Stego-Malware Attribution: Simple Signature and Content-based Features Derived and Validated from Classical Image Steganalysis on Five Exemplary Chosen Algorithms

Bernhard Birnbaum, Christian Kraetzer and Jana Dittmann
Contact author: christian.kraetzer {at} ovgu.de

Advanced Multimedia and Security Lab (AMSL)
Department of Computer Science
Otto-von-Guericke University Magdeburg
Magdeburg, Germany

The 17. International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2023)
September 25, 2023 to September 29, 2023 - Porto, Portugal

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Short resume of the presenter

Christian Kraetzer is a researcher at the Multimedia and Security Group of the Otto-von-Guericke University Magdeburg (OVGU) in Germany. He received his Ph.D. in Computer Science in 2013 for a dissertation project focused on applied pattern recognition for audio forensics tasks. His current research interests include explainable AI solutions for multimedia forensics, steganalysis, biometrics, and cryptography.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Outline

- Introduction to steganography and steganalysis
- Summary of the contributions of this paper
- Evaluation setup
- Evaluation results discussion
- Summary and conclusion

Steganography

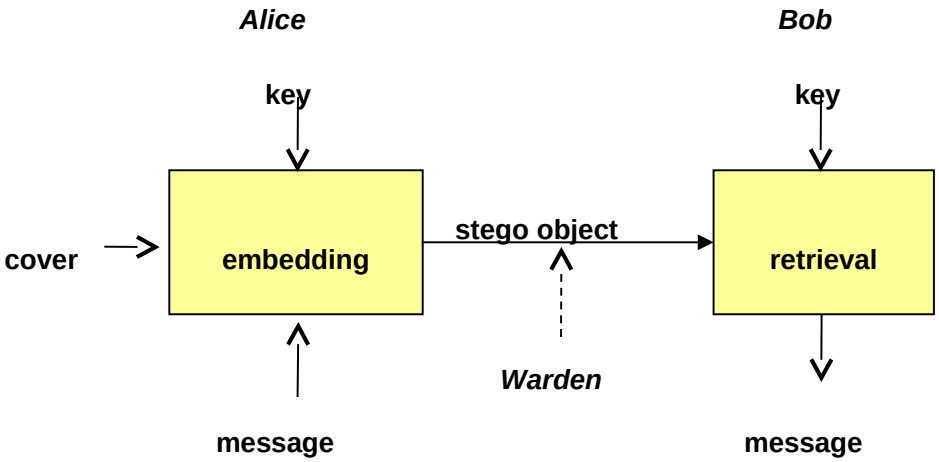
- Art and science of hidden communication

- Definition of steganographic paradigms (e.g. [Fridrich2010]):
  - steganography by cover modification
  - steganography by synthesis
  - steganography by selection
- Massive domination of the academic literature by steganography by cover modification, assumedly due to capacity and message coding aspects

[Fridrich2010] Jessica Fridrich: “Steganography in digital media: principles, algorithms, and applications”. Cambridge, UK: Cambridge University Press. 2010.

Traditional channel model for end to end steganography (A-B scenario)

- Est. by Simmons in 1983 ([Simmons1983]; adopting a metaphor from Wilkins from 1694)
- Hypothetic 3-party setup with Alice and Bob as communicating entities and Warden as observer
- Simplified basic assumptions:
  - key syncronisation problem solved
  - Warden is not capable of cover-stego-attacks
- Warden might be active (i.e., might change elements in the communication channel)



[Simmons1983] Gustavus J. Simmons: The Prisoners' Problem and the Subliminal Channel. Proc. Advances in Cryptology, Proceedings of CRYPTO '83, Santa Barbara, California, USA, August 21-24, 1983.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Stego-Malware: A recent trend in steganography and Malware development

- Current malware (short for Malicious Software) usually created using specialized integrated development environments (so called malware creation kits)
- More and more modern malware families come with plug-ins for steganographic channels in their malware creation kits
- Examples:
  - 2011 Duqu („*Most intriguing, though, is the way Duqu transmits information back to its control centre. It first encrypts this information and then embeds it in a JPEG file* […]" - https://www.technologyreview.com/s/529071/the-growing-threat-of-network-based-steganography/ - last access 11. September 2023)
  - 2015 Duqu 2.0
- Goals of the steganographic channels: Covert data exfiltration, covert command and control (C&C) communication or stealthy infiltration of malicious code to bypass endpoint security (anti-virus) solution

