Journal of the Korea Academia-Industrial cooperation Society
Vol. 23, No. 11 pp. 608-614, 2022

https://doi.org/10.5762/KAIS.2022.23.11.608
ISSN 1975-4701 / eISSN 2288-4688

디지털 서명과 메시지 인증코드를 이용한 영상의 무결성 검증에 관한 연구

우찬일
서일대학교 정보통신공학과

A Study on the Image Integrity Verification using Digital Signature and Message Authentication Code

Chan-Il Woo
Department of Information and Communication Engineering, Seoil University

요 약 디지털 멀티미디어 콘텐츠는 쉽게 조작 되거나 복사 될 수 있기 때문에 디지털 멀티미디어 콘텐츠에 대한 출처와 변형 여부를 확인하는 것이 매우 중요하다. 이를 위해 디지털 멀티미디어 콘텐츠에 대한 소유권 증명이나 무결성을 검증하기 위한 목적으로 시각적으로 인지 할 수 없는 정보를 디지털 멀티미디어 콘텐츠에 삽입 할 수 있는 디지털 워터마킹 기술이 제안되었다. 본 논문에서는 영상에 대한 인증과 무결성을 검증하기 위해 영상 블록을 기반으로 하여 디지털 워터마크를 삽입하는 방법을 제안한다. 제안 방법에서는 원 영상을 16×16 크기를 갖는 여러 블록으로 분할한 후 각 블록을 다시 4개의 8×8 서브 블록으로 세분화하여 분할된 영상 블록을 이용하여 생성된 디지털 서명과 메시지 인증 코드를 블록 내에 삽입한다. 제안 방법은 특정 블록을 복사하여 붙여 넣는 콜라주 공격을 확인할 수 있고 변형이 발생된 영역을 쉽게 검출할 수 있는 장점이 있다. 그리고 제안 방법의 안전성은 암호학적 알고리즘의 안전성에 의존하고 있다.

Abstract Because digital multimedia content can easily be manipulated and duplicated, it is important to ensure that multimedia content is original and not manipulated. For this purpose, digital watermark technology has been proposed to insert visually unrecognizable information into digital multimedia content for the purpose of verifying ownership or ensuring the integrity of digital multimedia content. This paper proposes a block-based digital watermark method for image authentication and integrity verification. The proposed method divides the original image into 16×16 non-overlapping pixel blocks, with each block further subdivided into four 8×8 sub-blocks. The digital signature or message authentication code generated from the image block is inserted. The proposed method can detect a collage attack, and it is possible to easily detect manipulated areas. The safety of the proposed method depends on the security of the cryptographic algorithm.

Keywords : Integrity, Authentication, Tamper detection, Digital Signature, HMAC.

1. 서론

디지털 워터마킹은 디지털 콘텐츠에 대한 저작권 보호나 인증과 무결성을 검증하기 위한 방법으로 연구가 이루어져 왔다. 저작권을 보호하기 위한 방법에서는 디지털 콘텐츠에 삽입된 워터마크가 저작권자의 워터마크임을 증명할 수 있어야 한다. 그러나 워터마크가 삽입된 영상에 대하여 필터링과 같은 다양한 후처리 기술을 이용

본 논문은 서일대학교 학술연구비에 의해 연구되었음.
*Corresponding Author : Chan-Il Woo(Seoil Univ.)
email: ciwoo@seoil.ac.kr
Received September 13, 2022 Revised October 21, 2022
Accepted November 4, 2022 Published November 30, 2022

할 경우 삽입된 워터마크를 쉽게 제거할 수 있는 문제점이 발생한다. 따라서 저작권 보호 기술에서는 이러한 문제점을 해결하는 것이 매우 중요하며 이를 위해 다양한 연구가 진행되고 있다[1]. 디지털 워터마킹에 대한 또 다른 연구 분야로는 디지털 콘텐츠에 대한 진위 여부를 확인하기 위한 무결성 검증 방법이 있다. 무결성 검증 기술에서는 워터마크가 삽입된 영상에 작은 변화라도 발생할 경우 이를 감지할 수 있어야 한다.

