A Review of Encryption Schemes Used in Modern Ransomware[1]

CECC 2020

Peter Švec Roderik Ploszek

Institute of Computer Science and Mathematics
Slovak University of Technology

24–26 June 2020

[1] This research was sponsored by Slovak Republic under grant VEGA 1/0159/17.

Contents

Ransomware

- special type of malware that encrypts personal user data
- after encryption is finished, the malware provides instructions how to pay a **ransom**
- ransomware was reported as a top threat in 2019 IOCTA

Analysis process

Objective

Analyse modern ransomware and compare the findings with the previous generation. Findings can be used to understand current trends in ransomware and improve detection.

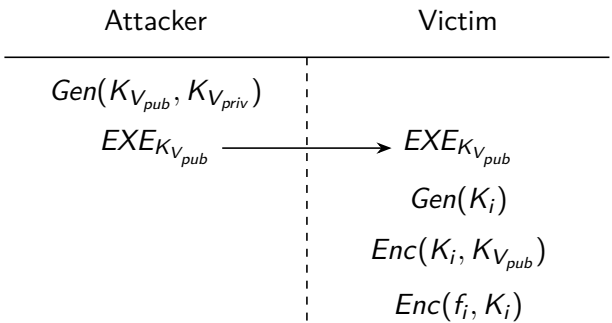
Steps

1. download a sample from malware archives, check known hash value
2. dynamic analysis – run the sample and analyse impact on the system
3. static analysis – use disassembler/decompiler to analyse inner workings of ransomware

Encryption schemes

1. Shipped public key to encrypt random symmetric keys, generated for each file to be encrypted.
2. Generated public and private key pair on the victim side. Private key is sent back to the attacker. Public key is used to encrypt random symmetric keys, generated for each file to be encrypted.
3. Shipped public key to encrypt global symmetric key, used for victim data encryption.
4. Three-tier trust model, where shipped public key A is used for encrypting private key B. Public key B is used for symmetric keys encryption, generated for each file.

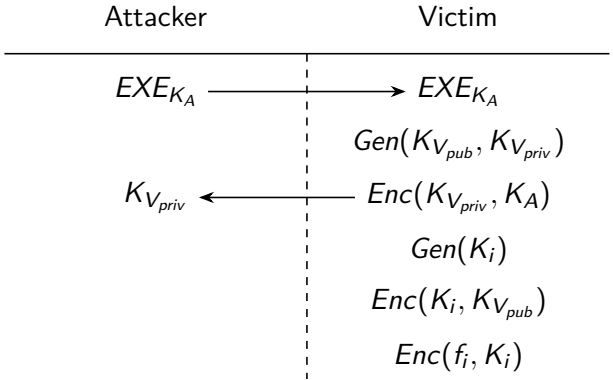
Encryption scheme 1



This scheme is used by Ryuk, Dharma, LockBit...

A	attacker
V	victim
K	key
$Enc(msg, K)$	encrypt msg with K
EXE_K	executable with embedded key K
$Gen(K)$	generate key K
f_i	file from system, $i \in N$, where N is total number of files

Encryption scheme 2



A	attacker
V	victim
K	key
$Enc(msg, K)$	encrypt msg with K
EXE_K	executable with embedded key K
$Gen(K)$	generate key K
f_i	file from system, $i \in N$, where N is total number of files

This scheme is used by GandCrab...

Symmetric ciphers in older ransomware

Ransomware name	Algorithm	Mode	IV
OpenToYou	RC4	—	—
Amnesia	AES-128	CBC	—
Globe V3	AES-256	ECB	—
Xmas	Custom cipher	—	—
LeChiffre	BlowFish	—	—
Petya	Salsa20	—	—

Craciun, Mogage, Simion: *Trends in Design of Ransomware Viruses* (2019)

Symmetric ciphers in today's ransomware

Ransomware name	Algorithm	Mode	IV
Ryuk	AES-256	CBC	Zero
Dharma	AES-128	CBC	Random
LockBit	AES-128	CBC	Random
SamSam	AES-128	CBC	Random
GandCrab	Salsa20, RC4	—	—
Lilocked	ARIA-128	—	—
Clop	Custom cipher	—	—

Asymmetric ciphers in older ransomware

Ransomware name	Algorithm
Petya	ECC
GpCode	RSA-660
CTB-Locker	ECC
Shade	RSA-3072
CryptoMix	RSA-2048
Cerber	RSA-2048

Subedi, Budhathoki, Dasgupta: *Forensic Analysis of Ransomware Families using Static and Dynamic Analysis* (2018)

Asymmetric ciphers in today's ransomware

Ransomware name	Algorithm
Ryuk	RSA-2048
LockBit	RSA-2048
SamSam	RSA-2048
GandCrab	RSA-2048
Dharma	RSA-1024
Clop	RSA-1024

Key generation in older ransomware

Ransomware name	Key
Petya	secp192k1
OpenToYou	hardcoded
Amnesia	C `rand()` function
Xorist	C `rand()` function
Xmas	C `rand()` function
LeChiffre	hardcoded

Craciun, Mogage, Simion: *Trends in Design of Ransomware Viruses* (2019)

Key generation in today's ransomware

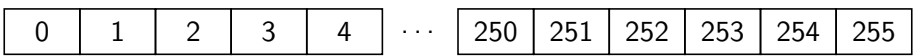
Ransomware name	Symmetric key	Asymmetric key
Ryuk	`CryptGenKey`	hardcoded
Dharma	`get_random_NZ`	hardcoded
LockBit	`CryptGenRandom`	hardcoded
SamSam	`RNGCryptoServiceProvider`	argument
GandCrab	`CryptGenRandom`	`CryptGenKey`
Clop	`CryptGenKey`	hardcoded

Cryptography implementation in today's ransomware

Ransomware name	Implementation
Ryuk	Windows CryptoAPI
Dharma	axTLS embedded SSL
LockBit	Windows CryptoAPI + AES-NI instruction set + Optimized Rijndael
SamSam	.NET System.Security.Cryptography
GandCrab	Windows CryptoAPI
Clop	Widnows CryptoAPI

Custom ciphers (Clop)

Cipher state is a permutation of 256 bytes (0–256):



Initial state is determined by a random 117 byte key.
For every byte of the input file, two cipher state bytes are exchanged and used to determine cipher state byte that is xored with the input byte.

Common Mistakes: Then & Now

Then

Previously, ransomware authors used `rand()` function to generate keys, CBC mode using zero initialization vector, ECB mode or custom ciphers.

Now

Nowadays, ransomware creators are more careful and depend mostly on correct cryptography implementations.

Quantum ransomware: The future

As quantum computers are improving, we can expect the need to move away from standard asymmetric algorithms such as RSA.
Ransomware authors will certainly follow this trend.

Conclusion

Future plans

- Acquire and analyze more samples.
- Use gained information in detection systems (e.g. based on ontology)

Discussion

peter.svec1@stuba.sk
roderik.ploszek@stuba.sk