**Outguess detection visual analysis**

## This page is not finished yet

and my anylsis needs a lot more testing to be sure, if there are no squares on uniform background it does not necessarily mean that there is no outguess, it appears that if jpg is big enough and message hidden with outgues is short , background is not touched by outguess, as you can see on first jpg of 2014, outguess is in and background is fully uniform black

Page 6: probaly no outgues, i am pretty sure the re isn none, but further tests needed

Page 4: i havent even open it in PS, no time atm

## Outguess JPEG creation code review

Outguess works as follow:

*"For JPEG images, OutGuess preserves statistics based on frequency counts. As a result, statistical tests based on frequency counts are unable to detect the presence of steganographic content. Before embedding data into an image, OutGuess can determine the maximum message size that can be hidden while still being able to maintain statistics based on frequency counts."*

As can be noticed, Outguess creates a new JPEG statistcally equivalent to the one given as an input. When creating the new JPEG, the default compression is 75, and can be set using -p to a value between 75 and 100. Consequently, the compression level of an image may give a clue about the presence of outguess, but this is not enough to draw any conclusion yet.

Let's rephrase the original question from "Is there valid outguess data in this image ?" to "Can this image be an output of the Outguess program ?". The latter question is much more easier to answer and provides a clear answer about the possibility of valid Outguess data in a picture.

Outguess uses its own JPEG wrapper code around libjpeg. When looking for clues about which program was used to generate a given output, one has to look at the header construction. When looking at the outguess JPEG wrapper code for the writing of the output, this looks like that:

int

main(int argc, char \*\*argv)

{

/\* ... \*/

if (!doretrieve) {

/\* ... \*/

dsth->write(fout, image);

} else {

/\* ... \*/

}

/\* ... \*/

}

handler jpg\_handler = {

"jpg",

init\_JPEG\_handler,

read\_JPEG\_file,

write\_JPEG\_file,

bitmap\_from\_jpg,

bitmap\_to\_jpg,

preserve\_jpg

};

void

write\_JPEG\_file (FILE \*outfile, image \*image)

{

/\* ... \*/

jpeg\_set\_defaults(&cinfo);

/\* ... \*/

}

GLOBAL(void)

jpeg\_set\_defaults (j\_compress\_ptr cinfo)

{

/\* ... \*/

cinfo->JFIF\_major\_version = 1; /\* Default JFIF version = 1.01 \*/

cinfo->JFIF\_minor\_version = 1;

cinfo->density\_unit = 0; /\* Pixel size is unknown by default \*/

cinfo->X\_density = 1; /\* Pixel aspect ratio is square by default \*/

cinfo->Y\_density = 1;

/\* ... \*/

}

It has to be noticed that the APP0 marker will appear as JFIF 1.1 with a void density unit and X Y densities set to 1x1. This is interesting to notice since image processors in general tend to give useful values for these fields as they will be used for printing and later editing. The APP0 header will thus look like this:

ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00

Now lets look at the APP0 marker for images given to 2014 (the exercise for earlier images is left to the reader):

Image APP0 header Readable

Ouguess data

----------------------------------------------------------------------------------

Epiphany: APP0 missing (imgur) YES

Cover: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Liber Primus: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Chapter 1: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Runes page 1: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Status page: ff e0 00 10 4a 46 49 46 00 01 01 01 01 90 01 90 00 00 NO

Runes page 2: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Runes page 3: ff e0 00 10 4a 46 49 46 00 01 01 01 01 90 01 90 00 00 NO

New runes p1: ff e0 00 10 4a 46 49 46 00 01 01 01 01 90 01 90 00 00 NO

New runes p2: ff e0 00 10 4a 46 49 46 00 01 01 01 01 90 01 90 00 00 NO

New runes p3: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

New runes p4: ff e0 00 10 4a 46 49 46 00 01 01 01 01 90 01 90 00 00 NO

Portrait: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Last runes p1: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Last runes p2: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Last runes p3: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

Last runes p4: ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 YES

As can be seen, images with a density unit set to unknown and X Y densities set to 1x1 correspond to images with readable outguess data in them. From the previous paragraph we now that the images with these fields set to any other value cannot have been output from a vanilla version of Outguess. This can be fair to then say they do not contain any valid, encrypted or not, Outguess data in them.

