SARI: Self-Authentication-and-Recovery Image Watermarking System

Ching-Yung Lin
IBM T. J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598
cylin@watson.ibm.com

Shih-Fu Chang
Dept. of EE, Columbia University
500 W120th St.
New York, NY 10027
sfchang@ee.columbia.edu

ABSTRACT

In this project, we designed a novel image authentication system based on our semi-fragile watermarking technique. The system, called SARI, can accept quantization-based lossy compression to a determined degree without any false alarm and can sensitively detect and locate malicious manipulations. It's the first system that has such capability in distinguishing malicious attacks from acceptable operations. Furthermore, the corrupted area can be approximately recovered by the information hidden in the image. The amount of information embedded in our SARI system has nearly reached the theoretical maximum zero-error information hiding capacity of digital images. The software prototype includes two parts - the watermark embedder that's freely distributed and the authenticator that can be deployed online as a third-party service or used in the recipient side.

Keywords

Authentication, watermarking, recovery, multimedia security, information hiding.

1. INTRODUCTION

SARI (Self-Authentication-and-Recovery Images, demos and test software at http://www.ctr.columbia.edu/sari) is a semi-fragile watermarking technique that gives "life" to digital images [1]. Like a gecko can recover its cut tail, a watermarked SARI image can detect malicious manipulations (*e.g.*, crop-and-replacement) and approximately recover the original content in the altered area. Another important feature of SARI is its compatibility to JPEG lossy compression within an acceptable quality range. SARI authenticator can sensitively detect malicious changes while accepting alteration introduced by JPEG lossy compression. The lowest acceptable JPEG quality factor depends on an adjustable watermarking strength controlled in the embedder. SARI images are secure because the embedded watermarks are dependent on the image content (and on their owner's private key).

Traditional digital signatures, which utilize cryptographic hashing and public key techniques, have been used to protect the authenticity of traditional data and documents [2]. However, such schemes protect every bit of the data and do not allow any manipulation or processing of the data, including acceptable ones such as lossy compression. To the best of our knowledge, the SARI technique is the only solution that can verify the authenticity of images/videos and at the same time accept desired manipulations such as JPEG compression and brightness adjustment. It also has the unique capability to sensitively detect unacceptable manipulations, correctly locate the manipulated positions and partially recover the corrupted area. This technique differs from traditional digital signatures in that (1) it uses invisible watermarking, which becomes an integral part of the image, rather than external signatures, (2) it allows some pre-defined acceptable manipulations, (3) it locates the manipulation areas, and (4) it can partly recover the corrupted areas in the image. A comparison of SARI and traditional digital signature method is shown in Table 1.

ACM Multimedia '01, Sep 30 – Oct. 5, 2001, Ottawa, Canada.

2. SYSTEM DESCRIPTION

SARI is based on the following techniques. Basically, two invariant properties of quantization-based lossy compression are the core techniques in SARI. The first property shows that if a transform-domain (such as DCT in JPEG) coefficient is modified to an integral multiple of a quantization step, which is larger than the steps used in later JPEG compressions, then this coefficient can be exactly reconstructed after later JPEG compression. The second one is the invariant relationships between two coefficients in a block pair before and after JPEG compression. In SARI, we use the second property to generate authentication signature, and use the first property to embed it as watermarks. These properties provide solutions to two major challenges in developing authentication watermarks (a.k.a., integrity watermarks): how to extract short, invariant, and robust information to substitute fragile hash function, and how to embed information that is guaranteed to survive quantization-based lossy compression to an acceptable extent. In additional to authentication signatures, we also embed the recovery bits for recovering approximate pixel values in corrupted areas. SARI authenticator utilizes the compressed bitstream, and thus avoids rounding errors in reconstructing transform domain coefficients.

The SARI system was implemented in the Java platform and is currently operational on-line. Users can download the embedder from the SARI website and use it to add the semi-fragile watermark into their images. He can then distribute or publish the watermarked SARI images. The counterpart of the embedder is the authenticator which can be used in the client side or deployed on a third-party site. Currently, we maintain the authenticator at the same website so that any user can check the

(a) (b)

Figure 1: (a) User Interface of the SARI embedder (b) an example of the watermarked SARI image (Size: 256x384, PSNR = 41.25 dB, Embedded semi-fragile info bits: 20727)

authenticity and/or recover original content by uploading the images they received.