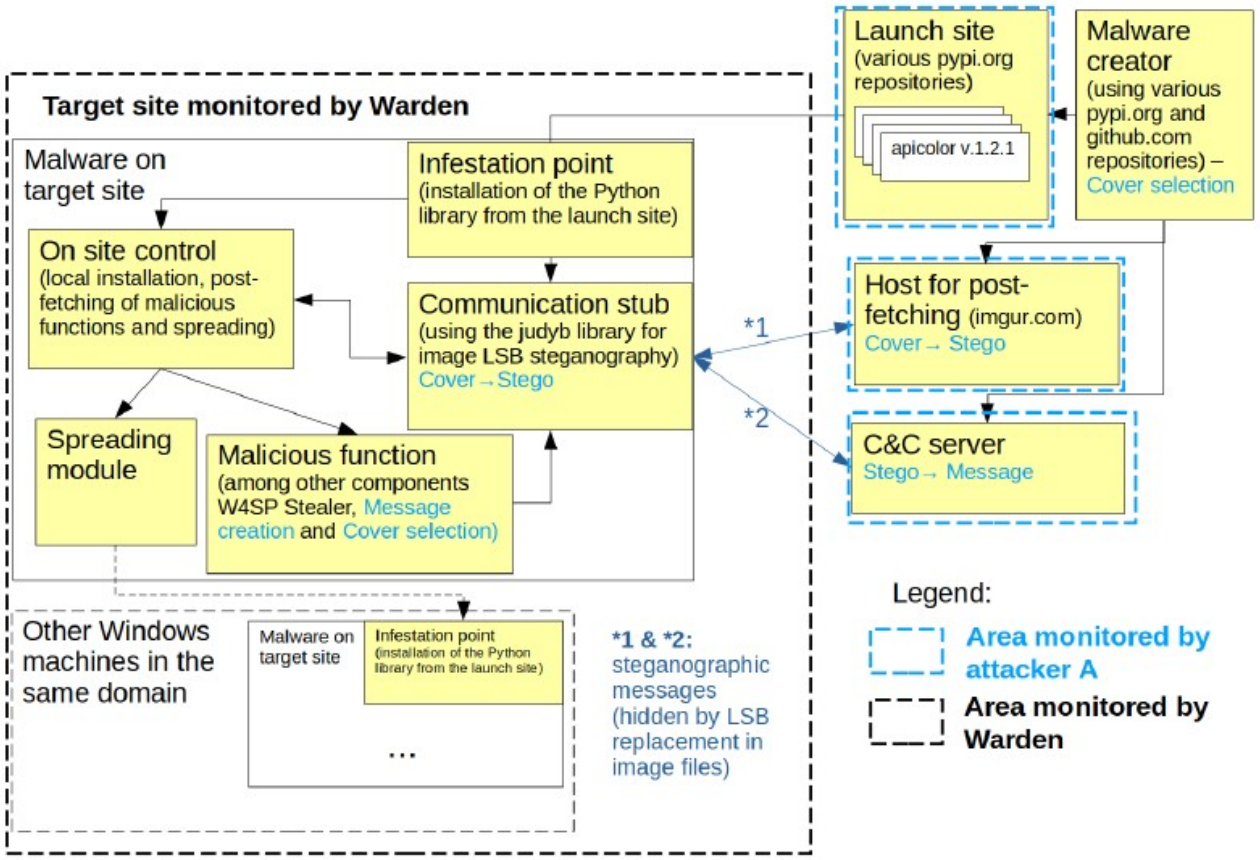
Important characteristics of stego-malware

- Significant deviations from the traditional ‘Alice and Bob’ (A-B) end-to-end communication scheme:
  - The classical ‘Alice and Bob’ (A-B) end-to-end communication scenario with a ‘Warden’ monitoring the channel is changed to a A-A scenario with the attacker A controlling both ends of the communication and the Warden observing the target system and its incoming and outgoing communication
  - As the attacker A activities at the target system are fully observable, all attacker actions should be as limited as possible to raise no suspicion
  - As a result, non-Kerckhoffs’ setups (key-less techniques, hard-coded keys or key infiltrated together with the payload of the hidden communication) are often used
  - Simple embedding and retrieval techniques are used natively in the target domain or are put in place with supply-chain-attacks
  - The cover selection and/or cover embedding at the target system can be observed by the warden, therefore blind as well as non-blind analysis of cover and stego objects used are possible for the Warden

