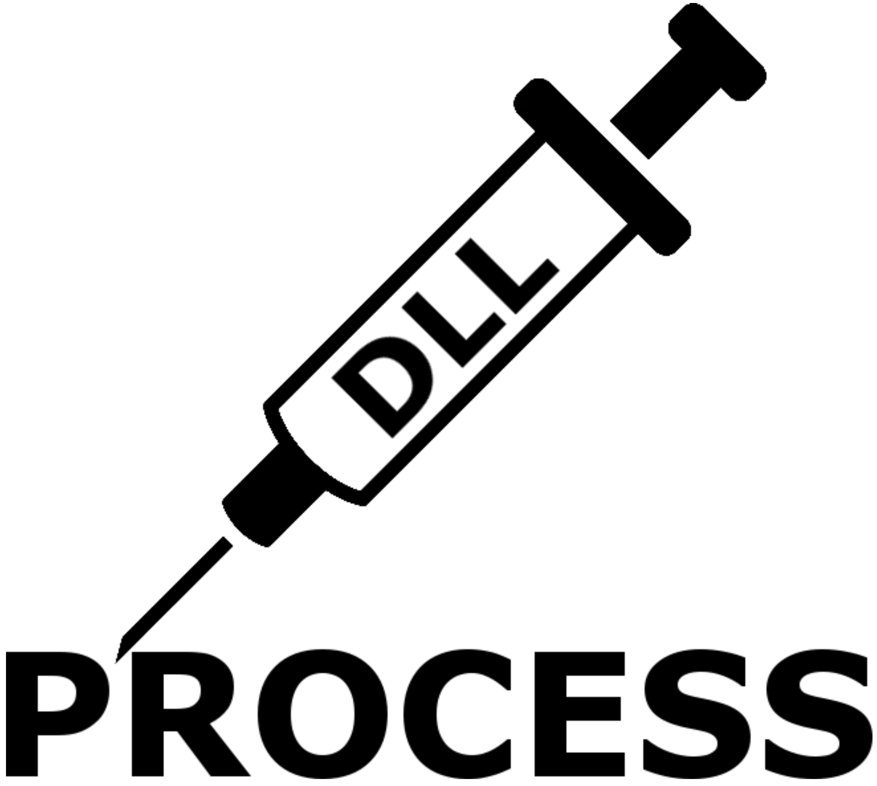
Secure an Editor

Yaakov Cohen

DLL INJECTION



DLL INJECTION

- We will discuss about two methods:
  - AppInit_DLLs registry key.
  - CreateRemoteThread.

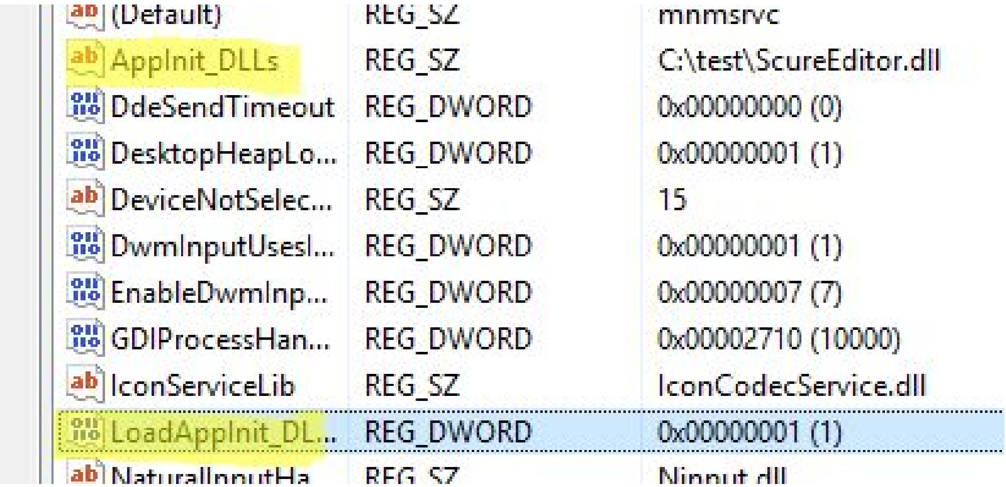
AppInit_DLLs registry key

- Legitimate way to do DLL Injection.
- Documented by Microsoft.
- Will inject the DLL to all processes that load user32.dll (almost all processes in Windows).
- Must be done by Admin user.

"Today, only a small set of legitimate applications use this mechanism. Unfortunately, a larger set of malware use this mechanism. Applications and malicious software both use AppInit DLLs for the same basic reason, which is to hook APIs”
Microsoft.

AppInit_DLLs registry key

- Add the DLL path to the registry key.
- The key path is diffrent for x86 or x64 Windows OS and by the dll build architecture.

