ARL South Cyber Rapid Innovation Group

Ransomware Key Recovery

Developed by Dr. Jaime C. Acosta, Mr. Adrian Belmontes

This training exercise is meant to be completed in groups.

In May 2017, the WannaCry ransomware crypto worm spread worldwide, encrypting a victims' data and demanding a ransom to be paid in the form of bitcoin. Over 200,000 victims were affected by this attack worldwide including the National Health Service hospitals in England and Scotland. Although a kill switch has been addressed to combat the worm, the danger continues in the different emerging variants.

You and your partner are part of an elite research group, developing the next vaccine for the flu. The machine being used to store data requires software that is only compatible with an older version of Windows 7, making it vulnerable to attacks. In an attempt to look for a better alternative, your coworker, Alice, comes across a free program. Unfortunately, the program turned out to be ransomware and infected the network.

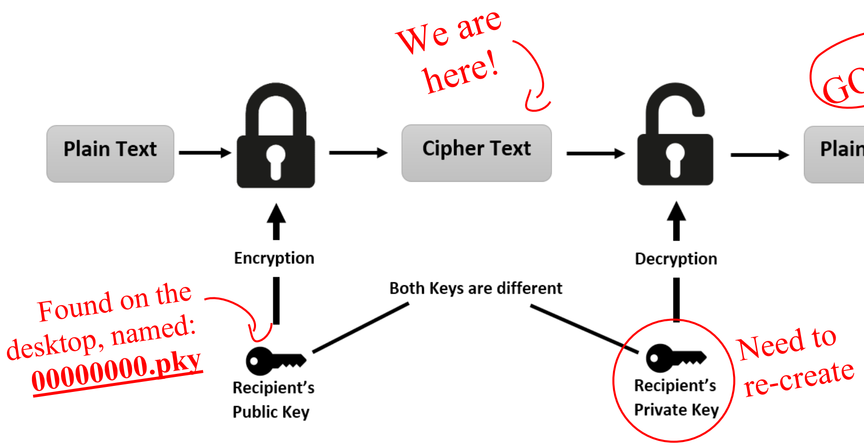


Figure 1 RSA encryption protocol.

Luckily, your personal computer was updated to include the latest security patch and you decide to use it to analyze the malware. Alice helps by drawing out the plan above. You and your partner need to help Alice recover the data to finish the vaccine on time!

Your RIG consists of the following.

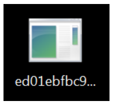
- **Win7-keyRecovery VM**: the machine that you will infect with the WannaCry ransomware.
- **CleanWin7 VM**: the machine that you will use for analysis and to generate a decryption key.
- A directory called **Shared folder** used for sharing files between the machines. This folder is located on the Desktop of both machines.

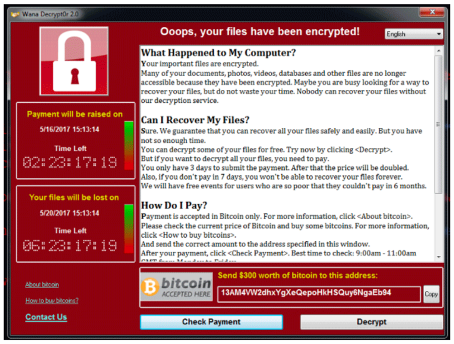
Step 1 - Dynamic Analysis of the Process Memory (Win7-keyRecovery Machine)

Run the WannaCry malware in your sandbox and observe its behavior on the machine.

1. Start the Win7-keyRecovery by clicking on the corresponding link in your browser. You should see a Windows Desktop. If prompted, login using the following credentials:

   Username: **Reversing**
   Password: **malware**

2. Open ***WannaCry*** by double clicking on this file:  ed01ebfbc9... on the desktop. Click ***No*** on any prompts that may come up.
3. The program will execute and will finish once you receive the message shown below.



Wana Decrypt0r 2.0

Ooops, your files have been encrypted!

English

What Happened to My Computer?
Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Payment will be raised on
5/16/2017 15:13:14
Time Left
02:23:17:19

Your files will be lost on
5/20/2017 15:13:14
Time Left
06:23:17:19

Send $300 worth of bitcoin to this address:
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

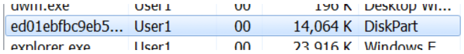
****

You are smart, so you want to freeze and save a copy of the computer so you can analyze it later. You do this as soon as possible to make sure the ransomware or the operating system don't remove important artifacts that are essential for undoing the chaos.

****

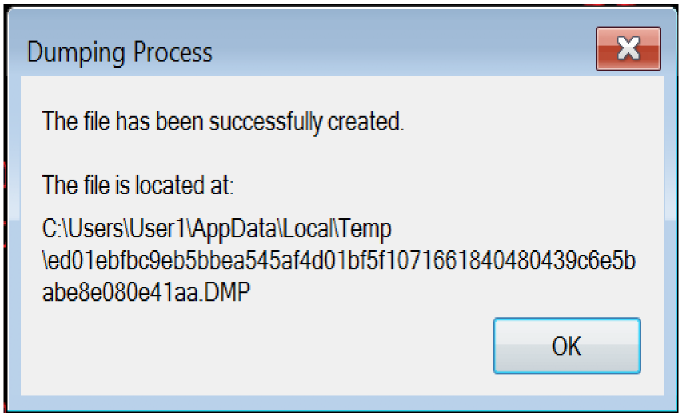
4. Open ***Task Manager*** by pressing the *windows key* and searching *task manager*. Clicking on '***View running processes with Task Manager***'

5. Click on the ***Processes*** tab and click on ***Show processes from all users.*** Find the WannaCry process (it starts with ***ed01ebfbc…***)



```
dwm.exe            User1   00     196 K  Desktop Wi...
ed01ebfbc9eb5...   User1   00  14,064 K  DiskPart
explorer.exe       User1   00  23,916 K  Windows E
```

6. Right-click and select ***Create Dump File***. You should see the following screen.



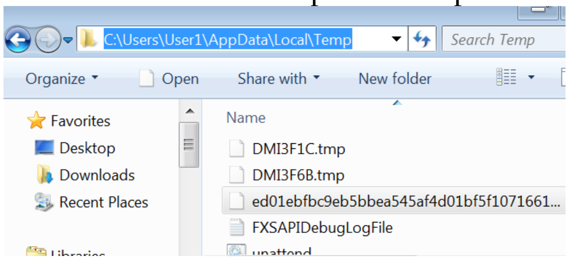
Dumping Process

The file has been successfully created.

