

# DMA Locker: New Ransomware, But No Reason To Panic

[blog.malwarebytes.com/threat-analysis/2016/02/dma-locker-a-new-ransomware-but-no-reason-to-panic/](http://blog.malwarebytes.com/threat-analysis/2016/02/dma-locker-a-new-ransomware-but-no-reason-to-panic/)

hasherezade

February 3, 2016

## All your personal files are LOCKED!

### WHAT'S HAPPENED?

- \* All your important files(including hard disks, network disks, flash, USB) are encrypted.
- \* All of files are locked with asymmetric algorithm using AES-256 and then RSA-2048 cipher.
- \* You are not possible to unlock your files because all your backups are removed.
- \* Only way to unlock your files is to pay us 536 GBP in Bitcoin currency ( 2.0 BTC ). After payment we will send you decryption key automatically, which allow you to unlock files .

DMA Locker is another ransomware that appeared at the beginning of this year. For now it has been observed to be active only on a small scale (source) – but we just want to warn you that it exists.

### [UPDATE] READ ABOUT THE LATEST VERSION OF DMA LOCKER: 4.0

**UPDATE [4 Feb 2016]:** I apologize to everyone misguided by my rush conclusions about the crypto. After further analysis and consultation with other analysts (special thanks to [@fwosar](#) and [@maciekkotowicz](#)) I confirmed that in reality it is AES in ECB mode. Low entropy was just caused by the fact, that it encrypts separately 16 byte chunks, that are small enough to give this effect. Authors of the malware told many lies in their ransom note, but this one was true, just my mistake. The only way to recover the key is to find the original sample with key included. My goal is always to provide best quality analysis – this time I failed, but I tried to fix it as soon as possible and not let the false information spreading.

## Analyzed samples

- [d35344b1f48764ba083e51438121e6a9](#) – Polish version type 2 (from Jan 2016) <- main focus of this analysis
- [4190df2af81ece296c465e245fc0caea](#) – English version type 2 (from Jan 2016)
- [6fbd3cdcafd6695c384a1119873786aa](#) – Polish version type 1 (from Dec 2015)

*// Special thanks to malware hunters: [@PhysicalDrive0](#) , [@JAMESWT\\_MHT](#) and [@siri\\_urz](#) for their respective help in collecting the samples!*

## Behavioral analysis

When deployed, the ransomware moves itself into **C:\ProgramData** (or **C:\Documents and Settings\All Users\Dokumenty\**), renamed to **fakturax.exe** and drops another, modified copy: **ntserver.exe**. File **faktura.exe** is removed after execution. Depending on its version, it may also drop some other files in the same location.

Name	Date modified	Type	Size
Start menu	2009-07-14 00:00	File folder	
Templates	2009-07-14 06:53	File folder	
cryptinfo.txt	2016-02-02 15:02	Text Document	1 KB
date_1.txt	2016-02-02 15:02	Text Document	1 KB
fakturax.exe	2016-01-28 15:16	Application	96 KB
ntserver.exe	2016-02-02 14:59	Application	96 KB

Symptoms of this ransomware can be recognized by a red window popping up on the screen. So far, it has been observed in two language versions – Polish or English. An example of the English is below:

**DMA Locker**

**All your personal files are LOCKED!**

**WHAT'S HAPPENED?**

- \* All your important files(including hard disks, network disks, flash, USB) are encrypted.
- \* All of files are locked with asymeric algorithm using AES-256 and then RSA-2048 cipher.
- \* You are not possible to unlock your files because all your backups are removed.
- \* Only way to unlock your files is to pay us 536 GBP in Bitcoin currency ( 2.0 BTC ). After payment we will send you decryption key automatically, which allow you to unlock files .

