

Practical Cryptographic Key Recovery

Jesse Kornblum

Outline

- **Introduction**
- **Targets**
- **Current Methods**
- **Tool Marks**
- **Example - BitLocker**
- **Conclusion**

Targets

- **Documented Open Source**
 - TrueCrypt
- **Undocumented Open Source**
 - PGP Whole Disk Encryption
 - <http://www.pgp.com/downloads/sourcecode/>
- **Documented Closed Source**
 - BitLocker Drive Encryption*
- **Undocumented Closed Source**
 - PointSec
 - Previously unseen tools

Current Methods

- **Brute Force**
 - Try every block of bytes as possible key
 - "Last resort of the incompetent"
 - See "Linear Scan" paper by Hargreaves and Chivers
 - Doesn't work for split keys

Current Methods

- **Key Schedule Search**
 - Better brute force
 - Really identifying data that is not a *key schedule*
 - See "*Cold Boot*" paper by Halderman et al.

Current Methods

- **Source code analysis**
 - **Requires elbow grease**
 - **Can't be automated**
 - **Works great**
 - **May have to update for each version**
 - **See "Volatools" paper by Walters and Petroni, BlackHat Federal 2007**

Tool Marks



Image courtesy Flickr user grendelkhan, <http://flickr.com/photos/grendelkhan/118876699/>

- **Marks specific to individual tools**
- **Associated with physical forensics**

Tool Marks

- **Were the screwdrivers found in the suspect's house used on the screws found on the bank vault?**



Image courtesy Flickr user Uwe Hermann, <http://flickr.com/photos/uwehermann/92145964/sizes/m/>

Computer Forensics Tool Marks

- **Anything detectable that software stores in RAM or on disk that identifies the tool in question**
 - **Most Recently Used lists**
 - **Header and footer carving**
 - **Registry keys left after program removed**
 - **Preferences files in user directories**
 - **Wiping programs leave traces behind**

Cryptographic Tool Marks

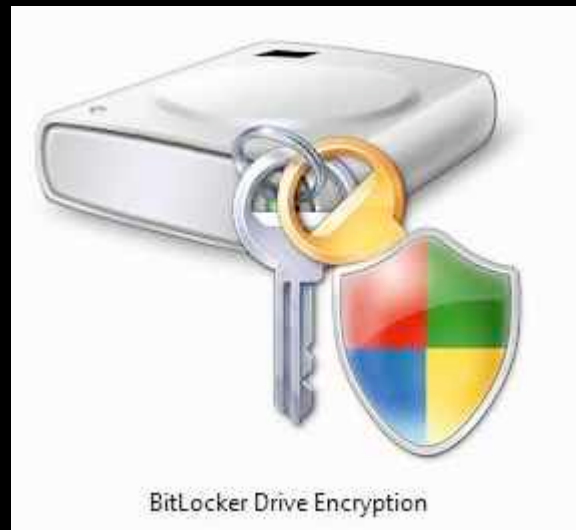
- **Hard to detect the keys themselves**
 - **Should be random**
- **Can detect the cryptographic tool itself**
 - **Programs**
 - **Drivers**
 - **Mounted volumes**
- **Can detect the structure surrounding the keys**

BitLocker Drive Encryption

- **Full Volume Encryption bundled with Windows Vista Ultimate**
- **Uses 128 bit AES-CBC + Elephant diffuser**
 - **Can configure for 256 bit and/or without diffuser**
- **Crypto developed by Niels Ferguson**
 - **also wrote Twofish, Helix, Fortuna RNG, CCM mode**
 - **Uses AES-CCM for key management**
- **Actual encryption work is done with 512 bit Full Volume Encryption Key (FVEK)**
 - **Key is 512 bits regardless of mode being used**

BitLocker Drive Encryption

- I am not aware of any backdoors in BitLocker Drive Encryption
- You cannot access a protected volume without the FVEK



BitLocker Drive Encryption

Image courtesy of the Microsoft Corporation.

BitLocker Drive Encryption is a registered trademark of the Microsoft Corporation.

