**What Happened Part 1 (2014)**

*We (the authors) have chosen not to name any of the individuals responsible for specific solutions here, and instead refer to the solving community as a whole. Shoutouts to you few who actually solved the puzzle -- you know who you are.*

**The Beginning**

The Twitter Image

In early January 2014, people gathered together on the net to wait for the next signs of life from Cicada 3301. After several fake puzzles, eventually a genuine message from Cicada was received. On January 6th, the T[witter](https://twitter.com/1231507051321) account used by Cicada in 2013's puzzle was re-examined; after being inactive for about a year, it suddnely tweeted a link to an image on [imgur](http://i.imgur.com/zN4h51m.jpg).



In line with earlier rounds, this image contained a steganographically hidden message recoverable using the program [outguess](http://www.outguess.org/). Executing

outguess -r zN4h51m.jpg zN4h51m\_output

yields

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

The work of a private man

who wished to transcend,

He trusted himself,

to produce from within.

1:2:3:1

3:3:13:5

45:5:2:3

20:3:20:5

8:3:8:6

48:5:14:2

21:13:4:1

25:1:7:4

15:9:3:4

1:1:16:3

4:3:3:1

8:3:26:4

47:3:3:5

3

13:2:5:4

1:4:16:4

.

o

n

i

o

n

Good luck.

3301

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.11 (GNU/Linux)

iQIcBAEBAgAGBQJSyjguAAoJEBgfAeV6NQkPsgAP/A3tMC3lpyFNAc/sj+Izu15S

CzUjZJMe20Gu9UMNokQ2UJabktv9w0GMyK17TrMkUcU+ZpjdzGNqKoE2ETVxLmD/

uBZtR5PnF9EE3D08tJUPN1vSrYNkYk+9zcaUJZMPNgYNCt/CACutPwrOci9i9FDO

7BIpnhGqT3ZruqrSwO2Y73LJI1xxUt1XUqh1NQ+fJeAFMRkJBZZazkxRlgk3GGsF

fLrcEKrS+KBipV1EQaaKxjISc9hc2c1TfxE66evlkN+zLcoyDcYuyruNM5wiZzgM

2uR58c+xgWQgG5UuLFClfvjDxUvDkrKt4mzEeaYSUm1MsYueuYklz4ydlg5Mf6l2

p1WyAxO52XfXVUZASk6VmaEQ0WjODTXvLeFTxUSDoKDMkvxDVxX6wGkufS9JwakB

nTZizZ8Ypv8GcNCuNNGd6gZ1Vk2MYntggXdX8INd0Itcd3QnLqbBnATDOinDxlOs

5zTrtyTHNaxxDagPfAbU1jMXM0aHd7PFAzjjp7kgCTWqMyBch+8Vt80bjkdL9iw8

Q3hxuanq8mh6nUGc+tNe0UfqKHEbE+jWIezYqgawJB0M9R5OhxWE+E+jPXtZKkXQ

JHYndPDrrsV8q27b7p0KN0+oblTkjqsItIAuLu7FNd0B4xb1jjp1Sbh7WJdZ/rbi

mCO0vN/obU9qK1Vfapy0

=6Gxk

-----END PGP SIGNATURE-----

This message is signed with the [PGP key](http://pgp.mit.edu/pks/lookup?search=0x7A35090F&op=index) used by Cicada in past years, verifying its authenticity. This message marks the start of 2014's recruitment puzzle.

**The Book Cipher**

The message looked like a book cipher. Cicada has used book ciphers [in past years.](https://uncovering-cicada.fandom.com/wiki/What_Happened_Part_1_%282013%29#The_Law_.28Liber_AL_vel_Legis.29) Thanks to the hint in the beginning of the message the book was easy to find -- [Self-Reliance and Other Essays](http://www.math.dartmouth.edu/~doyle/docs/self/self.pdf) , by Ralph Waldo Emerson.

To illustrate how to solve this cipher, take the first line of the message:

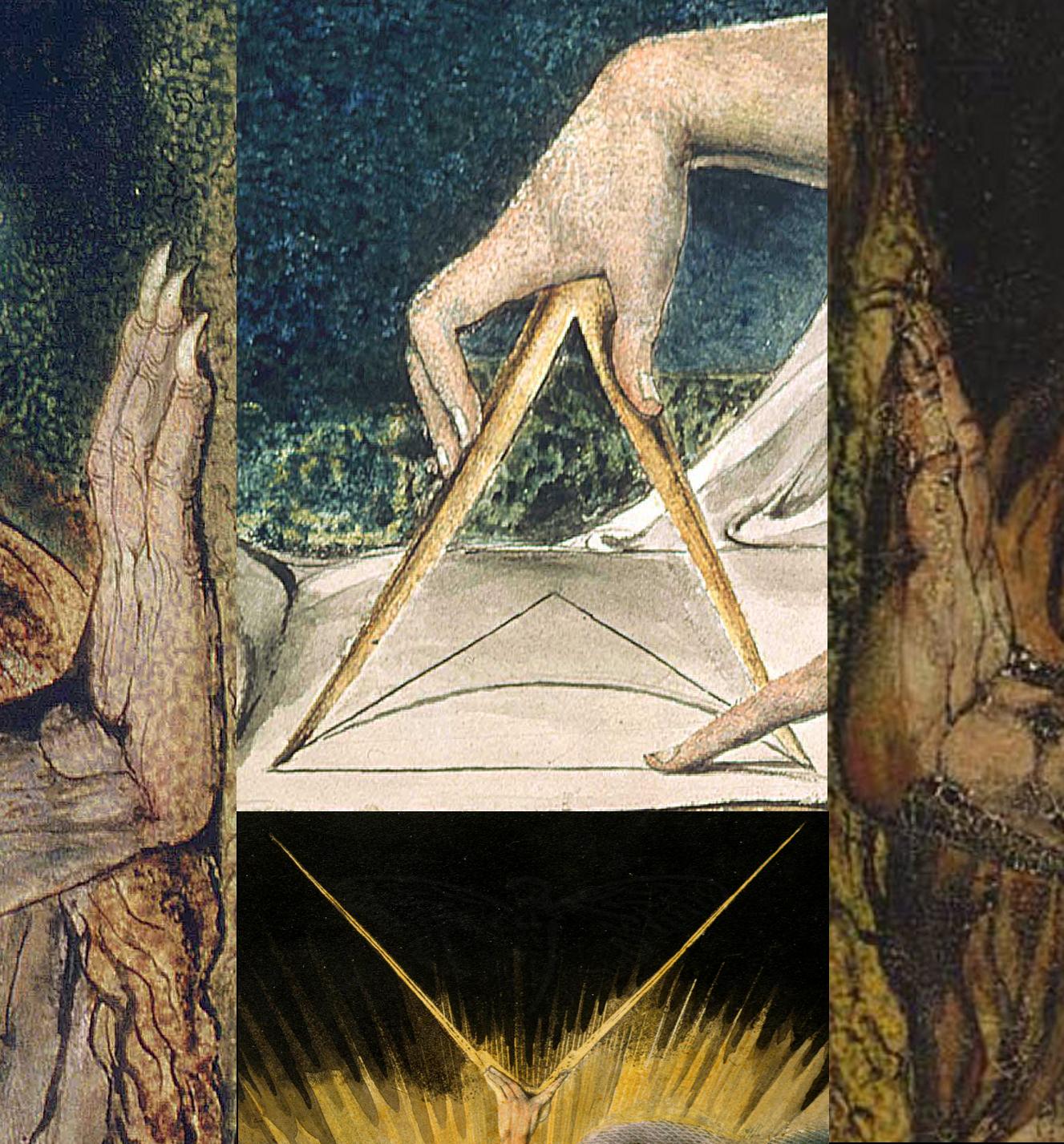
1:2:3:1

This line references the first paragaph of the text, the second sentence in that paragraph, the third word in that sentence and finally the first letter in that word. In this case the first character of the third word of the second sentence of the first paragraph in *Self-Reliance and Other Essays* is 'a'. Following this scheme for the rest of the cipher (the solitary 3 points to the third character in "Ralph Waldo Emerson", 'l') and appending ".onion" to the result yields

auqgnxjtvdbll3pv.onion

which is the address of a Tor hidden service. Visiting the service we found...

