Hindawi
Security and Communication Networks
Volume 2020, Article ID 4836109, 10 pages
https://doi.org/10.1155/2020/4836109

Research Article

Protecting Metadata of Access Indicator and Region of Interests for Image Files

JeongYeon Kim

School of Business, Sangmyung University, 20, Hongjimun 2-gil, Jongno-gu, Seoul 110-743, Republic of Korea

Correspondence should be addressed to JeongYeon Kim; jykim@smu.ac.kr

Received 28 April 2019; Accepted 27 December 2019; Published 22 January 2020

Guest Editor: Rajkumar Soundrapandiyan

With popularity of social network services, the security and privacy issues over shared contents receive many attentions. Besides, multimedia files have additional concerns of copyright violation or illegal usage to share over communication networks. For image file management, JPEG group develops new image file format to enhance security and privacy features. Adopting a box structure with different application markers, new standards for privacy and security provide a concept of replacement substituting a private part of the original image or metadata with an alternative public data. In this paper, we extend data protection features of new JPEG formats to remote access control as a metadata. By keeping location information of access control data as a metadata in image files, the image owner can allow or deny other's data consumption regardless where the media file is. License issue also can be resolved by applying new access control schemes, and we present how new formats protect commercial image files against unauthorized accesses.

1. Introduction

With the increase of data consumption and content creation through the Internet, data security and privacy are one of the major concerns for data sharing. Many technical solutions and frameworks such as network-based protection, access control with authentication, and data encryption have been introduced. However, any of the suggested technologies cannot fully meet all security requirements for various content types and usages. Nowadays, social network services make easier for the Internet users to share multimedia contents rather texting with others. It causes additional security concerns because sensitive personal images or messages shared through social network services could be delivered to strangers [1]. The proliferation of use of digital files brings nonintended release of privacy information, while the owner of the files who posted them on social media only intended to share with limited acquaintances [2]. Image files may contain additional metadata about location, events, and individual relationships between persons in the picture [3], which can be used for secondary hacking.

Moreover, for commercial stocks, the issues of copyright violation and ownership identification get more industry attentions due to frequent occurrences of data abuse and malicious attacks over communication networks. Usually, the attacks are trying to modify or delete specific information in the document to claim legal ownership or proper authorities for content usage.

Addressing these challenges has been an interesting problem for secure communication over open networks. Most multimedia content providers are actively engaged on the preservation of the integrity of their own metadata, by which the Intellectual Property Right (IPR) and the access right information is reserved. Encryption for metadata stored in a specific area named EXIF of an image file helps for the issue [4]. It is not standardized, and popular image decoders cannot support it though.

In this paper, we suggest new metadata management methods based on JPEG standards for Privacy and Security (ISO/IEC JTC 1/SC 29/WG 1/Systems Part4). New JPEG format enhances protection schemes for the sensitive part of the image and important metadata. In addition to adopting new media standards, we suggest to keep access control as a

remote data and store its reference as a metadata in the media file. By using remote access control and keeping its information as a metadata in each image file, the image owner can manipulate the access control data to allow or deny other's data consumption regardless of the place the media file is stored at. License issue also can be resolved by applying new access control schemes. A remote system can decide to allow or deny the given user's access to the media file by the license status under the system login ID. Besides, as users copy a media content file into several networked places, we discuss how new approach protect them against unauthorized accesses.

In the following, we give a description on multimedia-related security issues and ongoing efforts to resolve them. After that, we introduce new JPEG formats and metadata management schemes for security. Also we provide examples of media content, adopting suggested image file formats and explanations on data protection schemes.

2. Related Studies

Security should be a part of data management strategy reflecting all users' information usage. The NIST Cybersecurity Framework, a well-known framework used by many business areas and organizations, helps organizations to be proactive about information security risks [5]. To protect the value of data assets, the framework suggests to have content repositories and identify data assets' location first. Each asset has been assigned access permissions for each current and potential user. The framework also suggests to keep important data in an encrypted form preventing it from unauthorized users' accesses even in the data leak cases.

However, media content management needs additional schemes and efforts because the general risk management approaches cannot resolve multimedia specific security issues, such as privacy or license issues. Images should be handled differently from text data for the issues.

2.1. Security of Multimedia Data. For required additional functionalities of multimedia contents, MPEG standardization group (ISO/IEC JTC1 SC29/WG11) defines a suite of standards for design and implementation of media-handling features, MPEG-M (ISO/IEC 23006). MPEG-M enables easy design and implementation of media-handling value chains with common APIs, protocols, and interfaces for service aggregation mechanisms.

