

Mobile steganography: Looking to the future

Team members

Dr. Jennifer Newman¹ - PI

Dr. Yong Guan¹ - Co-PI

Dr. Roy Maxion²

Dr. Min Wu³

¹ Iowa State University

² Carnegie Mellon University

³ University of Maryland-College Park

Post docs¹

Li Lin

Graduate students¹

Li Lin, Stephanie Reinders, Wenhao Chen, Abby Martin

Undergraduate students¹

Seth Pierre; Yangxiao Wang

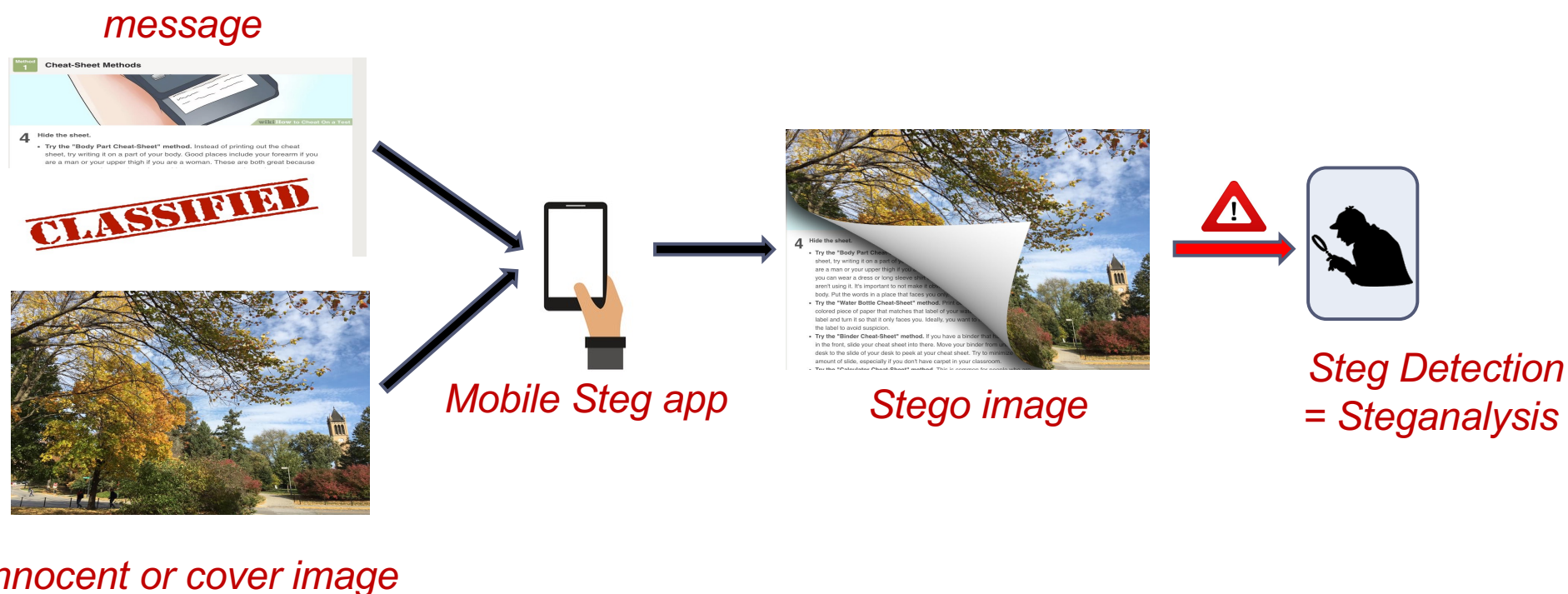
This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreements 70NANB15H176 and 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

Overview

- What is steganography and steganalysis
- Motivation for the use of mobile steganography and for a mobile app database
- Example of steganography embedding algorithm (LSB)
- How to use stego apps to create a mobile steg image database
- Data in StegoAppDB

What is steganography? *Hiding in plain sight*

- **Steganography:** to camouflage a message (payload) with an ordinary-looking object to avoid suspicion of covert communication
- **Goal:** make payload visually and statistically **undetectable** so there is no evidence of communication



Steganography as a threat

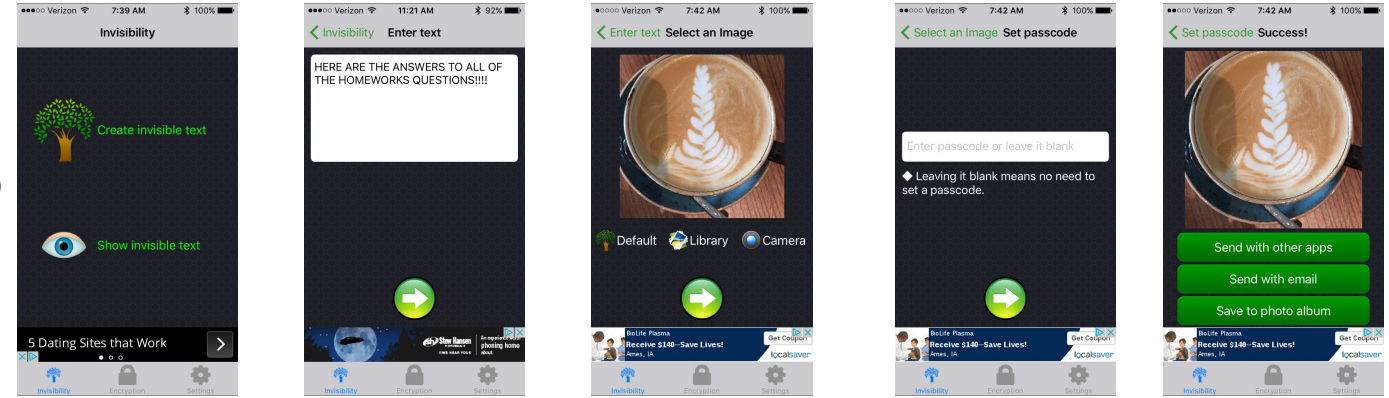
- National / Foreign intelligence
- Steal intellectual property
- Pornography
- Spying
- Communication for criminal activity
- Hide notes from illegal activities like financial transactions, etc.