**HOW TO PAY US AND UNLOCK YOUR FILES?**

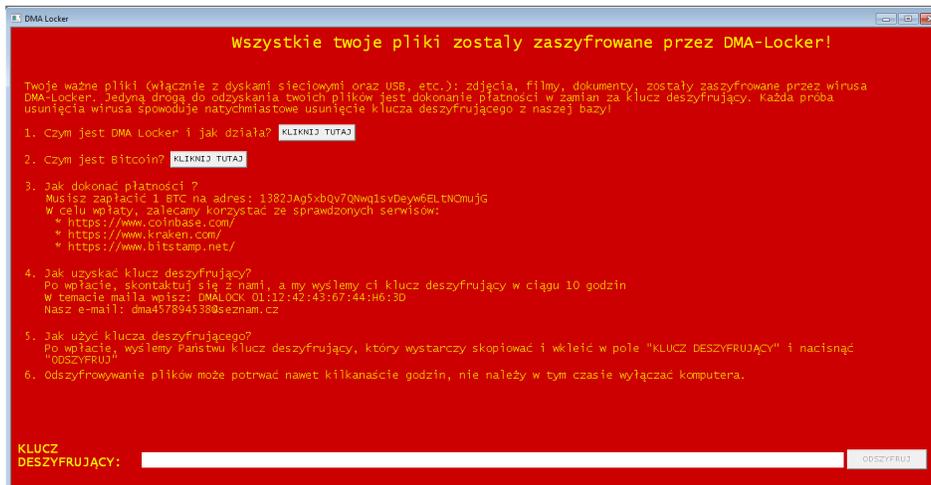
- To pay us, you have to use Bitcoin currency. You can easily buy Bitcoins at following sites:
  - \* <https://www.coinfloor.co.uk>
  - \* <https://www.coinbase.com/>
  - \* <https://www.bitstamp.net/>
- If you already have Bitcoins, pay us 2.0 BTC (536 GBP) on following Bitcoin address:  
**1KXw7aJR4THWAxtnxZYzmysdLXVhLfa97n**
- After payment, necessarily contact with us to get your decryption key:  
**january0030@gmx.com** . In mail title write your unique ID:  
**DMALOCK 49:15:61:11:84:76:67:71**
- We will automatically send you decryption key after bitcoin transfer .  
 When you receive your decryption key, copy and paste it to "DECRYPTION KEY" field  
 Then, press the DECRYPT button to UNLOCK ALL YOUR FILES.

IF FILES UNLOCKING PROCEDURE IS ALREADY WORKING, YOU CAN EASILY TURN OFF YOUR COMPUTER AND CONTINUE FILES UNLOCKING AFTER NEXT STARTUP. TO CONTINUE HEALING YOUR FILES, COPY AND PASTE THE SAME DECRYPTION KEY TO THE "DECRYPTION KEY" FIELD AND PRESS "DECRYPT" BUTTON. THE FILES RECOVERING WILL BE CONTINUED!

**DECRYPTION KEY:**  **DECRYPT**

**3/2/2016 1:57**

Earlier version comes with a bit different GUI (also Polish or English variant):

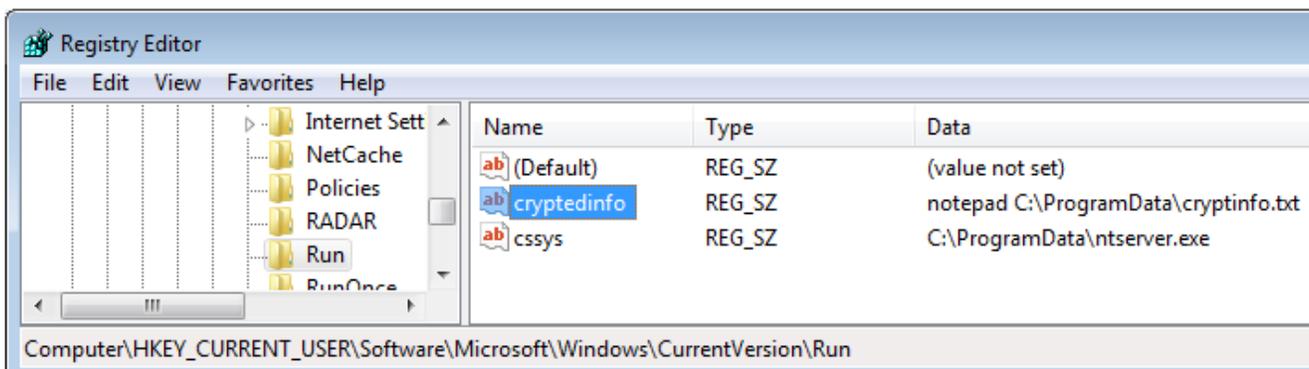


In contrast to other ransomware that are offering a separate decrypter, **DMA Locker** comes with a decrypting feature built-in. It is available from the GUI with ransom note. If the user enters a key (32 characters long) in the text field and clicks the button, the program switches to the decryption mode (using supplied key):



The program is not very stable and may crash during encryption. An older version has been observed to sometimes crash after finishing encryption – but before displaying any info about what happened, which may be very confusing for the victim. What makes things worse is the fact that it does not change file extensions. So, in such a case the only visible symptom will be that the attacked person cannot open some of his/her files.

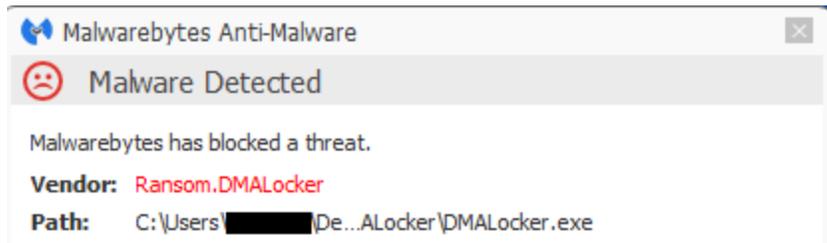
Newer versions also add keys to the autorun. One is to deploy a dropped copy of the program, and the other to display a ransom note in TXT format (via notepad). However, the copy of the program (DMALOCK 41:55:16:13:51:76:67:99ntserver.exe) – is not always dropped successfully and then only the TXT note may be displayed.