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Example of the stego-malware communication scenario (A-A scenario) discussed in the paper



Steganalysis in general

- Counter science to steganography, typically modeled as a 2-class decision problem: unmodified cover vs. stego object
- Very few existing publications (e.g. [Provos2001]) model steganalysis as a multi-class problem (one class per potentially used steganographic embedding scheme plus one class for unmodified cover data)
- Technical solutions: either trained detector or signature based detection, the first option is far more often considered in literature

[Provos2001] N. Provos and P. Honeyman, "Hide and seek: an introduction to steganography." in IEEE Security & Privacy, vol. 1, no. 3, pp. 32-44, May-June 2003, doi: 10.1109/MSECP.2003.1203220.

Steganalysis in the stego malware case

- Due to the inherent characteristics of the A-A scenario and the increased capabilities of the Warden in this context, blind as well as non-blind steganalysis approaches are possible for the Warden
- Detection can be signature based (e.g., using characteristic patterns in the stego file structure or metadata) or could be model-based, focussing on the detection of embedding patterns in the actual signal (e.g., traces/artefacts created by the embedding in the pixel values of a PNG image)
- For many of the simple embedding and retrieval techniques found in stego malware, readily usable detection methods and tools already exist, e.g., the methods described in [Verma2022] and [Cohen2020] or the tools provided in steganalysis toolsets like Aletheia (https://github.com/daniellerch/aletheia)

[Verma2022] V. Verma, S. K. Muttoo, and V. B. Singh, “Detecting stegomalware: Malicious image steganography and its intrusion in windows,” in Security, Privacy and Data Analytics (U. P. Rao, S. J. Patel, P. Raj, and A. Visconti, eds.), (Singapore), pp. 103–116, Springer Singapore, 2022.

[Cohen2020] A. Cohen, N. Nissim, and Y. Elovici, “Maljpeg: Machine learning based solution for the detection of malicious jpeg images,” IEEE Access, vol. 8, pp. 19997–20011, 2020.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Our contributions

- Contribution(s) of this paper:
  - Addressing the attribution task for image steganalysis: Finding traces indicating on the potential source of a stego-malware, based on the characteristics of the stego objects
  - Providing an exemplary empirical study on multi-class, signature-driven steganalysis with five existing methods for image steganography: *jphide*, *jsteg*, *outguess*,
    steghide and *f5*
  - Introduction of a set of light-weight (i.e., easy to compute) blind and non-blind features to enable the attribution of aforementioned image steganography methods
  - Determination of estimates for detection rates as well as false alarm rates on two established image test sets: the Alaska2 [Alaska2] image steganalysis reference database and the Flickr30k data set [Young2014]

[Alaska2] Kaggle: “Alaska2 image steganalysis set,” https://www.kaggle.com/competitions/alaska2-image-steganalysis/data - last accessed: August 21st, 2023, July 2020.

[Young2014] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” TACL, vol. 2, pp. 67–78, 2014.

Evaluation setup (1/2)

- Image steganography tools (taken from [Breuker2020]): *jphide*, *jsteg*, *outguess*, *steghide* and *f5*
- General analysis tools: *exiftool*, *binwalk*, *foremost*, *strings*, *imagemagick* modules ‘identify’ and ‘compare’

- Image sets for evaluation:
  - 1000 randomly chosen specimen are sampled as covers from the established image steganalysis reference dataset ‘Alaska2’ [Alaska2]
  - To provide a significant amount of wide variance image data to establish potential false positive rates for attribution the ‘Fickr30k’ dataset from [Young2014] is used

[Breuker2020] D. Breuker: “Steganography-toolkit,” (2020) https://github.com/DominicBreuker/stego-toolkit - last accessed: Sept. 11th, 2023.

[Alaska2] Kaggle: “Alaska2 image steganalysis set,” https://www.kaggle.com/competitions/alaska2-image-steganalysis/data - last accessed: August 21st, 2023, July 2020.

