Journal of The Institute of Electronics and Information Engineers Vol.52, NO.11, November 2015 http://dx.doi.org/10.5573/ieie.2015.52.11.079
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2015-52-11-9

Emulab 테스트베드 환경에서의 분산 스테가노그래피 연구

(Research on Steganography in Emulab Testbed)

정 기 현*, 석 우 진**

(Ki-Hyun Jung and Woo-Jin Seok©)

요 약

스테가노그래피는 비밀 데이터가 숨겨져 있다는 그 자체를 숨기는 방법을 말한다. Emulab은 연구자가 언제든지 원하는 운영체제 시스템과 네트워크 토폴리지를 구성할 수 있도록 제공하는 프레임워크이다. 본 논문에서는 Emulab 환경에서 스테가노그래피 기법을 처음으로 적용하여 분산 처리가 가능함을 보이고자 한다. 칼라 비트맵 이미지를 사용하여 Emulab 환경에서 한 대의 서버와 여러 대의 클라이언트별로 나누어 처리하게 함으로써 알고리즘의 성능을 평가한다. 커버 이미지로 사용하는 칼라 이미지는 RGB 영역으로 각각 나누어지고, 각각의 영역에 대해서 비밀 데이터를 각 클라이언트에서 분산처리하게 하고, 성능을 비교하게 된다. 실험결과에서는 커버 이미지의 크기가 증가함에 따라 클라이언트/서버 구조를 가진 Emulab 환경에서 실행시간이 지속적으로 향상됨을 보여주고 있다.

Abstract

Steganography is to conceal the existence of secrete data itself. The Emulab is a framework to provide real systems and network topology that can set up at anytime by researchers. In this paper, we show that steganography techniques can be applied in the Emulab environment. Steganography methods are evaluated on a standalone and sharing environments using the color bitmap images. The cover image is divided into RGB channels and then embedded the secret data at each client. The experimental results demonstrate that execution time is better in client/server environment as cover image size is increasing.

Keywords : Information Hiding(정보은닉), Steganography(스테가노그래피), Emulab(에뮬랩), Data Hiding(자료은닉), Information Security(정보보호)

* 정회원, 경일대학교 사이버보안학과
(Department of Cyber Security, Kyungil University)
** 정회원, 한국과학기술정보연구원 첨단연구망센터
(Department of Advanced KREONET Center, Korea Institute of Science and Technology Information)
※ 본 논문은 한국과학기술정보연구원(KISTI)에서 수행중인 “첨단연구망 기반협업플랫폼 서비스 기술개발 및 적용(K-15-L01-C04-S03)” 사업과 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2015R1D1A1A01058019)
© Corresponding Author(E-mail: wjseok@kisti.re.kr)
Received ; September 22, 2015 Revised ; October 21, 2015
Accepted ; November 3, 2015

Ⅰ. 서 론

인터넷의 발전으로 멀티미디어 콘텐츠가 빠르게 확산됨에 따라 불법적인 사용 및 변경으로 피해가 확산되고 있는 가운데, 최근에 정보보호분야에서 암호학과 더불어 정보은닉기법이 활발하게 연구되고 있다. 스테가노그래피는 멀티미디어 콘텐츠와 같은 커버 객체에 비밀 메시지를 숨겨서 스테고 객체를 생성하게 되는데, 이때, 스테고 객체에 비밀 메시지가 숨겨져 있다는 그

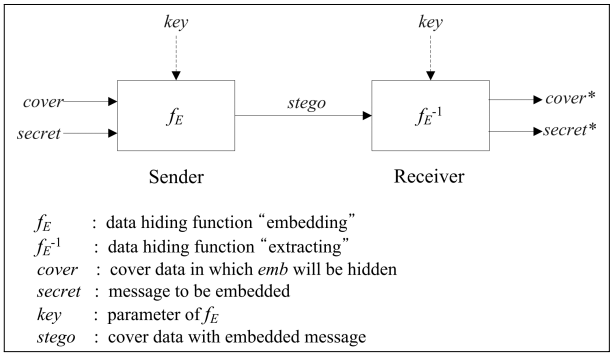

그림 1. 정보은닉 모델
Fig. 1. Model of Information Hiding.

