Journal of The Institute of Electronics and Information Engineers Vol.52, NO.4, April 2015 http://dx.doi.org/10.5573/ieie.2015.52.4.099
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2015-52-4-13

안드로이드 스마트기기에서의 스테가노그래피 연구

(Steganography on Android Smart Devices)

정 기 현*, 이 준 호**, 유 기 영***

(Ki-Hyun Jung, Joon-Ho Lee, and Kee-Young Yoo©)

요 약

스마트폰 사용의 확대로 iOS와 Android 운영체제에 대한 관심이 높아지고 있다. 본 논문에서는 안드로이드 플랫폼을 기반으로 한 스테가노그래픽 기법으로 안드로이드에서 기본적으로 제공하는 그래픽 라이브러리인 스키아를 기반으로 한 영상 포맷을 분석하고, 분석된 포맷에 기반한 알고리즘을 제안한다. 제안하는 알고리즘은 Alpha, Red, Green, Blue 각각 8비트씩 총 32비트를 사용하는 트루 칼라를 기반으로 네 개의 각 8비트 영역을 활용하여 비밀자료를 숨긴다. 또한 이미지 왜곡이 덜 민감한 Alpha 영역을 활용하여 최대한 비밀자료를 숨길 수 있도록 알고리즘을 제안함으로써, 안드로이드를 기반으로 하는 스마트기기에 모두 사용이 가능할 것으로 보인다. 실험결과에서는 Alpha값의 변화에 따른 비밀자료 삽입용량과 이미지 왜곡 정도를 보임으로써 제안하는 알고리즘의 우수성을 증명하고 있다.

Abstract

As increasing the use of smart phones, the interest of iOS and Android operating system is growing up. In this paper, a novel steganographic method based on Android platform is proposed. Firstly, we analyze the skia based image format that is supporting 2D graphic libraries in Android operating system. Then, we propose a new data hiding method based on the Android bitmap image format. The proposed method hides the secret data on the four true color areas which include Alpha, Red, Green, Blue. In especial, we increase the embedding capacity of the secret data on the Alpha area with a less image distortion. The experimental results show that the proposed method has a higher embedding capacity and less distortion by changing the size of the secret bits on the Alpha area.

Keywords : Information Hiding(정보은닉), Steganography(스테가노그래피), Watermarking(워터마킹), Reversible Data Hiding(가역정보은닉), Android(안드로이드)

Ⅰ. 서 론

안드로이드 모바일 운영체제로 구글이 2007년 안드로이드사를 인수하여 스마트기기용 오픈 소스로 공개 운영하고 있는데, 전세계 모바일 운영체제 시장의 77%를 안드로이드가 차지하고 있다. 우리나라의 경우는 안드로이드 비중이 93.4%로 세계에서 안드로이드 의존도가 높은 편에 속한다. 안드로이드 1.0이 2008년에 오픈된 이후 현재 안드로이드 5.0 롤리팝(Lollipop) 버전이 제공되고 있다[1~2].

한편으로 인터넷의 발전으로 멀티미디어 데이터에 대한 송수신이 확대됨에 따라 저작권 보호와 데이터 무결성 문제 등이 대두되었다. 이러한 문제점을 극복하기 위하여 암호학과 함께 정보은닉(Information Hiding, Data Hiding) 기법이 발전하게 되었다. 암호학에서는 공격자가 숨겨진 비밀자료 자체를 해독할 수 없도록 하

* 정회원, 경일대학교 사이버보안학과
(Department of Cyber Security, Kyungil University)
** 정회원, 국방과학연구소 제6기술연구본부
(6th R&D Institute, Agency for Defense Development)
*** 정회원, 경북대학교 컴퓨터공학부
(School of Computer Science and Engineering, Kyungpook National University)
© Corresponding Author(E-mail: yook@knu.ac.kr)
Received ; August 27 2014 Revised ; November 12, 2014
Accepted ; March 5, 2015

