PE32 Password

Kyungpook National University
Taekwan Yang (CSE GSWC, KERT)

ytk2128@gmail.com

https://github.com/ytk2128/pe32-password

Contents

1. Outline

2. Implementation

3. Conclusion

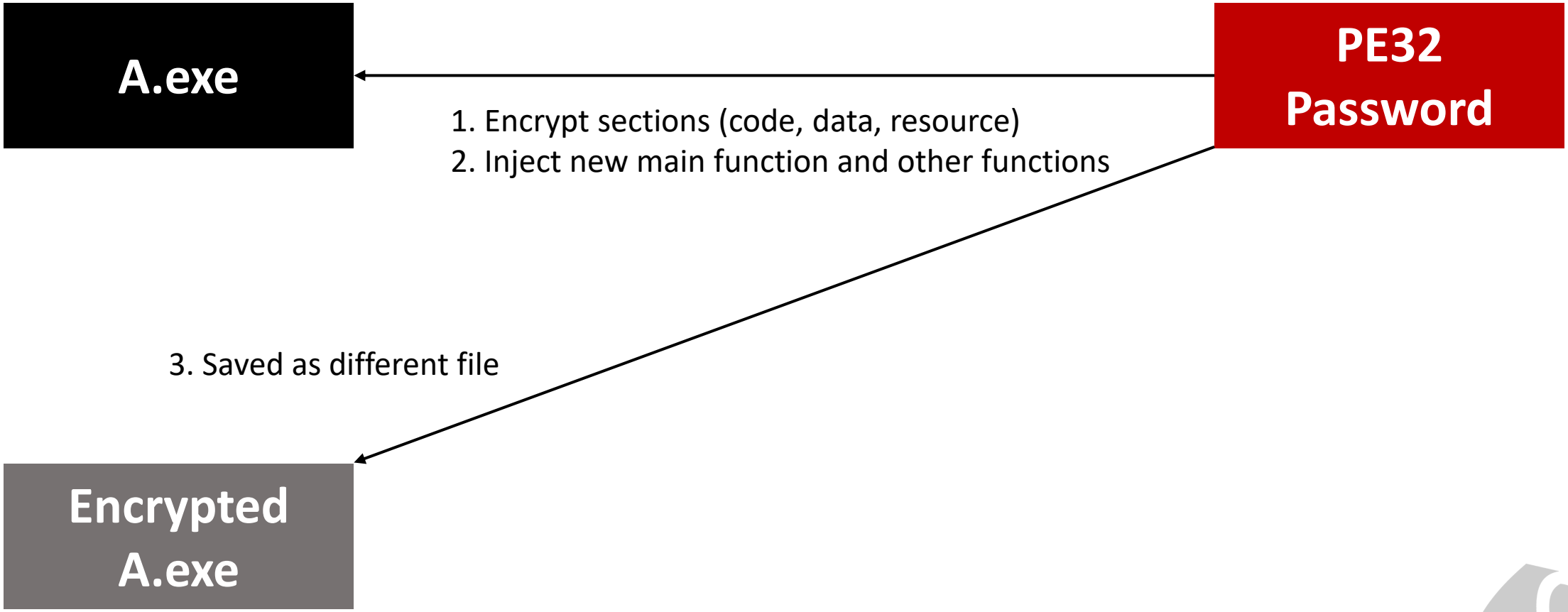
Outline

Outline

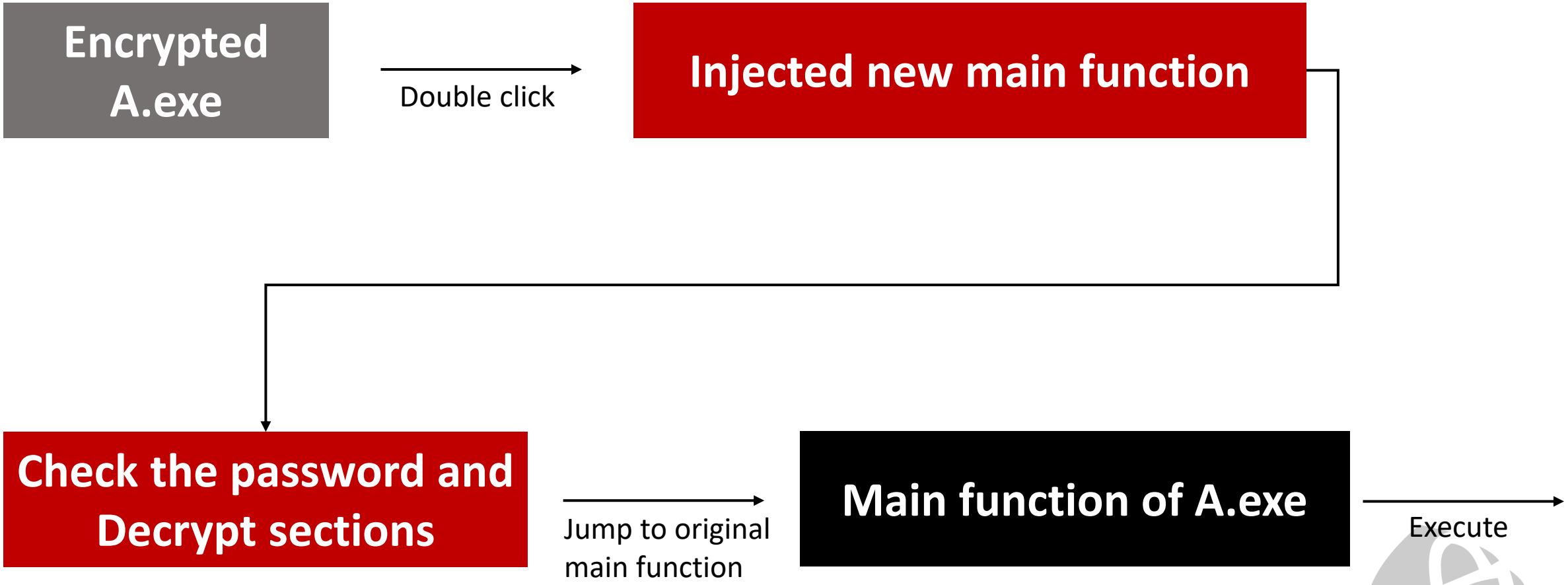
Project	**PE32** Password
Description	Encrypting x32 executable files (.exe) to lock with the password
Programming Language	C++, x86 Assembly
Operating System	Windows 10 Pro x64
Goals	1. Deeper understanding of **Windows Internals** (format of executable files and how operating system executes files) and **SEED128** (symmetric encryption algorithm provided by Korea Internet & Security Agency) 2. Developing **a sample project** based on **my library** (for modifying windows executable files)

