Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.6, Dec. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.6.1339

2019년 랜섬웨어 암호화 프로세스 분석 및 복호화 방안 연구*

이 세 훈,[1†] 윤 병 철,[1] 김 소 람,[1] 김 기 윤,[1] 이 영 주,[3] 김 대 운,[3] 박 해 룡,[3] 김 종 성[1,2‡]
[1]국민대학교 금융정보보안학과, [2]국민대학교 정보보안암호수학과, [3]한국인터넷진흥원

A Study on Encryption Process and Decryption of Ransomware in 2019*

Sehoon Lee,[1†] Byungchul Youn,[1] Soram Kim,[1] Giyoon Kim,[1] Yeongju Lee,[3]
Daeun Kim,[3] Haeryong Park,[3] Jongsung Kim[1,2‡]
[1]Dept. of Financial Information Security, Kookmin University,
[2]Dept. of Information Security, Cryptology, and Mathematics, Kookmin University
[3]Korea Internet & Security Agency

요 약

랜섬웨어는 사용자의 파일을 암호화하고, 이를 복구하는 대가로 금전을 요구하는 악성 소프트웨어다. 랜섬웨어의 수가 늘어남과 동시에 사용되는 암호화 프로세스 또한 정교해지며 보안 강도도 높아지고 있다. 이에 따라 랜섬웨어의 분석은 점점 어려워지고 복구 가능한 랜섬웨어의 수도 줄어들고 있다. 그러므로 지능화된 랜섬웨어의 암호화 프로세스 및 복호화 방안에 관한 연구는 필수적이다. 본 논문은 2019년 주요 신규 랜섬웨어 5종에 대해 역공학하여 암호화 프로세스를 밝히고 이를 기반으로 복구 가능성에 대한 연구를 진행하였다.

ABSTRACT

Ransomware is a malicious software which requires money to decrypt files that were encrypted. As the number of ransomware grows, the encryption process in ransomware has been more sophisticated and the strength of security has been more stronger. As a result, analysis of ransomware becomes more difficult and the number of decryptable ransomware is getting smaller. So, research on encryption process and decryption method of ransomware is necessary. In this paper, we show encryption processes of 5 ransomwares which were revealed in 2019, and analyze whether or not those ransomwares are decryptable.

Keywords: Ransomware, Decryption, Reverse Engineering

I. 서 론

랜섬웨어(Ransomware)는 불특정 다수의 PC를 대상으로 시스템 자체를 잠그거나 문서, 사진, 동영상과 같이 중요 파일을 암호화하며 복구 조건으로 일정한 몸값(Ransom)을 요구하는 악성 소프트웨어다. 초기 랜섬웨어는 낮은 PC 보급률로 의해 피해 규모가 작았지만, 현재는 PC 보급률의 증가, IT 기술 발전 그리고 익명성과 추적 불가능한 거래가 가능한 비트코인이 등장하여 랜섬웨어 공격자가 직접적으로

Received(10. 08. 2019), Modified(12. 10. 2019), Accepted(12. 10. 2019)
* 이 논문은 2019년도 암호이용활성화의 재원으로 한국인터넷진흥원의 지원을 받아 수행된 연구 사업임(KISA2019-0079)
* 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2019R1F1A1060634)
† 주저자, dreamtree304@kookmin.ac.kr
‡ 교신저자, jskim@kookmin.ac.kr(Corresponding author)

금전적 이윤을 얻을 수 있는 구조가 갖춰짐에 따라서 랜섬웨어의 규모가 기하급수적으로 증가하게 되었다. 한국랜섬웨어침해대응센터에 따르면 랜섬웨어로 인한 국내 피해량은 2017년 7,000억 원에서 2018년 1조 2,500억 원으로 약 두 배가량 증가하였다[1]. 또한, 개인을 대상으로 감염시켰던 과거와 달리 기업이나 관공서를 대상으로 하는 표적형 랜섬웨어의 수가 증가했다. 해외의 경우, 2019년 3월 노르웨이의 알루미늄 생산 업체인 Norsk Hydro가 랜섬웨어 감염되었다. 이로 인해 금속 압출 공정 다수가 가동 중단되었으며, 피해액은 4,100만 달러로 추산된다[2]. 또한, 같은 달 미국 조지아 주 잭슨 카운티(Jackson County)의 컴퓨터들이 감염되어 40만 달러 피해를 보았다[3]. 같은 해 5월에는 미국 볼티모어시의 병원, 공항, 현금 자동 인출기 등의 서비스가 랜섬웨어에 의해 피해를 보았으며 이를 복구하기 위해 1,800만 달러의 비용을 지불한 것으로 추산된다[4]. 국내의 경우, 2019년 5월 특정 학원 관리 프로그램이 랜섬웨어에 감염되어 해커와의 협상을 통해 부분 복구가 되었다[5].