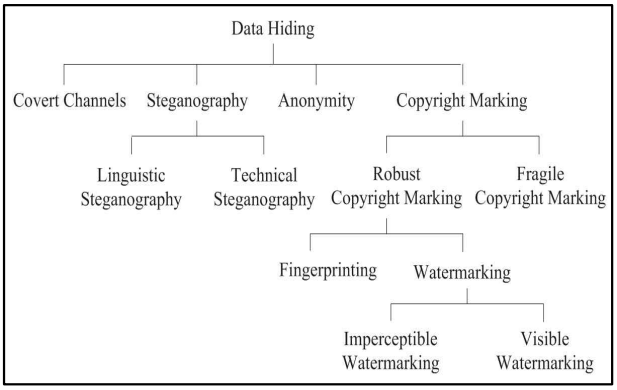


그림 1. 정보은닉 기법 분류
Fig. 1. Hierarchy of Information Hiding.

기 위한 특징을 가지고 있는 반면에 정보은닉 기법은 비밀자료가 숨겨졌다는 그 자체를 숨기는 것과 동시에 공격자가 숨겨진 비밀자료를 해독할 수 없도록 하는 것을 특징으로 한다[3]. 정보은닉 기법은 기준에 따라서 다양하게 분류될 수 있으나, 일반적으로 그림 1과 같이 나눌 수 있다[4~7].

또한 정보은닉 기법에서 비밀자료가 숨겨진 객체에서 비밀자료 추출과 더불어 원본 객체를 복원하느냐의 여부에 따라서 비가역 정보은닉과 가역 정보은닉(Reversible Data Hiding) 기법으로 나눌 수 있다. 비가역 정보은닉 기법에는 LSB(Least Significant Bit) 교체 기법과 PVD(Pixel-Value Differencing) 기법이 대표적이다[8~9]. LSB 교체 기법은 커버 객체(Cover Object)의 최하위 비트를 비밀자료로 교체하는 기법으로 결과적으로 사람의 시각으로 그 차이를 인지할 수 없다는 것을 기반으로 하고 있다. PVD 기법은 연속된 두 픽셀 차이값을 이용하여 비밀자료를 더 많이 숨길 수 있도록 제안되었다. 가역 정보은닉 기법에는 DE(Difference Expansion)와 히스토그램 시프팅(Histogram Shifting) 등이 대표적이다[10~11]. 히스토그램 시프팅 기법에서는 이미지 히스토그램의 최소값과 최대값을 이용하였고, DE에서는 두 픽셀에 대한 평균값과 차이값에 대한 확장을 통한 방법을 제안하였다. 이를 기반으로 다양한 개선 방법이 제안되었다.

최근 안드로이드를 기반으로 하는 스테가노그래피 기법에 최근 많이 연구되고 있다[12~16]. 이미지를 기반으로 하는 자료은닉 방법은 BMP 파일이나 RBG 영역을 활용한 LSB 기법을 기반으로 하고 있으며, QR 코드나 문자의 특성을 이용한 자료은닉 기법을 제안하고 있다.

본 논문에서는 안드로이드 모바일 운영체제를 탑재한 스마트기기에서 2D 그래픽 라이브러리인 스키아(Skia) 플랫폼을 기반으로 하는 자료은닉 기법을 제안하고자 한다. 스키아 플랫폼에서 기본적으로 제공되는 트루 칼라(True Color) 포맷을 분석하고, 안드로이드 이미지 포맷에 이미지 왜곡 정도를 식별하기 힘들도록 비밀자료를 숨길 수 있는 알고리즘을 제안한다.

본 논문은 다음과 같이 구성되어 있다. 제Ⅱ장에서는 제안하고자 하는 기법과 관련된 내용들을 살펴보고, Ⅲ장에서 제안방법에 대해서 상세하게 설명한다. 제안된 방법에 대한 실험결과를 Ⅳ장에서 다루고, 마지막으로 Ⅴ장에서 결론을 맺는다.

Ⅱ. 관련 연구

본 장에서는 제안된 방법과 관련되는 내용으로 안드로이드에서 기본적으로 채택하고 있는 그래픽 라이브러리인 스키아(Skia)와 안드로이드에서 이미지 처리를 위해 제공되는 비트맵 포맷에 대해서 설명한다.