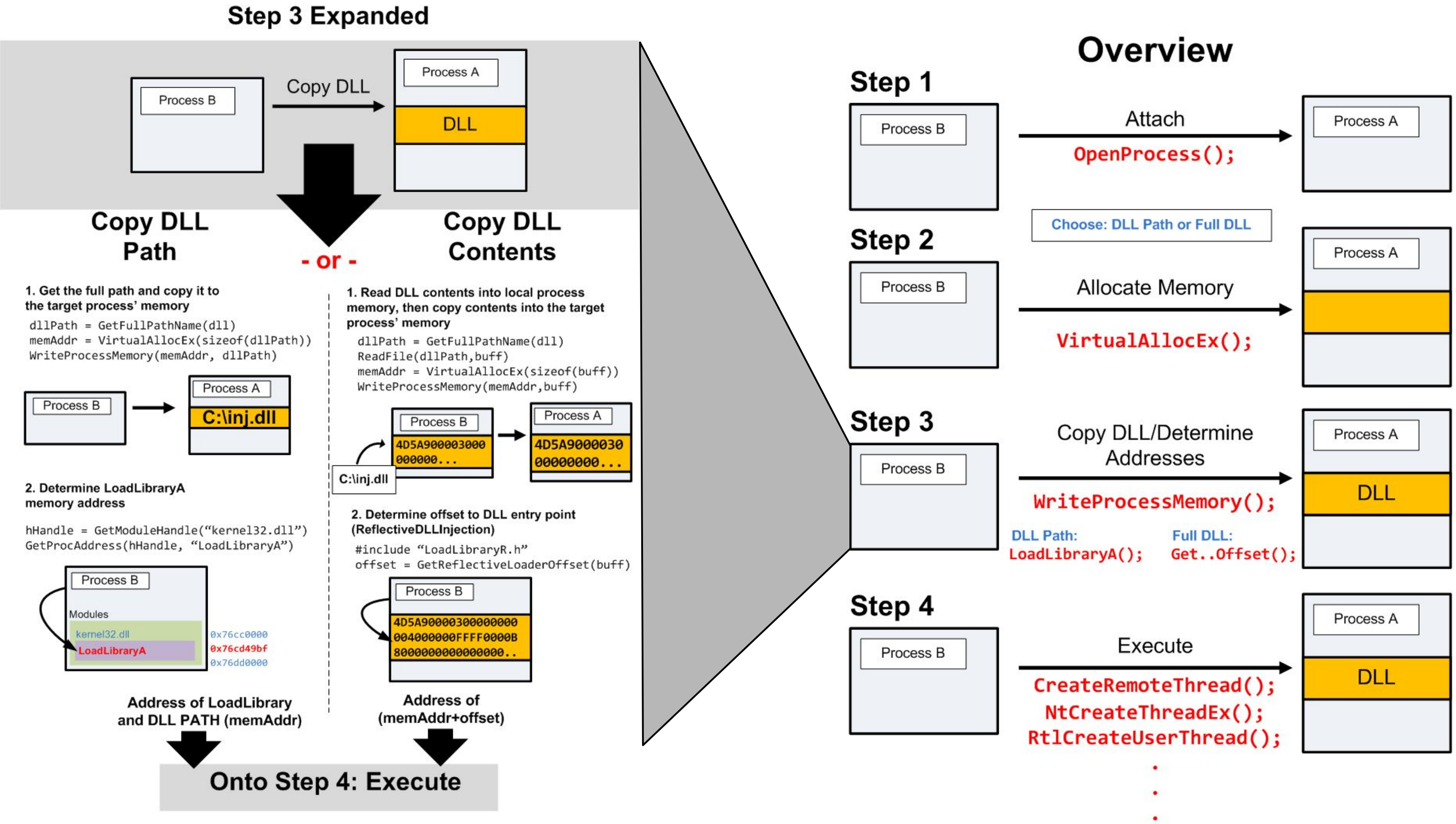


(Default)	REG_SZ	mnmsrvc
AppInit_DLLs	REG_SZ	C:\test\ScureEditor.dll
DdeSendTimeout	REG_DWORD	0x00000000 (0)
DesktopHeapLo...	REG_DWORD	0x00000001 (1)
DeviceNotSelec...	REG_SZ	15
DwmInputUsesI...	REG_DWORD	0x00000001 (1)
EnableDwmInp...	REG_DWORD	0x00000007 (7)
GDIProcessHan...	REG_DWORD	0x00002710 (10000)
IconServiceLib	REG_SZ	IconCodecService.dll
LoadAppInit_DL...	REG_DWORD	0x00000001 (1)
NaturalInputHa	REG_SZ	Ninput.dll

CreateRemoteThread

- Not need Admin Privileges.
- The user can decide to use it or not.
- Inject the dll by WINAPI function.
- This method works only when the process is already running.

CreateRemoteThread



Step 3 Expanded

Copy DLL Path

- or -

1. Get the full path and copy it to the target process' memory

```
dllPath = GetFullPathName(dll)
memAddr = VirtualAllocEx(sizeof(dllPath))
WriteProcessMemory(memAddr, dllPath)
```

2. Determine LoadLibraryA memory address

```
hHandle = GetModuleHandle("kernel32.dll")
GetProcAddress(hHandle, "LoadLibraryA")
```

Address of LoadLibrary and DLL PATH (memAddr)

Copy DLL Contents

1. Read DLL contents into local process memory, then copy contents into the target process' memory

```
dllPath = GetFullPathName(dll)
ReadFile(dllPath,buff)
memAddr = VirtualAllocEx(sizeof(buff))
WriteProcessMemory(memAddr,buff)
```

2. Determine offset to DLL entry point (ReflectiveDLLInjection)

```
#include "LoadLibraryR.h"
offset = GetReflectiveLoaderOffset(buff)
```

Address of (memAddr+offset)

