Authenticating Encrypted Data

Sachin Kashyap
Electronics and Communication Engineering Dept.
Indian Institute of Technology Guwahati
Guwahati, Assam 781039
Email: k.sachin@iitg.ernet.in

Kannan Karthik
Electronics and Communication Engineering Dept.
Indian Institute of Technology Guwahati
Guwahati, Assam 781039
Email: k.karthik@iitg.ernet.in

***Abstract*—Traditionally message authentication codes (MACs) and the process of encryption are treated as orthogonal security mechanisms, where MACs are used to prove data authenticity while encryption is used to preserve confidentiality. In this paper we propose to integrate the two in a framework where the parameterized hash value of an encrypted image is designed to be the same as the hash value of the parent unencrypted original image. This form of privacy preserving authentication has applications in database searches containing sensitive files such as sets of patient's X-ray records. The main challenge here is to develop a parametric hashing algorithm that is invariant to encryption by allowing a small part of the statistical signature of the original image to emerge despite the encryption process. Since the hash value is computed without decrypting the original data, one can prove authenticity without actually revealing the information. Our contribution is twofold: (1) Formulation of this new problem of authenticating encrypted data and (2) Development of a simple hashing algorithm applicable to encrypted images.**

I. Introduction

With the unparalleled growth of the internet and the multimedia technology becoming ubiquitous more than ever, the security of digital images and videos have become more and more important. Multimedia data such as images, videos or audios can be easily copied by unauthorized users. They have become vulnerable to illicit signal processing operations. The security mechanisms which are employed to protect the multimedia data from unauthorized operations are (1) Multimedia encryption to prevent eavesdropping, (2) Watermarking for copyright protection and tracking and (3) Parametric multimedia hashing for content authentication.

Multimedia authentication is a relatively new research area which aims at ensuring that the particular data were sent by the authorized user and have not subsequently been altered or substituted. Message validation schemes such as watermarking and hashing have provided solutions for copyright protection and content authentication. Message authentication verifies the integrity of a message and that the purported identity of the sender is valid. Multimedia watermarking aims at embedding an invisible or visible information into the multimedia for the purpose of copyright protection, content authentication and forensic tracking [1]. Watermarking can be divided into two types: robust watermarking and fragile watermarking. Robust watermarks are resistant to various signal processing operations and they are mainly used for copyright protection. On the contrary, fragile watermark is altered if the multimedia is slightly manipulated and they are used for content validation [2], [3].

Multimedia hashing is another method employed for authenticating the messages. Hashing is essentially a many-to-one mapping which is used to authenticate the message and to ensure that it came from the true source. They are also called as one-way functions because given a message M and a hash function H, it is very easy to compute the hash $h = H(M)$ but given the hash h, it is impossible to compute M such that $h = H(M)$. The robustness requirement of the hash against manipulations is different for content authentication and copyright protection applications as it is for watermarks. The former requires the hash to be fragile against even minute manipulations while the later requires it to be robust even against major manipulations.

Multimedia encryption aims at rendering the information unintelligible under the influence of a key. Encryption aims at removing the redundancy and the correlation of the data by scrambling it. In order to protect data privacy, images and videos need to be encrypted before being transmitted to the destination. Because the multimedia data such as videos form large streams of data, partial encryption that encrypts only some significant parts of the data is employed [4]. Multimedia has some special requirements like perceptual security, efficiency, cryptographic security, format compliance and signal processing in the encrypted domain etc [5].