Steganography as a threat

- National / Foreign intelligence
 - Steal intellectual property
 - Pornography
 - Spying
 - Communication for criminal activity
 - Hide notes from illegal activities like financial transactions, etc.
- Many ways to obtain images
 - Many ways to embed messages
 - Smartphones offer advantages

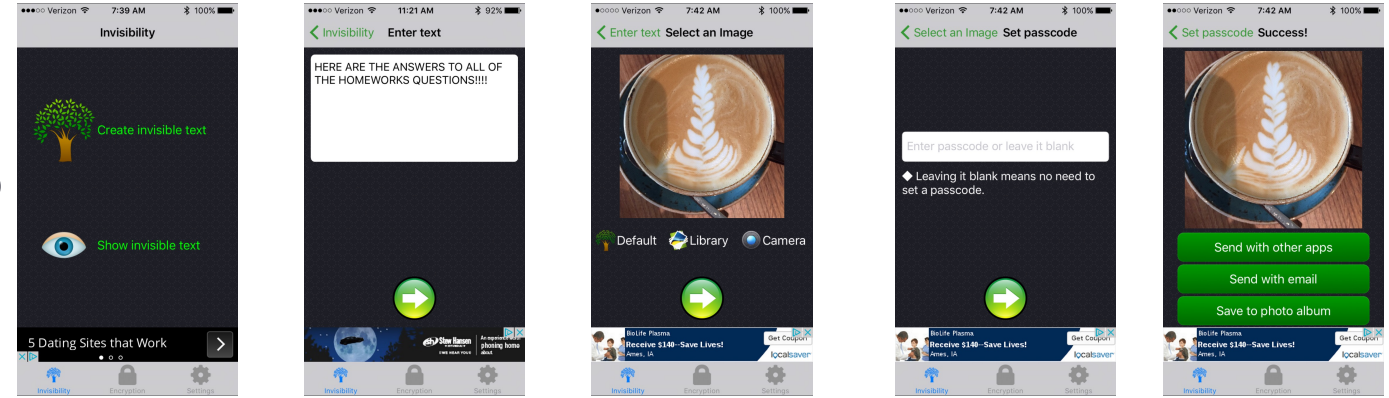
Scenario: How it works

- Alice wants to send a secret message to Bob
 - Download mobile steg app onto smartphone
 - Chooses a photo
 - Types in secret message
 - The app produces a stego image visually indistinguishable from the original
- Send to Bob
 - Uses same app to extract the hidden message



Scenario: How it works

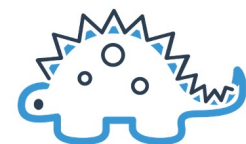
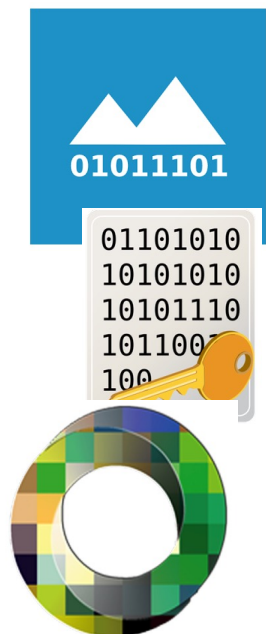
- Alice wants to send a secret message to Bob
 - Download mobile steg app onto smartphone
 - Chooses a photo
 - Types in secret message
 - The app produces a stego image visually indistinguishable from the original
- Send to Bob
 - Uses same app to extract the hidden message



- How to detect these kinds of stego images?
- Can we extract messages?

Selected Popular Steganography Apps on Google Play

App Name	Installs	Open Source	Output Format	Image Resizing	Payload Pre-processing			Embedding Technique
					Encryption	Signature Strings	Length Data	
PixelKnot	100,000+	Yes	JPG	Downsampling	Yes	No	Yes	F5
Steganography Master	10,000+	No	PNG	No	No	Yes	No	1's digit replacement
Steganography_M	10,000+	No	PNG	No	No	Yes	No	LSB replacement
DaVinci Secret Image	5,000+	No	PNG	User specified	No	Yes	Yes	Alpha channel encoding
Steganography_T	5,000+	No	PNG	No	No	No	Yes	LSB replacement
Stegais	1,000+	No	JPG	Downsampling	No	No	Yes	Unknown
PocketStego	1,000+	No	PNG	Downsampling	No	Yes	No	LSB
MobiStego	1,000+	Yes	PNG	Downsampling	Yes	Yes	No	RGB channels LS2B
NiaStego	1,000+	No	PNG	Upsampling	Yes	Yes	No	RGB channels LSB
Passlok	1,000+	Yes	JPG	No	Yes	Yes	No	Non-shrinkage F5



Why focus on smartphones?

- Many mobile stego apps available on smartphones
- Easy to use images for steganography on a smartphone
- Easier to use smartphone apps than computer
- Smartphones are present in almost all forensic investigations
- This is rationale for creating a steganography database entirely in mobile domain:
 - Image data originating from smartphone cameras
 - Stego images originating from smartphone apps
- We want to study steganography in this domain
 - Detect these kinds of stego images, extract messages, etc.
- Need special database: **StegoAppDB***

*StegoAppDB: A steganography apps forensics image database," IS&T Int'l. Symp. on Electronic Imaging, Media Watermarking, Security, and Forensics 2019, Burlingame, CA, pp. 536-1-536-12 (12), 2019.

Design criteria for StegoAppDB database

- *To be useful in a forensic context:*
 - **Authentication:** Provenance for each image
 - **Representation:** Includes practical representatives found in crime cases
 - **Evaluation:** Data can evaluate and benchmark algorithms
 - **Free public access, no copyright or privacy issues**
- *What we need in a steganography database*
 1. Variety of embedding algorithms / apps
 2. Variety of smartphones where apps execute (Android, iOS)
 3. Variety of different images
 4. Payload
 5. Stego images