MIPAMS (Multimedia Information Protection and Management System) [6, 7], a service-oriented content management platform, follows a relevant part of the MPEG-M engines. It includes the rights expression language, license, orchestrator, metadata, content protocol, event reporting, content search, security, intellectual property management, and protection engines. The content licensing scenario is a subsystem for specialized content willing to trade, where content files are distributed under copyright with license templates chosen by users.

MIPAMS implements most required functions with the external system including the license management, which makes the system complicated. Instead of keeping license data in the media file itself, additional system to keep user's license information needs user and content identification.

Recently, there are attempts to apply Blockchain technology to protect media contents [8]. KODAKOne [9] is one of the examples for image rights management platform. KODAKCoin is a kind of cryptocurrencies, which can be used to buy the license of images in KODAKOne, enabling photographers to take more control in image rights management. It is a digital ledger of rights ownership for photographers' works. Photographers upload their images to the platform and the records of license purchases are recoded in the ledger. Referring to public Blockchain records, everyone can check if a certain user has a license for digital assets. It enables to track licensing records and prevent illegal uses. However, Blockchain technology is in an early stage for development having limited capacities and not fully defined details, including general identification methods for users and digital assets to records license transactions in the ledger.

2.2. JPEG Privacy and Security. As we have reviewed, privacy and license issues of digital assets are not well addressed and it is an inhibiting factor in the further digital content distribution. To resolve the issues, the JPEG group decided to develop another image file format to enhance security and privacy features [10].

The JPEG format is one of widely used multimedia standards. After the file format of JPEG known as JPEG Interchange Format (JIF), additional standards have evolved. JPEG File Interchange Format (JFIF) and Exchangeable image file format (Exif), both formats use JIF byte layout employing the application markers which is one of the JIF standard's extension points [11]. JFIF uses APP0, and Exif uses APP1.

JPEG XT (ISO/IEC 18477-3 for JPEG-1/JPEG XT) defines a file format to embed boxes in a basic structure of a JPEG-1 image file in order to achieve a common framework for future extensions across JPEG standards. With this box structure, future standards can focus on the definition of boxes and provide compliances across the family of JPEG standards [12].

Adopting these schemes, the JPEG working group suggested a new file format for privacy and security features employing box structure with different application markers. The purpose of the new standards is not suggesting new image coding methods based on better mathematical model, but rather additional boxes for metadata useful for security and privacy in an image file. By using different markers for added boxes, current image decoders just skip new box formats while new image decoders can support new features. Some image codes can be stored in the boxes for metadata, but they can be decoded as the existing image data stored in other areas. Figure 1 shows the compatibility between current JPEG decoders and new decoders.

JPEG privacy and security format suggest concept of replacement as a main method of image privacy protection [13–15]. Replacement means the private part of the image or

metadata can be substituted by given public data. The image owner decides the sensitive area of the image and replaces the original image with the public one. As a result of image replacement, the sensitive data is stored in a created replacement box to substitute data stored in the target area where public data is placed. When image decoder encounters a replacement box in the image file, the associated replacement action should be applied and the resulting file is decoded as usual if the user has a proper authorization for the file. The standards define four types of replacements: box structure in same file, app marker segment, region of interest in image data, and whole file.

Figure 2 shows box replacement, region of interest replacement, and file replacement cases. In a box replacement case, byte offset of the target box counting from the beginning of the file should be provided. In a ROI replacement case, start position and end position of the ROI should be provided. Current standards define the position in the image with vertical offset from the top in pixels and horizontal offset from the left in pixels.

The Privacy and Security standards define the additional method using data encryption to secure data from an unauthorized user. The replacement method has sensitive data in its box format and prevents unauthorized users from accessing it with image decoders, but data should be encrypted to avoid additional accesses.

As described in Figure 3, encrypted data is also stored in a box structure. Encryption related parameters, such as encryption method and initial vector used for key generation, should be stored in the box. The protected content shall be decrypted using provided parameters if the encryption method is supported by the decoder and authorization is granted. Otherwise, the entire protected content shall be ignored.

2.3. JUMBF (JPEG Universal Metadata Box Format). For described protection methods in Section 2.2, new standards guide to keep security-related data as metadata and to use JUMBF (JPEG Universal Metadata Box Format), new box format, for the metadata [16]. The JUMBF box contains exactly one JUMBF description box and one or more content Boxes. According to the definitions of the box structure, a box contains other boxes in it and is called a super box. JUMBF is a kind of super box and also can be nested, but in its first place, there is always a description box.