In order to boost the data security, the researchers are focussing on the marriage of the two levels of security [5], [6], [7]. Multimedia encryption and multimedia authentication schemes serve two different purposes but they can be merged together in one system to protect confidentiality and to check the authenticity of the data. It is indeed a very challenging problem but if we can integrate the two functionalities simultaneously, it will revolutionize the area of multimedia distribution. In [5], a commutative watermarking and encryption has been proposed that embeds watermark into the encrypted media directly, which avoids the decryption-watermarking-encryption triples. The encryption and watermarking operations are done on the same data part. In [6], commutative watermarking and encryption scheme is proposed for media data protection. In the scheme, the partial encryption algorithm is adopted to encrypt the significant part of media data, while some other part is watermarked. In [5], it has been reported that commutative schemes based on partitioning the

data are vulnerable to replacement attacks. Since one data part is not encrypted, it leads to leakage of information and it is vulnerable to watermark attacks. In [7], the possibilities of inserting and detecting of watermark in the encrypted domain have been discussed. The various issues pertaining to such a scenario and its potential applications have been discussed. In [7], a watermark detection scheme has been proposed which detects the watermark irrespective of whether it has been embedded in the plaintext and then the watermarked data has been encrypted or first it has been encrypted and then watermarked. Their scheme achieves the challenging goal of achieving the two levels of security simultaneously.

The paper is organized as follows: Section II formulates the problem of authenticating encrypted data, talks about the invariant watermark and invariant hashes and discusses about the zero-knowledge authentication. Section III presents the chaos based encryption that we have employed on the data points to scramble the information. We then propose and analyze the novel method for calculating the hash in section IV . Section V summarizes the experimental simulations and carries out the discussion on the results.

II. Problem Formulation

Suppose I have a piece of information. Without revealing the information, I want to prove to you that I have it. The way I do it should not in any way leak information about my actual data. Here comes the problem of authentication of encrypted data. This is termed as zero knowledge proof. Message authentication assures that data received are exactly as sent by and that the purported identity of the sender is valid. So, we have to look for a watermarking-encryption dual in which the watermark remains invariant to the encryption process. Let the original media be P, the encryption process be represented as E , the watermark embedding algorithm be represented as W_{embed}, watermark extraction algorithm be represented as $W_{extract}$, the watermark be W, the watermark key be K_w, the encryption key be K, watermarked media be P_W then mathematically,

$$E(W_{embed}(P, W, K_w), K) \;=\; E(P_w, K) \;=\; P_{w,encrypt} \tag{1}$$

We want the watermark to remain invariant to the encryption process. That is to say, we want a scheme wherein we can extract the watermark without decrypting the received data. Mathematically,

$$W_{extract}(P_{w,encrypt}, K_w) \;= W \tag{2}$$

If such a scheme is achieved, it will be possible to extract the watermark directly from the encrypted media. The watermark can be embedded directly in the encrypted domain. Zero knowledge proof will be achieved. In order to extract a watermark and embed a new one, one will not have to go through the decryption-watermark embedding-encryption triples. In [7], watermarking detection algorithm has been proposed which is able to detect the watermark irrespective of whether it is embedded in the plaintext and then the watermarked data is encrypted or first the plaintext is encrypted and then the encrypted data is watermarked. Watermark is detected without the knowledge of the decrypting key. But, the encryption that they have used permutes only the first 25 DCT coefficients. This is a weak encryption and the encrypted image leaks some information about the original image.

Hash functions are one way functions which accepts a variable length message M as input and produces a fixed size output called the hash code H (M). The hash value is appended to the message at the source. The receiver authenticates the message by recalculating the hash value. When we are concerned with the confidentiality of communication, we need to scramble the information using an encryption algorithm and a secret key K. While scrambling the data, if we can preserve some of its attributes, then those attributes can be used to generate the hash. So, the hash calculated using the plaintext data and the encrypted data will remain the same. Mathematically, it can be stated as,

$$H(E(M, K), K_{hash}) = H(M, K_{hash}) \tag{3}$$

where, M the message to be transmitted,E is the encryption function, K the encryption key, K_{hash} the hashing key and H the hashing algorithm. Note here by introducing a hashing key K_{hash} we have parameterized the hash value. Thus the hash digest which is obtained becomes equivalent to a message authentication code (MAC). Only the users who share the key K_{hash} with the source and also possess either the encrypted or original image will be able to verify the hash value. We also have to ensure that

$$H(E(M, K_1), K_{hash}) = H(E(M, K_2), K_{hash}) \tag{4}$$

where, K_1 and K_2 are two encryption keys.

