A NEW SEMI-FRAGILE IMAGE AUTHENTICATION FRAMEWORK COMBINING ECC AND PKI INFRASTRUCTURES

Qibin Sun[1], *Shih-Fu Chang*[1], *Kurato Maeno*[2] *and Masayuki Suto*[2]

[1] Department of E.E., Columbia University, New York City, NY10027, USA
[2] Oki Electric Industry Co., Ltd. Gunma 370-8585, Japan

ABSTRACT

This paper presents a new semi-fragile framework aiming at extending public key signature scheme from message level to content level. The content signing procedure includes signature generation and watermark embedding while the content authentication procedure includes watermark extraction and signature verification. One main unique contribution is the novel use of error correction coding (ECC) to address the incidental distortions caused by some acceptable manipulations such as lossy compression. Another unique feature is integration of PKI security infrastructure and the hashing mechanism to achieve security and short signatures/watermarks. In the signing procedure, block-based invariant features are extracted from the image content and then encoded by ECC to obtain their corresponding parity check bits (PCB). All PCBs are then embedded back into image as watermarks for the purpose of authentication and locating content alteration. In addition, codewords are concatenated, hashed and finally signed by content owner's private key to generate a global cryptographic signature. The authentication procedure is the inverse procedure of signing except using content owner's public key for signature verification. After describing the proposed algorithm in details, an implementation example is given by combining our system with invariant features explored in earlier systems.

Keywords digital signature, watermark, authentication, integrity protection, PKI, error correction coding

1. INTRODUCTION

Semi-fragile mage authentication concerns with verifying authenticity of a received image while allowing some acceptable manipulations such as lossy compression. Typical approaches can be categorized as: signature based, watermark based, or combinations of both. Signatures typically are based on image content in order to represent the invariant "essence" of the image. To enhance security and reduce the signature length, it's desired to apply hashing and public key infrastructure (PKI).

Application of hashing and PKI to image authentication has been shown in the fragile watermarking system proposed by Wong and Memon[1]. In that system, the signatures are generated from hashing all relevant information including image itself by setting all LSBs to zero and user ID. Then the formed signatures are signed and embedded back into all image block LSBs as watermarks. However, such system is fragile- any change to the non-LSB part will modify the signature. For content-based image signatures and the corresponding watermarking techniques, the main challenge has been to find "adequate" content features that can be used to distinguish between malicious attacks and acceptable manipulations. In [2,3], semi-fragile authentication solutions were developed considering lossy compression as acceptable. In [2], the authors discovered a mathematical invariant relationship between two coefficients in a block pair before and after JPEG compression. In [3], the authors simply took the mean value of each block as the feature. Both of these systems have difficulties in integrating their techniques with hashing and PKI, as discussed below.

- Acceptable manipulations will cause changes to the content features, though the changes may be small compared to content-altering attacks. Such "allowable" changes to the content features make the features non-hashing. Any minor changes to the features may cause significant difference to the hashed value due to the cryptographic nature of the hashing method.
- As a result of the incompatibility with hashing, the generated signature size is proportional to the size of the content, which is usually very large. This will result in a time-consuming signing procedure as the size of the formed signature is much greater than 1024bits. The formed signature has to be broken into small pieces (less than 1024bits) for signing.
- Because of the possible variations caused by acceptable manipulations, decision of authenticity is usually based on comparison of the feature distance (between original one and the one extracted from the received image) against a threshold value, which is hard to determine.

Usually ECC is used for tolerating bit errors when transmitting messages in a noisy channel by adding the redundancy into original messages[5]. In this paper, we

propose a new semi-fragile image authentication framework by utilizing ECC in a novel way. Instead of directly embedding whole ECC encoded features into image, we only take their associated parity check bits as a kind of Message Authentication Code (MAC[4]) and embed them into image as watermarks for authentication purposes. Its error correction capability allows us construct a stable cryptographic hash value even facing feature variations caused by acceptable manipulations (such as lossy compression, single pass or multiple iterations). Note that similar idea on using ECC to tame acceptable distortions has been explored in other research fields such as biometrics[6].

In Section 2, we will describe with details our proposed framework. Our framework is general and can be used together with any invariant or almost invariant features extractable from images. Section 3 includes an implementation example using the invariant features that has been explored in a well-known semi-fragile authentication system[2,7]. Section 4 concludes the paper.