The file is located at:

C:\Users\User1\AppData\Local\Temp
\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5b
abe8e080e41aa.DMP

7. Open an explorer window and then enter the path in the path bar as shown below.



C:\Users\User1\AppData\Local\Temp

8. Drag and drop the memory dump file to the Desktop.

9. Now let's copy a few files to a shared folder so we can analyze them on a separate machine. Start by double clicking the folder on the desktop named: ***Shared Folder***

10. Locate the ***public key file***, named: ***00000000.pky*** on the Desktop**.** Copy both this ***public key file*** and the ***memory dump file*** to the shared folder.

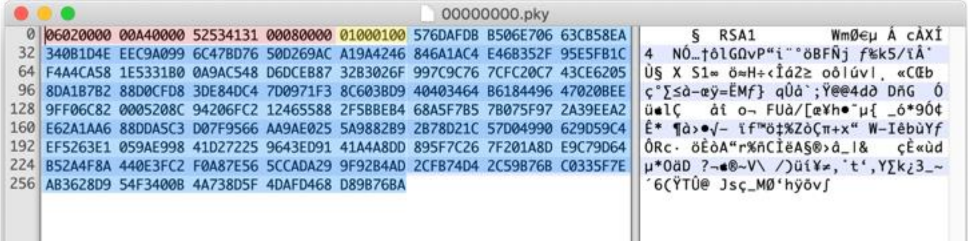
11. Open ***HxD Hex Editor*** by ***double-clicking*** on on the desktop. Then click ***File->Open*** and select the ***public key*** located on the desktop.

The next big step is to figure out the bad guy's private key. We'll need to look at a few values from the public key: a **special header**, the **exponent**, and the **modulus**.

****

The public key can be seen in the image below and consists of a ***special header*** (red), the public ***exponent*** (yellow), and the ***modulus*** (blue). Figure 2 is colored to identify components easier. When using HxD, the components will not be colored. There is a bit of an issue as the values are actually in the reverse order that we will need them. For example, if we encounter the following 8 digits ***12 34 56 78***, we would need to convert them to the following instead: ***78 56 34 12***. (This is actually a known issue in computing called little versus big endianness).

For example, consider the public exponent bytes in little-endian: 01 00 01 00. The actual value of these bytes would be 00 01 00 01. The bytes are reversed.



00000000.pky

```
  0 06020000 00A40000 52534131 00080000 01000100 576DAFDB B506E706 63CB58EA        §   RSA1         WmØ€µ Á cÀXÍ
 32 340B1D4E EEC9A099 6C47BD76 50D269AC A19A4246 846A1AC4 E46B352F 95E5FB1C   4   NÓ…†ôlGΩvP“i¨°öBFÑj ƒ‰k5/ïÂ˙
 64 F4A4CA58 1E5331B0 0A9AC548 D6DCEB87 32B3026F 997C9C76 7CFC20C7 43CE6205   Ù§ X S1∞ ö≈H÷‹Îá2≥ oô|úv|¸ «CŒb
 96 8DA1B7B2 88D0CFD8 3DE84DC4 7D0971F3 8C603BD9 40403464 B6184496 47020BEE   ç°∑≤à–œÿ=ËMƒ} qÛà`;Ÿ@@4d∂ DñG  Ó
128 9FF06C82 0005208C 94206FC2 12465588 2F5BBEB4 68A5F7B5 7B075F97 2A39EEA2   ü●lÇ   åî o¬ FUà/[æ¥h•˜µ{ _ó*9Ó¢
160 E62A1AA6 88DDA5C3 D07F9566 AA9AE025 5A9882B9 2B78D21C 57D04990 629D59C4   Ê* ¶à>•√– ïf™ö‡%ZòÇπ+x“ W–IêbùYƒ
192 EF5263E1 059AE998 41D27225 9643ED91 41A4A8DD 895F7C26 7F201A8D E9C79D64   ÔRc· öÈòA“r%ñCÌëA§®›â_|&   çÈ«ùd
224 B52A4F8A 440E3FC2 F0A87E56 5CCADA29 9F92B4AD 2CFB74D4 2C59B76B C0335F7E   µ*OäD ?¬●®~V\ /)üí¥≠,˙t‘,Y∑k¿3_~
256 AB3628D9 54F3400B 4A738D5F 4DAFD468 D89B76BA                              ´6(ŸTÛ@ Jsç_MØ‘hÿõv∫
```

Figure 2 Public Key (Color-Coded) – Consists of header (in red), exponent (in yellow), and modulus (in blue).

****

12. Look at your hex editor and identify your public exponent. Write down your public exponent, ***as displayed in the hex editor***? _______________________

13. ***Reverse*** the bytes from number 1215 as described above and then write them here. Hint: it should match the example explained above. _________________________

14. Look at your hex editor and identify your modulus. What are the ***first 8 characters (or 4 bytes)*** of the modulus, ***as displayed in the hex editor***? _______________________

15. Look at your hex editor and identify your modulus. What are the ***last 8 character (or 4 bytes)*** of the modulus, ***as displayed in the hex editor***? ___________________________

16. ***Reverse*** the bytes from number 15 as described above and then write them here. _________________________

****

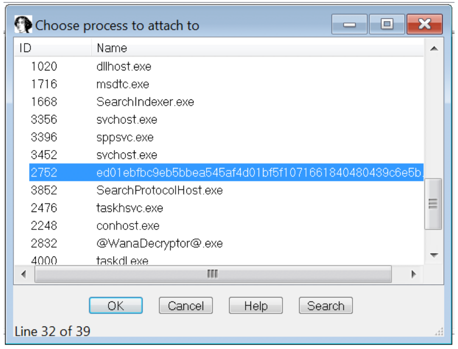
Now you will use IDA Pro to search for a few more numbers that were present in the memory dump for the ransomware process.

You will also simulate the effects of restarting the machine, as a first instinct, in the hopes that everything will go back to normal.

****

17. Open up ***IDA Pro Free*** by ***double-clicking*** on IDA Pro Free on the desktop. Then click on ***GO.***

18. On the toolbar, click on ***Debugger -> Attach -> Local Windows Debugger***. Then look for the WannaCry process (it starts with ***ed01ebfbc…***) and click ***OK***.