Once we ensure that, we have a scheme wherein zero knowledge proof of data being authentic or not can be given. Such a hash remains invariant to encryption. Hence, we achieve two layers of security where each works independently. The purpose of the hash function is to generate the fingerprint of the block of data. The hash must be able to detect any changes in the data. The image hash is a bit-string that somehow captures the essentials of the data (image in our case). Our aim is to propose a key-dependent function that returns the same bits or numbers from images and their encrypted versions. By incorporating encryption and hashing in one system, we can have two levels of security and we can prove the authenticity of the data without actually revealing the information.

The possible application of such a scheme is when we have several nodes between the transmitter and the receiver. These nodes are points at which we want to check whether the data coming down the stream have not been manipulated with and they are coming from the true source but at the same time we don't want these nodes to have the knowledge of the actual information. Now, the question is why we have nodes between the transmitter and the receiver. The answer is that encryption-decryption is a time taking process. If the scrambled data has been manipulated, there should be same way to find it out by remaining in the encrypted domain itself so that we don't have

to go for decryption-hash calculation-encryption triples. This not only saves time but ensures that the nodes don't come to know what they are authenticating.

III. Encryption

In [8], it has been said that secrecy systems are systems where the meaning of the message is concealed by cipher or code. Encryption enables the scrambling of the data in such a way that it conceals the statistical information and the dependencies in the original plaintext. Traditional encryption algorithms like AES, DES have been employed for multimedia signals but have been found to be computationally intensive particularly when applied to large streams of data such as videos and audio clips. Chaos based encryption algorithms which are much simpler have been used in recent times. It excites the scientific community because of the various similarities between chaos based systems and encryption. Such systems possess the avalanche property which states that even when the smallest change occurs in the plaintext, the ciphertext changes dramatically. Classical encryption algorithms are sensitive to keys, while chaotic maps are sensitive to initial conditions and parameters. Cryptographic algorithms shuffle and diffuse data by rounds of encryption, while chaotic maps spread the initial region over the entire phase space via iterations [9]. Permutations are important mathematical building blocks for symmetric encryption systems in general, and block ciphers in particular. Permutation is a bijective map whose domain and range are the same. Permutation ciphers based on chaos have been proposed [10].

Let S be a set. A map $f : S \rightarrow S$ is a permutation if f is bijective (i.e. injective and surjective). The set of all permutations of S is denoted by $Perm^{S \rightarrow S}$. We employ a permutation cipher based on the Cat map. The Cat map is given by,

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod N \tag{5}$$

where, x_n and y_n represent the rows and columns of the data points respectively, N is the number of columns in data block to be permuted. We have taken a block size of 16×16 where the data points are actually the DCT coefficients of the image. The Cat map is employed for a number of iterations for each 16×16 block. The secret key K decides the number of iterations for which Cat map will be employed for each block. The secret key also decides the values of parameters p and q.In our simulations, we have considered 256×256 image. So, in total we are having 256 blocks on which Cat map has to be employed.The inverse Cat map for decryption is given by,

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} pq+1 & -p \\ -q & 1 \end{bmatrix} \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} \bmod N \tag{6}$$

A. Need for softening the encryption

Note here that if a stronger cipher is to be implemented then this cat-map can be extended to the pixel values also. Thus a mixed cipher (substitution + permutation) could have been implemented. This will completely scramble the image (pixel values and positions). However we will later find that we need to allow a portion of the statistical signature of the original image to percolate. For instance, this statistical signature can be just the first order and second order statistics. We will show in the next section that if we constrain the encryption process to a simple permutation cipher, then the mean and variance of the individual blocks remain unchanged. The residual statistical signature can be then be processed into a hash value. If the chosen statistical measures before and after encryption remain the same, then the hash value will remain invariant to encryption.

