JPSEC for Secure Imaging in JPEG 2000

Frederic Dufaux [a,b], Susie Wee [c], John Apostolopoulos [c] and Touradj Ebrahimi [a,b]

[a] Signal Processing Institute, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland
Frederic.Dufaux@epfl.ch, Touradj.Ebrahimi@epfl.ch

[b] Emitall S.A., Rue du Théâtre 5, 1820 Montreux, Switzerland
Frederic.Dufaux@emitall.com, Touradj.Ebrahimi@emitall.com

[c] Multimedia Communications and Networking Dept., HP Labs, Palo Alto, CA, USA
swee@hpl.hp.com, japos@hpl.hp.com

ABSTRACT

In this paper, we first review the on-going JPSEC standardization activity. Its goal is to extend the baseline JPEG 2000 specification to provide a standardized framework for secure imaging, in order to support tools needed to secure digital images, such as content protection, data integrity check, authentication, and conditional access control. We then present two examples of JPSEC tools. The first one is a technique for secure scalable streaming and secure transcoding. It allows the protected JPSEC codestream to be transcoded while preserving the protection, i.e. without requiring unprotecting (e.g. decrypting) the codestream. The second one is a technique for conditional access control. It can be used for access control by resolution or quality, but also by regions of interest.

Keywords: JPEG 2000, JPSEC, secure imaging, encryption, secure transcoding, scrambling

1. INTRODUCTION

The fast-paced progress in computer technologies combined with the availability of powerful imaging software has made digital imaging a tremendous success with consumers. JPEG 2000 is the latest standard for still image coding [1][2]. On top of a very efficient image compression scheme, JPEG 2000 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 a major issue in many imaging applications, JPEG has initiated a work item known as Secure JPEG 2000 or JPSEC. The purpose of this activity is to provide with a framework for secure imaging. This paper presents an overview of the current status of JPSEC [3]. We then describe two examples of JPSEC applications.

The first example describes a technique for secure scalable streaming and secure transcoding [4][5]. A media delivery system may need to deliver a media stream to a number of clients with diverse device capabilities and connection qualities. This may require mid-network nodes, or proxies, to perform stream adaptation, or transcoding, to adapt streams for downstream client capabilities and time-varying network conditions. This example describes a JPSEC application that provides protection of JPEG 2000 codestreams while maintaining a level of scalability and transcodability in the JPSEC codestream. This allows the protected JPSEC codestream to be transcoded while preserving the protection, i.e. without requiring unprotecting the codestream.

The second example presents a technique for conditional access control [6]. For instance, to browse a large database of images, it is desirable to give free access to a small resolution thumbnail while access to higher resolutions is subject to authorization. Similarly, one can give free access to a low quality version of an image, while the higher qualities are protected. The technique described in this example basically adds a pseudo-random noise to the codestream. Authorized users know the pseudo-random sequence and can therefore remove the noise. Conversely, the decoded image appears

severely distorted for an authorized user who does not know how to remove the noise. The scrambling is selectively applied on the code-blocks composing the JPEG 2000 codestream. Consequently, the distortion level introduced in specific parts of the image can be controlled, enabling access control by resolution, quality or regions of interest. In [7], this conditional access control technique is applied in a video surveillance system to protect regions corresponding to people, while the remaining of the scene is clear, hence preserving the anonymity of the people under surveillance.

Other examples of JPSEC applications and tools are presented in [8][9][10].

This paper is structured as follow. In Sec. 2, we first give an overview of the current status of JPSEC. The technique for secure scalable streaming and secure transcoding is presented in Sec. 3. The technique for conditional access control is described in Sec. 4. Finally, we draw some conclusions in Sec. 5.

2. JPSEC OVERVIEW

The Joint Photographic Experts Group (JPEG) has recently completed a new still image coding standard called JPEG 2000, approved as an International Standard in December 2000. The baseline version, more formally referred to as part 1 of the JPEG 2000 specifications, integrates an efficient image compression scheme along with new features required by multimedia applications, such as progressive coding up to lossless, scalability, region of interest coding, and error resilience. A more detailed description of JPEG 2000 can be found in [1][2].

Recognizing that security is an important concern in many applications, JPEG recently initiated an activity known as Secure JPEG 2000 or JPSEC, formally referred to as part 8 of the JPEG 2000 specifications. Its goal is to extend the baseline specifications to provide a standardized framework for secure imaging. This framework enables the efficient integration and use of the tools needed to secure digital images, such as content protection, data integrity check, authentication, and conditional access control. The framework is open and flexible, hence ensuring a straight path for future extensions.

In this section, we review the current status of the JPSEC specifications. At the time of this writing, the JPSEC standard has reached the Committee Draft stage [3]. While in a stable phase, it is still subject to change.

2.1. Scope

JPSEC enables the use of security tools supporting a number of security services. Currently, the following security services are considered:

- **Confidentiality:** transformation of the image data (and/or the associated metadata) into an encrypted/ciphered form that conceals its original content. This includes selective encryption meaning that only parts of the image data are encrypted.

- **Integrity verification:** detection of manipulations to the image data (and/or the associated metadata) to verify its integrity. The integrity verification may also identify locations in the image data where the integrity is in doubt. There are two classes of integrity verification: image data integrity verification and image content integrity verification. In the first case, a bit exact verification of the image data is performed. In the second case, some minor alteration of image data results in success of the integrity verification as long as the alteration does not change the perceptual meaning of the image content. Integrity verification includes cryptographic methods such as Message Authentication Codes (MAC), digital signatures, cryptographic checksums or keyed hash, and watermarking.