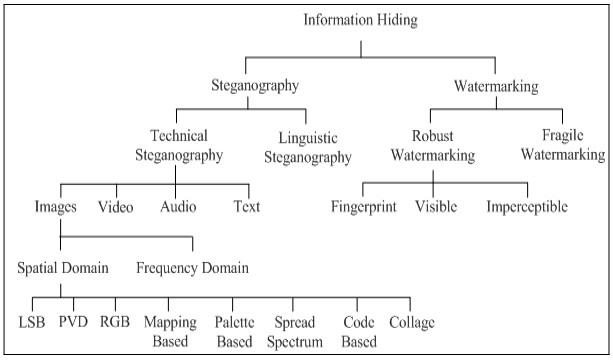

그림 2. 스테가노그래피 분류
Fig. 2. Hierarchy of Steganography.

자체를 숨기는 방법을 일컫는다[1-4]. 일반적으로 스테가노그래피는 그림 1에서 보는 바와 같이 스테고 객체를 받은 수신자측에서 비밀 메시지 추출과 함께 커버 객체의 복원 여부에 따라서 비가역정보은닉기법과 가역정보은닉기법으로 나눈다[5, 9~11].

한편, Emulab은 미국 유타대학에서 개발된 연구 프레임워크로 연구에 필요한 시스템 할당과 네트워크 토폴로지를 자유롭게 생성할 수 있는 서비스를 제공하는 테스트베드이다. 국내에서는 2012년에 KISTI에서 Emulab을 자체적으로 구축하여 운영하고 있다[6~8].

본 논문에서는 스테가노그래피 기술에 대한 응용으로 다양한 플랫폼 환경을 구축하고 실험한 사례는 찾아보기가 힘든 상황에서 Emulab 테스트베드 환경을 구축하고 현재 연구되고 있는 스테가노그래피 기술들이 효율적으로 처리될 수 있음을 보이고자 한다.

본 논문은 다음과 같이 구성되어 있다. 제Ⅱ장에서는 제안하고자 하는 기법과 관련된 내용들을 살펴보고, Ⅲ장에서 Emulab 환경에서 스테가노그래피 알고리즘이 적용될 수 있는 방법을 설명한다. 제안된 방법에 대한 실험결과를 Ⅳ장에서 다루고, 마지막으로 Ⅴ장에서 결론을 맺는다.

Ⅱ. 관련 연구

본 장에서는 제안된 방법과 관련되는 내용으로 Emulab 테스트베드 환경과 스테가노그래피에 대해서 설명한다.

1. 스테가노그래피

정보은닉기법은 크게 아래 그림 2와 같이 나눌 수 있는데, 기본적으로 커버 객체로 이미지를 사용하고 있는데, 그 이유는 확장성이 용이하기 때문이다[1~4].

다른 분류 기준으로 보면, 비가역정보은닉기법은 커버 객체는 복원할 수 없으나, 숨겨진 비밀 메시지를 추출할 수 있는 것으로 LSB(Least Significant Bit) 교환과 PVD(Pixel-Value Differencing) 방법이 대표적이고, 가역정보은닉기법은 양자화기반, 히스토그램 변경 기반, 확장 기반, 압축 기반, 듀얼 이미지 기반 등으로 분류할 수 있으나, 많이 활용되고 있는 알고리즘으로 DE (Difference Expansion), HS(Histogram Shifting), PEE (Prediction Error Expansion) 등이 있다.

2. Emulab 테스트베드

유타대학의 Flux 그룹에 의해 진행되고 있는 Emulab은 전 세계 37개가 구축되어 네트워크, 분산시스템, 보안, 교육 등의 목적으로 사용되고 있다[6-8].