## Detection

---

It is detected by Malwarebytes Anti-Malware as **Ransom.DMALocker**:



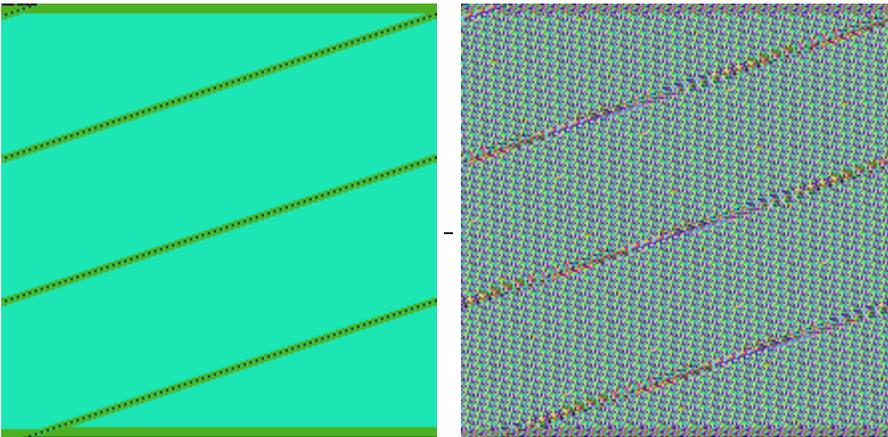
## Experiment

---

In the ransom note, the authors mention that the data is encrypted by **AES and RSA**. Let's look at the files.

After the first look at encrypted content we can see repetitive patterns and entropy is relatively low.

Left – raw bytes of original BMP, right – the same BMP encrypted by DMA Locker:



Let's compare some more files and see how they changed after being encrypted by DMA Locker.

### Example 1 – HTML files:

comparison of original files:

```

C:\Users\tester\Desktop\rav_samples\index.html >> C:\Users\tester\Desktop\rav_samples\modules.html
Compare Next difference Previous difference Font Binary
Edit mode Copy -> Copy <- Undo ANSI<->ANSI
00000: 3C 21 44 4F 43 54 59 50 |<!DOCTYPE
00008: 45 20 48 54 4D 4C 20 50 |E HTML P
00010: 55 42 4C 49 43 20 22 2D |UBLIC "-
00018: 2F 2F 57 33 43 2F 2F 44 |//W3C//D
00020: 54 44 20 48 54 4D 4C 20 |TD HTML
00028: 34 2E 30 31 20 54 72 61 |4.01 Tra
00030: 6E 73 69 74 69 6F 6E 61 |nsitiona
00038: 6C 2F 2F 45 4E 22 3E 0A |l//EN">.
00040: 3C 68 74 6D 6C 3E 3C 68 |<html><h
00048: 65 61 64 3E 3C 6D 65 74 |lead<met
00050: 61 20 68 74 74 70 2D 65 |a http-e
00058: 71 75 69 76 3D 22 43 6F |quiv="Co
00060: 6E 74 65 6E 74 2D 54 79 |ntent-Ty
00068: 70 65 22 20 63 6F 6E 74 |pe" cont
00070: 65 6E 74 3D 22 74 65 78 |ent="tex
00078: 74 2F 68 74 6D 6C 3B 63 |t/html;c
00080: 68 61 72 73 65 74 3D 69 |harset=i
00088: 73 6F 2D 38 38 35 39 2D |so-8859-
00090: 31 22 3E 0A 3C 74 69 74 |l">.<tit
00098: 6C 65 3E 50 69 6E 3A 20 |le>Pin:
000A0: 50 69 6E 20 32 2E 31 33 |Pin 2.13
000A8: 20 55 73 65 72 20 47 75 | User Gu
000B0: 69 64 65 3C 2F 74 69 74 |ide</tit
000B8: 6C 65 3E 0A 3C 6C 69 6E |le>.<lin
000C0: 6B 20 68 72 65 66 3D 22 |k href="
000C8: 64 6F 78 79 67 65 6E 2E |doxygen.
00000: 3C 21 44 4F 43 54 59 50 |<!DOCTYPE
00008: 45 20 48 54 4D 4C 20 50 |E HTML P
00010: 55 42 4C 49 43 20 22 2D |UBLIC "-
00018: 2F 2F 57 33 43 2F 2F 44 |//W3C//D
00020: 54 44 20 48 54 4D 4C 20 |TD HTML
00028: 34 2E 30 31 20 54 72 61 |4.01 Tra
00030: 6E 73 69 74 69 6F 6E 61 |nsitiona
00038: 6C 2F 2F 45 4E 22 3E 0A |l//EN">.
00040: 3C 68 74 6D 6C 3E 3C 68 |<html><h
00048: 65 61 64 3E 3C 6D 65 74 |lead<met
00050: 61 20 68 74 74 70 2D 65 |a http-e
00058: 71 75 69 76 3D 22 43 6F |quiv="Co
00060: 6E 74 65 6E 74 2D 54 79 |ntent-Ty
00068: 70 65 22 20 63 6F 6E 74 |pe" cont
00070: 65 6E 74 3D 22 74 65 78 |ent="tex
00078: 74 2F 68 74 6D 6C 3B 63 |t/html;c
00080: 68 61 72 73 65 74 3D 69 |harset=i
00088: 73 6F 2D 38 38 35 39 2D |so-8859-
00090: 31 22 3E 0A 3C 74 69 74 |l">.<tit
00098: 6C 65 3E 50 69 6E 3A 20 |le>Pin:
000A0: 4D 6F 64 75 6C 65 20 49 |Module I
000A8: 6E 64 65 78 3C 2F 74 69 |index</ti
000B0: 74 6C 65 3E 0A 3C 6C 69 |tle>.<li
000B8: 6E 6B 20 68 72 65 66 3D |nk href=
000C0: 22 64 6F 78 79 67 65 6E |"doxygen
000C8: 2E 63 73 73 22 20 72 65 |.css" re