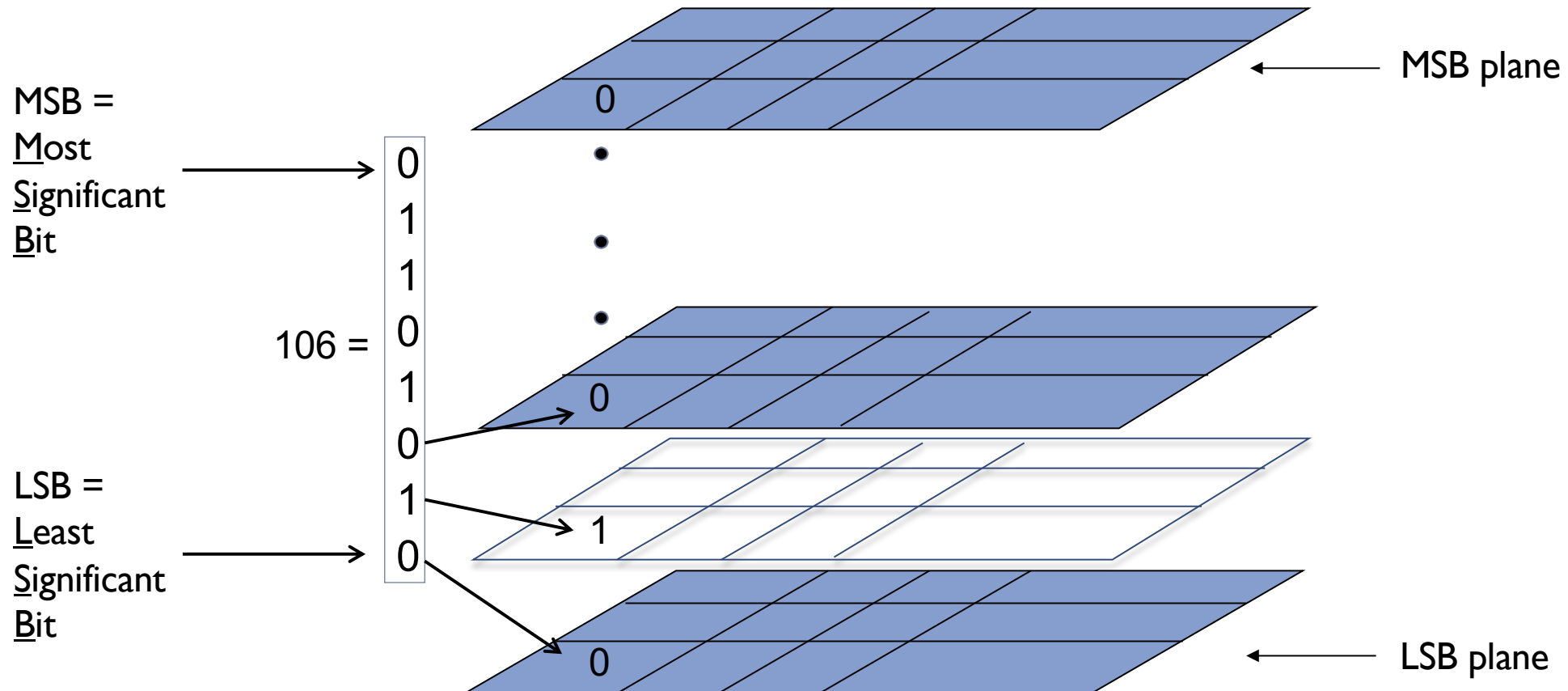
What is in StegoAppDB database

- Phones (property of CSAFE)
 - 10 different smartphone models, 2-4 devices per model, 28 distinct devices
 - Provides variety of models and more than one sample of each model
 - Provides variability in sensor response to scene content
- Protocol of image capture, scene content
 - A range of ISO and exposure times, 20 images of the same-scene: 10 JPEG, 10 DNG
 - Inside scenes only
- Used *Cameras* camera app to collect images according to protocol
 - Developed in our lab expressly for this purpose
- Stego images created using apps
- Message content verified in each stego image: extract and compare with embedded

LSB Replacement embedding

Review: The 8 bit-planes of an 8-bit image (grayscale)

Gray scale value of 106 represented by 8 bits



Example: Steganography embedding algorithm

LSB Replacement

- Take a grayscale image and observe its LSB plane

230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Cover image values

0	1	0	0	1	0	0
1	0	0	0	1	0	0
1	1	1	1	0	0	1
1	0	1	1	0	1	0
0	0	1	1	1	0	0
0	1	1	0	1	1	1
0	0	0	1	0	1	0

LSB values of cover image

LSB Replacement

Payload bit string: 010010000100010101010010010001010010000001000001010100...

230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Cover image values

0	1	0	0	1	0	0
1	0	0	0	1	0	0
1	1	1	1	0	0	1
1	0	1	1	0	1	0
0	0	1	1	1	0	0
0	1	1	0	1	1	1
0	0	0	1	0	1	0

LSB values of cover image

1	0	1	1	1	0	1
0	1	0	1	1	0	1
1	1	0	0	1	1	0
0	1	0	0	0	1	1
1	0	1	1	0	1	0
1	1	0	0	1	1	1
0	1	1	1	0	0	0

Payload bits - array

- Take a grayscale image and observe its LSB plane
- **Convert payload into sequence of binary bit values (0s and 1s) and form into array the size of the image**

LSB Replacement

230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Cover image values

231	228	233	235	235	232	149
236	237	236	235	233	234	153
255	255	254	250	231	237	160
98	91	66	36	94	247	131
223	152	255	129	128	247	132
155	199	254	150	189	241	147
216	133	163	163	170	238	122

Stego image values

0	1	0	0	1	0	0
1	0	0	0	1	0	0
1	1	1	1	0	0	1
1	0	1	1	0	1	0
0	0	1	1	1	0	0
0	1	1	0	1	1	1
0	0	0	1	0	1	0

LSB values of cover image

1	0	1	1	1	0	1
0	1	0	1	1	0	1
1	1	0	0	1	1	0
0	1	0	0	0	1	1
1	0	1	1	0	1	0
1	1	0	0	1	1	1
0	1	1	1	0	0	0

Payload bits - array

- Take a grayscale image and observe its LSB plane
- Convert payload into sequence of binary bit values (0s and 1s), uniformly distributed and form into array the size of the image
- Replace the (cover) image's LSB values with the payload bits
- The stego image's gray value is the (new) base-10 number if the LSB bit is changed

LSB Replacement

230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Cover image values

231	228	233	235	235	232	149
236	237	236	235	233	234	153
255	255	254	250	231	237	160
98	91	66	36	94	247	131
223	152	255	129	128	247	132
155	199	254	150	189	241	147
216	133	163	163	170	238	122

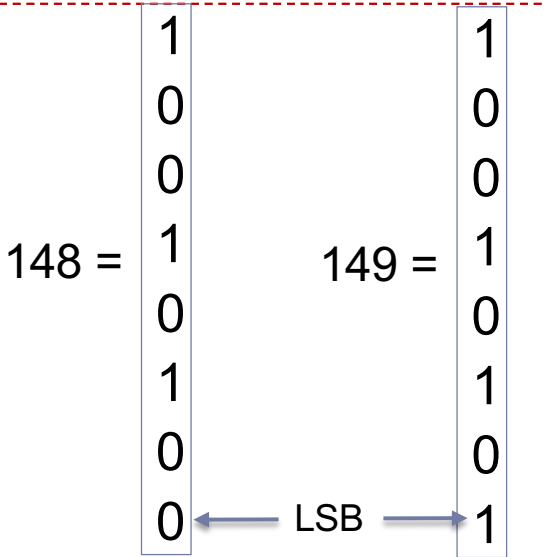
Stego image values

0	1	0	0	1	0	0
1	0	0	0	1	0	0
1	1	1	1	0	0	1
1	0	1	1	0	1	0
0	0	1	1	1	0	0
0	1	1	0	1	1	1
0	0	0	1	0	1	0

LSB values of cover image

1	0	1	1	1	0	1
0	1	0	1	1	0	1
1	1	0	0	1	1	0
0	1	0	0	0	1	1
1	0	1	1	0	1	0
1	1	0	0	1	1	1
0	1	1	1	0	0	0

Payload bits - array



cover

stego

- LSB replacement simply overwrites the existing bit with the payload bit

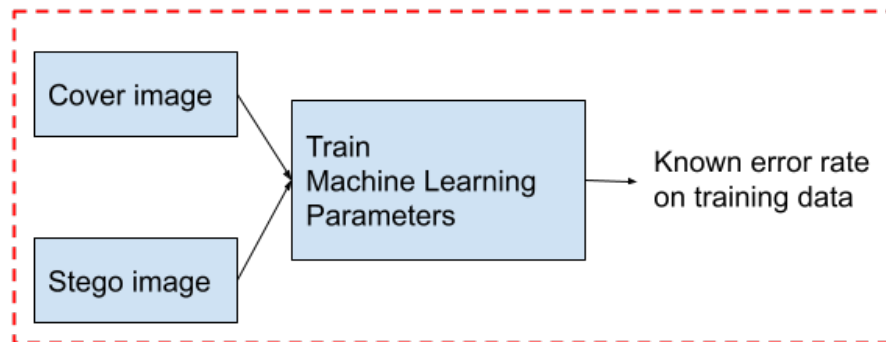
Analyzing an image for message content

- Analyze an image
 - Is it a cover image or stego image? *Classify an image as cover or stego*
 - How large (bits) is the message (if stego)? *Quantitative steganalysis*
 - Where are the message bits? *Which pixel locations contain message bits?*
 - Can we extract the message? *Do we know the order of bit embedding?*