1. 스키아(Skia) 플랫폼

안드로이드는 구글의 2차원 그래픽 라이브러리인 스키아 라이브러리를 지원하고 3차원 그래픽은 크로노스(Khronos) 그룹의 OpenGL ES를 기반으로 하고 있다.

스키아에서 독자적인 GPU 지원 구조를 만들어서 사용하고 있는 것과 같이 안드로이드 뷰(View) 시스템도 GPU를 직접 사용해서 하드웨어를 가속시킨다. 우리가 많이 사용하는 캔버스(Canvas)는 그림 2에 나타난 것과

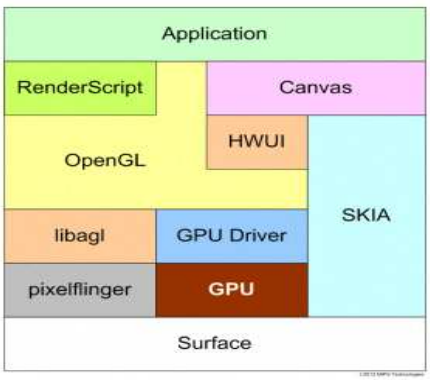


그림 2. 안드로이드 2D 그래픽 아키텍쳐
Fig. 2. Android 2D Graphics Architecture.

같이 HWUI와 스키아를 이용해서 서피스(Surface)로 요청을 하는 구조로 이루어져 있다. 안드로이드 3.0 부터는 화면이 큰 태블릿에 대한 지원으로 HWUI나 렌더스크립트(Renderscript) 등이 추가되었다[17~18].

2. 안드로이드 비트맵(Bitmap)

안드로이드에서 많이 사용하는 리소스 중 하나는 비트맵이다. 안드로이드에서는 Bitmap.Config.ALPHA_8, Bitmap.Config.ARGB_4444, Bitmap.Config.ARGB_8888, Bitmap.Config.RGB_565 등과 같이 4가지 형식의 비트맵 포맷을 제공한다. 높은 해상도를 유지하기 위해서는 Bitmap.Config.ARGB_8888 포맷을 사용하고 있는데, 이 구조는 그림 3과 같이 정의되어 있다.

안드로이드 운영체제의 프로그램 수준에서 선언하고 있는 Bitmap.Config.ARGB_8888 형식을 보면 그림 4와 같은데, 이는 스키아에서 사용하는 디폴트 설정값을 따르고 있다.

본 논문에서는 안드로이드에서 제공하는 이미지용 비트맵 포맷을 분석하여 Red, Green, Blue 영역뿐만 아니라, 육안으로 구별하기 힘든 Alpha 영역을 이용하여 비밀자료를 숨기고자 한다. 즉, Alpha 영역에는 더 많은 비트를 변경하더라도 이미지 왜곡에는 덜 민감하다는 원리를 이용하여 자료은닉 알고리즘을 제안한다.

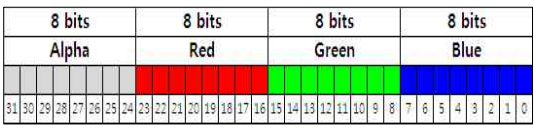


그림 3. Bitmap.Config.ARGB_8888 포맷
Fig. 3. Bitmap.Config.ARGB_8888 Format.

```
#if !defined(ANDROID)
  #define SK_A32_SHIFT 24
  #define SK_R32_SHIFT 16
  #define SK_G32_SHIFT  8
  #define SK_B32_SHIFT  0
#endif
```

그림 4. 안드로이드 Bitmap.Config.ARGB_8888 매크로
Fig. 4. Android Bitmap.Config.ARGB_8888 Macro.