Choose process to attach to

ID	Name
1020	dllhost.exe
1716	msdtc.exe
1668	SearchIndexer.exe
3356	svchost.exe
3396	sppsvc.exe
3452	svchost.exe
2752	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5b
3852	SearchProtocolHost.exe
2476	taskhsvc.exe
2248	conhost.exe
2832	@WanaDecryptor@.exe
4000	taskdl.exe

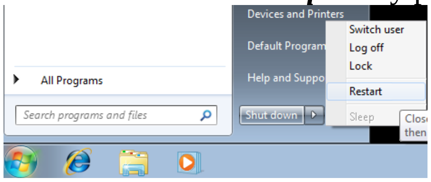
Line 32 of 39

19. When the main IDA Pro Window opens, on the toolbar click on the ***Play*** button just below the toolbar and then click the ***Pause*** button.

20. Now open the search Windows by clicking ***Search -> sequence of bytes***. This will take a few seconds to complete.

21. In the String box, type in your ***answer for number 16***. Make sure ***Find all occurrences*** is selected. How many occurrences did it find? __________

22. Click on the ***Windows button*** and select ***Restart***. ***Force quit*** any processes that are active.



23. After restart, run the ***WannaCry process*** on the Desktop again.

24. ***Repeat steps 17 – 21***.

25. How many occurrences did it find this time? __________

26. Did the number of occurrences change? Write down a few reasons why you think this is the case.

____

____

____

Step 2 – Extracting Encryption Parameters from the Memory Dump

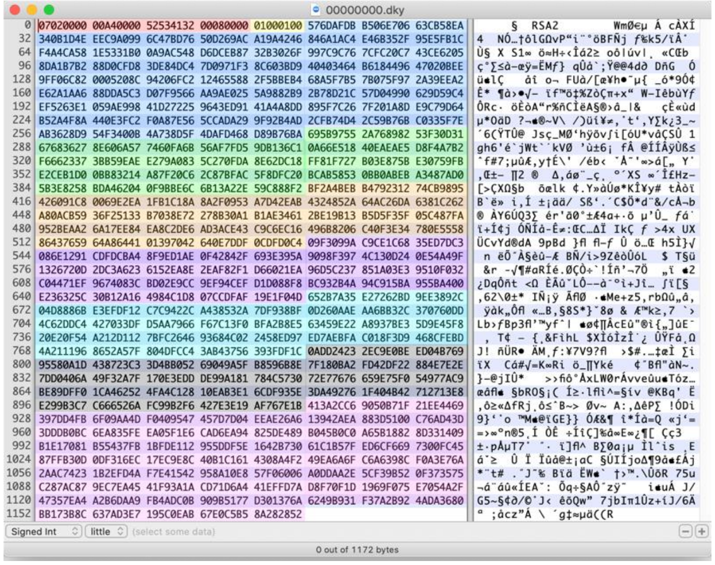
(CleanWin7 Machine)

Recall that in the previous step, you froze a copy of the process data before you restarted. Now you can look in this image to find all of the values you need to recreate the decryption key!

****

Before building the private key, we'll take a closer look at the full structure of the RSA keys. The RSA structure contains several items in addition to the ones you captured previously; many of which we'll need eventually to re-create the private key as shown in Figure 1. The memory dump will provide us with the two prime numbers and the rest can be obtained with a calculator.

Name	Size	Variable name in RSA equation	Color-code
Heading	16 bytes (32 characters)	N/A	Red
Public Exponent	4 bytes (8 characters)	**E**	Yellow
Modulus	256 bytes (512 characters)	**N**	Blue
Prime Number 1	128 bytes (256 characters)	**P**	Green
Prime Number 2	128 bytes (256 characters)	**Q**	Orange
CRT Exponent 1	128 bytes (256 characters)	**dP**	Purple
CRT Exponent 2	128 bytes (256 characters)	**dQ**	Light Blue
CRT Exponent Coefficient	128 bytes (256 characters)	**qInv**	Black
Private Exponent	256 bytes (512 characters)	**D**	Pink