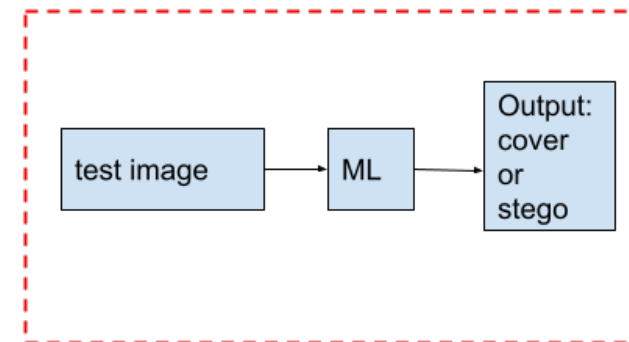
Steg Detection/Steganalysis: Machine Learning

- Steganalysis Machine Learning (ML) algorithms depend on training pairs of data:
 - One cover image (no message), and corresponding stego image with hidden message
- Take pairs & train a ML algorithm to “learn” the difference between the two types
- Then test: run other data that has not been used to train the ML algorithm through the ML algorithm, determine the error rate

Training Phase:
Training data in pairs
(known, generated by user)

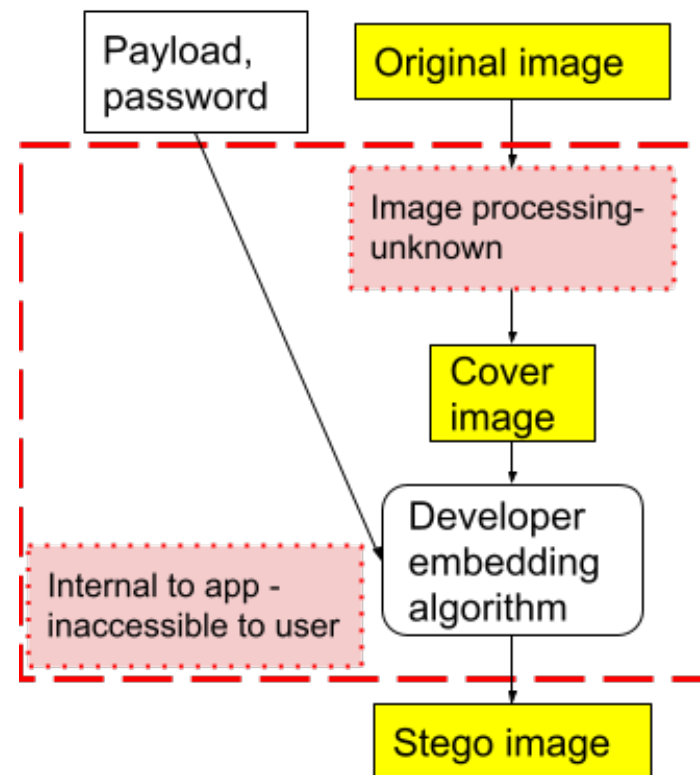


Testing Phase:
Class of Data Unknown



How a stego app works on a smartphone

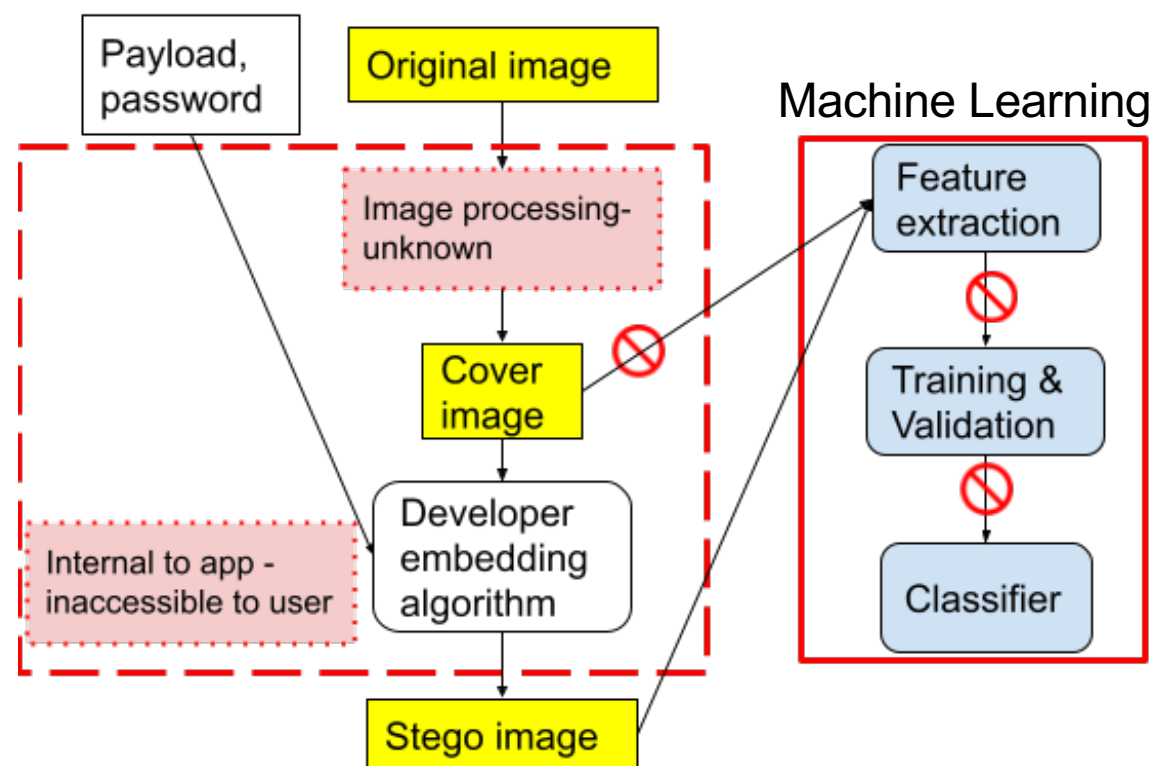
- Original image: on phone, use camera
- Some apps require password
- Almost all apps perform image processing on the image input to the app: resizing, cropping, format change, etc.
- The cover image: inaccessible to user
- Embedding algorithm: inaccessible to user/researcher (generally)
- Choice of embedding rate not available to user/researcher