As a bonus, it has to be noted that the 400x400 dpi we have for the other images is the default for GIMP.

## All cicada image files, jpgs and pngs from 2012, 2013 and 2014

(archiver’s note: the file in the below link can be found in the same directory)

<http://www.mediafire.com/download/ip1nih9otmqjno1/Cicada3301Pics2012until2014.7z>

Images are in 2012 2013 and 2014 maps.

Files are supposed to be originals, but i havent double chack every single one.

Txt files in maps explain few details.

In map 2014, page4.jpg is image with magic square. That image was released on onion 3 server-status page.

They gave us the last page when we unexpected got on the "server-status" page and that probably wasnt preplanned. We think that that is actually last page of the book.

## Table with data on all pics:

name resolution size number of Outguess Outguess Outguess

in bytes outguess chars. found bacground details

2012 509x503 29,279 535 yes yes not sure yet

everywhere 509x503 28,892 1328 yes yes not sure yet

2nd ver. 509x503 no source 1405 yes yes not sure yet

2013 571x577 34,490 1459 yes yes not sure yet

valete 503x509 31,004 1398 yes yes not sure yet

2014 547x557 41,731 1150 yes !NO! not sure yet

1033.jpg 1327x1427 342,361 1498 yes unconclusive

page 1 2400x3600 168,876 2899 yes !NO! not sure yet

page 2 2400x3600 175,159 2899 yes !NO! not sure yet

page 3 2400x3600 1,476,614 2899 yes !NO! not sure yet

page 4 2400x3600 336,353 X no !NO! not sure yet

(5x5)

page 5 2400x3600 892,759 31809 yes YES not sure yet

page 6 2400x3600 1,223,337 7524 nonASCI random !NO!

of random false positive

garbage

onion 4 jpgs:

page 7 2400x3600 400dpi no ourguess

page 8 2400x3600 400dpi no ourguess

page 9 2400x3600 1dpi outguess

page 10 2400x3600 400dpi no ourguess

Neded to make more tests to figure out when outguess starts to change backgtund pixels:

[03:25] <masso> at ~25-30% of max possible size you should be fine without background scrambling

[02:07] <masso> NiceLurk: ya, I did a lot of things with outguess. It decreases the rgb values of px by max 1 for R and B, 2 for G

P setting for outguess:

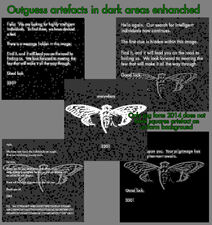
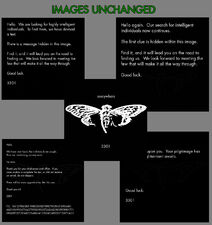
* -p <param> parameter passed to destination data handler
* it influences if outguess changes background or not, basically it is jpg compression if i understood correctly