따라서 디지털 워터마킹의 저작권 보호 기술에서는 삽입된 워터마크가 다양한 공격으로부터 최대한 훼손되지 않아야 하는 것이 중요하다면, 무결성 검증을 위한 워터마킹에서는 영상에 공격이 발생하였을 경우 삽입된 워터마크가 쉽게 파괴되어 변형 여부를 감지하는 것이 중요하다[2-5]. 이 두 가지 기술은 디지털 콘텐츠에 워터마크를 삽입한다는 측면에서는 유사한 기술로 볼 수 있으나 활용하는 측면에서는 매우 다른 특성을 가지고 있다. 일반적으로 저작권을 보호하기 위한 기술에서는 주파수 영역에서 워터마크를 삽입하는 것이 공간영역에서 삽입하는 것보다 강인성을 향상시킬 수 있다. 그러나 무결성을 검증하기 위한 기술은 삽입된 워터마크가 쉽게 부수어져야 되는 특성을 가져야하기 때문에 주파수 영역뿐만 아니라 공간영역에서도 많은 연구가 이루어지고 있다[1,6].

공간영역에서 워터마크를 삽입하는 방법 중 대칭키 암호를 사용하여 워터마크를 삽입하는 방법에서는 워터마크를 검증하기 위해, 워터마크 생성에 사용된 비밀키를 알아야 하기 때문에 키가 노출될 수 있는 단점이 있다. 이러한 문제를 해결하기 위해 공개키 암호를 이용한 방법에서는 개인키로 워터마크를 생성하고 공개키로 검증하기 때문에 워터마크 삽입에 사용된 개인키를 알아내는 것이 매우 어려운 장점이 있다. 그러나 키의 안전성 보장을 위해 키의 길이가 길어질수록 생성되는 워터마크의 길이도 길어질 수 있기 때문에 영상 블록에 저장할 경우 블록의 크기가 커지게 되는 단점이 있다. 따라서 삽입되는 워터마크가 작은 크기의 블록에 삽입될 수 있다면 공개키 암호를 사용하는 방법이 대칭키 암호를 사용하는 방법보다 매우 효과적일 수 있다[7-9]. 워터마크를 생성하기 위해 사용되는 개인키와 검증을 위한 공개키가 서로 다른 장점을 활용하면서 공개키 암호의 단점인 블록의 크기를 줄일 수 있는 방법으로 디지털 서명을 이용하는 방법이 있다. 디지털 서명은 해시 함수와 공개키 암호를 이용하여 생성할 수 있으나 DSA, KCDSA, Schnorr 서명 등을 이용할 경우 보다 작은 크기의 디지털 서명을 얻을 수 있다. 따라서 입력 데이터의 크기가 크더라도 작은 크기의 서명을 생성하기 때문에 블록 단위로 구성되는 워터마킹에 효과적으로 활용할 수 있다.

본 논문에서는 전체 영상을 작은 크기를 갖는 여러 블록으로 나누고 각각의 블록 정보를 이용하여 디지털 서명과 메시지 인증 코드를 생성한 후 블록 내의 하위 2개의 LSB에 삽입하는 방법을 제안한다. 제안 방법에서 블록의 변형 여부는 삽입된 디지털 서명을 공개키로 검증하고, 변형이 발생된 블록은 작은 크기의 서브 블록으로 분할하여 변형 위치를 확인한다. 제안 방법에서는 암호학적으로 안전한 디지털 서명과 메시지 인증 코드를 이용하여 인증과 무결성을 검증할 수 있는 방법을 제안하며, 이를 위해 기존 방법들에 대한 문제점을 분석하고 해결하기 위한 방법을 제시한다.