Onto Step 4: Execute

Overview

Step 1

Attach

`OpenProcess();`

Step 2

Choose: DLL Path or Full DLL

Allocate Memory

`VirtualAllocEx();`

Step 3

Copy DLL/Determine Addresses

`WriteProcessMemory();`

DLL Path: `LoadLibraryA();`

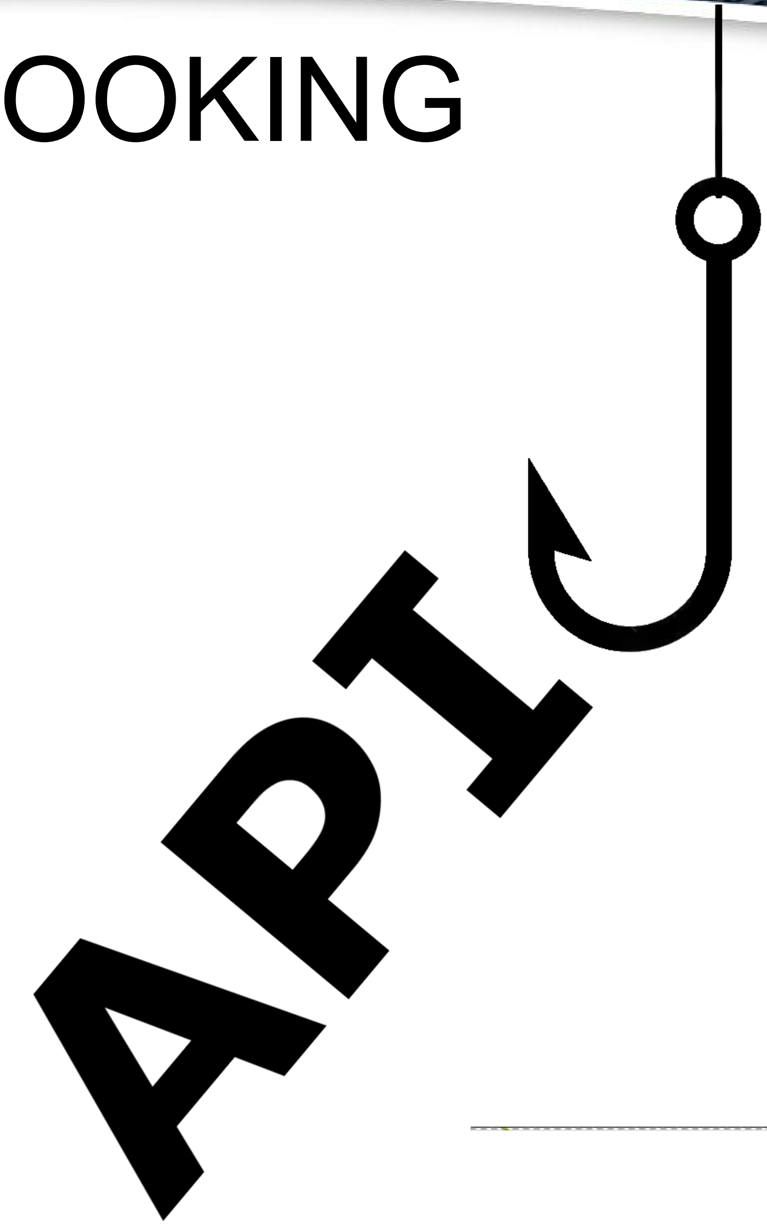
Full DLL: `Get..Offset();`

Step 4

Execute

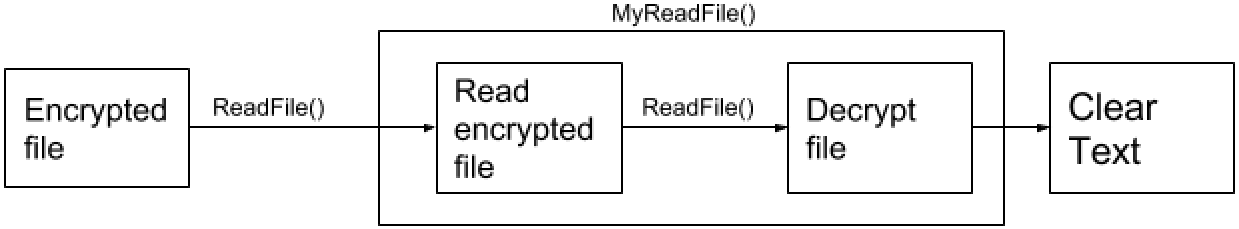
```
CreateRemoteThread();
NtCreateThreadEx();
RtlCreateUserThread();
.
.
.
```

API HOOKING



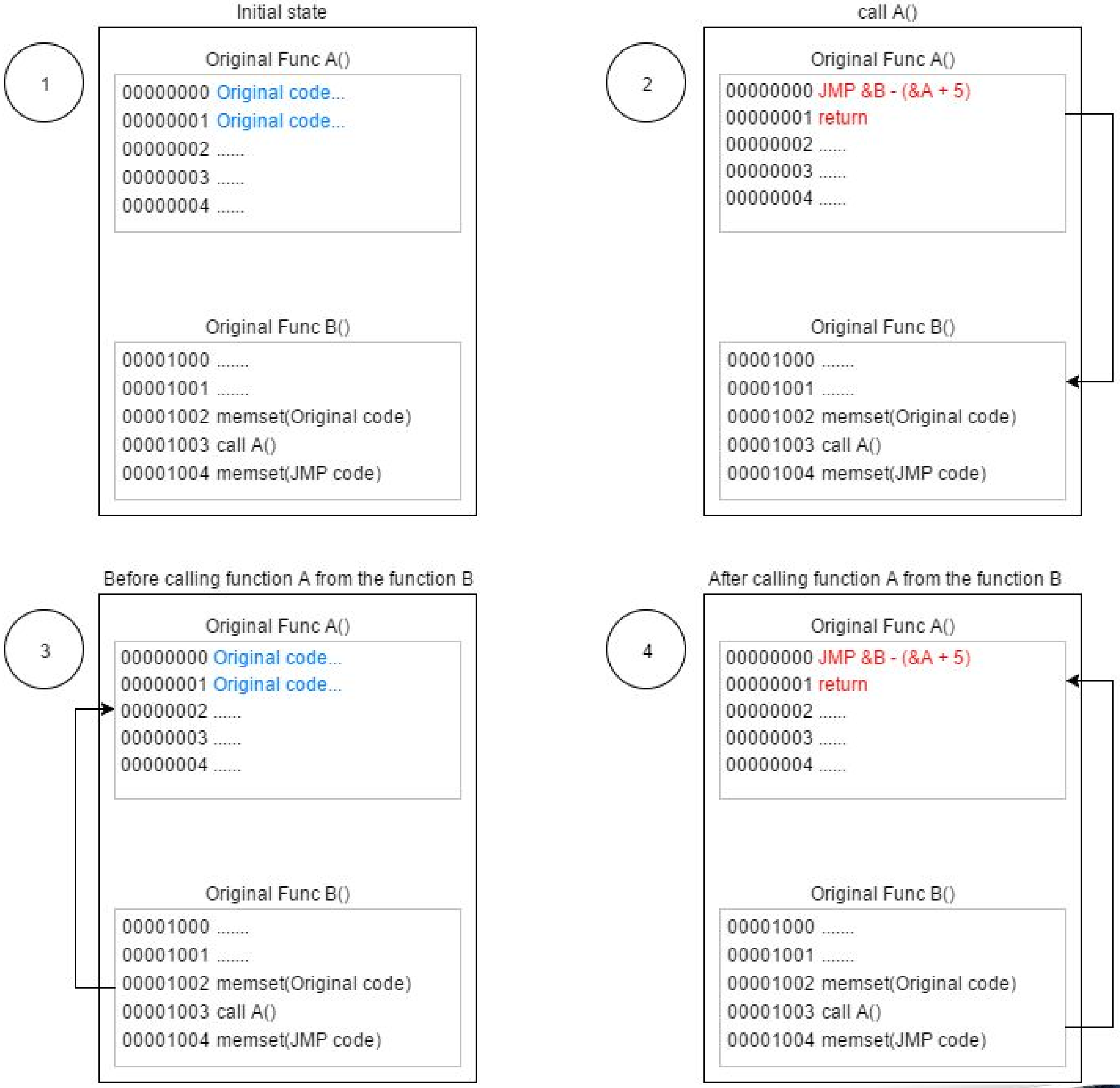
API HOOKING

- Alter or augment the behavior of software components by intercepting function calls.
- We want to hook all the read and write calls.
- For example WINAPI ReadFile func:



API Hooking Using JMP Instruction

- Because of the calling convention if we don't change the stack we can get all the func's args just by declaring the same function prototype.