```

comparison of the same files encrypted:

```

C:\Users\tester\Desktop\encrypted_samples\index.html >> C:\Users\tester\Desktop\encrypted_samples\modules.html
Compare Next difference Previous difference Font Binary
Edit mode Copy -> Copy <- Undo ANSI<->ANSI
00000: 04 90 4E E3 E7 38 A3 30 |.Năç8Ł0
00008: BF DE 4A 71 3B 7E 4A EB |žTJq;~Jē
00010: 0E E8 AE 91 44 44 34 76 |.čö'DD4v
00018: 28 61 C5 A6 A6 E5 D1 53 |(aL!;!ÍNS
00020: E3 F9 85 08 A6 EC FE 55 |šš...!ětU
00028: B2 FE 12 71 56 7A 32 1A |.č. qVz2.
00030: 38 16 4C 69 16 0F D7 1A |8.Li...x.
00038: D4 D7 48 E3 0D 6C 8B 70 |Ō×Hă.l<p
00040: 83 A4 8C 26 A0 82 A9 39 |xšš ,@9
00048: 31 08 D5 2E 51 02 D2 B4 |l.Ō.Q.Ň'
00050: 87 66 3E C7 69 1C 9D CF |žž>či.tĎ
00058: CE AC EC 58 D2 2E 2C 77 |Ī-ěXŇ.,w
00060: D0 B9 90 1B 71 B3 E4 29 |Đa.qšā)
00068: BB 3A 36 38 D5 B1 99 90 |»:68Ō±™
00070: 16 EA AA 44 F3 EC BB 37 |.ešDóě»7
00078: 17 D5 7B 59 0D F1 51 14 |.Ō{Y.nQ.
00080: 15 7C 31 16 E3 21 1A F7 |.|l.ă!.+
00088: 76 4C 21 BD 7C E8 74 DD |vL!"|čtÝ
00090: 09 F9 E0 94 27 03 27 F8 |.šž".'.ž
00098: 1D 80 DD 4F 86 71 CC 6D |.čÝO+qšm
000A0: 61 DC BF 6A 23 AE D5 B7 |aŮžž±ŌŌ.
000A8: 77 E8 AB E3 6F 4C 3B 13 |wč«ăoL;.
000B0: 8B 70 83 AD 72 8F C6 72 |<p-ržčr
000B8: 49 03 6E 36 A1 49 D0 EF |I.n6~IĐđ
000C0: 66 C1 EB E8 F5 36 E8 4F |žĂăčš6čŌ
000C8: D0 A6 5B 66 55 C4 D6 6B |Đ;[fUĂŌk
00000: 04 90 4E E3 E7 38 A3 30 |.Năç8Ł0
00008: BF DE 4A 71 3B 7E 4A EB |žTJq;~Jē
00010: 0E E8 AE 91 44 44 34 76 |.čö'DD4v
00018: 28 61 C5 A6 A6 E5 D1 53 |(aL!;!ÍNS
00020: E3 F9 85 08 A6 EC FE 55 |šš...!ětU
00028: B2 FE 12 71 56 7A 32 1A |.č. qVz2.
00030: 38 16 4C 69 16 0F D7 1A |8.Li...x.
00038: D4 D7 48 E3 0D 6C 8B 70 |Ō×Hă.l<p
00040: 83 A4 8C 26 A0 82 A9 39 |xšš ,@9
00048: 31 08 D5 2E 51 02 D2 B4 |l.Ō.Q.Ň'
00050: 87 66 3E C7 69 1C 9D CF |žž>či.tĎ
00058: CE AC EC 58 D2 2E 2C 77 |Ī-ěXŇ.,w
00060: D0 B9 90 1B 71 B3 E4 29 |Đa.qšā)
00068: BB 3A 36 38 D5 B1 99 90 |»:68Ō±™
00070: 16 EA AA 44 F3 EC BB 37 |.ešDóě»7
00078: 17 D5 7B 59 0D F1 51 14 |.Ō{Y.nQ.
00080: 15 7C 31 16 E3 21 1A F7 |.|l.ă!.+
00088: 76 4C 21 BD 7C E8 74 DD |vL!"|čtÝ
00090: 09 F9 E0 94 27 03 27 F8 |.šž".'.ž
00098: 1D 80 DD 4F 86 71 CC 6D |.čÝO+qšm
000A0: BB CB D3 11 51 37 75 3D |<ĒŌ.Q7u=
000A8: 41 0C E2 3B 26 2E 43 8D |A.ă;š.CĪ
000B0: A4 94 99 55 92 3D AF 68 |x™™U'=.žh
000B8: C5 E6 53 67 1F 23 01 37 |ÍčSg.ž.7
000C0: 0F 94 1B 0F CE 7F 8E 45 |.".ŮŮĒE
000C8: D9 8A A5 DF 1F 27 E2 A2 |ŮŠĂB. 'ă'