BitLocker Drive Encryption

- **Good documentation, but not complete**
 - **Key management systems not described**
 - **No implementation of elephant provided**
- **Reverse engineered by Kumar and Kumar**
 - **Published paper, linux driver to mount protected volumes**
 - **<http://www.nvlabs.in/node/9>**

BitLocker Drive Encryption

- **Brute Force works**
 - FVEK is in RAM
- **Key schedule search works**
 - Finds several schedules
 - Two of the keys make up the FVEK
 - Some assembly required
- **Source code analysis**
 - Not an option for most of us

BitLocker Tool Marks

- **BitLocker AES key schedules**
 - Several schedules in memory at any given time
 - Some bits of FVEK used to generate sector keys
 - Other bits of FVEK used to encrypt/decrypt data
 - In default mode, some bits unused

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	3F	26	C8	B5	FF	87	47	B1	D5	26	12	43	EC	CD	78	C6
00000010	D5	09	AF	19	D1	5A	10	03	B5	4D	1B	73	0E	EC	0A	93
00000020	7A	16	05	EB	54	9F	39	1C	2E	5D	6A	DB	BC	67	C2	36
00000030	1C	3D	F3	60	AF	A1	EB	6F	E4	47	B2	E3	A5	B5	38	D9

BitLocker Tool Marks

- **AES key schedules**
 - **Encryption and Decryption schedules**

AES key
round key 1
round key 2
...
round key n-2
round key n-1
round key n

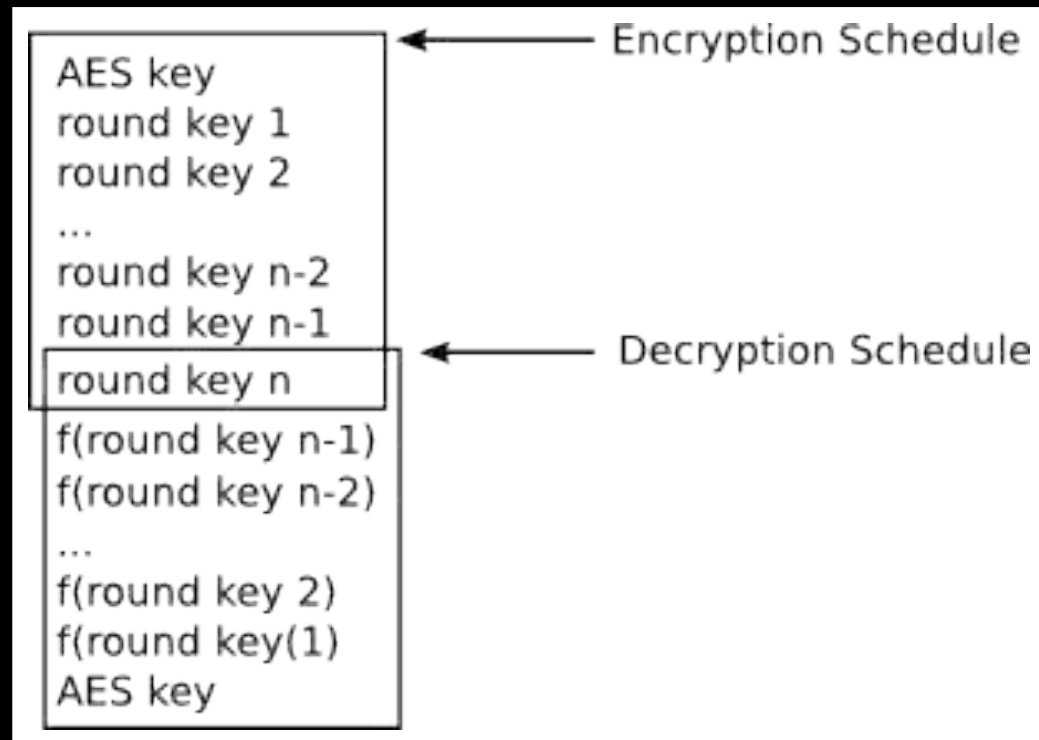
Encryption Schedule

round key n
f(round key n-1)
f(round key n-2)
...
f(round key 2)
f(round key(1))
AES key

