Open access • Book Chapter • DOI:10.1007/978-3-319-54876-0_2

Ransomware and the Legacy Crypto API — Source link

Aurélien Palisse, Hélène Le Bouder, Jean-Louis Lanet, Colas Le Guernic ...+1 more authors

Institutions: French Institute for Research in Computer Science and Automation

Published on: 05 Sep 2016 - Conference on Risks and Security of Internet and Systems

Topics: Ransomware and Malware

Ransomware and the Legacy Crypto API

Aurélien Palisse, Hélène Le Bouder, Jean-Louis Lanet, Colas Le Guernic, Axel Legay

Ransomware and the Legacy Crypto API

Aurélien Palisse[1], Hélène Le Bouder[1], Jean-Louis Lanet[1], Colas Le Guernic[1,2], and Axel Legay[1]

[1] High Security Laboratory - INRIA TAMIS, Rennes, France
[2] DGA Maîtrise de l'Information, Bruz, France

Abstract. Ransomware are malicious software that encrypt their victim's data and only return the decryption key in exchange of a ransom. After presenting their characteristics and main representatives, we introduce two original countermeasures allowing victims to decrypt their files without paying. The first one takes advantage of the weak mode of operation used by some ransomware. The second one intercept calls made to Microsoft's Cryptographic API. Both methods must be active before the attack takes place, and none is general enough to handle all ransomware. Nevertheless our experimental results show that their combination can protect users from 50% of the active samples at our disposal.

Keywords: Malware, Ransomware, ECB mode, Replay attacks, Microsoft's Cryptographic API.

1 Introduction

Motivation. Ransomware constitute a hot topic since they are rapidly spreading. The number of ransomware victims has increased significantly recently as shown in reports of various actors [1–3]. For example, Kaspersky [4] observes a 5-fold increase in the number of individuals infected between 2012 and 2015. Moreover victims are not limited to individuals, but important organizations are targeted too, such as hospitals [5, 6].

Despite this, to the best of our knowledge there are only three scientific publications about ransomware. In our opinion, too few works have been done on this topic that is rapidly becoming one of the main security threat.

State of the art. The first publication [7] demonstrates how to make an experimental ransomware with the Microsoft's Cryptographic API. At that time cryptographic libraries and capabilities were generally restricted by governments's legislation. The first solution against its malicious usage was zero-knowledge authentication. A second proposal was to allow public-key encryption only for trusted certificates verified by the kernel (and checked against revocation lists).

In fact the first ransomware could be bypassed by reverse-engineering [8]. At that time, the motto in ransomware development was: "*few investments, few incomes, few risks*". So only weak custom cryptography, RC4 encryption, or factorisable RSA key were used.

A more recent research [9], proposes an efficient and practical approach for ransomware detection. The authors perform an insight analysis on file system activities during infection. Distinguishable patterns have been found concerning Input/Output (I/O) operations on top of the Windows file system driver. Their solution is to place a filter driver between userland applications and the NTFS driver to intercept all I/O Request Packets (IRPs), to block ongoing attacks. Alternatively they suggest to monitor the Master File Table (MFT) with the advantage that files can be recovered. Unfortunately none of these protections have been implemented or experimentally evaluated.

Contribution. In this paper, new protections against ransomware are presented and we make a comparative study of the threat evolution dedicated to filecryptor. The first protection is based on the principle of a replay attack [10]. The main idea is to benefit from a weak chaining mode with cipher algorithm against the ransomware. The second protection uses the Microsoft's Cryptographic API to prevent malicious alteration of user files.

Organization. This paper is organized as follows. Section 2 introduces ransomware. A collection of ransomware and our classification are described in section 3. Then our protections are detailed in section 4 and 5. In section 6, experimental results are presented. Finally the conclusion is drawn in section 7.

2 Ransomware Overview

2.1 Definition

A malware is a software designed by an attacker to perform undesirable actions on the victim's computer, and usually without the knowledge of legitimate users. A ransomware is a form of malware that prevent legitimate users from accessing their device or data and asks for a payment in exchange for the stolen functionality. They have been used for mass extortion in various forms, but the most successful seem to be encrypting ransomware: most of the user data are encrypted and the key can be retrieved with a payment to the attacker.

To be widely successful a ransomware must fulfill three properties [8]:

Property P1: The hostile binary code must not contain any secret (*e.g.* deciphering keys). At least not in an easily retrievable form, indeed white box cryptography can be applied to ransomware [11].

Property P2: Only the author of the attack should be able to decrypt the infected device.

Property P3: Decrypting one device can not provide any useful information for other infected devices, in particular the key must not be shared among them.

2.2 Infection distribution

The most common infection vectors to data have been through malicious email attachments, compromised software and drive-by downloads[3] exploit kits [12, 13]. Other vectors include malvertisements,[4] hacking or social engineering. It can also be downloaded as a payload by another malware.

2.3 Payment method

The goal of a ransomware is to earn money. First ransomware used bank transfers and prepaid cards (MoneyPak, Amazon and Apple gift cards). In rare cases SMS or call to a overtaxed premium mobile number have been encountered. These methods are traceable by law enforcement agencies, thus large-scale campaigns were limited in order to not attract attention. That is why, new generation of ransomware relies on Bitcoin (BTC) almost exclusively. Bitcoin release has undeniably stimulated ransomware threats for massive attacks thanks to: confidentiality, rapidity of transfer, absence of central banking group. The Bitcoin protocol have been extended in 2014 and can be used to store 80 bytes not related to the transaction. The new variant of CTB-Locker uses this field as a side channel to send back the decryption keys once the ransom paid [14].