00000000.dky

```
   0  07020000 00A40000 52534132 00080000 01000100 576DAFDB B506E706 63CB58EA
  32  340B1D4E EEC9A099 6C47BD76 50D269AC A19A4246 846A1AC4 E46B352F 95E5FB1C
  64  F4A4CA58 1E5331B0 0A9AC548 D6DCEB87 32B3026F 997C9C76 7CFC20C7 43CE6205
  96  8DA1B7B2 88D0CFD8 3DE84DC4 7D0971F3 8C603BD9 40403464 B6184496 47020BEE
 128  9FF06C82 0005208C 94206FC2 12465588 2F5BBEB4 68A5F7B5 7B075F97 2A39EEA2
 160  E62A1AA6 88DDA5C3 D07F9566 AA9AE025 5A9882B9 2B78D21C 57D04990 629D59C4
 192  EF5263E1 059AE998 41D27225 9643ED91 41A4A8DD 895F7C26 7F201A8D E9C79D64
 224  B52A4F8A 440E3FC2 F0A87E56 5CCADA29 9F92B4AD 2CFB74D4 2C59B76B C0335F7E
 256  AB3628D9 54F3400B 4A738D5F 4DAFD468 D89B76BA 695B9755 2A768982 53F30D31
 288  67683627 8E606A57 7460FA6B 56AF7FD5 9DB136C1 0A66E518 40EAEAE5 D8F4A7B2
 320  F6662337 3BB59EAE E279A083 5C270FDA 8E62DC18 FF81F727 B03E875B E30759FB
 352  E2CEB1D0 0BB83214 A87F20C6 2C87BFAC 5F8DFC20 BCAB5853 0BB0ABEB A3487AD0
 384  5B3E8258 BDA46204 0F9BBE6C 6B13A22E 59C888F2 BF2A4BEB B4792312 74CB9895
 416  426091C8 0069E2EA 1FB1C18A 8A2F0953 A7D42EAB 4324852A 64AC26DA 6381C262
 448  A80ACB59 36F25133 B7038E72 278B30A1 B1AE3461 2BE19B13 B5D5F35F 05C487FA
 480  952BEAA2 6A17EE84 EA8C2DE6 AD3ACE43 C9C6EC16 496B8206 C40F3E34 780E5558
 512  86437659 64A86441 01397042 640E7DDF 0CDFD0C4 09F3099A C9CE1C68 35ED7DC3
 544  086E1291 CDFDCBA4 8F9ED1AE 0F42842F 693E395A 9098F397 4C130D24 0E54A49F
 576  1326720D 2DC3A623 6152EA8E 2EAF82F1 D66021EA 96D5C237 851A03E3 9510F032
 608  C04471EF 9674083C BD02E9CC 9EF94CEF D1D088F8 BC932B4A 94C915BA 955BA400
 640  E236325C 30B12A16 4984C1D8 07CCDFAF 19E1F04D 652B7A35 E27262BD 9EE3892C
 672  04D8886B E3EFDF12 C7C9422C A438532A 7DF938BF 0D260AAE AA6BB32C 370760DD
 704  4C62DDC4 427033DF D5AA7966 F67C13F0 BFA2B8E5 63459E22 A8937BE3 5D9E45F8
 736  20E20F54 A212D112 7BFC2646 93684C02 2458ED97 ED7AEBFA C018F3D9 468CFEBD
 768  4A211196 8652A57F 804DFCC4 3AB43756 393FDF1C 0ADD2423 2EC9E0BE ED04B769
 800  95580A1D 438723C3 3D4BB052 69049A5F B8596B8E 7F180BA2 FD42DF22 884E7E2E
 832  7DD0406A 49F32A7F 170E3EDD DE99A181 784C5730 72E77676 659E75F0 54977AC9
 864  BE89DFF0 1CA46252 4FA4C128 10EAB3E1 6CDF935E 3DA49276 1F404B42 712713E8
 896  E299B3C7 C666526A FC99B2F6 427E3E19 AF767E1B 413A2CC6 9050B71F 21EE4469
 928  397DD4FB 6F09AA4D F0409547 457D7D04 EEAE26A6 13942AEA 883D5100 C76AD43D
 960  3DDDB0BC 6EA835FE EA05F1E6 CAD6EA94 825DE489 B045B0C0 A65B1882 8D331409
 992  B1E17081 B55437FB 1BFDE112 955DDF5E 1642B730 61C1B57F ED6CF669 7300FC45
1024  87FFB30D 0DF316EC 17EC9E8C 40B1C161 4308A4F2 49EA6A6F C6A6398C F0A3E76A
1056  2AAC7423 1B2EFD4A F7E41542 958A10E8 57F06006 A0DDAA2E 5CF39B52 0F373575
1088  C287AC87 9EC7EA45 41F93A1A CD71D6A4 41EFFD7A D8F70F1D 1969F075 E7054A2F
1120  47357EA4 A2B6DAA9 FB4ADC0B 909B5177 D301376A 6249B931 F37A2B92 4ADA3680
1152  BB173B8C 637AD3E7 195C0EAB 67E0C5B5 8A282852
```

Signed Int | little | (select some data)

0 out of 1172 bytes

Figure 3 Format with Color-coded values

****

1. Exit the current Windows machine by clicking the Back button in your Browser. Now, start the CleanWin7 by clicking on the corresponding link.

2. Open ***HxD Hex Editor*** by ***double-clicking*** on HxD the desktop. Then open the ***public key*** that you stored in the shared folder (00000000.pky).

****

When WannaCry creates the public key, it uses two prime numbers to calculate a key value. The primes are stored in memory until they are overwritten or cleared. Creating the dump file in a timely manner decreases the chance of losing that data (Note: newer machines clear the memory right away).

****

3. Click on ***File -> Open*** and selecting the ***DMP file*** on the shared folder (it starts with ***ed01ebfbc…***).

4. On the toolbar click on ***Search -> Find.*** Then click on the ***Hex-values*** tab and select ***All*** for the search directions.

5. In the search bar, enter your ***answer from Step 1, number 15*** and click on ***Search all***.

6. Double-click through the occurrences and find the one that contains the following characters right after the value you searched: ***11000110***.

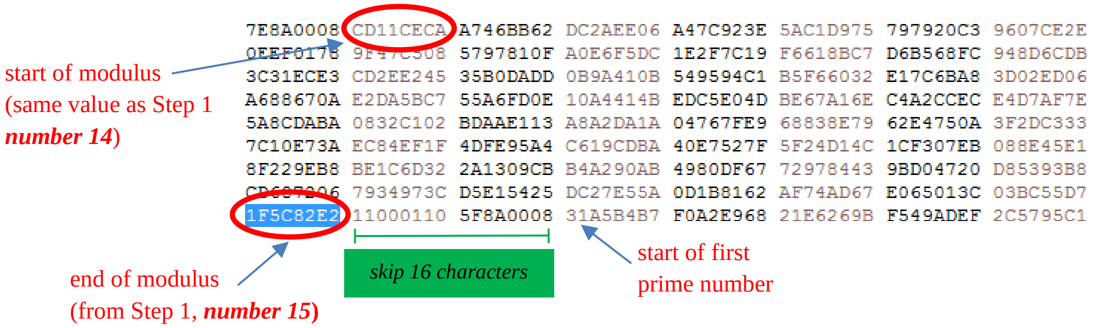
   Now we need to extract a few numbers from the rest of this memory dump. They're mixed with other numbers, so we'll have to skip a few numbers and paste only what we need into a temporary spreadsheet file named ***Calculations Template***.

7. Open ***Calculations Template*** by ***double-clicking*** on Calculations Template on the desktop we'll start filling in highlighted the green and blue cells.

8. Enter you answer from ***Step 1, number 12*** under ***Public Exponent (little endian).***

9. Go back to the HxD application. See Figure 4 on the next page (your values and their placement will be different). From the found value (the ***end of modulus***). Look roughly ***8 rows up*** to find the value from ***Step 1 number 14***. This is the start of the modulus.