The type of content Boxes is implied by the JUMBF TYPE field in the JUMBF description box. In the standards, there are several predefined JUMBF types with exactly one content box. Table 1 has the currently defined JUMBF TYPE with one content box in the JUMBF super box. A XML or JSON-type JUMBF box has just one content box, where XML or JSON format text data is containing. Also code stream-type JUMBF contains just one content box having image code stream data, while UUID-type JUMBF contains a box containing box identification data to be referred to from inside or outside of the image file. If JUMBF has more than one content box, it should be defined in related specifications.

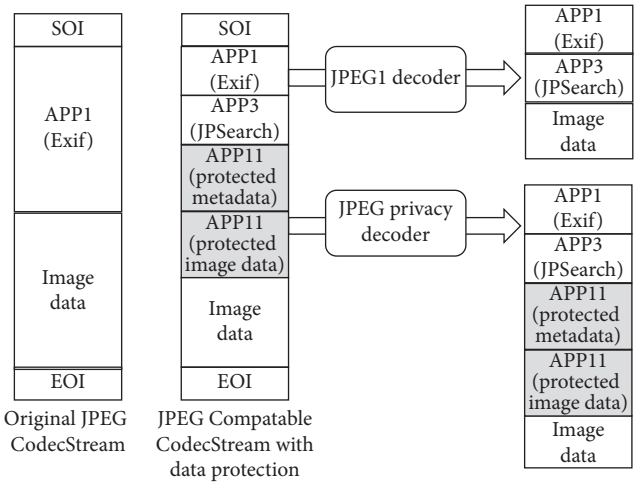

Figure 1: JPEG privacy and security requirements.

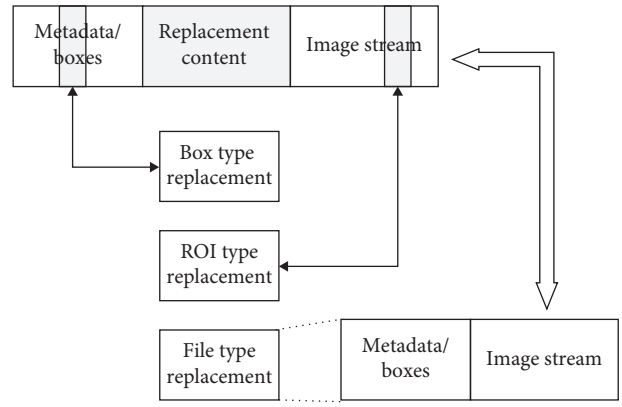

Figure 2: Box structure for replacement methods.

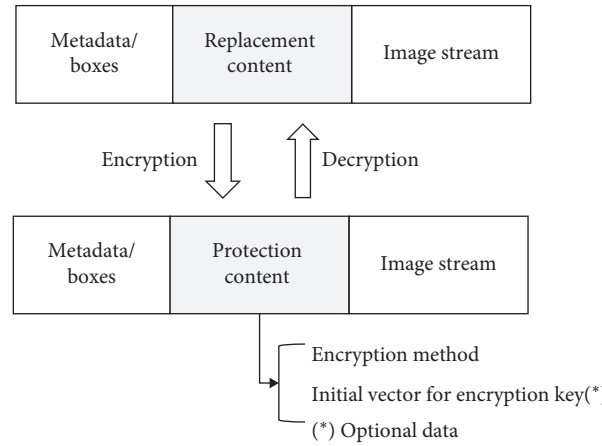

Figure 3: Box structure for protection methods.

Replacement and Protect methods also keep the related data in image files as metadata using predefined super boxes, called replacement-type JUMBF and protection-type JUMBF. As described in Figure 4, the replacement-type and protection-type JUMBF boxes have a JUMBF description box, their own description box, and data boxes. Data for

Table 1: JUMBF content types.

