Approaches for steganography detection with Benford's Law

From various papers with contributions from:

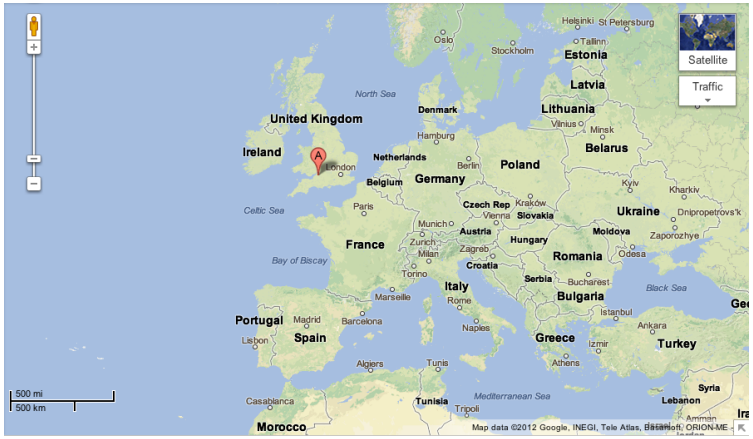
Panos Andriotis, Alex Zaharis, Dini Martini, Theo Tryfonas, George Oikonomou et al.

@theotryfonas, @PanosAndriotis

SPI 2013, Brno
Thu., 23rd May 2013

Where are we from?

University of BRISTOL



Smartphone Forensics and Content Verification

Cryptography Group, Department of Computer Science, University of Bristol

Motivation

In our on-going quest to secure our networks and systems we must first be able to detect and understand illegal actions as they happen, discover the attack infrastructures the miscreants are using and dissect the results of compromised systems. We aim to develop digital forensics tools that will identify, analyse and visualise illegal activities on related devices that use the Internet. Our main objective is to design a toolkit that will detect illegal activities in a post incident fashion, to identify their source, to profile the expertise and motive of the attackers and present the relevant information in a way that will be usable by investigating authorities.

Android Smartphone Forensic Analysis

Logical and physical acquisition of data stored in smartphones running Android OS.

Data Examination and Verification

SQLite Data Analysis, Pattern Lock brute force, JPEG Steganography Detection and Content Analysis.

Contribution and Outlook

We propose methods for acquiring forensic-grade evidence from Android smartphones using open source tools. We investigate cases where the suspect has made use of the smartphone's Wi-Fi or Bluetooth interfaces. The analysis of several case studies reveals traces left in the inner structure of mobile devices and also highlights security vulnerabilities. We perform physical acquisition of data and examine them safely in order to discover any activity associated with wireless communications.

We also present a novel approach to the problem of steganography detection in JPEG images by applying a statistical attack, which can be useful for content verification in numerous cases. We introduce a blind steganographic scheme that can flag a file as a suspicious stego carrier. Our method achieves very high accuracy and speed and is based on the distributions of the first digits of the quantised DCT coefficients present in JPEGs. Furthermore, we demonstrate that not only can we detect steganography but we can also reveal which steganographic algorithm was used to embed data in a JPEG file.

Many users prefer to utilise Android's 'pattern lock' mechanism instead of traditional text-based codes. Notable are methods that recover the lock patterns using the oily residues left on screens when people move their fingers to reproduce the unlock code. We performed a study on user perceptions of the security of patters they form when setting a graphical password for their phones. We are now able to use our survey's results to establish a scheme, which combines a psychology-based attack and a physical attack on graphical lock screen methods, aiming to reduce the search space of possible combinations forming a pattern, to make it partially or fully retrievable.

Funding and Collaboration

European Union's Prevention of and Fight against Crime Programme "Illegal Use of Internet" - ISEC 2010 Action Grants,
grant ref. HOME/2010/ISEC/AG/INT-002.

University of BRISTOL

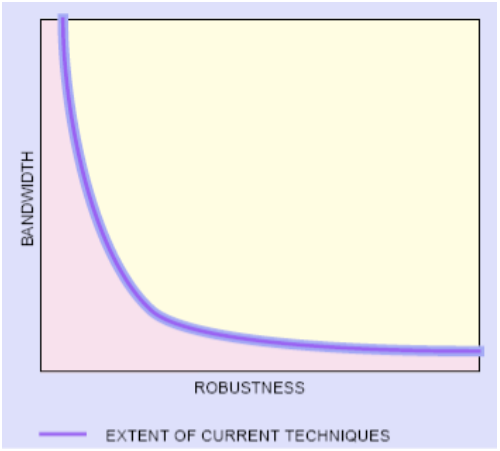

BROMPTON

Outline

- Briefly on steganography and steganalysis
- Briefly on Benford's Law and applications
- Applying Benford's Law to detect JPEG steganography
  - Raw byte values
  - DCT coefficients
- Further work

Data hiding

- Data insertion into existing data with the intention of:
  - fingerprinting
  - digital watermarking
  - covert communication



The robustness of the host signal reduces with the bandwidth (volume) of embedded data

Types of Data Hiding

- Media Management Layer
  - Use of areas that the OS is unaware of (Unallocated space, Host Protected Area, Partition Gap, MBR-area)
- File System Layer
  - Exploitation of file system structures vulnerabilities (Slack Space, NTFS Alternate Data Streams, Reserved inodes - EXT2/3)
- Application Layer
  - Steganography

Embedding secret messages in images

- “Fuse”:
  - Embedding the secret information within the file exploiting its file structure.
  - Could be used with multiple file types.
- “Least Significant Bit (LSB) Encoding”:
  - Hiding 1 bit of data in every pixel of 8-bit images.
  - Hiding 3 bits of data in every pixel of 24-bit images
    - Very sensitive in change of format and encoding of the images (e.g. save from .GIF to .JPEG).

Example of LSB encoding manipulation

- Hiding the letter G in the following bit stream:

  10010101 00001101 11001001 10010110

  00001111 11001011 10011111 00010000

- G → 01000111

  10010100 00001101 11001000 10010110

  00001110 11001011 10011111 00010001

Embedding secret messages in images (cont'd)

- Takes advantage of the limitations of the human vision system (HVS).
- Anything that can be coded into a bit stream can be embedded in an image.
- 8-bit:
  - Small.
  - Only 256 colours available.
- 24-bit:
  - Better for steganography
  - Large number of possible colours (>16M) exceeds HVS capabilities for differentiation.
- Compression:
  - "lossy", the secret message may lose integrity because the compression algorithm reduces the image fidelity (JPEG).
  - "lossless, retains image properties at the expense of image size - good for steganogrphy (GIF, BMP).