Figure 1 shows the user interface of the embedder in which the user can open image files in various formats, adjust the acceptable compression level, embed the watermarks, check the quality of the watermarked images and save them to files in desired formats (compressed or uncompressed). The user interface of the authenticator includes the functions that open image files in various formats, automatically examine the existence of the SARI watermark, and authenticate and recover the manipulated areas.

3. EXAMPLE and SUMMARY

Figure 2 and 3 show an example of using SARI. In Figure 2, we first embed watermarks in the image, then use Photoshop 5.0 to manipulate it and save it as a JPEG file. Figure 3 shows the authentication result of such manipulations. We can clearly see that the manipulated areas can be located by the SARI authenticator. In Figure 3(b), we can see that the corrupted area has been recovered.

SARI has been extensively tested on-line and has shown its effectiveness in protecting the content-integrity of images. It can effectively reconstruct the trustworthiness of digital multimedia data that have been threatened by various state-of-the-art manipulation tools. Besides the security applications, SARI can be used for error concealment in a wireless environment [3].

4. REFERENCES

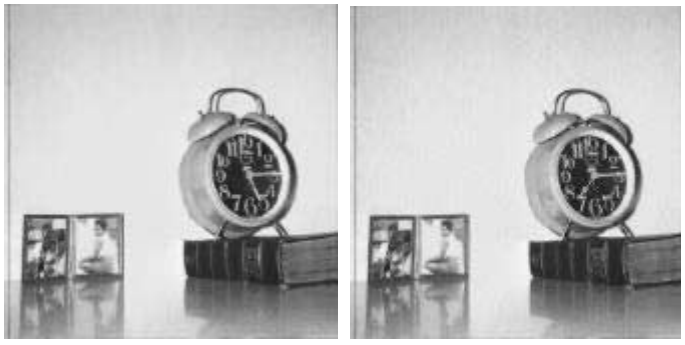
[1] C.-Y. Lin and S.-F. Chang. Semi-Fragile Watermarking for Authenticating JPEG Visual Content. *SPIE Security and Watermarking of Multimedia Content II*, Jan 2000.

[2] W. Diffle and M. E. Hellman. New Directions in Cryptography . *IEEE Trans. on Information Theory*, Vol. 22, No. 6, pp-644-654, Nov 1976.

[3] C.-Y. Lin, D. Sow, and S.-F. Chang. Using Self-Authentication-and-Recovery for Error Concealment in Wireless Environments. *Proceedings of SPIE*, Vol. 4518, Aug. 2001.

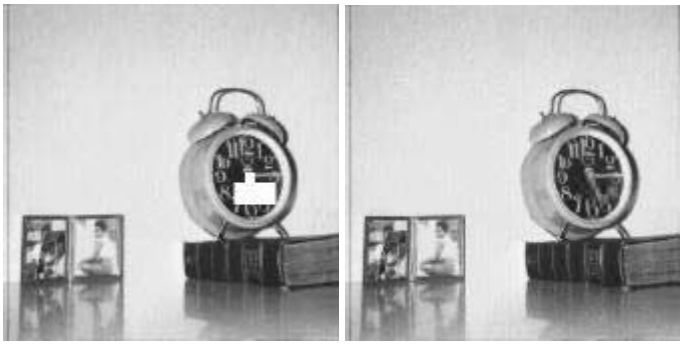
Table 1. Comparison of Digital Signature and SARI

	Digital Signature	SARI
Characteristic	Single-stage authentication	End-to-end, content-based authentication
Robustness	No single bit of the data can be changed	Accept various content-preserving manipulations
Sensitivity	Detect any change	Detect malicious changes, e.g., crop-and-replacement
Security	Use public key methods	Use secret mapping function and/or public key methods
Localization	Cannot localize manipulated areas.	Can localize the manipulated areas.
Convenience	Need a separate digital signature file.	No additional file is required.
Recovery	Not feasible.	Corrupted regions can be approx. recovered.
Visual Quality	Not affected	Not affected, but may degrad if require strong robustness



(a) (b)

Figure 2. (a) Original image after adding SARI watermark; (b) manipulated image by crop-and-replacement and JPEG lossy compression.



(a) (b)

Figure 3: (a) Authentication result of the image in Figure 2(b); (b) Authentication and Recovery result of the image in Figure 2(b).