Compliant Asymmetric Authenticated Encryption Scheme for JPEG2000 Code-streams

Jinyong Fang[†, †††], Jun Sun[†], and Haifeng Qian[††]

† Institute of Image Communication and Information Processing,Shanghai JiaoTong University,Shanghai,210030,China
†† Department of Computer Science and Engineering,Shanghai JiaoTong University,Shanghai,210030,China
††† Faculty of Electronic and Information Engineering, Zhejiang Wanli University, Ningbo, 315100,China

Summary
The JPEG2000 syntax requires that any two consecutive bytes in the encrypted packet body should not be larger than 0xFF8F. This stringent requirement has plagued researchers for a few years. In this paper, we present a novel secure encryption and authentication scheme for JPEG2000 code-streams, which does not introduce superfluous JPEG2000 markers in the protected code-stream. The scheme achieves nearly 99.6% of the information protection for data confidentiality and it is computational efficiency. We develop a new public key method. It also provides source authentication without appending additional bits into the raw JPEG2000 code-streams, thus the compliant authenticated encryption is achieved.

Key words:
Authenticated encryption, Compliant encryption structure, JPSEC, JPEG2000

1. Introduction

JPEG2000 is the latest international still image compression standard [1,2]. On top of a very efficient image compression scheme, it also offers new compelling functionalities required by multimedia applications, such as progressive transmission up to lossless coding, seamless scalability, region of interest and error resilience.

However, the ease to manipulate digital images and to copy and distribute them at a negligible cost also raises the issues of content protection, authentication and data integrity. Recognizing that security is major issue in many imaging applications, JPEG (Joint Photographic Experts Group) has initiated a work item known as Secure JPEG2000 or JPSEC. It's the part 8 of JPEG2000 standard. Security and authentication are two major technical issues in JPSEC. A JPEG2000 code-stream is composed of markers and data packets. The markers with values restricted to the interval [0xFF90, 0xFFFF] are used to delimit various logical units of the code-stream, facilitate random access, and maintain synchronization in the event of error-prone transmission. The packets carry the content bit-streams whose codewords (i.e., any two contiguous bytes) are not in the interval [0xFF90, 0xFFFF]. Since the output of a good cipher appears "random", straightforward application of a cipher to encrypt code-stream packets is bound to produce encrypted packets, which include superfluous markers. Such markers will cause potentially serious decoding problems (such as loss of code-stream synchronization and erroneous or faulty image transcoding). To overcome the superfluous markers problem, the encryption method must be JPEG2000 code-stream syntax compliant. Such a compliant encryption method does not introduce superfluous markers in the encrypted packets and maintains all the desirable properties of the original code-streams.

Usually, the confidentiality of delivered data is provided by encryption algorithm, and the authentication of messages is guaranteed by digital signature. To encrypt and authenticate the code-stream simultaneously without changing the structure of JPEG2000 code-stream is a challenging work.

This paper presents a compliant authenticated encryption scheme. The scheme achieves nearly full protection for data confidentiality (99.6% of the information) and computational efficiency. By using a public key method, it provide source authentication simultaneously. In addition, the protected code-streams inherit all the desirable properties of the original JPEG2000 code-streams, e.g. error resilience and scalability.

The rest of this paper is organized as follows. Section 2 illustrates the related work on multimedia encryption. Section 3 introduces our compliant authenticated encryption scheme for JPEG2000. Section 4 shows the performance and analysis. In Section 5, we conclude this paper.

2. Related work on multimedia encryption

Due to the popularity of Internet and digital library applications, the intellectual property right (IPR) protection is becoming increasingly important. Encryption is frequently used to protect the multimedia content. A big

challenge for authenticated multimedia encryption (in the compressed domain) is to maintain syntax compliance.

Some shemes [3,4] selectively shuffle MPEG streams using shuffling tables so as to maintain syntax compliance. This shuffling method was generalized to index mapping [5]. However, the schemes are not applicable to JPEG2000 code-stream because the shuffling tables are too large to be implemented in practice.

