A Public-Key Approach of Selective Encryption for Images

Feng Jiang, Paul Salama, and Brian King
(Corresponding author: Brian King)

Department of Electrical and Computer Engineering, Indiana Univeristy-Purdue University Indianapolis
420 University Blvd, Indianapolis, IN 46202, USA
(Email: briaking@gmail.com)

(Received Oct. 3, 2015; revised and accepted Dec. 7, 2015 & Feb. 28, 2016)

Abstract

Data security protection is essential for most multimedia data transmissions today. Classical multimedia content protection is dominated by symmetric encryption methods. This paper explores the possibility of public-key media content encryption. The key problem to the asymmetric selective encryption is formalized and defined as the "bounded plaintext problem". Possible solutions to this problem are proposed. A public-key multimedia encryption model is developed and the implementation results are provided.

Keywords: Bounded plaintext problem, EZW, public-key encryption, selective encryption, SPIHT

1 Introduction

Today, society has an insatiable desire for multimedia (such as video or images). The computational and bandwidth demand of multimedia can be significant. Multimedia data usually requires enormous storage and real time computation capabilities. Consequently, efficient multimedia compression techniques are necessary. Further, multimedia content is frequently transmitted over an insecure network [8, 37]. Many applications like military image databases, video conference, medical imaging system, etc. require efficient and secure digital image transmissions [4, 6, 12, 13]. Encryption is the essential technology used to provide confidentiality of the multimedia content. One typical multimedia encryption technique is to merge the compression and encryption methods together, a process called *selective encryption* [19, 25, 32, 35].

The classical multimedia encryption systems usually utilizes symmetric encryption schemes or hybrid encryption [5, 7] schemes. The latter encrypts the symmetric encryption key using public-key encryption schemes [33] and then encrypts the multimedia data using symmetric key encryption. If a public-key encryption scheme is used to protect the multimedia data sequence, then the encryption process needs to be implemented repeatedly, because of the capacity of a single public-key encryption. This adds a significant computational burden on the sender, more so for the receiver.

Using the selective encryption, a desired security can be achieved by encrypting part of the multimedia data. Based on Uhl's [21, 22] analysis of JPEG2000, at least 20% of the image data need to be encrypted to achieve a reliable security. Brahimi [2] showed in his work that 11% of a medical image encrypted leads to a suitable PSNR (indicating low quality). Lian [18] presented a result that 15.3% of the image data encrypted generates a poor decompressed image, which is secure for common applications. However, these plaintext sizes significantly exceed the common plaintext sizes of the public-key encryption (as listed in Table 1). It is accepted by most scholars [21, 22, 39] that the public-key encryption schemes should not be used for multimedia content protection.

In networked applications of multimedia delivery, the symmetric key is usually transmitted from the receiver to the sender using some public-key cryptographic scheme. Thus, at least one public-key cryptographic algorithm must be applied, even when the multimedia content is protected using symmetric key encryption. It would be much more efficient if one can deliver the secured multimedia content using only one public-key encryption. Motivated by this, we explored the possibility of public-key multimedia encryption in this paper. Here we formally define a problem, the "bounded plaintext problem" (see Definition 1), which will be the essential problem of using public-key methods for multimedia encryption. We discuss a solution, as well as results from our implementation experiments.

The rest of the paper is organized as follows. In Section 2, the bounded plaintext size problem is defined based on public-key cryptosystems and the selective encryption for multimedia data. Section 3 discusses the solution to the bounded plaintext problem. Experimental results are provided in Section 4. In Section 5 we provide a conclusion.

2 The Bounded Plaintext Problem

Common public-key cryptosystems include: RSA, ElGamal, Elliptic Curve El Gamal, etc. In a public-key cryptosystem there are two keys, a public key and a secret key. The two keys are mathematically related, but it is "computationally infeasible" to determine the secret key from the public-key [33]. The "feasibility" is related to a computationally "hard" problem. Many of the public-key cryptosystems are related to one of three hard problems: the integer factoring problem, the discrete-log problem and the elliptic curve discrete log problem.

Typically the security of a secure cryptosystem is based on a time period. That is, if there is no known attack on a cryptosystem, then the only remaining attack is some type of key search (such as a factoring algorithms, Pohlig-Hellman, etc.). Such algorithms tend to run in exponential time and are useful when they are able to run for sufficient amount of time or when the key is small. There has been several studies concerning public-key cryptosystems and the appropriate keysize for a given time period [9, 11, 14, 16].

A parameter that determines the security of a secure public-key cryptosystem is typically a fixed year, such that the time difference between the end of the fixed year and the current date provides a duration for a key search algorithm to determine the key. Thus, once one fixes a year, then a lower bound on the key size is provided.

We refer to a secure plaintext size as the "bound" on the plaintext size for keys secured until a given year. The secure plaintext size for these three problems, for the year 2015, is given in Table 1. Here, the problems have been analyzed with several analysis approaches such as Lenstra [16], etc.