Ⅲ. 제안 방법

본 논문에서는 Alpha 영역의 비트 변화에 따른 이미지 왜곡 정도를 비교하여 최대한 비밀자료를 숨길 수 있는 자료은닉 기법을 제안한다. 안드로이드 모바일 운영체제가 제공하는 트루 칼라에서 RGB 영역뿐만 아니라 Alpha 영역을 이용하는 것이다. 본 장에서는 먼저 Alpha 영역에서 비트값 변화에 따른 이미지 왜곡 정도를 분석하고, 커버 객체에 비밀자료를 숨기는 자료은닉 알고리즘과 수신된 스테고 객체에서 비밀자료를 추출하는 알고리즘을 살펴보고자 한다.

1. ALPHA값 변화에 따른 이미지 변화

한 픽셀을 기준으로 8비트를 가지는 Alpha 영역에서 최하위 비트값에서부터 최상위 비트값을 변경함에 따라서 결과 이미지가 어떻게 보여지는지를 살펴보면 그림 5와 같다. 그림 5에서 보는 것과 같이 RGB 영역을 제외한 Alpha 영역이 0xFF에서 0xC0까지 값을 변경하더라도 원래의 이미지를 확인할 수 있을 정도를 보여주고 있다. 즉 한 픽셀에 대한 8비트 영역에서 상위 2비트를 제외하고 하위 6비트까지 비밀자료를 숨길 수 있음을 보여주는 것이다. 결과적으로 다른 세 개 영역에 대한 자료은닉뿐만 아니라 Alpha 영역의 변화에 따라 더 많은 비밀자료를 숨길 수 있으므로 이에 대한 고려를 통하여 시스템을 설계하여야 한다.

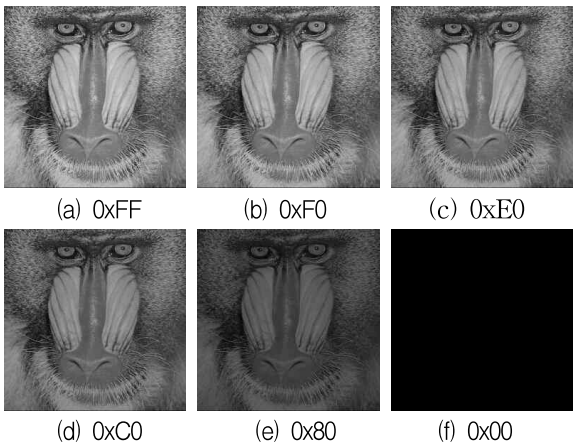


그림 5. Alpha값에 따른 이미지 왜곡정도
Fig. 5. Image Distortion for Alpha.

2. 자료 은닉 알고리즘

먼저 $W \times H$ 크기를 가진 커버(Cover) 이미지에서 하나의 픽셀값 p_{ij}에 대하여 Alpha, Red, Green, Blue 값은 수식 (1)과 같이 정의할 수 있다.

$$C = \{ p_{ij} \in \{0, 1, \ldots, 255\} \} \quad (1)$$

여기에서 $0 \le i < W, 0 \le j < H$ 조건을 만족한다. 다음으로 숨기고자 하는 n 비트를 가진 비밀자료는 수식 (2)와 같이 0과 1의 값으로 구성된다. 여기에서 s_k는 한 픽셀에 숨길 비트값을 나타낸다.

$$S = \{ s_k \mid 0 \le k < n,\ s_k \in \{0, 1\} \} \quad (2)$$

비밀자료를 숨기기 전에 주어진 비밀자료 S는 l 비트 크기를 가진 배열로 수식 (3)과 같이 재배치하게 된다.

$$S' = \{ s'_k \mid 0 \le k < n,\ s'_k \in \{0, 1, ..., 2^l - 1\} \} \quad (3)$$

다음으로 네 개의 영역에 대하여 비밀자료를 숨기는 과정은 수식 (4)를 따르게 된다. 하나의 픽셀값 p_{ij}에 대하여 비밀자료를 삽입한 결과값 p'_{ij}을 구하게 된다.

$$C' = \{ p'_{ij} \mid p'_{ij} = p_{ij} - (p_{ij} \bmod 2^l) + s'_k \} \quad (4)$$

위의 과정을 n 크기의 비밀자료를 모두 숨길 때까지 반복하게 된다.