Conan[6] described a technique which selectively encrypt JPEG2000 code-streams in order to generate compliant encrypt JPEG2000. In this scheme, if any byte, say *X*, has a value less then 0xF0, the four LSBs (Least Significant Bits) of *X* are encrypted with a block cipher. Clearly, the security of this scheme is weak. Canon inc. proposed another word-level scheme [7], which encrypts a word recursively until the ciphertext is compliant. This scheme not only has to check the value of the current encrypted word, but also it's proceeding byte and succeeding byte. Later Ma [8] pointed out that Canon's proposal was not reversible, i.e., portion of a plaintext could not be recovered from the corresponding ciphertext.

Wu[9] proposed two packet-level encryption schemes based on stream ciphers and block ciphers, respectively. They showed that the two schemes protect 99% of a code-stream. However, the schemes are not able to regain synchronization when some transmission error occurs and they could not achieve authentication while encrypting the packet streams.

Later Wu and Deng[10] gave a code-block level compliant scheme. They claimed that this scheme could provide full protection of code-streams. However, this algorithm has a small probability of not generating conditional-satisfied encrypted code-stream forever. Even if it can generate compliant out streams in many time, its iterative method is computationally inefficient.

2.1 Structure of JPEG2000 Code-Stream

Here we give a brief depiction of the JPEG2000 code-stream structures. The detail description can be obtained from [1,2]. The JPEG2000 coded image data is referred to as code-stream, instead of bit-stream as in JPEG and MPEG. The term "code-stream" refers to both the coded image data and the signaling markers and marker segments which are used to locate and describe coding parameters and auxiliary information.

In the simplest case, a JPEG2000 code-stream is structured as a main header followed by a sequence of tile-streams. The code-stream is terminated by a two-byte marker EOC (end of code-stream). This is depicted graphically in Fig.1. The main header consists of markers and marker segments containing global information necessary for decompression of the entire code-stream. Each tile-stream consists of a tile header followed the compressed packet-stream data for a single tile. Each tile header consists of markers and marker segments containing the information necessary for decompressing the packet-stream of its associated tile. Finally, the packet-stream of a tile consists of a sequence of packets.

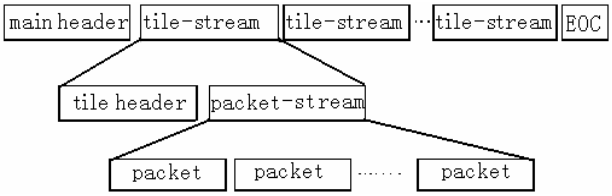

FIG. 1. JPEG2000 CODE-STREAM

The structure of a packet is depicted in Fig.2. A packet consists of a packet header followed by a packet body. To note, the Standard ensures none of the code-stream's delimiting marker codes (these all lie in the range 0xFF90 through 0xFFFF) can appear in the packet-stream except marker segment SOP (start of packet) and marker EPH (end of packet header).

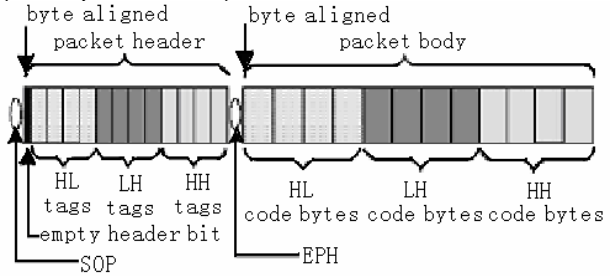

FIG. 2 JPEG2000 PACKET STRUCTURE

2.2 Authenticated Encryption Scheme

An authenticated encryption scheme [11] allows messages to be encrypted and authenticated simultaneously. We mainly consider the scenario of communication on a public channel. It is dangerous to accept JPEG2000 pictures if we can't confirm the sender's identity, for nowadays the JPEG virus has been flooding on INTERNET. In [12], the authors proposed a general method (digital signature) to authenticated the JPEG2000. In that scheme, additional data have been attached to the JPEG2000 code-stream. The Authenticated Encryption Scheme allows us to identify the source information, which indicate the identity of the sender. We use a very simple technique to achieve this property. In order to keep privacy of the JPEG2000 pictures the sender must send encrypted JPEG2000 pictures to receiver online.