[Young2014] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” TACL, vol. 2, pp. 67–78, 2014.

Evaluation setup (2/2)

- Embedding options used:
  - two different message capacities (embedding data: ASCII text of 26 Bytes (‘low’ capacity scenario) and 2.1 kBytes (‘high’ capacity scenario) length)
  - two keys of different length (4 Bytes (=‘short’) and 128 Bytes (=‘long’) are used.
  - Only *jsteg* does not support a key as a parameter and therefore the embedding that case is key-less (‘no key’).

Feature space
Blind / signature-based attributes (1/2)

- ba1: JFIF version information as extracted by *exiftool*; this feature is considered anomalous if the JFIF version value cannot be successfully retrieved from the JPEG header
- ba2: image data type as extracted by *binwalk*; considered anomalous if file header is corrupted
- ba3: output of the file carving tool *foremost*; anomalous if image headers appear to be damaged or the JPEG image format integrity is otherwise violated
- ba4 : All tools seem to leave specific traces in the COM sections of the JPEG file header or the quantization tables. This is a weakness that many stego tools share, because they use in many cases non-standard JPEG libraries and do not write correct or plausible JPEG/JFIF metadata

Feature space

Blind / signature-based attributes (2/2)

SIGNATURES FOR FEATURE $\mathtt{BA}_4$ - CHECKED ON ITS PRESENCES: IF $\mathtt{sig}_1$, $\mathtt{sig}_3$ AND $\mathtt{sig}_5$ ARE NOT PRESENT AND $\mathtt{sig}_2$ AND $\mathtt{sig}_4$ ARE PRESENT THE CORRESPONDING STEGO ALGORITHM IS DETECTED

signature id	stego algorithm	signature content
$\mathtt{sig}_1$	*jsteg*	JFIF
$\mathtt{sig}_2$	*f5*	(((
$\mathtt{sig}_3$	*jphide*	56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
$\mathtt{sig}_4$	*outguess*, *steghide*	!22222222222222222222222222222222222222222222
$\mathtt{sig}_5$	*outguess*, *steghide*	Exif

Feature space

Non-Blind / content-based attributes (1/2)

- nba1: The idea behind this feature is to compare original and stego-image file sizes, as the stego tools *jphide* and *steghide* do not perform an image re-compression during embedding. This results in a stego file size that is much closer to the original compared to other stego algorithms. The following rule was derived by empirical analysis:

$$\texttt{diff. size} < \frac{\texttt{orig. size}}{2} \wedge \texttt{stego size} > \frac{\texttt{orig. size}}{2}$$

Feature space

Non-Blind / content-based attributes (2/2)

- nba2 : This feature is also motivated by the fact that *jphide* and *steghide* do not apply a re-compression So, when creating a differential image of the stego-manipulated image (by *jphide* or *steghide*) and the original, all visible changes are directly related to the embedded data.
  Usually, the amount of changes is much smaller than the changes related to a re-compression, except when embedding large amount of data. As stego manipulations are also visible in the different colour channels of the image, but changes in the RGB channels due to a re-compression are much higher than for stego embeddings, the following detection rule is derived by empirical analysis:

$$\texttt{diff. mean} > 127 \wedge \texttt{diff. mean} == \texttt{rgb diff. mean} \pm 1\%$$

Feature space
Summary

THE ATTRIBUTION FEATURES USED IN THIS PAPER WITH THE CORRESPONDING ANALYSIS TOOLS AND THE ENCODING OF THE RELEVANCE FOR THE FIVE CONSIDERED STEGANOGRAPHY TOOLS (`r`: RELEVANT, `u`: UNIQUE, `m`: MOTIVATED FROM, `n.a.`: NOT APPLICABLE)