How a stego app works on a smartphone

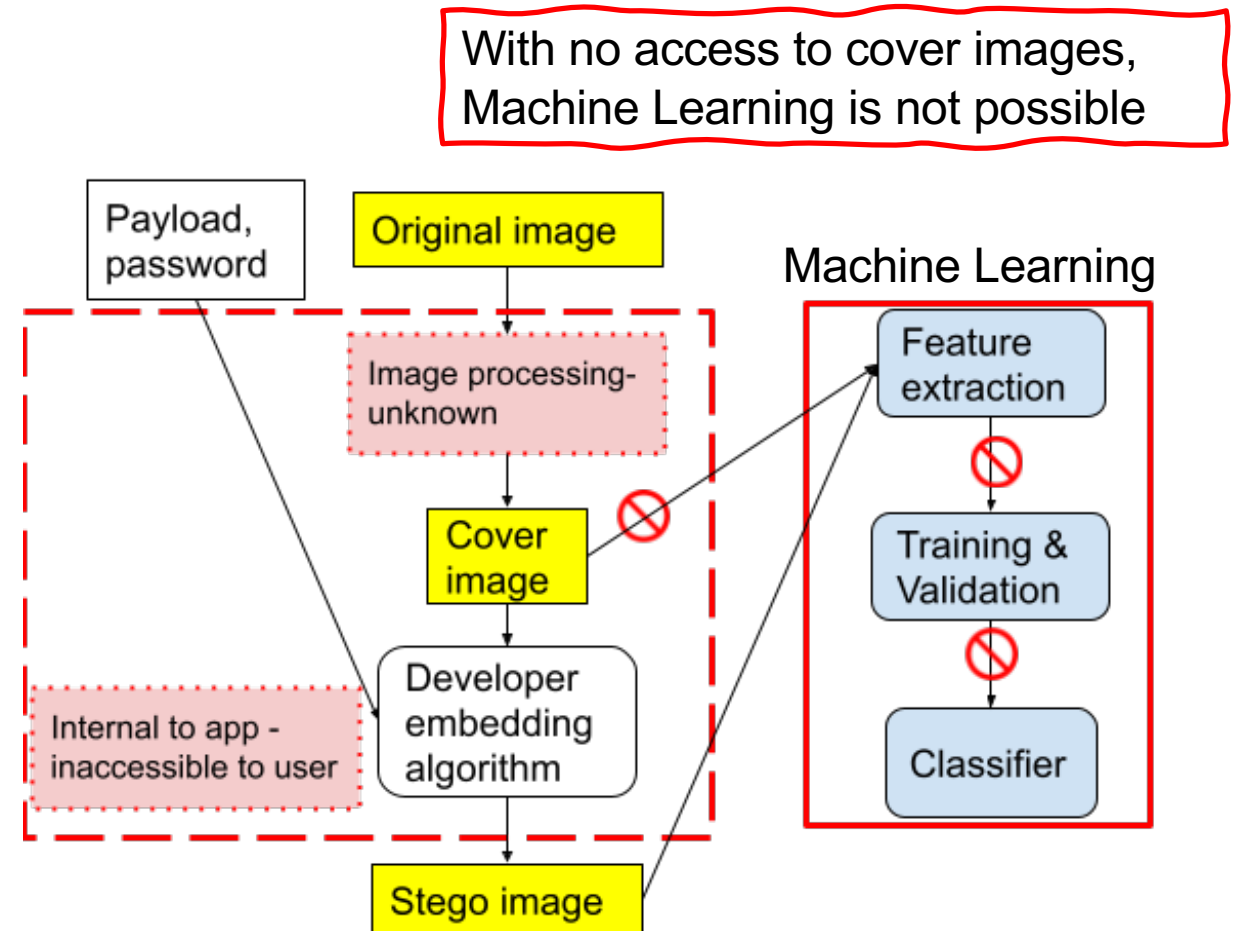
- Original image: on phone, use camera
- Some apps require password
- Almost all apps perform image processing on the image input to the app: resizing, cropping, format change, etc.
- The cover image: inaccessible to user
- Embedding algorithm: inaccessible to user/researcher (generally)
- Choice of embedding rate not available to user/researcher

With no access to cover images, Machine Learning is not possible



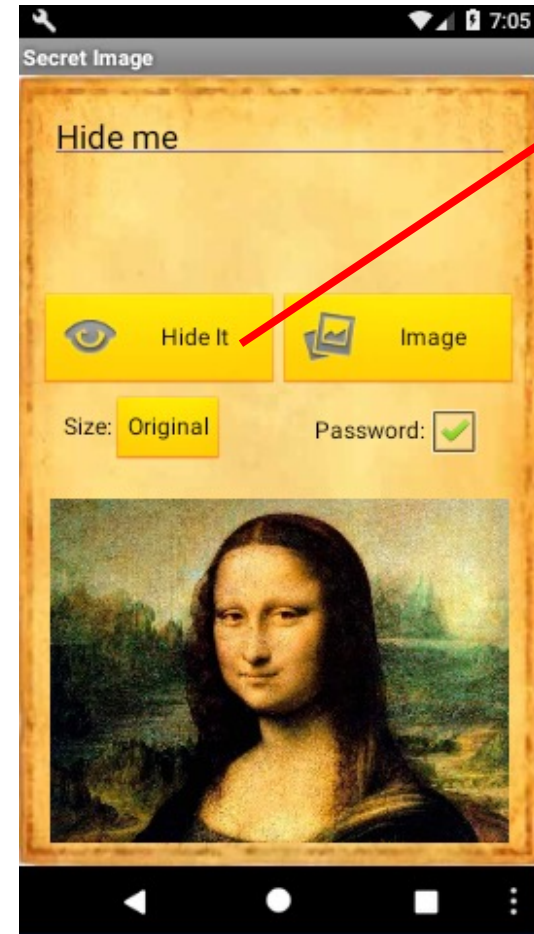
Other issues

- How to produce stegos with specified embedding rates?
- How to generate large amounts of stegos?
- **Answer:**
- We reverse-engineered several android apps
- We used some open-source code (PixelKnot, Pictograph)



Reverse Engineer Android Stego Apps

- Reverse engineering process*
 1. Identify important GUI widgets
 2. Locate corresponding callback methods
 3. Analyze binary code with control flow graphs
- Tools used:
 - APKTool, Smali



```
<node resource-  
id="id/action"  
text="Hide It"/>
```

```
id/action = 0x7f050003
```