1

Initial state

Original Func A()

```
00000000 Original code...
00000001 Original code...
00000002 ......
00000003 ......
00000004 ......
```

Original Func B()

```
00001000 .......
00001001 .......
00001002 memset(Original code)
00001003 call A()
00001004 memset(JMP code)
```

2

call A()

Original Func A()

```
00000000 JMP &B - (&A + 5)
00000001 return
00000002 ......
00000003 ......
00000004 ......
```

Original Func B()

```
00001000 .......
00001001 .......
00001002 memset(Original code)
00001003 call A()
00001004 memset(JMP code)
```

3

Before calling function A from the function B

Original Func A()

```
00000000 Original code...
00000001 Original code...
00000002 ......
00000003 ......
00000004 ......
```

Original Func B()

```
00001000 .......
00001001 .......
00001002 memset(Original code)
00001003 call A()
00001004 memset(JMP code)
```

4

After calling function A from the function B

Original Func A()

```
00000000 JMP &B - (&A + 5)
00000001 return
00000002 ......
00000003 ......
00000004 ......
```

Original Func B()

```
00001000 .......
00001001 .......
00001002 memset(Original code)
00001003 call A()
00001004 memset(JMP code)
```

API Hooking Using JMP Instruction

- JMP Instruction on Multithreading process:
  - Race condition.
  - Critical section.
- Helpful C++ tools:
  - Mutex.
  - Recursive mutex.
  - Lockguard.

Encryption Mechanism

Requirements :

- User friendly.
- Create unique key for each file (Security in depth).
- Can be opened by authorized third-party (Transfer files, Backup, IT Department).

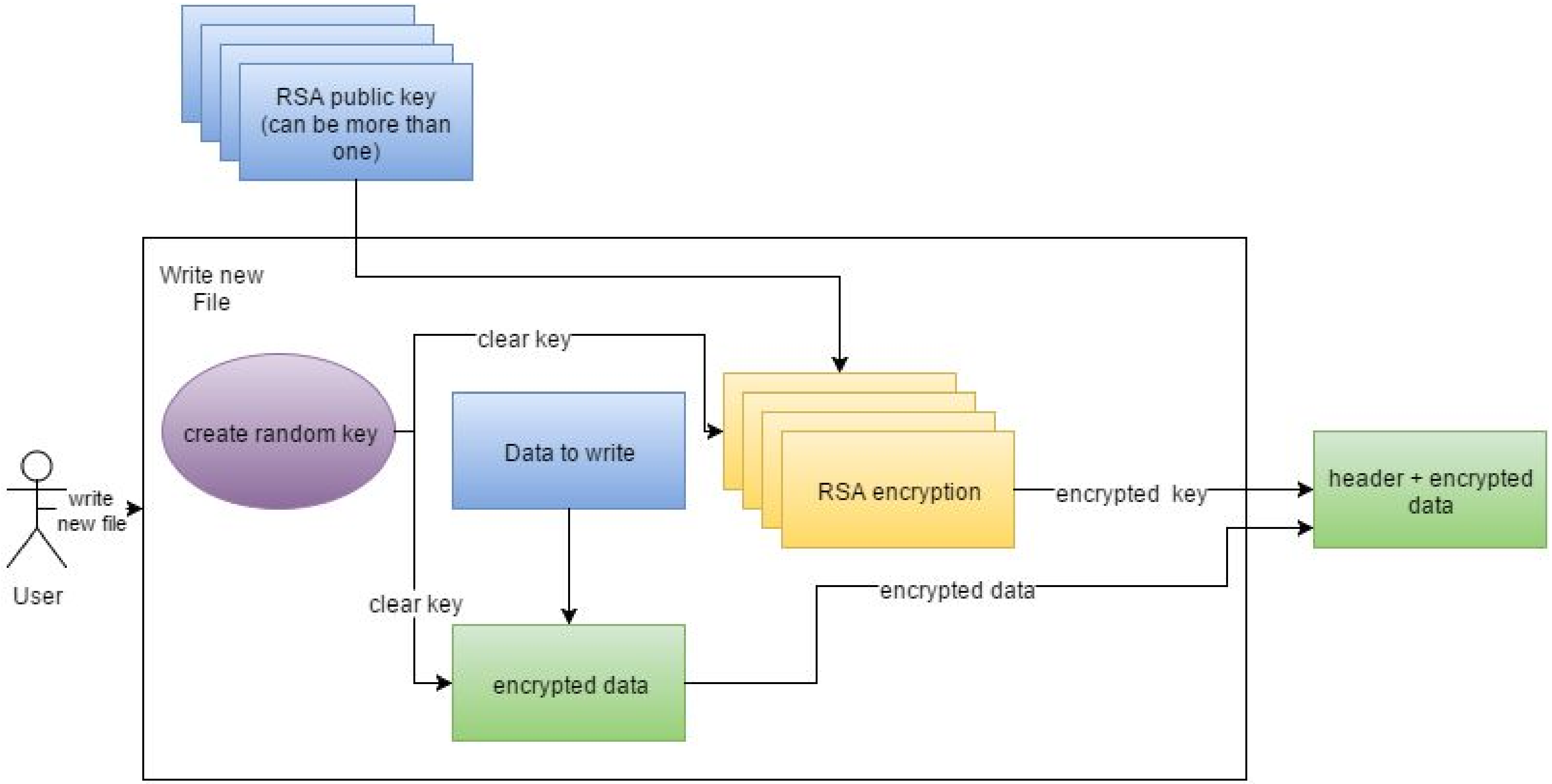
Key Hierarchy

Key Hierarchy

- A-Symetric Key for each user (Public and Private).
- Symetric Key for each file.
- Encrypt the file by the Symetric Key.
- Encrypt the Symetric Key by the user's Public Key.
- The Private key will be encrypted by symetric algorithm with the user's password as key.