2.4 Command and Control

In order to fulfill property P2, the communication with the attacker is necessary. This is usually realize through a Command and Control (C&C) server. The minimal functionality is to provide the decryption key. In order to fulfill property P3, decryption key must be unique to each victim. Thus a unique identifier is computed. Different kinds of ransomware have been found, some of them require an initial connection to C&C in order to generate per-user key-pair or just to check its availability. A static central C&C can be easily disabled by law enforcement. Malware authors can circumvent this by Domain Generation Algorithm (DGA). Still it is not rare to analyze ransomware which tries to contact C&C without success, so their malicious payload stay inactive.

2.5 Obfuscation

Ransomware authors pack and obfuscate their payload to bypass anti-virus detection and evade analysis tools. Some modern ransomware are highly sophisticated pieces of code. Malware authors use Do-It-Yourself (DiY) malware kits and additional armoring techniques to generate new executable files based on the same code. An infinite number of variants based on an original sample can be generated. Automating the generation is named malware factory and can

[3] Drive-by download is a term used to describe how a piece of malware is installed on a user's computer without his knowledge when browsing a compromised website.

[4] Porte-manteau of malware and advertisement.

be compared to an assembly line. Some ransomware use malware factories techniques and morphing code to defeat hash based signature. New variants of Cerber are generated every 15 seconds thanks to on the fly server-side factories [15].

2.6 File encryption

The seminal paper of Youg *et al.* [7] describes the use of both symmetric and asymmetric cryptography in conjunction with computer virus and trojan horse technology. They demonstrated with an experimental ransomware this concept on a Windows target using Triple DES [16] and RSA [17] implementation with a 1024-bits public-key. This proof of concept is based on public cryptographic services given to userland applications through the Microsoft's Cryptographic API (MS CAPI) framework. Distributed in 1996, MS CAPI provides a high degree of abstraction and significantly facilitates development of software with cryptographic primitives needs.

Implementation of standard algorithms (DES, Triple DES, RSA, PGP) in the 90s were not publicly available due to government legislation. At that time, cryptovirus required some crypto experts, and multiple precision library to manipulate large integer. The OpenSSL project only started two years later in 1998.

Nowadays plenty of free third-party cryptographic libraries can be used by criminals for malicious encryption purposes (OpenSSL, mbed TLS, libsodium). More recent cryptographic designs involve Elliptic Curve Cryptography (ECC) for shared secret and per-victim key-pair generation [18]. File encryption is performed almost exclusively by Advanced Encryption Standard (AES) with recent ransomware, difference can be pointed only on the chaining mode used. Ransomware authors might use the default setting of the system (MS CAPI) or library. In most situations files encryption is performed only superficially. The main objective is to make files unusable. In the remaining of the section, we present the two most used chaining mode.

Block cipher modes of operation A block cipher encrypts data of fixed-length n bytes. If the data are bigger than this fixed-length, data are split in different block of size n bytes each. If the size of data is not a multiple of n bytes, the last block is padded. The mode of operation describes how to repeatedly apply the encryption. The ransomware used two classic modes: Electronic Codebook (ECB) mode, Cipher Block Chaining (CBC) mode. In this paper, $\mathcal{F}$ denotes an encryption algorithm, T a plain-text, C a cipher-text and K a cipher-key. Plain-text is split in a set of blocks B_i of size n bytes.

Electronic Codebook mode The ECB mode encryption is a simple mode. The data are divided into blocks, and each block is encrypted separately (1).

$$C_i = \mathcal{F}(B_i, K_i) \quad . \tag{1}$$

Figure 1 illustrates this mode.

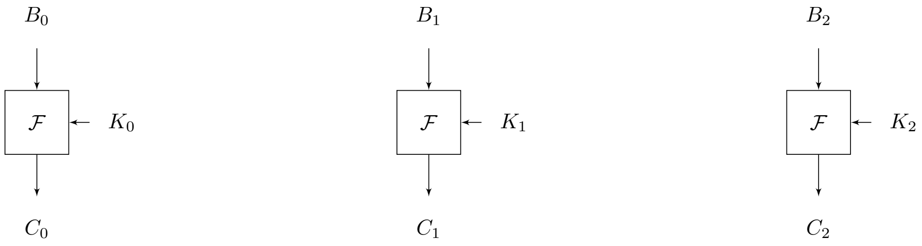

Fig. 1: Scheme of ECB mode encryption

Cipher Block Chaining mode In CBC mode, each block of plain-text is xored with the previous cipher-text block before being encrypted. An Initialization Vector (IV) is xored with the first block B_0. The relation is defined in (2).

$$\begin{cases} C_0 = & \mathcal{F}\left(B_0 \oplus IV, K_0\right) \\ C_{i+1} = & \mathcal{F}\left(B_{i+1} \oplus C_i, K_i\right) \end{cases} . \tag{2}$$

Figure 2 illustrates this mode.