2. PROPOSED FRAMEWORK

In our proposed solution, signature generation / verification modules are mainly employed for the role of content signing and authentication. Watermark embedding / extraction modules are only used for storing signatures. Refer to Figure 1 (Upper part)., the procedure of content signing can be depicted as follows. Although we use DCT block-based structure for feature extraction and watermark embedding in the following explanation, it should be noted that the proposed method is general so that features from different representations (such as wavelet transform, JPEG-2000 etc) and non-block structures (such as wavelet subbands, the pixel domain, etc) can be used.

The input original image is firstly partitioned into non-overlapping blocks. Transform such as DCT is usually needed for each block. Block-based invariant features are then extracted and mapped onto binary values if the subsequent ECC scheme is binary. After ECC encoding, their corresponding parity check bits (PCBs) of each block-based feature set can be obtained. Taking PCBs as the seed of watermark to form the block-based watermark. One necessary condition in selecting watermarking scheme is that the embedded watermark should be robust enough for extraction from received images under acceptable manipulations. Therefore a simple method for generating watermark data is to use ECC again: PCBs are encoded by another ECC scheme and then the ECC encoded output is embedded as watermark data. The watermark data for each block is embedded either into the same block or into a different block. In the meantime, all codewords (features together with their corresponding PCBs) are concatenated and hashed by typical cryptographic hashing function such as MD5[4]. Finally, content owner's private key is used to sign the hashed value. The encrypted hashed value can be stored in a place external to the image as embedded into the image again as an watermark. The proposed method can be used with various invariant features, such as the object-based features used in [3,8], the invariant transform coefficient relationships [2.7], and the invariant fractionalized bit planes in JPEG-2000 images [9].

The content signing algorithm is further described using the following structured codes.

System setup:
Content owner requests a pair of keys (private key and public key) from the PKI authority.
Select an adequate ECC scheme (*N, K, D*) given domain-specific acceptable manipulations. Here *N* is the length of output encoded message, *K* is the length of original message and *D* is the error correction capability.
Select another ECC scheme (*N', K', D'*) for watermark formation as described above (optional)

Input:
Original image to be signed I_o

Begin
Partition image into non-overlapping blocks (1..*B*).
For block 1 **to** block *B*, **Do**
Conduct block-based transform such as DCT.
Extract invariant feature.
Map and fold (if necessary) each feature set into one or more binary messages each of which has length *K*.
ECC encode each binary vector to obtain its codeword *W* and PCBs. Their lengths are *N* and *N-K*.
i) The PCBs can be used as watermark or they can used to form watermark through another ECC (*N', K', D'*); where *K'* = *N* – *K*;
Embed watermark into selected block;
Inverse transform to obtain watermarked image I_w;
ii). Collect codewords from all blocks *W* (1..*B*) and concatenate them to form a single bit sequence Z
End
Hash the concatenated codeword sequence *Z* to obtain *H*(*Z*);
Sign on *H*(*Z*) by the owner's private key to obtain the Signature *S*;
End

Output:
Watermarked image *Iw*;
Content-based encrypted hashed signature *S*.

As described above, only the PCBs are embedded as watermarks and are used later in verification stage for correcting potential changes in the signatures (i.e., content features). As shown in the example ECC in Table 1, we can see that the first 4 bits in a codeword are from the original message bits. Assume we want to protect the message 0001, its corresponding codeword is 0001111. We only need to use the last 3 bits (PCBs) to form MAC and use it as watermark data. Later assume we receive a message like 0011 (one bit change compared to the original message 0001). By checking with its original parity check value: 111, we can detect and correct the code 0011 back to 0001 and obtain its correct corresponding codeword 0001111 again. It is clear that by using a simple

ECC scheme, the system robustness is improved. It's likely that minor changes caused by acceptable manipulations (e.g., lossy compression or codec implementation variation) can be corrected by the ECC method. However, the use of ECC also brings some security concerns. Since the mapping from messages to PCBS is a multi-to-one mapping, the reverse mapping is not unique. In the example shown in Table 1 one PCB is shared by two messages. It results in some security problems. For example, given the original message 0001, its PCB is 111. We can replace the original message with a faked one: 1111, its corresponding PCB also is not affected, still 111. Hence it will pass the authentication. This case will become worse in practice, as the length of message (i.e., extracted feature) usually is mush longer than that of parity check bits. In practical implementations, we can partly reduce such system security risk by employing some methods, such as making the MAC formation adaptive to the location of the image block or by randomizing MAC assignment. However, to add another layer of security, we augment the above PCB-based ECC watermark by using a cryptographically hashing of the concatenated codewords, not just the PCBs.