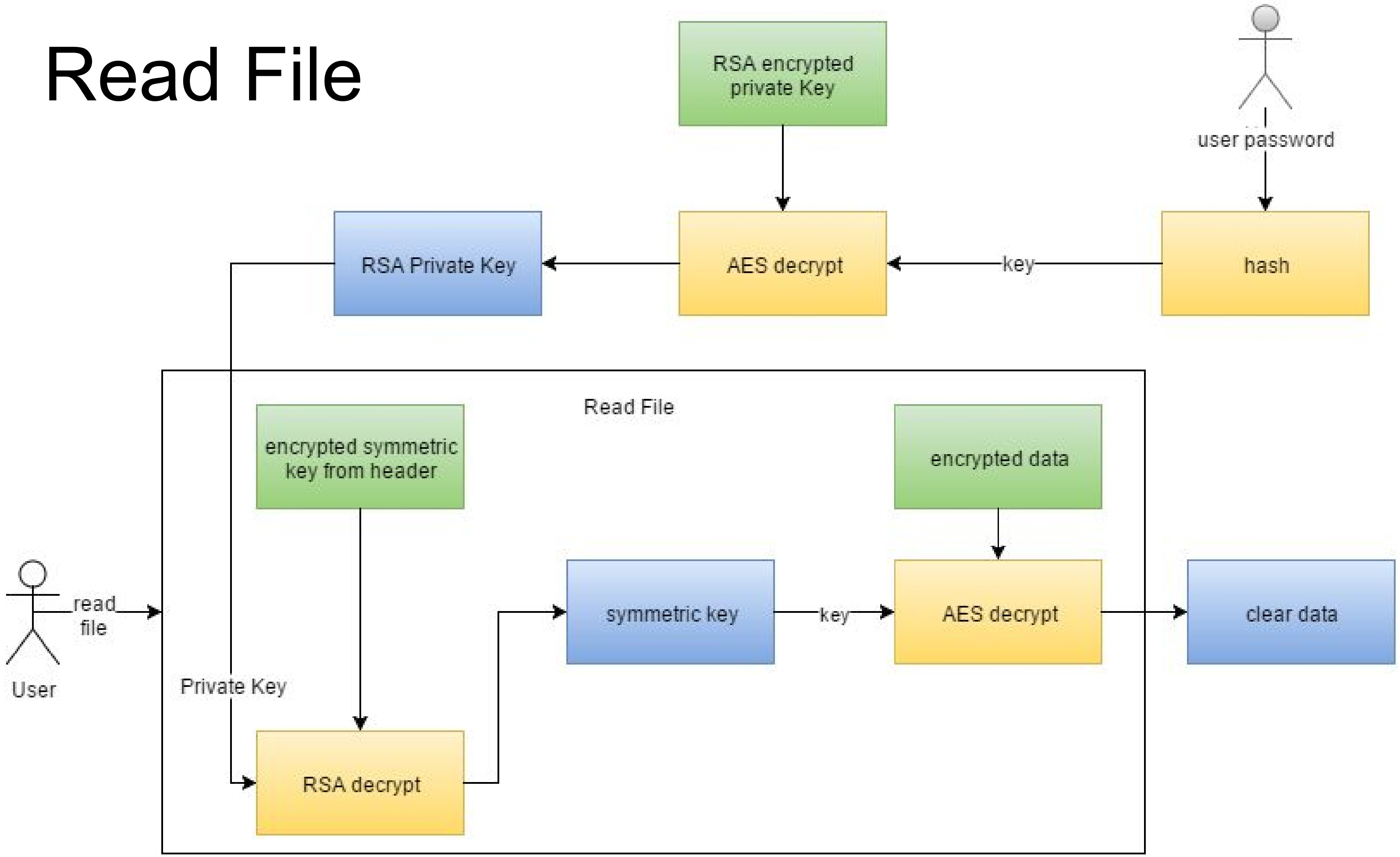
Key Hierarchy

- Write File



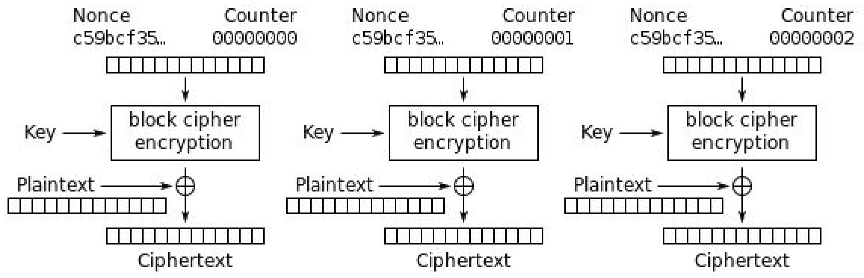
Key Hierarchy

- Read File

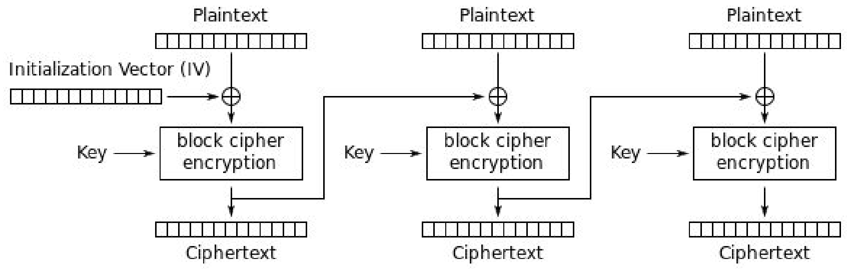


Why not CBC mode?

