Signal and Image Processing in the Encrypted Domain

Hitoshi Kiya[1] and **Masaaki Fujiyoshi**[2], Non-members

ABSTRACT

This paper describes signal processing in the encrypted domain, i.e., that after encryption but before decryption. In this framework, signal processing operations can be directly applied to encrypted signals without decrypting of encrypted signals, whereas the ordinary framework encrypts signals for transmission and/or storing but it decrypts them before signal processing operations are applied to. The described framework befits contemporary cloud computing in which not only transmission but also storing and processing are done in the public Internet. Addition to brief survey, two tangible application scenarios are also demonstrated in this paper where a new signal processing algorithm is introduced each.

Keywords: Image Compression, Security, Discrete Cosine Transformation, Sign Correlation, Image Identification, JPEG 2000, Zero-Bit-Plane

1. INTRODUCTION

With a development and spreading of digital devices such as still cameras, video cameras, audio recorders, and other sensors, tons of signals are acquired in every second all over the world. Even signals are privacy sensitive such as videos in surveillance systems and/or commercially sensitive such as digital cinema movies, not only processing but also transmitting of signals are necessary to exploit them, in particular, in this Internet era. Moreover, cloud computing [1, 2], a very recent application on the Internet, saves your resource such as CPU power, storage, memory, and so on, but it stores signals somewhere in the Internet to process them. To overcome such situation, it is desired that signal processing in the encrypted domain. That is, a signal is firstly encrypted, and processing operations are directly applied to the encrypted signal, whereas the encrypted signal should be decrypted to be processed in the conventional framework.

Though signal processing in the encrypted domain was firstly discussed in 1987 [3], the big movement has been in the last half decade. In Dec. 2006, a three year-long project dedicated to this topic, called SPEED (Signal Processing in the EncryptEd Domain) [4], started in Europe with the planned budget of about 1900 million €. In the very next year, 2007, EURASIP Journal on Information Security published the special issue on signal processing in the encrypted domain [5]. Workshops dedicated to this topic were also held in 2007 [6] and in 2009 [7]. Later, of not only EURASIP but also IEEE, major international conferences had a special session on signal and image processing in the encrypted domain [8, 9], and a keynote speech was also given in an international workshop [10].

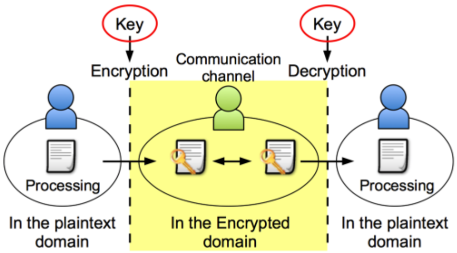


Fig.1: *Signal processing in the plaintext domain.*

Through the above mentioned promotions of and events for research activities, some fundamental signal processing operations have been implemented: linear filtering [11, 12], sum-product of two signals [13], discrete Fourier transformation [14–16], and so on. These operations, however, cannot be used in all scenarios and/or applications, so further research and development are desired. In addition, application oriented/specific approach as well as the above mentioned generalizing approach is also encouraged to achieve the practical operations.

This paper describes signal and image processing in the encrypted domain. The rest of the paper is organized as follows. In Section 2.signal and image processing in the encrypted domain is briefly reviewed. Sections 3and 4show a tangible example of the topic in each. Finally, conclusions are drawn in Section 5.

2. SIGNAL PROCESSING AND ENCRYPTION

This section describes two frameworks in which signals will be encrypted for transmission and storage; one is an ordinary framework in which encrypted signals are decrypted before signal processing operations will be applied to signals, and the other is a framework in which signal processing operations are

Manuscript received on March 31, 2012 ; revised on April 20, 2012.

[1,2] The authors are with Department of Information and Communication Systems, Tokyo Metropolitan University, Japan., E-mail:kiya@sd.tmu.ac.jp and fujiyoshi-masaaki@tmu.ac.jp

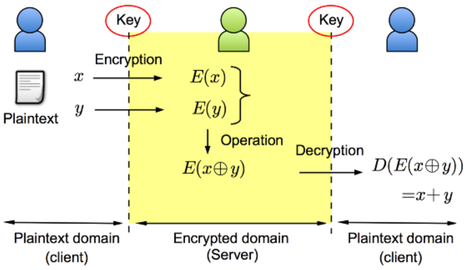
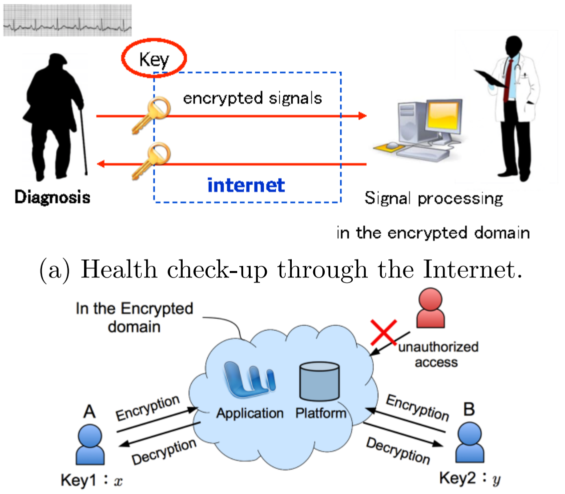


Fig.2: *Signal processing in the encrypted domain.*



(a) Health check-up through the Internet.