Let's re-check Table 1 again. We can see that, although given one PCB, there are multiple messages sharing the PCB. However, their corresponding codewords are different (0001111 and 1111111 respectively). In other words, each syndrome (message and PCB), is uniquely defined. Any change in the message or the PCBs will make the syndrome different. Given the uniquely defined concatenated codeword sequence, we can apply cryptographic hashing (e.g., MD5) to the codeword sequence and form a much shorter output (about a few hundreds of bits).

Refer to Figure 1 (lower part), to authenticate a received image content, in addition to the image itself, two other pieces of information are needed: the signature associated with the image (transmitted through external channels or as embedded watermarks), and the content owner's public key. The image is processed, in the same way as feature generation, decompose image into blocks, to extract features for each block, to form finite-length messages. From the embedded watermarks, we also extract the PCBs corresponding to messages of each block. Note the messages are computed from the received image content, while the PCBs are recovered from the watermarks that are generated and embedded at the source site. After we combine the messages and the corresponding PCBs to form codewords, the whole authentication decision could be made orderly. First, we calculate the syndrome block by block to see whether there exists any blocks whose codewords are uncorrectable. If yes, then we could claim that the image is unauthentic. Secondly, assume all codewords are correctable, we replace those erroneous codewords with their corrected ones. Then we repeat the same process at the source site: concatenate all corrected codewords into a global sequence and cryptographically hash the result sequence. By using owner's public key, the authenticator can decrypt the hashed value that's generated at the source site. The final authentication result is then concluded by bit-by-bit comparison between these two hashed sets: if there is any single bit different, the authenticator will report that the image unacceptable ("unauthentic").

It's interesting and important to understand the interplay between the decision based on the block-based signatures and the global hashed signature. The local signatures are the PCBs corresponding to the features of each block. They can be used to detect any unrecoverable changes in a block. However, since we do not transmit the entire codeword, there exist changes of a block that cannot be detected (as the case 0001111 vs. 1111111 discussed earlier). However such changes will be detected by the global hashed signature, because the hashed signature is generated by using the entire codewords, not just the PCBs. Therefore, there exist such possibilities: the image is deemed as unauthentic because of inconsistence between hashed sets while we are unable to indicate the locations of attacks because there are no uncorrectable codewords found. In such case, we still claim the image is unauthentic although we are not able to indicate the exact alternation locations.

3. AN IMPLEMENTATION EXAMPLE

As a proof-of-concept exercise, we describe how the proposed framework can be applied to a prior system, SARI [2,7], for semi-fragile image authentication. In SARI, two invariant properties are utilized for signature generation and watermark embedding respectively[2,7]. The first property, used in generating invariant features, is based on the invariant relationship between two coefficients in a block pair before and after JPEG compression. The second property, used for watermarking embedding, is if a coefficient is modified to an integral multiple of a quantization step which is larger than the steps used in later JPEG compression, this coefficient can be exactly reconstructed after later JPEG compression. We use the same properties for signature generation and watermarking in our system. The only difference is instead of directly embedding the feature sets, we apply ECC and embed PCBs as watermarks. In each 8x8 block, we take the first 10 coefficient pairs to generate the signature and embed watermark back into 11^{th} to 20^{th} coefficients. ECC for generating PCBs is BCH (15,11,1) where one bit error is allowed in a block. (The length of PCB is 4). The concatenated codewords are cryptographically hashed to form a global signature, which is embedded as watermark as well. ECC for watermarking is based on another BCH

(7,4,1). (Therefore the length of watermark in a block is 7). More detailed testing results will be published soon.

4. CONCLUSIONS AND FUTURE WORK

In this article, we have proposed a new semi-fragile image authentication watermarking framework combining ECC and PKI security infrastructure. By using ECC, we provide a mechanism allowing minor variations of content features caused by acceptable manipulations (such as lossy compression or watermarking). We also developed a novel approach combining local block-based signatures and a global signature. The former can be used to detect locations of attacks in specific blocks, while the latter uses cryptographic hashing and PKI ensure the global authenticity of the whole image. As a proof-of-concept example, we also described the procedure of converting a prior system, SARI, to utilize the proposed framework.