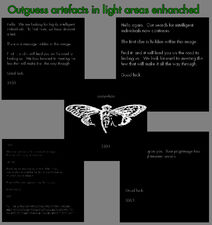
P setting for outguess

<andycactus> do you have a few minutes  
<andycactus> re image analysis of outguess and image backgrounds  
<andycactus> i saw the table you put up on wiki and the lack of background artefacts in this years images  
<Lurker69> yeah  
<Lurker69> not sure about that yet  
<Lurker69> but we asume that outguess if message is short enough doesnt  put artefacts on background  
<andycactus> did you realise that this happens if you put -p"100" the background effect goes away  
<Lurker69> but  i/we need to test that with example jpgs in which we will put outguee mssages of different sizes  
<andycactus> this is the quality setting for outguess  
<andycactus> for jpg compression  
<Lurker69> no havent try that yet  
<Lurker69> oh so on -p100 this doesnt happen  
<Lurker69> what is default outguess jpg compresion?  
<Lurker69> looks like cca 50  
<andycactus> if you look at the image headers, the quantize header tables are full of ones eg 1st image  
<Lurker69> judgign the jpgs  
<andycactus> default is 75  
<andycactus> when quantise tables are full of ones, outguess had a high eg -p"100" set  
<andycactus> so effect is the 'ditthering' of output gets put around say the text or image details  
<andycactus> in tiny 'squares'  
<Lurker69> yeah, so when big squaress happen?  
<andycactus> then in those images the quantisation tables are not full of ones  
<andycactus> so a lower n or default of 75 is used for -p"n"  
<Lurker69> ok so basicly it is possible that page 6 has outguess but cicada used -p"100" tag for it, although this is unlikely  
<Lurker69> i have to try to cretae same message as cicada did, one on twitter, then out in exact same text we got outt of it, and  observe filesize and dithering effects  
<andycactus> well I dont know. whenever I have tried my own tests even on white bg with -p"100" then the background around the letters is affected  
<andycactus> just around letters say  
<Lurker69> yeah, if you ahve effect only areoung letter it is very hard to determine if it is jpg or outguess  
<andycactus> jpegsnoop shows you the quantisation tables full of ones, most images this year  
<andycactus> except for one, page five which has the old style artefacts  
<Lurker69> <http://i.imgur.com/MYMoWUX.png>  here is massos experiment, here you can distingush outguess from jpg ditherind  
<andycactus> and I would say he used default quality on that one  
<andycactus> the quantisation table will not be full of ones  
<Lurker69> yeah he did, ithink  
<Lurker69> <http://prntscr.com/2kd1x6> here  also, you can notice alot more continuous vretical ines of pixels near vertical parts of runes  
<andycactus> those artefacts would go to the 'higher def' type if he used -p"100"  
<Lurker69> <http://static3.wikia.nocookie.net/__cb20140116131909/uncovering-cicada/images/d/d0/Page4artefactsinlight.jpg>  
<Lurker69> but here on page 4... you can sy for certaint  
<Lurker69> there arent many vertical lines, but there are few..  
<Lurker69> so maybe p100 and very short mesage?  or most probbybly nothing  
<andycactus> well that page 4 does look like a -p"100" type of effect rather than just jpeg artefacts  
<andycactus> and for some reason page five did not use -p"100"

## Cicada black jpg

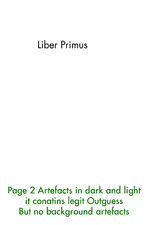
All of those conatin Legit Outguess, but one frno 2014 does not have background artefacts.



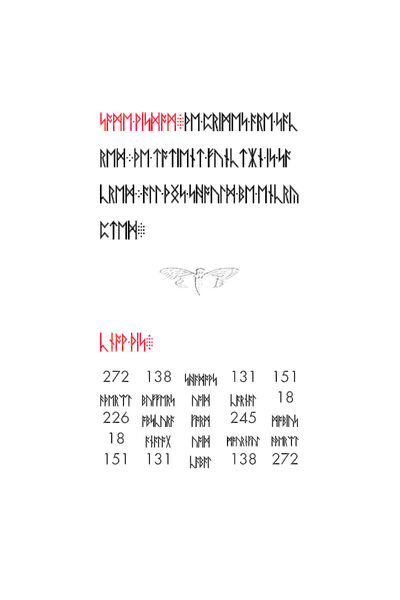
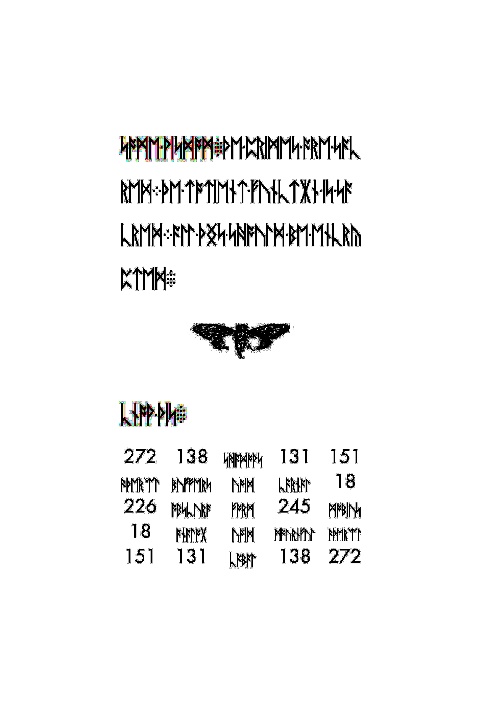
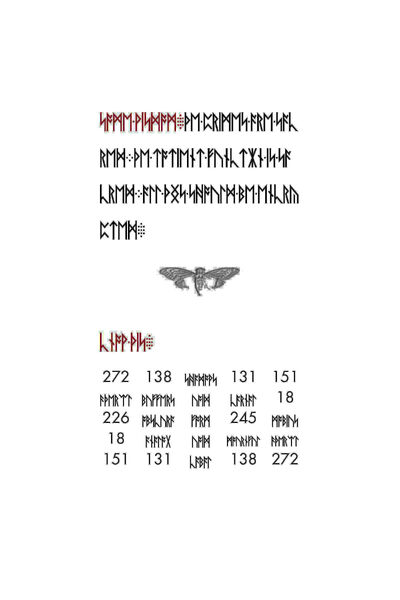