Outline – Process of Normal A

Outline – Encryption Process

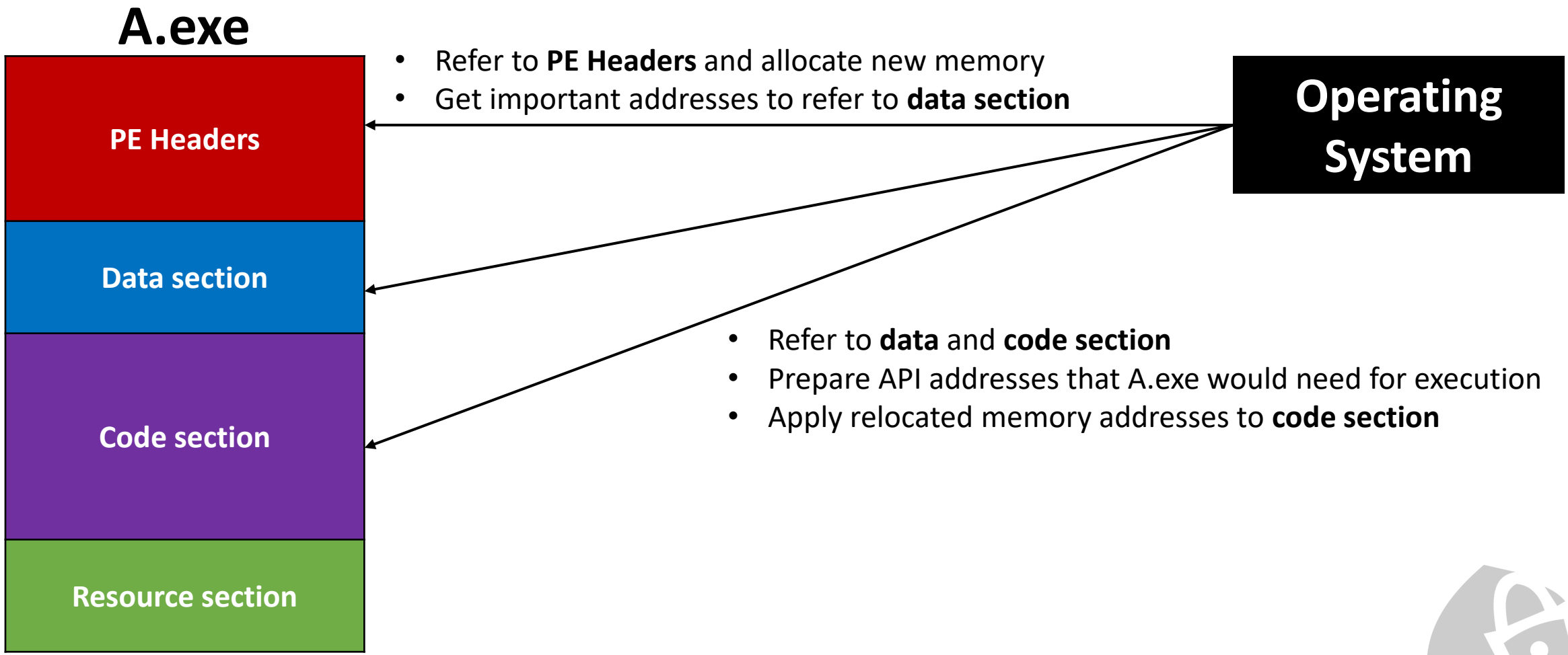


Outline – Process of Encrypted A

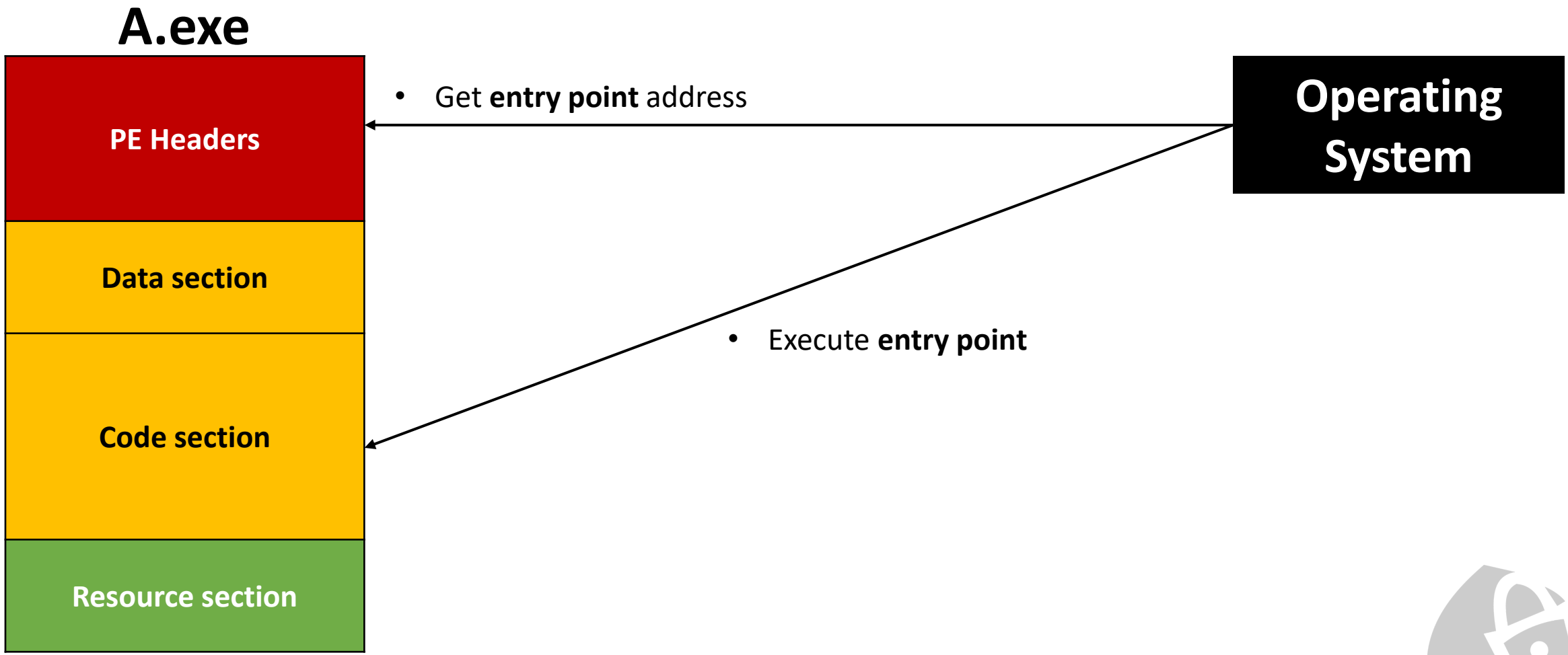


Implementation

Implementation – Normal Case

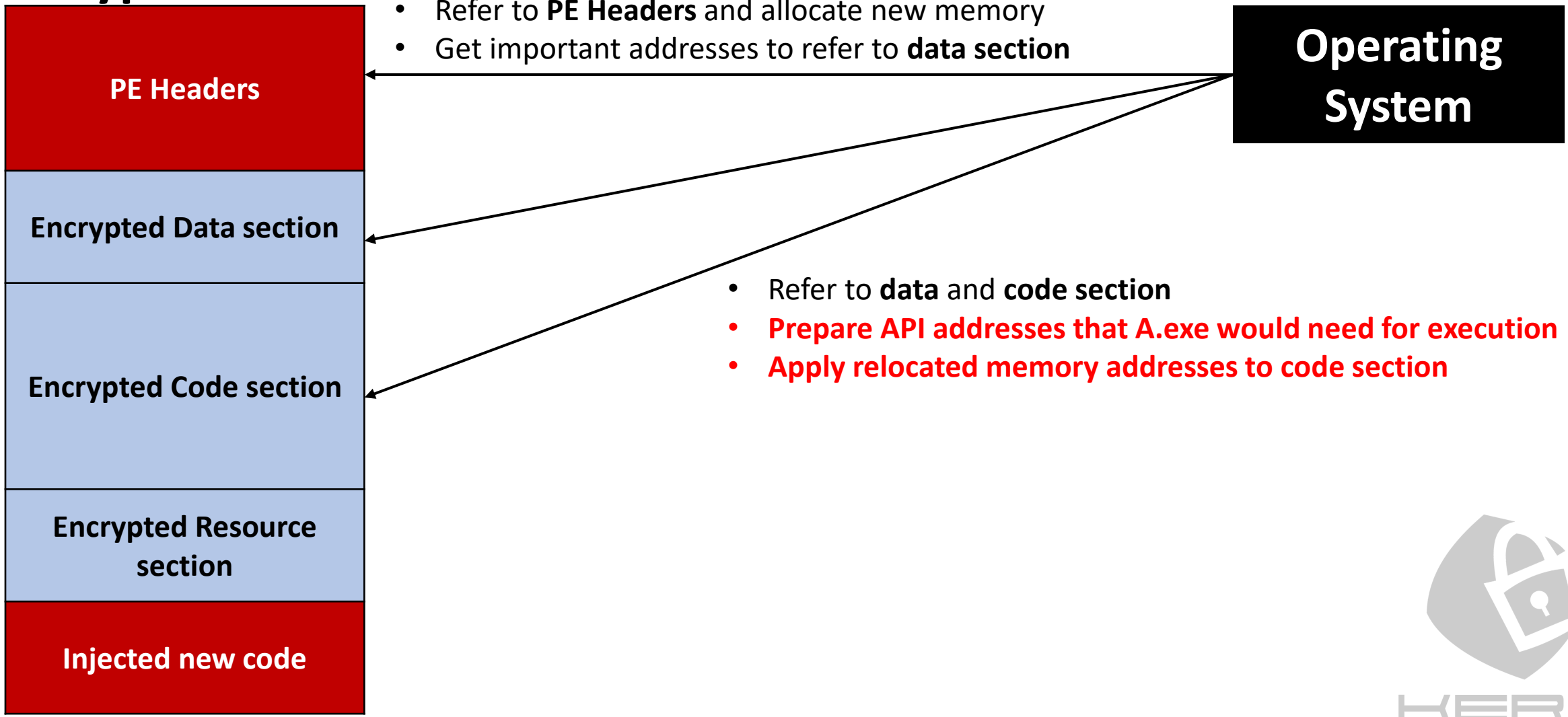


Implementation – Normal Case

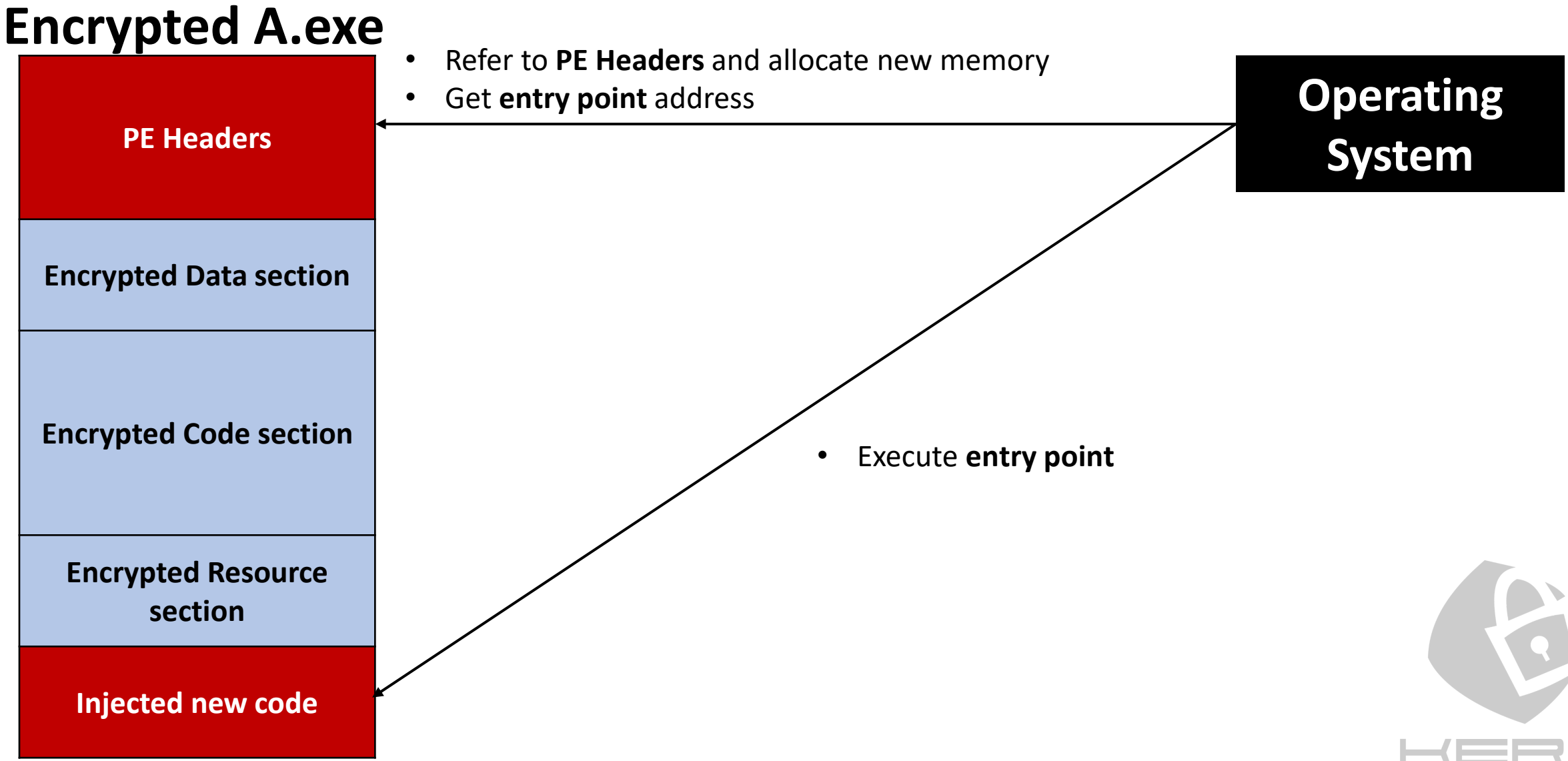