Steganalysis

- Steganalysis is the process of detection and extraction of hidden messages from a carrier.
- It uses statistical and mathematical techniques to reduce as much as possible the range of suspicious files.
  - But sometimes all files may be suspected.
  - Embedded content may be encrypted.

Types of steganalysis

- **Stego only attack – where available is only the stego-object (carrier).**
- Known cover attack – initial cover object and corresponding stego object available to the analyst
- Known message attack – the secret message is available along with the stego object.
- Chosen stego attack – the algorithm (stego tool) and the stego-object are available.
- Chosen message attack – for given secret message we can create the corresponding stego object.
- Known stego attack – the algorithm (stego tool), the cover object and the stego-object are available.

Steganography tools

Text

Text Steganographic Tools	Plain Text	Other	Source Code	License	Production
PGPn123		Yes		Shareware	Yes
Nicetext	Yes		Yes	Open Source	Yes
Snow	Yes		Yes	Open Source	Yes
Texto	Yes		Yes	Open Source	Yes
Sam's Big Play Maker	Yes		Yes	Open Source	Yes
Steganosaurus	Yes		Yes	Open Source	Yes
FFEncode	Yes			Open Source	Yes
Mimic	Yes			Open Source	Yes
wbStego	Yes	HTML, PDF	Yes	Open Source	Yes
Spam Mimic	Yes			Not Specified	Yes
Secret Space	Yes			Not Specified	Yes
WitnesSoft	Yes	Yes		No longer in production	
MergeStreams		Hides excel file in word		Freeware	Yes
Steganos	Yes	HTML		Commercial	Yes
Invisible Secrets		HTML		Commercial	Yes

Image

Image Steganographic Tools	BMP	JPEG	GIF	PNG	TGA	Other	Production	License
Crypto123	Yes	Yes					Yes	S
Hermetic Stego	Yes						Yes	S
IBM DLS	Yes	Yes	Yes	Yes			Yes	S
Invisible Secrets	Yes	Yes		Yes			Yes	S
Info Stego	Yes	Yes	Yes				Yes	S
Syscop		Yes					Yes	S
StegMark	Yes	Yes	Yes	Yes	Yes	TIF	Yes	S
Cloak	Yes							S
Contraband Hell	Yes						Yes	F
Contraband	Yes						Yes	F
Dound	Yes						Yes	F
Gif it Up			Yes				Yes	F
Camouflage				Yes	Yes		Yes	F
Hide and Seek	Yes		Yes				Yes	F
InThePicture	Yes						Yes	F
S-Tools	Yes						Yes	F
Jpegx		Yes					Yes	F
Steganos	Yes					DIB	Yes	F
BMP Secrets	Yes							
DCT-Steg		Yes						
Digital Picture Envelope	Yes							
EikonAmark		Yes						
Empty Pic			Yes					
Encrypt Pic	Yes							
EzStego			Yes					
BMP Embed	Yes							
BMPTable	Yes							
StegoTif					Yes	TIF		
Hide Unhide						TIF		
In Plain View	Yes							
Invisible Encryption			Yes					
JK-PGS						PPM		
Scytale						PCX		
appendX		Yes	Yes	Yes				
Total	20	10	9	5	3	6	17	

S – Shareware License
F – Freeware License

Sound

Audio Steganographic Tools	MP3	WAV	Others	Production	License
Info Stego	Yes			Yes	Shareware
ScramDisk		Yes		Yes	Shareware
MP3Stego	Yes			Yes	Open Source
StegoWav		Yes		Yes	Open Source
Hide4PGP	Yes		VOC	Yes	Open Source
Steghide		Yes	AU	Yes	Open Source
S-Tool		Yes		Yes	Open Source
Invisible Secrets		Yes		Yes	Commercial
Paranoid			Yes	Yes	Commercial
Steganos		Yes	VOC	Yes	Commercial

Image

Image Steganographic Tools	JPEG	BMP	Others	Embedding Approach	Production
Blindside		Yes		SDS	Yes
Camera Shy	Yes			SDS	Yes
dc-Steganograph			PCX	TDS	
F5	Yes	Yes	GIF	TDS	Yes
Gif Shuffle			GIF	Change the order of the color map	Yes
Hide4PGP		Yes		SDS	Yes
JP Hide and Seek	Yes			SDS	Yes
Jsteg Jpeg	Yes			SDS	Yes
Mandelsteg			GIF	SDS	Yes
OutGuess	Yes		PNG	TDS	Yes
PGM Stealth			PGM		Yes
Steghide		Yes		SDS	Yes
wbStego		Yes		SDS	Yes
WnStorm			PCX		Yes

TDS - Transform Domain Steganography
SDS - Spatial Domain Steganography (LSB Replacement and LSB Matching)

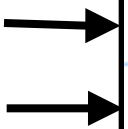
Disc and filesystem

File System Steganographic Tools	Location of Embedding	Source Code	License	Production
Disk Hide	Windows Registry	No		No
Drive Hider	Windows Registry	No		No
Easy File & Folder Protector	VXD driver, Windows Kernel	No	Shareware	Yes
Invisible Files 2000	Hard Disk	No	Shareware	Yes
Magic Folders	File System	No	Shareware	Yes
Dark Files	File system	No	Shareware	Yes
bProtected 2000	File system	No	Shareware	Yes
BuryBury	File system	No	Shareware	Yes
StegFS	File system	Yes	Open Source	Yes
Folder Guard Jr	File System	No	Freeware	Yes
Dmagic	File System	No	Freeware	Yes
BackYard	File System	No		No
Snowdisk	Disk space			No
Masker	Any file (Image, Text, Audio, Video)	No	Shareware	Yes
Anahtar	3.5-inch diskette	No		No
Hide Folders		No	Shareware	Yes
Hidden		No		No
Paranoid		No		No
Diskhide		No		No

Misc.

