Digital Watermark Embedding with Elliptic Curve Cryptography

Arieza Nadya
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
ariezanadya@gmail.com

Abstract—**The use of the internet has completely removed the barrier of distance as now people are able to communicate with each other despite the distance that once hinders interaction with others that live on the other side of the earth. Granted, the existence of blogs, image sharing sites, art sharing sites and the likes means that any person can express and publicize their works and ideas freely for little to no cost. This also means that any person can take a digital art off the internet and claims it as their own. To prevent this from happening, we can use a public-key cryptography to embed a signature image, in this case the owner's identification image to the digital image.**

Keywords—digital watermarking, public key algorithm, elliptical curve cryptography

I. INTRODUCTION

Art theft is a huge issue in the artistic community in the internet. It is mainly because digital images of arts are so easily shared and taken with the existence of internet, even more so since art sharing websites become popular. Unfortunately, this convenience also drives some people to claim someone else's work as their own. Lots of visual arts showcased on the internet comes in the form of digital image. That is to say, we are able to 'sign' a digital image with a digital watermark.

A digital watermark is, in the context of digital image, something that is embedded in the digital image, which also can be extracted from its digital image. In this sense, digital watermarking comes close with the techniques in steganography.

Wong [3] described a public key scheme in his paper to use digital watermarking with the RSA algorithm to ensure authenticity of a digital image. The type of watermark used in this scheme is fragile watermarking and so it will show if the image has been tampered with. It can also be used to prove ownership of a digital image because the signature image can only be extracted with the public key of the owner. The public key can be published without compromising the security of the system. Thus, anyone is able to verify if an image claimed by someone is indeed claimed by the rightful owner.

While RSA algorithm has been around for a while, ECC on the other hand is still a relatively new approach in the public key cryptosystem. Even so, ECC has some advantages over RSA which we will discuss on the second section.

II. FUNDAMENTAL THEORIES

A. Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is an approach to public-key cryptosystem which is based on an elliptic curve equation over a finite field [1]. An elliptic curve is defined in the following form (Weierstrass normal form):

$$y^2 mod\ p = (x^3 + ax + b)\ mod\ p$$

with the condition that the value of $4a^3 + 27b^2\ mod\ p$!= 0. Different values of a and b yields different elliptic curves. The elements of the finite field ranges between 0 and p-1. All points operations performed on an elliptic curve also involve only integers that ranges between 0 and p-1 (since it works in modular arithmetic). These points operations have specific set of rules:

a. Point addition

Addition of points in an elliptic curve is given by the following rule: given three aligned, non-zero points P, Q, and R, their sum is $P + Q + R = 0$. The points in the elliptic curve with the operator <+> is an abelian group so we can rewrite it as $P + Q = –R$.

Consider two non-zero, non-symmetric points $P = (x_p, y_p)$ and $Q = (x_q, y_q)$. If P and Q are distinct, the line through them has a slope, m, where m is the value of $\frac{yp-yq}{xp-xq}$. The intersection of the line with the elliptic curve is a point $R = (x_r, y_r)$, where $x_r = m^2 – x_p – x_q$ and

$y_r = m\ (x_p – x_r) – y_p$. Since $P + Q = –R$, so the value of $(x_p, y_p) + (x_q, y_q) = (x_r, –y_r)$.

b. Point doubling

Point doubling is essentially adding a point in an elliptic curve by itself. For example, let $P = (x_p, y_p)$ be a point

in an elliptic curve over *Fp*. The gradient of the line, *m*, is defined as $m = \frac{dy}{dx} = \frac{3x^2_p + a}{2y_p}$. The intersection of the line with the elliptic curve is a point $R = (x_r, y_r)$ where $x_r = m^2 - 2x_p$ and $y_r = m(x_p - x_r) - y_p$. If $y_p = 0$ then *m* is not defined, thus $2P = O$, where O is a point at infinity.

c. Scalar multiplication

Another operation that can be performed on the points on an elliptic curve is scalar multiplication, *nP*, that can be written as $nP = \sum_0^n P$. One way to do this is to add the value *P n* times, but this seemingly naïve method is not efficient given that *n* has *k* binary digits, the algorithm would be $O(2^k)$. A faster way to do this is to use both point addition and point doubling to aid the operation.

Elliptic Curve Cryptography's security is as strong as RSA cryptography with the advantage of shorter key. This fact makes it a good cryptography option for devices that have a relatively small memory. The security of ECC lies in what is said as the discrete logarithm problem. Given *n* and *P*, we have at least one polynomial time algorithm to compute *Q* = *nP*. But the other way around is significantly more difficult to compute.