```

### Example 2 – PNG files:

comparison of original files:

comparison of the same files encrypted:

As we can see, when the beginnings of original files are identical, the beginnings of encrypted outputs also are. But it seems that encryption is done in some chunks – possibly 8 or 16 bytes at once. Look at the comparison of PNG files – from 0x10 they have been encrypted differently – although they both have zeros at positions 0x10, 0x11...

## Inside

This ransomware is distributed without any packing and no defense against analysis has been observed. All the used strings and called API functions are in plain text. In fact, the malware even “helps” the analyst by providing a lot of debug strings describing all its activities (original + translation):

```

[+] Plik jest aktualnie zaszyfrowany, pomijanie.. //The file is already encrypted,
skipping..
[*] Rozmiar pliku = %I64d bajtow.. //File size = %I64d bytes..
[+] Rozpoczeto szyfrowanie pliku: %s //Started encrypting the file: %s
[+] Zakonczone szyfrowanie pliku: %s //Finished encrypting the file: %s
[+] Rozpoczeto zapisywanie z pamieci do pliku: %s //Started dumping from memory to a
file: %s
[+] Zakonczone zapisywanie z pamieci do pliku: %s //Finished dumping from memory to a
file: %s
[*] Plik jest aktualnie odszyfrowany, pomijanie.. //The file is already decrypted,
skipping..
[+] Rozpoczeto deszyfrowanie pliku: %s //Started decrypting file: %s
[+] Zakonczone deszyfrowanie pliku: %s //Finished decrypting file: %s
Alokacja, error: %d //Allocation error: %d
DMA Locker
Otwieranie pliku: %d //Opening file: %d

```

Thanks to the logs, finding important part of the code is trivial!

At the beginning of the execution a new thread is deployed – whose role is to check for the presence of following processes:

- rstrui.exe
- ShadowExplorer.exe
- sesvc.exe
- cbengine.exe

If any of them is detected, malware tries to terminate it. Just after deploying this thread malware logs (in Polish):

“[+] *Blocking processes of system recovery*”

```

004043AD call    init_filenames
004043B2 call    drop_ransom_note_txt
004043B7 xor     ebx, ebx
004043B9 push   ebx                ; lpThreadId
004043BA push   ebx                ; dwCreationFlags
004043BB push   ebx                ; lpParameter
004043BC push   offset terminate_blacklisted_processes ; lpStartAddress
004043C1 push   ebx                ; dwStackSize
004043C2 push   ebx                ; lpThreadAttributes
004043C3 call    ds:CreateThread
004043C9 test   eax, eax
004043CB jz     short skip_logging

```

```

004043CD push   offset aBlokowanieProc ; "[+] Blokowanie procesow przywracania s"...
004043D2 call    printf