**The First Onion**



*This onion has since gone offline.*

**The William Blake Collage**

The hidden service featured [this image](https://uncovering-cicada.fandom.com/wiki/File:1033.jpg), which is a collage of four [paintings](https://uncovering-cicada.fandom.com/wiki/DEBATE_ABOUT_MEANING_OF_THE_PICTURE_ON_ONION) by [William Blake](https://en.wikipedia.org/wiki/William_Blake).

Following the usual procedure for investigating images, it was noticed that this image contains a message, again, extractable by outguess. The following is its content, omitting the PGP header and signature:

e = 65537

n = 75579125746085351644267182920580212556413102071876330957950694457000592\

10248050757270234679993673844203148013173091173786572116639

- -----BEGIN COMPRESSED RSA ENCRYPTED MESSAGE-----

Version: 1.99

Scheme: Crypt::RSA::ES::OAEP

eJwBswBM/zEwADE2MgBDeXBoZXJ0ZXh0LE2jxJS1EzMc80kOK+hra1GKnXgQKQgVitIy8NgA7kxn

2u8jNQDvlu0uymNNiu6XVCCn66axGH0IZ9w4Af3K/yRgjObsfA1Q7QqpXNALJ9FFPgYl5rh07cBP

M9kbSH6DynU/5cYgQod2KymjWcIvKx3FkjV4UOGakDnBf1eQp1uwvn3KxDVwTyzPqbMnZvOA06Ec

AfKtyz1hEK/UBXkeMeVrnV5SQQ==

=yTUshDMKN65aPaKAR0OU8g==

- -----END COMPRESSED RSA ENCRYPTED MESSAGE-----

The message can be seperated into two parts. The first part, spanning the first 3 rows, declares two values, an *N* and an *e*. The next part, hugged by "*BEGIN COMPRESSED RSA ENCRYPTED MESSAGE*", contains information about an encryption scheme, also called a chiffre, as well as data encoded in [base64](https://en.wikipedia.org/wiki/Base64). Base64 is a scheme to encode unprintable bytes into printable characters.

The *Scheme* line tells us that the following message is encrypted using the cipher RSA. The next step was clear: decrypt the message. To do that, we needed something we didn't have: the private key. A brief explanation is in order.

**A Brief Overview of RSA**

RSA is a moderately complex cipher to understand; [Numberphile](https://www.youtube.com/watch?v=M7kEpw1tn50&feature=youtu.be&t=54s) provides a good introduction to the topic. Its main advantage is that, as a [public key encryption](http://en.wikipedia.org/wiki/Public_key_encryption) scheme, it allows sharing of encryption keys without transmitting the key in plaintext or agreeing on a key in advance. Public key cryptography solves the problem of how two parties can communicate securely without a pre-existing secure channel of communication.

In RSA, *N* and *e* are variables commonly used in the mathematical aspects of the cipher. In fact, together they constitute the public key of an RSA keypair. The public and private keys are mathematically related. Without going too much into detail, they both are related to *N. N is* the product of two large primes, called *p* and *q*. If an attacker can factor *N,* which is publicly available, into its two prime factors *p* and *q*, then he can calculate the corresponding private key.

In reality, factoring large integers that have only two prime factors is a [computationally hard problem](http://en.wikipedia.org/wiki/RSA_problem) . In RSA, an *N* with 2048 or 4096 bits is typically used as the large size provides enough complexity to make the factorization of *p* and *q* computationally infeasible in the short term.

**Finding the Private Key (or, Brute Forcing RSA)**

To our luck, the *N* used to encrypt the message was far smaller than 2048 or 4096 bits. The *N*we were given was 432 bits (130 decimal digits) long.

The solving community exhausted a lot of options in an attempt to find the private key. They searched for suspicious information in the data provided until that point, investigated images, brainstormed correlations, followed connections and so forth. As time went on and nothing was discovered, the solvers began to discuss the worst case scenario: finding *p*and *q* via brute computational force.

After a while it was agreed that a parallelized approach would render the best results, as a breakthrough in finding the hidden data would not impede on factorizing the number and vice versa. So a small group of people banded together to think about the most effective way to share their computing power in order to factor *N*. It was decided that a distributed approach would be the only feasible option, since even a 432 bit key requires an enormous computational effort not suited for a single processor. However, distributing the workload could achieve results in way less time, and the community was eager to help. We quickly agreed on [cado-nfs](http://cado-nfs.gforge.inria.fr/), a program designed to obtain the prime factors of large integers using a distributed network of machines (convenient to say the least). After about 8 hours of debugging, fixing, patching and testing as well as additional 9 hours of distributively working on the prime, the results were in:

p = 97513779050322159297664671238670850085661086043266591739338007321

q = 77506098606928780021829964781695212837195959082370473820509360759

These individuals successfully brute forced a 432-bit RSA key. Make no mistake -- this is an incredible feat for a group of people spread across the world with consumer-grade hardware. At this point, the writer would like to take a moment to express their gratitude for the people participating in this effort. Without the people donating their time, efforts and resources, this would have taken a lot longer than originally anticipated. Also, a special thanks needs to go to the people who managed the servers, especially the one unnamed person who rented a server from Amazon to complete the last, locally computed phase of the calculation. With all words of thanks being said, let's continue.

**Putting The Private Key To Use**

Now that we have the information, we have to make something out of it. To make sure we do it right, it would be wise to simply do what cicada did to encrypt the message and reverse it. We also have a clue for this in the outguessed message. The line

Scheme: Crypt::RSA::ES::OAEP

contains a dead giveaway. The programming language [Perl](https://en.wikipedia.org/wiki/Perl) uses [double colons](http://perldoc.perl.org/perlmod.html#Packages) to achieve modularization. Therefore, we can safely conclude that a Perl program was used to encrypt this message. A decryption program in Perl using the found p and q values is available [here](https://uncovering-cicada.fandom.com/wiki/Program_to_decrypt_RSA_message).

**The Second Onion**

*This onion has since gone offline.*

**The Growing String**

After the successful decryption of the RSA message, we had a single resource:

cu343l33nqaekrnw.onion

This is the address of a Tor Hidden Service. Upon visiting the service, the following document was displayed:

<!--Patience is a virtue-->



The page appeared to be static. However, after some time, we noticed that the string was slowly growing. Every few minutes, two characters were appended to the end of the string. This process continued for approximately 23 hours. The time intervals between new bytes were found to be multiples of five. Various users recorded the minutes between updates, the time that they occurred, and the data appended at those times.

<http://pastebin.com/5bTLHqCN>

<http://imgur.com/lTRRxTT>

<http://pastebin.com/qn8jmPJr> (GMT +1)

<http://i.imgur.com/prAeqPS.png>

The above datasets are most likely not complete, and no guarantee of accuracy is made.

After 23 hours, the process stopped and no more characters/bytes were appended to the string. The final string was:



This string is 512 characters long. We assume it is hexadecimal.

However, about an hour after the string finished growing, at approximately 05:31:40 GMT the document changed. The old 512-character code was gone, along with the HTML comment. In its place was the following document:

<https://infotomb.com/oyfhl.txt>

backup: [sprunge](http://sprunge.us/VHRH)

We link to it because it is very large.

Note that the HTML comment was changed from

<!--Patience is a virtue-->

to

<!--761-->

This is significant. We noticed that, by applying the Gematrius Primus from the 2013 puzzle to the phrase 'Patience is a virtue' and summing the result, one obtains 761. The number is also a palindromic prime.

The new string was 3641299 (?) characters long. We noticed that it contained a significant amount of repeated text.

Analsysis of this [new string](https://infotomb.com/oyfhl.txt) revealed that it contained three JPEG image files. This was discovered when we:

1. Converted the string into its binary representation
2. Flipped all the bits (i.e. 0 becomes 1, 1 becomes 0) [Note: this is equivalent to XOR 111111]

Looking at that XOR'd binary string, we noticed that the first two bytes were

0xFF 0xD8

Wikipedia tells us that these are the first two bytes of a JPEG image. We walked through the data left to right, and later on in the string, we discovered the same byte sequence again. This indicated the presence of more than one JPEG image. We analysed the rest of the string in a similar fashion, and discovered a total of three JPEG images. The third JPEG image was in reverse order, and so had to be un-reversed.