JUMBF type	Meaning
Code stream content type	Exactly one codestream box.
XML content type	Exactly one XML box.
JSON content type	Exactly one JSON box.
UUID content type	Exactly one UUID box.
Other content types	Other content types defined in other specifications

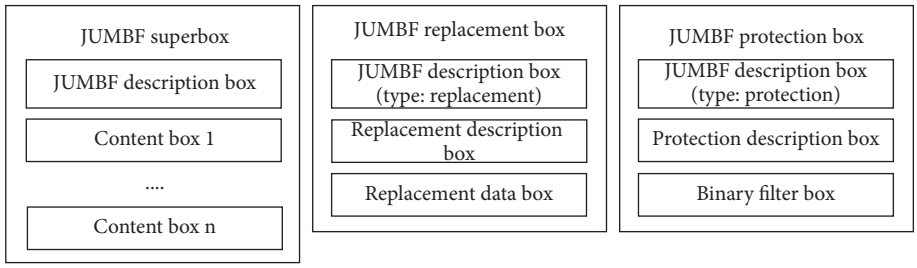

Figure 4: JUMBF superbox (replacement/protection box).

replacement such as target area offsets or data for encryption such as encryption method with an initial vector should be placed in the own description box, while real image data for original images or encrypted data is stored in each data box.

Besides, the protection JUMBF box has several labels referring to another JUMBF boxes just in case more data are required. Figure 5 shows additional labels in the JUMBF protection box.

ENC label is an optional label referencing to a JUMBF Box used to keep additional data encryption-related parameters. There are many encryption methods and some of them may need more parameters. The encoder creates an additional box to store them and record its label in the ENC label space. Image decoders find the parameters by following the label. The AR label is also an optional label referencing to a JUMBF Box having access policy rules of the data in the Binary Data box. Besides, the Initial Vector label is an optional space for an initial value used to start encryption key related process.

3. Image Protection

With conversions of current JPEG image files into new format, sensitive image data such as Personal Identifiable Information (PII) can be hidden by overlapping the related image area with another images. Metadata in images also can be handled as important data by using protection JUMBF boxes.

In this section, we check resolve image license issues applying access control information protected as metadata. Also we take sample images from a movie titled "Hana Restaurant" and explain how to create access control data within a new image format.

3.1. Metadata for Access Control. To admit user's access to certain data, an application program will check user authentication and verify user authorization for the given local resource. Usually access permissions of the resource are generated by a policy or rules which depend on the group the user belongs to. Access rules may be expressed using eXtensible Access Control Markup Language (XACML).

New image format reserves a space of the AR (Access Rule) label in the protection JUMBF box linking to another JUMBF box keeping the actual XACML data. In our examples, we will provide location information of the XACML instead of the data to allow image owners to manage it wherever the actual content file is.

Figure 6 gives an example for using the XML JUMBF box linked to a Protection Box. With XACML in the XML box, the system can provide policy-based access control to image files. The following is an example of XACML policy for date-based access control. It permits any user's view action to an image file named "Sample.jpg" before the end of year 2019. The policy or policy sets defined in XACML 3.0 may have remote references Algorithm 1.

Also, encrypted data can be decrypted by user authorization. Otherwise, the image owner can assign a password. The ENC Label in Figure 5 links to another JUMBF box having parameters for data decryption. The following is an example using the JSON box for the encryption parameters Algorithm 2.

If the encryption method in the linked JUMBF box has password-based encryption, the system will generate a secret key based on Password-based Key Derived Function (PBKDF) [17] to use for data encryption and decryption. The users who provide proper password can decrypt images.

Also access control can be performed with user authorization. The user should provide additional login information to the external image management system, where it keeps image identification and each user's authorization of the image. The system allows or denies a given user's access to images through policies. XACML has a request and response form for resource accesses or a reference to remote policy to get user's current authorization.

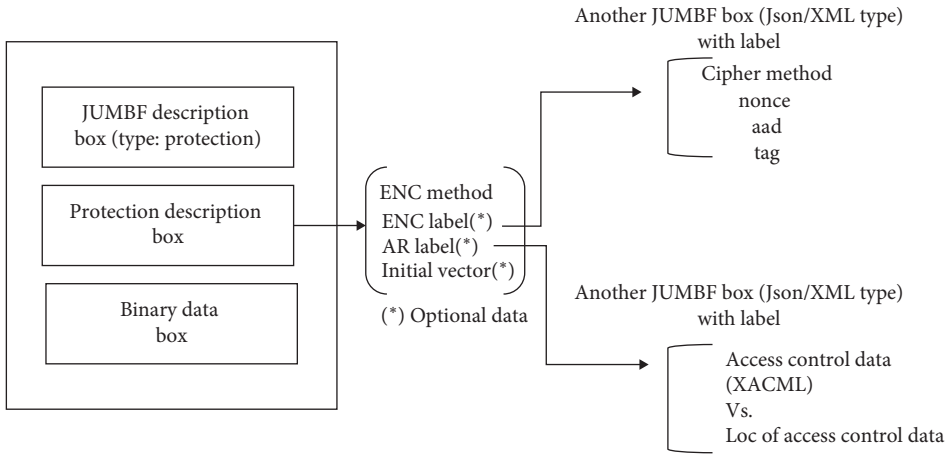

Figure 5: Protection JUMBF box for encrypted document.

