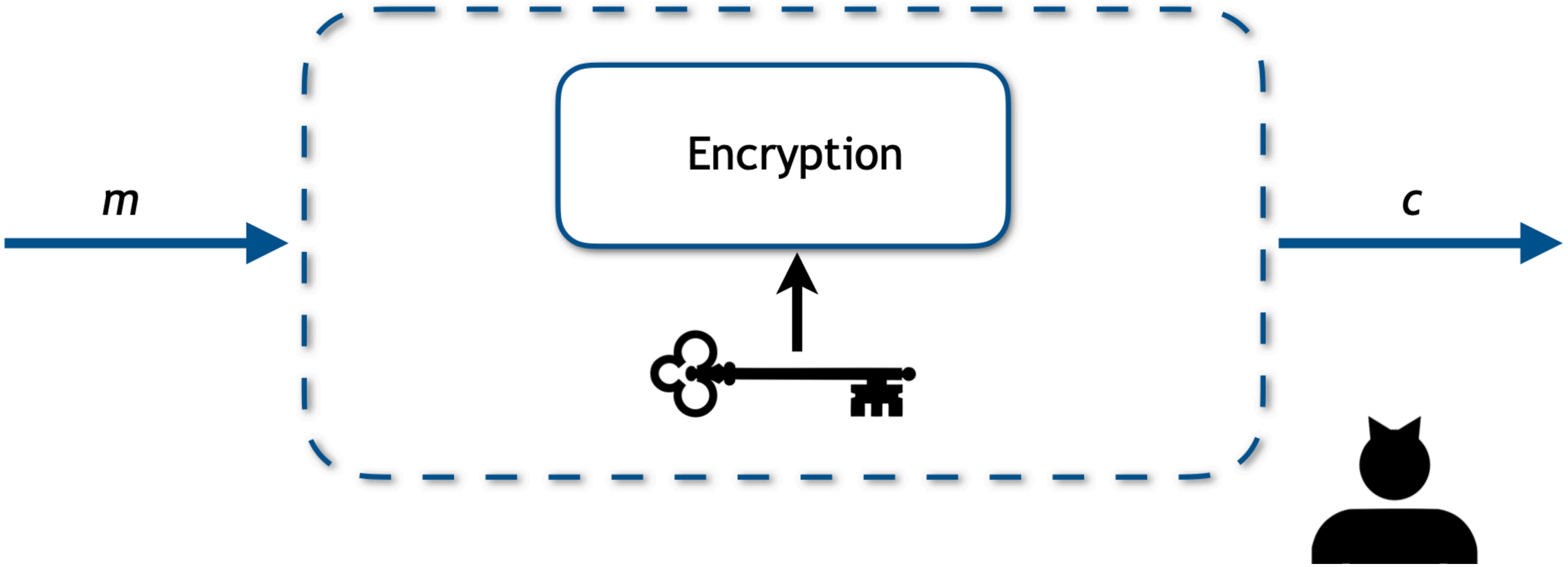
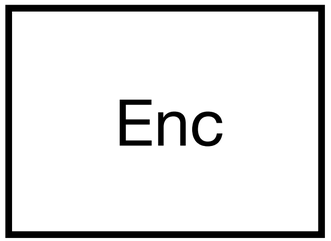
On Provable White-Box Security in the Strong Incompressibility Model