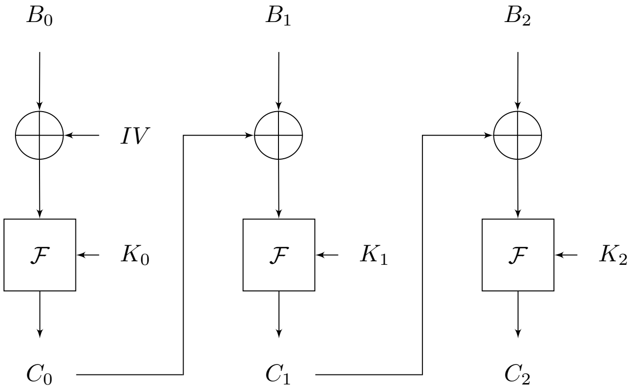

Fig. 2: Scheme of CBC mode encryption

Infection mitigation In order to mitigate the attacks, it is mandatory to realize backups daily to reduce the gap between the systems and the snapshots. The ransomware targets the network drives. Some attacks against Synology storage products have been done through a vulnerability exploited by the SynoLocker ransomware in 2014 [19]. To reduce the risks, off-site or cold backups need to be performed.

3 Taxonomy

With the advent of cryptographic libraries and APIs, the last decade saw a steady increase in the number of ransomware. We present a few of them in this section and try to address the most significant improvements from old to state of the art ransomware.

3.1 Ransomware Collection

One of the first studies about ransomware was published in December 2004 by Nazarov *et al.* [20]. This report make an insight analysis of **PGPCoder/GPCode**. This ransomware was the first implementation of the model proposed by Young *et al.* encrypting over 80 different types of files on the disk.

CryptoLocker was first reported in September 2013 [21]. It is the first to implement all of the properties P1, P2 and P3. Files are encrypted using AES with a random key which is then encrypted thanks to a 2048-bits RSA public key. The corresponding private key, needed to decrypt the AES key, can be obtained by paying the ransom. Distribution was initially through spam messages containing malicious attachments. Then the main targets were the victims of the botnet Gameover ZeuS. It was disrupted by international law enforcement agencies during the authorities operation Tovar [22] in June 2014.

CryptoWall appeared in early 2014 [23]. It continues to gain notoriety and is still intensively developed [24]. It encrypts files using RSA and uses The Onion Router (Tor)[5] to obfuscate communications with the C&C. Each victim is registered and identified by a unique identifier and get from the C&C the corresponding RSA public-key. Infection vector was performed initially via exploit kits and then moved to more traditional spam email campaigns. File encryption does not occur if the RSA public-key is not received. It is one of the first requiring payment in Bitcoin only, and deleting the Shadow Copy. Moreover a list of all the encrypted files is stored in the registry to ease the decryption process.

Version 3.0 upgrades to AES symmetric encryption with Cipher Block Chaining (CBC) mode for files and communication with C&C was done over the Invisible Internet Project (I2P)[6] network. Due to the lack of reliability, they back up with a combination of compromised sites forwarding traffic to Tor server.

In version 4.0, an additional step in disruption is made by renaming files, including extension, with random characters. In order to avoid encrypting the same file again following superinfection, a hash corresponding to the RSA public-key given to the victim is prepended to the beginning of the file.

TorrentLocker ransomware was found in February 2014 [25] and shares many similarities with CryptoLocker. It adds captcha code and redirection to a

[5] `https://www.torproject.org/`

[6] `https://geti2p.net/fr/`

spoofed site for infection. At the beginning, AES encryption was performed on all files with the same key and Initialization Vector (IV) using Counter (CTR) mode [26]. This can be exploited with one known plain-text and cipher-text in order to extract the keystream and then apply it to a cipher-text to recover any original file (RannohDecryptor[7]). Recently, the authors have fixed this flaw with CBC mode. In certain network, Tor traffic can be forbidden or might trigger warnings, instead TorrentLocker uses HTTPS like any legitimate connections made by the browser. New features were added like: harvesting email contacts, free decryption service for a single file, and partial encryption in order to speed up the attack.

The **CTB-Locker** (Curve-Tor-Bitcoin) appeared in June 2014 [27], it is the first to use ECC to compute public keys and shared secrets based on secret keys generated at runtime [28]. An active Internet connection is no longer required to begin the encryption process, moreover a complex cryptographic design is used. It is the first one which preceded AES file encryption by a compression step using ZLib. Communication with C&C is established by proxy websites like the Tor2web service which acts as a relay to the back-end infrastructure build on Tor hidden service. Infection distribution is realized thanks to an affiliate program to a network of partners motivated by gain.

A new variant appeared in February 2016 that is designed specifically against websites [29]. Infection begins with the site hack and replaces the legitimate `index.html` with one that performs file encryption and displays a ransom note. To be successful the site needs to use a php module.

TeslaCrypt ransomware was uncovered in February 2015 [30], distributed through websites that redirect victims to an exploit kit (drive-by download). Angler is one of them which takes advantage of Adobe Flash (CVE-2015-0311[8]). Contrary to other ransomware, TeslaCrypt keep encryption keys on the disk during the attack. It is one of the first ransomware which specifically targets files used by video games. Similarly to CTB-Locker, encryption process occurs irrespectively of any communication with the C&C, but their cryptosystems design differ slightly. TeslaCrypt embeds an ECC public-key in its binary, shared across plenty of samples which is used to compute a shared secret involved in the generation of an AES session key. The system has an ECC master private key. It was surprisingly broadcast in May 2016 by the malware authors themselves during the ransomware shutdown process.