Implementation – Error Case

Encrypted A.exe

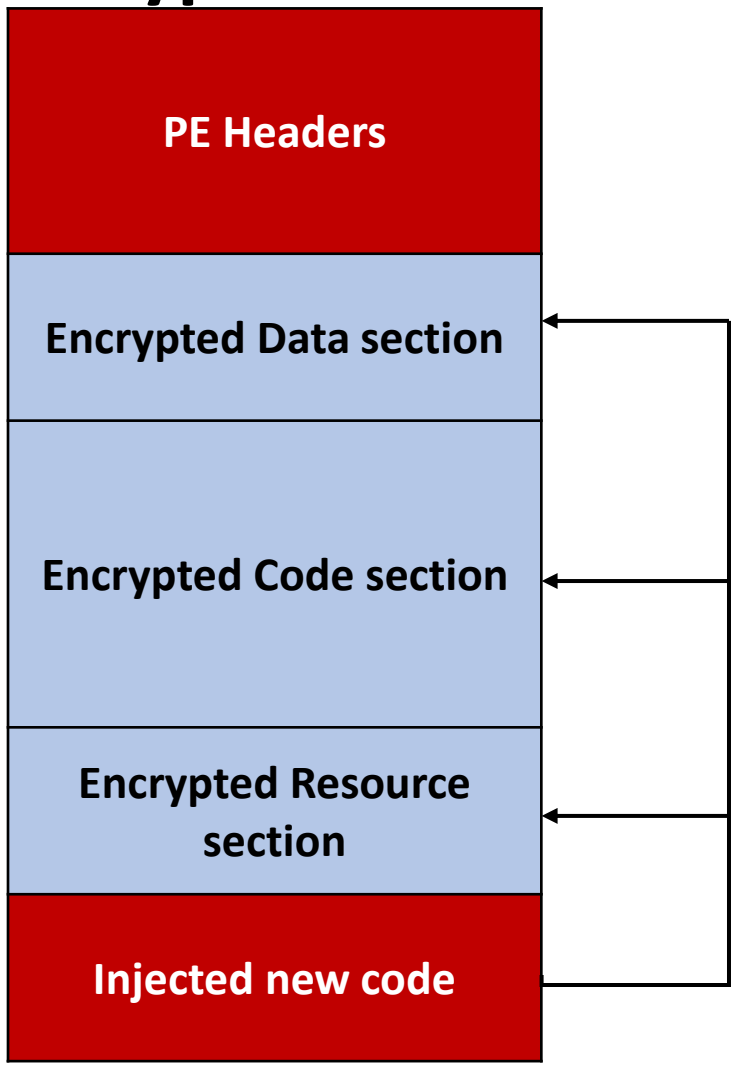


Implementation – Encrypted Case



Implementation – Encrypted Case

Encrypted A.exe

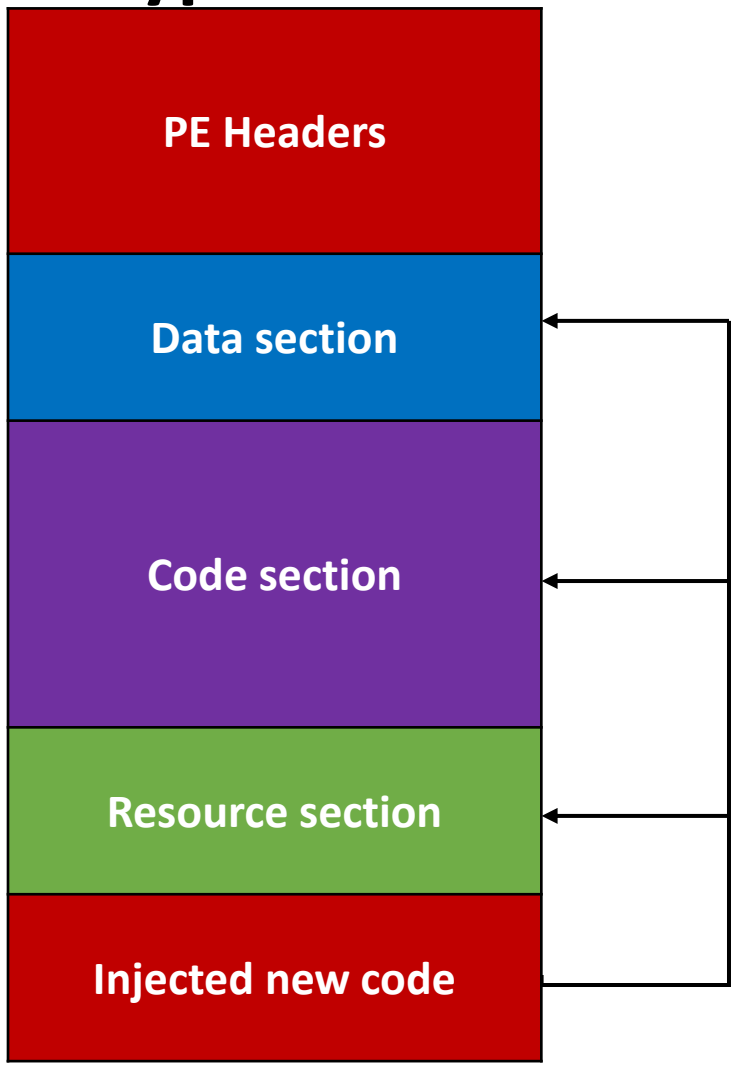


Operating System

- Decrypt sections if the password is correct

Implementation – Encrypted Case

Encrypted A.exe



