Ransomware Using Smart Contracts

Team : Green’s Club

Submitted By

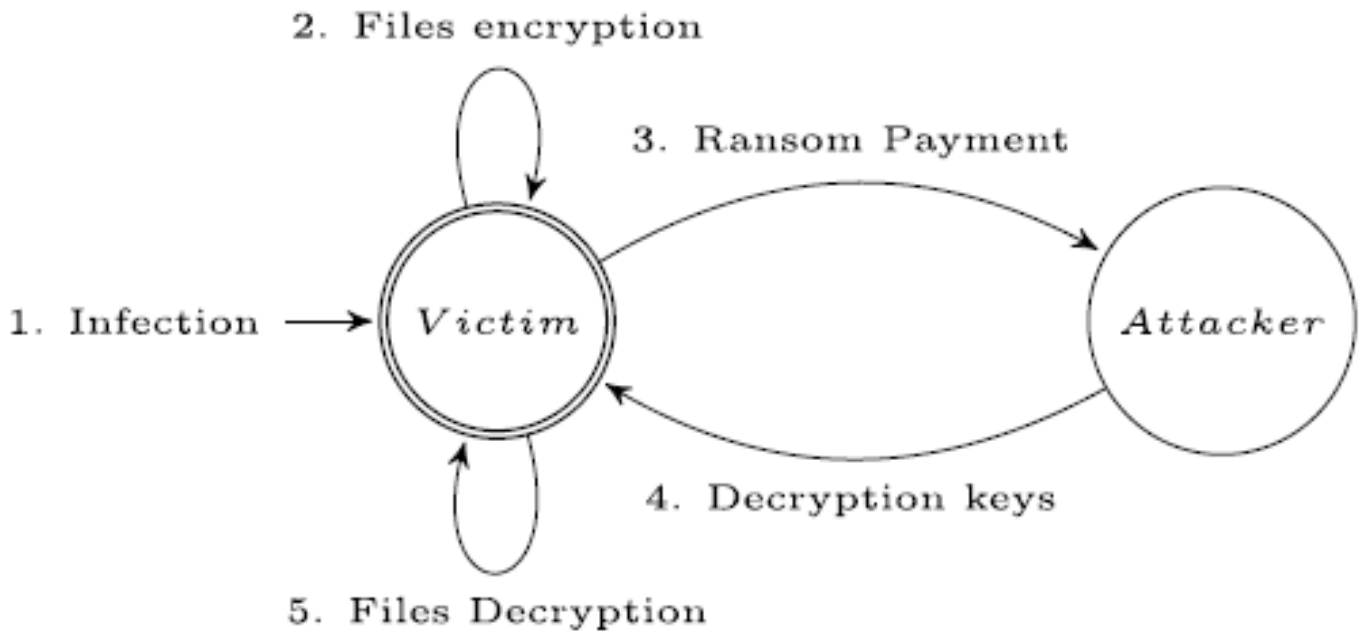
Mohit Vaid

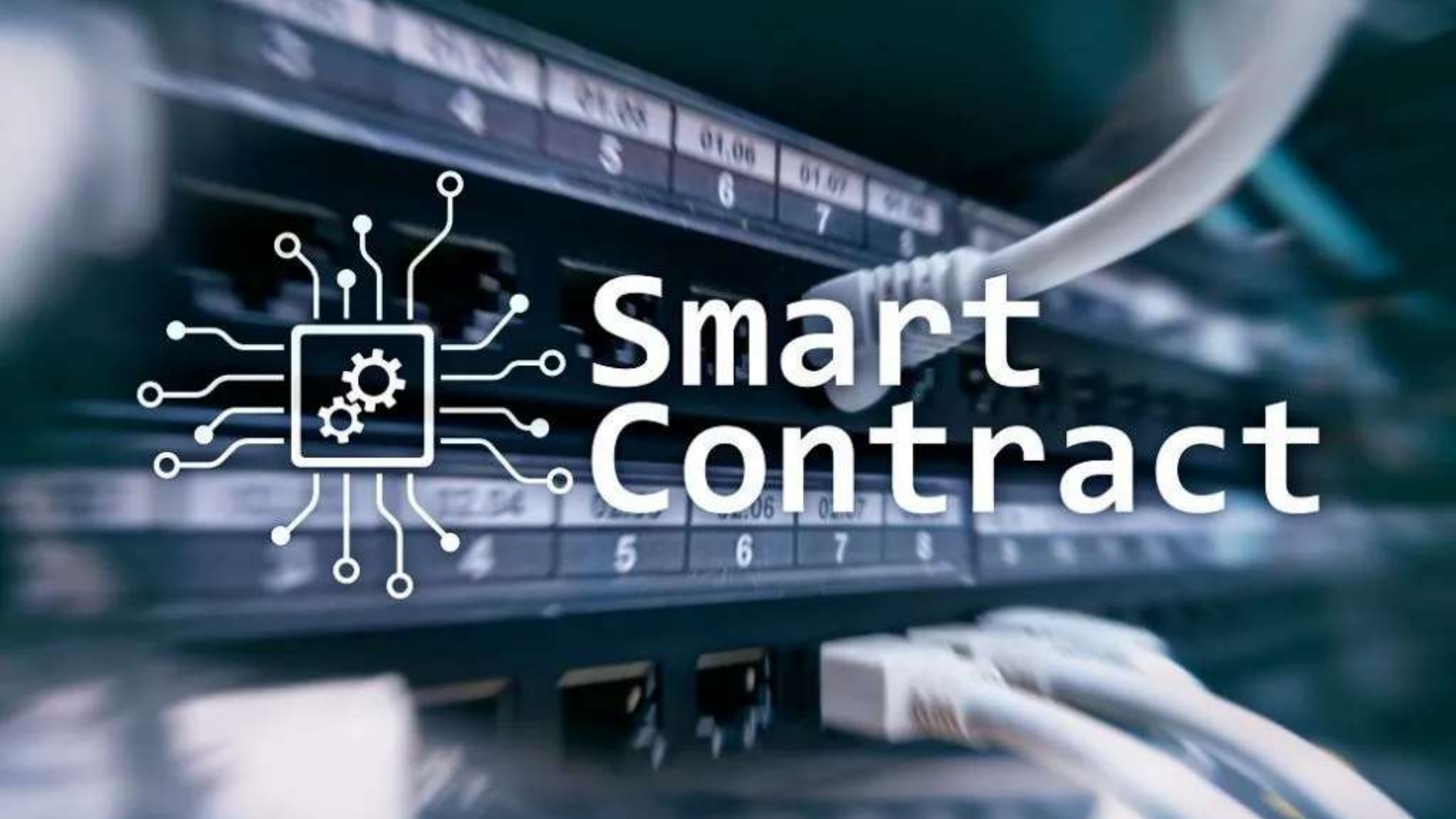
Akash Gupta

Saty Mohammad

Introduction

- Common Ransomware Scheme




