StegoAppDB: A Benchmark Database for Mobile Steganography

Li Lin, Wenhao Chen, Stephanie Reinders, Min Wu*, Yong Guan, Jennifer Newman

*Iowa State University, *University of Maryland, College Park*

Project Rationale & Goals

Why we need a reference StegoAppDatabase :

1. Stego apps are popular, simple, and stealthy.
2. No current database has images reflective of mobile stego cases.
3. No algorithms/software have successfully detected images from stego apps.

What we can do with StegoAppDatabase:

1. Test current tools/software on mobile stego images.
2. Create/benchmark new algorithms.
3. Potential applications to other digital image forensics problems.

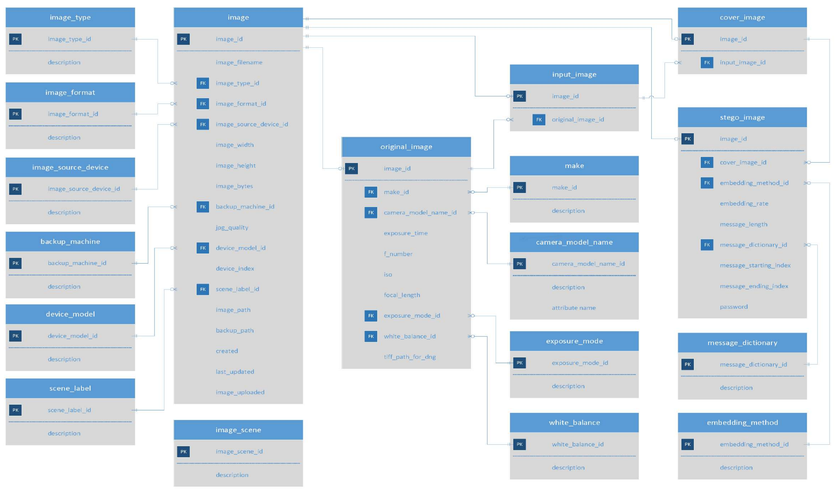
What challenges we are facing:

1. Forensics applications require data authentication and copyright/privacy-free.
2. Each app has unique characteristics and requires reverse-engineering.
3. Need a fast and efficient method to batch-generate many images.

Example: Steganography Apps in Google Play Store

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

Entity Relationship Diagram of the StegoAppDatabase



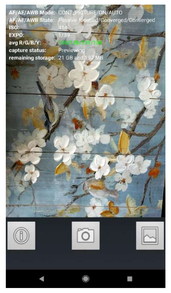
Data Collection and Current Status of the Database

24 Lab Smartphones (2+ phones per model)

Google Pixel 1
Google Pixel 2
HTC 10
OnePlus 5
Samsung Galaxy S7
Samsung Galaxy S8

iPhone 6S
iPhone 6S Plus
iPhone 7
iPhone 7Plus
iPhone 8
iPhone X

48,240 original JPEGs and DNGs
100 images per scene, 2,412 labeled scenes
10 different exposure settings for each scene

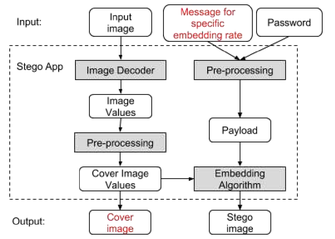


We developed a camera app for efficient data collection:

- **"Cameraw" for Android and iOS**
- **Takes 20 images with one click**
- **Captures both JPEG and DNG**
- **Photographer assigns scene label after capture session**
- **Image metadata are saved along with the collected images**

595,400 stego images on 5 different embedding rates, generated from 7 stego apps:

We reverse-engineer the stego apps and batch-generate cover and stego images, with specific embedding rates.



Evaluation, Results & Discussion

Evaluation of existing steg detection programs on 2000 cover images and 2000 stego images created by the standard F5 embedding algorithm:

Error Type	Programs		
	Stego Hunt	DC3 StegDetect	Provos StegDetect
False Alarm	0.0%	0.0%	24.6%
Missed Detection	5.2%	0.0%	47.4%
Average Error	2.6%	0.0%	36.0%

Evaluation of existing steg detection programs on 4818 cover images and 4818 stego images created by the StegoApp *PixelKnot:*

Error Type	Programs		
	Stego Hunt	DC3 StegDetect	Provos StegDetect
False Alarm	0%	0%	24%
Missed Detection	100%	100%	76%
Average Error	50%	50%	50%

Conclusion: Existing steg detection programs do not perform well on detecting Stego Images created by Apps. [***Appeared in IFIP 11.9 2018***] [1]

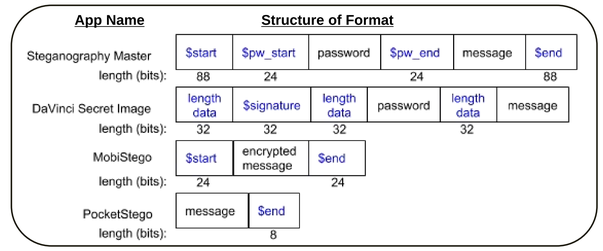
Type of Detection	Advantages
Machine Learning	• Reverse engineering is not required • Run with zero knowledge of signature • Can be retrained to include other stego embedding methods
Signature/Hash-Based	• Very fast to execute, high accuracy • Identifies many types of files, not just images

ML-based Steg Detection: Detecting StegoApp *PixelKnot* using *CC-JRM* features and *FLD* ensemble classifier training on 500 cover-stego pairs of images at 10% embedding rate.

Data Source	Original Format	Input Image Size	Output Image Size	Accuracy
Google Pixel	JPEG	4048*3036	1280*960	99.0%
Samsung S7	JPEG	4032*3024	1280*960	92.4%
OnePlus 5	JPEG	4608*3456	1280*960	99.7%
All	JPEG	-	1280*960	96.8%

Conclusion: Knowing the data source and stego apps, an academic ML-based detection method performs well on the stego images generated from *PixelKnot*. ***[Appeared in APSIPA 2018]*** [2]

Signature-based Steg Detection. It is possible to detect stego images from mobile apps that embed signatures.



Stego App	Test Images	Image Count	Accuracy
Steganography Master	SM Stego Images	42,100	100%
	Other Images	159,980	100%
DaVinci Secret Image	DV Stego Images	42,100	100%
	Other Images	159,980	100%
MobiStego	MS Stego Images	42,100	100%
	Other Images	159,980	100%

1. W. Chen, Y. Wang, Y. Guan, J. Newman, L. Lin, and S. Reinders. "Forensic analysis of android steganography apps," In G. Peterson and S. Shenoi, eds., Advances in Digital Forensics XIV, Cham. Springer Int'l. Publishing, 2018, pp. 293-312.

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

The material presented here is based upon work supported by the National Institute of Science and Technology under Award No. 70NANB15H176. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Institute of Science and Technology, nor of the Center for Statistics and Applications in Forensic Evidence.