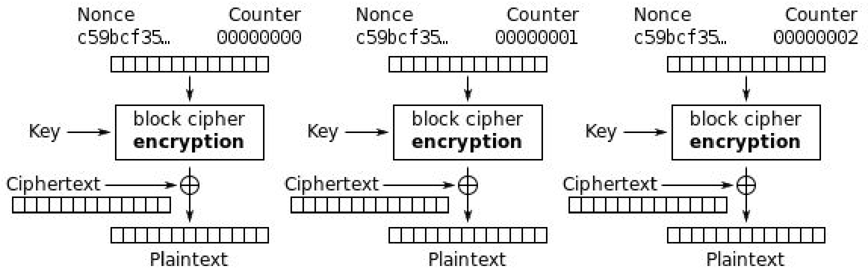
- We read only one block at the time, not the all file.



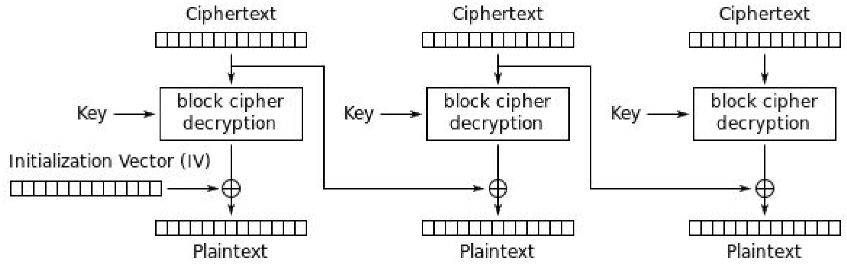
Counter (CTR) mode encryption