- **Source authentication:** verification of the identity of a user/party which generated a JPSEC stream. This includes methods such as digital signatures or Message Authentication Code (MAC).

- **Conditional access:** mechanisms to grant or restrict access to image data or parts of it. This allows for instance to view a low resolution (preview) of an image without being able to visualize a higher resolution.

- **Secure scalable streaming and secure transcoding:** methods such that a node can perform streaming and transcoding of JPSEC stream without requiring decryption or unprotecting the content. An example is the case

where protected JPEG 2000 content is streamed to a mid-network node or proxy that in turn transcodes the protected JPEG 2000 content in a manner that preserves end-to-end security.

- **Registered content identification:** registration of a JPSEC stream with a Registration Authority. This includes the matching of a claimed image data to the registered image data. For example, this could involve reading a file identifier (Licence Plate) placed inside the image metadata, and checking the coherence between this Licence Plate and the information that has been uploaded when the registration process was done in order to verify that the file corresponds to the identifier.

2.2. JPSEC framework

JPSEC defines an open and flexible framework for secure imaging as illustrated in Figure 1. A JPSEC protector application provides a number of security services (e.g. confidentiality, integrity verification, source authentication,…). In order to secure an image, it applies one or more JPSEC protection tools (e.g. encryption, digital signature,…). The resulting JPSEC codestream is generated by inserting in the stream the corresponding JPSEC syntax, signaling the JPSEC tools which have been used and how they have been applied to the image.

Conversely, in order to unprotect and consume a JPSEC codestream, a JPSEC unprotector application applies the corresponding JPSEC unprotection tools (e.g. decryption, digital signature verification,…). To do so, it has to parse and interpret the JPSEC syntax in order to identify the security services associated with the image data and the corresponding JPSEC tools required to process it.

The JPSEC tools fall into two categories. The first category encompasses well-known cryptographic methods such as AES, DES, 3DES, RC4, RSA, MD5, SHA-1, etc… In this case, a number of templates are defined in order to specify method specific parameters. The tools in this category are therefore referred to as template protection tools. The syntax contains all the required information relative to the protection tool and how it has been applied. It is therefore sufficient to enable a JPSEC application to unprotect the image data. The second category consists of proprietary tools. These tools have to be registered with the JPSEC Registration Authority (RA), they are then referred to as registration authority protection tools. Upon registration, a tool is assigned a unique identification number. In this case, the syntax contains the unique identification number along with private parameters. A JPSEC application may have to query the JPSEC RA in order to get a description of the tool and be able to unprotect the image data. With this registration process, provision is made for future tools to be identified and registered.

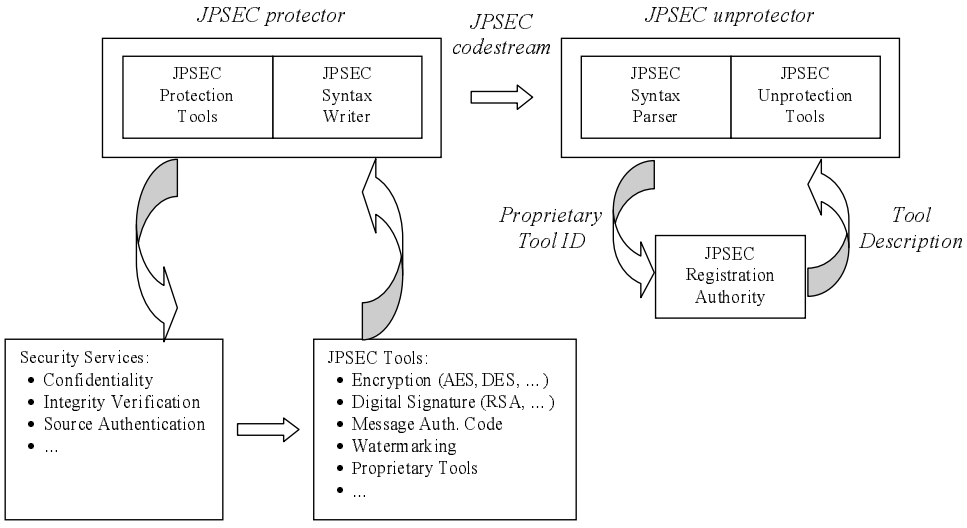

Figure 1 – JPSEC framework.

A JPSEC codestream can be created and consumed in a number of ways. The encoding and protection can be performed simultaneously, or the security tools can be applied directly on a previously encoded JPEG 2000 codestream. Multiple security tools can also be successively applied to the content, e.g. the content is signed and encrypted in order to both protect the content and guarantee its integrity. In this last case, the JPSEC codestream is therefore generated from another JPSEC codestream.

2.3. Normative and informative parts

JPSEC defines a normative codestream syntax which specifies the information required for interpreting a secure image data. More specifically, the syntax signals which security services are associated with the image data, which JPSEC tools are required in order to fulfill the corresponding services, and which parts of the image data are protected. JPSEC also defines a normative process for registering security tools with the JPSEC RA.

The specific tools to secure an image are out of the scope of JPSEC. Instead, JPSEC gives informative examples of security tools in typical use cases, along with guidelines on how to implement security services.