이러한 랜섬웨어에 의한 경제적 손실을 막기 위해 다양한 랜섬웨어의 예방 및 탐지 모델에 대한 연구가 진행되어 왔다. Lu, Tianliang 등은 Crypto API 사용률이 랜섬웨어가 일반 프로그램보다 높다는 것을 착안하여 랜섬웨어를 탐지하는 모델을 제안했다[6]. Nikolai Hampton 등은 윈도우 환경에서 Windows API를 사용하는 일반 프로그램과 랜섬웨어의 빈도분석을 통해 랜섬웨어가 일반 프로그램에 비해 Crypto API 호출 빈도수가 매우 높음을 밝혀냈다[7]. 또한, Amin Azmoodeh 등은 IoT 환경에서 랜섬웨어에 감염된 디바이스의 CPU 전력 소모량이 일반 상태보다 크다는 점을 착안하여 랜섬웨어 탐지 방안을 제시하였다[8]. Natanzon 등은 LU (Logical Unit)에 저장된 I/O 의 히스토리와 랜섬웨어 감염 시의 I/O 요청횟수에 대한 패턴을 분석하여 랜섬웨어를 탐지하는 모델을 제안하였다[9]. 마지막으로 Eugene Kolodenker 등은 Crypto API의 CryptEncrypt, CryptExport, CryptSetKeyParam 등의 함수들을 후킹하여 랜섬웨어 실행 시 암호화에 사용된 키를 획득하였고, 해당 암호키를 이용하여 암호화된 파일을 복호화하는 모델을 제안하였다[10]. 위에 소개한 내용과 같이 랜섬웨어에 대한 대부분의 연구 결과는 랜섬웨어 탐지와 같은 사전 대응에 대한 내용이 주를 이룬다. 그러나 랜섬웨어 감염 후 복호화를 위한 사후대응에 관한 연구도 필요하다. 따라서 본 논문에서는 다양한 랜섬웨어 분석을 통해 암호화 프로세스를 파악하고 그에 따른 복호화 방안을 연구하였다.

본 논문에서는 2019년 상반기 점유율이 높은 랜섬웨어 3종(Gandcrab v5, Clop, Sodinokibi)과 신규 랜섬웨어 2종(Phobos, LooCipher)을 역공학 분석하여 암호화 프로세스를 분석하였다. 2장에서 랜섬웨어 5종에 대한 암호화 프로세스를 분석하며, 3장에서는 복호화가 가능한 시나리오를 분류하고 복호화 방안을 제시한다. 그리고 마지막 4장에서는 결론을 맺는다.

II. 랜섬웨어 암호화 프로세스

본 장은 랜섬웨어 5종에 대한 암호화 프로세스에 대해 서술하며, 분석을 위해 아래 Table 1과 같이 분석환경을 구축하였다. 동적 분석을 위해 OllyDbg를 사용했으며 정적 분석을 위해 IDA 및 Ghidra를 활용하였다. 암호화된 파일의 구조 분석을 위해 Hxd Editor를 사용하고 WireShark를 통해 네트워크 패킷 분석을 진행하였다. 먼저 정적 분석을 통해 암호화 프로세스 및 패킹 유무를 확인한 뒤, 동적 분석 과정을 통해 패킹 해제, 암호화된 문자열 및 암호화에 사용되는 인자들을 확인하였다.

Table 1. Analysis Environment

	Name	Role
Virtual Machine	VMware Workstation 12 PRO	-
OS	Windows 7 x86 service pack 1	-
	Windows 7 x64 service pack 1	
	Windows 10 x86 Redstone 4	
Analysis Tool	OllyDbg v1.10	disassembler
	IDA v7.0	decompiler
	Ghidra v1.9.4	decompiler
	HxD Editor v2.0	hex code editor
	WireShark v3.0.3	packet analysis

2.1 Gandcrab v5

2018년 1월 처음 등장한 Gandcrab 랜섬웨어는 서비스가 종료된 2019년 6월까지 50만 2천 건에 달하는 피해를 준 서비스형 랜섬웨어(RaaS, Ransomware as a Service)다[11]. 해당 랜섬웨어는 멀버타이징, 악성 메일 등에 의해 유포되었고, 랜섬웨어가 실행되면 사용자 PC의 로컬 드라이브뿐 아니라 로컬과 연결된 네트워크 드라이브 또한 암호화를 진행한다. Gandcrab v5.0 ~ v5.2의 암호화 프로세스는 Fig. 1과 같으며, Windows API인 Crypto API를 사용해 암호화를 진행한다.

① 제작자의 공개키 복호화
랜섬웨어 실행 파일 내에 하드코딩 되어있는 RC4 복호키와 제작자의 RSA 공개키를 RC4 암호 알고리즘과 XOR 연산자를 사용하여 복호화한다.

② 로컬 RSA 공개키, 개인키 쌍 생성
파일 암호키 암호화를 위해 CryptGenKey 함수를 사용하여 공개키와 개인키 쌍을 생성한다.

③ Salsa20 암호키, nonce 생성 및 로컬 RSA 개인키 Salsa20 암호화
Salsa20의 암호키와 nonce를 CryptGenRandom 함수를 사용하여 생성한다. 그 후, 로컬 RSA의 개인키를 Salsa20의 암호키와 nonce를 사용하여 암호화한다.

④ 공격자의 공개키로 Salsa20 암호키, nonce 암호화 및 레지스트리 저장
③에서 사용한 암호키 및 nonce를 CryptEncrypt 함수를 사용하여 공격자의 공개키로 암호화한다. 암호화된 Salsa20 암호키, nonce 및 로컬 개인키는 레지스트리에 저장된다.

⑤ Salsa20 암호키와 nonce 생성 및 파일 암호화
CryptGenRandom 함수를 사용하여 Salsa20의 암호키와 nonce를 생성하고 파일을 암호화한다.

⑥ Salsa20 암호키와 nonce 생성 후 RSA로 암호화 및 EOF (End Of File)에 저장
파일 암호화에 사용한 Salsa20의 암호키 및 nonce를 로컬에서 생성된 RSA 공개키로 암호화하고 암호화된 파일의 EOF에 저장한다.