IV. Proposed Algorithm for calculating the Invariant Hash

We propose a novel algorithm to calculate the hash value of the data so that the encryption process remains transparent to the Hash function. Mathematically,

$$H(E(M,K),K_{hash}) = H(M,K_{hash}) \tag{7}$$

where, M the message to be transmitted,E is the encryption function, K the encryption key, K_{hash} the hashing key and H the hashing algorithm. We also have to ensure that

$$H(E(M,K_1),K_{hash}) = H(E(M,K_2),K_{hash}) \tag{8}$$

where, K_1 and K_2 are two encryption keys. To start with, first we compute the 16x16 block DCT of the image. In order to do that, first we divide the image into blocks each of dimension 16x16.In total, we have 256 such blocks. For the k^{th} block A_k , the 2-dimensional DCT B_k is given as,

$$B_{k_{ij}} = a_i a_j \sum_{m=0}^{15} \sum_{n=0}^{15} A_{k_{ij}} \cos \frac{\pi(2m+1)i}{2M} \cos \frac{\pi(2n+1)j}{2N} \tag{9}$$

where, $0 \leq i \leq 15, 0 \leq j \leq 15$ and

$$a_i = \left\{ \begin{array}{l} \frac{1}{4}, i = 0 \\ \frac{1}{\sqrt{8}}, 1 \leq i \leq 15 \end{array} \right\}, a_j = \left\{ \begin{array}{l} \frac{1}{4}, j = 0 \\ \frac{1}{\sqrt{8}}, 1 \leq j \leq 15 \end{array} \right\}$$

After calculating the DCT coefficients for each block, we apply the Cat map individually to each of the block. The secret key decides the values of the parameters p and q and the number of iterations for which Cat map will be employed for each of the block. Since, we are applying a permutation cipher which scrambles only the positions of the DCT coefficients within the block, the statistics of the block like its mean and variance remain the same. So, in order to generate the hash, we select the means and variances of the blocks as our feature space. This feature space remains invariant to the encryption process. Hence, the hash of the original image and the scrambled image remain the same. The mean of the k^{th} block B_k is given by,

$$mean_{B_k} = \frac{1}{256} \sum_{i=0}^{15} \sum_{j=0}^{15} B_k(i,j) \tag{10}$$

The variance of the k^{th} block B_k is given as,

$$Var_{B_k} = \sum_{i=0}^{15}\sum_{j=0}^{15}\{B_k(i,j) - mean_{B_k}\}^2 \tag{11}$$

The means and the variances are normalized using the following equations,

$$norm\_mean_{B_k} = \frac{mean_{B_k}}{\max\limits_{k\in\{1,2.....256\}}\{mean_{B_k}\}} \tag{12}$$

$$norm\_Var_{B_k} = \frac{Var_{B_k}}{\max\limits_{k\in\{1,2.....256\}}\{Var_{B_k}\}} \tag{13}$$

So, we have as our feature space the normalized means and variances of 256 blocks where, $norm\_mean_{B_k}, norm\_Var_{B_k} \in (0,1)$. Next we sort out the blocks based upon the increasing values of the normalized variances. K_{hash} has two components- K_{hash1} and K_{hash2}.The top n blocks are selected based upon the secret key K_{hash1}. The means and variances of the selected blocks are then quantized. Suppose one of the normalized mean values is 0.7924. The binary equivalent of 0.7924 is 0.11001010110110101. The bit patterns that we obtain after quantizing the variance and mean are concatenated together. The concatenated bit patterns obtained for each block are stacked together in a vector P Then, using the key K_{hash2}, we generate the random sequence between 0 and 1 of length 100. Then, we multiply each of the random number with the size of vector P and round off the products. The bits corresponding to the indices indicated by the rounded-off products are selected from the vector P. This is our required hash. The above calculations are done both for the plaintext data as well as the encrypted data and the resultant hashes are found to be the same as shown in Fig. 1.