3. 자료 추출 알고리즘

비밀자료가 숨겨진 스테고 이미지에 대하여 수신자 측에서는 별도의 추가 정보없이 비밀자료를 추출할 수 있다. 스테고 이미지에서 비밀자료를 추출하는 방법은 수식 (5)를 따른다.

$$S' = \{ s'_k \mid 0 \le k < n,\ p'_{ij} \bmod 2^l \} \quad (5)$$

이러한 과정은 하나의 픽셀값에 대하여 네 개의 영역에 각각 반복적으로 수행함으로써 얻어지게 된다.

Ⅳ. 실험 결과

본 논문의 실험에서는 그림 6에서 보는 바와 같이 256 x 256 크기의 이미지를 커버 이미지로 사용하고, 비밀자료는 랜덤함수를 사용하여 얻어진 값을 사용하여 어떠한 형태의 비밀자료로 사용가능함을 보였다.

$$PSNR_{a,r,g,b} = 10 \times \log\left(\frac{255^2}{MSE}\right)$$
$$PSNR_{rgb} = \frac{PSNR_r + PSNR_g + PSNR_b}{3} \quad (6)$$
$$PSNR_{argb} = \frac{PSNR_a + PSNR_r + PSNR_g + PSNR_b}{4}$$

(a) Airplane (b) Baboon (c) Boat
(d) Gatbawi (e) House (f) Lena
(g) Man (h) Peppers (i) Tiffany

그림 6. 실험에 사용된 커버 이미지
Fig. 6. Cover Images.

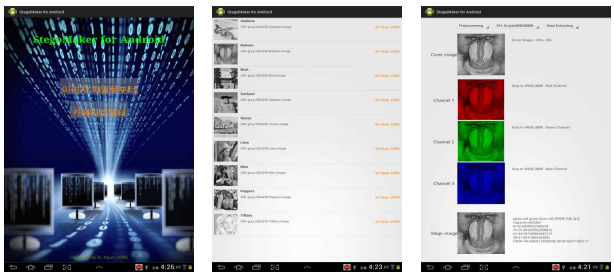


그림 7. 구현된 사용자 인터페이스
Fig. 7. Implemented User Interface.

실험에 대한 성능 평가를 위해 정보은닉 기법에서 주로 사용하는 비밀자료 삽입용량(Capacity)과 이미지 왜곡 정도를 나타내는 $PSNR$값을 사용하였다. 실험에 사용된 커버 이미지는 트루 칼라로 저장되기 때문에 $PSNR$값을 수식 (6)과 같이 분리하여 사용하였다. 여기서 MSE는 평균제곱오차를 나타낸다. 통상적으로 $PSNR$값이 $30dB$ 이상이면 사람의 육안으로 구별하기 힘들게 된다고 알려져 있다.

그림 7은 안드로이드로 구현된 결과를 보여주는 화면으로 갤럭시탭(GalaxyTab) 10.1인치 안드로이드 4.1.2 운영체제 환경에서 수행되었다.

제안한 알고리즘에서 각 영역별로 숨기는 비트 수를 $e = (\alpha, r, g, b) = (l_a, l_r, l_g, l_b)$ 라고 두면, 가변길이 v

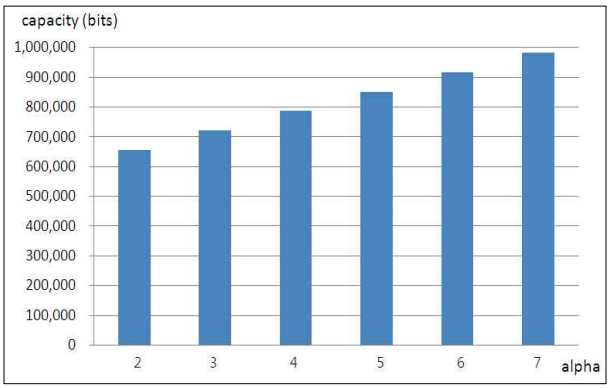
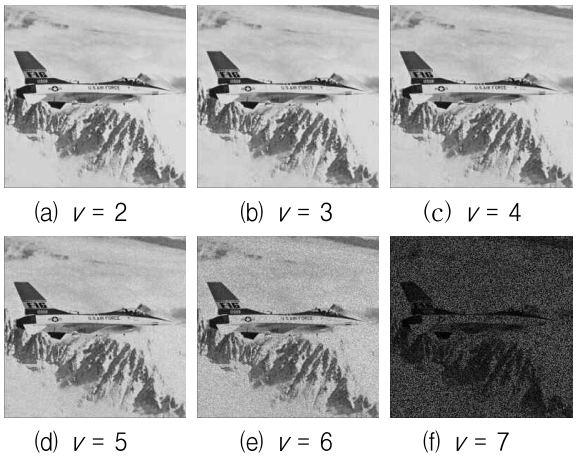