Smart
Contract

Smart Contracts

Guarantees to the Victim

Pay Per Decrypt

More Resilient

Scheme

Preliminary

Infection Vectors

Generate

Master SK

Victim

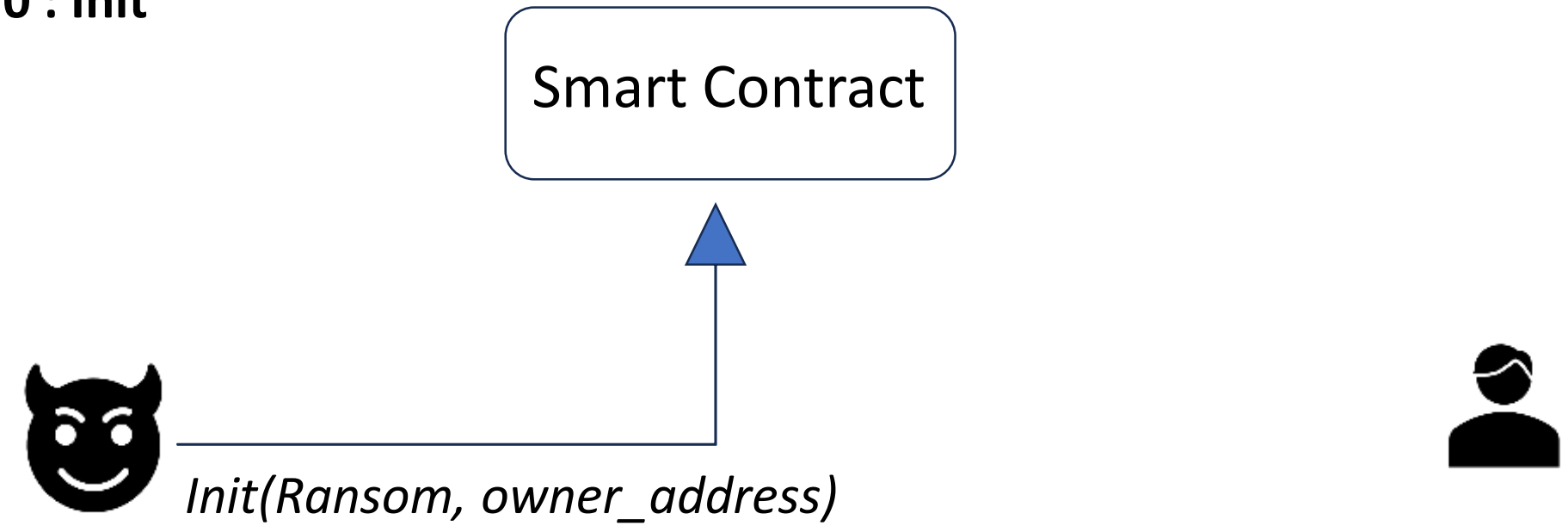
+ ID

Victim