It can be noticed that **Crypvault** [31] ransomware uncovered in April 2015 uses a legitimate cryptographic tool GnuPG to process file encryption. Additionally, a password dump utility is used as well as a software provided by Sys-

[7] `http://www.kaspersky.com/internet-security-center/threats/torrentlocker-malware`

[8] `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-0311`

internals in order to definitively delete files on disk.

In February 2016 **Locky** ransomware [32] hit the Internet and spread to more than 100 countries worldwide. Attackers can get statistics on the encrypted files but also the corresponding list and their path. The United States and France are the most infected countries, authors concentrate the best of the previous ransomware to achieve a highly skilled threat.

The **Petya** ransomware discovered in May 2016 [33] infects the hard drive partition table and prevents the operating system to be launched. Victims are redirected to a special boot screen asking for a ransom with persuasive font. Besides partition table overwriting which is not a new feature for malware, a special structure managed by the file system driver called Master File Table (MFT) is also encrypted preventing file recovery from a live cd. To the best of our knowledge it is the first significant ransomware to possess an entire offline cryptosystem design and which is placed at low-level. The disk encryption is performed with the stream cipher *salsa20* [34]. First versions presented some cryptographic flaws [35] that were corrected later.

3.2 Classification

In this paper, these ransomware are classified according to the cipher algorithms and Command and Control (C&C) communication channel. Traditionally, encryption algorithms used by ransomware are the standards: RSA [17] for metadata encryption and AES [36] for file encryption. Communication with C&C is more heterogeneous, each group of authors can choose a different option in function of the degree of reliability and confidentiality. Moreover, offline cryptosystems begin to appear and are a way for criminals to reduce exposure.

Table 1: Ransomware collection

Family	First seen	Most recent	Encryption algorithm	C&C
Gpcode	2004	2014	AES - ECB	~ HTTP
CryptoLocker	2013	2014	AES	~ HTTP
CryptoWall	2014	2016	AES - CBC	Tor
CTB-Locker	2014	2016	AES - ECB	Tor
TorrentLocker	2014	2016	AES - CTR|CBC *	Tor
TeslaCrypt	2015	2016	AES - ECB|CBC *	Tor
CrypVault	2015	2016	RSA - OAEP	
Locky	2016	-	AES - CTR|ECB *	~ HTTP
Petya	2016	-	Salsa20	No

* Samples variation.

4 Protection against ransomware using smart encoding

In this part, we propose a protection against ransomware using block ciphers. The ECB chaining mode can be exploited. On the other hand CBC exploitation involves more processing and requires poor cryptographic usages.

4.1 Ransomware using ECB mode

In cryptography, the ECB mode has a big drawback. Identical plain-text blocks are encrypted into identical cipher-text blocks; thus, it does not correctly hide data patterns. Fig. 3 illustrates this problem.

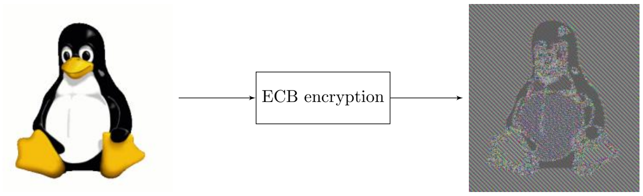

Fig. 3: ECB mode limitation (Tux pictures come from [37])

With such a mode, a replay attack [10] is possible. The main idea of our protection is to use the disadvantage of ECB mode to protect our data against ransomware. For this purpose, our protection consists in a smart data encoding.

The first step is to expend the data. Each byte of data is padded with 0 bytes to have block size n. In the case of the AES, $n = 16$. So a data file T of size m bytes, such as (3):

$$T = B_0, B_1, \cdots B_m \quad ; \tag{3}$$

is transformed in an expended data file such as (4):

$$\text{expended}T = B_0, \underbrace{0 \cdots 0}_{n-1}, B_1, \underbrace{0 \cdots 0}_{n-1}, \cdots, B_m, \underbrace{0 \cdots 0}_{n-1} \quad . \tag{4}$$

Additionally a dictionary is created:

$$\text{dic} = \underbrace{0 \cdots 0}_{n}, 1, \underbrace{0 \cdots 0}_{n-1}, \cdots, 255, \underbrace{0 \cdots 0}_{n-1} \quad . \tag{5}$$

If the ransomware uses the same key to encrypt all files, it also encrypts the dictionary file “dic”. The user can retrieve all files thanks to it by matching the encrypted blocks of a file with the encrypted blocks of the dictionary for which the unencrypted values are known.

If the ransomware uses a different key for each files, the dictionary can be created at the beginning of each of them. So a file T as (3) is expended in "dic + expended$T$":

$$\text{dic} + \text{expended}T = \underbrace{0 \cdots 0}_{n}, 1, \cdots, 255, \underbrace{0 \cdots 0}_{n-1}, B_0, \underbrace{0 \cdots 0}_{n-1}, \cdots, B_m, \underbrace{0 \cdots 0}_{n-1} \quad . \tag{6}$$

Another solution is to use the entropy of the original file. If T is a text, it can be supposed that the user knows the language used. In this case, no dictionary is necessary, text files are just expended (4) and the entropy is used to retrieve data with a classic basic cryptanalysis. Usually the file name is not encrypted, so the entropy could be stored in the file name to avoid the use of a dictionary.