그림 8. Alpha값 변화에 따른 삽입용량
Fig. 8. Embedding Capacity for Alpha.



(a) v = 2 (b) v = 3 (c) v = 4

(d) v = 5 (e) v = 6 (f) v = 7

그림 9. v값 변화에 따른 Airplane 이미지
Fig. 9. Airplane Image for v value.

에 대해서 $(l_a, l_r, l_g, l_b) = (v, 3, 2, 3)$ 로 설정하여 Alpha 영역에 더 많은 비밀자료를 숨길 수 있음을 보이고자 한다. 즉 RGB 영역은 이미지 왜곡 정도를 구별하기 힘든 LSB2와 LSB3으로 고정하고, Alpha 영역의 변화에 따른 실험을 보이고자 하였다. 사람의 육안으로 구별하기 힘든 범위 내에서 많은 양의 비밀자료를 숨기기 위한 목적으로 그림 8은 비밀자료 삽입용량을 v 값의 변화에 따른 그래프로 $v = 2$인 경우에는 평균 655,360 비트 크기의 비밀자료를 숨길 수 있었고, $v = 7$인 경우에는 평균 983,040 비트 크기의 삽입용량을 가질 수 있었다.

삽입용량의 변화에 따른 Airplane 이미지에 대한 왜곡 정도를 그림 9에서 보여주고 있다. 그림 9 (e)와 (f)에서 보는 바와 같이 각각 917,504 비트와 983,040 비트를 숨길 수 있었으나, 이미지 왜곡을 쉽게 확인할 수 있으므로 최대 5비트를 숨기는 것이 바람직할 것으로 판단된다.

표 1. e=(2,3,2,3)에 대한 *PSNR* 결과
Table 1. *PSNR* Results for e=(2,3,2,3).

성능 / 커버	$PSNR_a$	$PSNR_r$	$PSNR_g$	$PSNR_b$	$PSNR_{rgb}$	$PSNR_{argb}$
Airplane	42.70	37.91	44.15	37.98	40.69	40.01
Baboon	42.66	37.89	44.10	37.89	40.64	39.96
Boat	42.69	37.90	44.13	37.93	40.66	39.99
Gatbawi	42.67	37.86	44.11	37.85	40.62	39.94
House	42.70	37.76	44.08	37.75	40.57	39.86
Lena	42.70	37.90	44.13	37.93	40.67	39.99
Man	42.70	37.95	44.05	37.94	40.66	39.98
Peppers	42.67	37.85	44.14	37.87	40.63	39.95
Tiffany	42.65	37.93	44.15	37.94	40.67	40.01

표 2. e=(5,3,2,3)에 대한 *PSNR* 결과
Table 2. *PSNR* Results for e=(5,3,2,3).