2. 관련 연구

2.1 블록 기반 워터마킹

무결성을 검증하기 위한 워터마킹은 워터마크를 제거하기 위한 공격이 발생될 경우 삽입된 워터마크가 쉽게 부수어져야 한다. 이를 위해 공간영역 뿐만 아니라 주파수 영역에서 다양한 방법들이 연구되고 있으며, 일반적으로 공간영역에서 워터마크를 삽입할 경우 주파수 영역 방법보다 워터마크 제거가 용이할 수 있기 때문에 무결성 검증을 위한 다양한 방법들이 제안되고 있다.

공간영역 워터마킹에서는 워터마크 삽입을 위해 화소의 비트 값을 변경하게 되는데 화질 저하를 최소화하기 위해서는 최하위 비트에 워터마크를 삽입하는 것이 가장 효과적이다. 공간영역 워터마킹 중 영상을 특정 크기로 분할하여 블록 단위로 워터마크를 삽입하는 방법에서는 블록 정보만으로 워터마크를 생성할 경우 임의로 조작하는 것이 가능할 수 있기 때문에 암호학적으로 안전한 해시 함수나 디지털 서명 그리고 공개키 암호 등을 이용하면 이러한 문제를 해결할 수 있다.

공개키 암호를 이용하여 워터마크를 생성할 경우 RSA 공개키 암호에서는 암호문의 안전성을 위해 Eq. (1)에서 N의 크기를 최소 2,048 비트 이상으로 사용해야 한다. 이 경우 나머지 연산의 특성으로 인해 생성되는 암호문의 길이는 최대 N의 길이와 같은 2,048 비트가 될 수 있다. 만약 2,048 비트를 블록 내의 LSB에 삽입한다면 블록은 최소 46×46 이상의 크기를 가져야 한다. 그리고 화소의 하위 두 번째 LSB까지 확장하여 저장하더라도 32×32 이상의 크기를 가져야 2,048 비트를 저장할 수 있다.

$$C = M^E \bmod N \quad (1)$$

Where, M denotes message, E denotes encryption key, N denotes prime number × prime number

블록 단위로 수행되는 워터마킹에서는 블록의 크기를 크게 할 경우 많은 양의 정보를 저장할 수 있는 장점은 있으나 블록 내에서 일부 화소만 변형 되더라도 항상 큰 크기의 블록 단위로만 변형 위치가 검출되는 단점이 있다. 따라서 블록의 크기가 작으면 작을수록 변형 위치를 세부적으로 확인할 수 있는 장점이 있으나 워터마크의 크기가 제한되어 암호학적으로 안전한 공개키 암호의 사용이 어려울 수 있는 단점이 있다.

예를 들면, RSA 공개키 암호를 사용하여 8×8 블록의 LSB에 워터마크를 삽입할 경우 최대 64비트의 암호문만 저장할 수 있다. 이 경우 Eq. (1)에서 N은 작은 값을 사용해야 되는데 RSA 공개키 암호에서는 2,048 비트 이상의 키를 사용해야 개인키의 안전성이 보장될 수 있다고 알려져 있다. 따라서 공개키 암호를 사용할 경우 8×8 크기의 블록에 워터마크를 삽입하게 되면 쉽게 공격이 가능하여 워터마크 생성에 사용된 개인키를 찾을 수 있으며, 이 경우 공격자가 임의로 워터마크를 생성하여 삽입할 수 있는 문제점이 발생한다. 따라서 공개키 암호를 사용할 경우 워터마크 생성에 사용된 개인키의 안전성 보장이 가능하고 작은 크기를 갖는 블록에도 삽입할 수 있는 방법이 필요하다.

2.2 메시지 인증 코드

메시지의 무결성 검증은 암호학적으로 안전한 해시 함수(hash function)를 많이 사용하고 있으나 해시 함수는 메시지의 송신자를 인증하는 것이 불가능한 단점이 있다. 이러한 단점을 해결하기 위해 메시지 인증 코드(MAC: Message Authentication Code)가 개발되었으며, Fig. 1의 과정으로 생성되는 메시지 인증 코드는 메시지에 대한 무결성 검증과 Key를 통한 송신자 인증이 가능하다.