While in a symmetric encryption algorithm the key must be known by the two parties of the communication. Therefore, how to share a secret key becomes a very important thing in communication online. We modify the Diffie and Hellman's protocol [13] to an asymmetric

authenticated encryption algorithm to achieve the authentication and secrecy simultaneously. Due to the structure of JPEG2000 code-stream, how to design an authenticated encryption scheme is not a trivial thing.

In our scheme the sender and receiver both have a pair of public key and private key. The public key certificated by Certificate Authority (CA) implies the identity of the user. Moreover, there is no additional message appended to JPEG2000 code-stream, thus there is no need to send the encryption key secretly, while the scheme still keep high efficiency.

3. Compliant Authenticated Encryption Algorithm

In this section, we describe the algorithm of generating the shared secret key by using public key cryptography. Then, we show how to encrypt the packet body in a compliant way using the generated key. According to the JPEG2000 code-stream structures, all marker codes that are in the range of 0xFF90 to 0xFFFF should not appear in the packet body. The image data are compressed into the range between 0x0000 and 0xFF8F. So, the encryption process must be organized to obey this restrict. The following subsections elaborate how to adapt stream codes to generate compliant encrypted code-streams.

3.1 Encryption seed generation

Our construction is based on the static DH shared key between the communication parties. The scheme set as follows:

Setup:

1. Two large primes p and q, where $q \mid p-1$
2. An element $g \in Z_p^*$ which has the order of q
3. One-way hash function $H: \{0,1\}^* \rightarrow \{0,1\}^K$
4. The sender who has a private key $S_A \in Z_q^*$ and the corresponding public key $y_A = g^{S_A} \bmod p$
5. The receiver who has a private key $S_B \in Z_q^*$ and the corresponding public key $y_B = g^{S_B} \bmod p$

Generating the share key:

1. The sender randomly choose a number r
2. Compute $K = y_B{}^{S_A} \bmod p$ and $S = H(K \| r)$
3. The sender use the generated S as secret seed to our encryption algorithm and send r to receiver on a public channel.
4. The receiver compute $K' \equiv y_A{}^{S_B} \equiv y_B{}^{S_A} \equiv g^{S_B S_A} \equiv K \bmod p$ then compute $S' = H(K' \| r)$ to decrypt the encrypted JPEG2000 code-streams, where $S' \equiv S$.

3.2 Compliant Code-stream Encryption

Let M express a part of packet body, and $M = m_1 \| m_2 \| \cdots \| m_n$, where $\|$ denotes concatenation and each m_i depicts one byte in M. In the same way, we denote the ciphertext as $C = c_1 \| c_2 \| \cdots \| c_n$, where c_i depicts one byte and the key stream as $S = s_1 \| s_2 \| \cdots \| s_n$, where s_i denotes one byte. S_i is in the range of [0x0, 0xFF].

In our algorithm, we deal with any two consecutive bytes specially, if value of the first byte in this word is 0xFF. The scheme keeps value of the first byte as 0xFF, while the second byte is added with corresponding key (s_i) and then modulo 0x90. In this way, the processed value of the two consecutive bytes is less than 0xFF90, and value of the second byte is sure of not being 0xFF. Because in the unencrypted code-stream the value of a byte, which just followed after 0xFF, is less than 0x90, let the corresponding encrypted byte less than 0x90 is reasonable. This processing is a one-to-one mapping.

In the encrypting procession, any source byte m_i is encrypted by the corresponding key byte s_i. Even if a source byte is 0xFF, the corresponding s_i is skipped over. In this way, if there is an error byte in communication, the error propagation will be limited within two bytes at most. For example, if a byte is changed into 0xFF or a byte with value of 0xFF is changed into other value, the byte and its next are error decrypted. In other cases, the error is limited into one byte.