국내 연구자들의 연구 및 실험환경 고도화를 위한 공용 서비스로 구축된 KISTI Emulab은 현재 42개의 테스트 노드와 노드 제어용 서버 및 파일 서버를 포함한

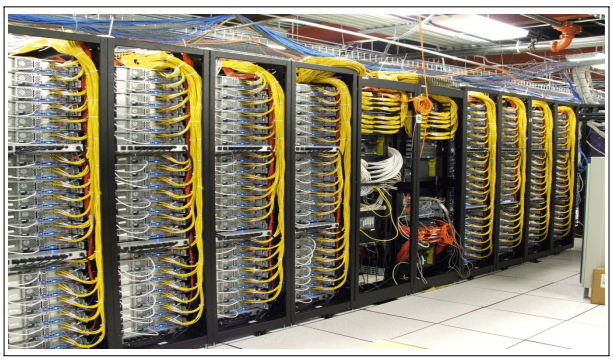

그림 3. Emulab 클러스터
Fig. 3. Emulab Clusters.

표 1. KREONET Emulab 서비스
Table 1. KREONET Emulab Service.

이미지 이름	세부 내용
CENTOS55-STD CENTOS63-65-STD	Emulab Standard CentOS 5.5 32-bit Emulab Standard CentOS 6.x 32-bit
UBUNTU12-64-STD UBUNTU12-64-JAVA	Ubuntu 12.04 LTS Ubuntu 12.04 LTS with JAVA
FBSD82-STD FBSD83-STD FBSD100-63-STD	FreeBSD 8.2 32-bit FreeBSD 8.3 32-bit FreeBSD 10.0 Emulab Standard Image
FEDORA15-STD	Standard 32-bit Fedora 15 Image
WIndos_7	Windows 7 Professional 32-bit

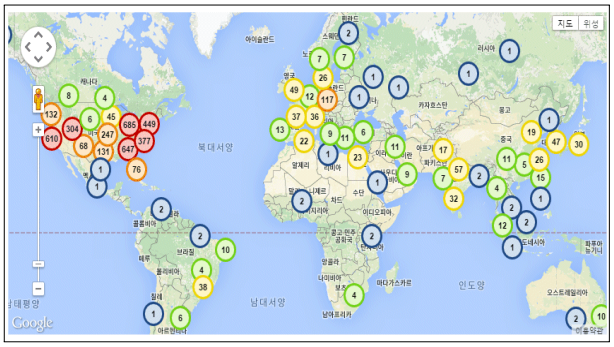

그림 4. Emulab 사용 현황
Fig. 4. Current Status of Emulab Usage.

서버 4대, 그리고 고성능 스위치 3대와 파워 컨트롤러 5개로 구축되어 있다. 표 1에서는 KISTI Emulab에서 사용할 수 있는 서비스를 나타내고 있는 것으로 다양한 운영체제뿐만 아니라, 노드수도 자유롭게 선택할 수 있도록 하고 있다.

이러한 Emulab 환경은 그림 4에서와 같이 다양한 실험환경을 손쉽게 구축할 수 있다는 장점으로 그 활용성이 확대되고 있다.

본 논문에서는 다양한 시스템과 네트워크 구성을 제공하는 Emulab 환경에서 스테가노그래피 알고리즘이 분산처리될 수 있는 가능성을 보이고자 한다.

Ⅲ. 제안 방법

본 논문에서는 Emulab 환경에서 RGB 칼라 이미지에 대하여 각 영역별로 나누어 처리하게 함으로써 스테가노그래피 알고리즘을 구현하고 실험하였다. RGB 이미지를 각 영역별로 나눌 경우, 3개의 그레이 이미지를 얻을 수가 있는데, 각각의 이미지에 대해서 그레이 이미지를 기반으로 하는 알고리즘을 적용할 수 있게 된다.

1. 분산처리 구성도

본 논문에서는 여러 노드 중에서 1대의 서버 노드와 3대의 클라이언트 노드를 구성하고, 아래 그림 5와 같이 자료은닉 과정에서 각 노드별로 역할 수행을 통하여 분산처리를 수행하고자 한다. RGB 각각에 대해서 하나의 클라이언트에서 독립적으로 수행시키고자 하였다.

다음으로는 자료 추출을 위한 과정으로 그림 6과 같이 수행하고자 한다.