feature id	feature name	feature type	analysis tools	*jphide*	*jsteg*	*outguess*	*steghide*	*f5*
signature-based features								
ba_1	*JFIF version*	blind	exiftool	n.a.	r, u, m	n.a.	n.a.	n.a.
ba_2	*binwalk extraction*	blind	binwalk	n.a.	r, u, m	n.a.	n.a.	n.a.
ba_3	*foremost carving*	blind	foremost	n.a.	r, u, m	n.a.	n.a.	n.a.
ba_4	*file header*	blind	strings	r	r, u	r	r	r, u, m
content-based features								
nba_1	*file size*	non-blind	exiftool	r	n.a.	n.a.	r, m	n.a.
nba_2	*difference color mean*	non-blind	imagemagick	r	n.a.	n.a.	r, m	n.a.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Attribution results for the five tested stego algorithms and implemented features (1/4)

feature	test config	in result list		positives		negatives		identification results				
		correct	incorrect	true p.	false p.	true n.	false n.	*jphide*	*jsteg*	*outguess*	*steghide*	*f5*
jphide (1000 images from Alaska2)												
ba_4	26B, long Key	1000	0	**1000**	0	0	0	**1000**	0	0	0	0
	rec; 26B, short Key	1000	0	**1000**	0	0	0	**1000**	0	0	0	0
	26B, short Key	1000	0	**1000**	0	0	0	**1000**	0	0	0	0
	2.1KB, short Key	1000	0	**1000**	0	0	0	**1000**	0	0	0	0
nba_1	26B, long Key	1000	1000	**0**	0	0	0	**1000**	0	0	1000	0
	rec; 26B, short Key	1000	1000	**0**	0	0	0	**1000**	0	0	1000	0
	26B, short Key	1000	1000	**0**	0	0	0	**1000**	0	0	1000	0
	2.1KB, short Key	1000	1000	**0**	0	0	0	**1000**	0	0	1000	0
nba_2	26B, long Key	923	923	**0**	0	0	77	**923**	0	0	923	0
	rec; 26B, short Key	775	775	**0**	0	0	225	**775**	0	0	775	0
	26B, short Key	921	921	**0**	0	0	79	**921**	0	0	921	0
	2.1KB, short Key	0	0	**0**	0	0	1000	**0**	0	0	0	0

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Attribution results for the five tested stego algorithms and implemented features (2/4)

feature	test config	in result list		positives		negatives		identification results				
		correct	incorrect	true p.	false p.	true n.	false n.	*jphide*	*jsteg*	*outguess*	*steghide*	*f5*
jsteg (1000 images from Alaska2)												
ba_1	26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	rec; 26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	2.1KB, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
ba_2	26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	rec; 26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	2.1KB, keyless	999	0	**999**	0	0	1	0	**999**	0	0	0
ba_3	26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	rec; 26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	2.1KB, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
ba_4	26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	rec; 26B, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
	2.1KB, keyless	1000	0	**1000**	0	0	0	0	**1000**	0	0	0
outguess (1000 images from Alaska2)												
ba_4	26B, long Key	1000	1000	**0**	0	0	0	0	0	**1000**	1000	0
	rec; 26B, short Key	1000	1000	**0**	0	0	0	0	0	**1000**	1000	0
	26B, short Key	1000	1000	**0**	0	0	0	0	0	**1000**	1000	0
	2.1KB, short Key	563	563	**0**	0	0	437	0	0	**563**	563	0

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Attribution results for the five tested stego algorithms and implemented features (3/4)

feature	test config	in result list		positives		negatives		identification results				
		correct	incorrect	true p.	false p.	true n.	false n.	*jphide*	*jsteg*	*outguess*	*steghide*	*f5*
steghide (1000 images from Alaska2)												
ba_4	26B, long Key	338	338	**0**	0	0	662	0	0	338	**338**	0
	rec; 26B, short Key	1000	1000	**0**	0	0	0	0	0	1000	**1000**	0
	26B, short Key	338	338	**0**	0	0	662	0	0	338	**338**	0
	2.1KB, short Key	241	241	**0**	0	0	640	0	0	241	**241**	0
nba_1	26B, long Key	1000	1000	**0**	0	0	0	1000	0	0	**1000**	0
	rec; 26B, short Key	1000	1000	**0**	0	0	0	1000	0	0	**1000**	0
	26B, short Key	1000	1000	**0**	0	0	0	1000	0	0	**1000**	0
	2.1KB, short Key	881	881	**0**	0	0	0	881	0	0	**881**	0
nba_2	26B, long Key	992	992	**0**	0	0	8	992	0	0	**992**	0
	rec; 26B, short Key	971	971	**0**	0	0	29	971	0	0	**971**	0
	26B, short Key	991	991	**0**	0	0	9	991	0	0	**991**	0
	2.1KB, short Key	0	0	**0**	0	0	881	0	0	0	**0**	0
f5 (1000 images from Alaska2)												
ba_4	26B, long Key	1000	0	**1000**	0	0	0	0	0	0	0	**1000**
	rec; 26B, short Key	1000	0	**1000**	0	0	0	0	0	0	0	**1000**
	26B, short Key	1000	0	**1000**	0	0	0	0	0	0	0	**1000**
	2.1KB, short Key	1000	0	**1000**	0	0	0	0	0	0	0	**1000**
all other		n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