2.2 Sodinokibi

2019년 4월 Oracle Weblogic의 취약점(CVE-2019-2725)을 악용한 Sodinokibi 랜섬웨어가 등장했다. 해당 랜섬웨어는 Gandcrab과 유사한 방식으로 유포되며, 로컬 PC뿐 아니라 로컬과 연결된 네트워크 드라이브 또한 암호화를 진행한다[12]. 본 랜섬웨어의 경우 파일 암호키 생성 및 파일 암호

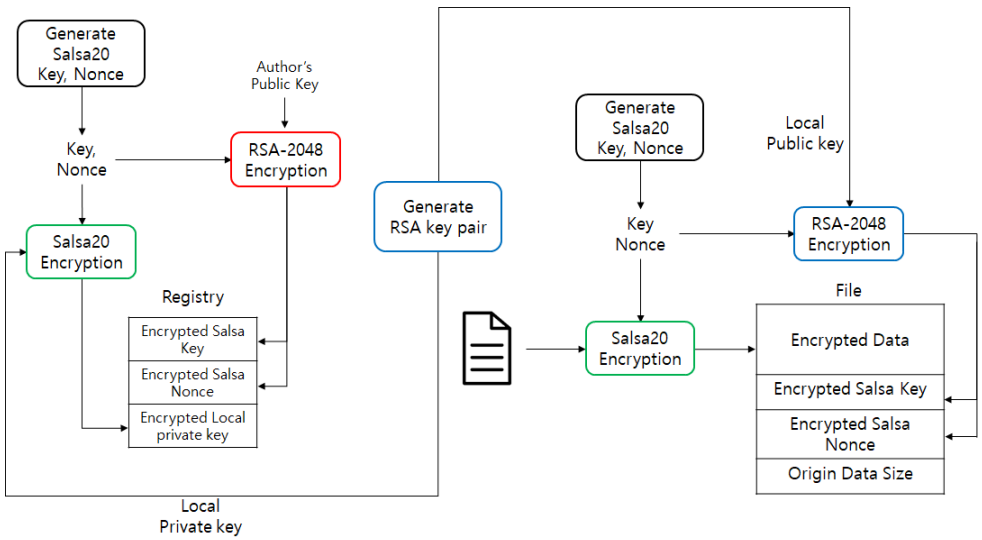

Fig. 1. Encryption Process of Gandcrab Version 5

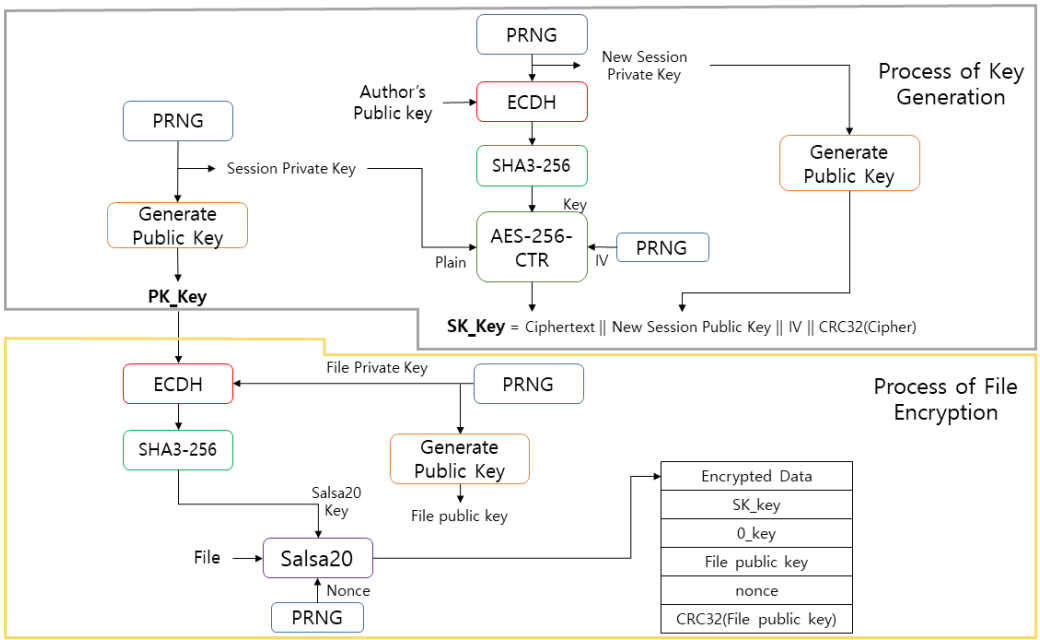

Fig. 2. Encryption Process of Sodinokibi

키 암호화를 위해 ECIES (Elliptic Curve Integrated Encryption Scheme)를 이용하고 암호화 과정에서 사용되는 모든 알고리즘은 Crypto API를 사용하지 않고 제작자가 직접 구현하였다. 해당 랜섬웨어의 암호화 상세 과정은 Fig. 2와 같다.

① Session Private Key 및 Session Public Key 생성

PRNG (Pseudo Random Number Generator)를 이용하여 Session Private Key를 생성한다. 그 후 생성된 Session Private Key와 Base를 인자로 사용하는 X25519 연산을 통해 Session Public Key를 생성한다. Session Public Key는 레지스트리에 PK_Key 이름으로 저장된다.

② New Session Private Key 및 Session Public Key 생성

①의 과정과 동일한 방식으로 New Session Private Key 및 New Session Public Key를 생성한다. New Session Private Key와 공격자의 Public Key를 ECDH (Elliptic Curve Diffie-Hellman)를 통해 공유키를 생성하고 그 값을 SHA3-256으로 해싱한다. 해싱한 값은 AES-CTR-256의 암호키로 사용한다.