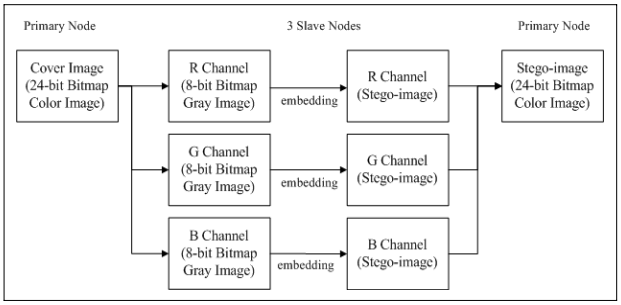

그림 5. 자료은닉 흐름도
Fig. 5. Flowchart of Data Embedding.

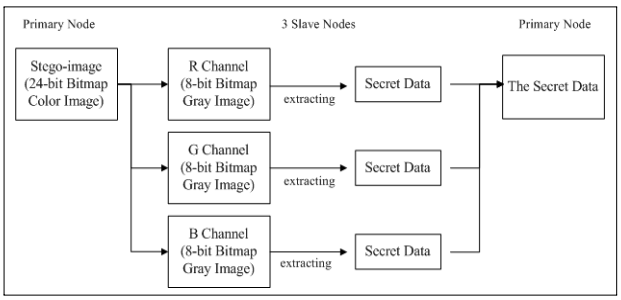

그림 6. 자료추출 흐름도
Fig. 6. Flowchart of Data Extracting.

2. 자료 은닉 방법

Emulab에서 구성한 노드를 각 영역별로 처리함으로써 스테가노그래피 알고리즘의 응용 범위를 확대할 수

Step 1. 칼라 이미지를 커버 이미지로 불러오기
Step 2. RGB 영역으로 분리하기
Step 3. 비밀자료와 영역별 이미지를 클라이언트로 보내기
Step 4. 클라이언트별로 비밀자료 숨기기
Step 5. 서버로 각 스테고 이미지 보내기
Step 6. 서버에서 칼라 스테고 이미지 생성하기

그림 7. 비밀자료 은닉 절차
Fig. 7. Secret Data Embedding Procedure.

있음을 보이고자 그림 7과 같은 절차로 구성하여 실험하였다. 사용된 비밀자료는 서버에서 랜덤함수를 이용하여 생성된 데이터를 사용하였다.

3. 자료 추출 방법

자료추출 절차는 수신자측에서 대상 이미지가 스테고 이미지인 것을 제외하면 자료은닉 절차와 유사하게 진행된다. 단, 실험결과에서도 알 수 있듯이, 1대의 서버에서 모든 과정을 수행하는 경우에는 그림 8에서 Step 3과 Step 5이 불필요하게 된다.

Step 1. 서버에서 칼라 스테고 이미지 불러오기
Step 2. RGB 영역으로 스테고 이미지 분리하기
Step 3. 각 영역별 스테고 이미지를 클라이언트로 보내기
Step 4. 클라이언트별로 비밀 데이터 추출하기
Step 5. 서버로 각 비밀 데이터 보내기
Step 6. 각 비밀 데이터를 합쳐서 비밀 메시지 완성하기

그림 8. 비밀자료 추출 절차
Fig. 8. Secret Data Extracting Procedure.

Ⅳ. 실험 및 분석

본 논문에서 제안한 내용은 자바 프로그램을 이용하여 실험하였다. 각 단계별로 시간 측정을 위하여 자바 1.8 버전 이상에서 java.time 패키지가 제공하는 Instant와 Duration 클래스를 사용하였다.