(b) Cloud computing [1, 2].

Fig.3: *Applications of signal processing in the encrypted domain.*

directly applied to encrypted signals, i.e., signal processing in the encrypted domain.

2.1 Ordinary Framework

In this framework, signal processing operations are applied to signals only when signals are not encrypted. This is referred to as signal processing in the plaintext domain in this paper. Figure 1 shows a conceptual diagram of this framework.

As shown in Fig. 1, at the transmitter side, a signal in its original form is processed and then be encrypted. So, the signal which is transmitted to the receiver side through a communication channel is encrypted. The signal which is arrived at the receiver is first decrypted and then signal processing operations are applied to the decrypted signal. That is, signals are processed in the plaintext domain even the signals are transmitted in those encrypted form.

This framework, thus, provides security only in communication channels, and signal decryption is required before signal processing. Consequently, the framework restricts signal processing tasks to only in the plaintext domain.

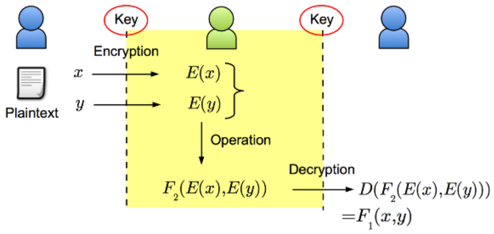


Fig.4: *Homomorphic encryption.*

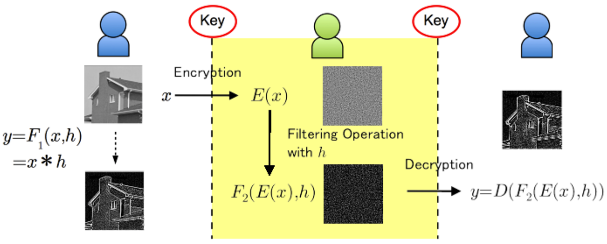


Fig.5: *Filtering in the encrypted domain.*

2.2 Signal Processing in the Encrypted Domain

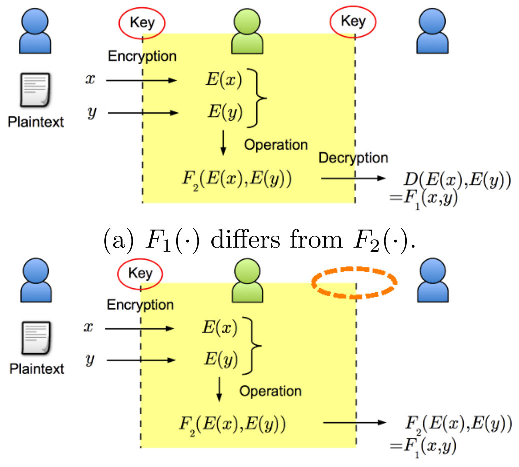
On the other hand, in the framework which is focused in this paper, signal processing operations are directly applied to encrypted signals. An example of this framework is shown in Fig. 2.

In Fig. 2, signals x and y are encrypted at the client side and encrypted signals $E(x)$ and $E(y)$ are transmitted to the server through a communication channel, where $E(\cdot)$ is an encryption operation. At the server side, an operation is directly applied to $E(x)$ and $E(y)$ to obtain a result, e.g., $E(x \oplus y)$ in Fig. 2. Processed encrypted signal $E(x \oplus y)$ is now returned to the client or further transmitted to another client, and $E(x \oplus y)$ is decrypted to a practical result, e.g., $x + y = D(E(x \oplus y))$ in Fig. 2 where $D(\cdot)$ is a decrypting operation which corresponds to $E(\cdot)$.

In this framework, signal processing operations are done without any keys, i.e., without any decrypting operations. Signals, thus, are securely protected when signal processing operations are applied to signals. Consequently, this feature serves secure applications even the service is in the Internet, i.e., this framework suits applications such as health check-up through the Internet or cloud computing as shown in Fig. 3.

Though signal processing in the encrypted domain is a desirable framework, it may seem to be impractical. To make this framework feasible, it is often based on homomorphic encryption in which specific types of computations can be directly applied to encrypted signals and the result of the computation is the same as the encrypted signal of the result of signal processing in the plaintext domain.