4.2 Ransomware using CBC mode

If a ransomware uses a block cipher in CBC mode, the solution described in 4.1 fails, because of the xor at the start of block encryption. This paragraph presents a more complex protection for this mode working if the ransomware encrypts newly created file with the same key as the file you want to retrieve.

The data files are expended exactly as in (4). But the construction of the dictionary is different. The user does not create a dictionary file, but 256 files dic_b^0, one for each possible value of byte $b \in [\![0, 255]\!]$ such as (7):

$$\text{dic}_b^0 = b, \underbrace{0 \cdots 0}_{n-1} \quad . \tag{7}$$

The ransomware encrypts all data files and the 256 dictionaries dic_b^0. At this step, the user can retrieve the first and only the first byte B_0 of each encrypted files. Then she can create new dictionaries, one for each encrypted file and for each possible value of byte $b \in [\![0, 255]\!]$ such as (8):

$$\text{dic}_b^1 = (b, \underbrace{0 \cdots 0}_{n-1}) \oplus C_0 \quad . \tag{8}$$

Then the ransomware encrypts the new dictionaries. At this step the user can retrieve all second bytes B_1 of the files.

So the user can retrieve all bytes B_i by recurrence on the dictionaries byte (9).

$$\text{dic}_b^i = (b, \underbrace{0 \cdots 0}_{n-1}) \oplus C_i \quad . \tag{9}$$

4.3 Limitation

The main problem to these countermeasures are the sizes of the files which significantly increase. The data size is multiplied by n. In our description, the size of dictionary elements is fixed to one byte. In practice, it can be any other size. The bigger the size element is, the bigger the dictionary and the file sizes. In future works, it would be interesting to optimize these sizes.

It is important to precise that the countermeasure against CBC mode is limited to a ransomware which:

- uses always the same key K and the same vector IV,
- encrypts all newly created files.

5 Monitoring Microsoft's Cryptographic API

Beginning with cryptoviral extorsion as presented by Young *et al.* in [7], ransomware may use Microsoft's Cryptographic API. This section presents a generic countermeasure against a malicious usage of this API.

5.1 Microsoft's Cryptographic API (MS CAPI)

Windows operating systems, starting with Windows 95, include an easy-to-use API that supply cryptographic services to userland applications through MS CAPI. Cryptographic primitives are embedded in dynamic link libraries and divided in Cryptographic Service Providers (CSPs) each one offering a set of primitives classified by their type (Hash, Signature, Encrypt). Its objective is to provide an abstraction layer for programming purpose by non cryptographic specialists. Moreover due to its extensible architecture design, it is possible to add cryptographic modules once Microsoft signs them. For example, Hardware Security Module (HSM) vendors can implement white-box or home-made cryptographic algorithms and comply to Federal Information Processing Standard (FIPS) approved algorithms standards. It can be noticed that no user authentication is performed in order to access the primitives, authentication relies exclusively on the operating system. The framework does not provide persistent storage of keys or archival directly, you should use separate API provided by Microsoft. Moreover each time a process instantiates a Dynamic Link Library (DLL), code is shared but data remains unique for each instance and thus no key is exposed at load time. Cryptomodule is dedicated to one process so compartmentalization is ensured. Extended details can be found in [7,38] which give a list of the exported functions and an in-depth documentation.

MS CAPI is old (appeared with Windows 95), but is still widely spread notably in banking infrastructure such as ATMs. For legacy or backward compatibility in newer Windows operating systems, MS CAPI is deprecated but still present; it is strongly recommended to use the Cryptography API Next Generation (CNG) beginning with Windows Vista. Despite this recommendation OpenSSL continues to initialize its Deterministic Random Bit Generator (DRBG) with some entropy coming from *CryptGenRandom* from MS CAPI.[9]

[9] see *RAND_poll* in crypto/rand/rand_win.c

No ECC is supported with default providers, National Institute of Standards and Technology (NIST) Suite B Cryptography is fully available in CNG as well as additional chaining modes (XTS, GCM, CCM) and DESX block cipher.

5.2 System Protection

Presentation The main question is how to prevent malicious usages of MS CAPI without intrusive methods. We want to provide a fully transparent solution as simple as possible, ransomware must not be informed about any analysis.

Contrary to Bromium report [13] where an intrusive method based on the instrumentation library Detours [39] is chosen, we implemented a CSP and then plug it in the system. When incorporating a cryptographic module in the system, we are located in a legitimate position to observe malicious cryptographic behavior but also legitimate calls. We have to provide real services to user application and so we used the OpenSSL library for cryptographic primitives.

Implementation The Software Development Kit (SDK) and the Cryptographic Service Provider Developer's Toolkit (CSPDK)[10] supplied by Microsoft are used. We integrated the OpenSSL primitives in collaboration with MS CAPI architecture sin order to get a functional provider. Version 1.0.2e dated from December 2015 has been used and the provider was statically linked with the resulting library. We provide the following services :

- AES with CBC and ECB chaining modes,
- RSA #PKCS1 v1.5 and v2.0,
- RSA textbook,
- MD5, SHA1 and SHA256,
- random source.

The above algorithms represent a set of cryptographic primitives sufficiently exhaustive to fit the needs of ransomware based on observations and reports (Table 1).