The compliant encryption process proceeds as follows:

(1) If the compressed data to be encrypted is the start data of a packet body:

if m_1 =0xFF, then $c_1 = m_1$;
else $c_1 = (m_1 + s_1) \bmod$ 0xFF;
for i = 2 to n,
  if m_{i-1} =0xFF ,then c_i =($m_i + s_i$)mod 0x90;
  else if m_i = 0xFF ,then $c_i = m_i$;
    if $m_i \neq$ 0xFF and $m_{i-1} \neq$ 0xFF,
    then c_i = ($m_i + s_i$)mod 0xFF;

(2) If it's not the start of a packet body, the algorithm judge value of the last byte in former ciphertext C in the first step. We sign this byte as c_n':

if c_n' = 0xFF, then $c_1 = m_1 + s_1$ mod 0x90;
else if m_1 = 0xFF, then $c_1 = m_1$;
  else $c_1 = m_1 + s_1$ mod 0xFF;
for i = 2 to n,
  if m_{i-1} =0xFF ,then c_i =($m_i + s_i$)mod 0x90;
  else if m_i = 0xFF ,then $c_i = m_i$;
    if $m_i \neq$ 0xFF and $m_{i-1} \neq$ 0xFF,

then $c_i = (m_i + s_i) \bmod 0xFF;$

3.3 Compliant Code-stream Decryption

The corresponding decryption process proceeds as follows:

(1) If the data to be decrypted is the start data of a packet body:

if c_1 =0xFF, then $m_1 = c_1$;
else $m_1 = (c_1 - s_1) \bmod 0xFF$;
for i = 2 to n,
if c_{i-1} =0xFF ,then $m_i = (c_i - s_i) \bmod 0x90$;
else if c_i = 0xFF ,then $m_i = c_i$;
if $c_i \neq$ 0xFF and $c_{i-1} \neq$ 0xFF,
then $m_i = (c_i - s_i) \bmod 0xFF$;

(2) If it's not the start of a packet body, the algorithm judge value of the last byte in former ciphertext C' in the first step. We sign this byte as c_n':

if c_n' = 0xFF, then $m_1 = (c_i - s_i) \bmod 0x90$;
else if c_1 = 0xFF, then $m_1 = c_1$;
else $m_1 = (c_i - s_i) \bmod 0xFF$;
for i = 2 to n,
if c_{i-1} =0xFF ,then $m_i = (c_i - s_i) \bmod 0x90$;
else if c_i = 0xFF ,then $m_i = c_i$;
if $c_i \neq$ 0xFF and $c_{i-1} \neq$ 0xFF,
then $m_i = (c_i - s_i) \bmod 0xFF$;

4. Performance and Analysis

4.1 Security

The scheme uses public key technique and the public key of the two communication parties is certificated by CA. The main security of the scheme is based on Computational Diffie Hellman (CDH) problem, which is a difficult mathematical problem if Discrete Logarithm problem is hard to solve. In key generation the sender randomly choose a number r, using the receiver's public key, his own secret key and a one-way hash function to generate the shared key. As the secret key has been used the receiver can confirm the sender's identity with the corresponding public key.

On the other hand, only the appointed receiver can get the shared key to decrypt the code-streams. For the scheme needs the secret key to generate the shared key. The attacker only break the CDH problem, then he can break our scheme. So our scheme's security is based on CDH problem and the scheme provide the properties (1) authentication; (2) data confidential.

4.2 Efficiency analysis

Our scheme has a relatively simple authenticated encryption structure. In our scheme, all image data are processed with a public key (S), except for the data with value of 0xFF. Denote the probability of this data occurs as $p_{FF.}$ We use many images to practically estimate this probability. Table 1 indicates the probability of several typical images. The average probability p_{FF} is nearly 0.398%. In other words, about 99.6% of the coded image data are protected, which is denoted as p_{pro}.