Figure 4 shows the conceptual diagram of homomorphic encryption. The aim of this encryption is obtaining $F_1(x, y)$ which is the output of function de-



(a) $F_1(\cdot)$ differs from $F_2(\cdot)$.

(b) $F_1(\cdot)$ is the same as $F_2(\cdot)$.

Fig. 6: *Two scenarios of signal processing in the encrypted domain.*

fined in the plaintext domain, $F_1(\cdot)$, whose inputs are signals x and y. It, however, should be given without any direct access to x and y themselves.

In Fig. 4, signal x and y are encrypted as $E(x)$ and $E(y)$, respectively. With a function defined in the encrypted domain, $F_2(\cdot)$, encrypted signals $E(x)$ and $E(y)$ are processed as $F_2(E(x), E(y))$. Homomorphic encryption gives $F_1(x, y)$ by decrypting $F_2(E(x), E(y))$, i.e.,

$$D\left(F_2\left(E(x), E(y)\right)\right) = F_1\left(x, y\right). \tag{1}$$

Though signal processing operation $F_2(\cdot)$ only access encrypted signals $E(x)$ and $E(y)$, the result given by $F_1(x, y)$ is obtained in homomorphic encryption.

Although all operations cannot be implemented in homomorphic encryption, addition or multiplication is given in homomorphic encryption. In addition, based on homomorphic encryption, linear filtering which is the most common operation in signal processing is served as shown in Fig. 5. Fig. 5 computes

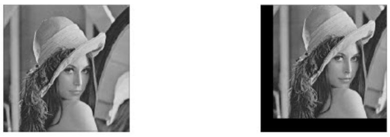
$$y = F_1(x, h) = x * h, \tag{2}$$

where signal y is the output of filtering operation $F_1(\cdot)$ whose inputs are signal x and filter h. However, x is first encrypted and it is then fed to filter operation $F_2(\cdot)$ in the encrypted domain, i.e., $F_2(E(x), h)$, instead applying $F_1(\cdot)$ to x with h in the plaintext domain. This filtering gives y by decrypting of $F_2(E(x), h)$, i.e.,

$$y = D\left(F_2\left(E(x), h\right)\right). \tag{3}$$

Thus, the filtering operation can be done before decryption.

In the signal processing in the encrypted domain framework, an original signal is always encrypted and the signal is processed without decryption, i.e., the original signal is protected against not only eavesdroppers in the Internet and unauthorized access but also a signal processing operator. The results of the signal processing operation, however, could be either encrypted or unencrypted, c.f., Fig. 6.



(a) Original. (b) Shifted.

Fig. 7: *Two images which (b) is a shifted version of (a) (20 pixels shifted in each axis).*

In Fig.6(a), operation $F_2(\cdot)$ which is defined in the encrypted domain differs from that in the plaintext domain, $F_1(\cdot)$. It is natural that $F_1(\cdot)$ for unencrypted signals such as x and y is different from $F_2(\cdot)$ for encrypted signals like $E(x)$ and $E(y)$. Decryption is required to obtain final result $F_1(x, y)$ from result in the encrypted domain $F_2(E(x), E(y))$. In other words, both signals and results are protected under this condition.

In contrast, Fig.6(b) does not need the decryption where two operations $F_1(\cdot)$ and $F_2(\cdot)$ are the same. This is a special form of the signal processing in the encrypted domain. Under this condition, final result $F_1(x, y)$ is the same as $F_2(E(x), E(y))$, i.e.,

$$F_2\left(E(x), E(y)\right) = F_1(x, y). \tag{4}$$

So, decryption of $F_2(E(x), E(y))$ is no need. In this form, only original signals are protected but the result is not protected.

From the next section, two tangible examples are given, namely, DCT sign correlation in the encrypted domain and identification of JPEG 2000 images in the encrypted domain.

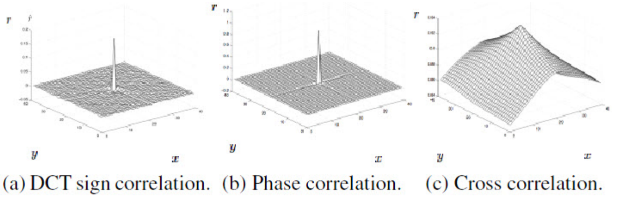
3. DCT SIGN CORRELATION IN THE ENCRYPTED DOMAIN

This section describes the DCT sign correlation [17] which is quite close to the phase correlation and the DCT sign correlation in the encrypted domain [18–20].