10. Highlight the bytes corresponding with the entire modulus the HxD program (the length of bytes you select can be seen at the bottom of application labeled ***"Length(d):"*** make sure it's ***256 bytes***). ***Copy and paste*** the modulus into the ***Calculations Template*** spreadsheet under ***Modulus (little endian).***



```
7E8A0008 CD11CECA A746BB62 DC2AEE06 A47C923E 5AC1D975 797920C3 9607CE2E
0EEF0178 9F47C508 5797810F A0E6F5DC 1E2F7C19 F6618BC7 D6B568FC 948D6CDB
3C31ECE3 CD2EE245 35B0DADD 0B9A410B 549594C1 B5F66032 E17C6BA8 3D02ED06
A688670A E2DA5BC7 55A6FD0E 10A4414B EDC5E04D BE67A16E C4A2CCEC E4D7AF7E
5A8CDABA 0832C102 BDAAE113 A8A2DA1A 04767FE9 68838E79 62E4750A 3F2DC333
7C10E73A EC84EF1F 4DFE95A4 C619CDBA 40E7527F 5F24D14C 1CF307EB 088E45E1
8F229EB8 BE1C6D32 2A1309CB B4A290AB 4980DF67 72978443 9BD04720 D85393B8
CD687206 7934973C D5E15425 DC27E55A 0D1B8162 AF74AD67 E065013C 03BC55D7
1F5C82E2 11000110 5F8A0008 31A5B4B7 F0A2E968 21E6269B F549ADEF 2C5795C1
```

start of modulus (same value as Step 1 ***number 14***)

end of modulus (from Step 1, ***number 15***)

skip 16 characters

start of first prime number

Figure 4 Encryption numbers in DMP file.

11. After the end of the modulus, skip 16 characters (***8 bytes***), and then the next 256 characters ***(128 bytes)*** make up the ***first prime number*** (see Figure 4)**.** Highlight the bytes corresponding with the first prime number the HxD program (the length of bytes you select can be seen at the bottom of application – make sure it's ***128 bytes***) and then ***copy*** and ***paste*** them into the spreadsheet under ***Prime 1 (little endian).***

12. After the first prime number, skip another 16 characters (***8 bytes***), and then the next 256 characters (***128 bytes***) make up the ***second prime number***. Highlight and copy the second prime number and place it under ***Prime 2 (little endian)***. The length of bytes you select can be seen at the bottom of the HxD application – make sure it's **128**.

We've found all of the values we need from our memory dump. The next step is to get the reverse of some of the numbers (or the ***big endian*** version) for the modulus and primes.

13. Open the ***Endian Converter*** by ***double-clicking*** on ConvertEnd... - Shortcut on the desktop. From the ***Calculations Template*** copy over the values under ***Modulus (little endian)***, click ***Convert*** and then paste the result back into the ***Calculations Template*** under ***Modulus (big endian).***

    If the converter prompts you that your number is **not even,** make sure there is no space at the end. If there is no space, add a 0 (zero) to the start of your number and try again.

14. Repeat the conversion process from ***number 13*** for all of the little endian values in your spreadsheet (**primes, public exponent, special header**). Hint: Public Exponent (big endian) should match your answer from Step 1, number 13.

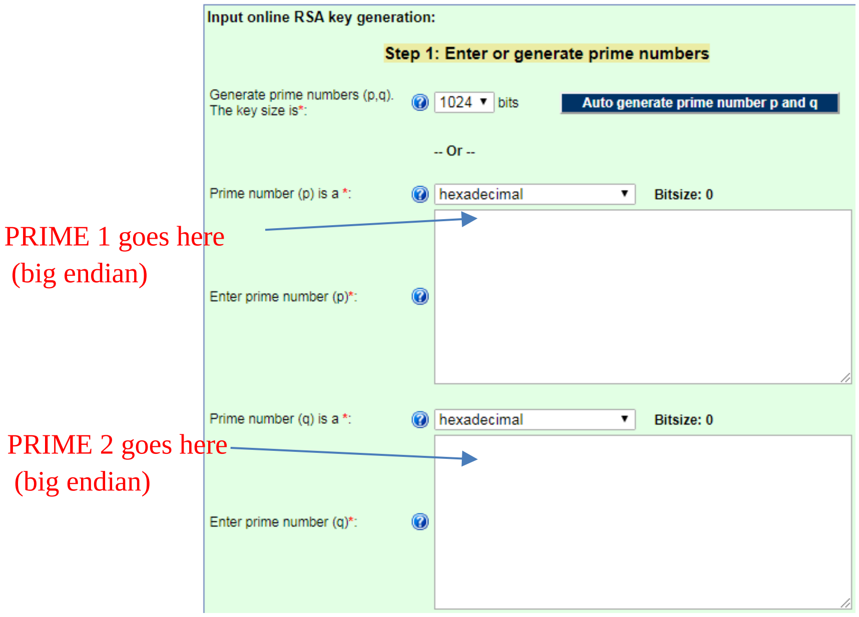
Step 3 – Calculating the Missing Encryption Values

The following shows the mapping between values and their variable name in RSA equations.

Name	Variable name in RSA equation	Status
Heading	N/A	Found
Public Exponent	**e**	Found
Modulus	**n**	Found
Prime Number 1	**p**	Found
Prime Number 2	**q**	Found
CRT Exponent 1	**dP**	Need to Calc
CRT Exponent 2	**dQ**	Need to Calc
CRT Exponent Coefficient	**qInv**	Need to Calc
Private Exponent	**d**	Need to Calc

You'll need to calculate a few more numbers with a script conveniently located on your Desktop.

1. Open the ***RSA key generation*** script by ***double-clicking*** on the desktop. Scroll down until you see the ***Input online RSA key generation form***.
2. ***Copy and paste*** the ***PRIME 1 (big endian)*** and ***PRIME 2 (big endian)*** from ***Calculations Template*** into the boxes.



Input online RSA key generation:

Step 1: Enter or generate prime numbers

Generate prime numbers (p,q). The key size is*: 1024 bits

-- Or --

Prime number (p) is a *: hexadecimal Bitsize: 0

Enter prime number (p)*:

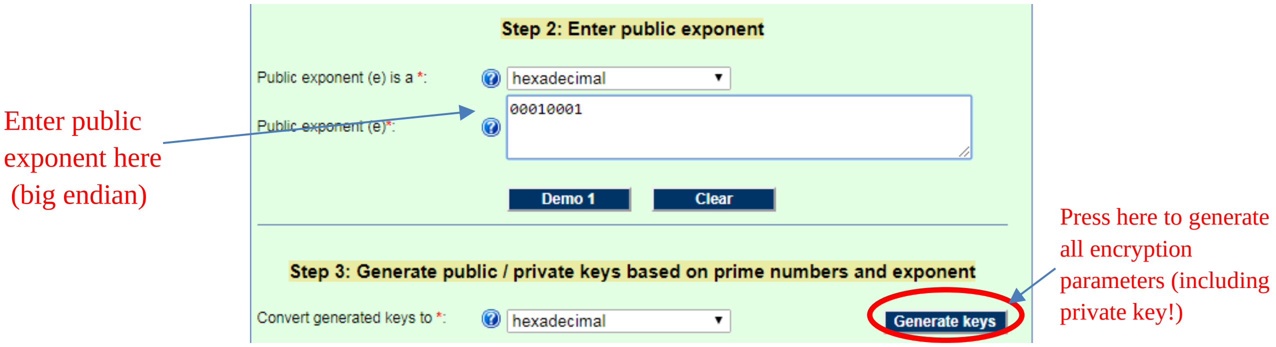
Prime number (q) is a *: hexadecimal Bitsize: 0

Enter prime number (q)*:

PRIME 1 goes here (big endian)

PRIME 2 goes here (big endian)

3. Scroll down further until you see ***Step 2: Enter public exponent.*** Enter your value for the ***public exponent (big endian)*** from the ***Calculations Template***, and then click on the ***Generate Keys*** button.