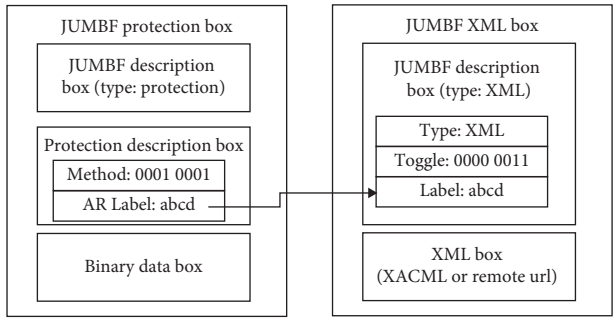

Figure 6: Connection to XML box for access control.

```
<Policy xmlns = "urn:oasis:names:tc:XACML:3.0:core:schema:wd-17"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
PolicyId = "urn:isdcm:policyid:1"
RuleCombiningAlgId = "urn:oasis:names:tc:XACML:1.0:rule-combining-algorithm:first-applicable"
Version = "1.0"
xsi:schemaLocation = "urn:oasis:names:tc:XACML:3.0:core:schema:wd-17
http://docs.oasis-open.org/XACML/3.0/XACML-core-v3-schema-wd-17.xsd">
<Description> Desert.jpg </Description>
<Rule Effect = "Permit" RuleId = "urn:oasis:names:tc:XACML:2.0:ejemplo:Desert">
<Description> Any user can view urn:mimage:Desert.jpg before the end of the year 2019 </Description>
.....
<!--resource-->
<Match MatchId = "urn:oasis:names:tc:XACML:1.0:function:regexp-string-match">
<AttributeValue
DataType = "http://www.w3.org/2001/XMLSchema#string"> urn:mimage:Sample.jpg </AttributeValue>
<AttributeDesignator
AttributeId = "urn:oasis:names:tc:XACML:1.0:resource:resource-id"
Category = "urn:oasis:names:tc:XACML:3.0:attribute-category:resource"
DataType = "http://www.w3.org/2001/XMLSchema#string" MustBePresent = "false"/>
</Match>
.....
<!--action-->
<Match MatchId = "urn:oasis:names:tc:XACML:1.0:function:string-equal">
<AttributeValue
DataType = "http://www.w3.org/2001/XMLSchema#string">
View </AttributeValue>
```

Algorithm 1: Continued.

```
<AttributeDesignator
AttributeId = "urn:oasis:names:tc:XACML:1.0:action:action-id"
Category = "urn:oasis:names:tc:XACML:3.0:attribute-category:action"
DataType = "http://www.w3.org/2001/XMLSchema#string"
MustBePresent = "false"/>
</Match>
. . .. . .
<Condition>
<Apply FunctionId = "urn:oasis:names:tc:XACML:1.0:function:date-less-than-or-equal">
<Apply FunctionId = "urn:oasis:names:tc:XACML:1.0:function:date-one-and-only">
<AttributeDesignator AttributeId = "accessDate"
Category = "urn:oasis:names:tc:XACML:3.0:date"
DataType = "http://www.w3.org/2001/XMLSchema#date" MustBePresent = "false"/> </Apply>
<AttributeValue DataType = "http://www.w3.org/2001/XMLSchema#date"> 2019-12-31 </AttributeValue>
</Apply>
</Condition>
. . .. . .
</policy>
```

Algorithm 1

```
{
  "jpeg_security": {
    "type": "protection",
    "cipher": {
      "method": "AES256-GCM",
      "nonce": "BdZbHABY/sytDTUB",
      "aad": "ZmFzb28uY29t",
      "tag": "1dsCuZ5XuanojwM/p6EoCA = = "
    }
  }
}
```

Algorithm 2

3.2. Metadata for Licensing. With a new image file format, we can manage the license information as a part of access control data or as another policy. It is much easier to manage license issues by access control compared to managing them by metadata, especially for several image copies in distributed environments. If license data is stored as metadata and there are several same files in networks, the system has to update each license metadata as user's authorization changes. However, keeping it as an access policy could be useful to identify the location of the same content files in networks.

In our examples, we integrate license data into access control data and keep the reference of access control policy in a XML JUMBF box.

3.3. Examples. Image format conversion process to generate new protected image has a sequence summarized in Figure 7. In this example, converting sample images, we assume that the image owner selects replacement methods and afterward data encryption methods.