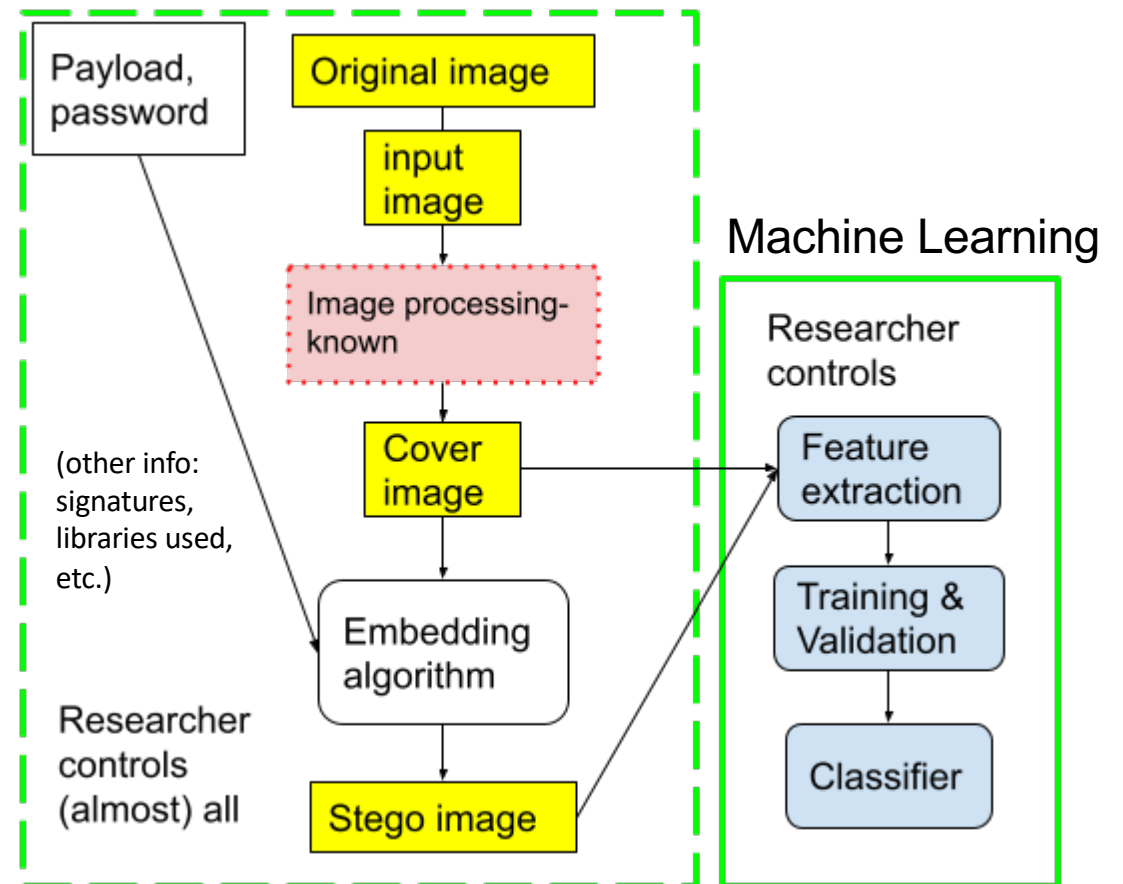
```
findViewById(0x7f050003)  
.setOnClickListener(...);
```

```
public void onClick(View  
v)  
{  
    ....  
}
```

*Forensic analysis of android steganography apps,” In G. Peterson and S. Shenoi, eds., Advances in Digital Forensics XIV, Cham. Springer Int’l. Publishing, pp. 293-312, 2018.

Modified app code allows generation of desired data*

- Researcher can:
 - Obtain intermediate images - **Covers**
 - **Identify the image processing** applied to the Original/Input image inside the app code
 - **Determine the precise embedding algorithm** the developer implemented
 - **Implement specific embedding rates of own choosing** not available through using the app directly
 - Create Machine Learning algorithms needing **pairs of cover-stego images**



*W. Chen, L. Lin, M. Wu, Y. Guan, and J. Newman. "Tackling Android Stego Apps in the Wild," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, pp. 1564-1573, 2018.

StegoAppDB image types: Defined

- **Original image**
 - Acquired by a device using Cameraw (our app), original dimensions native to camera app
 - “Original Image” is one selected by user of steg app (library, use phone camera)
- **Input image:** allows researcher to input specific images created by herself, like cropped
- **Cover image**
 - Processed input or original image using developer’s code
 - Cover image has same image file format as the Stego image that is output by app
 - Cover image is equivalent to 0% rate embedded Stego image
- **Stego image**
 - Output image with hidden message
 - Stego image has same dimensions as Cover image

Example of image types from StegoAppDB



Original image

Dimensions	4032 X 3024
Size	2.1 MB
Camera	Pixel2-1
Format	JPEG
Colorspace	RGB
JPEG Qual	90
ISO	500
Exposure time (s)	1/60



Input image

Dimensions	512 X 512
Size	188 KB
Format	PNG
Colorspace	Gray
Processing	Cropped
Processing	JPG -> Gray PNG



Cover image

Dimensions	512 X 512
Size	336 KB
Format	PNG
Colorspace	RGB/Gray
Processing	Known



Stego image

Dimensions	512 X 512
Size	337 KB
Format	PNG
Colorspace	RGB/Gray
Processing	Message embedded
Steg app	MobiStego
Emb. Rate	5 %



Stego image

Dimensions	512 X 512
Size	389 KB
Format	PNG
Colorspace	RGB/Gray
Processing	Message embedded
Steg app	MobiStego
Emb. Rate	10 %

Steganography images in StegoAppDB

- 20 images per scene
 - 10 JPEG (high-quality), 1 auto-exposure, 9 manual exposure
 - 10 DNG, 1 auto-exposure, 9 manual exposure
- Apps used to create steganography images
 - Android: PixelKnot, Passlok, MobiStego, PocketStego, Steganography (Meznik)
 - iPhone: Pictograph
- Center-cropped images used for stegos
- Embedding rates: 5%, 10%, 15%, 20%, 25%
- Message data: text from “The Complete Works of William Shakespeare” (online)
- Embedded message verified by extracting message for each stego after embedding



Count of images in StegoAppDB

Device Model	# Devices	ISO Range	Exposure Time Range	# Scenes	# Original Images	# Cropped Images	# Covers	# Stegos
Google Pixel 1	4	107 ~ 3735	1/120 ~ 1/12	284	5680	5680	25560	127800
Google Pixel 2	4	86 ~ 2927	1/249 ~ 1/12	286	5720	5720	25740	128700
Samsung Galaxy S8	2	57 ~ 6846	1/120 ~ 1/12	173	3460	3460	15570	77850
OnePlus 5	2	100 ~ 3000	1/9846 ~ 1/15	156	3120	3120	14040	70200
iPhone 6s	4	40 ~ 1600	1/60 ~ 1/3	284	5680	5680	5680	28400
iPhone 6s Plus	2	25 ~ 1250	1/66 ~ 1/3	181	3620	3620	3620	18100
iPhone 7	4	25 ~ 1000	1/60 ~ 1/3	285	5700	5700	5700	28500
iPhone 7 Plus	2	25 ~ 1000	1/80 ~ 1/3	185	3700	3700	3700	18500
iPhone 8	2	32 ~ 1250	1/60 ~ 1/3	142	2840	2840	2840	14200
iPhone X	2	20 ~ 1600	1/62 ~ 1/3	192	3840	3840	3840	19200
Total	28	20 ~ 6846	1/9846 ~ 1/3	2168	43360	43360	106290	531450
Total Images								724460

Image types in DB: difference between academic and mobile settings

Academic use
only

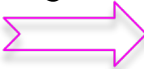


Image Type	Known to steganalyzer in academic setting	Known to steganalyzer in mobile app setting
Original – acquired by camera device	image values, meta data, dimensions, etc.	image values, meta data, dimensions, etc.
[Input – researcher processes from original (downsize, crop)]	[downsized/ cropped/ reformatted (i.e. tif -> jpeg)]	[downsized/ cropped/ reformatted (i.e. tif -> jpeg)]
Cover – processed by developer's app code	Known & equal to input	Unknown
Stego- image values & dimensions	Known	Known
Password	Known	Known
Embedding Algorithm	Known	Unknown

StegoAppDB webpages



<https://forensicstats.org/stegoappdb/>

[ABOUT](#) [RESEARCH AREAS](#) [RESOURCES](#) [LEARNING OPPORTUNITIES](#) [NEWS & EVENTS](#) [CONTACT](#) [Q](#)



StegoAppDB: A FORENSICS IMAGE DATABASE FOR MOBILE STEGANOGRAPHY

Contact Us
stegoappdb@iastate.edu

[jlnewman@iastate.edu](#)
Learn More About Steganography
Download Information and Instructions
License and Acknowledgements