Table 1: Plaintext length (bits) of public-key encryption secured for the year of 2015

Analysis Approach	Factoring	Discrete Logarithm	Elliptic Curve Logarithm
Lenstra [16]	1248	1613	154
Lenstra [14] Updated	1350	1245	156
ECRYPT II [11]	1248	1248	160
NIST [1]	2048	2048	224
ANSSI [9]	2048	2048	200

The "hardness of a problem" is often measured against the problem (input) size. For example RSA is related to the integer factoring problem. Currently, factoring a 1248 bit composite integer is considered infeasible. Consequently a secure plaintext length for RSA is approximately 1248 bits. In general, for a given public-key cryptosystem, if one fixes the parameter(s) to a security level for the underlying hard problem, then one has bounded the plaintext size to the given level[1].

For example, if once we adopt 1248 bits RSA in our multimedia data encryption, then we can support plaintext size up to 1248 bits. Throughout we will use Δ to represent the security level parameter.

Selective encryption ($\mathcal{SE}$) [19, 25, 32, 35] has been proposed and applied to multimedia encryption by scholars in recent years. The concept of selective encryption is to selectively encrypt part of the multimedia data. The two basic parts of selective encryption are the selection process S_{proc} and the encryption process E_{proc}. The E_{proc} is considered to be a typical cryptographic algorithm encrypting the plaintext M, where M is a subset of the multimedia sequence Seq. For example, in [24], a certain bitplane of the image sequence is selected to encrypt, the subset M, in this case, is the selected bitplane from all bitplanes.

It was shown by Hellman [10] that compressing data before encrypting it, increases security in the sense that more ciphertext will be needed to determine the encryption key than if the data had been encrypted directly. To improve the encryption efficiency, the selective encryption is usually performed in some transformed signal domain and integrated within the compression process. For example, [25, 40] implemented the encryption in the frequency domain, [19, 35, 38] integrated the selective encryption with compression techniques. In our previous work[28], only the beginning part of the compressed image sequence is encrypted to increase the encryption efficiency.

An efficient selective encryption is a selective encryption scheme:

$$\mathcal{SE} = (S_{proc}, E_{proc}, \Delta, Comp) \tag{1}$$

where selection process S_{proc} is accomplished in some domain D_m (multimedia compressed sequence), after using compression technique $Comp(\cdot)$, such that the most informative part of the original data is selected and encrypted generating a required security level Δ. Here, E_{proc} is the encryption process, $\mathcal{M}_{se}$ is the plaintext space and $\mathcal{C}_{se}$ is the cyphertext space.

As noted earlier, in a selective encryption scheme the encryption scheme used will be selected to satisfy the security parameter Δ.

However, to evaluate the security of the selectively encrypted data sequence (ciphertext), the whole sequence needs to be evaluated. We recognize that part of the sequence remains in the clear, so one cannot claim the data sequence is secure at a level of Δ. One will have to explicitly design the selection process S_{proc} and evaluate the security of the resulting data sequence accordingly.

When we use the term a *secure selective encryption scheme*, we mean that the entire multimedia data sequence, which includes the selectively encrypted data as well as the data that has not been encrypted, is secure against the known selective encryption attacks.

For example, $Comp$ = JPEG 2000, S_{proc}=first 20 % of the multimedia compressed sequence and E_{proc} =128 bit AES. The key size 128 for symmetric key is secure

[1] By security level, we are referring to a period of time or year, for which the secret key is secure against any known attacks.

according to [9] until year 2090, so security parameter $\Delta = \textit{year 2090}$.

Definition 1. ***Bounded Plaintext Problem:*** (E, Δ) *Given a public-key encryption algorithm E and a desired security level Δ, the plaintext size is bounded in bit length. For these fixed parameters, does there exist a secure selective encryption scheme $\mathcal{SE} = (S_{proc}, E_{proc}, \Delta, Comp)$ in which the encryption process E_{proc} is the public-key encryption algorithm E that is applied exactly once.*

For example, if one is given $E_{proc} = RSA$ and security parameter $\Delta = \textit{year 2015}$, then the plaintext size is limited to 1248 bits according to [16]. Thus one can only encrypt 1248 bits of the image. We have seen, via literature, that the compression scheme JPEG 2000 requires a substantial percentage of the image to be encrypted [21, 22]. Thus for (E, Δ)=(RSA,2015), if there exists a secure selective encryption scheme then the compression scheme cannot be JPEG 2000.

The essential task to solve the bounded plaintext problem is to determine the compression scheme $Comp(\cdot)$, which generates very high information rate [23] in a small locality, where the remaining data is provides significantly less information and then determine S_{proc}.

3 Solving The Bounded Plaintext Problem

3.1 Rate-scalable Wavelet Compression

A rate scalable compression [28, 31] technique allows compressing the multimedia data once and decompressing it at multiple data rates or quality. The user can stop the decompression process at any point of the compressed bitstream. A higher bit rate or bandwidth leads to a better decompression quality. There are several rate-scalable compression techniques, such as Embedded Zero tree Wavelet (EZW), Set Partitioning In Hierarchical Trees (SPIHT) and JPEG2000 [17]. EZW and SPIHT compressions are typical implementations of the initial rate scalable theory. Given a rate scalable compressed multimedia sequence, the more bitstream the user can decompress, the higher quality the user can achieve. The rate scalable compression process guarantees the user always get the "best" representation of the multimedia data with limited resources or computing capability. Typical rate scalable compression techniques are developed based on wavelet transform of the original data. The compression is performed to reduce the data dependency within the wavelet transformed frequency domain. As the inherent multi-resolution [27, 30] characteristic of the wavelet transform, the compression process is implemented from the most significant element to the least significant element successively. It is reasonable to expect that the information rate is much higher at the beginning part of the compressed sequences. We first performed the information rate test by using the EZW compression and then implemented both EZW and SPIHT compressions in our pilot encryption/decryption system.