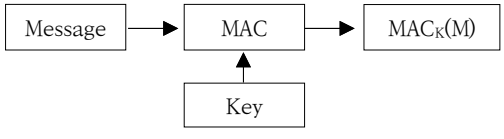

Fig. 1. Message authentication code generation process

메시지 인증 코드는 항상 일정한 크기의 출력을 생성하는 해시 함수와 유사하나 송신자 인증을 위해 송신자와 수신자 두 사람만이 공유하는 비밀키를 입력으로 사용하여 고정된 길이의 출력을 생성하기 때문에 메시지에 대한 무결성 검증과 송신자 인증이 가능하다[10].

메시지 인증 코드 중 해시 함수를 이용하여 생성된 메시지 인증 코드를 HMAC(Hash-based Message Authentication Code)이라고 하며, HMAC은 암호학적으로 안전하다고 알려진 모든 해시 함수를 사용할 수 있다. 본 논문에서는 디지털 서명이 삽입된 블록에 변형이 발생되었을 경우 변형 위치를 8×8 블록 단위로 확인하기 위해 HMAC을 이용한다.

2.3 디지털 서명

메시지 인증 코드는 송, 수신자가 공유하고 있는 비밀키를 이용하여 생성하기 때문에 부인방지와 제3자에 의한 인증이 불가능하다. 따라서 이러한 문제를 해결하기 위해 디지털 서명이 제안되었다. 디지털 서명은 메시지에 직접 서명하는 방법과 메시지의 해시 코드(해시 값)에 서명하는 방법으로 나눌 수 있으며, 메시지에 직접 서명하는 방법에서는 송신자의 개인키로 메시지를 암호화하여 디지털 서명을 생성하고 공개키로 복호화하여 검증한다. 이 방법은 메시지 전체를 암호화해야 되기 때문에 공개키 암호 알고리즘을 수행하는데 많은 시간이 소모되고 서명문의 크기가 커지는 단점이 있다.

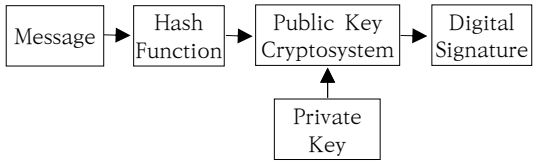

Fig. 2. Digital signature generation process

그러나 Fig. 2와 같이 메시지의 해시 코드를 이용하여 서명을 생성할 경우 짧은 길이의 해시 코드에 암호화를 수행하기 때문에 메시지에 직접 서명하는 방법과 비교하면 디지털 서명을 생성하는데 필요한 시간이 줄어들고 서명의 길이도 짧아질 수 있는 장점이 있다. 디지털 서명은 위의 방법 외에도 DSA나 KCDSA 그리고 Schnorr 서명 알고리즘을 이용할 경우 서명의 크기는 알고리즘마다 다를 수 있으나 작은 크기의 디지털 서명을 얻을 수 있다. Table 1은 위의 3가지 방법들에 대한 비교를 나타내고 있다.

Table 1. Compare the three methods

	Block Size	Safety
RSA	Big	Good
HMAC	Small	Middle (There is the possibility of losing the secret key)
Digital Signature Algorithm	Small	Good

3. 블록구조 및 워터마크 생성

3.1 블록 구조

본 논문에서는 두 종류의 워터마크를 사용한다. 첫 번째 워터마크는 Fig. 3의 1번부터 64번까지 16×16 크기를 가지는 블록에 대한 변형 여부를 검사하기 위한 워터마크로써 Fig. 4의 정보를 이용하여 생성하고 해당 블록의 LSB에 삽입한다. 이를 위해 전체 영상의 하위 두 개의 LSB를 0으로 초기화한 후 블록 단위로 워터마크를 생성하여 삽입한다. 따라서 첫 번째 워터마크를 검사하면 16×16 블록에 대한 변형 여부를 확인할 수 있다.