성능 / 커버	$PSNR_a$	$PSNR_r$	$PSNR_g$	$PSNR_b$	$PSNR_{rgb}$	$PSNR_{argb}$
Airplane	22.99	37.94	44.06	37.94	35.73	39.98
Baboon	23.00	37.89	44.01	37.90	35.70	39.93
Boat	23.01	37.91	44.03	37.89	35.71	39.94
Gatbawi	23.00	37.87	44.05	37.87	35.70	39.93
House	22.99	37.75	44.00	37.75	35.62	39.83
Lena	22.99	37.92	44.05	37.92	35.72	39.96
Man	23.02	37.91	44.05	37.92	35.73	39.96
Peppers	22.99	37.86	44.05	37.86	35.69	39.92
Tiffany	23.02	37.87	44.06	37.89	35.71	39.94

표 1과 표 2는 제안한 알고리즘에 대하여 e값의 변화, 특히 Alpha값의 변화에 따른 $PSNR$값을 보여주고 있다. 먼저 네 개의 각 영역에 대한 각각의 검토의견 감사드립니다. $PSNR$값과 함께 수식 (6)에서 정의한 트루칼라에 대한 $PSNR_{rgb}$와 $PSNR_{argb}$값을 보여주고 있다. 커버 이미지와 숨기고자 하는 비밀자료의 비트값 구성에 따라서 스테고 이미지가 생성되므로 $PSNR$값이 서로 약간의 차이를 보이고 있다.

표 1에서 보는 바와 같이 8개의 커버 이미지에 대해서 $e = (\alpha, r, g, b) = (2, 3, 2, 3)$ 을 적용한 경우에 평균 $PSNR_{rgb} = 39.97\,dB$ 와 $PSNR_{argb} = 40.64\,dB$ 값을 유지하였다. 또한 표 2에서와 같이 $e = (5, 3, 2, 3)$ 을 적용한 경우에 평균적으로 $PSNR_{rgb} = 39.93\,dB$ 와 $PSNR_{argb} = 35.70\,dB$ 값을 나타내었다. 즉, 결과적으로 $PSNR = 30\,dB$ 이상을 유지하고 있으므로 사람의 육안으로 이미지 왜곡을 구별하기는 쉽지 않다는 것을 보여주고 있다.

그림 10에서는 $e = (5, 3, 2, 3)$ 에 대한 스테고 이미지

그림 10. e=(5,3,2,3)에 대한 스테고 이미지
Fig. 10. Stego Images for e=(5,3,2,3).

를 보여주고 있다. 실험 결과를 살펴보면, 전체 8비트 중에 5-비트에 비밀자료를 숨긴 Alpha 영역에서는 $PSNR_a = 23.00\,dB$ 값을 유지하였다. 그리고 나머지 세 영역에서는 각각 $37.88\,dB$, $44.04\,dB$, 그리고 $35.70\,dB$ 를 유지하였다. 전체적으로는 왜곡 정도를 구별하기 힘들다는 것을 알 수 있다.

V. 결 론

본 논문에서는 안드로이드 플랫폼을 기반으로 하는 스마트기기에서 사용하고 있는 트루 칼라에 대한 비트맵을 분석하고, 이에 적합한 스테가노그래픽 알고리즘을 제안하였다. 안드로이드에서 사용하는 2D 그래픽 라이브러리인 스키아를 기반으로 사용되고 있는 트루 칼라인 Bitmap.Config.ARGB_8888에 비밀자료를 숨길 수 있는 알고리즘으로 Alpha, Red, Green, Blue 네 개의 영역에 각각 별도의 비밀자료를 숨길 수 있었다. 특히, Alpha 영역은 사람의 육안에 덜 민감하다는 점을 고려하여 더 많은 비밀자료를 숨길 수 있도록 설계하였다. 실험결과에서는 Alpha값의 변화에 따른 비밀자료 삽입 용량과 이미지 왜곡 정도를 분석함으로써 더 많은 비밀자료를 숨길 수 있는 방안을 제시하였다. 향후에는 안드로이드 모바일 운영체제에서 제공하는 이미지 처리 방법을 확대하여 다양하게 비밀자료를 숨길 수 있도록 적용하고, 안드로이드 기반 3D 이미지에 대한 정보은닉 방법을 연구하고자 한다.