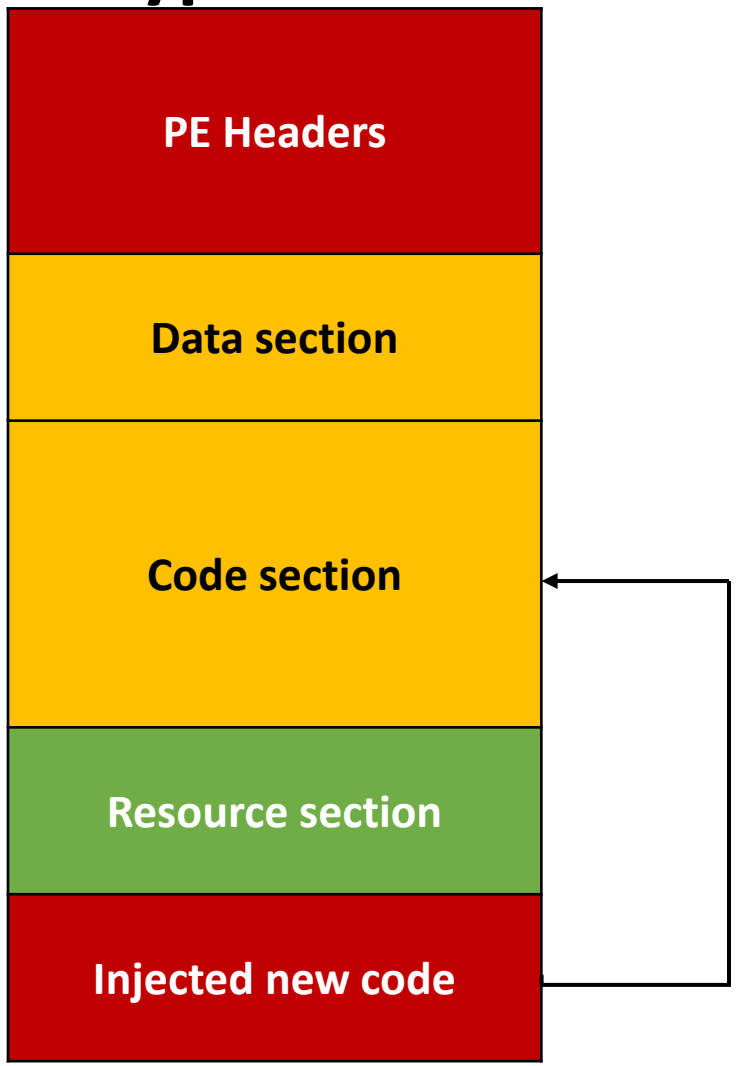
Operating System

- Decrypt sections if the password is correct

- Get important addresses to refer to **data section**
- Refer to **data** and **code section**
- Prepare API addresses that A.exe would need for execution
- Apply relocated memory addresses to **code section**

Implementation – Encrypted Case