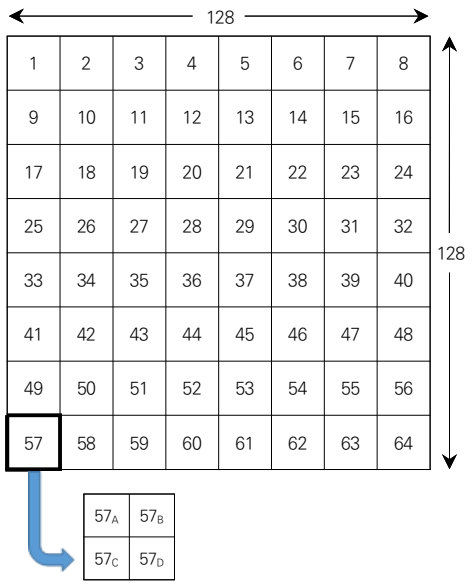

Fig. 3. Image block for watermark generation and embedding

Fig. 3은 128×128 크기의 영상을 16×16 크기의 블록으로 분할한 것을 나타내고 있으며, 이 경우 그림과 같이 64개의 블록으로 나눌 수 있다. 각 블록은 57번 블록의 57_A, 57_B, 57_C, 57_D 와 같이 8×8 크기를 갖는 4개의 서브 블록으로 구성할 수 있다.

Initialized block (16×16)	Block number	Mark

Fig. 4. Information for generating digital signature

16×16 블록에 변형이 발생 되었을 경우 해당 블록은 8×8 크기를 갖는 4개의 서브 블록으로 분할하여 변형 위치를 검사하게 된다. 이를 위해 서브 블록 정보를 이용하여 서브 블록에 삽입될 워터마크를 생성하고 해당 서브 블록의 하위 두 번째 LSB에 삽입한다. 이와 같이 두 종류의 워터마크를 삽입하게 되면 전체 영상에 대한 변형 여부와 변형이 발생된 부분을 16×16 블록 단위로 빠르게 찾을 수 있으며, 변형이 발생된 블록은 8×8 크기의 서브 블록 단위로 검사를 수행하기 때문에 변형이 발생된 부분을 보다 작은 영역으로 검출할 수 있는 장점이 있다.

3.2 블록 검증을 위한 워터마크

16×16 블록의 변형 검출을 위한 워터마크는 Fig. 4와 같이 초기화된 블록과 블록 번호 그리고 마크를 개인키와 함께 입력으로 사용하여 디지털 서명을 생성하고 블록 내의 LSB에 삽입한다. 각각의 블록은 동일한 키를 사용하여 서명을 생성하지만 생성되는 서명은 블록마다 서로 다른 값을 가지며, 디지털 서명 알고리즘에 따라 서명의 길이는 다를 수 있다. 디지털 서명 알고리즘 중 DSA의 경우 SHA1 해시 함수를 사용할 경우 작은 크기의 서명(r, s)이 생성되기 때문에 블록 내 LSB와 하위 두 번째 LSB 일부에 삽입할 수 있다. 그러나 서브 블록 검증을 위한 워터마크를 포함하여 하위 2개의 LSB에 삽입이 불가능할 경우에는 서명 중 하나를 LSB에 삽입하고 다른 하나는 공개키와 함께 공개한다.

3.3 서브 블록 검증을 위한 워터마크

영상에 대한 변형 유, 무는 16×16 블록 단위로 검사하고 변형이 발생된 블록에 대해서는 서브 블록(8×8) 단위로 변형 위치를 검사한다. 이를 위해 생성되는 워터마크는 서브 블록 내의 하위 두 번째 LSB에 삽입하며 다음과 같이 생성한다.