A JPSEC compliant application is one that is able to consume JPSEC codestreams. For this purpose, it has to implement one or more JPSEC tools fulfilling the security services targeted by the application.

2.4. Syntax

In this section, we present the current JPSEC syntax. JPSEC has introduced two new marker segments, SEC and INSEC.

The SEC marker segment is present in the Main Header and is mandatory. It gives overall information about the security tools which have been applied to secure the image. More specifically, SEC indicates the JPSEC tools used to secure the image, along with some parameters referring to the technique used. Among other things, these parameters can indicate which parts of the codestream have been secured. The syntax of the SEC marker segment is illustrated in Figure 2.

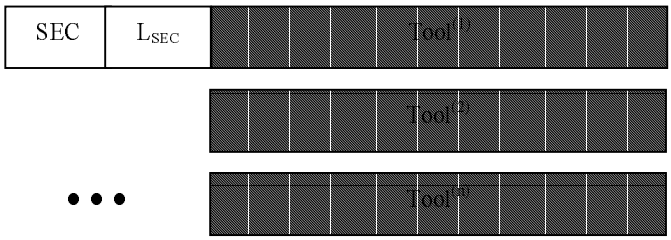

Figure 2 – SEC marker segment syntax.

SEC is a unique marker which unequivocally identifies the SEC marker segment, its value is 0xFF94. L_{SEC} is the length of the marker segment in bytes. Finally, $Tool^{(i)}$ contains the information about the protection tool i.

As mentioned earlier, there are two categories of protection tools: template protection tools which are specified with a template and registration authority protection tools which are specified by an identification number issued by the JPSEC RA. Figure 3 and Figure 4 show the syntax for these two types of tools respectively.

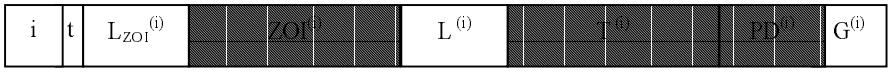

Figure 3 – Template protection tool syntax (t=0).

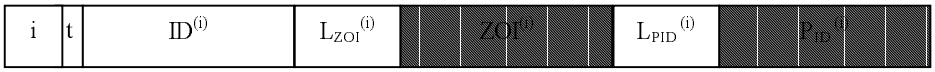

Figure 4 – Registration authority protection tool syntax (t=1).

In both cases, the meaning of the first byte is identical. The 7 most significant bits is a counter representing the instance index i of the protection tool. It can be used as a unique identifier of the tool instance. The least significant bit signals the tool type t where 0 identifies a template protection tool and 1 a registration authority template tool.

$ZOI^{(i)}$ specifies the zone of influence of the protection tool i, and $L_{ZOI}^{(i)}$ is the length of $L_{ZOI}^{(i)} + ZOI^{(i)}$ in bytes. The zone of influence can be expressed in image-related parameters, e.g. resolution levels, components, quality layers, regions of interest, or in non image-related parameters, e.g. bitstream byte ranges, packet indices. In addition, image-related parameters and non-image-related parameters can be used together in one ZOI to describe a *correspondence* between image features and bitstream areas. For example, the ZOI can specify that a protection tool is applied to bytes 0 through 2078 of the codestream data and that the protected data contains the lowest resolution component of the image. This ability to represent correspondences between image-related and bitstream-related parameters is a key feature of the ZOI. This feature enables low-complexity transcoding to be performed without decryption by representing the correspondence of bits or packets to image features, even when the data itself is encrypted.

In the syntax for template protection tools, $T^{(i)}$ is the protection method template parameters for protection tool i, $PD^{(i)}$ the processing domain, and $G^{(i)}$ the granularity, with $L^{(i)}$ the length of $L^{(i)} + T^{(i)} + PD^{(i)} + G^{(i)}$ in bytes. A number of templates are defined for methods such as decryption, authentication and integrity. These templates allow specifying a number of parameters which are relevant to the protection tool. Processing domain indicates in which domain the protection method is applied, e.g. wavelet coefficient domain, quantized wavelet coefficient domain, bitstream domain and packet domain. Granularity indicates the basic unit of protection for the protection method, for instance a tile, a component, a resolution level, a layer, etc…

Similarly, in the syntax for registration authority protection tools, $ID^{(i)}$ is the identification number issued by the JPSEC RA for protection tool i, $P_{ID}^{(i)}$ contains parameters for protection tool i, and $L_{PID}^{(i)}$ is the length of $L_{PID}^{(i)} + P_{ID}^{(i)}$ in bytes.

The INSEC marker segment provides with an additional mean to transmit parameters for one of the security tools declared in SEC, in order to complement the information in Main Header. It can be placed anywhere in the codestream data and is optional. It uses the fact that the arithmetic decoder in JPEG 2000 stops reading bytes when it encounters a termination marker (i.e. two bytes with a value greater than 0xFF8F). This interesting feature can be used to add extra bytes in the codestream without affecting the syntax compliance. The INSEC syntax is given in Figure 5.

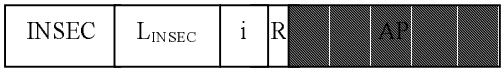

Figure 5 – INSEC marker segment syntax.

INSEC is a unique marker which unequivocally identifies the INSEC marker segment, its value is 0xFF95. L_{INSEC} is the length of the marker segment in bytes. The tool instance index i is 7 bits long, and identifies the tool instance declared in SEC for which INSEC is giving additional or alternative parameters. R is a one bit flag which signals whether the information carried in the INSEC marker segment is relevant to the preceding or following code-blocks. Finally, AP contains additional or alternative parameters for protection tool i.