E (Customized Executable)

Scheme

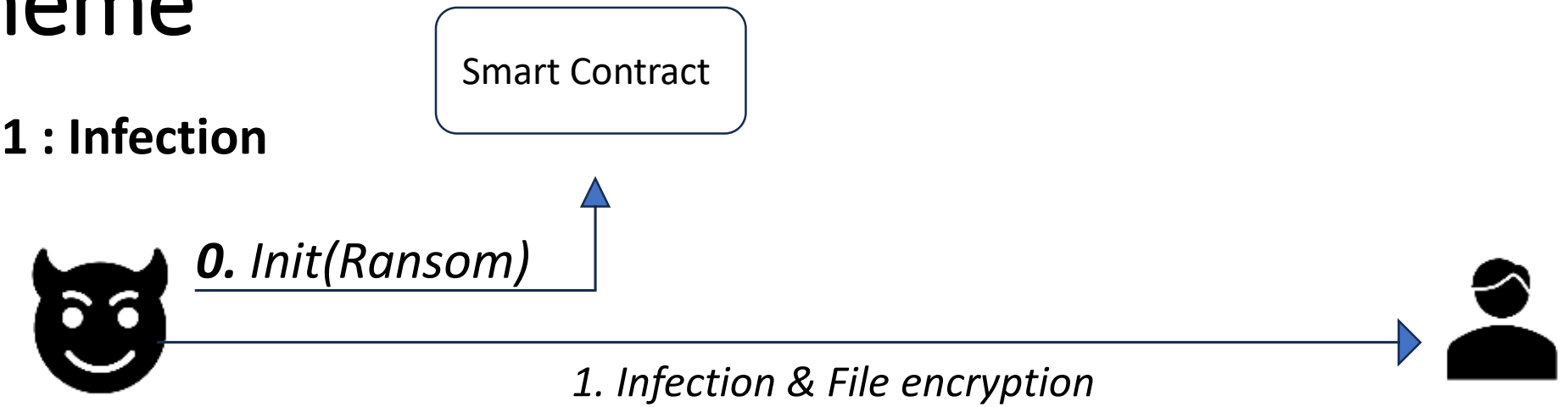
Step 0 : Init



Attacker deploys smart contract C to a public blockchain (Ethereum), and initializes it by calling init() function with a ransom amount