```

Instead of a list of attacked extensions, this malware contains two blacklists. One for directories:

```

004025A3 sub     esp, 2Ch
004025A6 push   esi
004025A7 mov     [ebp+var_2C], offset aWindows ; "\\Windows\\"
004025AE mov     [ebp+var_28], offset aWindows_0 ; "\\WINDOWS\\"
004025B5 mov     [ebp+var_24], offset aProgramFiles ; "\\Program Files\\"
004025BC mov     [ebp+var_20], offset aProgramFilesX8 ; "\\Program Files (x86)\\"
004025C3 mov     [ebp+var_1C], offset aGames ; "Games"
004025CA mov     [ebp+var_18], offset aTemp ; "\\Temp"
004025D1 mov     [ebp+var_14], offset aSamplePictures ; "\\Sample Pictures"
004025D8 mov     [ebp+var_10], offset aSampleMusic ; "\\Sample Music"
004025DF mov     [ebp+var_C], offset aCache ; "\\cache"
004025E6 mov     [ebp+var_8], offset aCache_0 ; "\\Cache"
004025ED xor     esi, esi

```

and another for file extensions:

```

00402627 mov     [ebp+var_30], offset a_exe ; ".exe"
0040262E mov     [ebp+var_2C], offset a_msi ; ".msi"
00402635 mov     [ebp+var_28], offset a_dll ; ".dll"
0040263C mov     [ebp+var_24], offset a_pif ; ".pif"
00402643 mov     [ebp+var_20], offset a_scr ; ".scr"
0040264A mov     [ebp+var_1C], offset a_sys ; ".sys"
00402651 mov     [ebp+var_18], offset a_msp ; ".msp"
00402658 mov     [ebp+var_14], offset a_com ; ".com"
0040265F mov     [ebp+var_10], offset a_lnk ; ".lnk"
00402666 mov     [ebp+var_C], offset a_hta ; ".hta"
0040266D mov     [ebp+var_8], offset a_cpl ; ".cpl"
00402674 mov     [ebp+var_4], offset a_msc ; ".msc"

```

Files that contain in their path blacklisted substrings are skipped.

Malware enumerates all the files – browsing first logical drives, after that network resources – trying to encrypt each and every file (except the blacklisted)

```

00404640 add     esp, 0Ch
00404643 lea    edx, [esp+90h+key_buf]
00404647 push   ebx
00404648 push   edx
00404649 call   enc_dec_logical_drives
0040464E add     esp, 8
00404651 push   ebx ; int
00404652 lea    eax, [esp+94h+key_buf]
00404656 push   eax ; int
00404657 push   ebx ; lpNetResource
00404658 call   enc_dec_net_resources
0040465D mov     ebx, ds:$leep
00404663 mov     esi, 0Ah
00404668 inn     char loc_404670

```

A single flag decides whether the malware is in encryption or decryption mode:

```

00402BA0 nov     edx, edi
00402BA2 push    edx           ; char *
00402BA3 call   _puts
00402BA8 add     esp, 4
00402BAB cmp     [ebp+mode_flag], 0
00402BAF jnz    short decrypt_mode

00402BB1 push    offset aFlaga0Szyfrowa ; "[+] Flaga = 0, szyfrowanie\n"
00402BB6 call   _printf
00402BB8 mov     eax, [ebp+cryptoKey]
00402BC1 add     esp, 4
00402BC4 push    eax           ; int
00402BC5 mov     ecx, edi       ; filename
00402BC7 push    ecx           ; char *
00402BC8 call   encrypt_file
00402BCD jmp     short loc_402C02

00402BCF decrypt_mode: ; "[+] Flaga = 1, deszyfrowanie\n"
00402BCF push    offset aFlaga1Deszyfro
00402BD4 call   _printf
00402BD9 mov     edx, [ebp+cryptoKey]
00402BDF add     esp, 4
00402BE2 push    edx           ; int
00402BE3 lea   eax, [ebp+filename]
00402BE9 push    eax           ; char *
00402BEA call   decrypt_file
00402BEF jmp     short loc_402C02

```

Encryption (as well as decryption) is deployed in a new thread:

```

00401E50 push    offset aRozpoczetoSzyf ; "[+] Rozpoczeto szyfrowanie pliku: %s\n"
00401E55 mov     [ebp+ThreadId+3], ebx
00401E58 call   _printf
00401E5D add     esp, 8
00401E60 mov     [ebp+var_1A0], ebx
00401E66 cmp     [ebp+nCount], ebx
00401E6C jle    short loc_401EDC

00401E6E jmp     short loc_401E76

00401E76
00401E76 loc_401E76: ; size_t
00401E76 push    30h
00401E78 call   _malloc
00401E7D mov     ecx, [ebp+var_198]
00401E83 mov     [eax+8], ecx
00401E86 lea   edx, [ebx+esi]
00401E89 mov     esi, [ebp+key]
00401E8C add     esp, 4
00401E8F lea   edi, [eax+10h]
00401E92 mov     [eax], edx
00401E94 mov     [eax+0Ch], ebx
00401E97 mov     ecx, 8
00401E9C rep movsd
00401E9E lea   ecx, [ebp+ThreadId+3]
00401EA1 push    ecx           ; lpThreadId
00401EA2 push    0             ; dwCreationFlags
00401EA4 push    eax           ; lpParameter
00401EA5 push    offset encrypting_thread ; lpStartAddress
00401EAA push    0             ; dwStackSize
00401EAC push    0             ; lpThreadAttributes
00401EAE call   ds:CreateThread