③ Session Private Key 암호화

②의 과정으로 생성된 암호키를 사용하여 Session Private Key를 AES-256-CTR로 암호화한다. 해당 Session Private Key는 파일 복호화 시, File Public Key와 ECDH를 사용하여 파일 복호화 키를 만들 수 있다.

④ File Private Key 및 File Public Key 생성

각 파일에 대해 공개키와 개인키 쌍을 생성한다. 생성과정은 ①의 과정과 동일하며 각 파일에 대해 서로 다른 키 쌍을 생성한다.

⑤ Salsa20 암호키 생성 및 파일 암호화

Session Public Key와 File Private Key는 ECDH를 통해 공유되며, 해당 정보를 기반으로 공유키를 생성한다. 공유키를 SHA3-256으로 해싱한 값을 Salsa20의 암호키로 사용하여 파일을 암호화한다.

2.3 Clop

2019년 2월 등장한 Clop 랜섬웨어는 개인이 아닌 국내 AD (Active Directory) 서버를 가진 기업을 타깃으로 피해를 주고 있는 랜섬웨어다[13]. 해당 랜섬웨어가 실행되면, 네트워크 드라이브를 탐지하여 서버에 연결된 모든 드라이브를 암호화한다. RC4 알고리즘을 제외한 키 생성 및 공개키 암호화에서 Crypto API를 사용하였다. Clop 랜섬웨어의 암호화 동작과정은 Fig. 3과 같다.

① RC4 암호키 생성

파일 암호화에 필요한 RC4 암호 알고리즘의 암호키를 CryptGenKey 함수를 사용하여 생성한다. RC4 암호키는 단일 파일에 대해 한 번만 사용하고 사용이 끝나면 파기하고 새로운 암호키를 생성한다.

② 파일 암호화

①의 과정을 통해 생성된 RC4 암호키를 이용하여 RC4로 파일을 암호화한다. 만일 파일의 크기가 300,000 bytes보다 크면, 300,000 bytes만 암호화를 진행한다.

③ RC4 암호키 암호화

파일 암호화에 사용된 RC4 암호키는 랜섬웨어 내에 하드코딩된 RSA 공개키로 암호화하여 암호화된 파일의 끝에 저장한다.

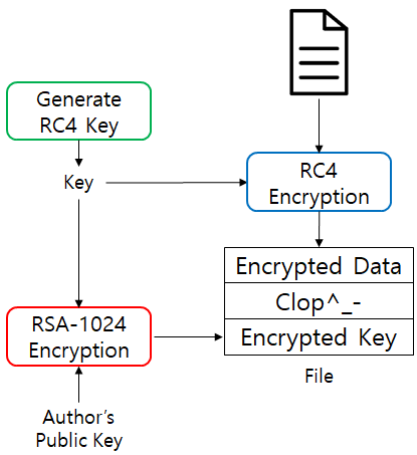

Fig. 3. Encryption Process of Clop

2.4 Phobos

Dharma와 Crysis 랜섬웨어의 변종인 Phobos는 2019년 1월 처음 발견되었다. 해당 랜섬웨어는 RDP (Remote Desktop Protocol) 방식을 사용하여 사용자 몰래 악성 파일을 실행시켜 파일 암호화를 진행한다[14]. 해당 랜섬웨어의 파일 암호화에 사용된 AES 알고리즘은 CryptoAPI를 사용하여 구현했지만, RSA 알고리즘을 CryptoAPI를 사용하지 않고 외부 라이브러리를 사용한다[15]. Phobos 랜섬웨어의 암호화 동작 과정은 다음 Fig. 4와 같다.

① AES 암호키 생성 및 암호키 RSA-1024 암호화

CryptGenRandom 함수를 통해 AES 암호키를 생성한다. AES 암호키는 암호화 기능을 가진 스레드마다 한 번씩만 생성되며 단일키로 사용한다. 이 암호키는 제작자의 공개키를 이용하여 RSA-1024로 암호화된다.

② IV 생성 및 AES-256-CBC 파일 암호화

개별 파일에 대해 AES-256-CBC에 사용되는 IV를 CryptGenRandom 함수를 통해 생성한다. IV와 ①의 과정을 통해 생성된 AES 암호키를 사용하여 파일을 암호화한다. 사용된 IV와 암호화된 AES 암호키는 EOF에 순차적으로 저장한다.

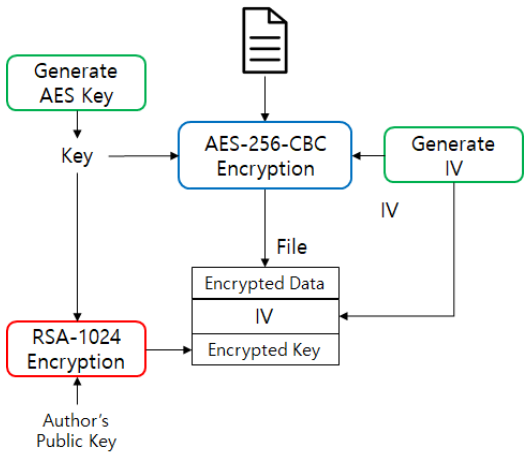

Fig. 4. Encryption Process of Phobos

2.5 LooCipher

2019년 6월 발견된 LooCipher는 스팸 메일을 통해 워드 문서를 다운받도록 요구한 뒤 해당 문서를 열면 매크로를 통해 해당 랜섬웨어를 다운받는다[16]. 해당 랜섬웨어는 C++ 기반 오픈소스 암호화 클래스 라이브러리인 Crypto++을 사용하여 파일을 암호화하고 공개키 기반 알고리즘을 사용하지 않는다. 그리고 취약한 난수 생성기를 통해 생성된 단일 암호키에 대해 모든 파일을 암호화한다. LooCipher의 암호화 동작 과정은 다음 Fig. 5와 같다.