Scheme

Step 1 : Infection



Infection and File encryption

- $dk_f = Hash\ (MasterSK, id_v||id_f)$
- Commitment $c\ = Hash\ (MasterSK)$

c, Encrypted Files

Stored in Victims System

Commitment will be used as a proof

id_v: $Id\ of\ the\ victim$
id_f: Id of the encrypted files
$C_f = Commitment$

Scheme

Step 2 & Step 3

Smart Contract

2. Challenge ( )

Challenge

Victims Asks for proof
by calling

$$getFreeDecryption\ (\ id_v, \{id_f\})$$

Verifies

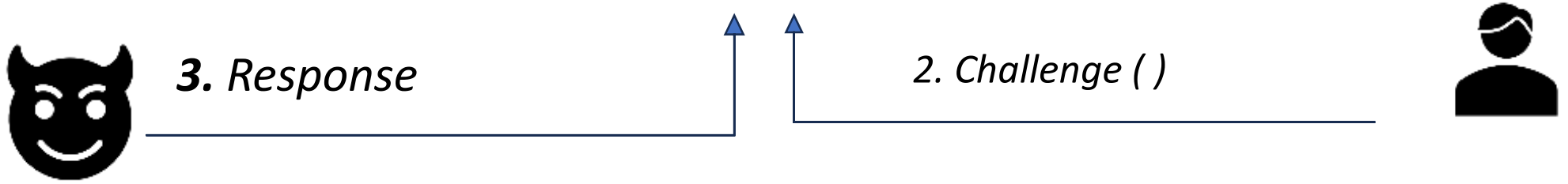
$$Number\ of\ elements\ in\ \{id_f\}\ < k$$

- If True then Smart Contract mark these files decrypted

$\{id_f\}$: Ids of file the victim wants to decrypt for proof

Scheme

Step 2 & Step 3



Challenge

Victims Asks for proof
by calling

$$getFreeDecryption\ (\ id_v, \{id_f\}, c)$$

Response

$$Attacker\ calls\ getID()$$

- Smart Contract will send the Identifier (Id) and the id of file
- Attacker calls $revealKey\ (id_v, id_f)$
- $dk_f = Hash(\ MasterSK\ , id_v\ ||\ id_f)$

Returns (id_v, dk_f)