3. SECURE SCALABLE STREAMING AND SECURE TRANSCODING

This section describes a method for protecting JPEG 2000 codestreams while enabling secure transcoding. Specifically, this section addresses encryption and authentication using JPSEC. We describe a JPSEC usage that provides privacy protection of JPEG 2000 codestreams while maintaining a level of scalability and transcodability [4][5]. This allows the resulting JPSEC codestream to be securely transcoded, i.e., *transcoded without decryption.* Furthermore, this usage describes how authentication can be applied so that receivers can verify that transcoding was performed in a valid and permissible manner. We also show how the JPSEC syntax can be used to signal these methods.

3.1. System description

We first describe a use case for privacy protection with JPEG 2000. A sender encrypts the JPEG 2000 content into a JPSEC codestream using a prescribed key. Allowed receivers can decode the JPSEC codestream using an appropriate key. If the client capabilities and network conditions permit, the content can be received and rendered in its full, original quality. Otherwise, the content may need to be adapted to match the client capabilities, e.g., by lowering the resolution, and the network conditions, e.g., by lowering the bit rate.

When sharing unprotected JPEG 2000 streams in this scenario, the solution is quite straightforward because of the inherent scalability properties that JPEG 2000 was designed to provide. Specifically, the JPEG 2000 standard uses structures such as tiles, precincts, resolution levels, quality layers, and color components and provides a syntax that allows easy access to these various components. For example, reducing the resolution simply requires extracting a subset of resolution levels. Likewise, reducing the bit rate simply requires extracting a subset of quality layers. Generally speaking, adapting or transcoding JPEG 2000 codestreams is a straightforward operation and results in rather straightforward implementations of mid-network transcoders. The scalability property of the JPEG 2000 codestream is one of the key differentiation features provided by the standard.

JPSEC protection methods may involve encrypting parts of the JPEG 2000 codestream. However, encryption can reduce or even destroy the scalability properties of the codestream. In this case, transcoding would require decryption and create a potential security threat. Thus, when designing JPSEC protection methods, careful consideration should be given for providing mechanisms for retaining various levels of scalability within the protected codestream.

Specifically, this should be done in a way that allows protected streams to be scaled or transcoded without decrypting the content. This allows mid-network nodes to perform this operation without requiring the key, and therefore without breaching the end-to-end security of the system. The ability to scale or transcode a protected stream without decryption is called *secure transcoding* and is a design goal for JPSEC. We present JPSEC protection methods that enable secure transcoding by allowing various levels of scalability and transcodability to be retained in the JPSEC codestream [4].

A JPSEC secure transcoding system is shown in Figure 6. An original image is encoded into a JPEG 2000 codestream. This is then locked with a key to form the JPSEC codestream. The JPSEC codestream can be unlocked back to a JPEG 2000 codestream by authorized entities using the appropriate key. This JPEG 2000 codestream can be decoded to reconstruct the image. The JPSEC codestream can also be securely transcoded or scaled to a transcoded JPSEC stream with a secure transcoding operation that does not require the key. The resulting transcoded JPSEC codestream can be unlocked by authorized entities using the key to form a transcoded JPEG 2000 codestream, which can then be decoded to reconstruct the image at a lower scale.

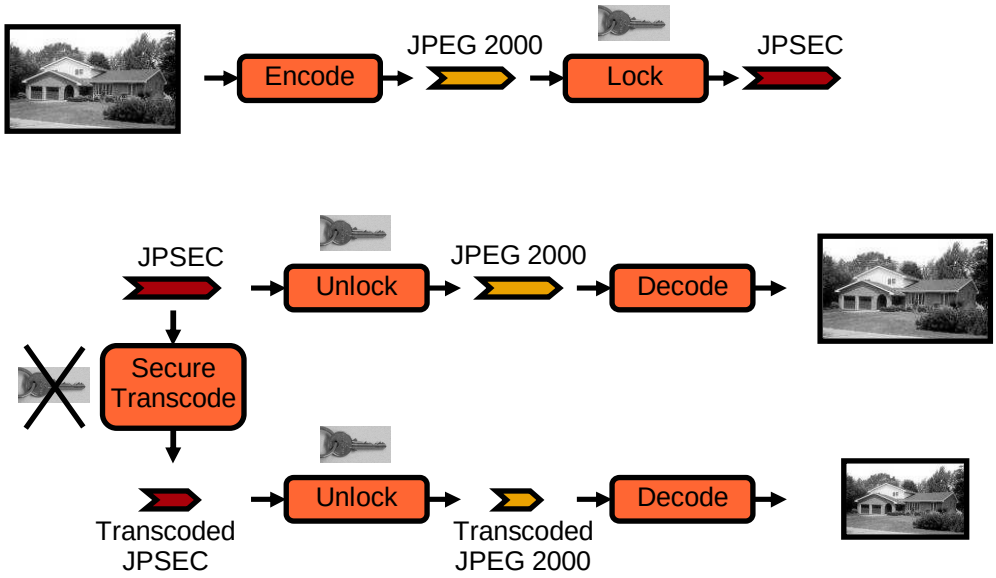

Figure 6 – Secure transcoding system diagram.