Integration Cryptographic providers on both 32 and 64-bit architectures are compiled as DLLs. Providers are then placed in the *System32* default path, and system integration is achieved through register entries. [11] Administrator rights are needed to plug a CSP in the system, registration is performed with *regsvr32.exe*. As explained in 5.1, system can not trust, and thus use, any provider until a valid signature from Microsoft is obtained. For our academic proof of concept, we patched the binary responsible for the authentication mechanism to successfully load our CSP. As described in Fig. 4, user applications start by calling MS CAPI exported functions *Crypt(*)* through *advapi32.dll*. In fact this is nothing more than an indirect jump to *cryptsp.dll* which contains the CSP authentication mechanism and most of the framework functionalities.

[10] http://www.microsoft.com/en-us/download/details.aspx?id=18512

[11] HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider

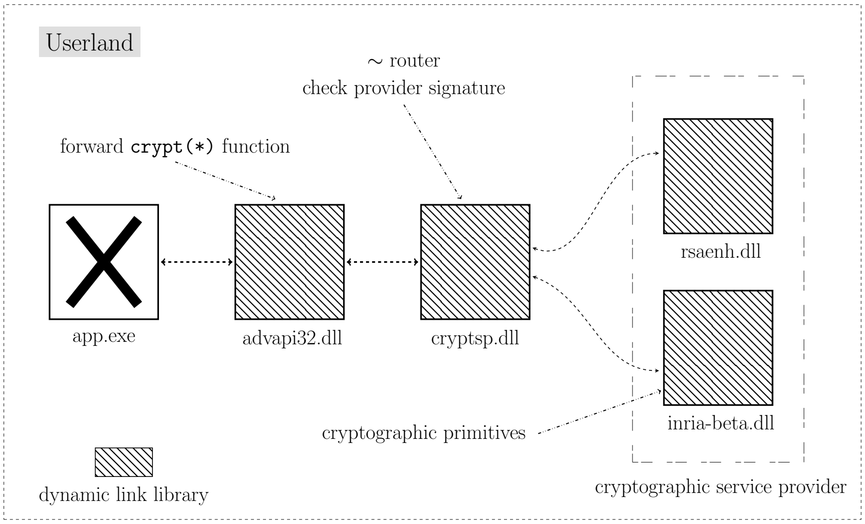

Fig. 4: Windows PE and Crypto API.

Still, a ransomeware may use another provider. Indeed, during the initialization phase, a call to *CryptAcquireContext* with the argument *pszProvider* lets the user choose between available providers.

```
BOOL WINAPI CryptAcquireContext(
  _Out_     HCRYPTPROV *phProv,
  _In_      LPCTSTR    pszContainer,
  _In_Opt   LPCTSTR    pszProvider,
  _In_      DWORD      dwProvType,
  _In_      DWORD      dwFlags
);
```

Listing 1.1: Provider selection

If the provider name is left *null*, a registry key indicates which is the default provider according to the requested type. So *cryptsp.dll* has been patched again to redirect all explicitly named provider to the default one. Final step is to set the name of the preferred cryptographic provider in registry as our own.

Detection The malicious behavior must be stopped and the damage limited. In addition file recovery is needed to render ransomware harmless. We have the control on key generation, encryption, signature, randomness and integrity. At load time, no key exists within a CSP and usually deletion, exportation or public-key encryption is performed once operations are done. We can take advantage of this opportunity to monitor and store the secrets, if malicious activities are detected.

Most interesting operations are logged in a special file prepended with a PE[12] header and extension to prevent ransomware encryption.

The difficulty is now to detect any abnormal behavior, no intelligence has been added to the module yet, but we investigate some possibilities. MS CAPI is marked as deprecated so very few applications on Windows down to Windows XP use its services. In any case, an intensive use is suspect, particularly encryption operations.

5.3 Limitation

As explained before, we had to place a patched *cryptsp.dll* in the ransomware directory. This is enough for our experimental purpose and characterization, but not for any real world deployment. Signing our provider will allow it to be loaded by the legitimate *cryptsp.dll*. Forcing its use is still a theoretical issue, in practice most ransomware use the default one: our own provider.

To complete the current solution we can consider ransomware detection based on a supervised machine learning classifier. The provider can be deployed on legitimate hosts then the normal behavior of MS CAPI can train a model. We do not want to see machine learning as a miracle oracle so, user opinion has to be requested when a malicious behavior is suspected. This solution can be incorporated in the provider and ransomware will never know about its existence. The solution presented in section 5.2 forces attackers to embed required cryptographic primitives in the binary. This approach avoids MS CAPI hijacking for manufacturers and aims to prevent malicious access.

6 Experimental Results

6.1 Environment

A bench of bare-metal hosts were used during the experiments. The Operating systems restoration is handled automatically in a few minutes. Moreover network traffic is captured through port mirroring and the hosts live in a completely open network. No malware analysis software were employed during ransomware execution, we aimed to be fully transparent regardless the threat complexity.

6.2 Results

The ransomware families considered are among the best known and the most diverse. Collection is composed of filecryptor (cf section 3). Unfortunately most of them do not trigger their malicious behavior. In most cases, C&C domains do not exist any more or are no more under attackers control. Due to this flaw, it seems important to note which one is active or inactive. CTB-Locker, TeslaCrypt and Petya do not require any C&C communication prior to begin encryption process. That is why good active rate is present with this families.

[12] PE: Portable Executable, Windows executable file format.