① 초기화된 서브 블록과 서브 블록 번호 그리고 비밀키를 입력으로 사용하여 HMAC을 생성한다.

② 메시지 인증 코드의 길이는 메시지 인증 코드 생성에 사용된 해시 함수에 의해 결정된다. 예를 들면

SHA-256 해시 함수를 이용할 경우 총 256 비트의 HMAC이 생성된다. 256 비트의 HMAC을 서브 블록에 저장하기 위해서는 하위 두 번째부터 다섯 번째까지의 공간이 필요하고, 이 공간에 HMAC을 삽입하게 되면 원 영상 정보의 손실이 매우 크게 발생하는 문제점이 있다. 따라서 HMAC을 Fig. 5와 같이 8비트씩 나눈 후 각 8비트 블록에 대해 XOR 연산을 수행하고 그 결과(W_1)를 블록 내 화소의 두 번째 LSB에 삽입한다.

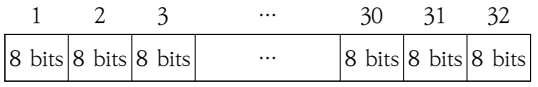

Fig. 5. Message authentication code

③ 서브 블록 화소의 하위 두 번째 LSB에 삽입되는 워터마크는 ②에서 구한 워터마크를 포함하여 8비트 정보 4개가 삽입된다. 따라서 나머지 3개의 워터마크는 Fig. 6과 같이 초기화된 현재 블록을 기준으로 이웃한 서브 블록을 해시 함수의 입력으로 사용하여 생성한다. 이 과정은 이웃한 서브 블록에 대해 모두 수행하고, 생성된 해시 코드는 ②의 과정처럼 8비트로 축약한다. 즉, 서브 블록 A에 삽입되는 정보는 초기화된 서브 블록 A와 이웃한 서브 블록(B, C, D)을 각각 입력으로 사용하여 생성된 8비트 정보(W_2, W_3, W_4)이다.

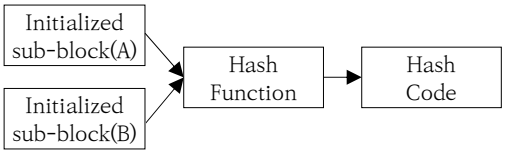

Fig. 6. Sub-block watermark generation process

이 과정은 삽입되는 각각의 서브 블록을 기준으로 모든 서브 블록들에 대해 수행한다. 따라서 A부터 D까지 각 서브 블록의 워터마크는 W_1부터 W_4까지 총 32비트로 구성되고 서브 블록의 하위 두 번째 LSB에 삽입한다. Fig. 7은 워터마크 삽입 과정에 대한 블록도를 나타낸다.

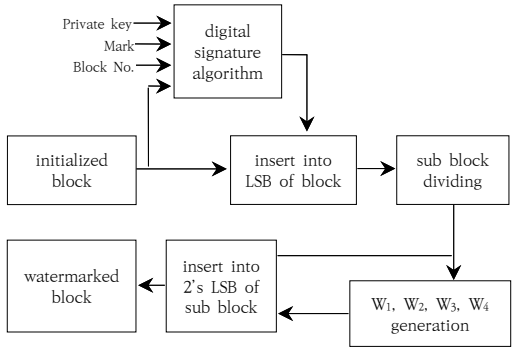

Fig. 7. Watermark generation and insertion process

4. 워터마크 추출 및 분석

워터마크가 삽입된 영상의 변형 유, 무와 변형 위치는 다음과 같이 검출한다.

① 전체 영상을 워터마크 생성 과정과 동일하게 16×16 블록으로 분할한 후 블록 내 하위 두 개의 LSB에서 워터마크를 추출하고 0으로 초기화한다.