① AES-128-ECB 암호키 생성

Fig. 6과 같이 현재 시간을 seed로 사용하는 rand 함수를 통해 생성된 난수를 73개의 문자로 대응시켜 16 bytes의 문자열을 생성한다. 그 후 Random_shuffle 함수를 사용하여 문자열을 무작위 배치한 뒤 AES의 암호키로 사용한다. 이 파일 암호키는 단일키로, 모든 파일에 대해 사용된다.

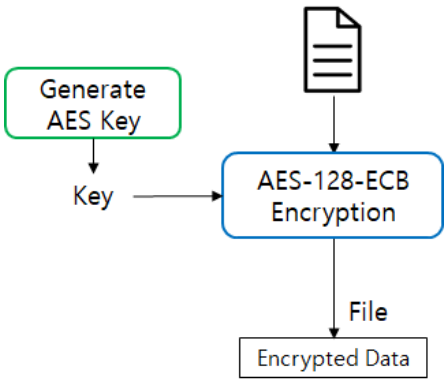

Fig. 5. Encryption Process of LooCipher

```
seed = time_64(0);
j__srand(seed);
memcpy(&v10, (int)"!@#%&?+-*/=_0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
v11 = 0;                abcdefghijklmnopqrstuvwxyz");
sub_5F5208((int)&v9);                          73 Characters
LOBYTE(v11) = 1;
while ( getStringLength((int)&v9) != a2 )
{
  v3 = j__rand();
  v8 = v3 % (getStringLength((int)&v10) - 1);
  v4 = sub_60290D((int)&v7, v8, 1);
  LOBYTE(v11) = 2;
  sub_5FA208(v4);
  LOBYTE(v11) = 1;
  sub_6051D0(&v7);
}
                    ⋮
random_shuffle_wrapper((char)v55, (int)v56, (int)v57, (char)v58);
```

Fig. 6. Code of Encryption key Generation

```
while ( length >= 16 )
{
  v48 = *(__m128i *)inBlocks;
  v49 = v48;
  if ( v81 )
  {
    v47 = *xorBlocks;
    v46 = _mm_xor_si128(v49, v47);
    v49 = v46;
  }
  if ( flags & 1 )
    ++*(_BYTE *)(inBlocks + 15);
  AESNI_Enc_Block(&v49, subkeys, rounds)
```

Fig. 7. AES Algorithm Using Crypto++ Library

② 파일 암호화

①의 과정을 통해 생성된 암호키를 사용하여 AES로 파일을 암호화한다(Fig. 7).

해당 랜섬웨어들의 분석 결과는 Table 2와 같다. LooCipher를 제외한 나머지는 모두 공개키를 사용하여 파일 암호키를 암호화한다. 또한, Sodinokibi는 ECIES를 사용하여 암호 강도를 높였으며 Gandcrab, Sodinokibi 그리고 Clop의 경우 스트림 암호를 사용한다. 이는 암호화할 데이터의 크기가 클 때 기존의 사용되던 AES 알고리즘보다 속도가 빨라 파일 암호화에 적합하기 때문이다[17]. 또한, Gandcrab 내의 Salsa20, Sodinokibi의 모든 암호 알고리즘, Clop의 RC4, Phobos의 RSA, LooCipher의 AES는 구현 시 Crypto API를 사용하지 않고 Crypto++과 같은 외부 라이브러리를 사용하거나 제작자가 직접 구현했다. 이는 [6,7,10]에서와 같이 Crypto API를 이용하여 랜섬웨어를 탐지하는 AV (Anti-Virus)를 우회하기 위함이다.

III. 복호화 방안

본 연구결과, 분석한 랜섬웨어 중 Gandcrab과 LooCipher는 감염된 데이터의 복호화가 가능하였다. 본 장에서는 Gandcrab과 LooCipher를 기반으로 복호화 방안에 대해 소개 및 제시한다.

3.1 취약한 난수생성기 재현

본 절에서는 2019년 6월 말에 발견된 신규 랜섬웨어인 LooCipher의 분석 결과를 토대로 암호키 재현 가능성을 확인하고 복호화 방안을 제시한다. 앞서

Table 2. The Result of Analysis Ransomware

		File Encryption	Encryption of File Encryption Key	Decryptable
Gandcrab v5.0~v5.2	Algorithm	Salsa20	RSA-2048	Exposed author's private key
	Encryption API	implemented by author	CryptEncrypt	
	Key Generation API	CryptGenRandom	CryptGenKey	
Sodinokibi	Algorithm	Salsa20	AES-256-CTR	X
	Encryption API	implemented by author		
	Key Generation API	ECDH implemented by author		
Clop	Algorithm	RC4	RSA-1024	X
	Encryption API	implemented by author	CryptEncrypt	
	Key Generation API	CryptGenKey	Used hard-coded public key	
Phobos	Algorithm	AES-256-CBC	RSA-1024	X
	Encryption API	CryptEncrypt	Used external library	
	Key Generation API	CryptGenRandom	Used hard-coded public key	
LooCipher	Algorithm	AES-128-ECB	X	Used vulnerable random generator
	Encryption API	Rijndael::Enc::AdvancedProcessBlocks in Crypto++ Library	X	
	Key Generation API	rand	X	

2.5절에서 언급한 바와 같이 LooCipher의 암호키는 현재 시스템 시간을 seed로 사용하는 취약한 난수 생성기인 rand 함수를 사용하여 생성된다. 따라서 키 생성 당시의 사용된 시스템 시간을 재현한다면 해당 암호키의 복구가 가능하다.