```

## Encryption key

The encryption key is 32 byte long. In newer version of the malware it is hard-coded at the end of the original file, and then read. However, there is a twist.

During execution, two copies of the original file are dropped: **fakturax.exe** and **ntserver.exe** – but only **fakturax.exe** contains the key – **ntserver.exe** have it cleaned. After reading the key, **fakturax.exe** is removed and the key is lost along with it. That's why, we can easily



```

C *G.P.U* - thread 00000AE8, module fakturax
004019CA . MOV DWORD PTR SS:[ESP+0xC], EAX
004019CE . MOV EDI, EDI
004019D0 > MOV EAX, DWORD PTR DS:[EBX]
004019D2 . MOV ECX, DWORD PTR DS:[EAX+EDI]
004019D5 . MOV EDX, DWORD PTR DS:[EAX+EDI+0x4]
004019D9 . ADD EAX, EDI
004019DB . MOV DWORD PTR SS:[ESP+0x74], ECX
004019DF . MOV ECX, DWORD PTR DS:[EAX+0x8]
004019E2 . MOV DWORD PTR SS:[ESP+0x78], EDX
004019E6 . MOV EDX, DWORD PTR DS:[EAX+0xC]
004019E9 . MOV DWORD PTR SS:[ESP+0x7C], ECX
004019ED . LEA EAX, DWORD PTR DS:[EBX+0x10]
004019F0 . LEA ECX, DWORD PTR SS:[ESP+0x10]
004019F4 . MOV DWORD PTR SS:[ESP+0x80], EDX
004019FB . CALL fakturax.004013B0
00401A00 . MOV EAX, ECX
00401A02 . PUSH EAX
00401A03 . LEA ESI, DWORD PTR SS:[ESP+0x78]
00401A07 . CALL fakturax.004014D0
00401A0C . ADD ESP, 0x4
00401A0F . LEA EAX, DWORD PTR SS:[ESP+0x30]
00401A13 . MOV ECX, 0x20
00401A18 . JMP SHORT fakturax.00401A20

```

Address	Hex dump	ASCII
0224FF10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
0224FF20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
0224FF30	31 31 31 31 31 31 31 31 31 31 31 31 31 31 31	1111111111111111
0224FF40	31 31 31 31 31 31 31 31 31 31 31 31 31 31 31	1111111111111111
0224FF50	07 F9 15 B3 01 AA F8 E6 BC 38 D9 55 AA FF 5C 2E	..s 0 °s0-U \.
0224FF60	43 38 82 76 B1 60 EC 0A 6E 11 79 58 67 5A 63 24	CS.v±'e.n(yXg2
0224FF70	00 00 00 00 00 00 4E 47 0D 0A 14 0A 00 00 00	....ePhg...+
0224FF80	49 48 44 52 D5 0A FE A3 94 FF 24 02 45 3C 03 77	IHDRN.µüö \$0E<w

after encryption (output marked gray):

