A, B) is a pair and exit Stage 1; otherwise, loop Step 3.3 until the first pair is found or all DCT values are tested.

3.3. For each untested quantized DCT coefficient C, if $Test(\{A+m+1, B-m, C+1\})$ is authentic, then (A, C) is a pair and exit Stage 1.

Stage 2: Finding the remaining pairs

Loop the following steps until all the block pairs are found.

1. For all the remaining unpaired blocks, find the coefficient E which has the largest value that can be added to it without resulting in out-of-range pixel values in spatial domain.
2. Find the maximum integer *m* ($m \geq 0$) to be added to E without triggering the SARI system. In this process, some pairs found earlier are used as balancing pairs to ensure the sum of resulting DCT coefficients is within the distance 1 of the original sum. The two DCT coefficients in a balancing pair are modified by the same amount.
3. Increase E by *m+1* and decrease each coefficient of a balancing pairs so the sum of modified DCT coefficients is within the distance 1 of the original sum.
4. Loop Step 5 until the partner of E is found or all untested DCT values are tested.
5. Select an untested DCT value F, and increase F by 1. Send the modified image to an oracle and observe the output. If the output is authentic, then (E, F) is a pair.

In both stages, a search for the maximum amount *m* ($m \geq 0$) that can be modified without triggering the SARI system is needed. This can be efficiently done with a binary search. For example, for Step 2 in the second stage, the following binary search can be used:

1. Let $m_p = 0$, and m_f = MAXINT+1, where MAXINT is the maximum integer that can be added to the DCT coefficient E under discussion.
2. while $m_p < m_f - 1$ {
   Let $e = \lfloor (m_p + m_f)/2 \rfloor$, where $\lfloor x \rfloor$ rounds x towards 0. Add e to E, send the modified image to an oracle and observe the output.
   If the output is authentic, then set $m_p = e$; otherwise set $m_f = e$.
   }
3. m_p is the maximum amount to be searched for, which can be added to E without triggering the SARI system.

It is possible that MAXINT is too small that the above binary search does not give the maximum amount we search for. For example, when MAXINT is added to E and the output of the oracle is authentic. In this case, we can reverse the direction, i.e., change addition to subtraction and vice versa, and search again. If the maximum amount still can not be found, frequency bins not used in generating the signature can be modified to enlarge the search range for the coefficient. Another solution is to choose another DCT coefficient to continue the test. The same method can be applied in other steps in both stages 1 and 2 when this situation occurs.

When a pair of DCT coefficients is found in the above procedure, one bound of the difference of the two coefficients without triggering the SARI system is also given. If the other bound is needed in producing a fraudulent image, we can reverse the direction and use the binary search described above to efficiently find the other bound.

3.2. Discussions

In this subsection, we shall first estimate roughly the number of oracle tests needed to find all the secret pairs, and then compare with the actual number of oracle tests from our experiments. Two parts of oracle tests are used in the proposed

oracle attack. The first one is to find the maximum amount that can be changed without triggering the SARI system. The second is to find a proper pair. We shall ignore the problem of out-of-range pixel values in the spatial domain after modification in the DCT domain in estimating the number of oracle tests required in finding the secret pairs. If we ignore the small number of oracle tests in the first part in Stage 1, the number of oracle test in this stage is about $N/2$ on average, where N is the number of DCT blocks in the image. In Stage 2, the number of oracle tests for each DCT coefficient E in Step 2 is about $\log_2(MAXINT)$, where *MAXINT* is the maximum allowed modification for the DCT value under discussion. The number of oracle test to find its pair in the rest steps is $N'/2$ on average, where N' is the number of untested DCT coefficients. Therefore the total number of oracle tests is estimated roughly as $\frac{N^2}{8} + \frac{N}{2}(1 + \log_2(\overline{MAXINT}))$, where $\overline{MAXINT}$ is the average of the maximum amounts for each selected coefficient E. For an image of 128 by 128 pixels, $N = (128/8) \times (128/8) = 256$. *MAXINT* is the maximum value after quantization so the second term is much smaller than the first term for 128 by 128 pixel or larger images, and therefore can be ignored. We conclude the number of oracle tests to find all the secret pairs is roughly about $\frac{N^2}{8}$, which is 8192 for images of 128 by 128 pixels.

We have implemented the proposed oracle attack, and tested on two grayscale images Lena and Woman of 128 by 128 pixels with different keys. The actual numbers of oracle tests are reported in Table 1. From this table, we can see that the actual numbers of oracle tests from the experiments are close to our above estimated value.

Table 1: Number of oracle tests for 128 by 128 pixel images with different mapping functions W_i

	W_1	W_2	W_3	W_4
Lena	8432	8503	7995	9256
Woman	8514	8467	8527	8779

3.3. A Solution

Uehara and Safavi-Naini [9] proposed a solution to the oracle attack by making each pair has exactly one block in common with one more pair. This would increase the signature size. Here we propose an alternative solution which does not increase the size of the signature. We use two independent mapping functions W_1 and W_2 to generate two sets of the feature codes of the same size, in exactly the same way as the SARI system. These two feature codes are then XOR each other, and the result plays exactly the same role as the feature codes in SARI system. This fix achieves the similar effect as the fix proposed in [9] without increasing the feature code size. It also makes the attack proposed in [7][8] ineffective.

4. CONCLUSIONS

We have proposed two new attacks to the SARI system. The first attack is a histogram attack which modifies DCT coefficients yet maintains the same relationship between any two DCT coefficients and the same mean values of DCT coefficients. The attack applies when non-zero thresholds are not used in image authentication. The second proposed attack is an oracle attack which uses an oracle to find the secret pairs used by SARI in generating the digital signature for an image. Fixes to thwart both attacks are also proposed in this paper.

REFERENCES

[1] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When Seeing Isn't Believing," to appear in *IEEE Signal Processing*, March 2004.

[2] B. B. Zhu and M. D. Swanson, "Multimedia Authentication and Watermarking," *Multimedia Information Retrieval and Management*, D. Feng, W. C. Siu, and H. Zhang, Eds., Springer-Verlag, Berlin, Heidelberg, New York, 2003, chap. 7, pp. 148-177.

[3] C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Surviving JPEG Lossy Compression," *SPIE Storage and Retrieval of Image/Video Database*, San Jose, vol. 3312, no. 37, pp. 296-307, Jan 1998.
[4] C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation," *IEEE Trans Circuits and Systems of Video Tech*., vol. 11, no. 2, pp. 153-168, 2001.
[5] C.-Y. Lin and S.-F. Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content," *SPIE Security and Watermarking of Multimedia Contents II*, San Jose, CA, Jan. 2000, pp. 140-151.
[6] Q. Sun, S.-F. Chang, K. Maeno, and M. Suto, "A New Semi-fragile Image Authentication Framework Combining ECC and PKI Infrastructure," *IEEE Int. Circuits & Systems*, 2002, vol. 2, pp. 440-443.
[7] R. Radhakrishnan and N. Memon, "On the Security of the SARI Image Authentication System," *IEEE Int. Conf. Image Processing*, vol. 3, pp. 971-974, 2001.
[8] R. Radhakrishnan, N. Memon, "On the Security of the Digest Function in the SARI Image Authentication System," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 1030-1033, Nov. 2002.
[9] T. Uehara and R. Safavi-Naini, "On (In) security of 'A Robust Image Authentication Method'," *Proc. IEEE Pacific-Rim Conf. on Multimedia*, pp. 1025-1032, Dec. 2002.