② 16×16 블록의 LSB에서 추출한 디지털 서명을 공개키와 Fig. 4의 정보를 이용하여 검증하고 블록의 변형 유, 무를 확인한다. 추출된 디지털 서명이 유효하면 변형이 발생되지 않은 것으로 판정하고 검사를 종료한다. 그러나 디지털 서명이 유효하지 않으면 변형이 발생된 블록으로 판정하고 다음 단계를 수행한다.

③ 변형이 발생된 블록을 네 개의 서브 블록으로 분할한 후 각 서브 블록의 하위 두 번째 LSB에 삽입된 워터마크를 생성하는 방법과 동일한 과정으로 W_1을 생성하고 추출된 값과 비교한다. 비교 결과 서로 다른 값을 가질 경우 변형이 발생된 블록으로 판단한다. 그러나 모든 서브 블록의 값이 동일할 경우에는 W_2부터 W_4를 비교하여 일치하는 값이 하나도 없는 블록을 변형이 발생된 블록으로 판정한다.

Fig. 8은 256×256 영상에 변형이 발생하였을 경우 검출될 수 있는 최대 크기의 블록을 나타낸다. Fig. 8의 (c)는 16×16 크기를 나타내고 2,048 비트 RSA 공개키 암호를 사용할 경우 (d)와 같이 64×64 크기로 검출될 수 있다.

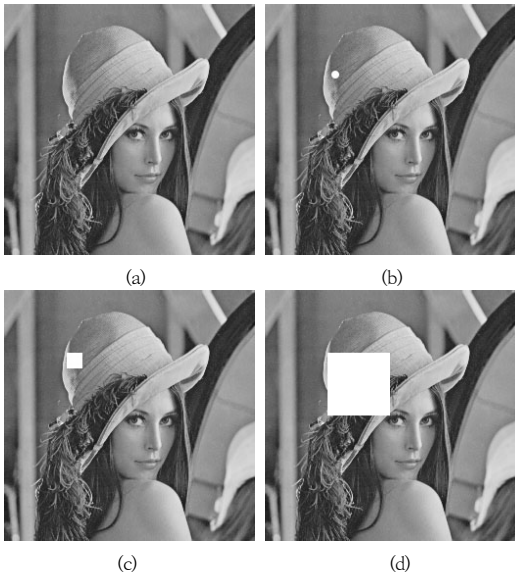

Fig. 8. Compare detectable areas: (a) original image, (b) tampered image, (c), (d) detectable area

DSA, Schnorr, KCDSA 등의 디지털 서명 알고리즘을 이용하면 워터마크 생성에 사용되는 개인키의 안전성을 보장할 수 있으면서 작은 크기의 블록으로 변형 영역을 검출할 수 있는 장점이 있다. 그러나 RSA 공개키 암호로 생성되는 디지털 서명은 안전성을 위해 키의 길이가 길어질수록 암호문의 길이가 길어지기 때문에 워터마크를 저장하기 위한 블록의 크기가 커지게 되는 단점이 있다.

따라서 블록이 작으면 작을수록 작은 변화를 감지하는데 효과적이다. 그리고 원 영상의 크기는 블록 크기의 정수 배가 되어야 워터마크를 삽입하는데 효과적이다. 2,048 RSA를 사용할 경우 블록의 크기는 46×46이 가장 적합하나, 이 경우 영상의 크기가 블록 크기의 정수 배가 되지 않기 때문에 서로 다른 크기의 블록이 생성되어 특정 패턴으로 워터마크를 삽입하는 것이 어려울 수 있다. 따라서 이러한 문제를 해결하기 위하여 블록 크기에 따른 삽입 방법의 연구가 필요하다.