Decryption Schedule

BitLocker Tool Marks

- Searching for BitLocker AES key schedules in RAM
 - *Overlapped slightly*



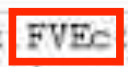
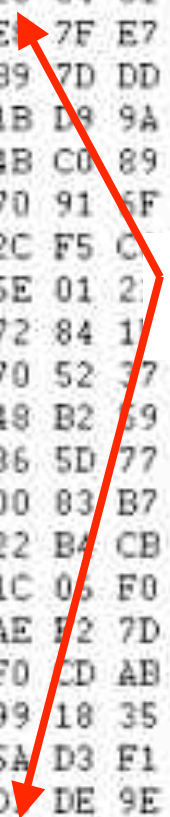
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
1EE5E000	00	00	7A	04	46	56	45	63	3C	FE	D8	83	E0	FE	D8	83	.. z FVEc p@ a@
1EE5E010	20	E0	E5	82	B0	03	00	00	00	80	00	00	00	00	00	00	ãã '.....
1EE5E020	20	84	01	68	02	D5	44	41	AB	E8	AA	47	5B	12	63	F5	[. cõ
1EE5E030	E1	7F	E7	51	EA	AA	A3	10					1A	50	6A	A2	è Pool Tag Pjc
1EE5E040	B9	7D	DD	F3	53	D7	7E	E1	Algorithm				08	C5	1D	16	!.....À..
1EE5E050	1B	D9	9A	C3	48	0E	E4	20					52	5E	8E	82	.Ü ÄH.ã Z IR^
1EE5E060	4B	C0	89	C3	03	CE	6D	E3	59	55	FE	77	0B	0B	70	F5	KÄ Ä. ImãYUpw. põ
1EE5E070	70	91	6F	E8	73	5F	02	0B	2A	0A	FC	7C	21	01	8C	89	p'cès_..*ü !
1EE5E080	2C	F5	C0										A1	BA	EB		.8E_?E.u 6bT!e
1EE5E090	5E	01	20										AA	67	A2		^ .15.«ë+t.YI ?gc
1EE5E0A0	72	84	10										8E	4A	42		ri s/õ@.\$-ã' JB
1EE5E0B0	70	52	37	4E	03	7D	C7	E7	04	59	EA	07	23	D7	A0	45	pR7N.}Çq.Yè.#x E
1EE5E0C0	48	B2	69	68	4B	CF	9E	8F	4F	96	74	88	6C	41	D4	CD	H?YhKI O t AÖI
1EE5E0D0	86	5D	77	F7	5A	D4	22	F2	6C	00	F9	25	73	DE	D2	6E]w-ZÖ"ö l.úxspÖn
1EE5E0E0	00	83	B7	5B	DC	89	55	05	36	D4	DB	D7	1F	DE	2B	4B	. [Ü U.60Üx.p+K
1EE5E0F0	22	B4	CB	16	DC	0A	E2	5E	EA	5D	8E	D2	29	0A	F0	9C	"`E.U.ã^ë)Ö).õ
1EE5E100	1C	06	F0	EE	FE	BE	29	48	36	57	6C	8C	C3	57	7E	4E	..õipã)H6Vl ÄW^N
1EE5E110	AE	F2	7D	47	E2	B8	D9	A6	C8	E9	45	C4	F5	00	12	C2	õö}Gã.Ü EãEÄõ..A
1EE5E120	F0	CD	AB	57	4C	4A	A4	E1	2A	51	9C	62	3D	E9	S7	06	õI«VLJ#ã*Q b=ëV.
1EE5E130	99	18	35	2F	BC	87	0F	B6	66	1B	38	83	17	B8	CB	64	.S/ã .#é.8 .,Ed
1EE5E140	5A	D3	F1	92	25	9F	3A	99	DA	9C	37	35	71	A3	F3	E7	ZÖH'z : Ü 75qföç
1EE5E150	D1	DE	9E	B2	7F	4C	CB	0B	FF	03	0D	AC	AB	3F	C4	D2	Öp ? LE.ÿ..~«?ÄÖ
1EE5E160	20	84	01	68	02	D5	44	41	AB	E8	AA	47	5B	12	63	F5	.h.ÖDA«ë?G[.cõ
1EE5E170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1EE5E180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1EE5E190	00	00	00	00	00	00	00	00					00	00	00	00
1EE5E1A0	00	00	00	00	00	00	00	00	Zeros				00	00	00	00
1EE5E1B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1EE5E1C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1EE5E1D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1EE5E1E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Pool Tag