In this example, the lock operation consists of two parts: encryption and authentication. The unlock operation also consists of two parts: verify authentication and decryption. A number of encryption/decryption methods can be used, where the selected method can be signaled with the protection method decryption template. Similarly, a number of authentication methods can be used, where the selected method can be signaled using the protection method authentication template. Furthermore, encryption and authentication can be performed in a number of ways, and the

specific manner in which each is applied can be specified by the image-based and non-image based description classes, including the processing domain and granularity.

Confidentiality may be provided by an encryption cipher, which may in principle include any of the JPSEC compliant block or stream ciphers. Illustrative examples of ciphers which can be used include the AES block cipher in Cipher Block Chaining (CBC) and Counter (CTR) modes, or RC4 stream cipher. Of course, other ciphers can also be used. Authentication may be provided using a number of authentication tools. As an illustrative example, the authentication can be performed using HMAC with SHA-1. Of course, other authentication methods can also be used.

Note that the encryption may be performed first, followed by authentication of the ciphertext. Or alternatively the authentication may be performed on the plaintext, followed by encryption. Both approaches have merits, and disadvantages, and the preferred approach often depends on the final application. JPSEC supports both orderings, and secure scalable streaming and secure transcoding can be performed with both. Generally, separate keys should be used for encryption and authentication. Of course, these keys may be related to a single master key, and various other key management strategies can be used.

The most significant aspect of this system is that the secure transcoding operation can be performed by a (potentially untrusted) stand-alone entity that does not have the key. Furthermore, receivers can perform authentication to verify that the received stream was transcoded in a valid and permissible manner.

3.2. JPSEC syntax

The JPSEC syntax can be used to create a secure transcoding system. This is achieved by using various encryption and authentication methods in conjunction with appropriate signaling. Specifically, the Zone of Influence (ZOI) parameters can signal information or metadata that might be difficult or impossible to infer because of the encryption. This information is placed in the SEC marker segment and can be used by transcoding nodes to perform secure transcoding.

One method for achieving secure transcoding is to protect JPEG 2000 codestreams by only encrypting JPEG 2000 packet bodies. This can be signaled in the SEC marker segment by specifying the ZOI as the entire image, the processing domain as the bitstream data, and the granularity of the encryption as the packet bodies. Since the JPEG 2000 header information and packet headers are left unencrypted, the JPSEC codestream can be almost fully manipulated using typical JPEG 2000 parsing. Its only limitation would be that it could not access the image data itself. This method provides full transcodability of the protected stream. However, the transcoding operation is relatively complex in that it does require detailed knowledge of the JPEG 2000 syntax and the encryption and decryption operations are relatively complex in that they require extracting and recombining packet headers and bodies.

Another class of methods for achieving secure transcoding use *segment-based encryption with metadata.* In this example, segments of data are encrypted together across packet headers and bodies, and metadata that describes this operation is signaled in the SEC marker segment. We now describe a usage example where the original JPEG 2000 codestream is in RLCP ordering and the goal is to protect this stream with encryption a nd authentication while enabling secure transcoding by resolution on the protected codestream. Figure 7 shows the syntax of a JPEG 2000 and JPSEC codestream. Since the original JPEG 2000 codestream used an RLCP ordering, each resolution component is represented by a contiguous data segment.

The specific syntax can be signaled in JPSEC using the template protection tool shown in Figure 3. The zone of influence, decryption template, processing domain, and granularity fields are detailed in Table 1. The ZOI specifies three zones, each uses image and non-image related parameters to specify the resolution level and corresponding byte range. The decryption template, processing domain, and granularity specify the decryption method and how it should be applied. Furthermore, authentication can be performed on each of the three data segments, either before or after encryption depending on the desired functionality. The resulting MAC values can be signaled in the SEC marker segment using the authentication template. An authorized JPSEC consumer with the appropriate key can use this information to decrypt and authenticate the JPSEC stream.

In order to perform secure transcoding on the JPSEC codestream, a transcoder simply reads and parses the SEC marker segment, identifies the locations of the resolution segments, and then retains or removes the appropriate data segments/resolutions. Notice that this transcoding operation corresponds to a simple parsing operation and does not

require unprotecting the data. Therefore, secure transcoding-by-resolution is achieved. Authentication is performed as well by authenticating the received transcoded data with the MAC values that are placed in the SEC marker segment during the JPSEC protection process.

In this example, the coded image data was ordered in contiguous transcodable units. If this is not the case, then the JPEG 2000 codestream can first be reordered into the desired ordering. This reordering can be signaled by transcoding the JPEG 2000 codestream (e.g., from CPRL to RLCP) or by using JPSEC signaling to represent the reordering.