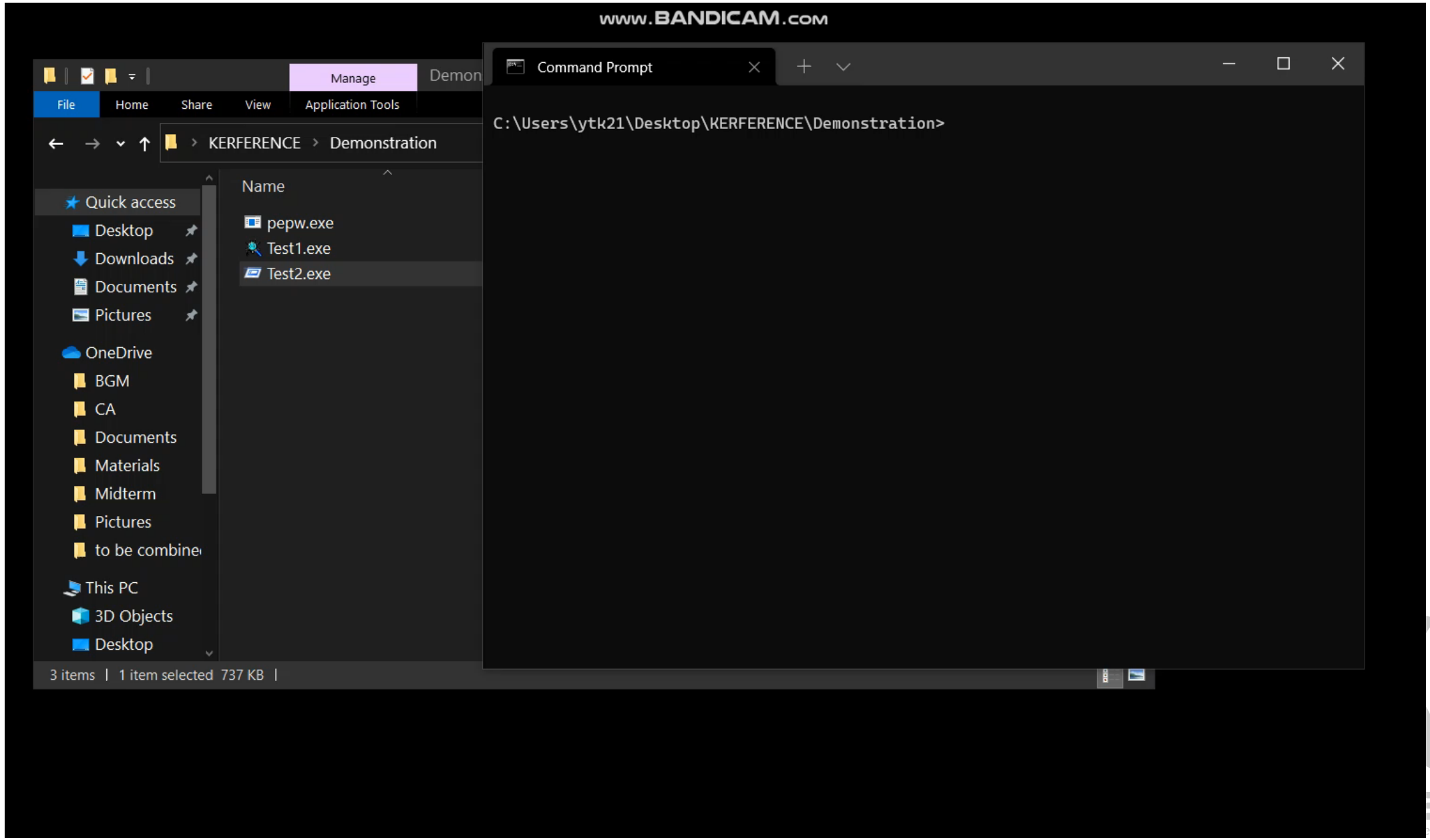
Encrypted A.exe



Operating System

- Get **original entry point** address
- Jump to **original entry point**

Implementation – Demonstration



Implementation – Comparison

Test1.exe

Name	Virtual Size	Virtual Add...	Raw Size	Raw Address	Reloc Address	Li
00000208	00000210	00000214	00000218	0000021C	00000220	00
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dw
.text	0005DC68	00001000	0005DE00	00000400	00000000	00
.rdata	0001321C	0005F000	00013400	0005E200	00000000	00
.data	000155A4	00073000	00001200	00071600	00000000	00
.rsrc	0005F6C0	00089000	0005F800	00072800	00000000	00
.reloc	000045E4	000E9000	00004600	000D2000	00000000	00