For further clarification, the order was this (.... = JPEG data, [ ] = one complete image):

[0xFF 0xD8........................][0xFF 0xD8........................][........................0xD8 0xFF]

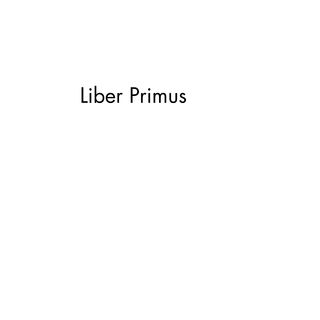
After transforming the bytes into their proper JPEG format (note that you must reverse the byte sequence of the third image before doing this):

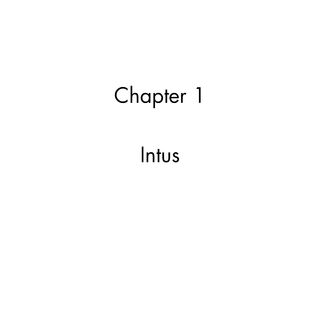
dd if=onioninvert.bin of=onion1.jpg bs=1 skip=0 count=168876

dd if=onioninvert.bin of=onion2.jpg bs=1 skip=168876 count=1476614

dd if=onioninvert.bin of=onion3rev.jpg bs=1 skip=1645490 count=175159

we obtained the following three images:

[](https://uncovering-cicada.fandom.com/wiki/File:Liber_primus.jpg)Liber Primus: Intus:

[](https://uncovering-cicada.fandom.com/wiki/File:Intus.jpg)

[](https://uncovering-cicada.fandom.com/wiki/File:Huh2.jpg)Runes:

**Image Analysis**

Each of the images contain hidden messages and other information. These were found and analyzed simultaneously. As a result, this section does not proceed in chronological order.

*Intus*

outguess -r intus.jpg out.txt

yields <https://infotomb.com/esd78.txt>

*Liber Primus*

outguess -r liber\_primus.jpg out.txt

yields <https://infotomb.com/hb0ba.txt>

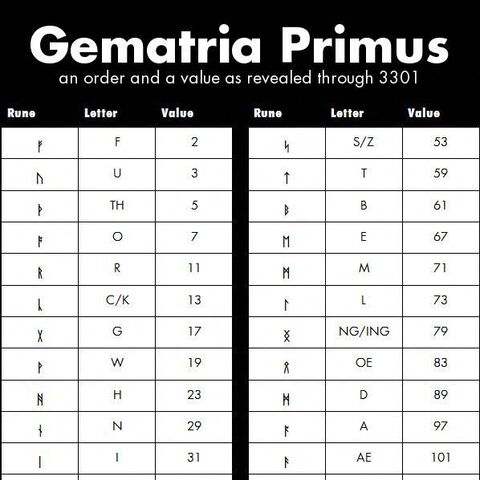
*Runes*

outguess -r runes.jpg out.txt

yields <https://infotomb.com/vmtyf.txt>

Here we break to explain something:

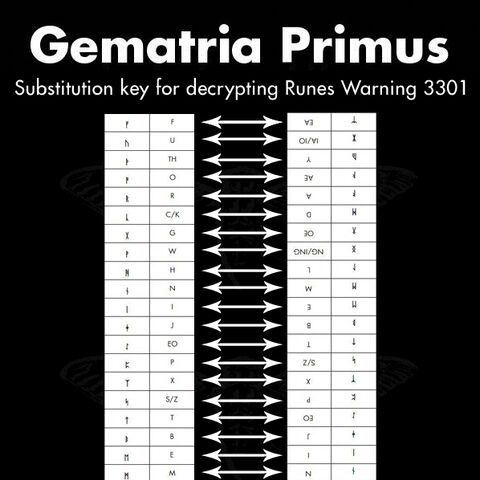
Below are two images. The first image is from 2013's puzzle and was obtained through use of XOR. It was used to solve portions of that year's puzzle. The second was created by puzzle solvers in 2014 and is derived from the first. Here we refer to 2013's image, which was released by Cicada 3301, as 'Gematria Primus 2013'. We refer to 2014's image as 'Gematria Primus 2014'. **It is paramount to understand that 2013's image was created by Cicada 3301, while 2014's was created by puzzle solvers and was created from rearranging 2013's image.**



Gematria Primus 2013

**2013 gematria primus that created by cicada3301:**

Gematria Primus 2014



2014's was created by puzzle solvers and was created from rearranging 2013's image

We now return to the puzzle. Using the Gematria Primus 2013, the runes in the 'Runes' image become:

R NGRA

JIHEIIAI MAEYW EAAAEN

YEP JAEAED IXDISEO NGLREO THAEIA

DMAENG EOAE JI EOAIAI EOIPEO YI D

MAENGHICOEI EAEMC THAEIAA EOAIAY IX

SIAEIMDI THAEIAA CFY CAE MAEEO ICEEO AE

A DLRWI YEP JAEAED AEA YI NICCROEI

DAEMEOREMIC NGEYEM IEYIA YI NGAE

ACC AEA YIEA MIANJIAC EAAEA RHH E

C CRDAIC

This stumped us for a short while. Then, someone created Gematria Primus 2014 by rearranging Gematria Primus 2013. From Gematria Primus 13, take the left block of three columns and stack it on top of the right block of three columns. Then, find the letter you wish to decode. Find the position of that letter in the vertical list, beginning at the top. Take that number and find the character that many characters into the list, from the bottom up. Applying this process to each letter/rune yields (newlines added for readability):

A WARNING

BELIEVE NOTHING FROM THIS BOOK

EXCEPT WHAT YOU KNOW TO BE TRUE

TEST THE KNOWLEDGE

FIND YOUR TRUTH

EXPERIENCE YOUR DEATH

DO NOT EDIT OR CHANGE THIS BOOK

OR THE MESSAGE CONTAINED WITHIN

EITHER THE WORDS OR THEIR NUMBERS

FOR ALL IS SACRED

By substituting each character for its respective value in Gematria Primus 2013, we noticed that the sum of each line adds up to a prime (emirps marked by \*):

A WARNING

BELIEVE NOTHING FROM THIS BOOK = 757\*

EXCEPT WHAT YOU KNOW TO BE TRUE = 1009\*

TEST THE KNOWLEDGE = 691

FIND YOUR TRUTH = 353\*

EXPERIENCE YOUR DEATH = 769\*

DO NOT EDIT OR CHANGE THIS BOOK = 911\*

OR THE MESSAGE CONTAINED WITHIN = 1051\*

EITHER THE WORDS OR THE NUMBERS = 859

FOR ALL IS SACRED = 677

**The 5 Gram Message**

Shortly after the warning in the previous section was found, it was discovered that by XORing the hexadecimal messages hidden within each of the three images the following message resulted:

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

IDGTK UMLOO ARWOE RTHIS UTETL HUTIA TSLLO

UIMNI TELNJ 7TFYV OIUAU SNOCO 5JI4M EODZZ

Good luck.

3301

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.11 (GNU/Linux)

iQIcBAEBAgAGBQJSy23PAAoJEBgfAeV6NQkPeJwP/0IoafJ1SbmhD+KNbL5I2EdH

jgPRnZNrKCyMpWFSIw1qs6ujuw6VnW/rfnOD+df4kpzoAwEFfZDcRnBVsvIzOJ31

Txj9jXD22ki/CNRY88NyIzW9fjKs+iOylsa7Tx+6PBb3ndoYNEwnQwLIq3K4S3kQ

tgMzE3LiVq2pQwqFNdN+zGqcq7POEs0GmnL1aNpqU+Wrba4gSfoWwQBWUDv3S/s8

vY0hEqhWNd76wphig6hH6OyIaX/t1eYfcsSYhzAE5oKKahGr1E7cX1GBpHCIr1WM

ZwNaGVArQAkyEzT++tmF01O9h218CiTUFoBM/Zxyra7vxI2UOYS/pLonuV+eXARY

YfPHaZZxfk3bUWXcxioRukFSY2+xNdPfuBIT8rcJqa1kPJOzeZVC/IcwHA2mmG4l

3ltiVcDnQrZgz6Im3/ugFg8bqW12qqZ6XizRP3EXm4EnyhpfKZnXKPLEOvPKCj6j

1kYCrLmGtTTPFx79fZfryGXQIEAmipRbjVS5sVbUCfgmqUagmdU6v9VI53n6+r0J

b2amxREA+2MflkEoVJUaLQJ1rKZLFFJ9J17zUaXKMllsDBWXJS4Mb54o2+8bkEcM

3cP+16XV9pf2wZBkJE0AwoXI4L8JEyjNZZcGSLy8BojlAupX3Fg9KKt71XXrm9FD

tuBhMYWo/TDz+4UzLB+I

=57tj

-----END PGP SIGNATURE-----

After some trial and error analysis, it was discovered that this ciphertext had been created using a simple column transposition cipher. By arranging the ciphertext into 14 columns like so:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
| I | D | G | T | K | U | M | L | O | O | A | R | W | O |
| E | R | T | H | I | S | U | T | E | T | L | H | U | T |
| I | A | T | S | L | L | O | U | I | M | N | I | T | E |
| L | N | J | 7 | T | F | Y | V | O | I | U | A | U | S |
| N | O | C | O | 5 | J | I | 4 | M | E | O | D | Z | Z |