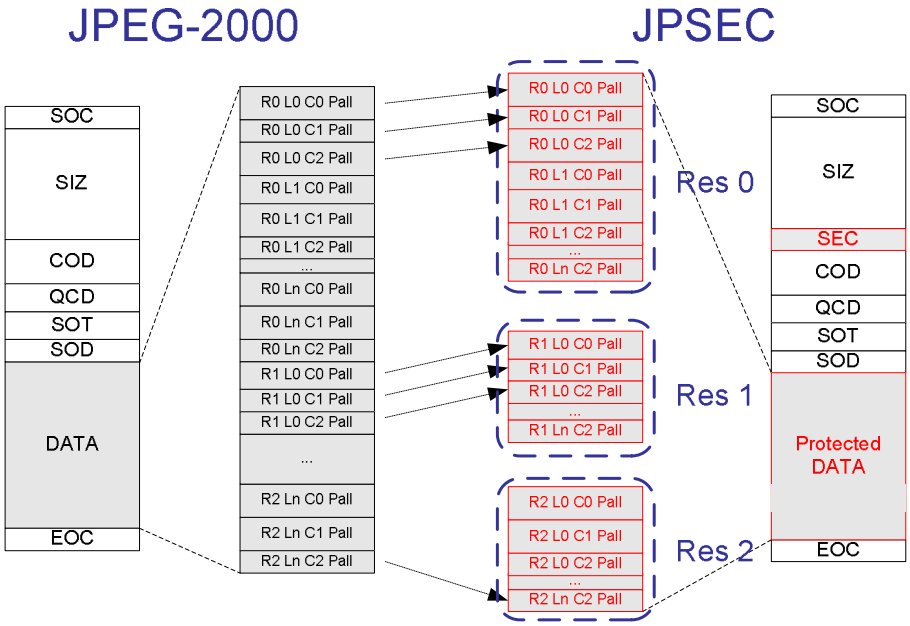

Figure 7 – Forming a JPSEC codestream from a JPEG 2000 codestream.

Table 1 – JPSEC signaling syntax for secure scalable streaming and secure transcoding.

Parameter		Derived meaning
NZzoi		Number of Zone is three.
Zone 0	DCzoi	Image description class, Resolution level.
		Non-image description class, Byte range.
	Pzoi[1]	Resolution level 0.
	Pzoi[2]	Byte range 0-1000
Zone 1	DCzoi	Image description class, Resolution level.
		Non-image description class, Byte range.
	Pzoi[1]	Resolution level 1.
	Pzoi[2]	Byte range 1000-2356.
Zone 2	DCzoi	Image description class, Resolution level.
		Non-image description class, Byte range.
	Pzoi[1]	Resolution level 2.
	Pzoi[2]	Byte range 2356-2998.

Parameter			Derived meaning
L			Length of T + PD + G.
T	PM_{ID}		Decryption template.
	ME_{decry}		Marker emulation is NULL.
	CI_{decry}		AES encryption
	CP_{decry}	M_b	CTR and nonce.
		P_{bs}	Padding not used in CTR
		SI Z_{bs}	Block size is 128.
		K_{bs}	Key size is 128.
PD			Bitstream-domain processing.
			Reserved
G			Packet headers are not protected.
			Units are ZOI zones.

4. SCRAMBLING FOR CONDITIONAL ACCESS CONTROL

In this section, a technique for conditional access control is presented. The method was initially proposed in [6]. Conditional access control is often a desirable functionality in imaging applications. For instance, free access to a low resolution thumbnail can be granted, while access to higher resolutions is subject to authorization. This allows browsing a large database of images. In a similar way, a highly compress low quality version of an image can be freely accessible, while the higher qualities are protected. In [7], conditional access control is used to preserve privacy in a video surveillance system. More specifically, the system protects regions of interest, e.g. corresponding to people. Henceforth, the scene is clear, but people under surveillance cannot be recognized unless an authorization is granted.

Basically, this technique adds a pseudo-random noise to selected parts of the codestream. Authorized users know the pseudo-random sequence and can therefore remove this noise. Conversely, unauthorized users do not know how to remove this noise and consequently have only access to a severely distorted image.

The system is composed of three main components: scrambling, pseudo-random number generator and encryption algorithm. In order to fully exploit and retain the properties of JPEG 2000, the scrambling is selectively applied on the code-blocks composing the codestream. Consequently, the distortion level introduced in specific parts of the image can be controlled. This allows for access control by resolution, quality or regions of interest in an image. In addition, several levels of access can be defined by using different encryption keys.

4.1. System description

The system is composed of three main components: scrambling, pseudo-random number generator and encryption which are discussed in more details hereafter.

Two approaches are proposed for scrambling. The scrambling is either performed on the wavelet coefficients or directly on the codestream, as illustrated in Figure 8 and Figure 9 respectively.

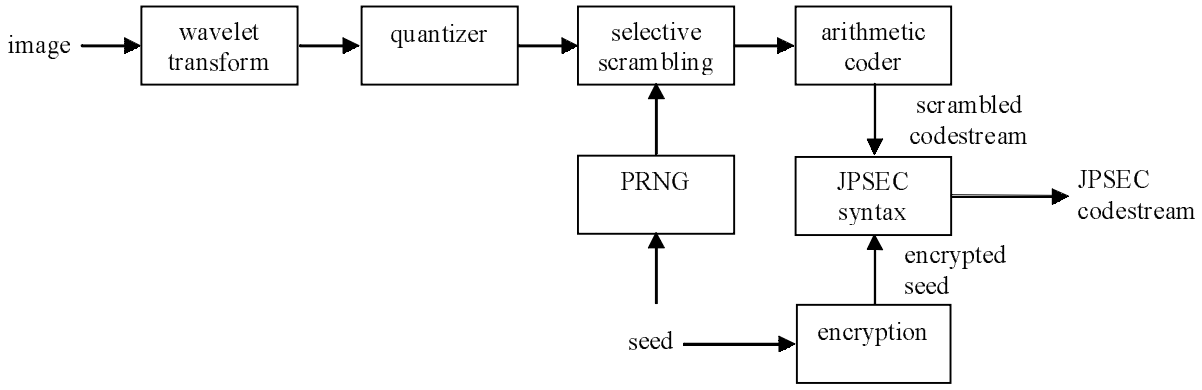

Figure 8 – Block diagram for wavelet domain scrambling.