Attribution results for the five tested stego algorithms and implemented features (4/4)

feature	test config	in result list		positives		negatives		identification results				
		correct	incorrect	true p.	false p.	true n.	false n.	*jphide*	*jsteg*	*outguess*	*steghide*	*f5*
imagemagick re-compression (1000 images from Alaska2, quality factor 75%)												
ba_1	genuine recompressed	0	0	0	0	**1000**	0	0	0	0	0	0
	genuine recompressed twice	0	0	0	0	**1000**	0	0	0	0	0	0
ba_2	genuine recompressed	0	0	0	0	**1000**	0	0	0	0	0	0
	genuine recompressed twice	0	0	0	0	**1000**	0	0	0	0	0	0
ba_3	genuine recompressed	0	0	0	0	**1000**	0	0	0	0	0	0
	genuine recompressed twice	0	0	0	0	**1000**	0	0	0	0	0	0
ba_4	genuine recompressed	0	1000	0	1000	**0**	0	1000	0	0	0	0
	genuine recompressed twice	0	1000	0	1000	**0**	0	1000	0	0	0	0
nba_1	genuine recompressed	0	1134	0	0	**433**	0	567	0	0	567	0
	genuine recompressed twice	0	2000	0	0	**0**	0	1000	0	0	1000	0
nba_2	genuine recompressed	0	650	0	0	**675**	0	325	0	0	325	0
	genuine recompressed twice	0	1960	0	0	**20**	0	980	0	0	980	0
genuine Flickr30k (31783 images from flickr)												
ba_1	original	0	58	0	58	**31725**	0	0	58	0	0	0
ba_2	original	0	0	0	0	**31783**	0	0	0	0	0	0
ba_3	original	0	0	0	0	**31783**	0	0	0	0	0	0
ba_4	original	0	31783	0	31783	**0**	0	31159	0	0	0	624

Performance of the attributes regarding different capacities and keys (1/2)

The blind signatures show:

- no influence from different capacities and keys in the performance for *jphide*, *jsteg* and *f5*, while *jphide* and *f5* have false classifications for the file header signatures of ba4, *jphide* with the re-compressed Alaska2 set and Flickr30k, *f5* with Flickr30k only
- for *outguess* and *steghide* the ba4 - file header signatures are sensitive: The high capacity with the short key influences the *outguess* file header signature (for the 2.1KB message, in 437 of the 1000 cases *outguess* could not successfully embed, which results for this tool in an empty (0 Byte) output file without a header – in the evaluations these cases are counted as false negatives since they are no genuine image files any longer but are also not flagged to be the output of this steganography tool). For *steghide*, the ba4 file header signature is (besides embedding problems that result in only 881 stego files being successfully created for the 2.1KB message) not only resulting in large numbers of false negatives for all cases except the re-compressed Alaska2 images with short capacity with the short key but it also lacks discriminatory power in regard to *steghide* and *outguess*

Performance of the attributes regarding different capacities and keys (2/2)

The non-blind (content-based) signatures show:

- both features are relevant for *steghide* and *jphide* but not in a unique manner
- the first content-based feature (nba1) of file size is in most cases capacities and keys independent but attributes *steghide* as well as *jphide* at the same time (in a not unique manner) with errors only in high capacity with the short key and with wrong classifications in the Alaska2 re-compression tests;
- the second non-blind feature of different color mean attributes (nba2) also *steghide* as well as *jphide* and is error prone to high capacity with the short key too and less sensitive for all other settings with wrong classifications in the Alaska2 re-compression tests