SAMPLE IMAGES



StegoAppDB, a steganography apps forensics image database, is a database of



Stego App DB

A Forensics Image Database for Mobile Steganography

[StegoAppDB FAQs](#)

Search For: ☒ Stego Images ☐ Original Images

Stego-related Images: ☒ Stego images
☐ Include pre-stego images (cover and input) [?](#)
☐ Include original images [?](#)

Embedding Program:

Android	Apple
<input type="checkbox"/> PixelKnot (JPG) <input type="checkbox"/> Passlok (JPG) <input type="checkbox"/> MobiStego (PNG) <input type="checkbox"/> PocketStego (PNG) <input type="checkbox"/> Steganography-Meznik (PNG)	<input type="checkbox"/> Pictograph (PNG)

Original Image Source Device:

	Select All	Deselect All	Device Number
			1 2 3 4
OnePlus 5	<input type="checkbox"/>	<input type="checkbox"/>	
Pixel 1	<input type="checkbox"/>	<input type="checkbox"/>	
Pixel 2	<input type="checkbox"/>	<input type="checkbox"/>	
Samsung Galaxy S7	<input type="checkbox"/>	<input type="checkbox"/>	
Samsung Galaxy S8	<input type="checkbox"/>	<input type="checkbox"/>	

	Select All	Deselect All	Device Number
			1 2 3 4
iPhone6s	<input type="checkbox"/>	<input type="checkbox"/>	
iPhone6sPlus	<input type="checkbox"/>	<input type="checkbox"/>	
iPhone7	<input type="checkbox"/>	<input type="checkbox"/>	
iPhone7Plus	<input type="checkbox"/>	<input type="checkbox"/>	
iPhone8	<input type="checkbox"/>	<input type="checkbox"/>	
iPhoneX	<input type="checkbox"/>	<input type="checkbox"/>	

Embedding Rate: ☐ 0% < rate ≤ 10% ☐ 10% < rate ≤ 20% ☐ 20% < rate ≤ 40%

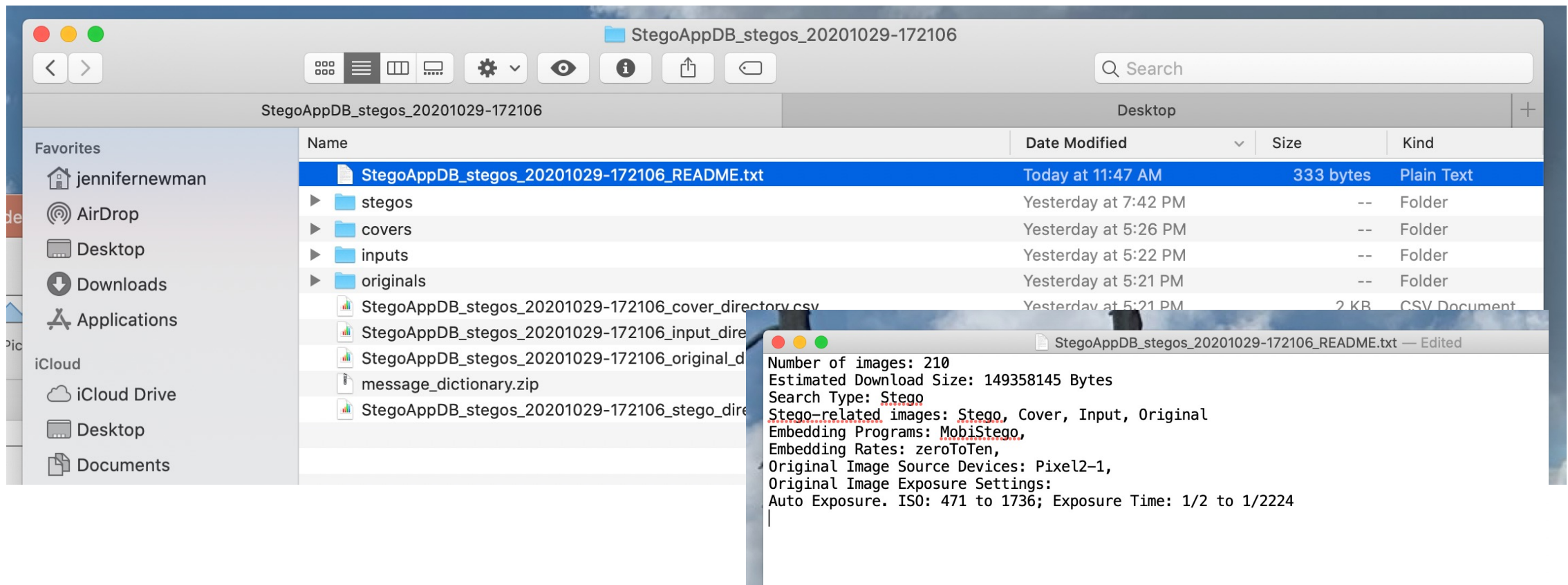
Original Image Exposure Settings:

	ISO	Exposure Time
<input type="checkbox"/> Auto Exposure	10 - 7000	1/11000 - 1/2
<input type="checkbox"/> Manual Exposure	10 - 7000	1/11000 - 1/2

[Search](#) [Clear Form](#)

Center for Statistics and Applications in Forensic Evidence
Center for Survey Statistics and Methodology, Iowa State University
Last updated: 3/29/2021