Table 1. The probability of information leakage in several images

	camer aman	barbar a	lena	rice
P_{FF}%	0.358	0.406	0.405	0.411
P_{pro}%	99.642	99.594	99.595	99.589
	tartan	belmo nt	tire	
P_{FF}%	0.418	0.395	0.393	

There are no iterations in our encryption algorithm, whereas many iterations occur in [10]. Therefore, more computational efficiency is achieved and it is easy to be realized in some real-time applications.

4.3 Error resilience

In our scheme, every data (m_i) in the original code-streams is correspondingly encrypted by the key (S_i). When a byte is changed into 0xFF or a byte with value of 0xFF is changed into others, the byte and its following byte will be decrypted improperly. In other cases, if there is a byte error occurring in the communication, it would only influence decryption of the byte itself. So the scheme can hold synchronous firmly and is error resilience.

5. Conclusion

JPSEC focuses on the security aspect of JPEG2000 standard. Encryption on JPEG2000 code-streams is of great importance in today's image and video communication applications. In this paper, we propose a novel public key encryption method, which can authenticate source data and encrypt code-streams simultaneously. The scheme is also very computational efficient and can protect about 99.6% of the information in images. Furthermore this encryption scheme generates fully compliant encrypted code-streams thus it maintains all the nice properties of original JPEG2000 code-streams, such as error resilience and scalability.

References

[1] "Information Technology – JPEG2000 image coding system", ISO/IEC International Standard 154441, ITU Recommendation T.800, 2000.

[2] Taubman D. and Marcellin M., "JPEG 2000: Image Compression Fundamentals, Standards and Practice", Kluwer Academic Publishers, 2002.
[3] Wen J.T., Severa M., Zeng W.J., M. H. Luttrell and Weiyin Jin, "A Format-compliant Configurable Encryption Framework for Access Control of Video," IEEE Transactions on Circuits and Systems for Video Technology, 12(6): 558-565,2002.
[4] Zeng Wenjun, Wen Jiangtao, Severa M., "Fast Self-synchronous Content Scrambling by Spatially Shuffling Codewords of Compressed Bitstreams," ICIP,pp. III 169-172, 2002.
[5] Wu Min and Mao Yinian, "Communication-friendly Encryption of Multimedia," IEEE Workshop on Multimedia Signal Processing, pp.292-295, 2002.
[6] Conan Vania, Sadourny Yulen and Thomann Stève, "Symmetric Block Cipher Based Protection: Contribution to JPSEC," ISO/IEC JTC 1/SC 29/WG1 N2771, Oct.2003.
[7] Canon inc. "Encryption Tool for JPEG2000 Access Control," ISO/IEC JTC 1/SC 29/WG1 N2839, 2003.
[8] Ma Di, Wu Yongdong and Deng Robert, "Analysis of Canon Encryption Scheme," Communications in JPEG2000 Security group (JPSEC), Nov. 14, 2003.
[9] Wu Hongjun and Ma Di, "Efficient and Secure Encryption Schemes for JPEG2000," ICASSP 2004, pp. V869-872, see also ISO/IEC JTC 1/SC 29/WG1 N2937, 2003.
[10] Wu Yongdong, and Deng Robert, "Compliant Encryption of JPEG2000 Codestreams", IEEE International Conf. on Image Processing, ISBN 0-7803-8555-1, IEEE Catalog Number 04CH37580C, Oct. 24-27, 2004, Singapore.
[11] Chen Tzer-Shyong, Huang Kuo-Hsuan and Chung Yu-Fang , "A practical authenticated encryption scheme based on the elliptic curve cryptosystem," Computer Standard & Interface, Volume 26, Issue 5, September 2004, pages 461-469.
[12] Grosbois Raphaël, Gerbelot Pierre and Ebrahimi Touradj, "Authentication and access control in the JPEG2000 compressed domain," Proceedings of SPIE Volume 4472, Applications of Digital Image Processing XXIV, Andrew G. Tescher, Editor, December 2001, pp. 95-104.
[13] Diffie W., Hellman, "M. New Directions in Cryptography," IEEE Transaction on Information Theory, VOL.22 pages:644-654,NOV 1976.