Comparison between RSA and ECC performance is illustrated in Fig. 1

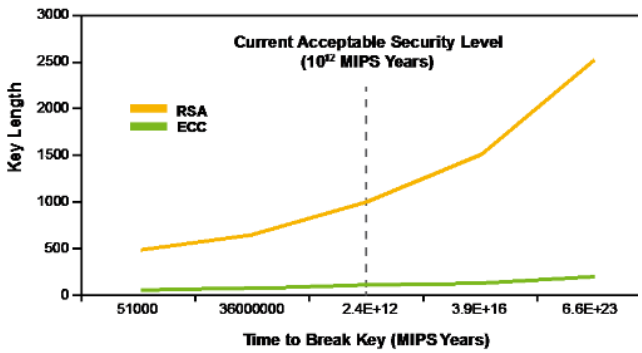

Figure 1 RSA and ECC performance [6]

The chart in Fig. 1 shows that for a given time needed to break a key, the ECC algorithm requires a much shorter key length compared with the key length needed in RSA. This on itself shows the superiority of ECC over RSA in terms of size. It is clear that ECC is more efficient than RSA.

B. MD5 Hash Function

The MD5 message digest algorithm is a one-way cryptographic hash function that is widely used. MD5 produced a 128-bit hash value generated from an arbitrary-length message. The application of a message digest such as MD5 is primarily to determine whether any changes are made to the original message by comparing the original message digest calculated before the transmission and after the transmission.

C. Digital Watermarking

Digital watermarking is a technique to insert a digital signature into an image so that the signature can be extracted for the purposes of ownership verification and/or authentication [3]. In this context we only want to use invisible watermark to sign our digital image, so the watermark needs to be inserted into the digital image itself. Digital watermarking employs steganographic techniques to embed data covertly in noisy signals. There are two general types of watermarking schemes that are used for different purposes, namely robust watermarking and fragile watermarking.

a. Fragile watermarking

A watermark is said to be fragile if the image hidden within the host signal is destroyed if the watermarked signal undergoes any manipulation [4].

With that fact, fragile watermarking is used for detecting if a manipulation has been performed on a digital image, which can also be used as a way to prove originality of the image. It is used mainly to ensure the integrity of a digital image.

b. Robust watermarking

Robust watermarking on the other hand has the purpose of protecting copyright of a digital image. This is due to the fact that robust watermark withstands common digital processing that is performed on a digital image, as opposed to fragile watermarking.

Image watermarking has different methods of embedding which can be classified into two categories below:

a. Spatial domain

A digital watermarking method is referred to as spatial method if it is embedding the watermark directly to the byte values of the digital image pixels.

b. Transform domain

A digital watermarking method is referred to as transform method (or in frequency domain) if it is embedding the watermark on the transformation coefficient of the digital image.

The following table describes the comparison between watermarking techniques in the spatial domain and techniques in the frequency domain.

TABLE I. SPATIAL DOMAIN AND FREQUENCY DOMAIN COMPARISON

	Spatial	**Frequency**
Computational Cost	Low	High
Robustness	Fragile	More robust
Perceptual Quality	High control	Low control
Capacity	High (depends on the size of the image)	Low
Example of Applications	Mainly authentication	Copyrights

[a] *Taken from* [4]

With the comparison in mind, we want our watermark to be fragile in order to expose any manipulation that the image undergoes. We are going to use a watermarking technique in the spatial domain to achieve this.

One digital watermarking technique in the spatial domain is the LSB (Least Significant Bit) technique. It is also one of the simplest watermarking techniques. LSB works by using two dimensional array of pixels in the container image to hold hidden data [5]. This method works by the fact that human eyes cannot detect small variance in color. Therefore, small differences in pixel values caused by LSB embedding will be hardly noticeable.

III. Proposed Solution

In this section we describe the mechanism of the Elliptic Curve Cryptography implementation to insert a digital watermark in the form of a binary image or a bi-level image into another 8-bit grayscale image. The general overall process is described in Wong's paper [3] but instead we will use ECC to insert our watermark image.

We consider a scenario of an 8-bit grayscale image $A_{m,n}$ where m and n are the dimension of the image. The watermark would be another image $B_{m,n}$, where m and n are the dimension of the watermark image and B is a binary image. In our example $B_{m,n}$ will have a smaller size compared to $A_{m,n}$. From $B_{m,n}$ we can form another image $B'_{m,n}$ with identical size to $A_{m,n}$ by tiling or repeating $B_{m,n}$ until we reaches the desired size, that is, the size of $A_{m,n}$. The image $B'_{m,n}$ will then be partitioned into blocks of i by j pixels. Then each block of $B'_{m,n}$ would be inserted to the corresponding block in image $B_{m,n}$ yielding the final watermarked image $A'_{m,n}$.

As opposed to using RSA for our public key cryptography, we will use ECC to embed the watermark image into the original image. Since ECC algorithm works with points, we first need to convert the bits into points before encrypting the bits, and convert it back to bits before decrypting. This can be achieved by using Koblitz's method of encoding and decoding message [7]. For the hash function we will use the widely used MD5.