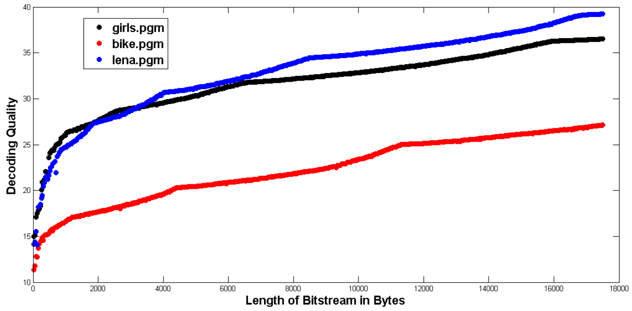


Figure 1: Decompression quality change through the whole bitstream

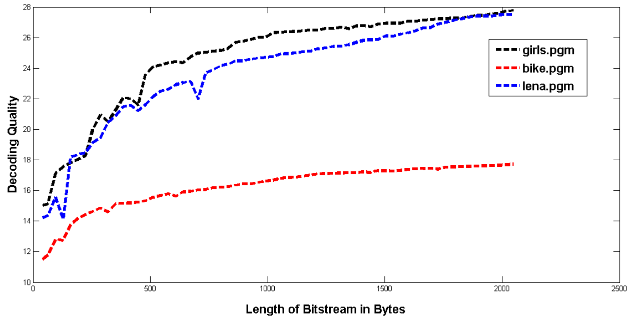


Figure 2: Decompression quality change within the first 2048 bytes

3.2 Testing Rate-scalable Wavelet Compression as a Solution to Bounded Plaintext Problem

3.2.1 Information Intensity Test

It is well know that the decompression quality increases by increasing decompression bitstream length. To test the information rate of the compressed bitstream, we successively increased the length of the selection of the bitstream and feed that into the decoder. The decoded image quality is calculated and plotted. As shown in Figure 1, Figure 2, three different images are tested, the decompression quality is gradually increased by increasing the decoding bitstream length. However, the decompression quality is not increased in a constant speed. Figure 2 is a concentrated version of Figure. 1. The decompression quality is increased greatly by only decoding the first 2048 bytes of the whole bitstream and the decompression quality increases at a much higher speed when the decoding bitstream length is small. Therefore, it is possible to find a small part of the entire bitstream representing the most significant information of the entire image. Our hypothesis was proven to be practical.

The experimental result also confirms the characteristics of wavelet compression in terms of applying such compression technique in this setting. Usually, the low frequency coefficients of the image are compressed first and the marked as the most important elements. The decoder can retrieve the background color of the image first and retrieve the detailed image features successively in the decoding process. The information rate at the beginning part of the compressed bitstream is much higher than the other parts.

3.2.2 Correlation Test

However, it is still too early to argue that hiding the initial part of the bitstream will secure all the useful information of the image, if it is possible to recover the beginning part by any other correlated bitstream. Our next experiment tested the correlation between a selected bit sequence at the initial part and a later part of the compressed sequence.

The first 224 bits (excluding the header part) of the compressed sequence is selected as a target sequence according to the public-key plaintext size in the Table. 1. To fully test the correlation between the target sequence and all the other bit sequences within the bitstream, we sampled through the entire bitstream uniformly and generated 250 sequences with same bit length as the selected targeted sequence. The correlation test was performed for three 512×512 grayscale images, "lena.pgm", "girls.pgm" and "bike.pgm". The goal of the correlation test is to check the relationship between two bit sequences in bit length n, denoted as bitstream X and bitstream Y.

In our experiment, we noticed that the general statistical correlation calculation

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (2)$$

was not a good fit here because the samples x_i and y_i can only take on value "0" or "1" in a bitstream. Thus different bitstream pairs with different patterns but same hamming weights will generate same correlation values. Consequently, we adopted the correlation coefficient C_{XY} test designed for bit sequences as described by Menezes et. al. in [20]. For $X = x_1, \ldots x_n$ and $Y = y_1, \ldots, y_n$, C_{XY} is defined as

$$C_{XY} = \frac{2(A(X,Y) - \frac{n}{2})}{\sqrt{n}}, \quad (3)$$

where

$$A(X,Y) = \sum_{i=1}^{n}(x_i \oplus y_i). \quad (4)$$

Here $A(X,Y)$ represents the Hamming distance between X and Y, $\oplus$ denotes the XOR operator and $\sum$ denotes the summing of bits which differ between X and Y.