3.1 DCT Sign Correlation

The DCT sign correlation is the correlation between the positive and negative sign of discrete cosine transformed (DCTed) coefficients of signals [17]. Let N-point DCT of N-point real signal $g_i(n)$ be $G_i(k)$. The type II DCT is defined as

$$\begin{aligned} G_i(k) &= \sqrt{\frac{2}{N}} C_k \sum_{n=0}^{N-1} g_i(n) \cos\left(\frac{\pi(n+1/2)k}{N}\right) \\ &= |G_i(k)|\, \sigma_i(k), \end{aligned} \tag{5}$$



(a) DCT sign correlation. (b) Phase correlation. (c) Cross correlation.

Fig. 8: *Three correlations for two images shown in Fig. 7. The peak position is at* $(20, 20)$*.*

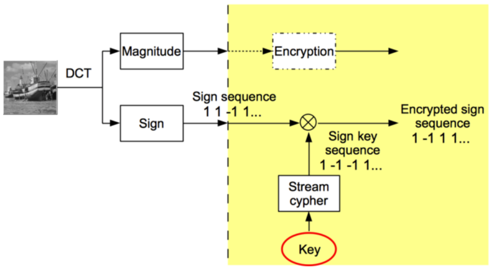


Fig. 9: *Encryption for DCT sign correlation.*

where

$$C_k = \begin{cases} 1/\sqrt{2}, & k = 0 \\ 1, & k \neq 0 \end{cases}, \tag{6}$$

and $|G_i(k)|$ and $\sigma_i(k)$ are the absolute value and the positive and negative sign of $G_i(k)$, respectively. The DCT sign product is, then, given as

$$R_\sigma(k) = \sigma_1(k)\sigma_2(k), \qquad k = 0, 1, \ldots, N-1, \tag{7}$$

and DCT sign correlation is defined based on $R_\sigma(k)$ as

$$r_\sigma(n) = \frac{1}{N}\sum_{k=0}^{N-1} K_k R_\sigma(k) \cos\left(\frac{\pi n k}{N}\right), \quad n = 0, 1, \ldots, N-1, \tag{8}$$

where K_k is the weight which is generally given as $K_k = (C_k)^2$.

For two 512×512-sized 8-bits grayscale images shown in Fig. 7 in which Fig. 7 (b) is a shifted version of Fig. 7 (a), the DCT sign correlation has a sharp peak value which the peak position indicates the displacement amount as well as the phase correlation, whereas the cross correlation which uses the amplitude and phase information has a smooth peak, c.f., Fig. 8. So, the DCT sign has the important information of images as well as the phase spectrum have.

This property of the DCT sign correlation is useful for applications such as similarity measurement, identification, and estimation of displacement amount, rotation angle, and scaling factor, similar to the phase correlation. In some applications, the DCT sign as well as phase spectrum should be protected because the DCT sign has the important information of its corresponding signal as mentioned above. So, the DCT sign correlation is expected to be done in the encrypted domain. The next section describes it.

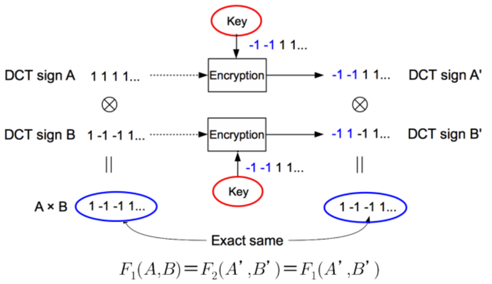
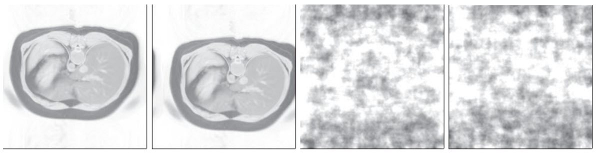


Fig. 10: *Basic principle of DCT sign correlation in the encrypted domain.*



(a) Original. (b) Shifted. (c) Encrypted (a). (d) Encrypted (b).

Fig. 11: *Test images* $(256 \times 256$ *pixels).*

3.2 DCT Sign Correlation in the Encrypted Domain

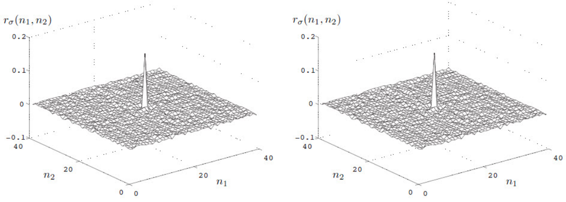
By introducing an appropriate encryption, DCT sign correlation can be held in the encrypted domain. Figure 9 shows the encryption for DCT sign correlation. This encryption first apply DCT to an original signal to obtain DCT coefficients $G_i(k)$, and then $G_i(k)$ are separated to those magnitude $|G_i(k)|$ and sign $\sigma_i(k)$. By using a sequence $s_{\alpha_i}(k) \in \{1, -1\}$ generated by a stream cipher with key α_i, sign $\sigma_i(k)$ are encrypted as

$$\tilde{\sigma}_i(k) = \sigma_i(k) s_{\alpha_i}(k), \tag{9}$$

where $\tilde{\sigma}_i(k)$ are encrypted signs. Applying the inverse DCT to encrypted DCT coefficient $\tilde{G}_i(k) = |G_i(k)|\,\tilde{\sigma}_i(k)$, scrambled signal $\tilde{g}_i(n)$ is given as

$$\tilde{g}_i(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C_k \tilde{G}_i(k) \cos\left(\frac{\pi(n+1/2)k}{N}\right). \tag{10}$$