Step 2: Enter public exponent

Public exponent (e) is a *: hexadecimal

Public exponent (e)*: 00010001

Step 3: Generate public / private keys based on prime numbers and exponent

Convert generated keys to *: hexadecimal

Enter public exponent here (big endian)

Press here to generate all encryption parameters (including private key!)

4. Scroll down and verify that the ***modulus*** matches what you have in your spreadsheet.

5. Look through the form and copy the private exponent (d) into the ***private exponent (big endian)*** on your spreadsheet. Continue through and also copy into the spreadsheet the ***big endian*** values for CRT exponent 1 (***dP***), CRT exponent 2 (***dQ***), and CRT coefficient (**qInv**).

6. You should now have all of the values in your spreadsheet under the ***big endian*** column filled.

   Use the Endian Converter to calculate and then fill in the missing values in the little endian column.

****

If you are asking yourself, where do these equations come from? These equations are standard for the creation of RSA keys; however, the formatting and layout may differ.

****

Once you're done, close all other windows except for the
Calculations Template and the ***HxD application***.

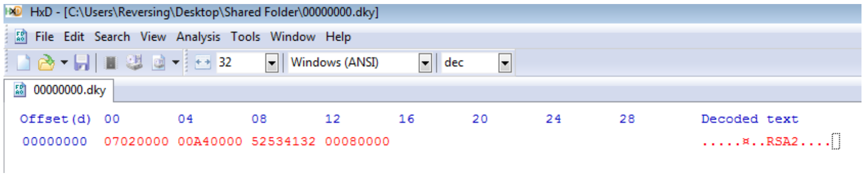
Step 4 – Regenerate the Private Key

At last, you have all of the values you need to regenerate the private key. Now you need to put them in the correct order (required by the ransomware) using the hex editor.

1. Create a new file by clicking ***File->New*** to begin creating the private key. Click on ***File->Save As…*** and name the file **00000000.dky** (8 zeroes) on the ***Desktop***. You can switch between the two files using the tabs at the top.

The key that WannaCry reads uses the ***little endian*** format; now we just have to create the key using the values that we've obtained in the previous steps.

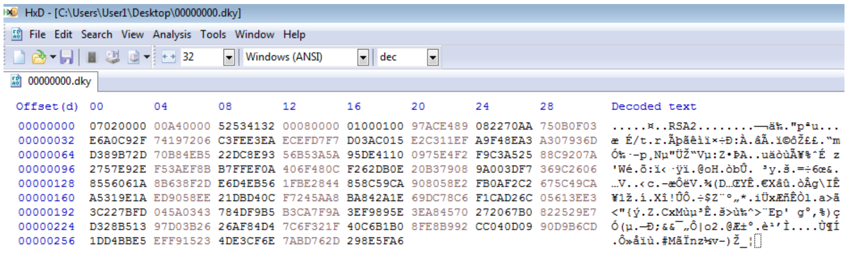
2. First, we'll create the ***special header.*** Make sure that your cursor is at the beginning of the file. Copy the value from your spreadsheet under the ***Special Header, Little Endian*** column. If a warning appears, select the ***don't ask again*** box and continue.



HxD - [C:\Users\Reversing\Desktop\Shared Folder\00000000.dky]
File Edit Search View Analysis Tools Window Help
32
Windows (ANSI)
dec
00000000.dky

```
Offset(d) 00       04       08       12       16       20       24       28         Decoded text
00000000  07020000 00A40000 52534132 00080000                                       ....¤..RSA2....
```

3. Next, copy over the little ***Public Exponent, Little Endian Column*** and the ***Modulus, Little Endian Column***. At this point, your file should have values through ***8 full rows*** and ***5 columns*** in the ***9th row***.