Miscellaneous Steganographic Tools	Cover Media	Source Code	License
GZSteg	.gz files	Yes	
InfoStego	Image, audio, video		Shareware
KPK File	Word, BMP		Shareware
S-Mail	.exe and .dll files		
Hiderman	Many different media		Shareware
StegMark	Image, audio, video		
Steghide	JPEG, BMP, WAV, AU	Yes	
S-Tools	BMP, GIF, WAV	Not sure	
Hydan	Program Binaries	Yes	Open Source
Covert.tcp	TCP/IP Packets	Yes	Open Source

Steganalysis tools



Hard Disk Steganographic Tools	Tools Analyzed	Detection Approach	Extraction Approach	Destruction Approach
2Mosaic	Removes stego content from any images			Break Apart
StirMark Benchmark	Removes stego content from any images			Resample
Phototile	Removes stego content from any images			Break Apart
Steganography Analyzer Real-Time Scanner	Analyzes Network Packets	Signature		
StegBreak	Jsteg-shell, JPhide, and Outguess 0.13b		Dictionary	
StegDetect	Jsteg, JPhide, Invisible Secrets, Outguess 01.3b, F5, appendX, Camouflage	Statistical		
StegSpy	Hiderman, JPHIde and Seek, Masker, JPegX, Invisible Secrets			
Stego-Suite	Detects Stego Image and Audio file		Dictionary	

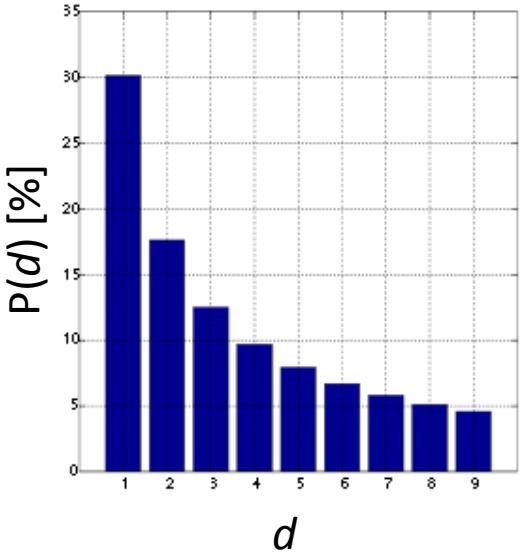
Benford's Law - historical facts

- 1881, Newcomb observed that the first pages of books with logarithmic tables, then heavily used for computation, were a lot more worn out than the last ones.
- Benford observed and abstracted formally this behaviour for random data sets around 1938.
  - Empirical law, a satisfactory explanation of which was provided by Hill (1996).
- This phenomenon can be observed and be of use in multiple domains and types of data sets.

The Law

The leading digit *d* (*d* E {1, …, *b* − 1}) of a number in base *b* (*b* ≥ 2) has a probability of occurrence that can be expressed as:

$$P(d) = \log_b(d+1) - \log_b(d) = \log_b\left(1 + \frac{1}{d}\right)$$



This represents the 'space' between the numbers *d* and *d* + 1, expressed in logarithmic scale.

For *b*=10 the following holds:

d	1	2	3	4	5	6	7	8	9
P(*d*)	30.1%	17.6%	12.5%	9.7%	7.9%	6.7%	5.8%	5.1%	4.6%

Lead digit distribution examples in natural data sets

col.	title	1	2	3	4	5	6	7	8	9	samples
A	Rivers, Area	31.0	16.4	10.7	11.3	7.2	8.6	5.5	4.2	5.1	335
B	Population	33.9	20.4	14.2	8.1	7.2	6.2	4.1	3.7	2.2	3259
C	Constants	41.3	14.4	4.8	8.6	10.6	5.8	1.0	2.9	10.6	104
D	Newspapers	30.0	18.0	12.0	10.0	8.0	6.0	6.0	5.0	5.0	100
E	Specific Heat	24.0	18.4	16.2	14.6	10.6	4.1	3.2	4.8	4.1	1389
F	Pressure	29.6	18.3	12.8	9.8	8.3	6.4	5.7	4.4	4.7	703
G	H.P. Lost	30.0	18.4	11.9	10.8	8.1	7.0	5.1	5.1	3.6	690
H	Mol. Wgt.	26.7	25.2	15.4	10.8	6.7	5.1	4.1	2.8	3.2	1800
I	Drainage	27.1	23.9	13.8	12.6	8.2	5.0	5.0	2.5	1.9	159
J	Atomic Wgt.	47.2	18.7	5.5	4.4	6.6	4.4	3.3	4.4	5.5	91
	Average	30.6	18.5	12.4	9.4	8.0	6.4	5.1	4.9	4.7	1011
	Probable Error	±0.8	±0.4	±0.4	±0.3	±0.2	±0.2	±0.2	±0.3		