Algorithm

AES key in schedule

Zeros



BitLocker Tool Marks

- **0x0 FVEc pool tag**
- **0x14 Algorithm ID, must be 0x8000-0x8003**
- **0x1C Start of first BitLocker AES schedule**
 - **AES key must be at start and end of schedule**
 - **bytes 0x1C-0x2C and 0x15C-0x16C for 128-bit**
 - **Zeros at end of schedule if 128-bit mode**
- **0x1EC Start of second BitLocker AES schedule**
 - **Same rules as above**

BitLocker Tool Marks

- Not perfect, but good enough
- Original

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	3F	26	C8	B5	FF	87	47	B1	D5	26	12	43	EC	CD	78	C6
00000010	D5	09	AF	19	D1	5A	10	03	B5	4D	1B	73	0E	EC	0A	93
00000020	7A	16	05	EB	54	9F	39	1C	2E	5D	6A	DB	BC	67	C2	36
00000030	1C	3D	F3	60	AF	A1	EB	6F	E4	47	B2	E3	A5	B5	38	D9

- Recovered

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	3F	26	C8	B5	FF	87	47	B1	D5	26	12	43	EC	CD	78	C6
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	7A	16	05	EB	54	9F	39	1C	2E	5D	6A	DB	BC	67	C2	36
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Finding Tool Marks

- How did we do this?
 - RTFM
 - FIPS certifications are great!
 - Ask developers for help
 - WinHex
 - IDA Pro
 - Checked builds
 - Debugging symbols
- Always trying to answer:
 - How does it know where to look?

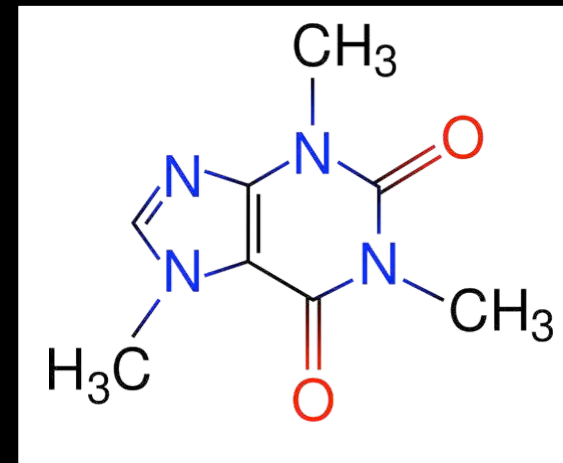


Image courtesy of User:Icey on Wikipedia and is public domain

Performance

- **Brute Force**
 - $O(nm)$
- **Key Schedule Search**
 - $O(nm)$
- **Source Code**
 - $X^* + O(n)$, where X^* may be infinite
- **Toolmarks**
 - $X + O(n)$

Forensics Tool Marks

- **Requires as much elbow grease as source code analysis**
 - Often more
 - Doesn't require the source code
- **May require updating for each version**
 - TrueCrypt
- **May be your only option for previously unseen tools**

Questions?

- **Introduction**
- **Targets**
- **Current Methods**
- **Tool Marks**
- **Example - BitLocker**
- **Conclusion**

Thank you



Image courtesy of the Microsoft Corporation.

- **Microsoft Corporation for keeping me employed**
- **Kumar and Kumar for their reverse engineering work**
- **AAron Walters for organizing**
- **ManTech International Corporation for letting me geek out**
- **You for staying!**

References

- **BitLocker Drive Encryption**

- N. Kumar and V. Kumar, "Bitlocker and Windows Vista",
<http://www.nvlabs.in/node/9>
- FIPS Security Policy:
<http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp947.pdf>

- **Brute Force Searches**

- C. Hargeaves and H. Chivers, "Recovery of Encryption Keys from Memory Using a Linear Scan",
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/4529302/4529303/04529504.pdf>

References

- **Key Schedule Searching**
 - **Cold Boot paper, <http://citp.princeton.edu/memory/>**
- **Source Code Analysis**
 - **AAron Walters and Nick Petroni, Volatools, Volatools: Integrating Volatile Memory Forensics into the Digital Investigation Process, <http://4tphi.net/fatkit/>**