1. 실험 환경

Emulab 환경에서 우분투(Ubuntu) 12.04 LTS 운영체제를 아래 그림 9와 같이 구성하였으며, 실제 테스트는 1대의 서버 노드와 3대의 클라이언트 노드 환경에서 실험하였다. 각 노드는 Quad-core Intel Xeon CPU와 12GB

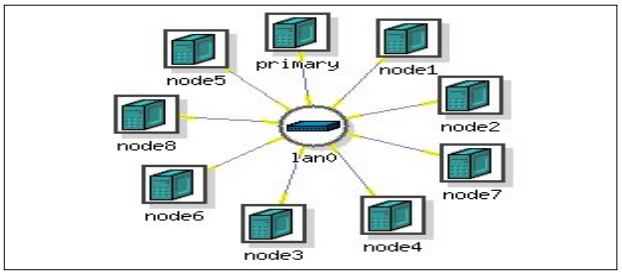

그림 9. 노드 구성 환경
Fig. 9. Node Configuration Environment.

```
#generated by Netbuild 1.03
set ns [new Simulator]
source tb_compat.tcl

set primary [$ns node]
set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]
set node5 [$ns node]
set node6 [$ns node]
set node7 [$ns node]
set node8 [$ns node]

tb-set-node-os $primary UBUNTU12-64-JAVA
tb-set-node-os $node1 UBUNTU12-64-JAVA
tb-set-node-os $node2 UBUNTU12-64-JAVA
tb-set-node-os $node3 UBUNTU12-64-JAVA
tb-set-node-os $node4 UBUNTU12-64-JAVA
tb-set-node-os $node5 UBUNTU12-64-JAVA
tb-set-node-os $node6 UBUNTU12-64-JAVA
tb-set-node-os $node7 UBUNTU12-64-JAVA
tb-set-node-os $node8 UBUNTU12-64-JAVA

set lan0 [$ns make-lan "$primary $node1 $node2 $node3 $node4 $node5
$node6 $node7 $node8 " 100Mb 0ms]

$ns rtproto Static
$ns run
#netbuild-generated ns file ends.
```

그림 10. 환경 설정 스크립트
Fig. 10. Configuration Setting Script.

Node ID	Name	Type	Default OSID	Node Status	Hours Idle[1]	Startup Status[2]	SSH URL	SSH mime	Console	Log
pc5	node5	dellR710	UBUNTU12-64-JAVA	up	6.14	none				
pc6	node2	dellR710	UBUNTU12-64-JAVA	up	0	none				
pc11	node3	dellR710	UBUNTU12-64-JAVA	up	4.4	none				
pc16	node8	dellR710	UBUNTU12-64-JAVA	up	0	none				
pc18	node1	dellR710	UBUNTU12-64-JAVA	up	4.18	none				
pc24	node6	dellR710	UBUNTU12-64-JAVA	up	14.69	none				
pc25	node4	dellR710	UBUNTU12-64-JAVA	up	0	none				
pc31	node7	dellR710	UBUNTU12-64-JAVA	up	14.69	none				
pc42	primary	dellR710	UBUNTU12-64-JAVA	up	0	none				

그림 11. 할당된 노드 세부화면
Fig. 11. Result of Allocated Nodes.

메모리를 가진 dellR710 PC로 구성되었다.

Emulab에서 노드 구성을 위한 스크립트는 그림 10과 같이 주어졌다.

Emulab에서 만들어진 결과 화면은 그림과 같다. 그림 11에서 보는 바와 같이 논리적인 구성을 물리적으로 맵핑된 결과를 알 수 있다.

2. 실험 결과 및 분석

스테가노그래피 실험을 위하여 상대적으로 하나의 노드에서 실험한 결과와 여러 노드를 사용한 결과를 비교하여 Emulab을 활용한 방법이 효율적임을 보이고자 하였다. 커버 이미지는 24-비트 비트맵 칼라 이미지를 사용하였다. 먼저, 1대의 서버로 구성한 경우에 대해서 각 단계별 수행시간을 비교한 결과를 표 2에 보여주고 있다.

다음으로 서버/클라이언트로 구성된 노드에서 클라이언트에서 수행하는 항목에 대한 평균 수행시간을 표 3에

표 2. 파일 크기별 서버 실행시간
Table 2. Execution Time on Server per File Sizes.

크기 / 단계	192KB	768KB	3,072KB	12,288KB	49,152KB	196,608KB
채널분리	382ms	729ms	1,868ms	5,792ms	21,512ms	81,070ms
자료은닉	204ms	514ms	1,734ms	5,953ms	20,904ms	80,595ms
스테고이미지 생성	23ms	81ms	304ms	1,152ms	4,591ms	18,209ms

표 3. 파일 크기별 클라이언트 실행시간
Table 3. Execution Time on Client per File Sizes.