The correlation test result C_{XY} approximately follows a Normal $N(0,1)$ distribution when $n \geq 10$ [20]. A two-sided test is applied. The hypothesis, "the two sequence are uncorrelated" serves as the null hypothesis H_0, and the alternative hypothesis, "the two sequences are correlated" is denoted by H_1.

The correlation between the target sequence X and the other 250 sampled sequences generated from the data in clear were tested. Among the 750 correlation values for the three testing images, the largest correlation value is 3.07 in absolute value, 747 out of the 750 absolute correlation values are lower than 2.5758. According to the standard normal distribution $N(0,1)$ [36], with the significance level $\alpha = 0.005$, we will rarely reject the null hypothesis, i.e. rejecting that the two tested sequences are uncorrelated. Thus, one can be very confident that the high information sequence is not correlated with the sequence generated from the data in the clear.

Table 2: Absolute correlation value between the selected bitstream and other bitstreams

image	largest absolute correlation	average absolute correlation
lena	2.539	0.8128
girls	3.0535	0.8054
bike	3.0735	0.77

Table 3: Plaintext length and decompression image quality of EZW sequences

image	size	plaintext length (bytes)	PSNR
lena	512 × 512	16	7.79
lena	512 × 512	16	7.77
lena	512 × 512	16	8.24

3.2.3 Decompression Quality Test

We then tested the decompressed image quality. We applied the decompression process to a sequence consisting of the image bitstream but where a small part of the initial bitstream is scrambled. As shown in Table 3, if a small part of the sequence is corrupted then this could cause severe intelligibility loss.

From the above experiments, we discovered that the rate scalable compression technique, such as EZW, generates a bitstream with decreasing information rate. In fact, the SPIHT compression technique, which is not discussed in this section, has proven to have the same characteristic.

It is included in the implementation results. The information rate is highest at the initial part of the bitstream. A small part of the bitstream lost or ruined in this part destroys the decompression process and the lost part cannot be recovered by any of the remaining bits of the bitstream.

3.3 Using Public-key Cryptosystems in a Selective Encryption Scheme

The encryption and decryption process of our public-key selective encryption model can be represented by $(E, D, \mathcal{K}_{sk}, \mathcal{K}_{pk}, \mathcal{M}, \mathcal{C})$ with the secret key space $\mathcal{K}_{sk}$ and public key space $\mathcal{K}_{pk}$. E is the encryption algorithm of selective encryption $\mathcal{SE}$ we defined in Equation (1). D is the compression output sequence of the compression technique *Comp* defined in Equation (1). The plaintext $\mathcal{M}$ is one plaintext of the plaintext space $\mathcal{M}_{se}$ defined in Equation (1) and Δ is the required security level. Let n denote the plaintext bound in bits and h denote the number of bits of the parameters within the header that need to be encrypted. The procedures of encryption and decryption are described in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Encryption process of the public-key selective encryption

```
Public key encryption: $(E, D, \mathcal{K}_{sk}, \mathcal{K}_{pk}, \mathcal{M}, \mathcal{C})$
user: public key pk
Security parameter: $\Delta$ which implies $n$ bit plaintext bound
input: Image $I$
  $D = Comp(I)$
  $H = header$ of $I$ and $h$ bits of $H$ need to be secured
  For bit $b \in D \cup H$,
      $S_{proc}(b) = \begin{cases} 1 & b \in H \text{ and } b \text{ needs to be secured} \\ 1 & b \in D \text{ and b is among first } n-h \text{ bits} \\ 0 & \text{otherwise} \end{cases}$
  Let $\mathbf{b} = (b_{1_1}, \dots, b_{i_n})$ where $S_{proc}(b_{i_j}) = 1$.
  Map $\mathbf{b}$ to $m_{\mathbf{b}} \in \mathcal{M}$
  Let $\mathbf{r}$ denote the set $(\nu_1, \dots \nu_t)$, $\nu_i \in D \cup H$ which $S_{proc}(\nu_i) = 0$.
  Let $\mathbf{o} = (E_{\mathsf{pk}}(m_{\mathbf{b}}) || \mathbf{r})$
  Return the selective encryption ciphertext $\mathbf{o}$
```

The public-key encryption algorithm we applied our work to is elliptic curve cryptography, which is well recognized as a bandwidth friendly type of public-key encryption. To form a general solution of the bounded plaintext problem, we select a public-key algorithm (elliptic curve encryption) with relatively small plaintext size.

Elliptic curve cryptography is defined over some finite field $\mathbb{F}$. For $a_i \in \mathbb{F}$ for $i = 1, ...5$, the elliptic curve E is a set of all points in $\mathbb{F} \times \mathbb{F}$, which satisfy an equation of the form

$$y^2 + a_1xy + a_2y = x^3 + a_3x^2 + a_4x + a_5, \qquad (5)$$

as well as the point of infinity $\mathcal{O}$.

There is a natural addition defined over E and so that E forms a finite additive abelian group, and the curve is selected so that a large prime q divides the group order of E.