And reordering the columns like so:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2** | **8** | **9** | **1** | **12** | **13** | **11** | **4** | **5** | **7** | **3** | **0** | **6** | **10** |
| G | O | O | D | W | O | R | K | U | L | T | I | M | A |
| T | E | T | R | U | T | H | I | S | T | H | E | U | L |
| T | I | M | A | T | E | I | L | L | U | S | I | O | N |
| J | O | I | N | U | S | A | T | F | V | 7 | L | Y | U |
| C | M | E | O | Z | Z | D | 5 | J | 4 | O | N | I | O |

A message is obtained:

GOOD WORK

ULTIMATE TRUTH IS THE ULTIMATE ILLUSION

JOIN US AT FV7LYUCMEOZZD5J4ONIO

Assuming that the final N was omitted in order to fit the bounds of the column transposition cipher, and applying proper formatting, the following Tor hidden service address is obtained:

fv7lyucmeozzd5j4.onion

And off we went.

**The Third Onion**

*This onion has since been taken offline.* *In this section the puzzle fractures into several directions. The author has chosen not to subdivide this section and instead opts for pure chronological formatting. Events in this section are detailed in the exact order they occurred.*

The first visitor to this hidden service was greeted with a blank page. The page remained blank for a short time and then changed to the following document:

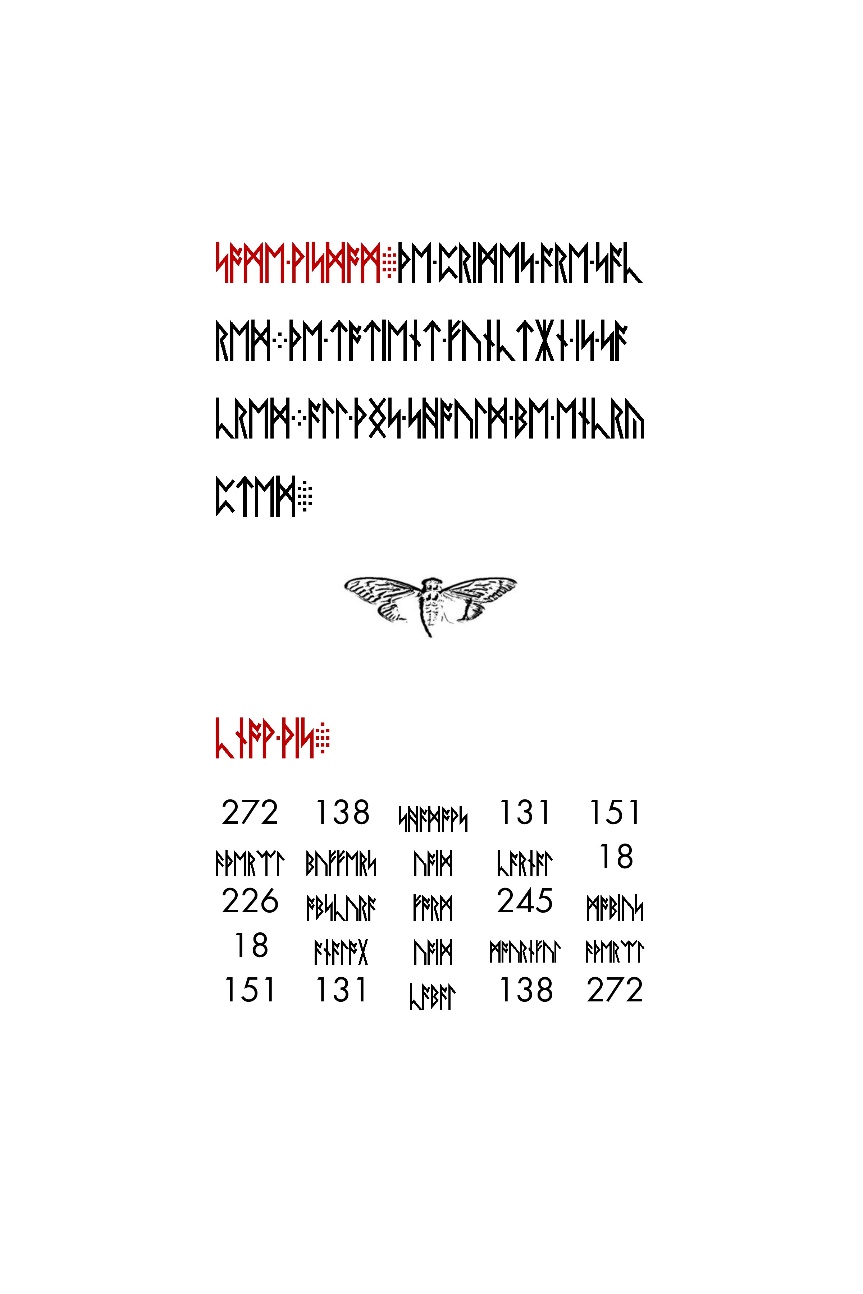
<!--1033-->

87de5b7fa2

As with the RSA onion, the string slowly grew over time, with two new characters (one byte) being added at widely varying intervals. Some timing data was collected for further analysis.

In the meantime, a solver ran the tool [DirBuster](https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project) against the hidden service and discovered an apparent misconfiguration of the backend Apache server. The server was leaking a system status page. Whether this was intentional is unknown. However, shortly after the page was discovered, it appears that Cicada was alerted that the status page had been discovered because the content of that page changed. Appended to the end of the server status was yet another very long string. This string was found to contain two image files in a similar ordering as the RSA onion, except that there was some data between them (OOB or Out Of Bounds data):

[0xFF 0xD8..............................] [Data in between JPGs] [..............................0xD8 0xFF]

After building the first JPG from the hex:

xxd -p -r < server-status.hex > server-status.jpg

One obtains the image shown below on the left. Doing the same for the reversed copy of the second JPG yields the same image as the first, shown -- except for that OOB data.

e

Comparing the first and second images

cmp -l server-status.jpg rev.server-status.jpg

one obtains the OOB data:

a02373230202020202833313020202020213433302020202021333130202020202135313a06363

330202020202939313020202020203331302020202020323330202020202028313a06323230202

020202534323020202020202139302020202025343230202020202632323a08313020202020203

2333020202020203331302020202029393130202020202636333a0135313020202020213331302

02020202134333020202020283331302020202022373230a0a

Note that all of these bytes are within the printable range of ASCII characters, and many of them appear to be ASCII for digits (e.g. 0x30, 0x39).

Converting this string to binary:

xxd -b oob.hex oob.bin

and reversing that::

xxd -r oob.bin oob-rev.bin

we obtain:

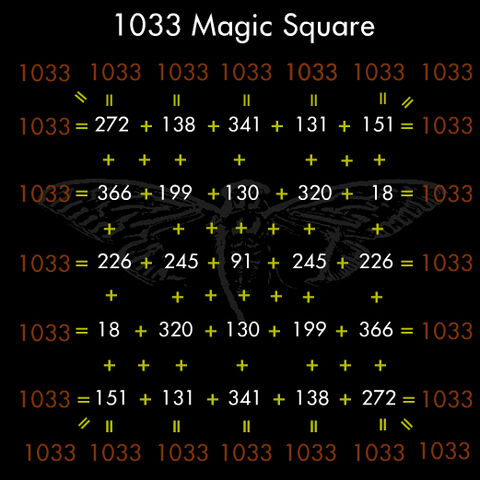
272 138 341 131 151

366 199 130 320 18

226 245 91 245 226

18 320 130 199 366

151 131 341 138 272

[](https://uncovering-cicada.fandom.com/wiki/File:Magicsquare.jpg)Among other things, this is a magic square whose magic number is -- you guessed it -- 1033.