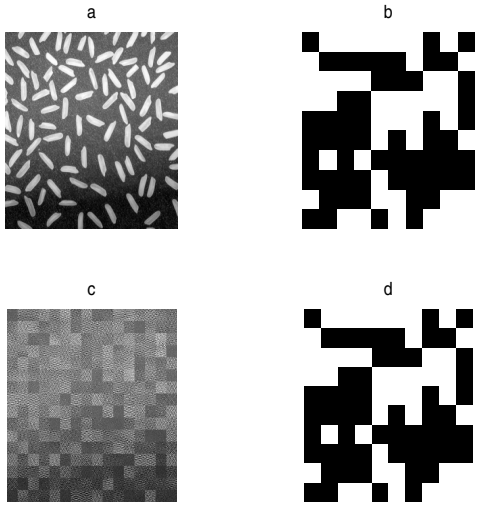

Fig. 1. Results showing the validity of the proposed algorithm.The hash of the original image and the encrypted image are same.(a) Original Rice image, (b) Hash derived from Original Rice image, (c) Encrypted Rice image, (d) Hash derived from Encrypted Rice image.

Thus, we achieve a hashing algorithm that remains transparent to the encryption process. In this way, in order to calculate the hash, we need not have to go through the decryption-hash calculation-encryption triples. Also, the load at the receiver side can be shared by a node. This node only calculates the hash value and in no way comes to know about the information being communicated between transmitter and receiver. This also saves time because if the data has been manipulated, the hash will change and we need not have to do decryption.

V. Simulation Results

In order to show the credibility of the proposed algorithm, it is tested across a variety of images for two reasons:

- The hash value should be unique to a given image. Different images should yield significantly different hash values. If the distance between hash values from two different images are significantly different, this can be used as a means of indexing the respective images.
- The hash invariance to encryption must be verified for different images inorder to justify this generalization.

The original images are shown in Fig. 2. For each image, first we compute the 16×16 block DCT. Then, each block is encrypted. Chaos encryption based on Cat map has been employed. The key K decides the values of p, q and the number of times the Cat map will be iterated for each of the blocks. The results obtained from four images are shown in Fig. 3. The security is strong because not only the parameters p and q are decided by the key but we also have randomized the number of iterations for the Cat map.

Our next step is to calculate the hash value of the original image and its corresponding encrypted version. As expected, they are found to be the same. The hashes obtained for each of the images is of 100 bits length. They are shown in the form of images of dimension 10×10 in Fig. 4. We also verify that the hash for each image obtained from the proposed algorithm is unique. In order to ensure that, we find the hamming distance between the hashes of different images. We XOR the two hashes and show the result in Fig. 5. From the figures we observe that the XORed results are virtually random patterns indicating that the parameterized hashes indeed capture the diversity of the chosen test images.

VI. Conclusions

In this paper we have explored a new framework for authenticating encrypted images. By allowing a portion of the statistical signature in the original image to surface despite the encryption operation, it becomes possible to validate the authenticity of the encrypted image without tapping into its contents. By constraining the encryption process to be a block DCT permutation cipher, we have observed that the mean and variances of the blocks remain the same even after encryption. We have used these two features to construct the hash value. This simple choice of features also depicts a significant variability across a variety of images. Future work entails developing hashes for much stronger ciphers.

Fig. 2. Original images used in simulations. (a) Pout, (b) Rice, (c) Cameraman, (d) Tire

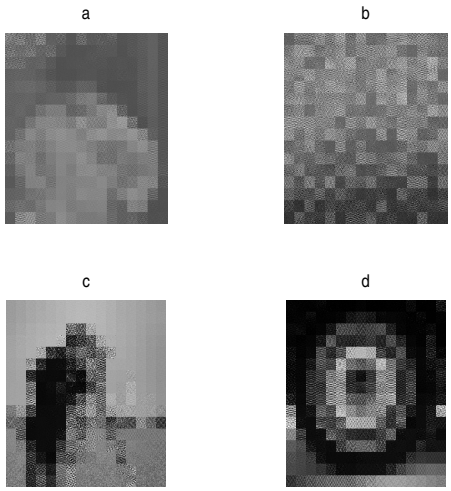

Fig. 3. Encrypted images after Cat map based encryption algorithm has been employed. The key K decides the values of parameters p and q. The key also decides the number of iterations for which Cat map will be employed for 256 blocks each of size 16x16. (a) Encrypted Pout, (b) Encrypted Rice, (c) Encrypted Cameraman, (d) Encrypted Tire