Algorithm 2 Decryption process of the public-key selective encryption

```
Public key encryption: $(E, D, \mathcal{K}_s, \mathcal{K}_p, \mathcal{M}, \mathcal{C})$
user: secret key sk
Security parameter: $\Delta$ which implies $n$ bit plaintext bound
input: ciphertext $\mathbf{o}$
  Let $\mathbf{c}$= first $n$ bits of $\mathbf{o}$ and let $\mathbf{r}$ = remaining bits of $\mathbf{o}$
  Let $m = Dec_{\mathsf{sk}}(\mathbf{c})$
  Map $m \in \mathcal{M}$ to a bit sequence $\mathbf{b}$
  Let $\mathbf{h}$ be the first $h$ bits of $\mathbf{b}$ and let $\mathbf{w}$ denote the remaining bits
  Insert $\mathbf{h}$ and $\mathbf{w}$ into $\mathbf{r}$ in the appropriate places, the result is $\mathbf{v}$. This step is defined by $S_{proc}$
  Let $T = Decompress(\mathbf{v})$
  Return image $T$
```

The scalar multiple of a fixed point P of E is the necessary cryptographic applications. That is, k is a positive integer $0 < k < q$ and the scalar multiple kP (which is $P + P + \cdots P$) is the essential calculation.

To achieve elliptic curve encryption one can use *Elliptic Curve El Gamal* [34]. If k is the user's secret key then kP is the public key. In order to encrypt the message W (which is an elliptic curve point), the sender gets the user public key kP, they select r randomly, where $0 < r < q$. They then compute

$$C_1 = rP \text{ and } C_2 = W + r(kP).$$

They then send (C_1, C_2) to the user. The user can decrypt (C_1, C_2) and can recover W by calculating

$$W = C_2 - kC_1.$$

According to Table 1, 224 bits plaintext size should be secure for elliptic curve encryption today. So q is approximately 224 bits in size. Thus elements in $\mathbb{F}$ are bounded by 224 bits. Though the message $M \in E \subset \mathbb{F} \times \mathbb{F}$, is a point , with approximately 448 bits. Because M satisfies Equation (5), the y-coordinate of the point has very little entropy (approximately one bit). Thus for this setting, the bounded plaintext problem is such that the plaintext is bounded by 224 bits. Concerning the conversion of the image sequence b to a point W. It is not trivial to map a bit-sequence to a point in an elliptic curve. Work in [15] and [34] show how such a mapping can be performed. The details of this mapping is outside the scope of this paper.

4 Analysis of the Implementation Results

Our analysis shows that it is difficult to retrieve any information from the compressed sequences with only a small part of the multimedia data sequence encrypted. Thus the bounded plaintext problem can be solved by utilizing a wavelet rate-scalable compression technique, such as EZW and SPIHT. The public-key multimedia encryption can be achieved by selectively encrypting a small important part of the multimedia data sequence. Here, the encryption process E_{proc} can be elliptic curve public-key encryption. The plaintext, in our work is bounded in 224 bits in order to utilize a elliptic curve public-key encryption secured for 2015.

To select the "most important part" from the multimedia sequence generated by compression $Comp(\cdot)$, the image sequence is first analyzed by per element.

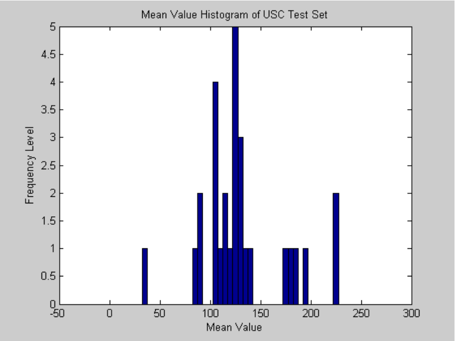


Figure 3: Image mean value distribution map of the USC test set

Usually there are two parts of a compression sequence, the general encoding settings (the header part) and the data part. For the header part, coding parameters such as compression rate, input output image dimensions, quality requirement and package indicators are not considered as necessary elements to be kept private, so no need to encrypt.

However, some important parameters related with image content or unique image identification information are necessary to be protected or tested to see if they should be protected.

For example, the initial threshold value and the mean value of the image in the EZW compression should be encrypted. The DC component of the wavelet transform in SPIHT compression should also be encrypted. The encoding threshold or DC component is considered as a necessary element because it is determined by both the original image features and the wavelet transform. Leaving the image mean value in the clear may result in leaking important information such as the background color or image shooting time. As shown in Figure. 3, it is natural to expect most images have the mean value near the middle point of the entire gray level, which is 128. However, an image with very dark or white background laying on the side parts of the map will leak information easily.

Both EZW and SPIHT compression techniques were implemented in our experiments. To satisfy the bounded plaintext requirement for the elliptic curve public-key encryption, 224 bits are selected from each rate scalable compression sequence. As listed in Table 4, for the EZW compression, the initial threshold, image mean value and the followed sequence data are selected to form 224 bits plaintext. 55 bits header information are skipped. For the SPIHT compression, all the header information except the DC component are skipped. The plaintext starts at the 105th bit. The reference EZW compression codec used in our experiment was developed by video and image processing laboratory at Purdue University [3, 26, 28, 29].