LooCipher 랜섬웨어의 복호화 과정은 Fig. 8과 같다. 암호화된 파일의 수정시간을 얻어 암호키 후보군을 생성한다. 이후 키 후보군을 사용하여 복호화를 진행한다. 만약 복호화한 결과가 해당 원본 파일의 고유 시그니처를 갖는다면 모든 파일을 대상으로 복호화를 진행한다.

암호키 후보군을 생성할 때 암호키의 생성과정은 Fig. 9와 같다. rand 함수의 seed를 감염된 파일의 수정시간으로 설정하고 하드코딩된 73개의 문자를 대응하여 16 bytes의 문자열로 만든다. 생성된 16 bytes의 문자열을 random_shuffle 함수를 사용하여 암호키를 생성한다.

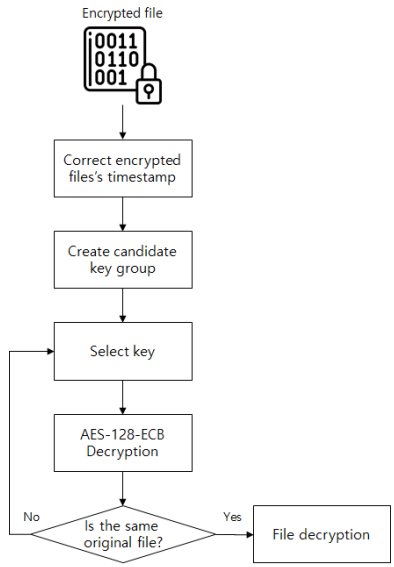

Fig. 8. Decryption Process of LooCipher

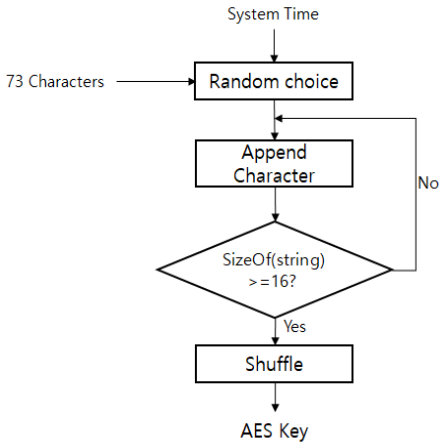

Fig. 9. Generation Process of File Encryption Key

제안한 방법으로 구현하여 암호화된 파일을 복호화한 결과, 암호화된 파일의 복호화에 성공하였다. 키 생성과정에서 사용된 시스템 시간의 단위는 초이며, 키 생성시간과 파일의 수정시간의 차는 최대 2시간(7,200초)을 넘지 않는다. 따라서, 키 무차별 대입 공격을 사용할 경우 2^{128}회의 연산량을 갖지만 제안한 방법을 사용하면 최대 7,200회 이내로 키를 찾을 수 있다.

3.2 메모리 포렌식을 통한 암호키 획득

LooCipher의 경우 파일 암호키를 생성하고 파일 암호화가 끝나면 메모리상에서 파일 암호키를 파기하여 메모리에서 암호키 추출을 못 하도록 한다. 하지만 Fig. 10과 같이 감염 PC에서 파일 암호키를 C2 (Command & Control) 서버와 통신하는 과정에서 주고받는 패킷이 메모리에 남아 키 복구가 가능하다.

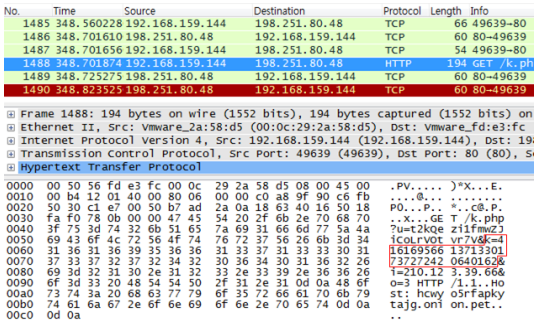

Fig. 10. File Encryption Key Sent in Plaintext to The C2 Server

따라서 LooCipher 랜섬웨어의 감염이 끝난 후 LooCipher의 프로세스 메모리를 추출하고 메모리 내 C2 서버에 보낸 Request를 파싱하여 파일 암호키를 찾을 수 있다(Fig. 11). 해당 방법의 경우, 휘발성 메모리에서 암호키를 찾기 때문에 시스템이 재부팅되거나 해당 프로세스 메모리 영역이 덮어진다면 키 획득이 불가능하다는 단점이 있다.