For those following along, here's a nice one-liner to get that matrix from the original image:

dd if=server-status.jpg bs=1 skip=$((0x00521e4)) count=357 status=noxfer | rev | xxd -p -r

Runes were unencrypted. With matrix and Gematria Primus 2013 in hand, we began to interpret the runes in the image. The runes in the upper half of the image read:

SOME WISDOM

THE PRIMES ARE SACRED

THE TOTIENT FUNCTION IS SACRED

ALL THINGS SHOULD BE ENCRYPTED

In the bottom half of the image, the red runes read:

KNOW THIS:

The table underneath, translated to value form using Gematria Primus 2013, yields:

272 138 shadows 131 151

aethereal buffers void carnal 18

226 obscura form 245 mobius

18 analog void mournful aethereal

151 131 cabal 138 272

272 138 341 131 151

366 199 130 320 18

226 245 91 245 226

18 320 130 199 366

151 131 341 138 272

which is the exact same matrix found earlier from the OOB data.

About a day went by with little activity other than speculation on the meaning of this matrix. Then the string from the main page stopped growing. According the the HTTP header, the final update occurred on January 11 at 01:09:01 GMT. The final document was:

<!--1033-->



The final string was 512 characters (assuming hex, 256 bytes or equivalently 2048 bits). It was observed that this string matched the growing string from the previous onion.

On January 11 at 10:07 UTC, the Apache server status page changed once again to display a new hexadecimal string:

<https://infotomb.com/laqs9.txt>

This string contained the following two images in the same style as before

[0xFF 0xD8...............] [...............0xD8 0xFF]





**First Image**

Outguessing the first image yields a signed message:

<https://infotomb.com/t5uuz.txt>

The hex string from that message encodes a JPG image:

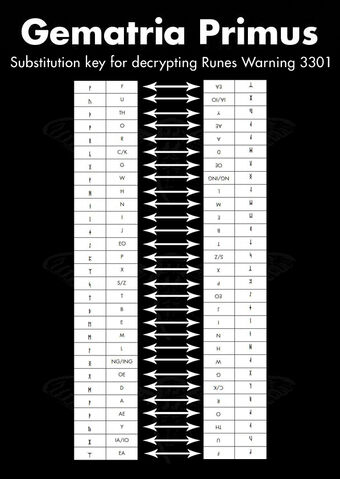
[](https://uncovering-cicada.fandom.com/wiki/File:OutguessfromLiberPrimusPage6.jpg)

The runes in this image were solved later (read on).

Translating the runes from the first image with Gematria Primus yields :

uWGsSfc rSugpWW fwxtclW ym WS tcnF GmXXmmw FpdGXr oW Xmi ff euG SuF yp rF ipF cF Fnw bxmd rXi fpc SSFc rTp fjmo ScwX bFw bWls ry jF r mcTSFtcpw mgS cGpu Sc rew Xpi bybx flir rm cgb yr cfu TpXjwtfW jgb FTffWpT pfax jmepGsosm wSjl wxuT FwmT dyjc sXxoGrmbw rmi dyjc xhuypl jGymfjpSuX wSj dwxu alasmXSx Fm Xmi py Fmuf

After some time with trial and error, it was discovered that this ciphertext had been created with a Vignere cipher.

[](https://vignette.wikia.nocookie.net/uncovering-cicada/images/6/66/Runes-warning.jpg/revision/latest?cb=20140108163159)

The key was DIUINITY (divinity) and used the reversed gematria as shown here

Every time you reached an F it was left unencrypted, and it reset the running key.

Reversing the cipher yields:

WELCOME:

WELCOME, PILGRIM TO THE GREAT JOURNEY

TOWARD THE END OF ALL THINGS.

IT IS NOT AN EASY TRIP, BUT FOR THOSE WHO

FIND THEIR WAY HERE IT IS A NECESSARY ONE.

ALONG THE WAY YOU WILL FIND AN END TO ALL

STRUGGLE AND SUFFERING, YOUR INNOCENCE, YOUR

ILLUSIONS, YOUR CERTAINTY, AND YOUR REALITY.

ULTIMATELY, YOU WILL DISCOVER AN END TO SELF.

**Second Image**

Outguessing the second image yields garbage output.

The runes on it translate to:

my yS Fxrjse ewn djusxytetm Sry ds neFdX pbunWGjXF jgb pTx pnwwilmF lpbuoWX rXWf rrSjm rmi dyj hlfu juXlTW SjoSrrm umsc WS liFFcl wi lt peup WXpTtb tme ulole Sjp uW lcg WgsXtm

bmrTfp wrj rxc G jWQ je ym dyjcFXuf pfa ccW r ujr ambp gpbunWGf nxe ygiWGumtcgWW jF bpwd fyx Fuf Sjp xlTWa lT cyuX

ce lFSixTsFhF Tyflcer pfax rbe Fcbf

After reversing the Vignere cipher:

IT IS THROUGH THIS PILGRIMAGE THAT WE SHAPE

OURSELVES AND OUR REALITIES.

JOURNEY DEEP WITHIN AND YOU WILL ARRIVE OUTSIDE.

LIKE THE INSTAR, IT IS ONLY THROUGH GOING

WITHIN THAT WE MAY EMERGE:

WISDOM:

YOU ARE A BEING UNTO YOURSELF.

YOU ARE A LAW UNTO YOURSELF.

EACH INTELLIGENCE IS HOLY.

FOR ALL THAT LIVES IS HOLY.

The red footer of that page reads

:AN INSTRUCTION: COMMAND YOUR OWN SELF :

**Back to the server page:**

Fourteen minutes after the status page update containing the two above images, on January 11 at 10:22 UTC the status page changed once again:

<https://infotomb.com/hw0l5.txt>

The first image from the previous update remained intact in the new string. However, the data of the second was replaced almost entirely with different hexademical. Attempting to render it as a JPG yields a corrupt and incomplete image.

**Outguessed Image Solved**

The runes in the small outguessed image obtained from the first new page were also encoded with a Vignere cipher. Using the key

welcome pilgrim to the

we obtained the Vignere offsets

22, 11, 9, 24, 26, 10, 11, 16, 19, 9, 23, 25, 19, 10, 13, 26, 27, 11

and using these offsets to decode the original runetext we obtained

A U O W Y F X L 5 L C S F J 3 N O N IA N

which formatted as a hidden service address yields

avowyfgl5lkzfj3n.onion

[](https://uncovering-cicada.fandom.com/wiki/File:Onion_4_adress_runes.jpg)

Small image cleartext illustration

**The Fourth Onion**

*This onion has since been taken offline.*

Upon visiting the fourth onion we were greeted with the following document:

</head><body><!--3301-->



<hr>

<address>Apache Server at 127.0.0.1 Port 5243</address>

</body>

</html>

This string is 512 characters (256 bytes). Shortly after the discovery of this string the onion went offline...

...until January 29 at 00:05 GMT, when it came back online. This time it contained no HTML. The sole content of the page was a signed message. The content of the message was a hex string (with newlines):

<https://infotomb.com/vnq3e.txt>

This string began with the bytes

0x1F 0x8B

indicating the string was a gzip file. Converting to binary

xxd -p -r onion4.hex onion4.gz

using file to confirm type

$ file onion4.gz

onion4.gz: gzip compressed data, was "data.out", from Unix, last modified: Fri Jan 24 15:10:12 2014

and unzipping

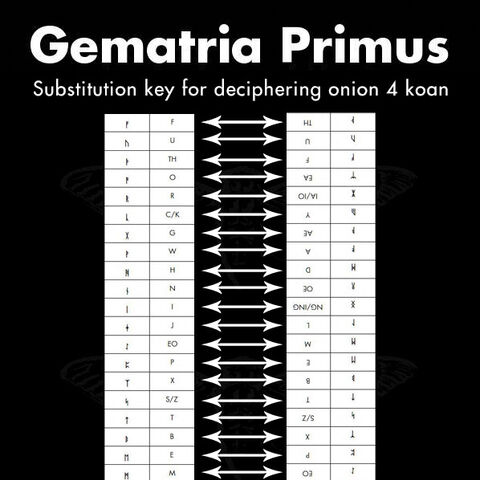
gzip -d onion4.gz

we are left with binary data containing four images:

[](https://uncovering-cicada.fandom.com/wiki/File:Onion-0.jpg)[](https://uncovering-cicada.fandom.com/wiki/File:Onion-1.jpg)

[](https://uncovering-cicada.fandom.com/wiki/File:Onion-2.jpg)[](https://uncovering-cicada.fandom.com/wiki/File:Onion-3.jpg)

Using a similar substitution to the one used in Onion 2

[](https://uncovering-cicada.fandom.com/wiki/File:SubstitutionKeyKoan.jpg)

and applying that to the runes in each image, we obtain a koan.