5. 결론

본 논문에서는 영상의 인증과 무결성 검증을 위해 암호학적으로 안전한 디지털 서명을 이용하여 블록 단위로 변형 유, 무를 확인하고 변형이 발생된 블록에 대해서는 작은 크기의 서브 블록으로 분할한 후 삽입된 메시지 인증 코드로 변형이 발생된 부분을 확인할 수 있는 방법을 제안하였다. 공개키 암호를 이용하여 워터마크를 생성하는 방법에서는 워터마크를 생성하는 개인키와 워터마크를 검증하는 공개키가 서로 다른 장점이 있으나 워터마크 삽입을 위해 블록의 크기가 작아질 경우 워터마크의 안전성 보장이 어려워 질 수 있기 때문에 블록의 크기가 커지게 되는 문제점이 발생한다. 그러나 디지털 서명을 이용한 제안 방법에서는 이러한 문제를 해결할 수 있기 때문에 공개키 암호를 이용한 방법보다 안전성과 효율성 측면에서 우수한 장점이 있다. 향후 연구과제로는 암호문을 해독하지 않고도 연산이 가능한 동형 암호 또는 준동형 암호를 이용한 방법과 다양한 크기를 갖는 영상에 대한 최적화된 블록 구성과 워터마크 삽입 방법에 대한 연구가 필요할 것으로 생각된다.

References

[1] H. S. Kim, Digital Watermarking, Green, pp.242-354, 2005.

[2] B. W. R. Agung, K. Adiwijaya, F. P. Permana, "Medical image watermarking with tamper detection and recovery using reversible watermarking with LSB modification and run length encoding (RLE) compression" IEEE Conf. on ComNetSat, pp. 167-171, 2012.
DOI: https://doi.org/10.1109/ComNetSat.2012.6380799

[3] H. Zhang, C. Wang, X. Zhou, "Fragile Watermarking Based on LBP for Blind Tamper Detection in Images" JIPS, Vol. 13, no. 2, pp. 385-399, 2017.
DOI: https://doi.org/10.3745/JIPS.03.0070

[4] M.U.Celik, G.Sharma, A.M.Tekalp, E.Saber, "Localized Lossless Authentication Watermark (LAW)," Proc. of SPIE-IS&T Electronic Imaging, Vol. 5020, pp. 689-698, 2003.
DOI: https://doi.org/10.1117/12.477312

[5] C. M. Wu, Y. S. Shin, "A Simple Image Tamper Detection and Recovery Based on Fragile Watermark with One Parity Section and Two Restoration Sections," Optics and Photonics Journal 3, pp. 103-107, 2013.
DOI: https://doi.org/10.4236/opj.2013.32B026

[6] P. L. Lin, C. K. Hsieh, P. W. Huang, "A hierarchical digital watermarking method for image tamper detection and recovery" Pattern Recognition 38, pp. 2519-2529, 2005.
DOI: https://doi.org/10.1016/j.patcog.2005.02.007

[7] G. Kaur, K. Kaur, "Image Watermarking using LSB," International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3,

no. 4, pp. 858-861, April 2013.

[8] P. W. Wong, “A Public Key Watermark for Image Verification and Authentication,” in Proc. of IEEE Conf. on Image Processing, pp. 425-429, 1998. DOI: https://doi.org/10.1109/ICIP.1998.723526

[9] H. Nyeem, W. Boles and C. Boyd, "Counterfeiting Attacks on Block-Wise Dependent Fragile Watermarking Schemes" in Proc. of the 6th International Conference on Security of Information and Networks, ACM Press and Digital Library, 2013. DOI: https://doi.org/10.1145/2523514.2523530

[10] J. K. Lee, T. I. Jeon, J. S. Jo, Information security and cryptography, p.585, infinity books, 2017, pp.309-341.

우 찬 일(Chan-Il Woo) [종신회원]

- 1995년 2월 : 단국대학교 대학원 전자공학과 (공학석사)
- 2003년 2월 : 단국대학교 대학원 전자공학과 (공학박사)
- 1995년 11월 ~ 1997년 2월 : LG 이노텍(주) 연구원
- 2004년 3월 ~ 현재 : 서일대학교 정보통신공학과 교수

〈관심분야〉
정보보호, 암호 프로토콜, 디지털워터마킹