Cipher Block Chaining (CBC) mode encryption

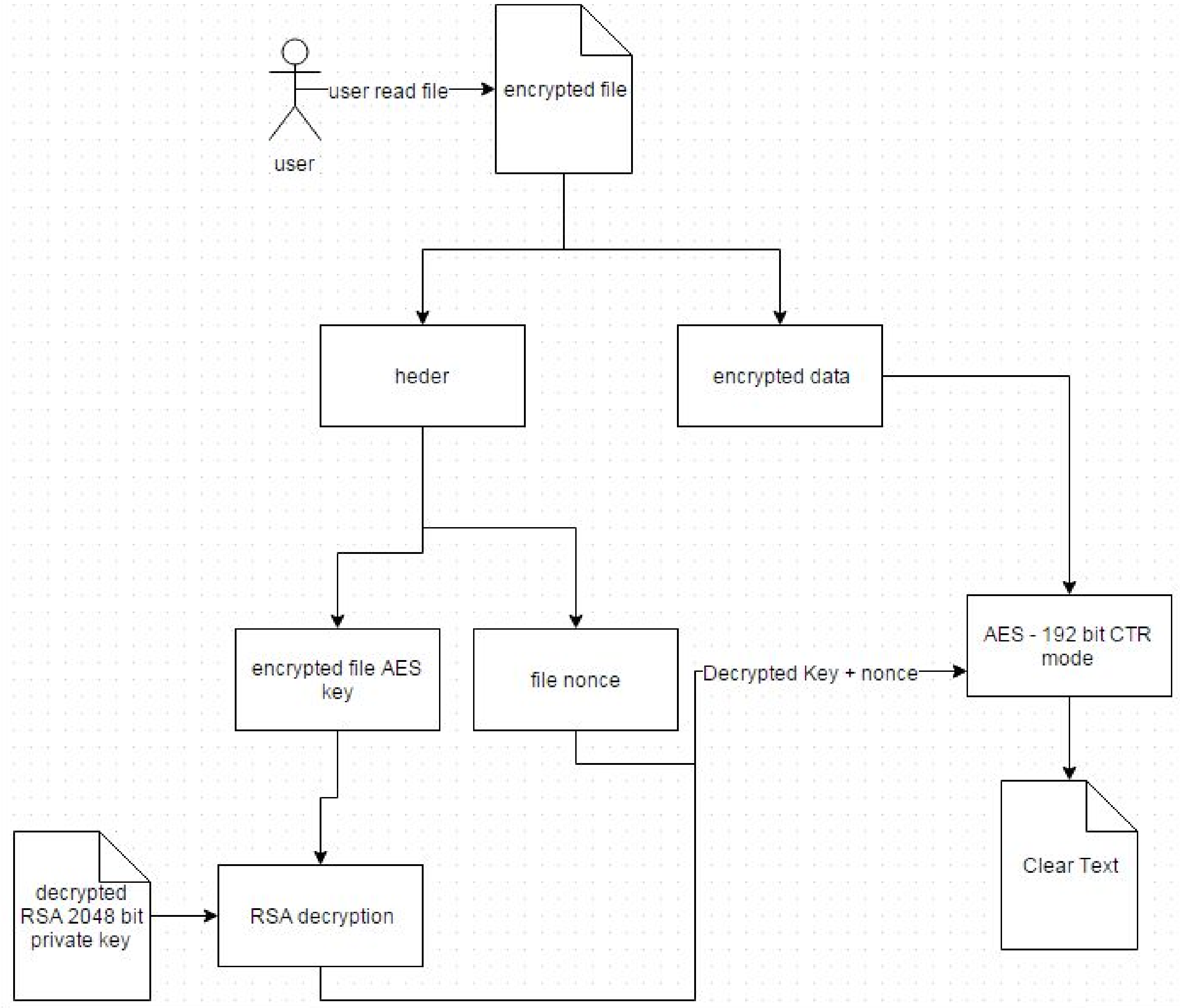


Counter (CTR) mode decryption



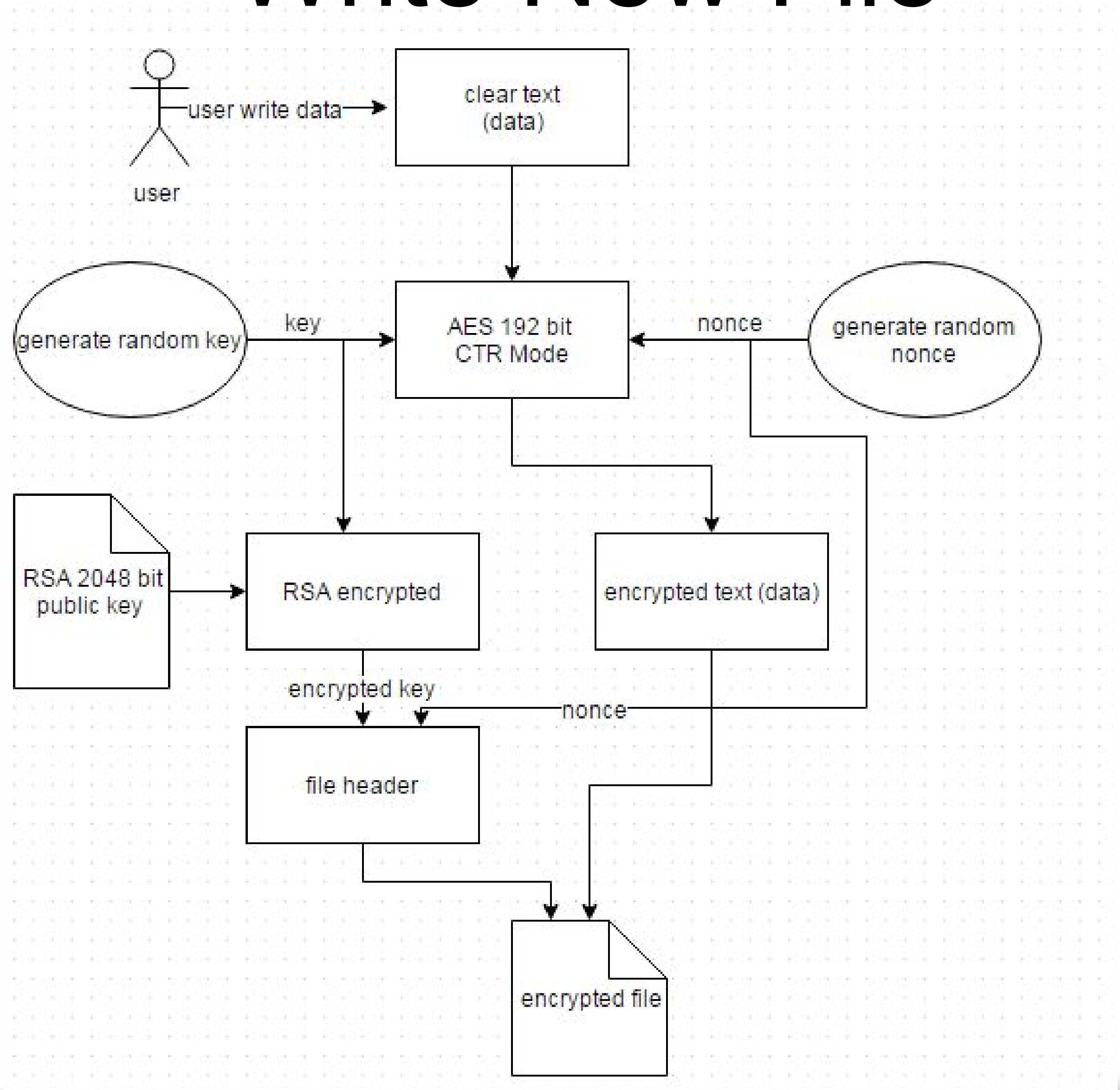
Cipher Block Chaining (CBC) mode decryption