REFERENCES

[1] NIPA, Android OS trends and implications, 2014.5.
[2] Android Developers, http://developer.android.com
[3] J. Zollner. H. Federrath. H. Klimant. A. Pfitzmann. R Piotraschke. A. Westfeld. G. Wicke. G. Wolf. Modeling the security of steganographic systems, 2nd Workshop on Information Hiding. pp. 345-355. 1988.
[4] B. Pfizmann. Information hiding terminology. Proc. First International Information Hiding Workshop. LNCS 1174. pp. 347-350. 1996.
[5] K.H. Jung, K.Y. Yoo, Data hiding using image interpolation, Computer Standards & Interfaces 31(2), pp. 465-470, 2009.
[6] K.H. Jung, K.Y. Yoo, Data hiding based on two-stage referencing for two-colour images, The Imaging Science Journal 61, pp. 475-483, 2013.
[7] W.J. Kim, P.H. Kim, J.H. Lee, K.H. Jung, K.Y. Yoo, Reversible data hiding method based on min/max in 2x2 sub-blocks, Journal of The Institute of Electronics and Information Engineers 51(4), pp. 69-75, 2014.
[8] R.Z Wang, C.F Lin, J.C Lin, Hiding data in images by optimal moderately significant-bit replacement, IEE Electron. Lett. 36 (25), pp.2069-2070, 2000.
[9] D. C. Wu and W. H. Tsai. A steganographic method for images by pixel-value differencing, Journal of Pattern Recognition Letters 24(9), pp. 1613-1626, 2003.
[10] J. Tian, Reversible data embedding using a difference expansion, IEEE Transactions on Circuits and Systems for Video Technology 13(8), pp. 890-896, 2003.
[11] L.C. Huang, L.Y. Tseng, M.S. Hwang, A reversible data hiding method by histogram shifting in high quality medical images, J Syst Software 86. pp. 716 - 727, 2013.
[12] D. Bucerzan, C. Ratiu, M.J. Manolescu, SmartSteg: a new android based steganography

application, Int J Comput Commun 8(5), pp. 681-688, 2013
[13] S. Sivakumar, B. Rajesh, Steganography on android based smart phones, IJCSMC 3(5), pp. 1051-1054, 2014
[14] H. Punjabi, D. Agrawal, S. Kumari, E. Bhagdev, StegoMMS for smartphones, IJETAE 4(3), pp. 829-832, 2014
[15] S. Mersal, S. Alhazmi, R. Alamoudi, N. Almuzaini, Arabic text steganography in smartphone, IJCIT 3(2), pp. 441-445, 2014
[16] W.C. Wu, Z.W. Lin, W.T. Wong, Application of QR-code steganography using data embedding technique, Information Technology Convergence 253, pp. 597-605, 2013
[17] Skia, https://sites.google.com/site/skiadocs/
[18] Skia Source, https://skia.googlesource.com/skia/

저 자 소 개

정 기 현(정회원)
1995년 경북대학교 컴퓨터공학과 (공학사).
1997년 경북대학교 컴퓨터공학과 (공학석사).
2007년 경북대학교 컴퓨터공학과 (공학박사).
1997년～2003년 국방과학연구소 선임연구원
2003년～2015년 영진전문대학 컴퓨터정보계열 교수
2015년～현재 경일대학교 사이버보안학과 교수
<주관심분야 : 정보보호, 디지털 워터마킹, 디지털 포렌식, 스테가노그래피, 게임/모바일프로그래밍>

유 기 영(정회원)
1976년 경북대학교 수학교육과 (이학사).
1978년 한국과학기술원 컴퓨터공학과 (공학석사).
1992년 미국 뉴욕 Rensselaer Polytechnic Institute 컴퓨터과학과 (공학박사).
1978년～현재 경북대학교 컴퓨터공학과 교수
1997년～1998년 한국정보과학회 영남지부장
1999년～현재 한국정보과학회 영남지부장
1999년～현재 한국정보과학회 이사
2006년～현재 제12대 한국정보보호학회 부회장
<주관심분야 : 암호학, 정보보호, 네트워크보안, 스테가노그래피>

이 준 호(정회원)
1996년 경북대학교 전자공학과 (공학사).
1998년 경북대학교 전자공학과 (공학석사).
1998년～현재 국방과학연구소 선임연구원
<주관심분야 : 정보보호, 스테가노그래피, 함정전투체계, 체계공학>