Starting from the original image, we proceed through the following steps:

(1) Select partial areas of the original image and alternative public images to substitute the selected areas. The alternative images should have exactly the same size and dimensions as the selected partial areas of the original image.

(2) Create alternative images open to public by replacing selected partial areas of the original image with given alternative images. The original image data is stored in a replacement JUMBF box.

(3) If needed, the replacement JUMBF box can be encrypted and stored as a protection JUMBF box. The protection JUMBF box may have labels for additional JUMBF box, where encryption parameters or access control policy or rules are stored.

For the sample image, we identify facial images as sensitive areas, and they are replaced with a smile image. In our Windows program shown in Figure 8, the user provides input

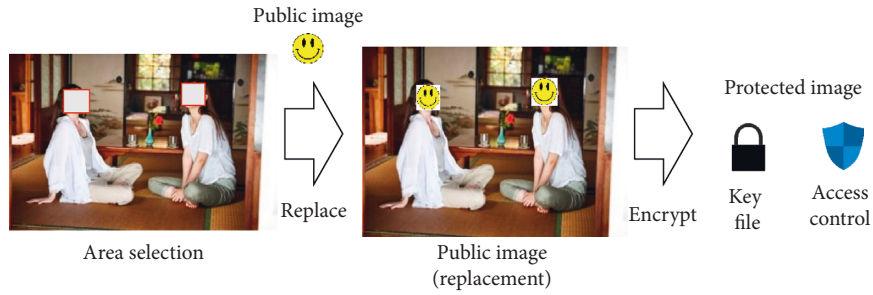

Figure 7: Protection JUMBF box for encrypted document (public images from Korean movie titled "Hana Restaurant" 2018).

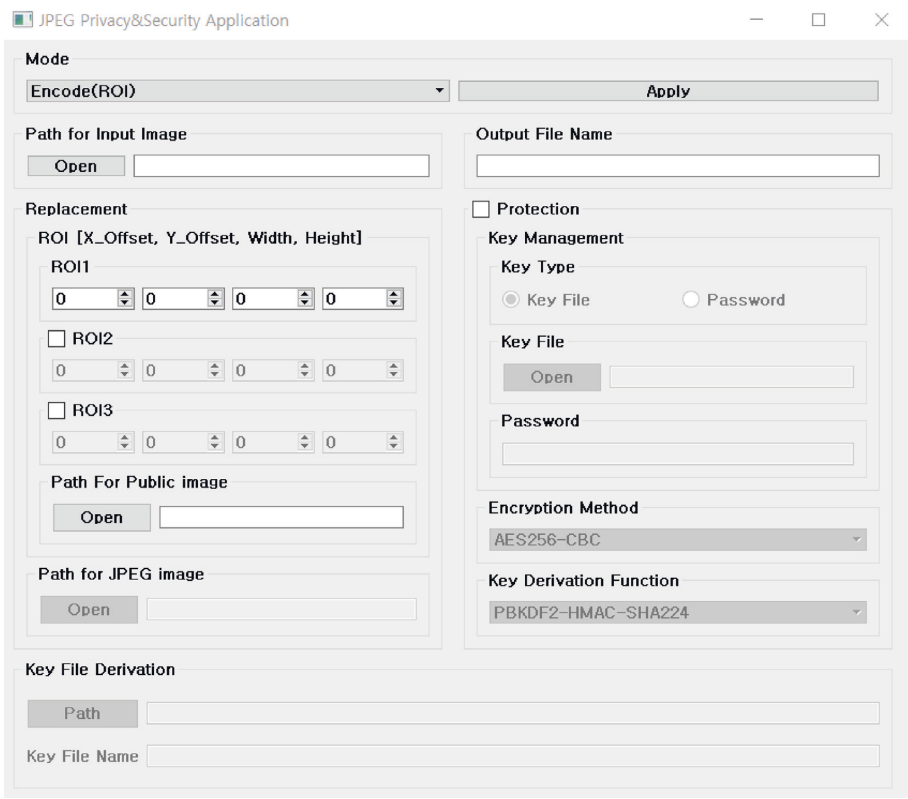

Figure 8: Application for replacement and protection JUMBF box.

file path, output file path, alternative image path, and ROI offsets to be replaced. After the file conversion, area's offsets and partial images are stored in the replacement JUMBF box, as shown in Figure 9.