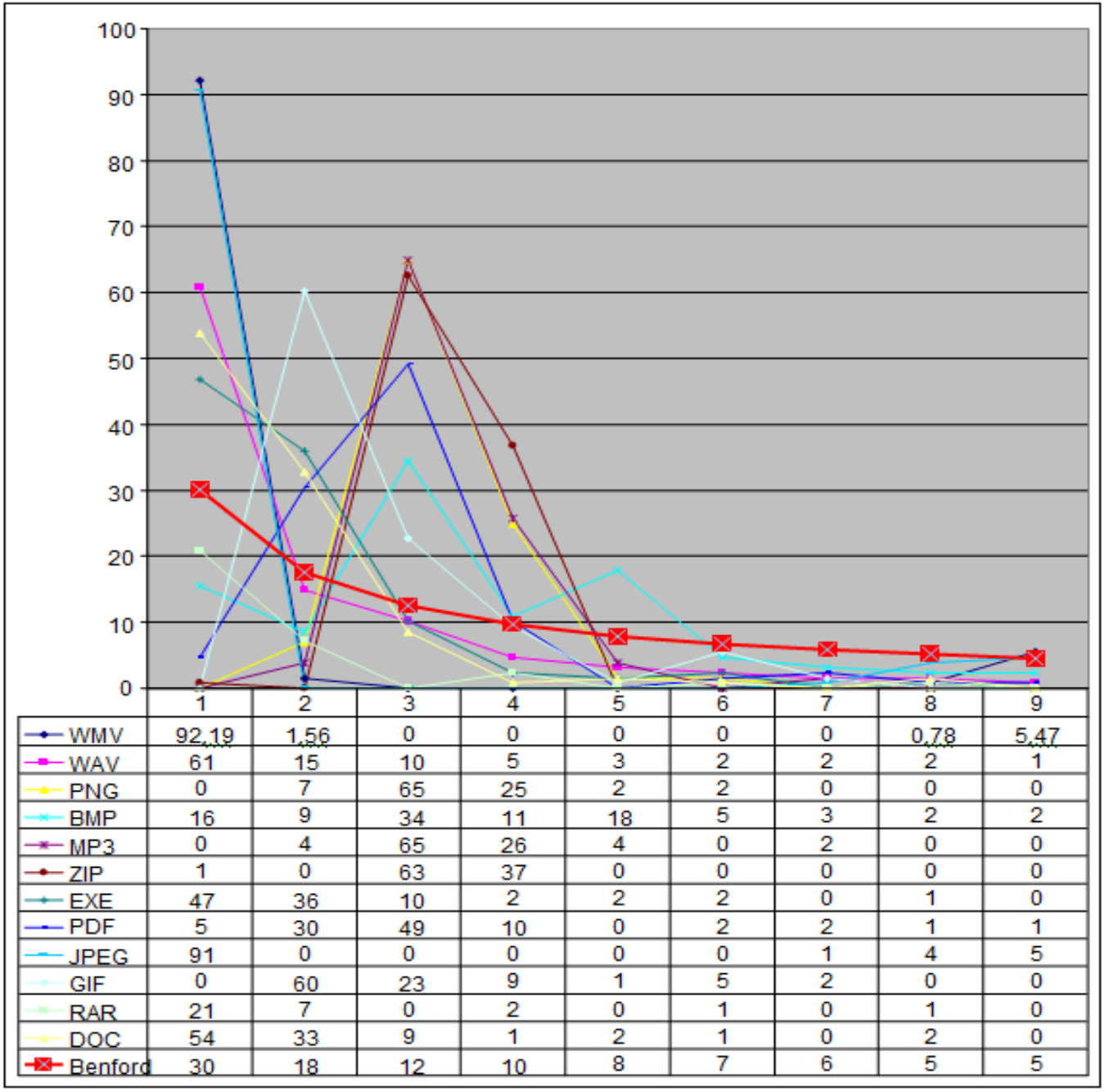
Various applications of Benford's Law

- Hal Varian (1972) proposed its use for detecting fraud in socio-economic data reporting.
- Used widely to detect fraud in transactional data (e.g. Nigrini, 2000 and others), as implemented within audit packages (ACL, IDEA etc.).
- Acceptable in courts of law in the US.
- Used to analyse the 2009 election results in Iran to prove rigging.

- **Limitation:** The law may be true for a set of items but not for a certain subset of it.

The first approach

- Use Benford's Law for detection of file anomalies on byte array sequences
  - Work from Karresand (2006) on byte value (eventually pairs) distribution in detection of image file format (and camera make) and
  - Work from Haggerty (2007) on file fingerprinting by byte value

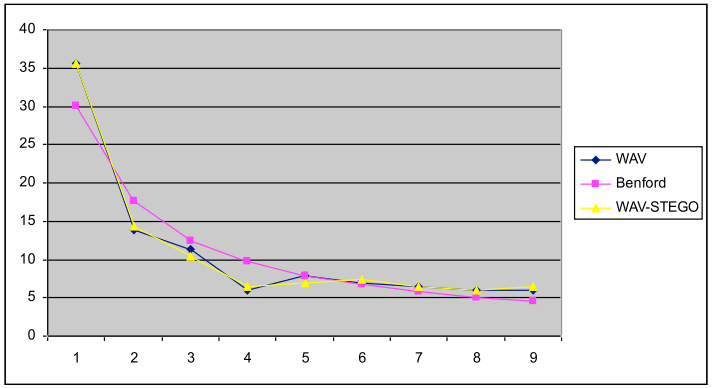


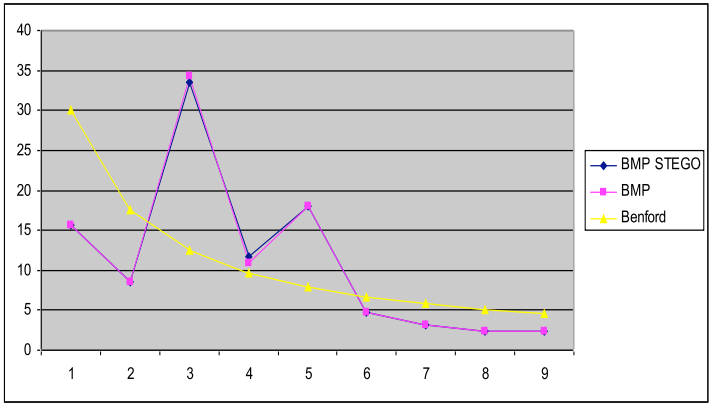
	1	2	3	4	5	6	7	8	9
WMV	92,19	1,56	0	0	0	0	0	0,78	5,47
WAV	61	15	10	5	3	2	2	2	1
PNG	0	7	65	25	2	2	0	0	0
BMP	16	9	34	11	18	5	3	2	2
MP3	0	4	65	26	4	0	2	0	0
ZIP	1	0	63	37	0	0	0	0	0
EXE	47	36	10	2	2	2	0	1	0
PDF	5	30	49	10	0	2	2	1	1
JPEG	91	0	0	0	0	0	1	4	5
GIF	0	60	23	9	1	5	2	0	0
RAR	21	7	0	2	0	1	0	1	0
DOC	54	33	9	1	2	1	0	2	0
Benford	30	18	12	10	8	7	6	5	5

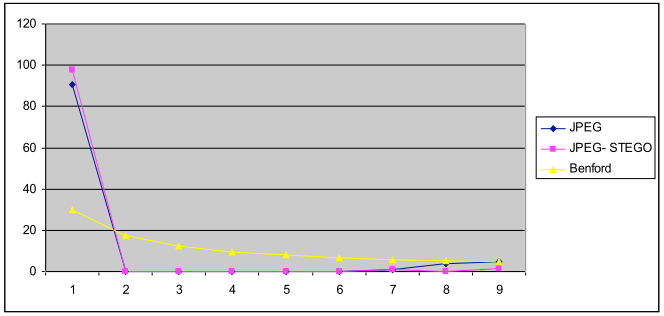
Steganography and alterations of file structure

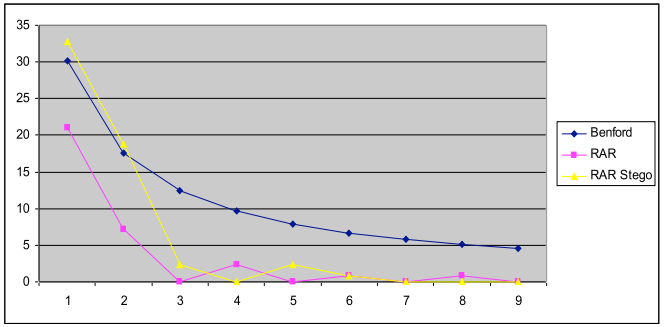
- We observed that the byte array representation's distribution was affected, in relation to the one of the original file types.
- Interestingly:
  - This was measurable for small size input secret files.
  - Increased with the size of the secret file.
  - It was detectable with no dependency of the type of stego algorithm used.