The domain parameters of the elliptic curve we will use for our algorithm will be:

$a = -1$

$b = 188$

$p = 98764321261$

A. Embedding Watermark

The steps to embed a watermark in the image are the following:

1. Define X, blocks of data within the image $A_{m,n.}$
2. Define X', blocks of data which is nearly identical to X, only the least significant bits are set to zero.
3. Compute the hash $H(m, n, X') = (p_1^r, p_2^r, \ldots, p_s^r)$ where p_i^r denotes the output bits from the hash function, s is the size of the output bits. We are using MD5, in which case the value of s is 128.
4. Define P_r as the first $i\ j$ bits from the bit stream
5. XOR P_r with the block of watermark
6. Encode the bits produced in step 5 to points
7. Encrypt the encoded bits
8. The binary blocks of data produced in step 7 is then embedded to the LSB of X' blocks

The diagram in fig. 2 illustrates the process in which the watermark is being embedded/encrypted into the digital image.

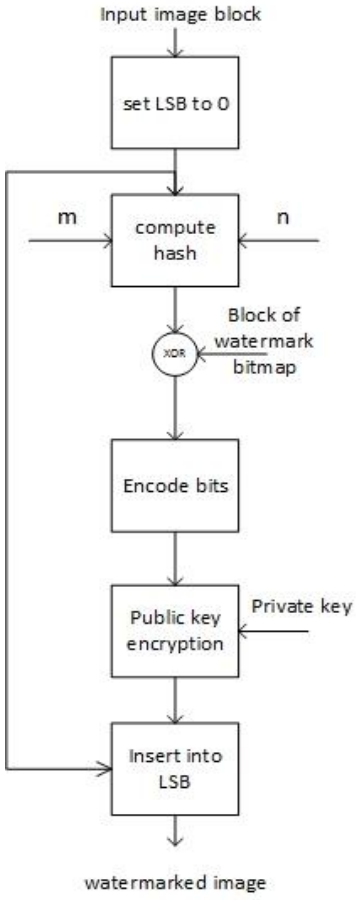

Figure 2 Inserting watermark

B. Extracting Watermark

Steps to extract the watermark back is described as below:

1. Split the input image block into two pieces, Z_r, that contains the LSBs, and G_r, that contains the pixel values with the LSBs set to zero.
2. Compute the hash $H(m, n, Z_r)$
3. Define Q_r as the first 64 bits of the hash output
4. Decrypt G_r by using the public key that corresponds to the private key used in embedding watermark

5. Decode it back again to the original bits
6. Compute the output by applying XOR between the output produced in step 5 and Q_r.

The diagram in fig. 3 illustrates the process of extracting/decrypting the watermark from the original image.

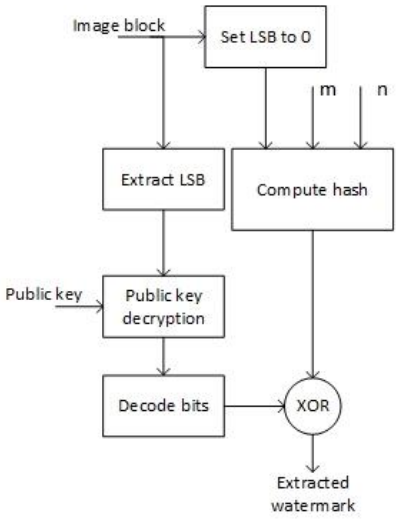

Figure 3 Extracting watermark

The extracted watermark, if decrypted using the correct public key will yield the watermark image. If decrypted using other public key, the resulting image would look like a random noise. This way ownership of an image can be verified by using just the public key.

This approach adds an additional process to encode and decode the bits as opposed to just encrypt and decrypt the bits as the original method using RSA but still provides better security for the same key length.

IV. Conclusion

This paper has presented the implementation of a digital watermark embedded with public key cryptography scheme by using elliptic curve cryptography. Compared with the original technique which uses RSA algorithm, the use of elliptic curve cryptography gives a smaller solution of encryption. This scheme also keeps the authentication and verification purpose of the digital watermark.

While the implementation on this paper is only applicable to grayscale images, application on RGB images can also be done by independently applying the method to the individual color planes of the image.

References

[1] Kapoor, et. al, "Elliptic Curve Cryptography", ACM Ubiquity, Volume 9, Issue 20.

[2] Padma Bh, et. al, "Encoding and Decoding of a Message in the Implementation of Elliptic Curve Cryptography using Koblitz's Method", International Journal on Computer Science and Engineering Vol. 02, No. 05, 2010, 1904-1907

[3] Wong, Ping Wah, "A Public Key Watermark for Image Verification and Authentication"

[4] El-Gayyar, Mahmoud, "Watermarking Techniques Spatial Domain Digital Rights Seminar"

[5] M. Nilay B, D. Dhruv, "Digital Watermarking Methods in Spatial Domain and Transform Domain," in International Journal for Scientific Research & Development, Vol. 1, Issue 10, 2013

[6] M. Alimohammadi, an A.A. Pouyan. "Performance Analysis of Cryptography Methods for Secure Message Exchanging in VANET," www.ijser.org/paper/Performance-Analysis-of-Cryptography-Methods-for-Secure.html. 2014.

[7] Bh. Padma, D. Chandravati, R. P. Prapoorna, "Encoding and Decoding of a Message in the Implementation of Elliptic Curve Cryptography using Koblitz's Method," IJCSE, Vol. 02, No. 05, 2010, 1904—1907.