After the creation of replacement JUMBF boxes, it can be encrypted. Assigned access control data and encryption related data can be stored in separate JUMBF boxes, and the protection JUMBF box keep their labels in the specific area. In this example, new image file has a XML JUMBF box having the location of access rule information, as shown in Figure 10.

As we explained in Section 3.2, the user will be a member of the licensed user group if the user has a license and the group has an authority to use the content.

Decoding the converted image files proceeds as follows:

(1) During sequential file scanning, a protection JUMBF box is decrypted for users who have the appropriate rights for the file. Otherwise, skip the box.

(2) If the decrypted content is a replacement JUMBF box, original images are merged at the position determined by offsets x and y.

(3) The resulting image file will be decoded as usual.

With described processes, users with proper rights can access to the original image and others access to alternative security images. We applied replacement and encryption methods with a password to public 44 images of a movie titled "Hana Restaurant" available from a Korean portal site. Figure 11 shows the selected results. Login-based authorization needs the additional management system, and we

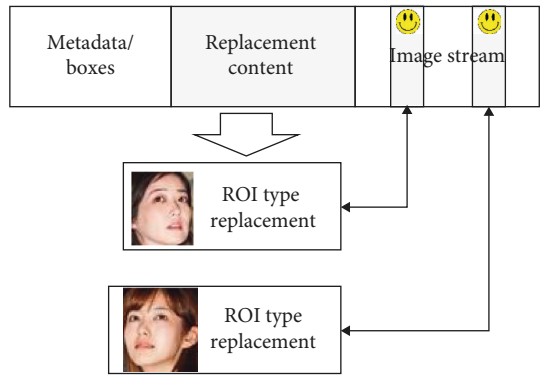

Figure 9: Replacement JUMBF box for ROI (public images from Korean movie titled "Hana Restaurant" 2018).

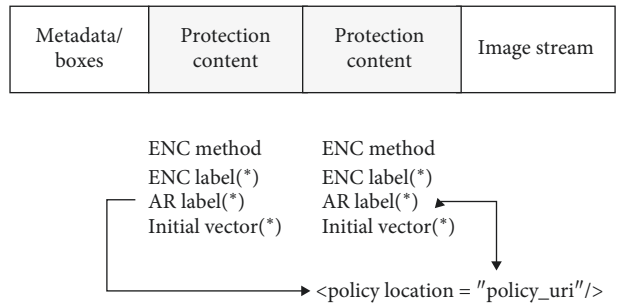

Figure 10: Protection JUMBF box for encrypted document.

Figure 11: Authorized view with password vs. public view (public images from Korean movie titled "Hana Restaurant" 2018).

encrypted the replacement boxes with the password-based method to simplify the access control.

Image encoding and decoding time varies depending on images' count of replacements and the size for encryption data. For images with 2 replacements and encryptions, the average encoding time of replacement is 0.35 seconds while password-based encryption takes 82% more time. Figure 12 shows the time ratio of encoding and decoding time between the original image, images with replacement boxes, images with password-based encryption boxes, and images with

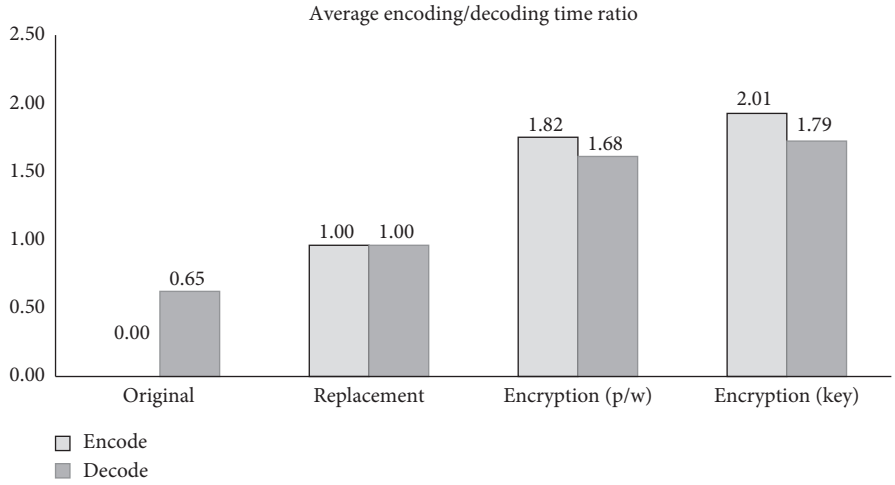

Figure 12: Performance of encoding and decoding.

encryption based on AES256-CBC algorithm with the initial vector.