In this first approach (see Figure 8), a known noise is added to selected quantized wavelet coefficients. For instance, in the case of access control by resolution, the scrambling is applied to the wavelet coefficients corresponding to high resolution levels. In order to have a very simple algorithm, the signs of the coefficients in each code-block are inverted pseudo-randomly. Note that this method modifies only the most significant bit-plane of the coefficients and can be performed on-the-fly during the entropy coding. The sign flipping takes place as follows. For each coefficient, a new pseudo-random value is generated and compared with a density threshold. If the pseudo-random value is greater than the threshold, the sign is inverted; otherwise the sign is left unchanged.

In the second approach (see Figure 9), scrambling is directly performed on the bitstream. Indeed, the slightest bit modification in the codestream usually leads to a totally different output from the arithmetic coder. For instance, in the case of conditional access by resolution, the part of the codestream corresponding to the high resolutions is scrambled. Alternatively, in the case of access control by quality, the scrambling is applied to the code-blocks belonging to high quality layers.

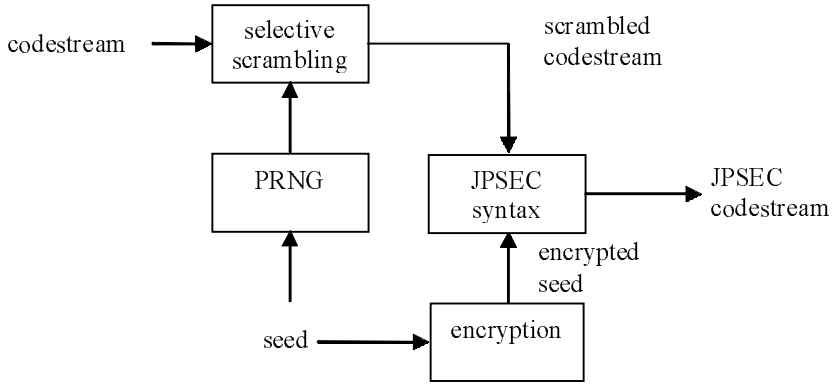

Figure 9 – Block diagram for bitstream domain scrambling.

Note that the scrambling should preserve the markers in the bitstream (i.e. two bytes with a value greater than 0xFF8F), and should not introduce any erroneous markers in the bitstream. For this purpose, the scrambling procedure is as follow:

1. if the byte is 0xFF, no modification
2. if the preceding byte is 0xFF, the current byte x is replaced by:
   - x'=(x+m) modulo 0x90, if x < 0x90, where m is an 8-bit pseudo-random number in the range [0x00, 0x8F]
   - no modification, if x >= 0x90
3. otherwise, the current byte x is replaced by x'=(x+n) modulo 0xFF, where n is an 8-bit pseudo-random number [0x00, 0xFE]

The following algorithm is used to obtain an 8-bit pseudo-random number n uniformly distributed in the [0,N] range:

do
n = 8bit pseudo-random number
while (n>N)

The scrambling process is driven by a pseudo-random number generator (PRNG). In the proposed implementation, the SHA1PRNG algorithm [11] with a 64-bit seed is used. To communicate the seeds values to authorized users, they are encrypted and inserted in the JPSEC codestream. In our implementation, the RSA algorithm is used for encryption [12]. Note that other PRNG or encryption algorithms could be used as well. The length of the key can be selected at the time the image is protected.

In order to improve the security of the system, the seed can be frequently changed. Also, several levels of access can be defined, using different encryption keys.

4.2. JPSEC syntax

We know describe the JPSEC syntax used with the proposed conditional access technique. In this example, both the SEC and INSEC marker segments are used. The SEC marker segment contains overall information about how the tool has been applied, namely the number of seeds used, an identification of the encryption key and the encrypted seeds. It is using the registration authority protection tool syntax. The INSEC marker segment is used to signal which codeblocks are scrambled and which seeds are used.

More specifically, we consider the general case where multiple keys are used, defining multiple levels of access. For this purpose, several instances i=0, 1, …, n of the tool are used in the SEC marker segment. Each instance has the s ame tool ID and corresponds to a specific encryption key. The resulting syntax for the SEC marker segment is shown in Figure 10.

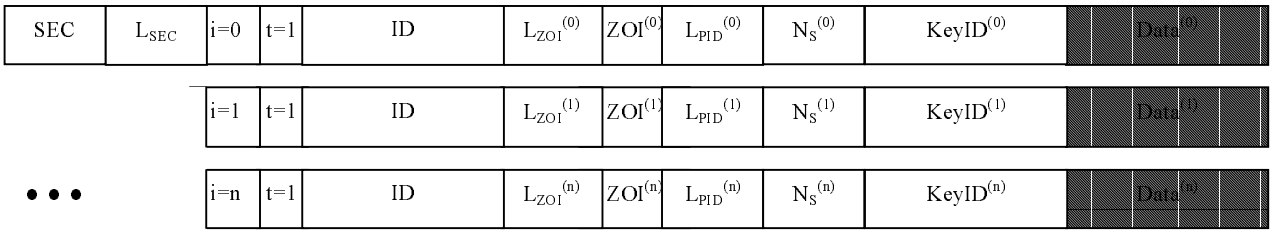

Figure 10 – Syntax for SEC marker segment.