This section contains:

Code Entry Point: 00013487

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	8B	FF	55	8B	EC	33	C0	85	C9	74	05	3B	4D	08	76	05	▌ÿU▌ì3À▌Ét│;M◘v│
00000010	B8	57	00	07	80	5D	C2	04	00	8B	FF	55	8B	EC	53	56	¸W.•▌]Â♦.▌ÿU▌ìSV
00000020	57	8B	D8	8B	F2	33	C0	33	FF	85	F6	74	1C	39	45	0C	W▌Ø▌ò3À3ÿ▌öt 9E♀
00000030	74	13	8A	13	84	D2	74	0D	88	11	41	43	4E	FF	4D	0C	t‼▌‼▌Òt.▌◄ÀCNÿM♀
00000040	47	85	F6	75	E8	85	F6	75	07	49	4F	B8	7A	00	07	80	G▌öuè▌öu•IO¸z.•▌
00000050	C6	01	00	8B	4D	08	85	C9	74	02	89	39	5F	5E	5B	5D	Æ┌.▌M◘▌Ét┐▌9_^[]
00000060	C2	08	00	8B	FF	55	8B	EC	8B	4D	0C	33	C0	85	C9	74	Â◘.▌ÿU▌ì▌M♀3À▌Ét
00000070	08	81	F9	FF	FF	FF	7F	76	05	B8	57	00	07	80	85	C0	◘ ùÿÿÿ▌v│¸W.•▌▌À
00000080	7C	40	8B	45	08	53	33	DB	56	8B	F1	85	C9	74	26	8B	│@▌E◘S3ÛV▌ñ▌Ét&▌
00000090	55	10	57	BF	FE	FF	FF	7F	2B	F9	2B	D0	8D	0C	37	85	U►W¿þÿÿ▌+ù+Ð ♀7▌
000000A0	C9	74	0D	8A	0C	02	84	C9	74	06	88	08	40	4E	75	EC	Ét.▌♀┐▌Ét─▌◘@Nui