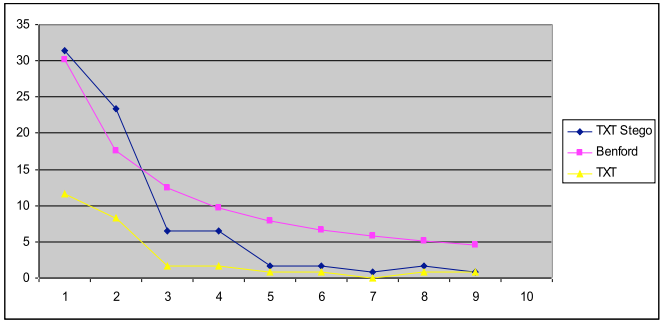
Variations for different file types









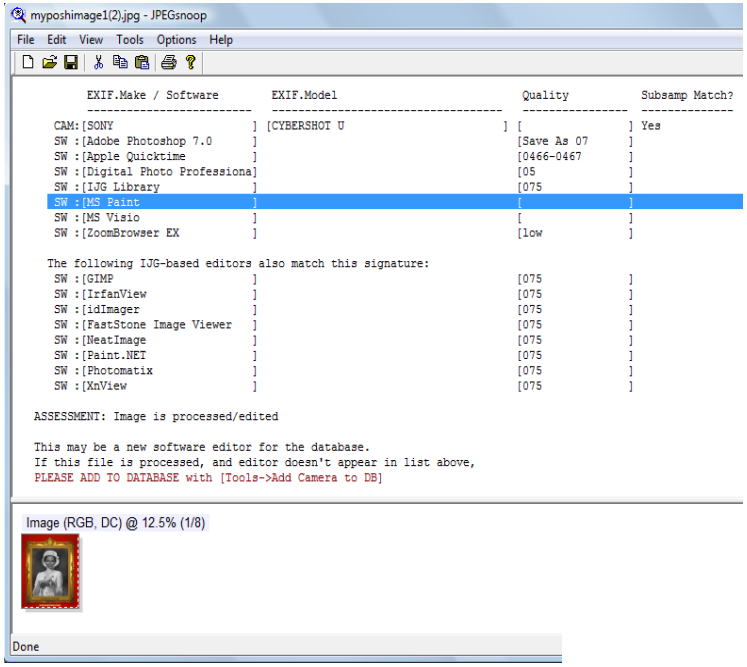


A key idea: Cover file generic reconstruction

- Generic reconstruction is a process whereby a file with similar properties to the original one is reconstructed from the stegocarrier.
- **Similar to what HVS does!**

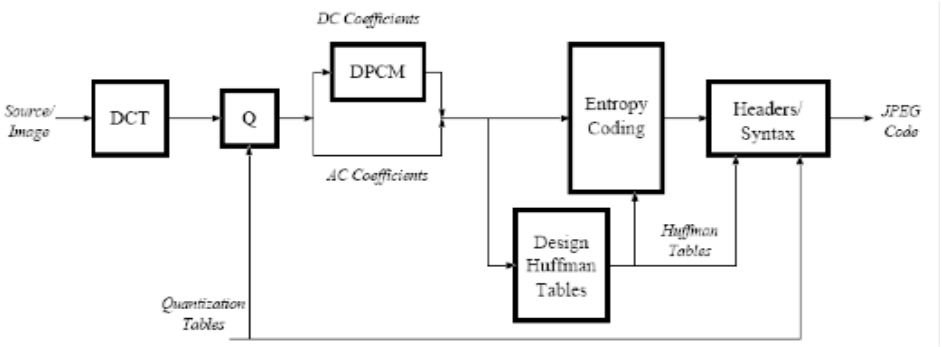
- Properties refer to:
  - Image quality.
  - File structure.
  - Content.
- Procedures that may change those could be :
  - Format alteration.
  - Copying reproduction (e.g. JPEG).
  - Use of stego tools.

File reconstruction




```
Precision=8 bits
Destination ID=0 (Luminance)
  DQT, Row #0:   8   6   5   8  12  20  26  31
  DQT, Row #1:   6   6   7  10  13  29  30  28
  DQT, Row #2:   7   7   8  12  20  29  35  28
  DQT, Row #3:   7   9  11  15  26  44  40  31
  DQT, Row #4:   9  11  19  28  34  55  52  39
  DQT, Row #5:  12  18  28  32  41  52  57  46
  DQT, Row #6:  25  32  39  44  52  61  60  51
  DQT, Row #7:  36  46  48  49  56  50  52  50
  Approx quality factor = 74.75 (scaling=50.51 variance=0.81)

 Precision=8 bits
 Destination ID=1 (Chrominance)
   DQT, Row #0:   9   9  12  24  50  50  50  50
   DQT, Row #1:   9  11  13  33  50  50  50  50
   DQT, Row #2:  12  13  28  50  50  50  50  50
   DQT, Row #3:  24  33  50  50  50  50  50  50
   DQT, Row #4:  50  50  50  50  50  50  50  50
   DQT, Row #5:  50  50  50  50  50  50  50  50
   DQT, Row #6:  50  50  50  50  50  50  50  50
   DQT, Row #7:  50  50  50  50  50  50  50  50
   Approx quality factor = 74.74 (scaling=50.52 variance=0.19)
```



