

# The Fairest Ransomware

Hikaru YOSHIMURA

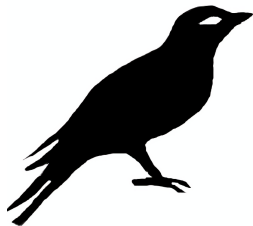
DWANGO Co., Ltd.  
[yyu@mental.poker](mailto:yyu@mental.poker)

June 31, 2017  
(Commit ID: a3196c6)

# Table of Contents

- 1 Who am I?
- 2 Introduction
- 3 Definition and Notation

# Who am I?

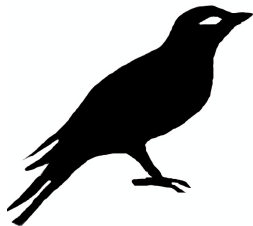


Twitter @\_yyu\_

Qiita yyu

GitHub y-yu

# Who am I?



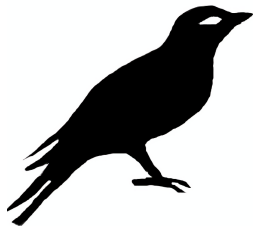
- University of Tsukuba (Undergraduate)

Twitter @\_yyu\_

Qiita yyu

GitHub y-yu

# Who am I?



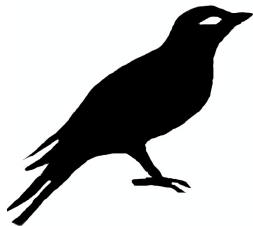
- University of Tsukuba (Undergraduate)
- Authentication Platform Section, DCS Dept

Twitter @\_yyu\_

Qiita yyu

GitHub y-yu

# Who am I?



- University of Tsukuba (Undergraduate)
- Authentication Platform Section, DCS Dept
- I'm interesting in Cryptography

Twitter @\_yyu\_

Qiita yyu

GitHub y-yu

# Introduction

# Introduction

- Recently, *Ransomwares* are being famous.



# Introduction

- Recently, *Ransomwares* are being famous.

## Ransomware

Ransomware is one of the malwares, which encrypts the data on a victim's computer and then make them pay ransom in exchange for decrypting.

# Introduction

- Recently, *Ransomwares* are being famous.

## Ransomware

Ransomware is one of the malwares, which encrypts the data on a victim's computer and then make them pay ransom in exchange for decrypting.

## Famous Ransomwares

- WannaCry
- Petya
- GoldenEye

# Introduction

- Recently, *Ransomwares* are being famous.

## Ransomware

Ransomware is one of the malwares, which encrypts the data on a victim's computer and then make them pay ransom in exchange for decrypting.

## Famous Ransomwares

- WannaCry
- Petya
- GoldenEye

Will ransomwares decrypt the data if the victim pays Bitcoin?

# The Fairest Ransomware

# The Fairest Ransomware

“If the victim pays some Bitcoins, their data will be decrypted under the probability on which they agreed.”

# The Fairest Ransomware

“ If the victim pays some Bitcoins, their data will be decrypted under the probability on which they agreed. ”

It's possible using cryptographic technique

# Encryption

# Encryption

**Symmetric Key Encryption (SKE)** is a cryptographic scheme that uses the *same* key to encrypt and decrypt data like AES. An encryption function is denoted  $\text{Enc}$ , a decryption function is denoted  $\text{Dec}$ . The following equation holds for the symmetric key  $k$ .

$$x = \text{Dec}_k(\text{Enc}_k(x))$$



# Encryption

**Symmetric Key Encryption (SKE)** is a cryptographic scheme that uses the *same* key to encrypt and decrypt data like AES. An encryption function is denoted  $\text{Enc}$ , a decryption function is denoted  $\text{Dec}$ . The following equation holds for the symmetric key  $k$ .

$$x = \text{Dec}_k(\text{Enc}_k(x))$$

**RSA Encryption** is a cryptographic scheme that uses the *different* keys between encrypting and decrypting data. The key using encryption is called *public key* and The key using description is called *secret key*. The following holds for a public key  $(e, N)$  and the secret key  $d$ .

$$x = (x^e)^d = (x^d)^e \pmod{N}$$

# Hash Function

# Hash Function

## Hash Function

A hash function  $H$  is a *one way function*, which means that:

- It's easy to calculate the output  $H(x)$  from input  $x$
- But it's hard to calculate the input  $x$  from output  $H(x)$

# Hash Function

## Hash Function

A hash function  $H$  is a *one way function*, which means that:

- It's easy to calculate the output  $H(x)$  from input  $x$
- But it's hard to calculate the input  $x$  from output  $H(x)$

A hash function  $H$  has the following properties:

# Hash Function

## Hash Function

A hash function  $H$  is a *one way function*, which means that:

- It's easy to calculate the output  $H(x)$  from input  $x$
- But it's hard to calculate the input  $x$  from output  $H(x)$

A hash function  $H$  has the following properties:

**Preimage Resistance** A hash value  $h$ , it's difficult to find any message  $m$  such that  $h = H(m)$ .

**Second Preimage Resistance** An input  $m_1$ , it's difficult to find different input  $m_2$  such that  $H(m_1) = H(m_2)$ .

**Collision resistance** It's difficult to find two different messages  $m_1$  and  $m_2$  such that  $H(m_1) = H(m_2)$ .

# Zero-Knowledge Proof

# Zero-Knowledge Proof

There are two people, Alice and Bob.

# Zero-Knowledge Proof

There are two people, Alice and Bob.

**Prover Alice** has the secret key  $d$  for RSA cipher text  $c$  encrypted by public key  $(e, N)$ . And she has cipher text  $s := \text{Enc}_k(c^d \bmod N)$  and its symmetric key  $k$ .



# Zero-Knowledge Proof

There are two people, Alice and Bob.

**Prover Alice** has the secret key  $d$  for RSA cipher text  $c$  encrypted by public key  $(e, N)$ . And she has cipher text  $s := \text{Enc}_k(c^d \bmod N)$  and its symmetric key  $k$ .

**Verifier Bob** has the cipher text  $c$ , its public key  $(e, N)$  and the cipher text  $s$ .

# Zero-Knowledge Proof

There are two people, Alice and Bob.

**Prover Alice** has the secret key  $d$  for RSA cipher text  $c$  encrypted by public key  $(e, N)$ . And she has cipher text  $s := \text{Enc}_k(c^d \bmod N)$  and its symmetric key  $k$ .

**Verifier Bob** has the cipher text  $c$ , its public key  $(e, N)$  and the cipher text  $s$ .

Bob want to verify as follows:

“ A preimage of the hash value  $H(k)$  is a symmetric key  $k$  that can decrypt SKE cipher text  $s$ . ”

without knowing the secret key  $d$  or symmetric key  $k$ .

# Zero-Knowledge Proof

We use *cut-and-choose protocol* where RSA cipher text  $c$  encrypted by public key  $(e, N)$  and its secret key  $d$ .

# Zero-Knowledge Proof

We use *cut-and-choose protocol* where RSA cipher text  $c$  encrypted by public key  $(e, N)$  and its secret key  $d$ .

**Alice**

**Bob**

---

 $F := \{\sigma_1, \dots, \sigma_n\}$  $R := \{d_1, \dots, d_m\}$  $\beta$  is a random permutationfor  $\{\sigma_1, \dots, \sigma_n, d_1, \dots, d_m\}$  $\beta$ 

---

 $f(\beta)$ 

finalize

Thank you for your attention!

Any question?