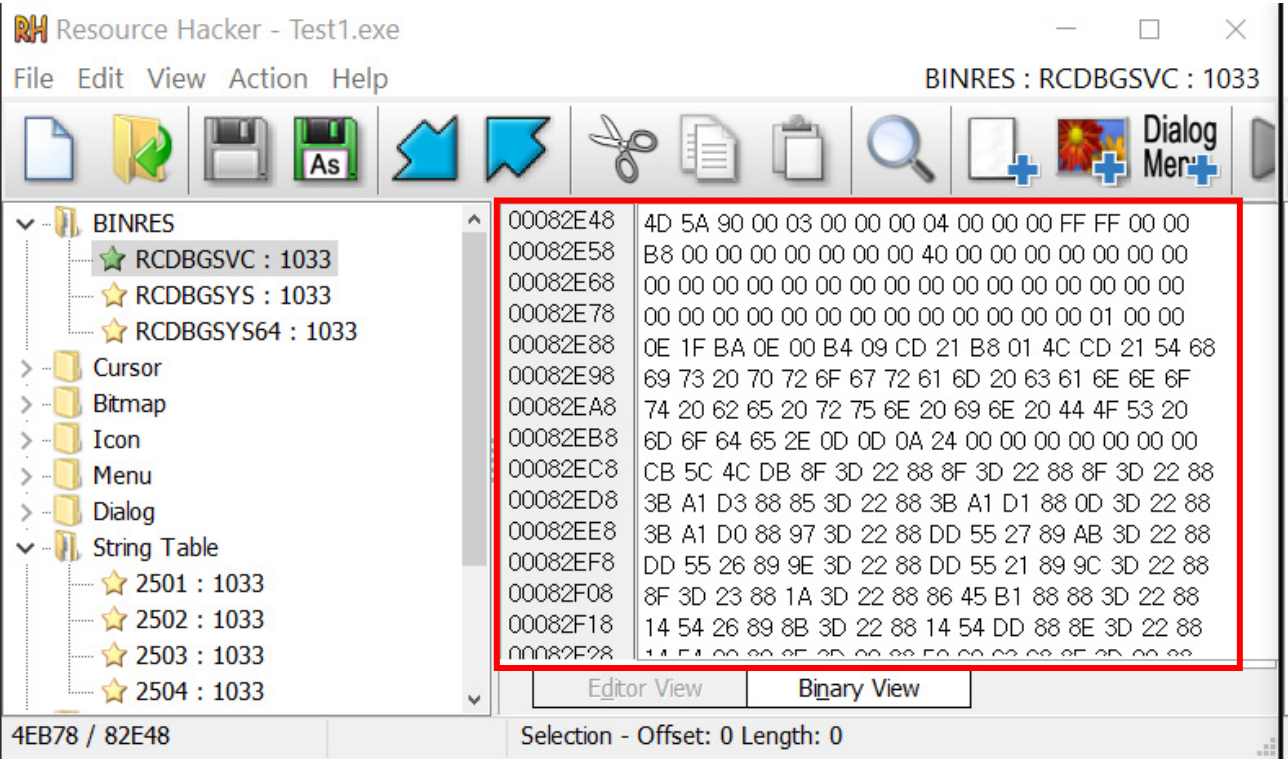
Test1.exe | Test1.exe_out.exe

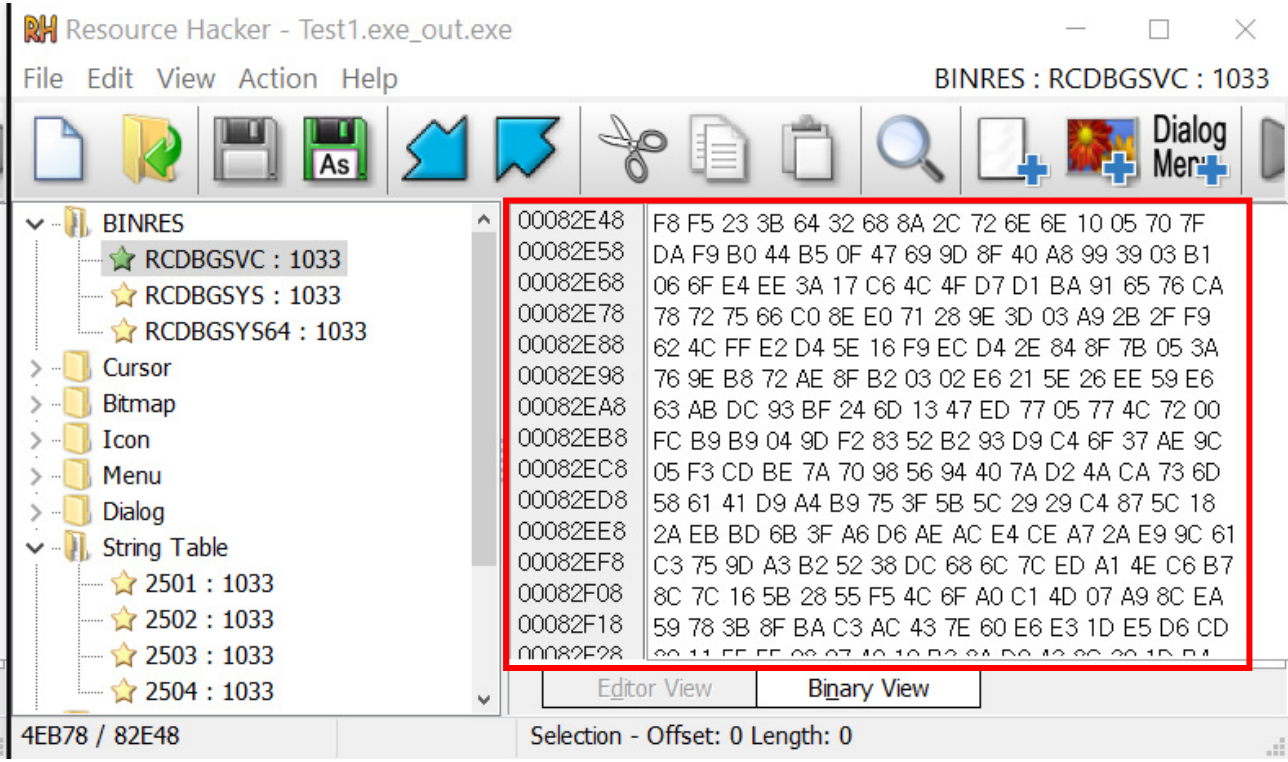
Name	Virtual Size	Virtual Add...	Raw Size	Raw Address	Reloc Address	Li
00000208	00000210	00000214	00000218	0000021C	00000220	00
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dw
.text	0005DC68	00001000	0005DE00	00000400	00000000	00
.rdata	0001321C	0005F000	00013400	0005E200	00000000	00
.data	000155A4	00073000	00001200	00071600	00000000	00
.rsrc	0005F6C0	00089000	0005F800	00072800	00000000	00
.reloc	000045E4	000E9000	00004600	000D2000	00000000	00
.pepw	00007000	000EE000	00007000	000D6600	00000000	00

This section contains:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	EE	F6	4B	51	2E	86	50	A7	D3	1D	DB	89	10	44	57	13	îöKQ.▌P§Ó Û▌┼DW‼
00000010	28	72	18	26	89	EA	72	DC	01	EC	D1	20	ED	E7	6A	EA	(r↑&▌êrÜ┌ìÑ.íçjê
00000020	BE	A6	0D	4B	97	96	57	52	F5	E1	DD	6D	46	5A	FF	3A	¾│.K▌▌WRõáÝmFZÿ:
00000030	5A	43	80	69	6F	32	1D	21	9D	25	CA	AB	11	F7	37	90	ZC▌io2 ! %Ê«◄÷7
00000040	1A	3E	1E	2B	4C	2D	5F	A1	69	DA	9C	90	6E	FE	50	C4	→> +L-_¡iÚ▌ nþPÄ
00000050	90	C6	37	8E	F0	54	5C	08	45	60	70	9F	5B	F1	A9	A3	Æ7▌ðT\◘E`p▌[ñ©£
00000060	A7	F2	FD	11	FF	1D	2B	71	C0	4A	59	52	F0	DF	0C	2F	§òý◄ÿ +qÀJYRð♀/
00000070	2F	98	60	03	5D	3A	5C	70	16	7F	88	D4	4D	65	6C	03	/▌`└]:\p┬▌▌ÔMel└
00000080	EC	84	2A	51	24	E6	E8	9A	AD	63	89	3C	EB	99	10	0F	ì▌*Q$æè▌-c▌<ë▌┼☼
00000090	A4	58	65	D2	A9	9C	25	BB	3E	5A	7A	B4	33	C7	12	F9	¤XeÒ©▌%»>Zz´3Ç↕ù
000000A0	45	CD	51	58	D9	0D	7F	C8	35	D7	79	5E	EF	0D	89	C2	EÍQXÙ.▌È5×y^ï.▌Â

Implementation – Comparison





Conclusion

Conclusion

What I learned from this project	1. How to deal with big and complex projects 2. How to make features more efficient in terms of the time complexity
What I regret the most during the project period	1. Being lazy 2. The part where I couldn’t learn SEED128 ECB Algorithm more deeply due to the deadline
Future goals	1. Releasing my library and this project on GitHub officially 2. Developing a better encryptor for software security using my library

Thank you