In a related work [9], we focused on semi-fragile authentication watermarking for JPEG-2000 images. We extracted features that's invariant against JPEG-2000 lossy compression, codec variations, and other acceptable manipulations. The proposed framework and test performance are found to be promising for the specific target application. Future work includes selecting and testing other invariant features as well as ECC schemes and extending the proposed framework to other media such as video and audio.

5. REFERENCES

[1] Ping Wah Wong and Nasir Memon, "Secret and public image watermarking schemes for image authentication and ownership verification", *IEEE Transactions on Image Processing*, Vol.10, No.10, pp.1593-1601, 2001.

[2] C.-Y. Lin and S.-F. Chang, "A robust image authentication method surviving JPEG lossy compression", *SPIE Security and Watermarking of Multimedia Content*, Vol.3312, pp.296-307, 1998.

[3] Der-Chyuan Lou and Jiang-Lung Liu, "Fault resilient and compression tolerant digital signature for image authentication", *IEEE Transactions on Consumer Electronics*, Vol.46, No.1, pp.31-39, 2000.

[4] B. Schneier, *Applied Cryptography*, New York: Wiley, 1996.

[5] S. Lin and D. J. Costello, JR., *Error control coding: Fundamentals and applications*, Prentice-Hall, 1983.

[6] G. I. Davida, Y. Frankel and B. J. Matt, On enabling secure applications through off-line biometric identification, *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, pp.148-157, 1998

[7] Ching-Yung Lin and Shih-Fu Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content," *SPIE Security and Watermarking of Multimedia Contents II EI '00*, SanJose, CA, Jan. 2000.

[8] M. Schneider and S.-F. Chang, "A Robust Content Based Digital Signature for Image Authentication", *IEEE International Conf. on Image Processing*, Laussane, Switzerland, Oct 1996

[9] Q.-B. Sun and Shih-Fu Chang, "Semi-Fragile Authentication of JPEG-2000 Images with a Bit Rate Control", Columbia University ADVENT Technical Report, 2002-101, also submitted to ICIP 2002.

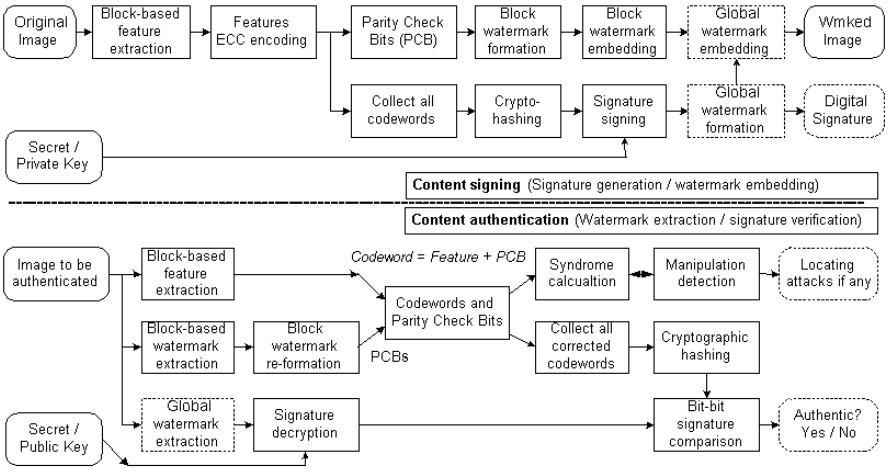

Figure 1. Proposed semi-fragile image authentication framework

Table 1. (7, 4) Hamming Codes with 1 bit error correction ability

Message	Codeword		Message	Codeword	
	Message	PCB		Message	PCB
0 0 0 0	0 0 0 0	0 0 0	1 0 0 0	1 0 0 0	0 1 1
0 0 0 1	***0 0 0 1***	1 1 1	1 0 0 1	1 0 0 1	1 0 0
0 0 1 0	0 0 1 0	1 1 0	1 0 1 0	1 0 1 0	1 0 1
0 0 1 1	0 0 1 1	0 0 1	1 0 1 1	1 0 1 1	0 1 0
0 1 0 0	0 1 0 0	1 0 1	1 1 0 0	1 1 0 0	1 1 0
0 1 0 1	0 1 0 1	0 1 0	1 1 0 1	1 1 0 1	0 0 1
0 1 1 0	0 1 1 0	0 1 1	1 1 1 0	1 1 1 0	0 0 0
0 1 1 1	0 1 1 1	1 0 0	1 1 1 1	***1 1 1 1***	1 1 1