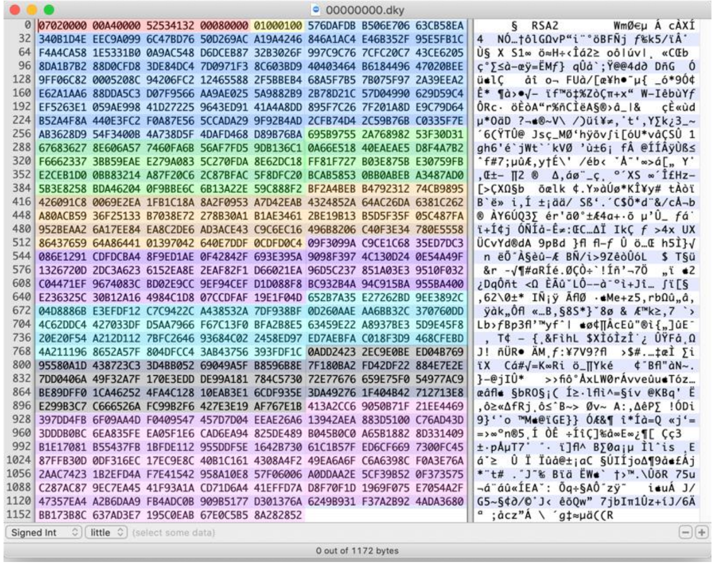


HxD - [C:\Users\User1\Desktop\00000000.dky]
File Edit Search View Analysis Tools Window Help
32
Windows (ANSI)
dec
00000000.dky

```
Offset(d) 00       04       08       12       16       20       24       28         Decoded text
00000000  07020000 00A40000 52534132 00080000 01000100 97ACE489 082270AA 750B0F03   .....¤..RSA2........—ä‰."p ª u...
00000032  E6A0C92F 74197206 C3FEE3EA ECEFD7F7 D03AC015 E2C311EF A9F48EA3 A307936D   æ É/t.r.Ãþãêìï×÷Ð:À.âÃ.ï©ôŽ££.“m
00000064  D389B72D 70B84EB5 22DC8E93 56B53A5A 95DE4110 0975E4F2 F9C3A525 88C9207A   Ó‰·–p¸Nµ"ÜŽ“Vµ:Z•ÞA..uäòùÃ¥%ˆÉ z
00000096  2757E92E F53AEF8B B7FFEF0A 406F480C F262DB0E 20B37908 9A003DF7 369C2606   'Wé.õ:ï‹·ÿï.@oH.òbÛ. ³y.š.=÷6œ&.
00000128  8556061A 8B638F2D E6D4EB56 1FBE2844 858C59CA 908058E2 FB0AF2C2 675C49CA   …V..‹c.–æÔëV.¾(D…ŒYÊ.€Xâû.òÂg\IÊ
00000160  A5319E1A ED9058EE 21DBD40C F7245AA8 BA842A1E 69DC78C6 F1CAD26C 05613EE3   ¥1ž.í.Xî!ÛÔ.÷$Z¨º„*.iÜxÆñÊÒl.a>ã
00000192  3C227BFD 045A0343 784DF9B5 B3CA7F9A 3EF9895E 3EA84570 272067B0 822529E7   <"{ý.Z.CxMùµ³Ê.š>ù‰^>¨Ep' g°‚%)ç
00000224  D328B513 97D03B26 26AF84D4 7C6F321F 40C6B1B0 8FE8B992 CC040D09 90D9B6CD   Ó(µ.—Ð;&&¯„Ô|o2.@Æ±°.è¹’Ì....ÙÍ
00000256  1DD4BBE5 EFF91523 4DE3CF6E 7ABD762D 298E5FA6                              .Ô»åïù.#MãÏnz½v-)Ž_¦
```

4. Recall the following Table and Figure from earlier. Continue by filling in the rest of the values starting with Prime Number 1 (***use little endian from here on***) to create the key structure as shown below.