크기 / 단계	192KB	768KB	3,072KB	12,288KB	49,152KB	196,608KB
자료은닉	223ms	347ms	772ms	2,374ms	7,580ms	27,819ms
채널별스테고 이미지 생성	62ms	163ms	496ms	1,458ms	5,594ms	16,310ms

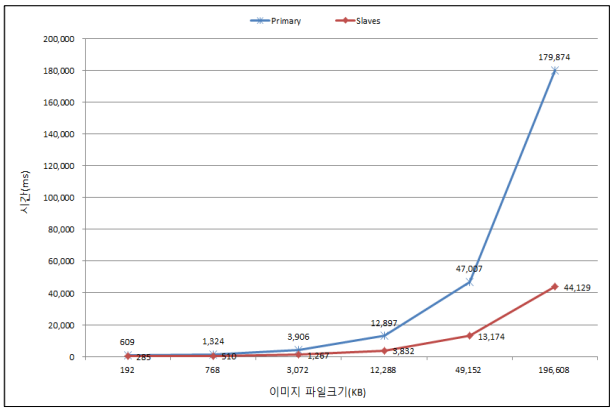

그림 12. 클라이언트/서버 수행시간 비교 그래프
Fig. 12. Graph of Client/Server Execution Time.

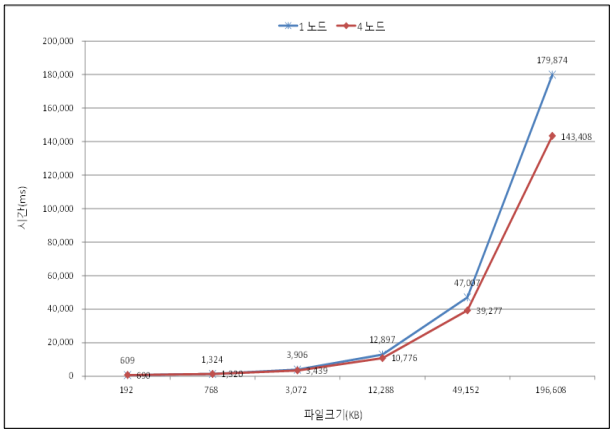

그림 13. 노드별 전체 수행시간
Fig. 13. Total Execution Time.

서 보여주고 있다. 파일 크기가 증가함에 따라 분산 처리가 효율적일 수 있음을 알 수 있다.

그림 12는 서버와 클라이언트에서 각각 수행되는 항목에 대한 전체 수행시간을 그래프로 나타낸 것이다.

또한, 그림 13에서는 본 논문에서 제안한 알고리즘을 분산 처리하였을 경우에 전체 수행시간을 나타낸 것이다. 즉, 1대의 서버에서 전체를 수행하는 시간과 노드를 클라이언트/서버로 구성하여 분산 처리할 경우, 스테고 이미지가 생성되는 전체 수행과정에 대한 시간을 그래프로 파일 크기가 커짐에 따라 여러 노드를 사용한 경우 처리 속도 측면에서 우수함을 알 수 있다.

본 실험에 사용된 스테가노그래피 기법은 기본적으로 이용되는 LSB, PVD, HS, DE 등을 사용하였으며, 각 방법이 수행되는 시간차이가 분산처리에 사용되는 시간과 비교하면 그 차이가 미미하여 모든 방법에 대한 결과는 제외시켰다. 실험 결과에 따르면, 이미지의 크기가 증가함에 따라 Emulab에서 제공하는 여러 클라이언트 환경으로 구성하여 처리하는 것이 더 효율적임을 알 수 있다.

위에서 살펴본 바와 같이, Emulab 환경에서 스테가노그래피 기법을 적용시킴으로써 분산처리가 가능함을 실험을 통하여 보였다. 또한, 스테가노그래피 측면에서 다양한 실험환경을 제공하는 Emulab을 활용할 수 있는 방안을 제시하였다.