Steganalysis method and proof of concept (for JPEG/MS Paint): Ben-4D

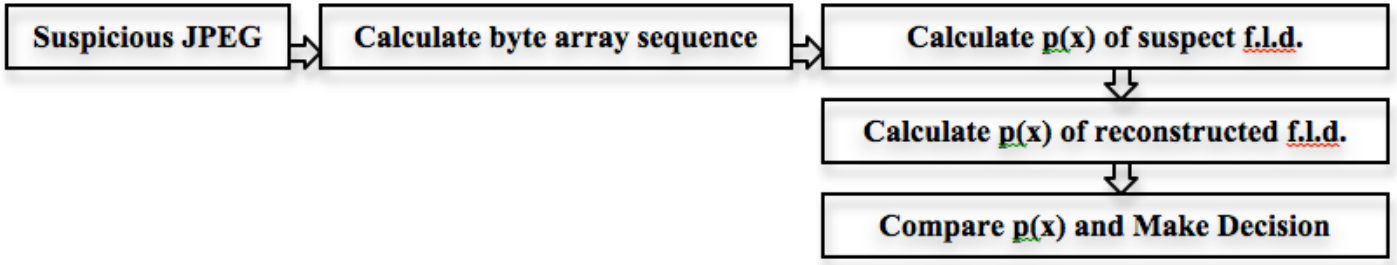


Figure 1: 'Ben-4D' Design Concept.

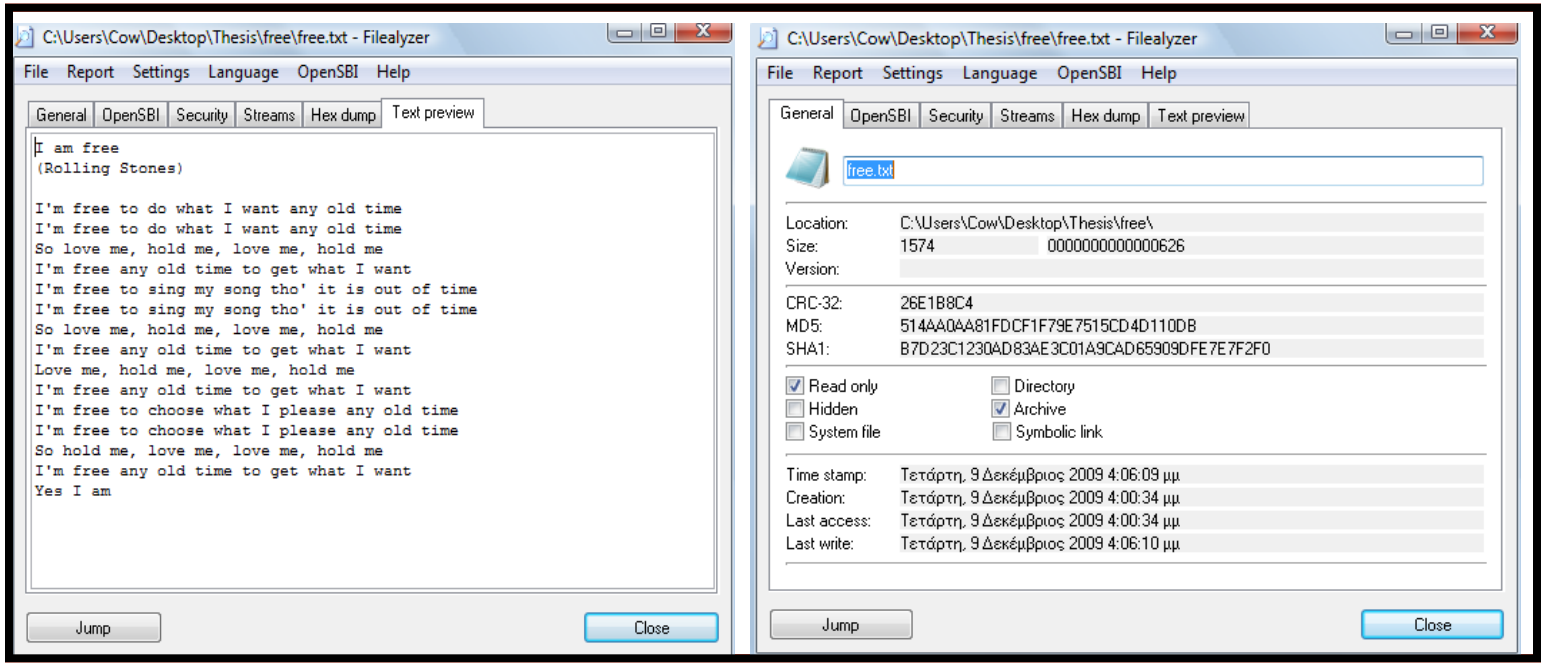
Similarity threshold

- Predefined constant value identified experimentally after applying Generalised Benford's Law on large numbers of reconstructed files.
- This value is encoding-specific, so MS Paint has a certain Similarity Threshold while Photoshop 9 has a different one.