A KOAN

A MAN DECIDED TO GO AND STUDY WITH A MASTER.

HE WENT TO THE DOOR OF THE MASTER

"WHO ARE YOU WHO WISHES TO STUDY HERE?"

ASKED THE MASTER.

THE STUDENT TOLD THE MASTER HIS NAME.

"THAT IS NOT WHO YOU ARE THAT IS ONLY WHAT YOU ARE CALLED.

WHO ARE YOU WHO WISHES TO STUDY HERE?" HE ASKED

AGAIN.

THE MAN THOUGHT FOR A MOMENT, AND REPLIED

"I AM A PROFESSOR."

"THAT IS WHAT YOU DO, NOT WHO YOU ARE"

REPLIED THE MASTER. "WHO ARE

YOU WHO WISHES TO STUDY HERE?"

CONFUSED, THE MAN THOUGHT SOME MORE.

FINALLY, HE ANSWERED, "I AM A HUMAN BEING."

"THAT IS ONLY YOUR SPECIES, NOT WHO YOU ARE.

WHO ARE YOU WHO WISHES TO STUDY HERE?"

ASKED THE MASTER AGAIN.

AFTER A MOMENT OF THOUGHT, THE PROFESSOR REPLIED

"I AM A CONSCIOUSNESS INHABITING AN ARBITRARY BODY."

"THAT IS MERELY WHAT YOU ARE NOT WHO YOU ARE"

WHO ARE YOU WHO WISHES TO STUDY HERE?"

THE MAN WAS GETTING IRRITATED. "I AM," HE STARTED,

BUT HE COULD NOT THINK OF ANYTHING ELSE TO SAY,

SO HE TRAILED OFF. AFTER A LONG PAUSE THE MASTER REPLIED

"THEN YOU ARE WELCOME TO COME STUDY."

AN INSTRUCTION

DO FOUR UNREASONABLE THINGS EACH DAY.

The following image

[](https://uncovering-cicada.fandom.com/wiki/File:Onion-1.jpg)

when run through outguess

outguess -r onion4image3.jpg out

contains a message

For those who have fallen behind:

TL BE IE OV UT HT RE ID TS EO ST PO SO YR

SL BT II IY T4 DG UQ IM NU 44 2I 15 33 9M

Good luck.

3301

This ciphertext was found to be encoded with a columnar transposition cipher. Period = 7, key = 1736254.

Cleartext:

TOBELIEVETRUTHISTODESTROYPOSSIBILITYQ4UTGDI2N4M4UIM59133

Which of course translates to

TO BELIEVE TRUTH IS TO DESTROY POSSIBILITY Q4UTGDI2N4M4UIM59133

Formatting that last bit as an onion url yields

q4utgdi2n4m4uim.onion Purpose of 9133 at the end of the string [was not discovered](https://uncovering-cicada.fandom.com/wiki/CICADA_3301_2014_PUZZLE_FACTS_PART_6#Outguess_and_a_Transposition_Cipher).

bringing us to

**The Fifth Onion**

The fifth onion contained the following (signed) message

<https://infotomb.com/ooxyo.txt> Interconnectedness.mp3 can be found [here](https://github.com/rtkd/idkfa/tree/master/assets/mp3).

Shortly afterwards it went offline.

Converting the hex string to binary

xxd -p -r onion5.hex onion5.bin

and checking with file

file onion5.bin

tells us that it is an mp3 file. Renaming as such

mv onion5.bin onion5.mp3

and reading the ID3 tags

id3v2 -l onion5.mp3

shows

id3v2 tag info for his5u.mp3:

TIT2 (Title/songname/content description): Interconnectedness

TPE1 (Lead performer(s)/Soloist(s)): 3301

his5u.mp3: No ID3v1 tag

Thus the name of the song is 'Interconnectedness', and the artist is '3301'.

Translating each character of the word 'Interconnectedness' to its corresponding numerical value with the Gematria Primus 2013 and summing yields 772 (a prime). The song itself is 277.133 seconds long.

On January 31, Onion 5 came back online. This time it contained an image

[](https://uncovering-cicada.fandom.com/wiki/File:Brotherboxportrait.jpg)

[](https://uncovering-cicada.fandom.com/wiki/File:Img_diff2.jpg)

Difference between original portarit and one from onion

This painting is *Portrait of Andrés del Peral* by Goya y Lucientes. The significance of this is indeterminate.

A subimposed image of a man can be seen in the upper right corner. Adjusting image filters reveals hidden information:

*--Image temporarily ommitted due to wikia server maintenance--*

The man in the upper right is supposed to be Grigori Rasputin. The significance of this is indeterminate. The two columns of numbers are (left to right)

181

7

15

16

966

456

351

7

which sums to 1033, and

966

1071

626

204

434

which sums to 3301. The significance of these numbers is indeterminate.

Applying outguess to the original portrait

outguess -r onion5portrait.jpg onion5portrait.outguess

and checking the output with file

file onion5portrait.outguess

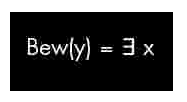
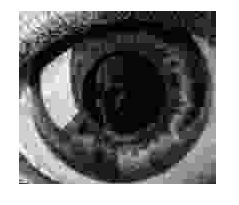
indicates that the retrieved data is a bzip compressed file. Decompressing

bzip2 -d onion5portrait.outguess

gives us a text file with a signed message.

<https://infotomb.com/772hf.txt>

Breaking up the three hex blocks in the message and using xxd to convert to binary yields two JPG images and an MP3 file.



The equation in the first image is part of the **Godel** incompleteness theorem. The mp3 file is a segment of **Bach'**s Trio Sonata in G Major (BWV 1039). ([Some say](http://www.clevcode.org/cicada-3301/) that it is BWV 1033) The picture of the eye is a painting by M.C. **Escher** called 'Eye', painted in 1946.

The last bit of the signed message

3PI:6:1:3

LML:1:1:1

3

ETOATS:19:9:1

...AF:5:3:1

AMO:13:10:1

CC:8:6:1

CBIA:3:7:2

CFAF:5:23:6

SPR:1:8:1

7

C[1]:4:5:3

AWDV:6:2:1

C[2]:2:17:5

SC:3:17:1

AOGS:2:8:1

ONION

is a book code. The key is ***Gödel, Escher, Bach****: An Eternal Golden Braid* by Douglas Hofstadter. The format of the code is

<chapter>:<line>:<word>:<letter>

Applying the code to the book:

3PI:6:1:3 Three-Part Invention 29 (u)

LML:1:1:1 Little Harmonic Labyrinth 103 (t)

3 (3)

ETOATS:19:9:1 Edifying Thoughts of a Tobacco Smoker 480 (q)

...AF:5:3:1 ... Ant Fugue 311 (t)

AMO:13:10:1 Introduction: A Musico-Logical Offering 3 (z)

CC:8:6:1 Crab Canon 199 (b)

CBIA:3:7:2 Canon by Intervallic Augmentation 153 (r)

CFAF:5:23:6 Chromatic Fantasy, And Feud 177 (v)

SPR:1:8:1 Six-Part Ricercar 720 (s)

7 (7)

C[1]:4:5:3 Contracrostipunctus 75 (d)

AWDV:6:2:1 Aria with Diverse Variations 391 (t)

C[2]:2:17:5 Contrafactus 633 (v)

SC:3:17:1 Sloth Canon 681 (z)

AOGS:2:8:1 Air on G's String 431 (p)