3.3 암호키 유출

파일 복호화를 위한 암호키는 해당 랜섬웨어의 제작자가 키 관리를 소홀하게 하는 경우 파일 암호키 또는 공개키로 암호화된 파일 암호키를 복호화하기 위한 개인키가 유출될 수 있으며 유출된 키를 사용하여 파일을 복호화할 수 있다. 본 절에서는 Gandcrab의 사례를 들어 암호키 유출 사례를 설명한다. Gandcrab의 복호화 프로세스는 Fig. 12와 같다. 제작자의 RSA 개인키를 통해 레지스트리에 저장된 암호화된 Salsa20의 암호키 및 nonce를 복호화한다. 그 후 복호화된 암호키 및 nonce를 사용하여

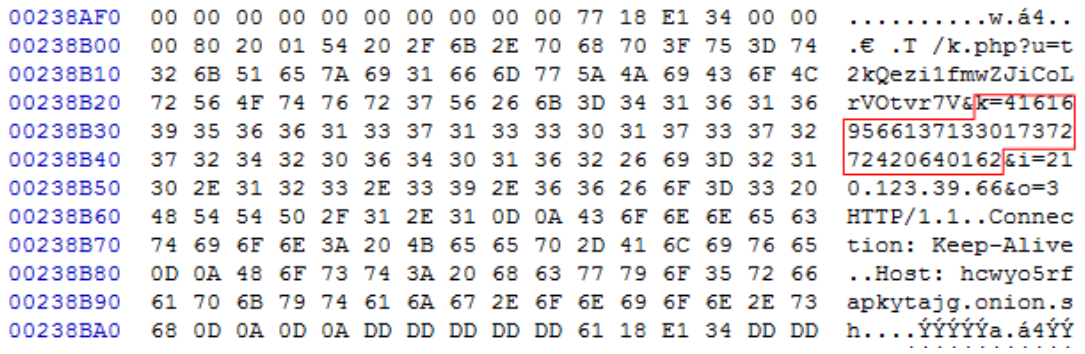

Fig. 11. File Encryption Key Left in Volatile Memory

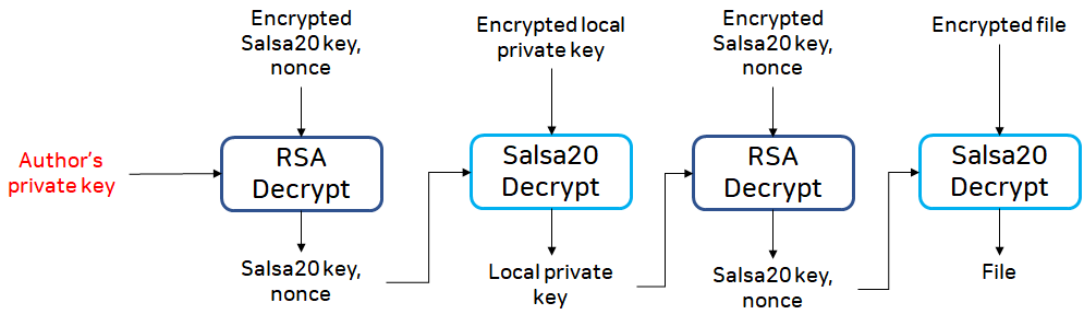

Fig. 12. File Decryption Process of Gandcrab version 5

암호화된 로컬 RSA 개인키를 Salsa20으로 복호화한다. 복호화된 로컬 RSA 개인키를 이용하여 각 파일에 대해 암호화된 Salsa20 파일 암호키 및 nonce를 복호화하여 파일을 복호화 할 수 있다.

Gandcrab의 경우 파일 암호화에 사용된 Salsa20의 암호키 및 nonce를 CSPRNG (Cryptographically Secure Pseudo Random Number Generator)를 사용하여 생성한다. 또한, 사용된 파일 암호키 및 로컬 개인키는 사용 후, 메모리에서 삭제되므로 제작자의 RSA 개인키를 알지 못하면 복호화가 어렵다. 그러나 예외적으로 Gandcrab 제작자가 2018년 10월경 시리아를 대상으로 RSA 개인키를 공개하여 시리아를 한정으로 파일 복구가 가능하다[18]. 또한, 2019년 6월경 Bitdefender, FBI, Europol 등 보안업체와 각 나라의 수사기관이 협력하여 C2 서버에 존재하는 제작자의 개인키 취득에 성공하여 현재 v2, v3를 제외한 v5.2까지 복호화가 가능하다[19].

IV. 결 론

본 논문에서는 2019년 주요 랜섬웨어 및 신규 랜섬웨어 5종에 대해 암호화 과정을 분석하고 복호화가 가능한 시나리오를 제시하였다. 공개키를 사용한 랜섬웨어는 제작자의 개인키가 유출되지 않는다면 복호화 가능성이 작지만, 암호키 생성과정 중 취약한 난수 생성기를 사용한 경우 난수 생성기의 seed를 재현해 암호키 복구 및 파일 복호화가 가능하다. 또한, 제작자가 파일 암호키를 메모리에서 삭제하지 않는 경우, 시스템 재부팅 혹은 해당 프로세스의 메모리가 변조되지 않는다면 메모리에서 암호키 복원이 가능하다.

암호화 알고리즘은 AES뿐만 아니라 스트림 암호를 사용하는 경우도 존재하는데, 이는 Crypto++과 같이 외부 라이브러리를 사용함으로써 윈도우에서 CryptAPI의 사용으로 인한 AV 탐지를 우회하기 위해 사용된 것으로 보인다. 또한 ECIES와 같이 보안 강도가 높은 암호 스킴을 채택함으로써 보안성을 높였다. 향후 나올 랜섬웨어는 암호화 과정이 더욱 복잡하고 보안성이 높아져 복호화가 가능한 랜섬웨어의 수가 줄어들 것이다. 따라서 암호화 프로세스 및 복호화 방안 연구가 활발히 논의되어야 할 것이다.