Table 4: Plaintext selection of EZW and SPIHT sequences

	Plaintext reconstruction	Bit location
EZW	image mean within the header	56 - 63
EZW	initial threshold value	64 - 71
EZW	package data	72 - 279
SPIHT	DC component within the header	105 - 112
SPIHT	package data	113 - 328

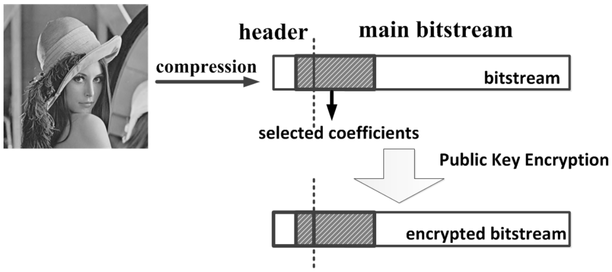


Figure 4: Proposed asymmetric encryption model-encryption process

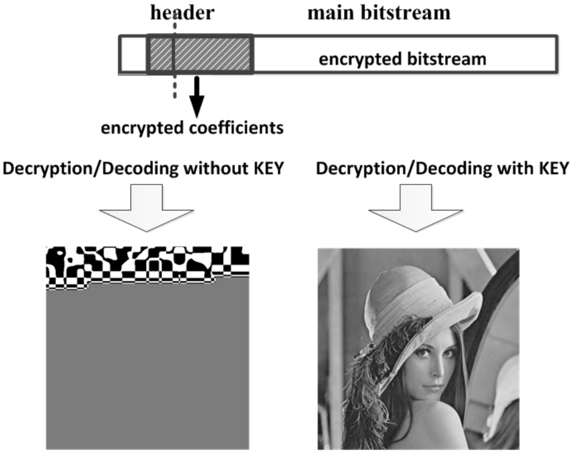


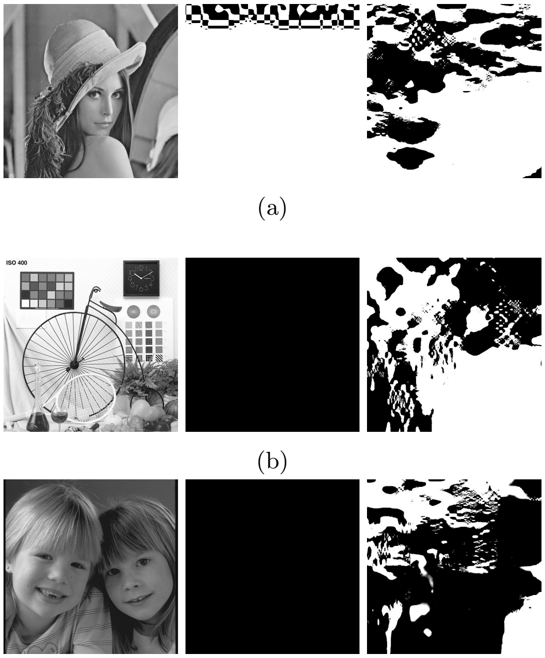
Figure 5: Proposed asymmetric encryption model-decryption process

Table 5: Data replacement attack of encrypted bit streams

PSNR	Bike	Girls	Lena
Inserting zeros	10.6713	9.6280	11.8726
Inserting ones	7.8563	14.8034	13.4597
Shifting over	10.3656	14.8034	14.3145
Inserting another bit-stream	9.6015	14.8034	12.0017

Table 6: Encryption data ratio compared with other existing methods

Method	Image	Encryption data ratio	Plaintext length (byte)	PSNR of decompressed image	Number of ECC encryption
Uhl[22, 21]	angiogram 512 × 512	20%	6554	9.90	234
Uhl[22, 21]	lena 512 × 512	20%	6554	9.77	234
Lian[18]	lena 128 × 128	13.4%	274	N/A	10
Lian[18]	village 512 × 512	16.8%	5505	N/A	197
Lian[18]	boat 256 × 256	10.8%	885	N/A	32
Brahimi[2]	radiological 256 × 256	11.6%	954	7.47	34
Salama[28]	bike 512 × 512	0.1%	32	9.48[2]	1.14
Salama[28]	barbara 512 × 512	0.1%	32	14.08[3]	1.14
Proposed Method	lena 512 × 512	0.09%	28	5.06	1
Proposed Method	girls 512 × 512	0.09%	28	7.34	1
Proposed Method	bike 512 × 512	0.09%	28	2.14	1



(a)

(b)

(c)

Figure 6: Decrypted images of EZW (middle) and SPIHT (right) sequences with and without a correct secret key (a)"lena" (b)"bike" (c)"girl"

The reference SPIHT compression codec used is developed by René Puchinger.

The compression bit rate is 1 bit per pixel for both EZW and SPIHT compression.

As shown in Figure 4 and Figure 5, the encryption process of our asymmetric encryption model reads and compresses one original image. The sequence analysis is performed in the compression process. Essential elements (h *bits*) within the header and the first $224-h$ bits within the data sequence are marked and then encrypted by the elliptic curve public-key encryption.