ONION

yields

ut3qtzbrvs7dtvzpONION

and formatting correctly yields

ut3qtzbrvs7dtvzp.onion

bringing us to

**The Sixth Onion**

ut3qtzbrvs7dtvzp.onion

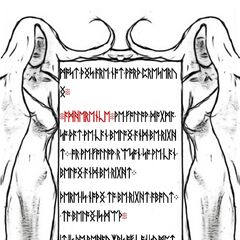
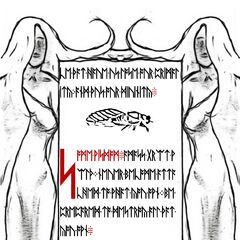
The first thing found at this onion address was a large block of hex after the HTML comment

The hex was

<https://infotomb.com/hrz8z.txt>

It contained four JPEG images in sequential, non-reversed order. They are

[](https://uncovering-cicada.fandom.com/wiki/File:Pg1.jpg)[](https://uncovering-cicada.fandom.com/wiki/File:Pg2.jpg)

[](https://uncovering-cicada.fandom.com/wiki/File:Pg3.jpg)[](https://uncovering-cicada.fandom.com/wiki/File:Pg4.jpg)

The runes were not enciphered, and they read:

Page 1:

THE LOSS OF DIVINITY: THE CIRCU

MFERENCE PRACTICES THRE

E BEHAVIORS WHICH CAUSE TH

E LOSS OF DIVINITY.

CONSUMPTION: WE CONSUME TOO

MUCH BECAUSE WE BELEIVE THE

FOLLWING TWO ERRORS WITHIN THE DEC

EPTION.

1 WE DO NOT HAVE ENOUGH

OR THERE IS NOT ENOUGH

Page 2

2 WE HAVE WHAT WE HAVE N

OW BY LUCK, AND WE WILL NOT

BE STRONG ENOUGH LATER T

O OBTAIN WHAT WE NEED.

MOST THINGS ARE NOT WORTH CONSUM

ING:

PRESERVATION: WE PRESERVE

THINGS BECAUSE WE BELIEVE WE AR

E WEAK. IF WE LOSE THEM WE WILL NO

T BE STRONG ENOUGH TO GAIN THEM

AGAIN. THIS IS THE DECEPTION.

Page 3

MOST THINGS ARE NOT WORTH PRESERV

ING:

ADHERENCE: WE FOLLOW DOGMA

SO THAT WE CAN BELONG AND BE RIGH

T. OR WE FOLLOW REASON SO WE CAN

BELONG AND BE RIGHT.

THERE IS NOTHING TO BE RIGHT ABOUT.

TO BELONG IS DEATH.

IT IS THE BEHAVIORS OF CONSUMPT

ION, PRESERVATION, AND ADHEREN

Page 4

CE THAT HAVE US LOSE OUR PRIMAL

ITY AND THUS OUR DIVINITY:

SOME WISDOM: AMASS GREAT W

EALTH. NEVER BECOME ATTA

CHED TO WHAT YOU OWN. BE

PREPARED TO DESTROY ALL THAT

YOU OWN:

AN INSTRUCTION: PROGRAM YOU

R MIND. PROGRAM REALITY

Each of these images contained a PGP signed outguess message. They're all essentially the same, following the form of

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Create one Tor hidden service that can accept CGI file uploads.

When this hidden service returns and can accept input, post the

three magic squares and the URL to your Tor hidden service here.

Work alone.

3333333333333333

310 12 103

3 3

312 14 123

3 3

310 12 103

3333333333333333

Good luck.

3301

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.11 (GNU/Linux)

iQIcBAEBAgAGBQJS24E0AAoJEBgfAeV6NQkPNPEP/jxtRsM2AOE3KRChpl1IHxGe

oMyd/YjXW7/o8X6Cv+AYhzQhRhqOQPL0N+fVC2WNO64CGnOmTLbhZPoXpV1giSmA

UguBIWZ59MmitGVmiz68M/i5H68h7s7eXoC5u7/iZjVMQBr86J1iJyeabVjJMSp+

OI8ouTuVAZ5lccvUy9UpV82GtGZoM7P1xxWJGpM3LIz6mv7VdfogrCNAW0jhZ2/x

8eiiHFuB5oktc9uTbIqhJQsESuc1u/uMkeb1OGXovKmD+zLtq+DPbWo8P0lnT70V

pyLOLM5CWsAjblU+5ohK57yFP6AV6x4l97BQyRmMOojh35QGKPVULZG4sRPKsuG5

nw93gRi6/eQ/aQvQuEvkf4lbj/V5G4kOj/YcQhyjAWdo1UPl9zkUXs2lKH7sUUms

P1WV6eyL6rAqpUXfwpDSfaPTPquIwuFsEl5z/d14IXnR3s+LQjlFDO3DE2d9QlEr

h5daiLFEvH+wyoJ5aPOsSkT+QJqCrVQNnbbQYzYKeMKAshu1LWuk1ZQ0XAEA6C2b

zbiPcXg0OMO+VWkhscZwxIHr1N5TVDj3NOszCfUe7lrYZhE0F/TL50NkGxw9+2qH

byDA8E4Yhe2c7pUVgs3OQLX46N4SOlbsH2MNXO9Y5jjI2Oj+OwLQ07F8jouNvXN4

kY3+nCV1PPLtiOu1CCP1

=/MnD

-----END PGP SIGNATURE-----

Note the number square (not magic) in the middle of the message:

10 12 10

12 14 12

10 12 10

The only difference between the messages in each image was the number used to make the border of the number square. Page 1's outguess (above) used '3', page 2 used '3', page 3 used '0', and page 4 used '1', spelling out 3301.

The onion went offline shortly afterwards. Six days later it came back online, sporting the following message:

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Hello. You have done well to come this far.

Please paste the magic squares into the appropriate textareas below, then

provide the URL to your Tor hidden service.

The path to your CGI script which accepts uploads should be '/cgi-bin/upload'

and the HTML form input which accepts file uploads should be named 'file'.

Additionally, please generate a GnuPG key pair, and place the public key

in the location '/key.asc'.

We will contact you soon.

Good luck.

3301

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.11 (GNU/Linux)

iQIcBAEBAgAGBQJS43VRAAoJEBgfAeV6NQkPVcQP/Rnli3AdTLAj28W1SMHTD6v0

Q67n89uGF6ZeD4U+dD2FHULAL9upNBRdzF7golqcfJCpeIKN0JYyilpGgSyTQmx+

yJXinlq4ZY+NNN45t8FtULvpVVO+L1ztF6dcohK+ZhAWWFj5u5WwEINx0mo+TE35

S7imfprBdk2C5B/E8ds7m35s74oWfdys8oY+vUHzOT4KB0SYFbankH6aLIe7fiTa

STB1Effelhg9F8YjDsopFHyF/kozI+eYk9yJcDEhlO4aiIkfZdNdLhXz80SIKw9v

ryjTCPUJfFrVpaelHxBefTOQHPfQEWnua7h6V6bx8Wiem7eiNyfXMAk1uoiu9zWW

FbA+MIFZ711kLvzD9Sg/0YGY97Gzzt0M1e8Pl/JYYzjOzFOH5pqmgMoOTBO0bvV+

d+QaLLiKmH0cYdTKmLv2xJ/1y7z0pakmgXCOhzIVQCDwoMxfGELLi9MNroaZFK3e

JzeDy828EfafrWpQ/LNzovb0XHyR1p4RRLq9vqMTFo0U0U7lLDKfWJvs4Y73o2Pn

QRNzn2+2GPQPj1CRMp5gxDFzwAMT0RBAfagkDiQu2uDxk8NZfSWkJVmsAUlLQLA9

7Wa5zuxPNvBf8Bws6y1C241FVfyttC4tNZEp2ShtbnHkZj1gFZf5v4rbq8wsxPrJ

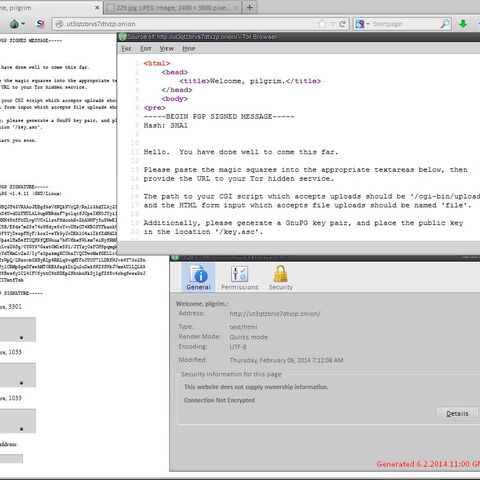
Jp8kkuLi0PWCITmtfTsb

=TwC1

-----END PGP SIGNATURE-----

and then three text boxes where the referenced 'magic squares' had to be pasted. At the bottom was a text box for the onion url of the hidden service mentioned in the previous message.