## Page 1, 2 and 3

All 3 of them contani Legit Outguess, but none of them had Background Artefacts. Strange that page 3 that is visually very similar to pages 5 and 6 does not have background artefacts.



## Page 4

We didnt find any Outguess in t yet, Image contains no background artefacts, but no sure if that means no outugess. Further tests needed to cocnlude anything

|  |  |  |
| --- | --- | --- |
| Page 4 Artefacts in Dark colors, so artefacts in letters. | Page 4 Artefacts in Light colors, so artefacts in background aruond text. | Artefacts on Light and Dark both shown on one picture. |

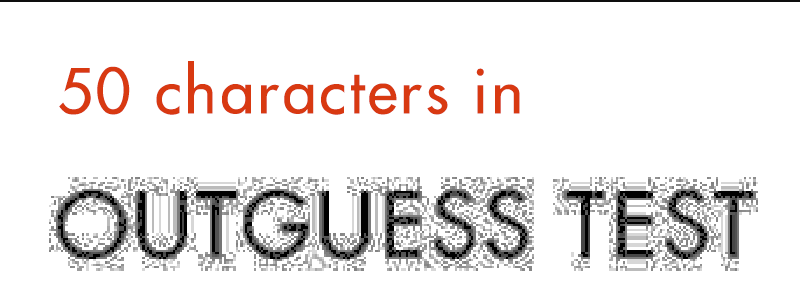
## Page 5 and 6

Half of each picture. Clearly shows that  Left (Page 6) Does not conatin same amount of dstenographicly outgueesed text as RIght one (Page 5), where we found Outguess message with 31809: short text, small jpg in hex, and PGP signature.

## Testing outguess by putting mesage in sample picture

I will make better explanation and more gifs some other day.