V. 결 론

본 논문에서는 Emulab 테스트베드에서 다양한 운영체제와 다양한 네트워크 토폴로지를 구성하고, 이러한 환경을 스테가노그래피 기법에 적용하여 분산처리 가능함을 보였다. 특히, 칼라 이미지의 크기가 늘어남에 따라 Emulab 환경을 사용하여 분산 처리하는 것이 효율적임을 실험을 통하여 증명하였다.

향후 연구에서는 Emulab에서 제공하는 다양한 네트워크 토폴로지에 따른 실험을 통하여 그 활용 범위를 확대하고자 한다. 또한, 새로운 스테가노그래피 기법을 Emulab 환경에 적용하여 다양한 실험 결과를 도출하고자 한다.

REFERENCES

[1] A. Khan, A. Siddiqa, S. Munib, S.A. Malik, A recent survey of reversible watermarking techniques, Information Sciences 279, pp. 252-272, 2014.
[2] M.S. Subhedar, V.H. Mankar, Current status and key issues in image steganography: a survey,

Computer Science Review 13(14), pp. 95-113, 2014.
[3] A. Cheddad, J. Condell, K. Curran, P.M. Kevitt, Digital image steganography: survey and analysis of current methods, Signal Processing 90, pp. 727-752, 2010.
[4] A. Nissar, A.H. Mir, Classification of steganalysis techniques: a study, Digital Signal Processing 20, pp. 1758-1770, 2010.
[5] J. Zollner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf, "Modeling the Security of Steganographic Systems," 2nd Workshop on Information Hiding, pp. 345-355, 1998
[6] KISTI Emulab, http://www.emulab.kreonet.net
[7] Emulab, http://www.emulab.net/
[8] M.H. Le, W.J. Seok, "Research on the trend of utilizing Emulab as cyber security research framework", Journal of The Korea Institute of Information Security & Cryptology 23(6), pp. 1169-1180, 2013.
[9] K.H. Jung, J.H. Lee, K.Y. Yoo, "Steganography on android smart devices", Journal of The Institute of Electronics and Information Engineers 52(4), pp. 99-105, 2015.
[10] 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.745-751, 2014.
[11] K.H. Jung, K.Y. Yoo, "High-capacity index based data hiding method", Multimedia Tools and Applications 74(6), pp. 2179-2193, 2015.
[12] J. Tian, "Reversible data embedding using a difference expansion", IEEE Transactions on Circuits and Systems for Video Technology 13(8), pp. 890-896, 2003.
[13] K.H. Jung, "Image steganographic method using variable length for data embedding", The Korea Institute of Military Science and Technology 11(32), pp. 115-122, 2008.
[14] K.H. Jung, I.T, Kim, J.C. Kim, "Steganographic method based on three directional embedding", The Korea Institute of Military Science and Technology 13(2), pp. 268-274, 2010.
[15] K.H. Jung, K.Y. Yoo, "Data hiding method in binary images based on block masking for key authentication", Information Sciences 277(1), pp. 188-196, 2014.
[16] J. Mielikainen, "LSB matching revisited", IEEE Signal Processing Letters 13, pp. 285 - 287, 2006.
[17] D.C. Wu, W.H. Tsai, "A steganographic method for images by pixel-value differencing", Pattern Regcognition Letters 24, pp. 1613 - 1626, 2003.
[18] H.C. Wu, N.I. Wu, C.S. Tsai, M.S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods", IEE Processing Visualization, Image Signal Process 152, pp. 611 - 615, 2005.
[20] X. Li, J. Li, B. Li, B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion", Signal Processing 93, pp. 198-205, 2013.

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, 모바일/게임프로그래밍>

석 우 진 (정회원)
1996년 경북대학교 컴퓨터공학과 (공학사).
2003년 Univ. North Carolina, Computer Science (이학석사).
2008년 충남대학교 컴퓨터공학과 (공학박사).
2003년～현재 한국과학기술정보연구원(KISTI) 첨단연구망센터 책임연구원
<주관심분야 : 미래인터넷, Federation, SFA, SDN, SDX, SDI, 클라우드>