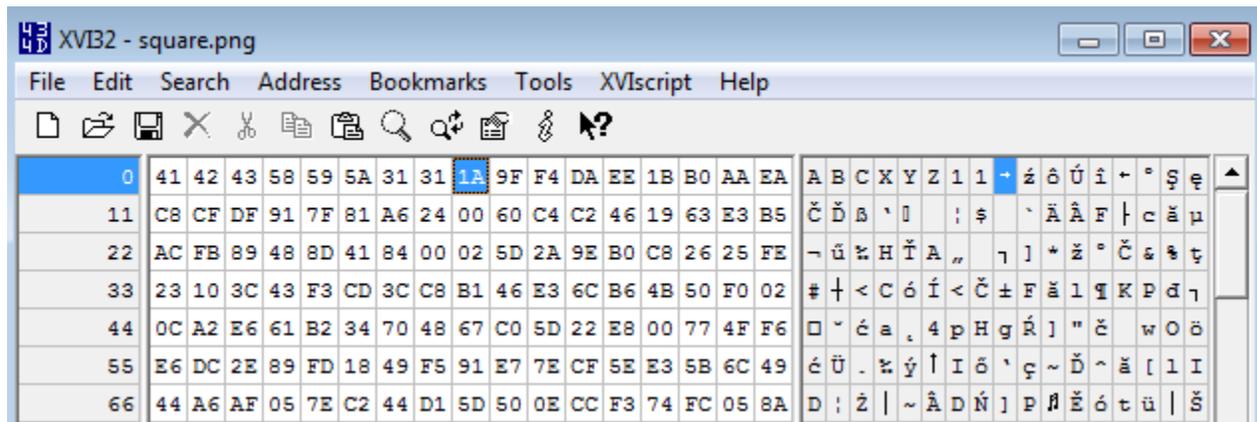
```

004019CA . MOV DWORD PTR SS:[ESP+0xC], EAX
004019CE . MOV EDI, EDI
004019D0 > MOV EAX, DWORD PTR DS:[EBX]
004019D2 . MOV ECX, DWORD PTR DS:[EAX+EDI]
004019D5 . MOV EDX, DWORD PTR DS:[EAX+EDI+0x4]
004019D9 . ADD EAX, EDI
004019DB . MOV DWORD PTR SS:[ESP+0x74], ECX
004019DF . MOV ECX, DWORD PTR DS:[EAX+0x8]
004019E2 . MOV DWORD PTR SS:[ESP+0x78], EDX
004019E6 . MOV EDX, DWORD PTR DS:[EAX+0xC]
004019E9 . MOV DWORD PTR SS:[ESP+0x7C], ECX
004019ED . LEA EAX, DWORD PTR DS:[EBX+0x10]
004019F0 . LEA ECX, DWORD PTR SS:[ESP+0x10]
004019F4 . MOV DWORD PTR SS:[ESP+0x80], EDX
004019FB . CALL fakturax.004013B0
00401A00 . MOV EAX, ECX
00401A02 . PUSH EAX
00401A03 . LEA ESI, DWORD PTR SS:[ESP+0x78]
00401A07 . CALL fakturax.004014D0
00401A0C . ADD ESP, 0x4
00401A0F . LEA EAX, DWORD PTR SS:[ESP+0x30]
00401A13 . MOV ECX, 0x20
00401A18 . JMP SHORT fakturax.00401A20

```

Address	Hex dump	ASCII
0224FF10	07 F9 15 B3 01 AA F8 E6 BC 38 D9 55 AA FF 5C 2E	..s 0 °s0-U \.
0224FF20	43 38 82 76 B1 60 EC 0A 6E 11 79 58 67 5A 63 24	CS.v±'e.n(yXg2
0224FF30	31 31 31 31 31 31 31 31 31 31 31 31 31 31 31	1111111111111111
0224FF40	31 31 31 31 31 31 31 31 31 31 31 31 31 31 31	1111111111111111
0224FF50	07 F9 15 B3 01 AA F8 E6 BC 38 D9 55 AA FF 5C 2E	..s 0 °s0-U \.
0224FF60	43 38 82 76 B1 60 EC 0A 6E 11 79 58 67 5A 63 24	CS.v±'e.n(yXg2
0224FF70	00 00 00 00 00 00 1A 9F F4 DA EE 1B 80 AA EA C8 CF DF	....+0°rt±:±0E
0224FF80	91 7F 81 A6 D5 0A FE A3 94 FF 24 02 45 3C 03 77	LO2R.µüö \$0E<w
0224FF90	78 4E 46 01 D4 FF 24 02 F5 37 58 77 78 4E 46 01	xNF0d' \$037XwxNF0
0224FFA0	83 F1 6B 76 00 00 00 00 00 00 00 00 00 00 00	ä'kv.....xNF0

output is then copied back to the original buffer, containing the full file. Every encrypted file has a content prefixed by “ABCXYZ11” – a magic value, used by the ransomware to recognize encrypted files (it has been introduced in the newer version). Below, we can see the sample file after being dumped on the disk.



16 byte long chunks of file are encrypted by AES in ECB mode.

## Conclusion

First of all, not all what malware authors tell is true. In this case the key was neither RSA encrypted, nor randomly generated – just stored in the original file.

Second – immediately removing the malware is not always the best solution – sometimes we may need it to recover the data.

If you encountered a ransomware, it is better to try to gather information about it before taking any steps. In case you cannot find any information, the best way is to make a topic on the forum of your favorite vendor or contact some known analyst. We are in a constant search of samples of new threats, trying to describe and solve the problems.

And remember: only some families are really nasty. Other, like i.e LeChiffre have implementation flaws allowing to recover files.

## Appendix

[https://forum.4programmers.net/Hardware\\_Software/264028-dma\\_locker\\_-\\_zaszyfrowane\\_pliki](https://forum.4programmers.net/Hardware_Software/264028-dma_locker_-_zaszyfrowane_pliki) – a thread on a Polish forum, created by a user infected by DMA Locker