`

Scheme

Step 5 – Pay Ransom

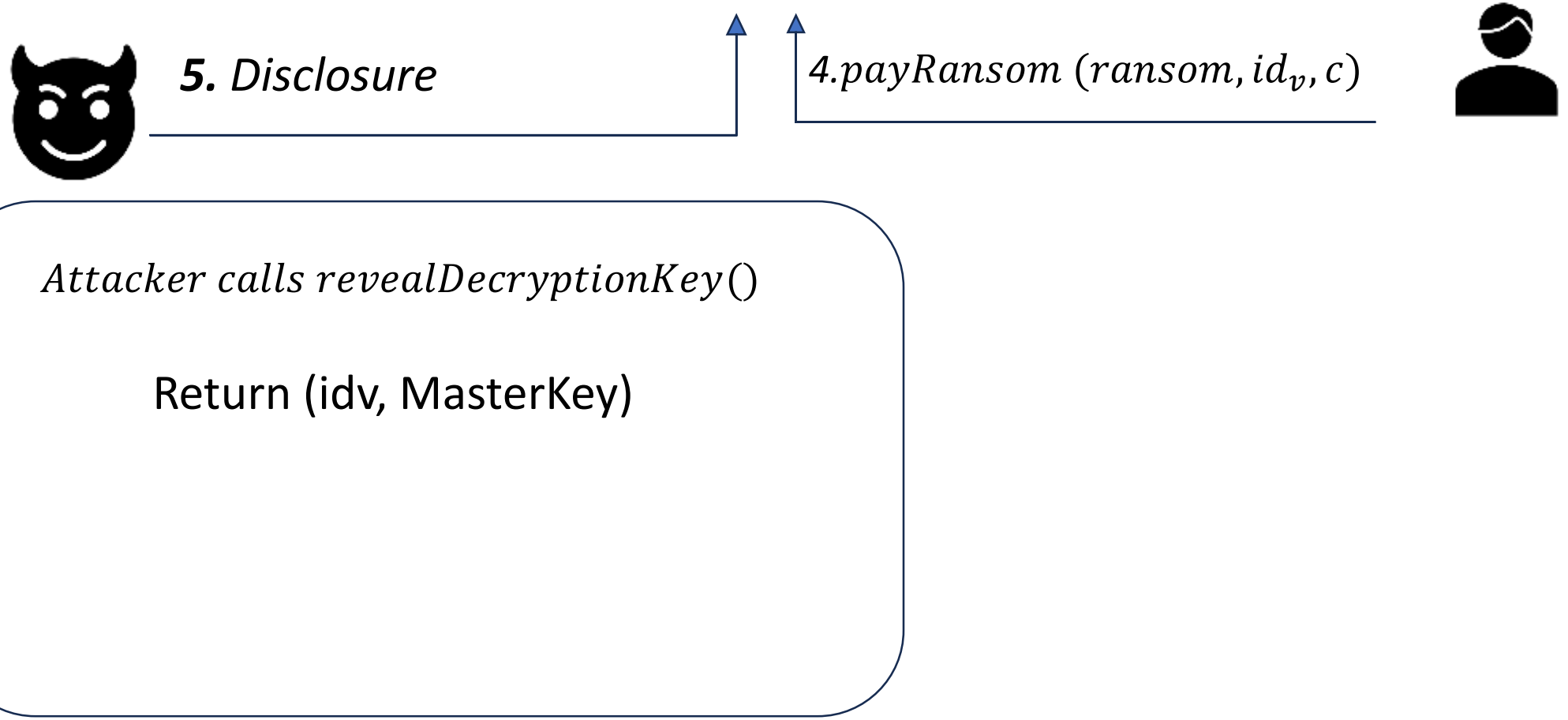
Verifies

- If ransom $\geq$ *Ransom set by attacker*

$$VictimID.Ransom == True$$

Scheme

Step 5 – Pay Ransom



Scheme

Verification

5. *Disclosure*

Smart Contract

Verifies

$$SMART\ CONTRACT$$

$$h_1 \rightarrow Hash\ (MasterKey)$$

$$If\ h_1 ==\text{c}$$

$$Returns\ MasterSK, to\ Victim$$

$$Else$$

Return the Ransom to the Victim

Conclusion

- Demonstrated how smart contract works enduring Guarentes to victim in Ransomware
- There are very few Countermeasures for this beyond causing an intentional Hard Fork in the block chain to eliminate smart contract
  - Very Difficult to perform, also puts whole system at risk
- More Challenging work
  - Zero Knowledge Proof : For proving legitimacy of the attacker
  - State Channels: This could help reduce number of exchanges messages on main blockchain

- attacker uses a symmetric encryption algorithm enc
- with a key k to encrypt x, such that enck(x) = c. He also uses a
- hash function h to compute h(k) = y. She then sends these values
- c and y to Victim, together with a zero-knowledge proof that c is the
- ciphertext of x under the key k and that h(k) = y.