Screenshot of the page:

[](https://uncovering-cicada.fandom.com/wiki/File:Onion6welcome.jpg)

The page accepted any magic square that matched the criteria (e.g. first box square had to be order 5 and sum to 3301). Some time after this, it was discovered that the mp3 'interconnectedness' found on Onion 5 contained steganographically hidden data. The tool used to hide this data was 'OpenPuff'. The data hidden was a text file named 'magicsquares.txt' containing the following:

434 1311 312 278 966

204 812 934 280 1071

626 620 809 620 626

1071 280 934 812 204

966 278 312 1311 434

7 375 236 190 27 17 181

351 223 14 47 293 98 7

456 232 121 114 72 23 15

16 65 270 331 270 65 16

15 23 72 114 121 232 456

7 98 293 47 14 223 351

181 17 27 190 236 375 7

272 138 341 131 151

366 199 130 320 18

226 245 91 245 226

18 320 130 199 366

151 131 341 138 272

Two order 5 and one order 7 magic square, summing to 3301, 1033, and 1033.

Upon submitting these squares and a link to your hidden service, the following page displayed:

< html >< head >< /head >< body >

< p > Thank you for you submission. < /p >

< img src="[view-source:<http://ut3qtzbrvs7dtvzp.onion/107.jpg> /107.jpg]" / >< br / >

< img src="[view-source:<http://ut3qtzbrvs7dtvzp.onion/167.jpg> /167.jpg]" / >< br / >

< img src="[view-source:<http://ut3qtzbrvs7dtvzp.onion/229.jpg> /229.jpg]" / >< br / >

[](https://uncovering-cicada.fandom.com/wiki/File:107.jpg)[](https://uncovering-cicada.fandom.com/wiki/File:167.jpg)[](https://uncovering-cicada.fandom.com/wiki/File:229.jpg)Note that the <html> tag is not properly closed.

The images on the page:

The final image is enormous, perhaps indicating the end of the book.

Decoding the runes on the page using a Vignere cipher and key CIRCUMFERENCE yields

A KOAN: DURING A LESSION: THE MAS

TER EXPLAINED THE I:"THE

I IS THE VOICE OF THE CIRCU

MFERENCE,"HE SAID.WHEN AS

KED BY A STUDENT TO EXPLAIN

WHAT THAT MEANT, THE MASTER SA

ID"IT IS A VOICE INSIDE YOUR H

EAD"."I DON'T HAVE A VOICE I

N MY HEAD," THOUGHT THE STUDENT,

AND HE RAISED HIS HAND TO TE

LL THE MASTER.THE MASTER STOP

--page change--

PED THE STUDENT,AND SAID"THE

VOICE THAT JUST SAID YOU HAV

E NO VOICE IN YOUR HEAD, IS THE

I."AND THE STUDENTS WERE ENL

LIGHTENED:

The text on the final page is not encoded. Transcription via Gematrius Primus yields:

AN INSTRUCTION:QUESTION ALL

THINGS: DISCOVER TRUTH INSIDE

YOURSELF: FOLLOW YOUR TRU

TH: IMPOSE NOTHING ON OTHERS.

KNOW THIS:

434 1311 312 278 966

204 812 934 280 1071

626 620 809 620 626

1071 280 934 812 204

966 278 312 1311 434

**The End**

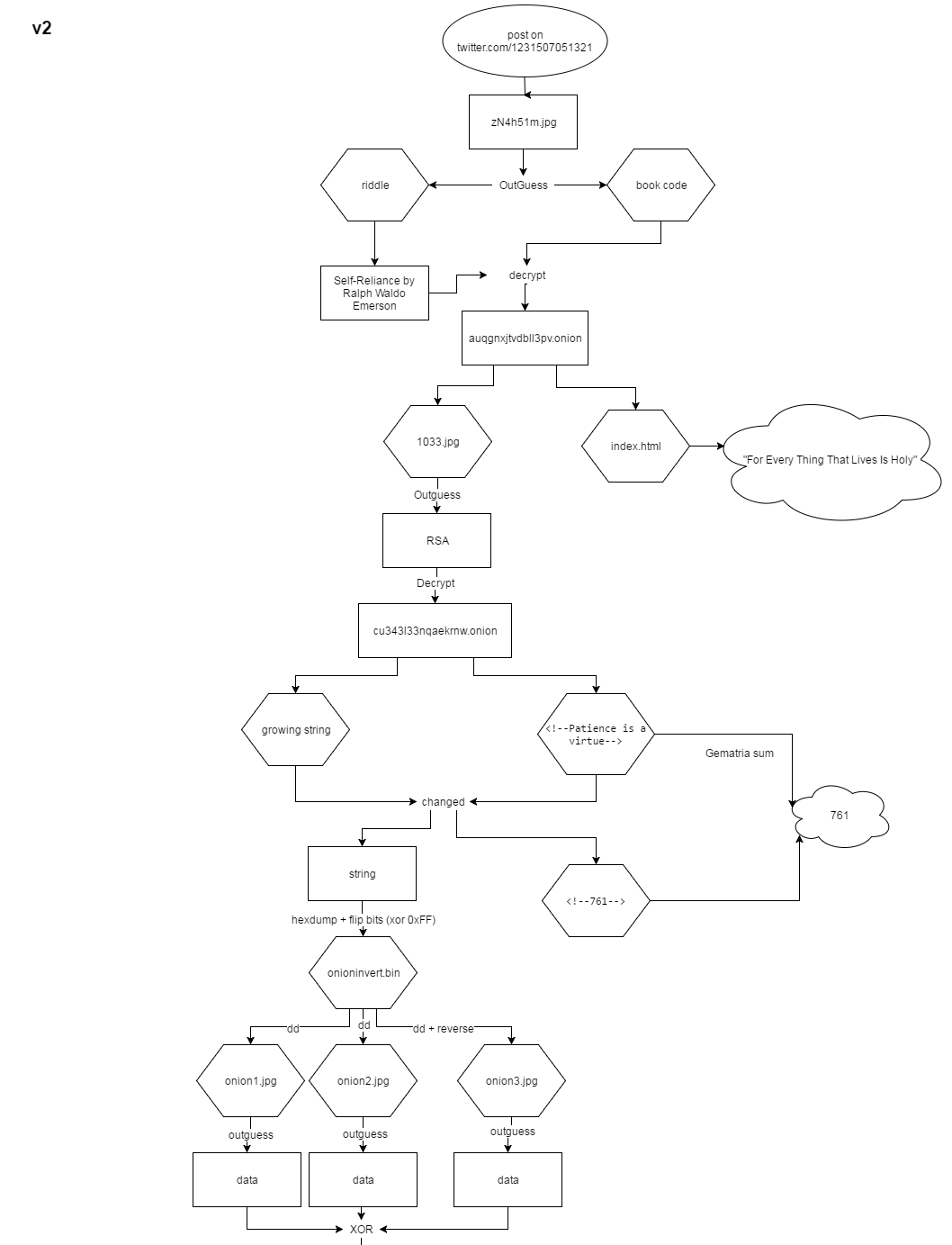
As usual, after the final submission we heard nothing. The IRC channels slowly died. The onions all went offline.

*Author's Note: when I say we heard nothing, 'we' references the collective of solvers. Cicada does not recruit groups, only individuals.*

Long after the community as a whole had given up, a response was finally received. Some got a message similar to the one from the end of 2013 (*or so they claim*), and the rest got a link to an onion, containing the remaining pages of the Liber Primus, a 58 page dump of images. These will be discussed [here.](https://uncovering-cicada.fandom.com/wiki/What_Happened_Liber_Primus_(Post_2014))

**Infographics**

Schematic presentation of all steps of 2014 puzzle: <http://imgur.com/42VKTWB>

2014 puzzle scheme