Read File

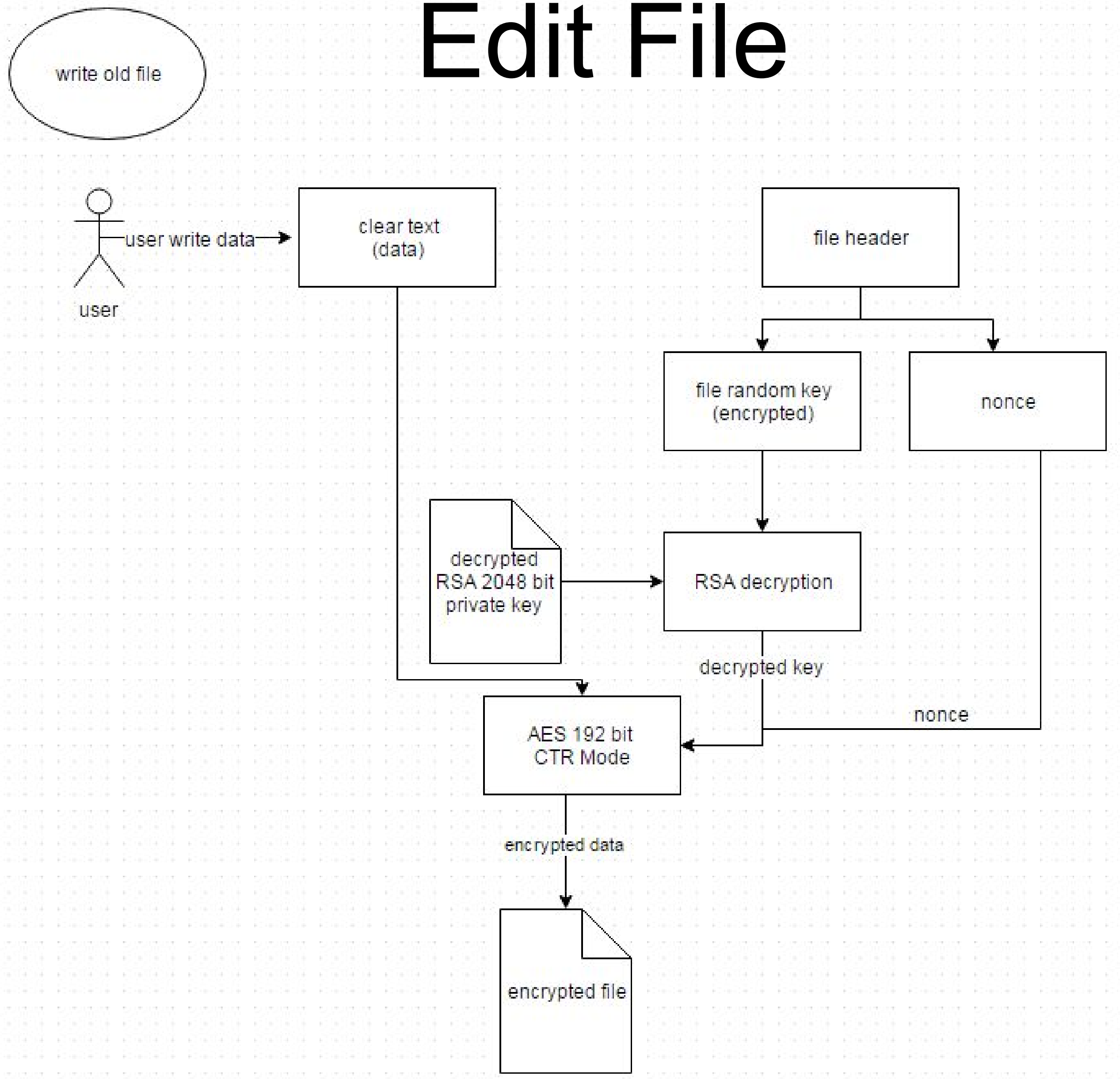


user
user read file
encrypted file
heder
encrypted data
encrypted file AES key
file nonce
Decrypted Key + nonce
AES - 192 bit CTR mode
decrypted RSA 2048 bit private key
RSA decryption
Clear Text

Write New File

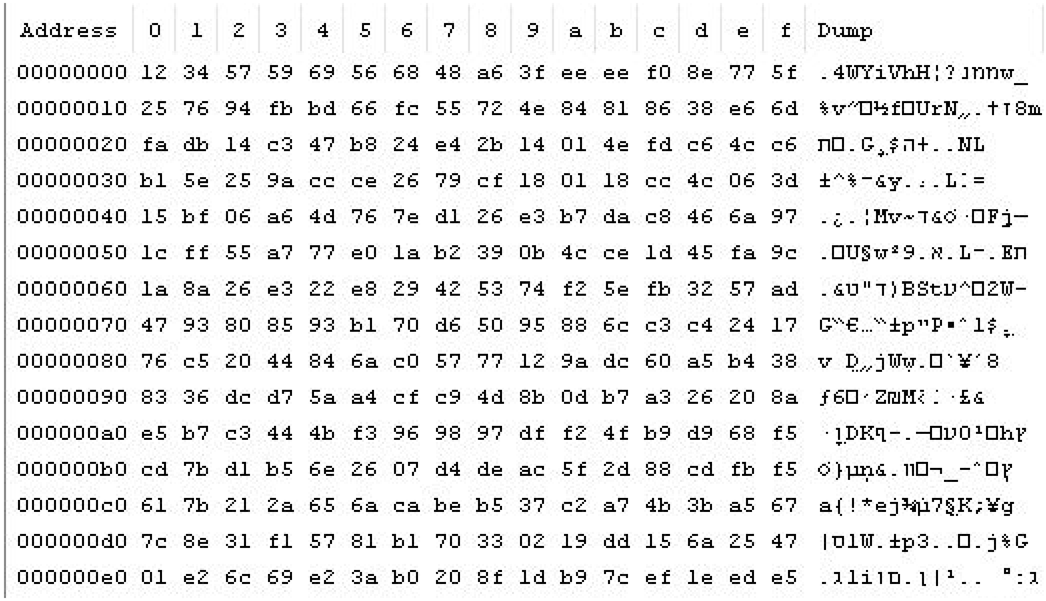


Edit File


write old file
user
user write data
clear text (data)
file header
file random key (encrypted)
nonce
decrypted RSA 2048 bit private key
RSA decryption
decrypted key
nonce
AES 192 bit CTR Mode
encrypted data
encrypted file

Encrypted File Structure

- Header.
    - Tag.
    - Encrypted Symetryc Key(s).
    - Nonce.
- Encrypted Data



```
Address   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
00000000 12 34 57 59 69 56 68 48 a6 3f ee ee f0 8e 77 5f  .4WYiVhH¦?ןחחw_
00000010 25 76 94 fb bd 66 fc 55 72 4e 84 81 86 38 e6 6d  %v″□½f□UrN„.†׀8m
00000020 fa db 14 c3 47 b8 24 e4 2b 14 01 4e fd c6 4c c6  ת□.G¸$ה+..NL
00000030 b1 5e 25 9a cc ce 26 79 cf 18 01 18 cc 4c 06 3d  ±^%-&y...L.=
00000040 15 bf 06 a6 4d 76 7e d1 26 e3 b7 da c8 46 6a 97  .¿.¦Mv~ד&ׂ·□Fj—
00000050 1c ff 55 a7 77 e0 1a b2 39 0b 4c ce 1d 45 fa 9c  .□U§wא²9.א.L-.Eת
00000060 1a 8a 26 e3 22 e8 29 42 53 74 f2 5e fb 32 57 ad  .&U"ד)BStע^□2W-
00000070 47 93 80 85 93 b1 70 d6 50 95 88 6c c3 c4 24 17  G"€…"±p"P•ˆl$¸
00000080 76 c5 20 44 84 6a c0 57 77 12 9a dc 60 a5 b4 38  v D„jWw.□`¥´8
00000090 83 36 dc d7 5a a4 cf c9 4d 8b 0d b7 a3 26 20 8a  ƒ6□·ZתM‹.·£&
000000a0 e5 b7 c3 44 4b f3 96 98 97 df f2 4f b9 d9 68 f5  ·ֹDKף-.—□ע0¹□hץ
000000b0 cd 7b d1 b5 6e 26 07 d4 de ac 5f 2d 88 cd fb f5  ׂ}µn&.װ□¬_-ˆ□ץ
000000c0 61 7b 21 2a 65 6a ca be b5 37 c2 a7 4b 3b a5 67  a{!*ej¾µ7§K;¥g
000000d0 7c 8e 31 f1 57 81 b1 70 33 02 19 dd 15 6a 25 47  |ס1W.±p3..□.j%G
000000e0 01 e2 6c 69 e2 3a b0 20 8f 1d b9 7c ef 1e ed e5  .גliגם.ן|¹.. °:ג
```

דוקטור,

אני רק שאלה

?