Download files csv files (≤ 3), folder structure, & readme



CSV File: original image information

StegoAppDB_stegos_20201029-172106_original_directory

original_image_id	original_image_filename	image_type	image_bytes	image_source_device	device_model	image_format	jpg_quality	make	camera_model_name	exposure_time	f_number	iso	focal_length	exposure_mode	white_balance	image_width	image_height	device_index	scene_label
611381	611381.JPG	original	2803793	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	604.0	4.4 mm	Auto	Auto	4032	3024	1	
611961	611961.JPG	original	2750633	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	536.0	4.4 mm	Auto	Auto	4032	3024	1	
615441	615441.JPG	original	3058133	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/30	1.8	844.0	4.4 mm	Auto	Auto	4032	3024	1	
616021	616021.JPG	original	2697701	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	624.0	4.4 mm	Auto	Auto	4032	3024	1	
616601	616601.JPG	original	2547881	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	512.0	4.4 mm	Auto	Auto	4032	3024	1	
617181	617181.JPG	original	2331672	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	628.0	4.4 mm	Auto	Auto	4032	3024	1	
618921	618921.JPG	original	2971292	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	616.0	4.4 mm	Auto	Auto	4032	3024	1	
619501	619501.JPG	original	2930355	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/30	1.8	896.0	4.4 mm	Auto	Auto	4032	3024	1	
620081	620081.JPG	original	3143739	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	520.0	4.4 mm	Auto	Auto	4032	3024	1	
622981	622981.JPG	original	3003488	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/30	1.8	728.0	4.4 mm	Auto	Auto	4032	3024	1	
624721	624721.JPG	original	3120012	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	820.0	4.4 mm	Auto	Auto	4032	3024	1	
627041	627041.JPG	original	2449145	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/120	1.8	552.0	4.4 mm	Auto	Auto	4032	3024	1	
629361	629361.JPG	original	2505865	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	636.0	4.4 mm	Auto	Auto	4032	3024	1	
629941	629941.JPG	original	2843700	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	680.0	4.4 mm	Auto	Auto	4032	3024	1	
631101	631101.JPG	original	2520249	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	764.0	4.4 mm	Auto	Auto	4032	3024	1	
632261	632261.JPG	original	2364639	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	592.0	4.4 mm	Auto	Auto	4032	3024	1	
632841	632841.JPG	original	2574457	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	504.0	4.4 mm	Auto	Auto	4032	3024	1	
633421	633421.JPG	original	2344791	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	548.0	4.4 mm	Auto	Auto	4032	3024	1	
634001	634001.JPG	original	2543152	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	552.0	4.4 mm	Auto	Auto	4032	3024	1	
634581	634581.JPG	original	2370022	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	500.0	4.4 mm	Auto	Auto	4032	3024	1	
636901	636901.JPG	original	2369295	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/60	1.8	568.0	4.4 mm	Auto	Auto	4032	3024	1	
642121	642121.JPG	original	3790868	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	544.0	4.4 mm	Auto	Auto	4032	3024	1	
644441	644441.JPG	original	3567800	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	944.0	4.4 mm	Auto	Auto	4032	3024	1	
646181	646181.JPG	original	2944128	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	836.0	4.4 mm	Auto	Auto	4032	3024	1	
646761	646761.JPG	original	2584564	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/40	1.8	944.0	4.4 mm	Auto	Auto	4032	3024	1	
647921	647921.JPG	original	2100108	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/24	1.8	1020.0	4.4 mm	Auto	Auto	4032	3024	1	
648541	648541.JPG	original	2440740	Pixel2-1	Pixel2	JPG	90	Google	Pixel 2	1/24	1.8	856.0	4.4 mm	Auto	Auto	4032	3024	1	

CSV File: input & cover image info

StegoAppDB_stegos_20201029-172106_input_directory

input_image_id	input_image_filename	image_type	original_image_id	input_image_bytes
611382	611382.PNG	input	611381	122542
611962	611962.PNG	input	611961	135185
615442	615442.PNG	input		
616022	616022.PNG	input		
616602	616602.PNG	input		
617182	617182.PNG	input		
618922	618922.PNG	input		
619502	619502.PNG	input		
620082	620082.PNG	input		
622982	622982.PNG	input		
624722	624722.PNG	input		
627042	627042.PNG	input		
629362	629362.PNG	input		

StegoAppDB_stegos_20201029-172106_cover_directory

cover_image_id	cover_image_filename	image_type	input_image_id	cover_image_bytes
611407	611407.PNG	cover	611382	197864
611987	611987.PNG	cover	611962	220071
615467	615467.PNG	cover	615442	213230
616047	616047.PNG	cover	616022	256997
616627	616627.PNG	cover	616602	219667
617207	617207.PNG	cover	617182	197194
618947	618947.PNG	cover	618922	223766
619527	619527.PNG	cover	619502	207928
620107	620107.PNG	cover	620082	217055
623007	623007.PNG	cover	622982	204369
624747	624747.PNG	cover	624722	245149
627067	627067.PNG	cover	627042	191813
629387	629387.PNG	cover	629362	241088

CSV File: stego image information

- Heavily provenanced
- Name of original image used in creation path to stego image
- Mobile app used; message content; embedding rate; etc.

StegoAppDB_stegos_20201029-172106_stego_directory																			
image_id	image_filename	image_type	image_bytes	cover_image_id	cover_image_filename	input_image_id	input_image_filename	original_image_id	original_image_filename	exposure_mode	exposure_time	iso	image_source_device	embedding_method	embedding_rate	message_length	message_dictionary	message_starting_index	password
611408	611408.PNG	stego	213528	611407	611407.PNG	611382	611382.PNG	611381	611381.JPG	Auto	1/60	604.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_henryv.txt	2938	N/A
611409	611409.PNG	stego	225209	611407	611407.PNG	611382	611382.PNG	611381	611381.JPG	Auto	1/60	604.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_cleopatra.txt	953	N/A
611988	611988.PNG	stego	233417	611987	611987.PNG	611962	611962.PNG	611961	611961.JPG	Auto	1/40	536.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_pericles.txt	833	N/A
611989	611989.PNG	stego	245577	611987	611987.PNG	611962	611962.PNG	611961	611961.JPG	Auto	1/40	536.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_othello.txt	1992	N/A
615468	615468.PNG	stego	227111	615467	615467.PNG	615442	615442.PNG	615441	615441.JPG	Auto	1/30	844.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_cymbeline.txt	2016	N/A
615469	615469.PNG	stego	238224	615467	615467.PNG	615442	615442.PNG	615441	615441.JPG	Auto	1/30	844.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_henryviii.txt	1924	N/A
616048	616048.PNG	stego	284309	616047	616047.PNG	616022	616022.PNG	616021	616021.JPG	Auto	1/40	624.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_comedy_errors.txt	354	N/A
616049	616049.PNG	stego	297822	616047	616047.PNG	616022	616022.PNG	616021	616021.JPG	Auto	1/40	624.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_comedy_errors.txt	1076	N/A
616628	616628.PNG	stego	242327	616627	616627.PNG	616602	616602.PNG	616601	616601.JPG	Auto	1/40	512.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_hamlet.txt	2086	N/A
616629	616629.PNG	stego	253974	616627	616627.PNG	616602	616602.PNG	616601	616601.JPG	Auto	1/40	512.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_tempest.txt	1155	N/A
617208	617208.PNG	stego	214804	617207	617207.PNG	617182	617182.PNG	617181	617181.JPG	Auto	1/60	628.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_richardiii.txt	832	N/A
617209	617209.PNG	stego	226413	617207	617207.PNG	617182	617182.PNG	617181	617181.JPG	Auto	1/60	628.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_othello.txt	4035	N/A
618948	618948.PNG	stego	240730	618947	618947.PNG	618922	618922.PNG	618921	618921.JPG	Auto	1/60	616.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_henryvi.txt	10723	N/A
618949	618949.PNG	stego	251936	618947	618947.PNG	618922	618922.PNG	618921	618921.JPG	Auto	1/60	616.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_taming_shrew.txt	2131	N/A
619528	619528.PNG	stego	224356	619527	619527.PNG	619502	619502.PNG	619501	619501.JPG	Auto	1/30	896.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_III.txt	2009	N/A
619529	619529.PNG	stego	235400	619527	619527.PNG	619502	619502.PNG	619501	619501.JPG	Auto	1/30	896.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_merchant.txt	313	N/A
620108	620108.PNG	stego	232952	620107	620107.PNG	620082	620082.PNG	620081	620081.JPG	Auto	1/60	520.0	Pixel2-1	MobiStego	0.049997967	9824	shakespeare_henryv.txt	2502	N/A
620109	620109.PNG	stego	244486	620107	620107.PNG	620082	620082.PNG	620081	620081.JPG	Auto	1/60	520.0	Pixel2-1	MobiStego	0.09999593	19654	shakespeare_III.txt	1389	N/A

Thank you!



Questions?