The decryption process reads the encrypted sequence and performs the sequence analysis to decide the plaintext location. The public-key decryption algorithm is then performed to recover the plaintext and merge it into the compression sequence. The decompression process generates a desired image or a severely destroyed image if the public-key decryption is not performed by a correct secret key. As listed in Figure 6, the decryption output images without a correct secret key represents no useful information at all.

To confirm the encrypted sequences provide security with a small plaintext length, the data replacement attack is tested also.

We assume the eavesdropper obtained an encrypted sequence with a small cyphertext, they can apply the data replacement [24] attacks to it by simply inserting constant bits or other sequence data. We performed several typical data replacement such as inserting zero, inserting one, shifting over and inserting another bitstream to the encrypted sequence. Table 5 illustrates the decompression image quality of these data replacement. The plaintext length is 224 bits, image size is 512 × 512, "bike", "girls" and "lena" testing images were tested. The poor decompression quality proved again that the encrypted part of the bitstream represented most important information of the image. Simple data replacements does not reveal any information.

To compare our proposed method with all the existing methods, the encryption parameters are listed in Table 6. Different images are selectively encrypted by different methods to be transmitted securely. The extremely poor reconstruction quality indicated that a satisfied security level can be achieved by encrypting by different data ratios. (Notice that, poor reconstruction quality can be shown by low PSNR or visually illustrated as well. The "N/A" in Table 6 means the reconstruction quality was visually illustrated but no PSNR values were measured.) The plaintext sizes are calculated by assuming all the images are compressed to one bit per pixel. Notice that the smaller size images require much lower plaintext size using same method. For the images in same size, our previous work [28] and this proposed method achieved good security by using much lower plaintext sizes. To apply common public-key encryption algorithms, such as elliptic curves encryption, our proposed method is the only one that enables public-key media content encryption without encrypting the original data repeatedly.

Common traditional media content encryption [2, 18, 21, 22, 28], simply selecting part of (for example, 20%) the original media content, are not designed to cooperate with asymmetric encryption. The plaintext size dramatically increases as the original image size increases. Targeting at finding only the essential part of the original image content,

Here we have formalized the key problem to asymmetric media content encryption and have solved it for the first time, which makes asymmetric media encryption practical.

5 Conclusion

As far as we know, this paper is the first paper that advocates using selective multimedia encryption with a public-key encryption scheme.

All other work in the selected encryption area have advocated to use symmetric encryption or some hybrid approach (use of public-key and symmetric key cryptosystems).

The key problem to asymmetric media content encryption has been formally defined and analyzed. Solutions to this problem have been proposed. The encryption security was tested by multiple methods such as image quality test, sequence correlation test and cryptographic security analysis. The public-key selective multimedia encryption model was proposed based on elliptic curve public-key encryption and two rate scalable compression techniques. Experimental results confirmed a successful implementation of a secured public-key selective multimedia encryption.

References

[1] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, *Recommendation for Key Management, Part 1: General (revised)*, NIST Special Publication, Citeseer, 2006.

[2] Z. Brahimi, H. Bessalah, A. Tarabet, and M. K. Kholladi, "A new selective encryption technique of JPEG2000 codestream for medical images transmission," in *IEEE 5th International Multi-Conference on Systems, Signals and Devices (IEEE SSD'08)*, pp. 1–4, 2008.

[3] Edward J Delp, Paul Salama, Eduardo Asbun, Martha Saenz, and Ke Shen, "Rate scalable image and video compression techniques," in *IEEE 42nd Midwest Symposium on Circuits and Systems*, vol. 2, pp. 635–638, 1999.

[4] N. F. El Fishawy and O. M. Abu Zaid, "Quality of encryption measurement of bitmap images with RC6, MRC6, and rijndael block cipher algorithms," *International Journal of Network Security*, vol. 5, no. 3, pp. 241–251, 2007.

[5] D. S. Abd Elminaam, H. M. Abdual-Kader, and M. M. Hadhoud, "Evaluating the performance of symmetric encryption algorithms," *International Journal of Network Security*, vol. 10. no. 3, pp. 216–222, 2010.

[6] J. B. Feng, I. C. Lin, C. S. Tsai, and Y. P. Chu, "Reversible watermarking: current status and key issues," *International Journal of Network Security*, vol. 2, no. 3, pp. 161–170, 2006.

[7] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Advances in Cryptology (CRYPTO'99)*, pp. 537–554, Springer, 1999.

[8] B. Furht, D. Socek, and A. M. Eskicioglu, *Fundamentals of Multimedia Encryption Techniques*, Multimedia Security Handbook, 2004.

[9] D. Giry and P. Bulens, "Keylength–cryptographic key length recommendation," 2009. (`http://www.keylength.com`)

[10] M. E. Hellman, "An extension of the shannon theory approach to cryptography," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 289–294, 1977.

[11] ECRYPT II and D. SYM, "Ecrypt II," 2011.

[12] I. A. Ismail, M. Amin, and H. Diab, "A digital image encryption algorithm based a composition of two chaotic logistic maps," *International Journal of Network Security*, vol. 11, no. 1, pp. 1–10, 2010.