References

[1] "The Dark Web Created by High Technology", Ntoday, 3. Mar. 2019 http://www.ntoday.co.kr/news/articleView.html?idxno=65667

[2] "Norsk Hydro hits LockerGoga, too late to release financial report", BoanNews, https://www.boannews.com/media/view.asp?idx=78806, Apr. 2019

[3] "Jackson County, Georgia Attacked Ransomware, decided to pay $400,000", DailySecu, https://www.dailysecu.com/news/articleView.html?idxno=46598, Mar. 2019

[4] "Baltimore City infected with ransomware", Etnews, http://www.etnews.com/20190607000209, Jun. 2019

[5] "Domestic specific school management program infected by Ransomware, "BoanNews, https://www.boannews.com/media/view.asp?idx=79785, May 2019

[6] Lu, T.; Zhang, L.; Wang, S.; Gong, Q. Ransomware Detection based on V-detector Negative Selection Algorithm. In Proceedings of the 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Shenzhen, China, 15 - 17 December 2017; pp. 531 - 536

[7] Hampton, N.; Baig, Z.; Zeadally, S. Ransomware Behavioural Analysis on Windows Platforms. J. Inf. Secur. Appl. 2018, 40, 44 - 51.

[8] Azmoodeh, Amin, et al. "Detecting crypto-ransomware in IoT networks based on energy consumption footprint." Journal of Ambient Intelligence and Humanized Computing 9.4 (2018): 1141-1152.

[9] Natanzon, Assaf, et al. "Ransomware detection using I/O patterns." U.S. Patent Application No. 15/275,759.

[10] Kolodenker, E.; Koch, W.; Stringhini, G.; Egele, M. PayBreak: Defense against Cryptographic Ransomware. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, UAE, 2 - 6 April 2017; pp. 599 - 611.

[11] AhnLab, "2019 ransomware trends" https://asec.ahnlab.com/1241, Jul. 2019

[12] KrCERT, "Sodinokibi", https://www.boho.or.kr/data/trendView.do?bulletin_writing_sequence=35018, May 2019

[13] "BoanNews", https://www.boannews.com/media/view.asp?idx=80028, May 2019

[14] Checkmal Blog, "Phobos Ransomware", http://blog.naver.com/PostView.nhn?blogId=checkmal&logNo=221520619442&parentCategoryNo=&categoryNo=7&viewDate=&isShowPopularPosts=true&from=search, Apr. 2019

[15] Github, "Crypto", https://github.com/joyent/syslinux/blob/master/gpxe/src/crypto/axtls/rsa.c, Mar. 2019

[16] Bleeping Computer, "LooCipher", https://www.bleepingcomputer.com/news/security/new-loocipher-rans omware-spreads-its-evil-through-spam/, Jun. 2019

[17] C. De Canniere, "eSTREAM Software Performance," LNCS vol. 4986, pp. 119-139, 2008

[18] Bleeping Computer, "Gandcrab" https://www.bleepingcomputer.com/news/security/gandcrab-devs-release-decryption-keys-for-syrian-victims/, Oct. 2019

[19] Bleeping Computer, "Gandcrab" https://www.bleepingcomputer.com/news/security/release-of-gandcrab-52-decryptor-ends-a-bad-ransomware-story, Jun. 2019

〈저자소개〉

이 세 훈 (Sehoon Lee) 학생회원
2019년 2월: 경북대학교 전자공학부 졸업
2019년 3월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 디지털 포렌식, 정보보호 등

윤 병 철 (Byungchul Youn) 학생회원
2019년 2월: 국민대학교 수학과 졸업
2019년 9월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 디지털 포렌식, 정보보호

김 소 람 (Soram Kim) 학생회원
2016년 2월: 국민대학교 수학과 졸업
2018년 2월: 국민대학교 금융정보보안학과 석사
2018년 3월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 디지털 포렌식, 정보보호

김 기 윤 (Giyoon Kim) 학생회원
2019년 2월: 국민대학교 정보보안암호수학과 졸업
2019년 3월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 디지털 포렌식, 암호학 등

이 영 주 (Yeongju Lee) 종신회원
2017년 2월: 전남대학교 지리학과 졸업
2018년 6월~현재: 한국인터넷진흥원
〈관심분야〉 암호학, 디지털 포렌식, 악성코드 분석

김 대 운 (Daeun Kim) 종신회원
2015년 2월: 전남대학교 컴퓨터공학과 졸업
2017년 2월: 전남대학교 정보보안협동과정 석사
2017년 3월~현재: 한국인터넷진흥원
〈관심분야〉 암호학, 디지털 포렌식, 악성코드 분석

박 해 룡 (Haeryong Park) 종신회원
1992년~1999년: 전남대 수학과 학사
1999년~2001년: 서울대 수리과학과 석사
2004년~2006년: 전남대 정보보안학과 박사
2000년~현재: 한국인터넷진흥원
〈관심분야〉 정보보안, 암호기술

김 종 성 (Jongsung Kim) 종신회원
2000년 8월/2002년 8월: 고려대학교 수학 전공 학사/이학석사
2006년 11월: K.U.Leuven, ESAT/SCD-COSIC 정보보호 전공 공학박사
2007년 2월: 고려대학교 정보보호대학원 공학박사
2007년 3월~2009년 8월: 고려대학교 정보보호기술연구센터 연구교수
2009년 9월~2013년 2월: 경남대학교 e-비즈니스학과 조교수
2013년 3월~2017년 2월: 국민대학교 수학과 부교수
2014년 3월~현재: 국민대학교 일반대학원 금융정보보안학과 부교수
2017년 3월~현재: 국민대학교 정보보안암호수학과 부교수
〈관심분야〉 정보보호, 암호 알고리즘, 디지털 포렌식