For encrypted signal $\tilde{g}_i(n)$, DCT sign product $\tilde{R}_\sigma(k)$ is given as



(a) In the plaintext domain. (b) In the encrypted domain.

Fig. 12: *Displacement amount estimation by the DCT sign correlation. (a) and (b) are identical.*

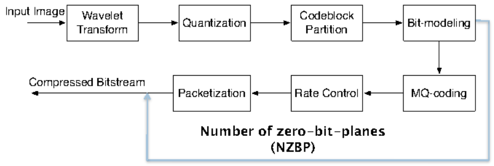


Fig.13: *Block diagram of JPEG 2000 encoder.*

$$\tilde{R}_\sigma(k) = \tilde{\sigma}_1(k)\tilde{\sigma}_2(k) = \sigma_1(k)s_{\alpha_1}(k)\sigma_2(k)s_{\alpha_2}(k). \tag{11}$$

If DCT sign of two signals are encrypted with one common ±1-sequence, i.e., $s_{\alpha_1}(k) = s_{\alpha_2}(k)$ for all k,

$$\tilde{R}_\sigma(k) = \sigma_1(k)\sigma_2(k) = R_\sigma(k), \tag{12}$$

and it results in that the DCT sign correlation of two encrypted signals, $\tilde{r}_\sigma(n)$, is the exact same as that of the original signals as shown in Fig. 10 [18–20].

3.3 Experimental Results

Applying the above mentioned encryption to Figs. 11 (a) and (b) yields encrypted images shown in Figs. 11 (c) and (d), respectively. It is noted that encryption is done with the same key, i.e., $\alpha_1 = \alpha_2$, and it results in encryption is done with the same sequence, i.e., $s_{\alpha_1}(k) = s_{\alpha_2}(k), \forall k$.

Figure 12 shows the displacement amount estimation by the DCT sign correlation. Fig. 12 (a) is for Figs. 11 (a) and (b), i.e., in the plaintext domain. On the other hand, for Figs. 11 (c) and (d), the estimation is shown as Fig. 12 (b).

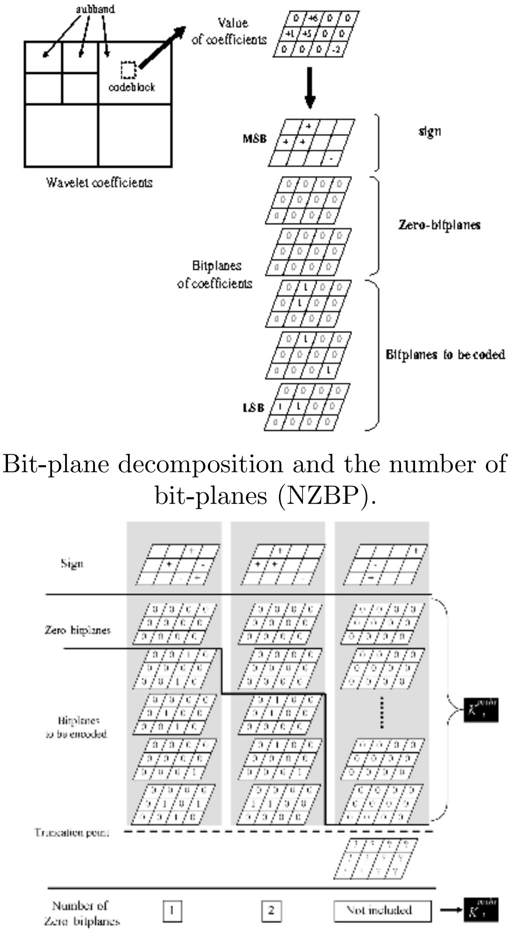
It is noted that DCT sign correlation in the encrypted domain $F_2(\cdot)$ is the same as DCT sign correlation in the plaintext domain $F_1(\cdot)$ by the introduced DCT sign encryption, i.e., $F_1(\cdot) = F_2(\cdot)$ as shown in Fig. 6 (b), because

$$F_1(A, B) = F_2(A', B') = F_1(A', B') \tag{13}$$

as shown in Fig. 10. Moreover, since JPEG for still images and MPEG for video sequences use DCT as those fundamental transformation and they put DCT sign to a compressed codestream separately from its corresponding DCT magnitude, DCT sign can be easily extracted from a compressed codestream by partial decoding. So, DCT sign correlation in the encrypted domain described in this section has close relations with compressed data.