free.txt
Έγγραφο κειμένου
1,53 KB

freeoriginal.txt
Έγγραφο κειμένου
634 byte

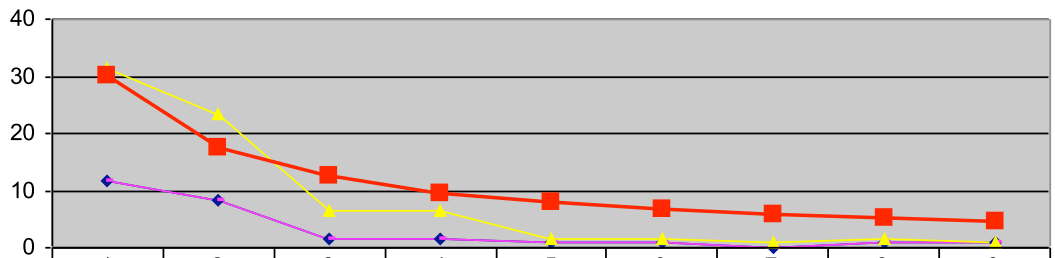
secret.txt
Έγγραφο κειμένου
85 byte



recreated.txt
Έγγραφο κειμένου
634 byte

STEGANOGRAPHY DETECTED

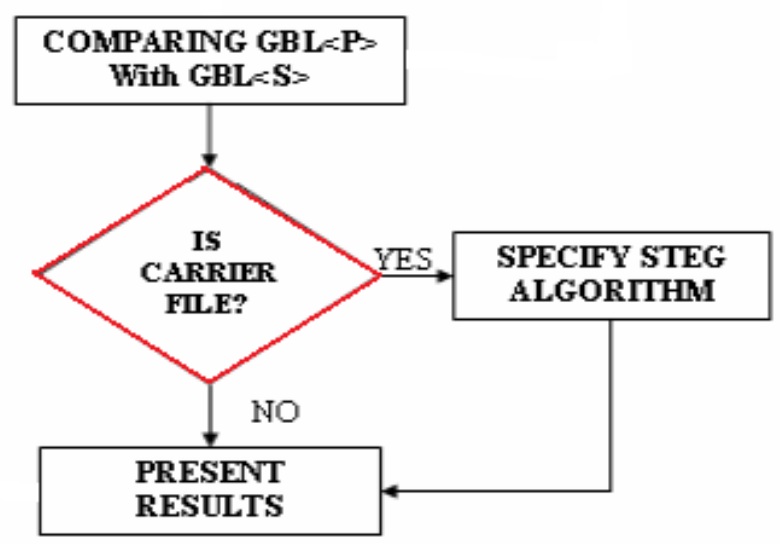
TXT Steganalysis



	1	2	3	4	5	6	7	8	9
freeoriginal	11,57	8,26	1,65	1,65	0,83	0,83	0	0,83	0,83
recreated	11,57	8,26	1,65	1,65	0,83	0,83	0	0,83	0,83
free	31,45	23,39	6,45	6,45	1,61	1,61	0,81	1,61	0,81
Benford	30	18	12	10	8	7	6	5	5

Improvement of detection rate by considering stego tools features

- Signatures/rules for the intended stego tool recognition:
  - Atypical or corrupted Huffman tables (JPHSWin).
  - Significant size difference of stegocarrier and reconstructed file (Camouflage, Invisible Secrets).
  - Specific headers manipulation (Invisible Secrets).
  - Issues with file termination (Camouflage).



Embedding these rules into the detection method leads to improvement of the *False – Positive detection rate from* **15% to 0.1%.**

Another approach

- Fu, Shi & Sub
  - examined the byte value distributions in the pixel domain (unsuccessful) as opposed to the Discrete Cosine Transform (DCT) values (that seems to obey Benford's law)
  - generalised the law to apply in detection of watermarked images
- Fu et al. worked on the distribution of first digits of DCT coefficients, but only on the luminance component of pictures
- We extended their work to chrominance and apply it comprehensively

StegBennie Algorithm

- After decompressing the image we read the metadata and find the compression quality factor.
- We are looking at the DCT blocks (8x8) that constitute the image and extract the first digit of each coefficient. For example, if the first row of an 8x8 block of coefficients is [211 22 12 6 1 0 0 0], the first digits are [x 2 1 6 1 x x x] (211 is the DC coefficient and it is excluded and also the zeros are not taken into consideration).
- We calculate the % percentage of appearance of each leading digit. Then we estimate the first digits expected distribution and finally compare the deviations between the expected and the calculated distributions.
- Any deviation between the expected and the estimated distributions will help to decide if the image is a stego or not.

StegBennie Algorithm (cont'd)

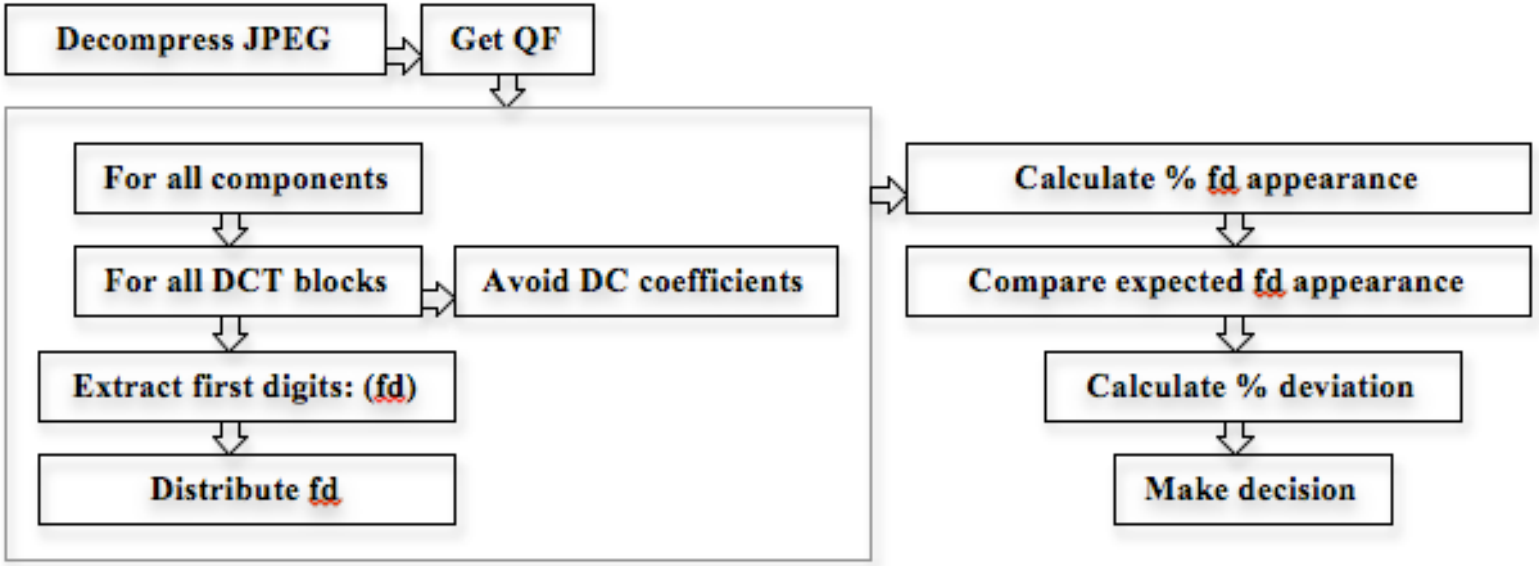


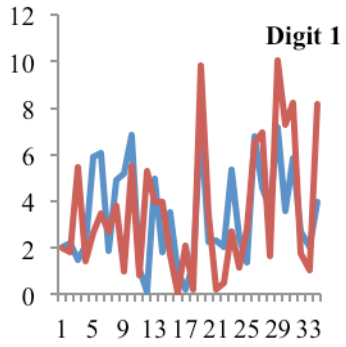
Figure 2: 'StegBennie' Design Concept.