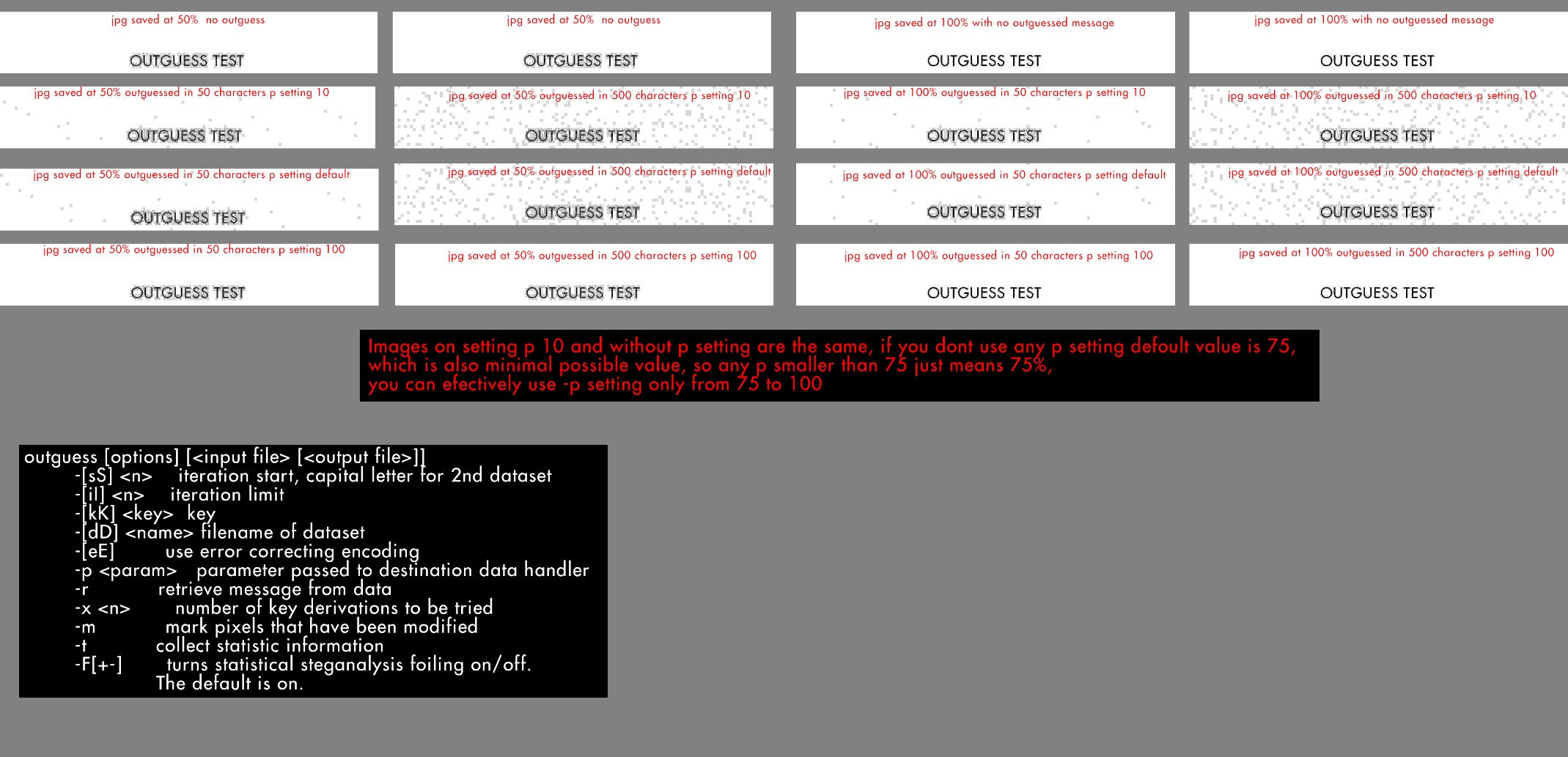
But most important things i figured out are. If you use setting -p100 while outguessing then outguess is visually undetectable. But so far we dont have any undeniable proof that cicada ever used anything than default -p 75

jpg saved at 50% with 50 hidden characters and you use default p=75% compression for outguess

This jpg shows different copmression rates and differnt lenght of messagess.

Mos notbale it that using -p 100  makes outguess images visually almost  imposible to distingush forom source jpg and precticly undetectable by this method.

Those images were mage by copying laybe two times in PS, then making one leyer levels up fully, to reveal dark artefacts; and other levels down fully to reveal light artefacts. thne turning visibility of top layer o 50% so both kind of atefacts are visible on same view.



## Some questions remain unanswered

Why initial jpg from 2014 doesnt have background artefacts like all previous white text on black backgrounfd cicada images is still ndetermined.

Why pages 2, 3, and 4 das no visual background artefacts is sitll undetermined.  We suspect lenght of message or using higher than default setting might  casued this, but firther tests  to replicate exact same situtaion are needed to  be sure.