4. IDENTIFICATION OF JPEG 2000 IMAGES IN THE ENCRYPTED DOMAIN

This section describes identification of JPEG 2000 images in the encrypted domain [21] which is signal processing in the compressed and encrypted domain.



(a) Bit-plane decomposition and the number of zero-bit-planes (NZBP).

(b) NZBP and rate control.

Fig.14: *Number of zero-bit-planes in JPEG 2000.*

4.1 JPEG 2000

Here, the overview of JPEG 2000 technology is given. JPEG 2000 is an international standard for compression of still images and video sequences. As shown in Fig. 13, JPEG 2000 first applies discrete wavelet transformation (DWT) to an input image to obtain DWT coefficients. By analyzing and encoding the DWT coefficients, a codestream consisting of not only rate-controlled encoded DWT coefficients but also the number of zero-bit-plane (NZBP) information is output, where identification of JPEG 2000 images in the encrypted domain described in this section is based on NZBP.

In JPEG 2000, several DWT coefficients are gathered to form a codeblock in each subband and DWT coefficients in a codeblock are divided to bit-planes as shown in Fig. 14 (a). A zero-bit-plane is a bit-plane whose elements are all zeros in a codeblock as shown in Fig. 14 (b). For a codeblock in which all bit-planes are zero-bit-planes, special information "not included" instead of the NZBP itself is stored to the compressed codestream. It is noted that the rate control in JPEG 2000 cuts down less significant bit-

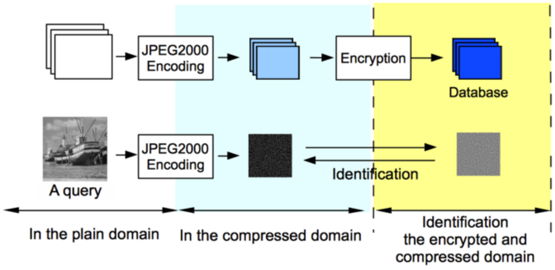


Fig.15: *Identification of JPEG 2000 image in the encrypted domain.*

planes based on rate-distortion curves. Since zero-bit-planes are more significant bit-planes as shown in Fig. 14 (b), the rate control in JPEG 2000 does not affect NZBP severely. That is, NZBP is almost independent of the compression ratio. So this identification utilizes NZBP [22–25].

JPEG 2000 is also the standard compression technology for digital cinema. In the digital cinema application, to edit and/or re-encoding of frames, a frame should be identified from a compressed codestream. Since video sequences are commercial in the digital cinema application, the identification is desired to be done in the encrypted domain. Moreover, JPEG 2000 codestreams in the digital cinema application have huge volume, and lightweight identification is expected [22–25].

4.2 Identification of JPEG 2000 Image in the Encrypted Domain

Identification of JPEG 2000 image in the encrypted domain described here first encodes images and/or video sequences by JPEG 2000 as shown in Fig. 15. Compressed images and/or video sequences are then encrypted and stored in a database. A query image is also encoded by JPEG 2000, and then is compared to compressed-and-encrypted images in the database.

To serve a secure and lightweight identification of JPEG 2000 image, a header parser for JPEG 2000 codestreams is introduced to extract NZBP from a header part in a JPEG 2000 codestream as shown in Fig. 16. It, thus, no full-decoding of JPEG 2000 codestreams is needed for either database setup or query. For database images, the body part of JPEG 2000 codestreams in which encoded DWT coefficients are stored is encrypted. In contrast, the NZBP is stored in the header part in JPEG 2000 codestreams as mentioned above, so neither decryption nor decoding is required to extract NZBP information from JPEG 2000 codestreams.

The query image is compared to an image in the database codeblock-by-codeblock based on the NZBP. For the focal codeblock, the K of differences in the NZBP between the focal codeblock and K of its surrounding codeblocks are firstly derived in the query and database images:

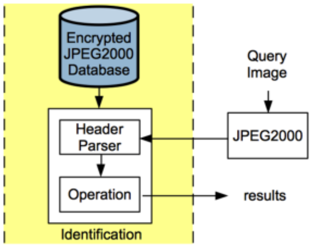


Fig.16: *NZBP-based identification of JPEG 2000 image in the encrypted domain.*

Table 1: *Specifications for standard evaluation material.*

Number of frames	14964
Frame rate	24 frames/sec
Spatial resolution	4096 × 1740 pixels
Color format	RGB(4:4:4), 12 bits/component

$$d^I_{c,k} = z^I_c - z^I_{c,k}, \tag{14}$$

where z^I_c is the NZBP of the c-th codeblock in image $I \in \{\text{query}, \text{database}\}$, $z^I_{c,k}$ represents that of the k-th surrounding codeblock ($k = 0, 1, \ldots, K-1$), and $d^I_{c,k}$ is the difference. Then, the positive and negative sign of $d^I_{c,k}$ given as

$$e^I_{c,k} = \begin{cases} 1, & d^I_{c,k} > 0 \\ 0, & d^I_{c,k} = 0 \\ -1, & d^I_{c,k} < 0 \end{cases}, \tag{15}$$

where $e^I_{c,k}$ is the sign of $d^I_{c,k}$. If $e^{\text{query}}_{c,k}$ and $e^{\text{database}}_{c,k}$ are the same for all c and k, the query image is identical to the image in the database. It is noted that $d^I_{c,k} = 0$ when either z^I_c or $z^I_{c,k}$ is “not included.”

4.3 Experimental Results

By using a total of 14964 frames out of 17239 (the excluded frames are fully black) from the standard evaluation material (StEM) DCI standard test sequences [26], c.f., Table 1, the described identification is evaluated. Table 2 summarizes the conditions, and the experiments of identification were performed for all possible combinations of a query and a database image. So, the total of the number of combinations was 14964 × 14964.

The false positive rate (FPR) and true positive rate (TPR) of all trials are shown in Table 3, where

$$\text{FPR} = \text{FP}/(\text{FP} + \text{TN}) \tag{16}$$

$$\text{TPR} = \text{TP}/(\text{TP} + \text{FN}), \tag{17}$$

and FP, TN, TP, and FN are the number of false positive, true negative, true positive, and false negative, respectively. Table 3 shows that the described identification of JPEG 2000 images in the encrypted domain produced an under 1.0 % FPR regardless of the

Table 2: *Conditions.*

(a)For query images.

DWT filter	9×7, 5×3
DWT level	5, 4
Base step size	1/256, 1/200
Codeblock size	32×32, 64×64, 128×32

(b) For database images.

	DWT filter	DWT level	Base step size	Codeblock size
DWT53	5×3	5	1/256	32×32
CB64	9×7	5	1/256	64×64
CB128	9×7	5	1/256	128×32
Res	9×7	4	1/256	32×32
Qstep	9×7	5	1/200	32×32

Table 3: *False positive rate and true positive rate given by Eqs.* (16) *and* (17).

		FPR (%)	TPR (%)
DWT53	$K = 4$	0.93	**100**
	$K = 8$	0.84	**100**
CB64		0.55	**100**
CB128		0.60	**100**
Res		0.41	**100**
Qstep	$K = 4$	0.79	**100**
	$K = 8$	0.72	**100**

difference of JPEG 2000 coding parameters. Moreover, it is noteworthy that no false negative match was produced by the described identification, i.e., any frame can be exactly identified even in the encrypted domain. In addition, the average processing time was about 0.6 msec/frame excluding disk accessing time on a Windows XP workstation with a Xeon 2.5 GHz processor and 4 GiB memory.

5. CONCLUSIONS

This paper describes about signal and image processing in the encrypted domain including a brief overview and two tangible examples. Signal and image processing in the encrypted domain still has challengeable problems and requires new approach and algorithms. Many participants are expected to join the development in this field to build trustworthy applications and services for our rich lives.

References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *Commun. ACM*, vol.53, no.4, pp.50–58, 2010.

[2] P. Mell and T. Grance, “The NIST definition of cloud computing,” *NIST Special Publication* 800-145, 2011.

[3] N. Ahituv, Y. Lapid, and S. Neumann, “Processing encrypted data,” *Commun. ACM*, vol.30, no.9, pp.777–780, 1987.

[4] SPEED Project, `http://www.speedproject.eu/`

[5] A. Piva and S. Katzenbeisser, eds., “Special issue on signal processing in the encrypted domain,” *EURASIP J. Inform. Security*, vol.2007, 2007.

[6] SPEED Workshop (in conjunction with European Symposium on Research in Computer Security), 2007.

[7] International Workshop on Signal Processing in the EncryptEd Domain, 2009.

[8] M. Barni and A. Piva, co-chaired, “Special session on signal processing in the encrypted domain,” *Proc. EURASIP EUSIPCO*, 2008.

[9] S. Rane and M. Barni, co-chaired, “Special session on secure signal processing,” *Proc. IEEE ICASSP*, pp.5848–5871, 2011.

[10] R.L. Lagendijk, “Secure signal processing: merging the worlds of signal processing and cryptography,” a Keynote speech, *IEEE MMSP*, 2009.

[11] T. Bianchi, A. Piva, and M. Barni, “Efficient pointwise and blockwise encrypted operations,” *Proc. ACM MMSec*, pp.85–90, 2008.