Performance of the attributes regarding the different algorithms (1/3)

- *jsteg*: Features ba1, ba2 and ba3 are motivated from artefacts observed after embedding and also ba4 file header motivated from *f5* can be used with sig1 in the file header to identify *jsteg* in a unique manner with best results. There are only a small number of errors at high capacity with the short key for ba2 and (58/30000) JFIF signature errors for ba1 in the Flickr30k tests
- *f5*: Feature ba4 with signature sig2 allows a unique identification of *f5* with only low (624/30000) errors in the Flickr30k tests

- In summary for signature sig2, it seems also to be naturally occurring in few JPEG samples in Flickr30k causing some classification errors: for ba1 of 0.18% (58/31783) and ba4 of 1.96% (624/31783)

Performance of the attributes regarding the different algorithms (2/3)

- *outguess*: Feature file header ba4 with signature sig4 is relevant for identification of *outguess* but it is not unique with errors in the high capacity embedding; it overlaps with the signature for *steghide* in the file header. For stego malware detection without the need of algorithm identification the sig4 usage is possible. There are no errors in the re-compression and Flickr30k tests.
- *steghide*: Feature file header ba4 with signature sig4 is relevant for identification of *steghide* but also outguess is attributed and therefore no unique algorithm identification is possible, but it allows with sig4 the general stego-malware detection. Only the re-compressed embedding has no errors. As for *outguess* there are also no errors in the re-compression and Flickr30k tests. The two blind content-based features nba1 and nba2 are motivated from *steghide* artefacts in JPEG files. Both features are relevant but do not allow algorithm individualization as also *jphide* causes similar artefacts in JPEG. Further it causes false positives in the re-compression and in the Flickr30k tests.

- In summary the ba4 sig4 allows stego-malware detection but no algorithm individualization.

Performance of the attributes regarding the different algorithms (3/3)

- *jphide*: ba4 with signature sig3 is relevant and unique for identification of *jphide* but the lack of signature sig3 (and the attribution based on this fact) also appears in the Flickr30k tests. The two non-blind content-based features nba1 and nba2 are also relevant but do not allow algorithm individualization as also *steghide* causes similar artefacts. Further it also causes (as for *steghide*) false positives in the re-compression and in the Flickr30k tests.

- In summary ba4 sig3 is unique for the algorithm *jphide*, but also occurs in non stego data in re-compression of Alaska2 and Flickr30k. Content based features are also relevant, but also occur during normal re-compression. Therefore, for these three features, it is difficult to used them for stego-malware detection.

Performance of the attributes

Summary

- The signature-based features ba2 and ba3 are capacity and key independent and perform best for *jsteg* algorithm identification with no false positives in re-compression as well as in the Flickr30k tests;
- Feature ba1 (JFIF Version) has a similar performance for *jsteg* with few errors of 0.18% in the Flickr30k tests;
- ba4 signatures allow algorithm identification with:
  - sig1 : *jsteg* – unique with no errors,
  - sig2 : *f5* – unique with sig2 but 1.96% errors in Flickr30k test,
  - sig3 : *jphide* – relevant but with 100% errors in Alaska2 re-compression and 98.04% errors in the Flickr30k test,
  - sig4 and sig5: *outguess* and *steghide* - relevant with errors depending on capacity and keys.
- The content-based features have high error rates when the images are re-compressed and are therefore not applicable if re-compression needs to be considered.

Paper summary and conclusion

- The set of light-weight attribution features used in this paper in an initial and also very simple evaluation with five existing algorithms shows a first positive tendency to potentially identify the stego algorithm used in a stego-malware scenario.
- The promising results motivate further work on attribution approaches, especially for a generalization for stego-malware detection and prevention scenarios.

- The interest in this field is caused by non-Kerckhoffs' setups, which are often found in combination with simple embedding techniques (like the ones practically evaluated in this paper, or even more trivial, like LSB embedding in pixel domain image formats such as PNG).
- Furthermore, the Warden can usually monitor all relevant communication channels as well as the potential cover objects available in the target domain.
- This last characteristic of such stego-malware scenarios also enables non-blind analysis and attribution methods which significantly simplify the detection and attribution tasks.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL

Advanced Multimedia and Security Lab

Thank you for your attention!