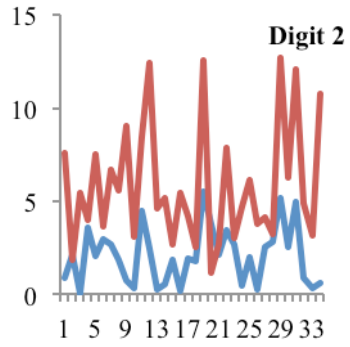
Quantised DCT coefficient-based analysis

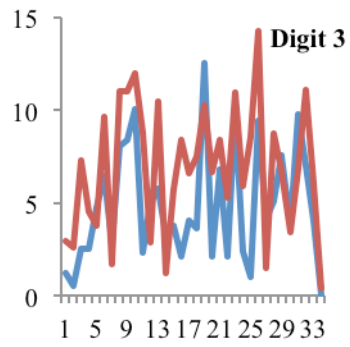
$$p(n) = N \cdot \log\left(1 + \frac{1}{s + n^q}\right), \quad n = 1, 2, \ldots, 9 \quad (2) \qquad \textit{(Generalized Benford's Law:" gBL")}$$

Quality Factor	*Model Parameters*			*Goodness-of-fit (SSE)*
	N	*q*	*s*	
100	1.608	1.605	0.0702	5.129e-06
90	1.25	1.585	-0.405	7.235e-07
80	1.344	1.685	-0.376	3.007e-06
75	1.396	1.731	-0.3549	3.986e-06
70	1.434	1.766	-0.339	4.455e-06
60	1.514	1.843	-0.3114	5.464e-06
50	1.584	1.909	-0.2875	5.119e-06

Table 1. Behavioural model of gBL for various quality factors.





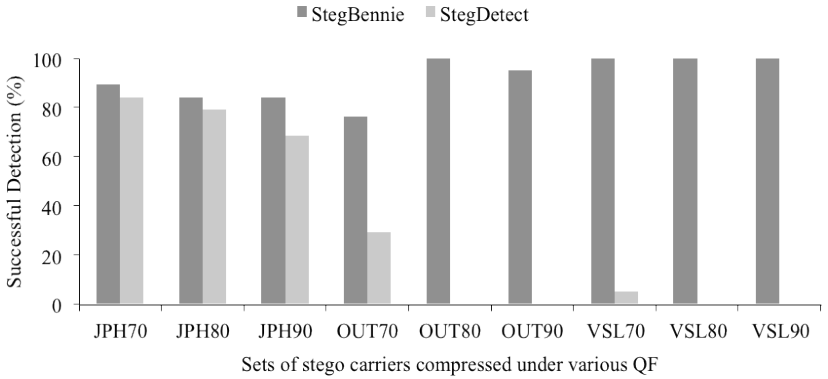


Comparisons with existing tools

- Ben-4D tested on 500 original images each with three stego variants (1,500 stego images), across 3 resolutions (320x240, 600x320 and 800x600)
- Three-stage testing of StegBennie versus StegDetect
  - existing image processing testing set
  - training data from the set above
  - own generated smartphone data set

File Type	*Original*	*JPHSWin*	*Camouflage*	*Invisible Secrets*
Number of files	1500	1500	1500	1500
Detection Rates (%)				
Ben-4D (full)	99	98	100	100
StegDetect	99.5	99.2	100	100
StegSpy	98	99	100	100

Table 5: Hit Rates Comparison among 'Ben-4D' and other tools.



Further work - Ben-4D

http://sourceforge.net/projects/ben4dstegdetect/

- Support for detection of more steganography tools.
- Other types of JPEG coding.
- Support for other popular image formats (BMP, GIF).
- Put on github.

Further work - StegBennie

http://www.fortoo.eu

- Consider the effect of the size of the embedded data and measure its impact on the overall validity of the method.
- Fu et al. (2007) observed that the distributions of first digits of the coefficients of the blocks of the JPEG images **before** the quantization step during the compression of the image adhere to the original Benford's Law.
- Apply supervised learning algorithms on the training set.
- Open source dissemination via ForToo website.

Sources

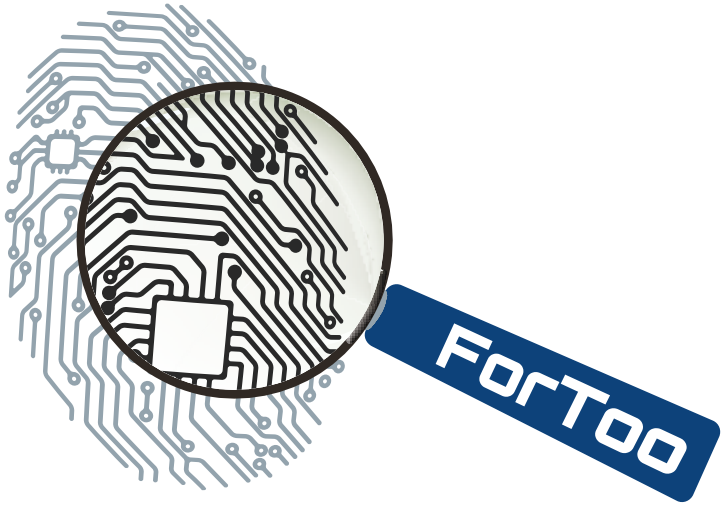
- Panagiotis Andriotis, George Oikonomou, Theo Tryfonas.
  JPEG Steganography Detection with Benford's Law.
  Digital Investigation, Vol. 9, pp. 246-257, 2013.
- A Zaharis, A Martini, T Tryfonas, C Ilioudis, G Pangalos.
  Lightweight Steganalysis based on Image Reconstruction & Lead Digit Distribution Analysis.
  International Journal of Digital Crime and Forensics, Vol. 3, pp. 29-41, 2011.
- A Zaharis, A Martini, T Tryfonas, C Ilioudis, G Pangalos.
  Reconstructive Steganalysis by Source Bytes Lead Digit Distribution Examination.
  Digital Forensics and Incident Analysis - WDFIA 2011, pp. 55-68, 2011.

Thank You

Any Questions?

Theo.Tryfonas@bristol.ac.uk

P.Andriotis@bristol.ac.uk



This work has been supported in part by the European Union's Prevention of and Fight against Crime Programme "Illegal Use of Internet" - ISEC 2010 Action Grants, grant ref. HOME/2010/ISEC/AG/INT-002