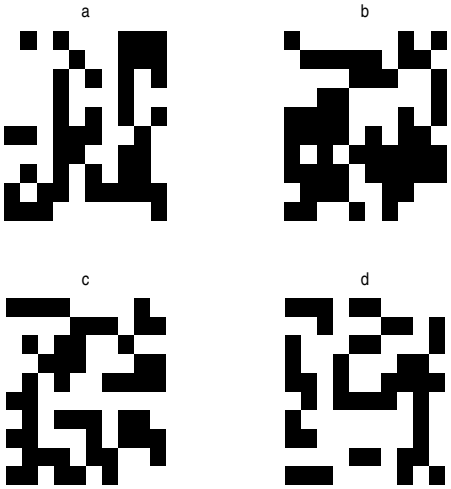

Fig. 4. The hashes obtained from the proposed algorithm. These hashes remain transparent to the encryption process. This is verified experimentally by finding out the hash of the original image and the encrypted image. Also, the hashes obtained from two different encrypted versions (same encryption algorithm but different keys used) of the same original image remain equal. (a) Hash of Pout, (b) Hash of Rice, (c) Hash of Cameraman, (d) Hash of Tire

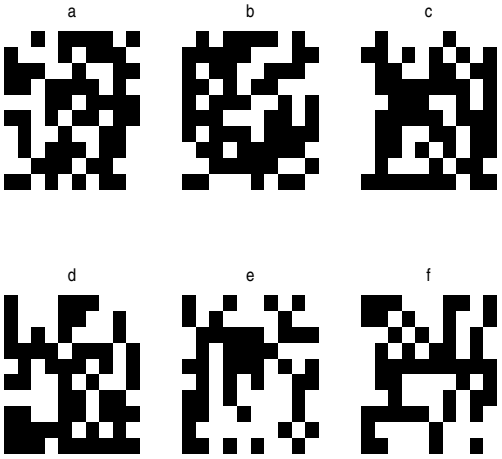

Fig. 5. The various hamming distance among the hashes obtained from the proposed algorithm. Hamming distance between hashes of (a) Pout and Rice, (b) Pout and Cameraman , (c) Pout and tire, (d) Rice and Cameraman, (e) Rice and Tire, (f) Cameraman and Tire.

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, "Watermarking applications and their properties," in *Proc. Int. Conf. on Information Technology: Coding and Computing*, pp. 6–10, March 2000.

[2] J. Fridrich, "Image Watermarking for Tamper Detection," in *Proc. ICIP*, vol. 2, (Chicago, Illinois), pp. 404–408, Oct 1998.

[3] E. T. Lin, C. I. Podilchuk, and E. J. Delp, "Detection of image alterations using semi-fragile watermarks," in *SPIE Intl. Conf. on Security and Watermarking of Multimedia Contents II*, Jan 2000.

[4] H. Cheng and X. Li, "Partial Encryption of Compressed Images and Videos," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2439–2451, 2000.

[5] S. Lian, "Quasi Commutative Watermarking and Encryption for Secure Media Content Distribution," *Multimedia Tools Appl, Springer*, vol. 43, pp. 91–107, 2009.

[6] S.Lian, Z. Liu, R. Zhen, and H. Wang, "Commutative Watermarking and Encryption for Media Data," *OE Lettters, SPIE*, vol. 45(8), 2006.

[7] G. Boato, V. Conotter, F. G. B. D. Natale, and C. Fontanari, "A joint asymmetric watermarking and image encryption scheme," in *Proceedings of SPIE Electronic Imaging*, vol. 6819, pp. 601–602, 2008.

[8] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, Oct 1949.

[9] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals, Elsevier*, pp. 749–761, 2004.

[10] Z. Lv, L. Zhang, and J. Guo, "A Symmetric Image Encryption Scheme Based on Composite Chaotic Dispersed Dynamics System," *Proc. of Second Symposium on Computer Science and Computational Technology*, pp. 191–194, 2009.