As the registration authority protection tool syntax is used, t is equal to 1. In our example, the zone of influence is not used (we use INSEC instead). $N_s^{(i)}$ is the number of seeds used by the instance i. $KeyID^{(i)}$ identifies the key used by the instance i to encrypt the seeds. Finally, $Data^{(i)}$ contains the encrypted seeds for the instance i.

The INSEC marker segment is used to signal which seed is used to protect which codeblocks. In this example, it is added before the secured codeblock(s), to indicate which seed has been used to protect this/these codeblock(s). Instead of indicating the seed itself, the marker contains an index which refers to the seeds in the SEC marker segment. The resulting syntax of the INSEC marker segment is given in Figure 11 and Figure 12 for the case of wavelet domain and bitstream domain scrambling respectively.

Figure 11 – Syntax for INSEC marker segment (case of wavelet domain scrambling).

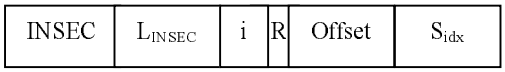

Figure 12 – Syntax for INSEC marker segment (case of bitstream domain scrambling).

S_{idx} indicates the index of the seed used to scramble the following code-blocks, and Offset specifies the offset of the first scrambled byte in the code-block codestream.

4.3. Results

Results obtained with the proposed scrambling technique are presented in Figure 13. In this example, an image has been encoded with 3 quality layers and 3 levels of wavelet decomposition. Scrambling in then applied to a region of interest with different strength. In the first case, all layers and resolutions have been scrambled, and in the second case, only layers 2 and 3 and resolution levels 1 to 3 have been scrambled.

Figure 13 – Scrambling results (left: all layers and resolutions have been scrambled, right: layers 2 and 3 and resolution levels 1 to 3 have been scrambled).

5. CONCLUSIONS

In this paper, we have first reviewed the on-going JPSEC standardization effort. Its goal is to extend the baseline JPEG 2000 specification to provide a standardized framework for secure imaging. JPSEC supports the tools needed to secure digital images, such as content protection, data integrity check, authentication, and conditional access control. The framework is open and flexible, hence ensuring a straight path for future extensions.

We have then presented two examples of JPSEC tools. The first example is a technique for secure scalable streaming and secure transcoding. It describes a JPSEC tool which provides protection of JPEG 2000 codestreams while maintaining a level of scalability and transcodability. This allows the JPSEC codestream to be transcoded while preserving the protection, i.e. without requiring unprotecting (e.g. decrypting) the codestream. The second example is a technique for conditional access control. The proposed technique is very flexible and can be used for access control by resolution or quality, but also by regions of interest. It introduces a pseudo-random noise to selected parts of the code stream, introducing severe distortions for unauthorized users. We have discussed two scrambling methods, applied either to the wavelet coefficients or directly to the codestream.

ACKNOWLEDGEMENT

EPFL's contribution to this work was partially supported by the European Project WCAM http://www.ist-wcam.org (IST Contract 507204) and the European Network of Excellence VISNET http://www.visnet-noe.org (IST Contract 506946), both funded under the European Commission IST 6th Framework Program.

The authors would like to thank all the individuals who participated in the JPSEC ad-hoc group activities for their contributions to the development of JPSEC.

REFERENCES

[1] A. Skodras, C. Christopoulos and T. Ebrahimi "The JPEG 2000 still image compression standard", IEEE Signal Processing Magazine , vol. 18, no. 5, pp. 36 -58, Sept. 2001.
[2] D. Taubman and M. Marcellin, "JPEG 2000: Image Compression Fundamentals, Standards and Practice", Kluwer Academic Publishers, 2002.
[3] JPSEC Committee Draft 1.0, ISO/IEC JTC1/SC29 WG1 N3297, April 2004.
[4] S. Wee and J. Apostolopoulos, "Secure scalable video streaming for wireless networks", In IEEE Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), May 2001.
[5] S. Wee and J. Apostolopoulos, "Secure scalable streaming and secure transcoding with JPEG-2000", In IEEE Proc. Int. Conf. on Image Processing (ICIP), September 2003.
[6] R. Grosbois, P. Gerbelot and T. Ebrahimi, "Authentication and access control in the JPEG 2000 compressed domain", In SPIE Proc. Applications of Digital Image Processing XXIV, San Diego, July/August 2001.
[7] F. Dufaux and T. Ebrahimi, "Video surveillance using JPEG 2000", In SPIE Proc. Applications of Digital Image Processing XXVII, Denver, CO, August 2004.
[8] Y. Sadourny and V. Conan, "A proposal for supporting selective encryption in JPSEC", In IEEE Transactions on Consumer Electronics, vol. 49, no. 4, pp 846-849, November 2003.
[9] Z. Zhang, Q. Sun, Y.Q. Shi, and Z. Ni, "A unified authentication framework for JPEG 2000", in IEEE Proc Int. Conf. on Multimedia and Expo, June 2004.
[10] Y. Wu, D. Ma, and R. Deng, "Progressive protection of JPEG2000 codestreams", in IEEE Proc. Int. Conf on Image Processing (ICIP), Singapore, October 2004.
[11] http://java.sun.com/j2se/1.4.2/docs/guide/security/CryptoSpec.html, Java Cryptography Architecture API Specification and reference.
[12] R.L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM (2) 21, 1978, Page(s): 120-126.