[12] T. Bianchi, A. Piva, and M. Barni, “Efficient linear filtering of encrypted signals via composite representation,” *Proc. IEEE DSP*, 2009.

[13] T. Bianchi, P.J.M. Veugen, A. Piva, and M. Barni, “Processing in the encrypted domain using a composite signal representation: Pros and cons,” *Proc. IEEE WIFS*, pp.176–180, 2009.

[14] T. Bianchi, A. Piva, and M. Barni, “Comparison of different FFT implementations in the encrypted domain,” *Proc. EURASIP EUSIPCO*, 2008.

[15] T. Bianchi, A. Piva, and M. Barni, “Implementing the discrete Fourier transform in the encrypted domain,” *Proc. IEEE ICASSP*, pp.1757–1760, 2008.

[16] T. Bianchi, A. Piva, and M. Barni, “On the implementation of the discrete Fourier transform in the encrypted domain,” *IEEE Trans. Inform. Forensics and Security*, vol.4, no.1, pp.86–97, 2009.

[17] I. Ito and H. Kiya, “DCT sign-only correlation with application to image matching and the relationship with phase-only correlation,” *Proc. IEEE ICASSP*, pp.I-1237–I-1240, 2007.

[18] H. Kiya and I. Ito, “Image matching between scrambled images for secure data management,” *Proc. EURASIP EUSIPCO*, 2008.

[19] I. Ito and H. Kiya, “A new class of image registration for guaranteeing secure data management,” *Proc. IEEE ICIP*, pp.269–272, 2008.

[20] H. Kiya and I. Ito, “Phase scrambling for image matching in the scrambled domain,” *Signal Processing*, S. Miron, ed., pp.397–414, InTech, 2010.

[21] O. Watanabe, T. Iida, T. Fukuhara, and H. Kiya, "Identification of JPEG 2000 images in encrypted domain for digital cinema," *Proc. IEEE ICIP*, 2009.

[22] T. Fukuhara, K. Hosaka, and H. Kiya, "Identifying method of JPEG2000 images in the code stream level for digital cinema," *IEICE Trans. Inform. Syst. (Japanese edition)*, vol.J91-D, no.9, pp.2305–2313, 2008.

[23] O. Watanabe, T. Fukuhara, and H. Kiya, "Fast identification of JPEG 2000 images for digital cinema profiles," *Proc. IEEE ICASSP*, 2011.

[24] O. Watanabe, T. Fukuhara, and H. Kiya, "Fast accurate identifying method of JPEG 2000 images with different coding parameters for digital cinema," *Proc. APSIPA ASC*, pp.298–301, 2010.

[25] O. Watanabe, T. Fukuhara, and H. Kiya, "Codestream-based identification of JPEG 2000 images with different coding parameters," *IEICE Trans. Inf.&Sys.*, vol.E95-D, no.4, pp.1120–1129, 2012.

[26] Digital Cinema Initiative, LLC Technology Committee, "StEM access procedures," http://www.dcimovies.com/StEM/, 2010.

Hitoshi Kiya is a Professor of the Department of Information and Communication Systems and a Board Member of the Faculty of System Design at Tokyo Metropolitan University, Japan. He also served as the chair of the Department of Information and Communications System. He received his B.E. and M.E. degrees from Nagaoka University of Technology, Japan, in 1980 and 1982 respectively, and a Dr.Eng. degree from Tokyo Metropolitan University in 1987, all in electrical engineering. In 1982, he joined Tokyo Metropolitan University as an Assistant Professor, where he became a Full Professor in 2000. He is also a collaborative researcher at Tohoku University, Japan. From 1995 to 1996, he attended the University of Sydney, NSW, Australia as a Visiting Fellow.

His research interests are in the areas of multirate signal processing and image processing including wavelets, video coding, compressed-domain video manipulation and security for multimedia. In these areas, he has published over 300 refereed papers in leading international conferences and journals. He authored seven books and co-authored three books. He also holds ten patents in the US and Japan. He received the Telecommunications Advancement Foundation Award in 2011, the IEICE Engineering Science Society Contribution Award in 2010 and the IEICE Best Paper Award in 2008. He is a Fellow of the IEICE and the ITE, a senior member of the IEEE, and a member of the APSIPA and the EURASIP.

Masaaki Fujiyoshi is an Assistant Professor of the Department of Information and Communication Systems at Tokyo Metropolitan University, Japan. He received his B.Arts, M.Eng., and Ph.D. degrees from Saitama University, Japan, in 1995, 1997, and 2001, respectively. In 2001, he joined Tokyo Metropolitan University as a Research Associate of the Department of Electrical Engineering. His research interests include image processing and secure communication. He received the IEICE Young Researchers Award in 2001. He is a member of the IEEE, the APSIPA, the IEICE, and the ITE.