[13] X. Kang, W. Zeng, and J. Huang, "A multi-band wavelet watermarking scheme," *International Journal of Network Security*, vol. 6, no. 2, pp. 121–126, 2008.

[14] A. K. Lenstra, *Key Lengths*, The Handbook of Information Security, Springer, 2004. (`https://infoscience.epfl.ch/record/164539/files/NPDF-32.pdf`)

[15] B. King, "Mapping an arbitrary message to an elliptic curve when defined over gf (2^n)," *International Journal of Network Security*, vol. 8, no. 2, pp. 169–176, 2009.

[16] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," in *Public Key Cryptography*, LNCS 1751, pp. 255–293, Springer, 2001.

[17] Z. N. Li, M. S. Drew, and J. Liu, "Image compression standards," in *Fundamentals of Multimedia*, pp. 281–315, Springer, 2014.

[18] S. Lian, J. Sun, D. Zhang, and Z. Wang, "A selective image encryption scheme based on JPEG2000 codec," in *Advances in Multimedia Information Processing (PCM'04)*, pp. 65–72, Springer, 2005.

[19] X. Liu and A. M. Eskicioglu, "Selective encryption of multimedia content in distribution networks: Challenges and new directions," in *Communications, Internet & Information Technology*, pp. 527–533, 2003.

[20] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 1996.

[2] Numbers are read from plotes in the paper [28].

[3] Numbers are read from plotes in the paper [28].

[21] R. Norcen, M. Podesser, A. Pommer, H. P. Schmidt, and A. Uhl, "Confidential storage and transmission of medical image data," *Computers in Biology and Medicine*, vol. 33, no. 3, pp. 277–292, 2003.

[22] R. Norcen and A. Uhl, "Selective encryption of the JPEG2000 bitstream," in *Communications and Multimedia Security*, LNCS 2828, pp. 194–204, Springer, 2003.

[23] J. Pieprzyk, T. Hardjono, and J. Seberry, *Fundamentals of computer security*, Springer Science & Business Media, 2013.

[24] M. Podesser, H. P. Schmidt, and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments," in *Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG'02)*, pp. 4–6, 2002.

[25] A. Pommer and A. Uhl, "Selective encryption of wavelet-packet encoded image data: Efficiency and security," *Multimedia Systems*, vol. 9, no. 3, pp. 279–287, 2003.

[26] M. Saenz, P. Salama, K. Shen, and E. J. Delp, "Evaluation of color-embedded wavelet image compression techniques," in *Electronic Imaging*, pp. 282–293, 1998.

[27] A. Said, W. Pearlman, et al, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.

[28] P. Salama and B. King, "Efficient secure image transmission: compression integrated with encryption," in *Electronic Imaging*, pp. 47–58, 2005.

[29] P. Salama, N. Shroff, and E. J. Delp, "Error concealment in embedded zerotree wavelet codecs," in *Proceedings of the International Workshop on Very Low Bit Rate Video Coding*, pp. 8–9, 1998.

[30] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *SIEEE Transactions on Ignal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[31] K. Shen and E. J. Delp, "Wavelet based rate scalable video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 109–122, 1999.

[32] T. Shi, B. King, and P. Salama, "Selective encryption for H.264/AVC video coding," in *Proceedings of SPIE*, vol. 6072, pp. 461–469, 2006.

[33] D. R. Stinson, *Cryptography: Theory and Practice*, CRC press, 2005.

[34] W. Trappe and L. C. Washington, *Introduction to Cryptography with Coding Theory*, Pearson Education India, 2006.

[35] M. V. Droogenbroeck and R. Benedett, "Techniques for a selective encryption of uncompressed and compressed images," in *Advanced Concepts for Intelligent Vision Systems (ACIVS'02)*, pp. 90–97, 2002.

[36] D. Wackerly, W. Mendenhall, and R. Scheaffer, *Mathematical statistics with applications*, Cengage Learning, 2007.

[37] C. P. Wu and C. C. Jay Kuo, "Efficient multimedia encryption via entropy codec design," in *Photonics West 2001-Electronic Imaging*, pp. 128–138, 2001.

[38] C. P. Wu and C. C. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 828–839, 2005.

[39] T. Xiang, J. Qu, C. Yu, and X. Fu, "Degradative encryption: An efficient way to protect spiht compressed images," *Optics Communications*, vol. 285, no. 24, pp. 4891–4900, 2012.

[40] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *MIEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.

Feng Jiang is a PhD student at Purdue West Lafayette. Her research interests include image processing, multimedia security, visual cryptography, and Watermarking.

Paul Salama received his Ph.D. from Purdue University in 1999. Research activities include Signal/image/video processing, biomed. image analysis, image/video compression, security, and restoration/reconstruction, digital communications, information theory and source coding, error resilience, and error concealment.

Brian King received a Ph.D. in Mathematics from University of Wisconsin-Milwaukee in 1990 and a Ph.D. in Computer Science from University of Wisconsin-Milwaukee in 2000. His research interests include Information security, computer & network security, wireless security, cryptography, Ubiquitous & Mobile Ad-hoc Network Security, algorithms, and applied mathematics.