## Soulseekah analysis:

We have theory hat jpg snop can detect outguess via fingerprint... but that needs to be tested:

<http://www.impulseadventure.com/photo/jpeg-snoop-identify-edited-photos.html>

[15:01] <soulseekah> <https://infotomb.com/kt5k8.txt> <- JPEG Analysis  
[15:02] <soulseekah> <https://infotomb.com/vmcfx.txt> <- fingerprints

<https://infotomb.com/kt5k8.txt>  
<https://infotomb.com/9yd96.txt>  
<https://infotomb.com/vmcfx.txt>  
<https://infotomb.com/8xopw.txt>  
<https://infotomb.com/y0mhn.txt>  
<https://infotomb.com/kcq8k.txt>

<https://infotomb.com/r39qe.txt> - imgur  
<https://infotomb.com/f3uyg.txt> - onion1  
<https://infotomb.com/9yd96.txt> - onion2 - liber primus  
<https://infotomb.com/g2f7j.txt> - onion2 - chapter 1  
<https://infotomb.com/cg5bf.txt> - onion2 - a warning  
<https://infotomb.com/nxbw8.txt> - onion3 - matrix  
<https://infotomb.com/3sfvq.txt> - onion3 - welcome welcome  
<https://infotomb.com/ana3x.txt> - onion3 - command yourself  
<https://infotomb.com/h9jl8.txt> - onion3 - onion4 encryption

## ppi of jpgs:

[14:29] <soulseekah> Can confirm, the second one is 400ppi  
[14:29] <soulseekah> vs first one is 74ppi  
[14:38] <masso> twitter 96  
[14:38] <masso> Blake 96  
[14:38] <masso> Liber 96  
[14:38] <masso> Chapter 96  
[14:38] <masso> Runes 96  
[14:38] <masso> Server status 72  
[14:38] <masso> 1st runes 96  
[14:38] <masso> 2nd runes 400  
[14:38] <masso> onion 400

Exif of all 4 pages from onion 4.

<http://regex.info/exif.cgi?dummy=on&imgurl=https%3A%2F%2Finfotomb.com%2Fqjiqh.jpg>  
<http://regex.info/exif.cgi?dummy=on&imgurl=https%3A%2F%2Finfotomb.com%2F3vw2u.jpg>  
<http://regex.info/exif.cgi?dummy=on&imgurl=https%3A%2F%2Finfotomb.com%2F609xy.jpg>  
<http://regex.info/exif.cgi?dummy=on&imgurl=https%3A%2F%2Finfotomb.com%2F8do0b.jpg>

First Second anf Fourth show Resolution    400 pixels/inch  
Third one that contained outguess shows Resolution 1 pixels/None, like all previous jpgs with outguess did

WE NEED TO ADD dpi resolution in table on top

## New stufff:

for now I'm 80% sure that we don't have outguess with password on page 6, but more tests needs to be done:

half legit proofs:

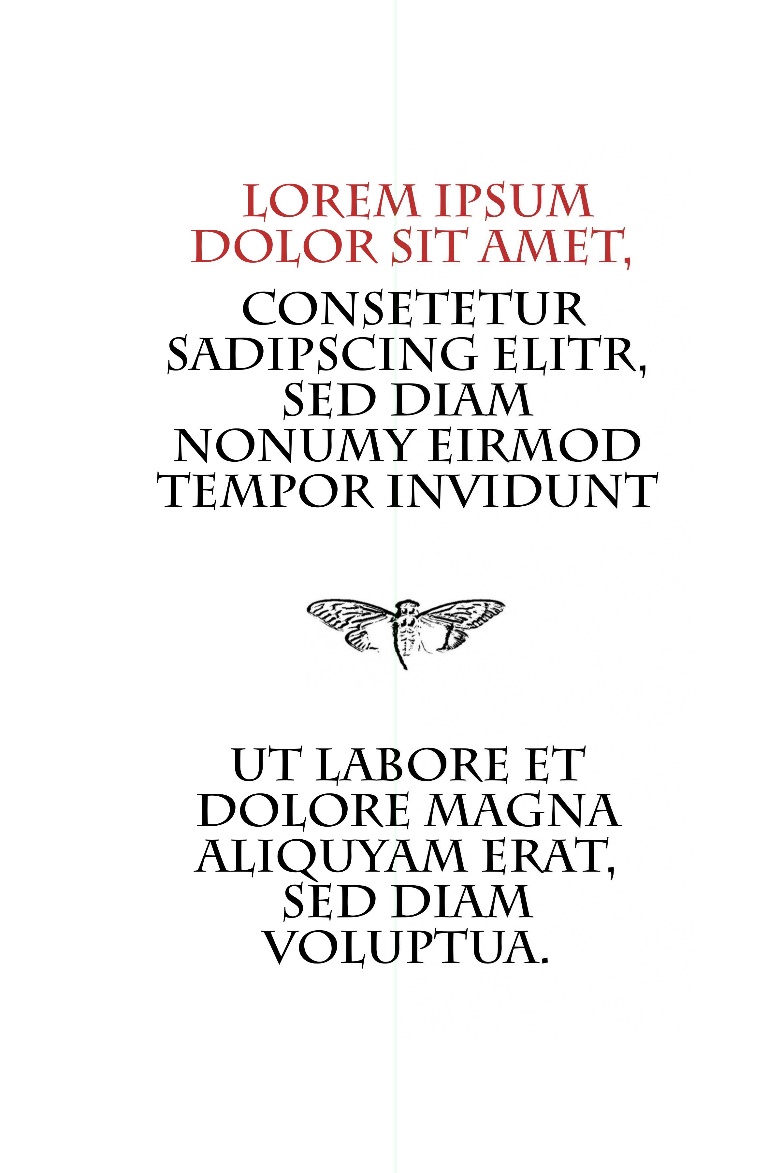
[18:45] <Lurker69> <http://imgur.com/BtHz9g7>

[03:20] <Lurker69> i am 90%  
[03:20] <Lurker69> or 95% sure that page 6 has NO outguess  
[03:21] <Lurker69> <http://imgur.com/a/pWUAW>  
[03:21] <Lurker69> but i need like 2 hours more  so probbabyl 3 days to finih it  
[03:22] <Lurker69> need to test exactly when outguess tarts to fuck with abckground  
[03:22] <Lurker69> sinc inital 2014 pic does not have square on background but it has outguess in it  
[03:23] <masso> depends on the amount of data to hide  
[03:25] <masso> at ~25-30% of max possible size you should be fine without background scrambling

## 

## ALL SORTS OF UNRELTATED STUFF AND MASSOS POSTS IN BETWEEN

[14:26] <durex> i use this for the images .. ive written it myself... so... <http://pastebin.com/7CDNECYx>  
[14:26] <TaiiwoBot> ^ [C++] #include <iostream>  #include <string.h>  #include <fstream>  #include <string>  - Pastebin.com ^  
[14:27] <essen\_> if you want a byte reverse tool, here is a quick and dirty c program: <http://pastebin.com/Auxu8czA>  
NiceLurk has changed topic for #CicadaWiki to: "Channel for people who want to help making Cicada 3301 Wikia page better:  [http://uncovering-cicada.wikia.com](https://uncovering-cicada.fandom.com/)   Main channel: #cicadasolers"  
NiceLurk has changed topic for #CicadaWiki to: "Channel for people who want to help making Cicada 3301 Wikia page better:  [http://uncovering-cicada.wikia.com](https://uncovering-cicada.fandom.com/)   Main channel: #cicadasolvers"  
<masso> <http://6666.99k.org/ogtest.jpg>  
[13:30] <masso> <http://prntscr.com/2jioe7>  
[14:23] <mlehmk> <http://pastebin.com/tW06q05q>  
[14:23] <TaiiwoBot> ^ [C#] Reverse - Pastebin.com ^  
[13:39] <Lurker69> uncovering-cicada.wikia.com/wiki/How\_to\_detect\_outguess\_in\_jpgs  
<Lurker69> "Observation: Gematria doesn't have a letter V in it. This is the only letter in the message without a value (the only other letter not defined in the Gematria is Q). If we take the latin convention that V = U  we get that 'divinity' = 376 => Line 3 = 1229."  
<glurk> Eve57 I promise you, if you take that file and reverse it, it will have a corrupt JPG.  Someone changed it to get "page 6"  
<masso> Lurker69: check my last link, put it in wikia or something for all the unbelievers... :)  
<NuclearPizzaCats> Lurker69: ok, thank you for the explanation !  
--> djs (47e42a74@gateway/web/freenode/ip.71.228.42.116) has joined #cicadasolvers  
<Lurker69> masso: why our picture looks like it ahs more jpg compression on top as towards bottom?  
<masso> Lurker69: but resave it somewhere please cause it won't be on my server for eternity  
<Lurker69> leeft and eight parts  
f[13:32] <masso> so, and here is the pic in complete, no correction at all, everybody can play around with contrast and things and see what outguess does. green line is cut, left side without, right with outguess  
[13:32] <masso> <http://imgur.com/okRX8bX>  
[13:30] <masso> Lurker69: Here is your pic, I created a pic in the same size as the matrix rune pic, text on it (~same % colour/white), right side has outguess, left side not, colour/contrast correction  
[13:30] <masso> <http://prntscr.com/2jioe7>