Name	**Size**	**Variable name in RSA equation**	**Color-code**
Heading	16 bytes (32 digits)	N/A	Red
Public Exponent	4 bytes (8 digits)	**e**	Yellow
Modulus	256 bytes (512 digits)	**n**	Blue
Prime Number 1	128 bytes (256 digits)	**p**	Green
Prime Number 2	128 bytes (256 digits)	**q**	Orange
CRT Exponent 1	128 bytes (256 digits)	**dP**	Purple
CRT Exponent 2	128 bytes (256 digits)	**dQ**	Light Blue
CRT Exponent Coefficient	128 bytes (256 digits)	**qInv**	Black
Private Exponent	256 bytes (512 digits)	**d**	Pink



```
   0 07020000 00A40000 52534132 00080000 01000100 576DAFDB B506E706 63CB58EA         §   RSA2         WmØ€µ Á cÀXÍ
  32 340B1D4E EEC9A099 6C47BD76 50D269AC A19A4246 846A1AC4 E46B352F 95E5FB1C   4  NÓ…†ôlGΩvP“i¨˚öBFÑj ƒ‰k5/ïÂ˙
  64 F4A4CA58 1E5331B0 0A9AC548 D6DCEB87 32B3026F 997C9C76 7CFC20C7 43CE6205   Ù§ X S1∞ ö≈H÷‹Îá2≥ oôlúvl¸ «CŒb
  96 8DA1B7B2 88D0CFD8 3DE84DC4 7D0971F3 8C603BD9 40403464 B6184496 47020BEE   ç˚∑≤à–œÿ=ËMƒ} qÛà`;Ÿ@@4d∂ DñG  Ó
 128 9FF06C82 0005208C 94206FC2 12465588 2F5BBEB4 68A5F7B5 7B075F97 2A39EEA2   üкачlÇ   àî o¬ FUà/[æ¥h•˜µ{ _ó*9Ó¢
 160 E62A1AA6 88DDA5C3 D07F9566 AA9AE025 5A9882B9 2B78D21C 57D04990 629D59C4   Ê* ¶à>•√– ïf™ö‡%ZòÇπ+x“ W–IêbùYƒ
 192 EF5263E1 059AE998 41D27225 9643ED91 41A4A8DD 895F7C26 7F201A8D E9C79D64   ÔRc· öÈòA“r%ñCÌëA§®>â_|&    çÈ«ùd
 224 B52A4F8A 440E3FC2 F0A87E56 5CCADA29 9F92B4AD 2CFB74D4 2C59B76B C0335F7E   µ*OäD ?¬®~V\ /)üí¥≠,˙t‘,Y∑k¿3_~
 256 AB3628D9 54F3400B 4A738D5F 4DAFD468 D89B76BA 695B9755 2A768982 53F30D31   ´6(ŸTÛ@ Jsç_MØ‘hÿöv∫i[óU*vâÇSÛ 1
 288 67683627 8E606A57 7460FA6B 56AF7FD5 9DB136C1 0A66E518 40EAEAE5 D8F4A7B2   gh6'é`jWt`˙kVØ ’ù±6¡ fÂ @ÍÎÂÿÙß≤
 320 F6662337 3BB59EAE E279A083 5C270FDA 8E62DC18 FF81F727 B03E875B E30759FB   ˆf#7;µûÆ,y†É\' /éb‹ ˇÅ¯'∞>á[„ Y˙
 352 E2CEB1D0 0BB83214 A87F20C6 2C87BFAC 5F8DFC20 BCAB5853 0BB0ABEB A3487AD0   ,Œ±– ∏2 ®  ∆,áø¨_ç¸ °´XS ∞´Î£Hz–
 384 5B3E8258 BDA46204 0F9BBE6C 6B13A22E 59C888F2 BF2A4BEB B4792312 74CB9895   [>ÇXΩ§b  õœlk ¢.Y»àÚø*KÎ¥y# tÀòï
 416 426091C8 0069E2EA 1FB1C18A 8A2F0953 A7D42EAB 4324852A 64AC26DA 6381C262   B`ë» i,Í ±¡ää/ Sß‘.´C$Ö*d¨&/cÅ¬b
 448 A80ACB59 36F25133 B7038E72 278B30A1 B1AE3461 2BE19B13 B5D5F35F 05C487FA   ® ÀY6ÚQ3∑ ér'ã0°±Æ4a+·õ µ’Û_ ƒá˙
 480 952BEAA2 6A17EE84 EA8C2DE6 AD3ACE43 C9C6EC16 496B8206 C40F3E34 780E5558   ï+Í¢j ÓÑÍà-Ê≠:ŒC…∆Ï IkÇ ƒ >4x UX
 512 86437659 64A86441 01397042 640E7DDF 0CDFD0C4 09F3099A C9CE1C68 35ED7DC3   ÜCvYd®dA 9pBd }ﬂ ﬂ–ƒ Û ö…Œ h5Ì}√
 544 086E1291 CDFDCBA4 8F9ED1AE 0F42842F 693E395A 9098F397 4C130D24 0E54A49F    n ëÕ˘À§èû–Æ BÑ/i>9ZêòÛóL   $ T§ü
 576 1326720D 2DC3A623 6152EA8E 2EAF82F1 D66021EA 96D5C237 851A03E3 9510F032    &r -√¶#aRÍé.ØÇÒ÷`!Íñ’¬7Ö  „ï 2
 608 C04471EF 9674083C BD02E9CC 9EF94CEF D1D088F8 BC932B4A 94C915BA 955BA400   ¿DqÔñt <Ω ÈÃû˘LÔ––à¯°ì+Jì… ∫ï[§
 640 E236325C 30B12A16 4984C1D8 07CCDFAF 19E1F04D 652B7A35 E27262BD 9EE3892C   ,62\0±* IÑ¡ÿ ÃﬂØ ·Me+z5,rbΩû„â,
 672 04D8886B E3EFDF12 C7C9422C A438532A 7DF938BF 0D260AAE AA6BB32C 370760DD    ÿàk„Ôﬂ «…B,§8S*}˘8ø & Æ™k≥,7 `›
 704 4C62DDC4 427033DF D5AA7966 F67C13F0 BFA2B8E5 63459E22 A8937BE3 5D9E45F8   Lb›ƒBp3ﬂ’™yfˆ| ø¢∏ÂcEû"®ì{„]ûE¯
 736 20E20F54 A212D112 7BFC2646 93684C02 2458ED97 ED7AEBFA C018F3D9 468CFEBD    , T¢ – {¸&FìhL $XÌóÎzÎ˙¿ ÛŸFà¸Ω
 768 4A211196 8652A57F 804DFCC4 3AB43756 393FDF1C 0ADD2423 2EC9E0BE ED04B769   J! ñÜR• ÄM¸ƒ:¥7V9?ﬂ  ›$#.…‡œÌ ∑i
 800 95580A1D 438723C3 3D4BB052 69049A5F B8596B8E 7F180BA2 FD42DF22 884E7E2E   ïX  Cá#√=K∞Ri ö_∏Ykè    ¢˘Bﬂ"àN~.
 832 7DD0406A 49F32A7F 170E3EDD DE99A181 784C5730 72E77676 659E75F0 54977AC9   }–@jIÛ*    >›ﬁô˚ÅxLW0rÁvveûuTóz…
 864 BE89DFF0 1CA46252 4FA4C128 10EAB3E1 6CDF935E 3DA49276 1F404B42 712713E8   œâﬂ §bRO§¡( Í≥·lﬂì^=§ív @KBq' Ë
 896 E299B3C7 C666526A FC99B2F6 427E3E19 AF767E1B 413A2CC6 9050B71F 21EE4469   ,ô≥«∆fRj¸ô≤ˆB~> Øv~ A:,∆êP∑ !ÓDi
 928 397DD4FB 6F09AA4D F0409547 457D7D04 EEAE26A6 13942AEA 883D5100 C76AD43D   9}‘˙o ™M@ïGE}} ÓÆ&¶ î*Íà=Q «j‘=
 960 3DDDB0BC 6EA835FE EA05F1E6 CAD6EA94 825DE489 B045B0C0 A65B1882 8D331409   =›∞°n®5¸Í ÒÊ ÷ÍìÇ]%â∞E∞¿¶[ Çç3
 992 B1E17081 B55437FB 1BFDE112 955DDF5E 1642B730 61C1B57F ED6CF669 7300FC45   ±·pÅµT7˙ ˇ· ï]ﬂ^ B∑0a¡µ Ìlˆis ¸E
1024 87FFB30D 0DF316EC 17EC9E8C 40B1C161 4308A4F2 49EA6A6F C6A6398C F0A3E76A   á˘≥  Û Ï Ïûà@±¡aC §ÚIÍjo∆¶9à£Åj
1056 2AAC7423 1B2EFD4A F7E41542 958A10E8 57F06006 A0DDAA2E 5CF39B52 0F373575   *¨t# .˝J˝‰ Bïä ËW` †›™.\ÛõR 75u
1088 C287AC87 9EC7EA45 41F93A1A CD71D6A4 41EFFD7A D8F70F1D 1969F075 E7054A2F   ¬á¨áû«ÍEA˘: Õq÷§AÔ˘zÿ˘   iuÁ J/
1120 47357EA4 A2B6DAA9 FB4ADC0B 909B5177 D301376A 6249B931 F37A2B92 4ADA3680   G5~§¢∂/©˙J‹ êõQw” 7jbIπ1Ûz+íJ/6Ä
1152 BB173B8C 637AD3E7 195C0EAB 67E0C5B5 8A282852                              ª ;àcz”Á \ ´g‡≈µä((R
```

5. Once done, save the key to your desktop.

6. Move the key you just created (00000000.dky) to the ***Shared Folder***.

Step 5 – Decrypting Files using the Re-Generated Private Key (Win7-keyRecovery Machine)

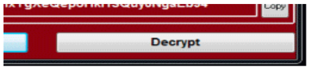
You created your own private key and now it is time to test it on the infected machine!

****

1. Exit the current Windows machine by clicking the Back button in your Browser. Now, click again on the link corresponding with the Win7-keyRecovery machine.

2. ***Move the private key*** (00000000.dky) from the Shared Folder to the ***desktop***.

3. Click on ***Decrypt*** on the message prompting you to pay.



4. Click ***Start***.

5. As a test to make sure it worked, open ***Important.txt*** to make sure you can read the data.

Great job! You've successfully analyzed a binary and decrypted your ransomed files!!!