Table 2: Results of our countermeasures on our ransomware collection. The first column gives the family. The second column indicates the number of ransomware collected. The third column indicates how many collected ransomware were actives, so tested in our experiments. The last two columns indicate if the ransomware families use ECB mode and/or the Crypto API, in positive cases ✓is shown and the corresponding countermeasure is efficient. Unfortunately for families that do not have active ransomware, we are not able to know if the countermeasures work. Some families are divided in multiple rows, the separation represent significant variation between samples of the same family.

Family	# Samples	No. of Actives	Attack details	
			AES - ECB	Crypto API
Gpcode	4	4	✓	✓
	1	1	✓	×
CryptoLocker	7	0	-	-
	2	2	×	✓
CryptoWall	5	0	-	-
CTB-Locker	4	4	×	×
TorrentLocker	3	0	-	-
TeslaCrypt	2	2	×	×
	1	1	✓	×
	1	0	-	-
CrypVault	2	0	-	-
Locky	5	0	-	-
Petya	2	2	×	×

No. of Samples
No. of Actives
	39	16(41%)	6(37%)
			8(50%)
				6(37%)

Active samples have not yet been found for four families. CryptoWall, as noted by antivirus reports, uses MS CAPI to perform file encryption at least in these older versions [13]. Similarly Locky rely on MS CAPI [40].

Gpcode and CryptoLocker employ MS CAPI quite simply. First they acquire the context, the provider initializes some internal structures. Then they perform one or several AES key generations and quickly export them outside provider memory. Lastly prior to begin encryption, chaining mode is chosen. Encryption process is then performed through intensive calls to *CryptEncrypt()*.

6.3 Discussion

The main problem with ransomware is their high level of volatility. If C&C communication is required, reversing part of the network protocol is the only way to trigger the malicious behavior without intrusive methods. Ransomware samples have been found on different online repositories [41–43]. The samples quality is erratic especially for CryptoWall and Locky. Results are not relevant enough due to the small collection considered.

The low use of MS CAPI in the recent ransomware families could be partly explained with marketing purposes, no modern algorithms are available. It is more attractive to employ ECC cryptosystems with state of the art curve like *Curve25519* than the veteran RSA public key exchange algorithm. Petya uses Salsa20, in comparison MS CAPI contains the weak RC4 algorithm. A second explanation is that the generated keys are kept under the provider control through opaque structure. It can be one indication against a move to CNG.

In the future MS CAPI will be rare. The attackers have little control on MS CAPI and his successor CNG. Thus security checks may be embed as we presented. Strong cryptographic primitives are freely available contrary to the 90s. One solution is to embed the cryptographic code in the binary. This approach is more stealthy and the attackers gain in control.

7 Conclusion

Few papers have studied ransomware threats and discussed about countermeasures. The taxonomy changes all days, protect files is complex within this context. Microsoft Windows is the principal target, on March 2016 the first significant ransomware on the OS X platform appeared. It would not be surprising that ransomware and worm capabilities will be mixed to enhance the incomes.

In this paper nevertheless, two new protections have been presented. One of them is based on the principle of a replay attack against a weak chaining mode in combination with a cipher algorithm. The second, more practical, exploit the Microsoft's Cryptographic API that many ransomware use. These two protections are placed at a different level and can be used complementary for best security, leading to a 50% success rate in our experiments. Unfortunately they are not efficient against all ransomware. Moreover these countermeasures are used after the attack, once the files have been encrypted. In our future works we would like to find an efficient protection that prevent files encryption by the ransomware.

Acknowledgments

The authors would like to thank Ronan Lashermes, Alexandre Gonzalvez and the anonymous reviewers for their valuable help and comments.

References

1. Trend Micro. By the numbers: Ransomware rising. http://www.trendmicro.com.ph/vinfo/ph/security/news/cybercrime-and-digital-threats/by-the-numbers-ransomware-rising.
2. Roland Dela Paz. *Cryptowall, Teslacrypt and Locky: A Statistical Perspective.* https://blog.fortinet.com/2016/03/08/cryptowall-teslacrypt-and-locky-a-statistical-perspective.
3. Lawrence Abrams. *The Week In Ransomware - June 24 2016.* http://www.bleepingcomputer.com/news/security/the-week-in-ransomware-june-24-2016-locky-returns-cryptxxx-apocalypse-and-more/.
4. Kaspersky. *Kaspersky Security Bulletin 2015.* https://securelist.com/files/2015/12/Kaspersky-Security-Bulletin-2015_FINAL_EN.pdf.
5. Sergey Lozhkin. *Hospitals are under attack in 2016*, March 2016. https://securelist.com/blog/research/74249/hospitals-are-under-attack-in-2016.
6. Seung Lee. *Ransomware Wreaking Havoc in American and Canadian Hospitals*, March 2016. http://europe.newsweek.com/ransomware-wreaking-havoc-american-and-canadian-hospitals-439714?rm=eu.
7. Adam L. Young and Moti Yung. Cryptovirology: Extortion-based security threats and countermeasures. In *1996 IEEE Symposium on Security and Privacy, May 6-8, 1996, Oakland,CA, USA*, pages 129–140, 1996.
8. Alexandre Gazet. Comparative analysis of various ransomware virii. *Journal in computer virology*, 6(1):77–90, 2010.
9. Amin Kharraz, William Robertson, Davide Balzarotti, Leyla Bilge, and Engin Kirda. Cutting the gordian knot: a look under the hood of ransomware attacks. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 3–24. Springer, 2015.
10. Syverson, Paul. A taxonomy of replay attacks [cryptographic protocols]. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, pages 187–191. IEEE, 1994.
11. Sébastien Josse. White-box attack context cryptovirology. *Journal in Computer Virology*, 5(4):321–334, 2009.
12. James Wyke and Anand Ajjan. *Sophos: the Current State of Ransomware*, December 2015. https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/sophos-current-state-of-ransomware.pdf?la=en.
13. Vadim Kotov and Mantej Singh Rajpal. *Bromium: Understanding Crypto-Ransomware*, 2014. https://www.bromium.com/sites/default/files/bromium-report-ransomware.pdf.
14. Denis Sinegubko. *How CTB-Locker Ransomware Uses Bitcoin And Blockchain.* https://www.cryptocoinsnews.com/how-ctb-locker-ransomware-uses-bitcoin-and-blockchain/.
15. Invincea endpoint security blog : Pat Belcher. *Hash Factory: New Cerber Ransomware Morphs Every 15 Seconds.* https://www.invincea.com/2016/06/hash-factory-new-cerber-ransomware-morphs-every-15-seconds/.
16. National Institute of Standards and Technology. *DATA ENCRYPTION STANDARD (DES).* http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

17. Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
18. Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.
19. Symantec. *Trojan.Synolocker*, 2014. `https://www.symantec.com/security_response/writeup.jsp?docid=2014-080708-1950-99`.
20. Denis Nazarov and Olga Emelyanova. *Blackmailer: the story of Gpcode*, 2006. `https://securelist.com/analysis/publications/36089/blackmailer-the-story-of-gpcode`.
21. Keith Jarvis and SecureWorks Counter Threat UnitTM Threat Intelligence. *CryptoLocker Ransomware*, December 2013. `https://www.secureworks.com/research/cryptolocker-ransomware`.
22. Federal Bureau Of Investigation (FBI). *GameOver Zeus Botnet Disrupted.* `https://www.fbi.gov/news/stories/2014/june/gameover-zeus-botnet-disrupted`.
23. Andrea Allievi and Earl Carter. *Ransomware on Steroids: Cryptowall 2.0.* Cisco, 2015. `http://blogs.cisco.com/security/talos/cryptowall-2`.
24. Yonathan Klijnsma. *The history of Cryptowall: a large scale cryptographic ransomware threat.* `https://www.cryptowalltracker.org/`.
25. Marc-Etienne M.Léveillé. *TorrentLocker: Ransomware in a country near you*, 2014. `http://www.welivesecurity.com/wp-content/uploads/2014/12/torrent_locker.pdf`.
26. Helger Lipmaa, Phillip Rogaway, and David Wagner. Ctr-mode encryption. In *First NIST Workshop on Modes of Operation*, 2000.
27. Zairon. *CTB-Locker encryption/decryption scheme in details*, February 2015. `https://zairon.wordpress.com/2015/02/17/ctb-locker-encryptiondecryption-scheme-in-details`.
28. Daniel J. Bernstein. *A state-of-the-art Diffie-Hellman function.* `http://cr.yp.to/ecdh.html`.
29. Lawrence Abrams. *CTB-Locker for Websites: Reinventing an old Ransomware.* `http://www.bleepingcomputer.com/news/security/ctb-locker-for-websites-reinventing-an-old-ransomware/`.
30. Talos Group. *Threat Spotlight: TeslaCrypt – Decrypt It Yourself*, April 2015. `http://blogs.cisco.com/security/talos/teslacrypt`.
31. Michael Marcos. *CRYPVAULT: New Crypto-ransomware Encrypts and "Quarantines" Files.* `http://blog.trendmicro.com/trendlabs-security-intelligence/crypvault-new-crypto-ransomware-encrypts-and-quarantines-files/`.
32. Fedor Sinitsyn. *Locky: the encryptor taking the world by storm*, 2016. `https://securelist.com/blog/research/74398/locky-the-encryptor-taking-the-world-by-storm`.
33. Fedor Sinitsyn. *Petya: the two-in-one trojan*, May 2016. `https://securelist.com/blog/research/74609/petya-the-two-in-one-trojan`.
34. Daniel J Bernstein. The salsa20 family of stream ciphers. In *New stream cipher designs*, pages 84–97. Springer, 2008.
35. Leo-stone. *Hack-petya mission accomplished.* `https://github.com/leo-stone/hack-petya`.
36. National Institute of Standards and Technology (NIST). *Specification for the Advanced Encryption Standard, FIPS PUB 197*, November 2001.
37. Wikipedia. *Block cipher mode of operation.* `https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation`.

38. Microsoft. *Microsoft Enhanced Cryptographic Provider, FIPS 140-1 Documentation: Security Policy*, 2005. `http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp238.pdf`.
39. Galen Hunt and Doug Brubacher. Detours: Binary interception of win 32 functions. In *3rd USENIX Windows NT Symposium*, 1999.
40. Hasherezade. Look into locky ransomware. `https://blog.malwarebytes.com/threat-analysis/2016/03/look-into-locky/`.
41. Malware online repository. `https://malwr.com`.
42. Malware online repository. `http://malwaredb.malekal.com`.
43. Malware online repository. `https://virusshare.com`.