## MORE POTENTIAL INFORMATION

ye7eech3eex on IRC notied a couple of odd patterns in some of the images.  He was running this:

<https://www.paste.to/smbBUBXk>

and the results of his investigations are here:

<https://www.paste.to/TANc9xqn>

I saw a similar anomaly with a different technique - processing the binary bits of the JPG files of the six pages of the book we have and viewing them as a bitmap show odd patterns in page 1 and page 2:

echo 'P1' > p1.pbm && echo '4691 288' >> p1.pbm && xxd -b ../Page\ 1\_Liber\_primus\_OK.jpg | cut -c "10-63" >> p1.pbm

echo 'P1' > p2.pbm && echo '3274 428' >> p2.pbm && xxd -b ../Page2\_Chapter\_Intus\_OK.jpg | cut -c "10-63" >> p2.pbm

echo 'P1' > p3.pbm && echo '4421 2672' >> p3.pbm && xxd -b ../Page3\_Runes\_Warning.jpg | cut -c "10-63" >> p3.pbm

echo 'P1' > p4.pbm && echo '1641 1640' >> p4.pbm && xxd -b ../page4.jpg | cut -c 10-63 >> p4.pbm

echo 'P1' > p5.pbm && echo '2866 2492' >> p5.pbm && xxd -b ../page5.jpg | cut -c 10-63 >> p5.pbm

echo 'P1' > p6.pbm && echo '3468 2822' >> p6.pbm && xxd -b ../page6.jpg | cut -c 10-63 >> p6.pbm

Note: The sizes for the pbm's were just factors of the file size in bits of the JPG (e.g. 4691x288=1351008, which is 168876kb which is the size of the Page\ 1\_Liber\_primus\_OK.jpg file).  The anomaly is visible whatever dimensions you choose.  File sizes are:

168876 16 Jan 12:25 Page 1\_Liber\_primus\_OK.jpg

175159 16 Jan 12:26 Page2\_Chapter\_Intus\_OK.jpg

1476614 16 Jan 12:29 Page3\_Runes\_Warning.jpg

336353 15 Jan 08:29 page4.jpg

892759 15 Jan 12:03 page5.jpg

1223337 12 Jan 15:31 page6.jpg

Note2: the jpg files used are those downloaded from the .7z archive linked in the wiki.

## jonsmif

The pbm files for pages were loaded into GIMP and exported for upload to tinypic - take a look at the first two pages:

<http://oi39.tinypic.com/28b5tme.jpg> (version from ye7eech3eex: <http://www.anonimg.com/img/67019d8adac7dfe129450d7464d55657.jpg>)

<http://oi44.tinypic.com/2jev5vt.jpg> (version from ye7eech3eex: <http://www.anonimg.com/img/91814fa25b5c0952fc8c4926699c3a33.jpg>)

There is clearly something odd in the images.  Could this be an artefact of the outguess data stored in the files (both of these files have a hidden PGP signed bunch of hex), or could it be some other anomaly?

-jonsmif