4. Conclusion

We reviewed image protection methods for region of interests and various metadata suggested in JPEG privacy and security standards. Suggested methods provide a mechanism to hide sensitive image parts of personal identifiable areas by overlapping other images. Compared to other JPEG standards for secure images, Secure JPEG 2000 (JPSEC), new standards are focusing on metadata and using JPEG XT box formats instead of suggesting new code stream methods. The JPSEC standards' specification targets protection of the entire code stream or segments of JPEG 2000 coded data and additional syntax to specify associated protection methods. However, new standards allow XML or JSON-type data and even code stream in the metadata areas and provide more flexibility applying additional security features such as XACML policies. Important metadata such as payment or licensing data also can be applied for access control policies.

We provide examples of JPEG image protection and privacy enhancement features using new standards' replacement and protection methods. The personal identifiable information in images can be replaced with public subimages. After deciding image areas to replace with another images, the image owner manages access control data by the following steps:

(i) Placing reference of access control policy in each image file

(ii) Placing and encrypting reference information of license policy in each image file if the original image has one

(iii) Deciding default access allowance for each image file

As a further study, we are planning to review other standards for multimedia data to cooperate with image metadata. Multimedia blockchain with smart contract functions is a platform to support various multimedia data.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by a 2018 Research Grant from Sangmyung University.

References

[1] M. Blaze, John Ioannidis, and A. Keromytis, "Experience with the keynote trust management system: applications and future directions," p. 1071, Trust Management, Berlin, Germany, 2003.

[2] T. Jing, Q. Chen, and Y. Wen, "A probabilistic privacy preserving strategy for word-of-mouth social networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6031715, 12 pages, 2018.

[3] Y. Liu, W. Zhang, and N. Yu, "Protecting privacy in shared photos via adversarial examples based stealth," *Security and Communication Networks*, vol. 2017, Article ID 1897438, 15 pages, 2017.

[4] H. Wijayanto, I. Riadi, and Y. Prayudi, "Encryption EXIF metadata for protection photographic image of copyright piracy," *International Journal of Research in Computer and Communication Technology*, vol. 5, pp. 237–242, 2016.

[5] J. Guinn, *Why You Should Adopt the NIST Cybersecurity Framework*, PwC, London, UK, 2015.

[6] S. Llorente, E. Rodriguez, J. Delgado, and V. Torres-Padrosa, "Standards-based architectures for content management," *IEEE MultiMedia*, vol. 20, no. 4, pp. 62–72, 2013.

[7] J. Delgado, S. Llorente, and E. Rodriguez, "Digital rights and privacy policies management as a service," in *Proceedings of the 2012 IEEE Consumer Communications and Networking Conference (CCNC)*, IEEE, Las Vegas, NV, USA, January 2012.

[8] D. Bhowmik and F. Tian, "The multimedia blockchain: a distributed and tamper-proof media transaction framework," in *Proceedings of the 2017 22nd International Conference on Digital Signal Processing (DSP)*, IEEE, London, UK, August 2017.

[9] I. Wenn DIgital, *KODAKOne|Image Rights Management Platform*, WENN DIgital, Inc., Los Angel, CA, USA, 2008, https://kodakone.com/.

[10] F. Temmermans, T. Ebrahimi, S. Foessel et al., "JPEG privacy and security framework for social networking and glam services," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 68, 2017.

[11] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18–34, 1992.

[12] T. Richter, "On the standardization of the JPEG XT image compression," in *Proceedings of the 2013 Picture Coding Symposium (PCS)*, IEEE, San Jose, CA, USA, December 2013.

[13] P. Schelkens, "Image security tools for JPEG standards," in *Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security*, ACM, Salzburg, Austria, June 2014.

[14] L. Yuan, P. Korshunov, and T. Ebrahimi, "Privacy-preserving photo sharing based on a secure JPEG," in *Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, Hong Kong, China, May 2015.

[15] L. Yuan, P. Korshunov, and T. Ebrahimi, "Secure JPEG scrambling enabling privacy in photo sharing," in *Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, May 2015.

[16] A. Kuzma, F. Temmermans, and T. Richter, "Emerging image metadata standards activities in JPEG," in *Proceedings of the Applications of Digital Image Processing XLI International Society for Optics and Photonics*, vol. 10752, San Diego, CA, USA, August 2018.

[17] K. Moriarty, B. Kaliski, and A. Rusch, "PKCS# 5: password-based cryptography specification version 2.1," 2017, RFC 8018, https://www.rfc-editor.org/info/rfc8018.