Estuardo Alpirez Bock, Chris Brzuska and Russell W. F. Lai

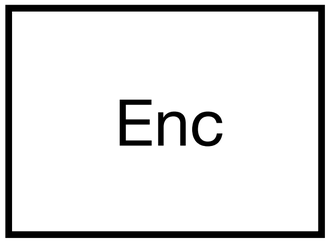
White-box attack model



Incompressibility

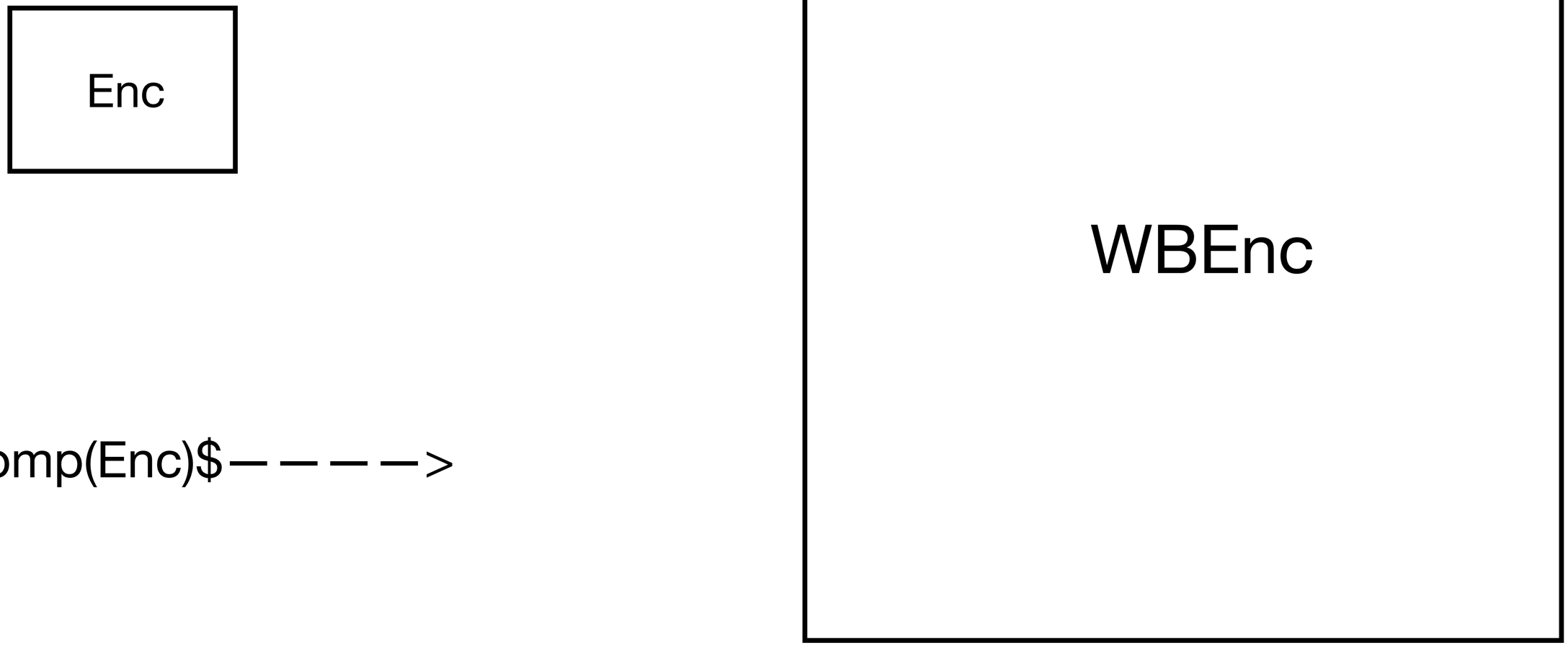


Incompressibility



Comp(Enc)$— — — —>

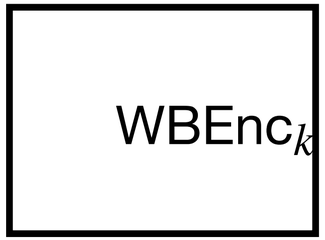
Incompressibility



White-box incompressibility

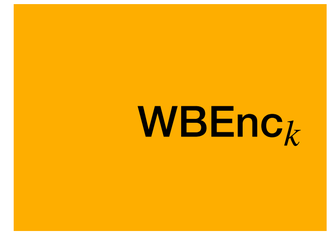
Code-lifting on white-box programs

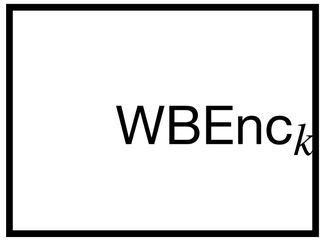
- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks



Code-lifting on white-box programs

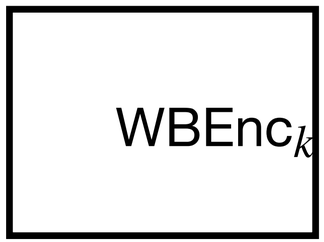
- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks

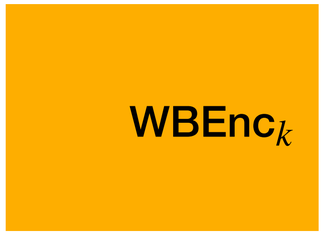




Code-lifting on white-box programs

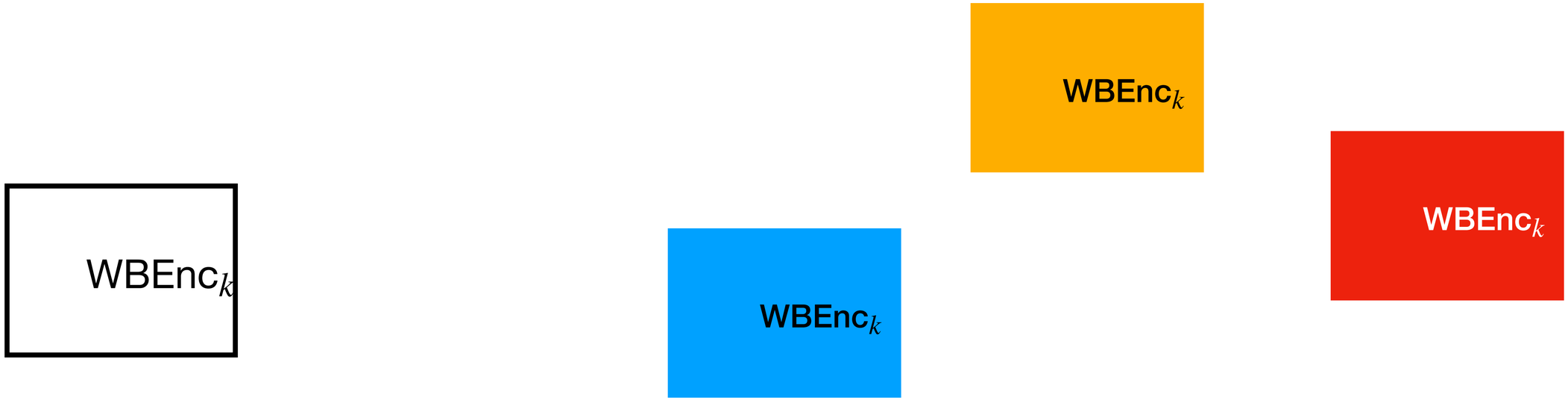
- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks





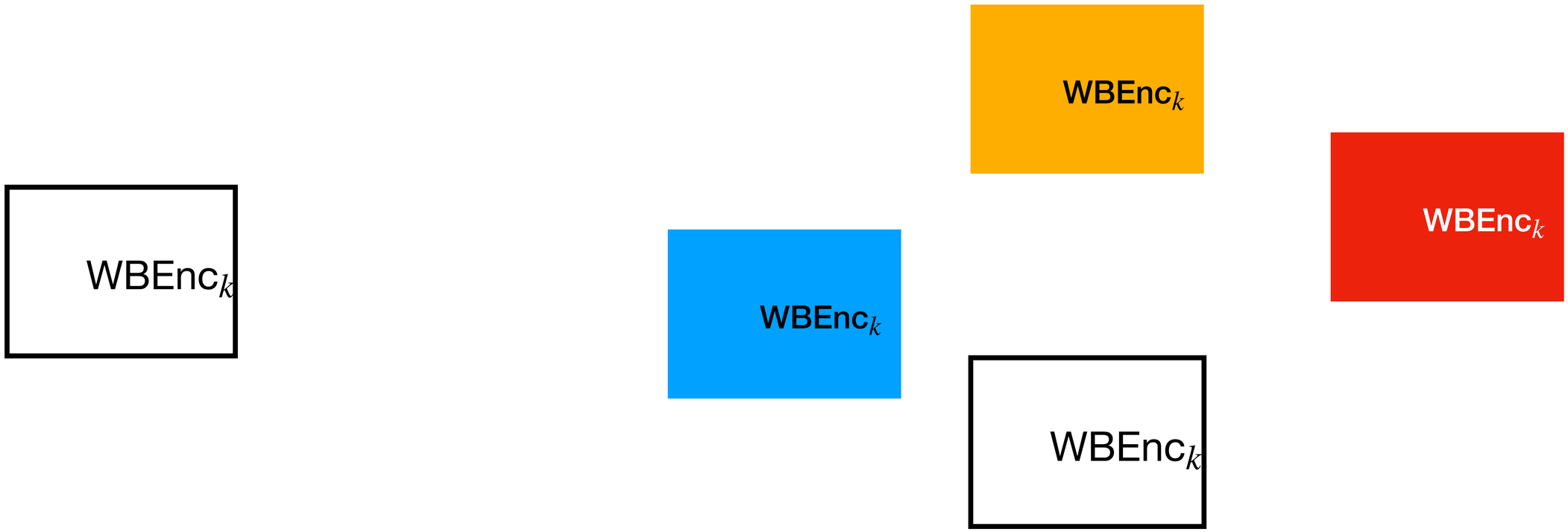
Code-lifting on white-box programs

- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks



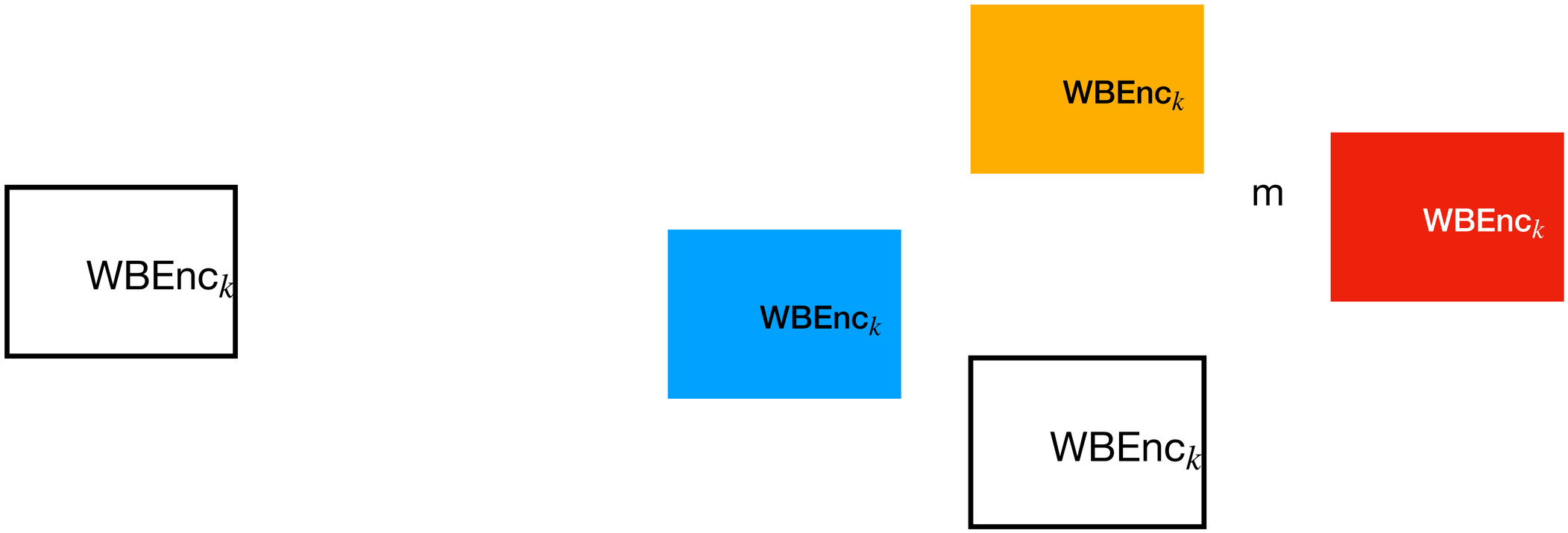
Code-lifting on white-box programs

- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks



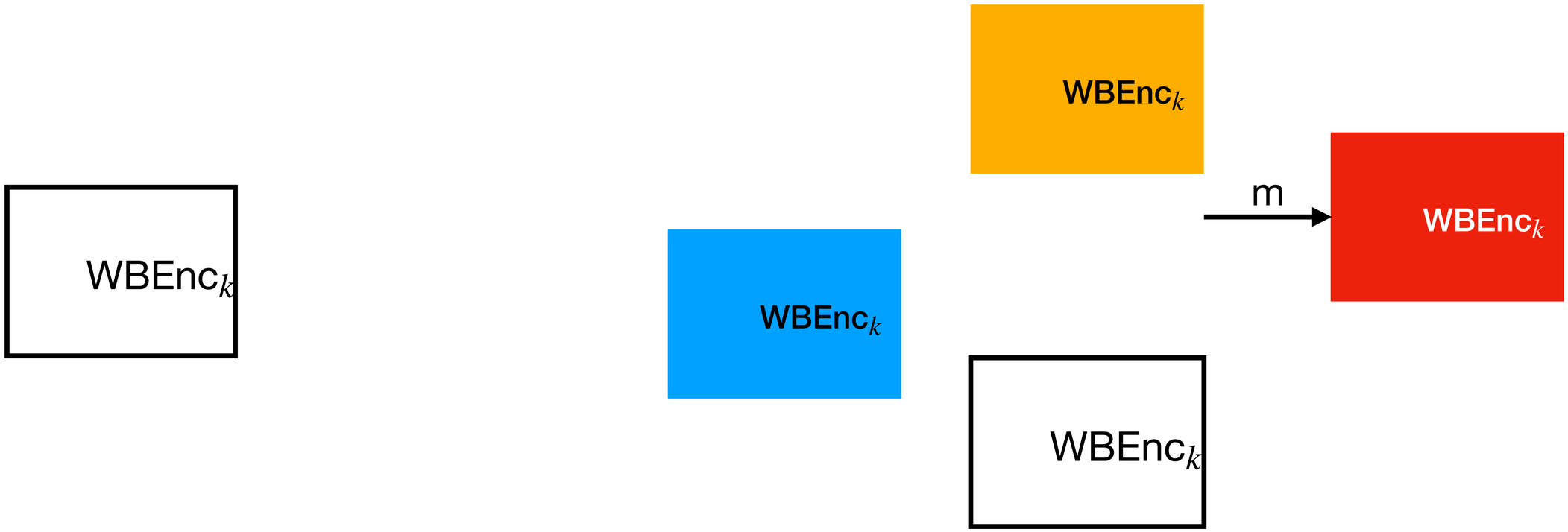
Code-lifting on white-box programs

- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks



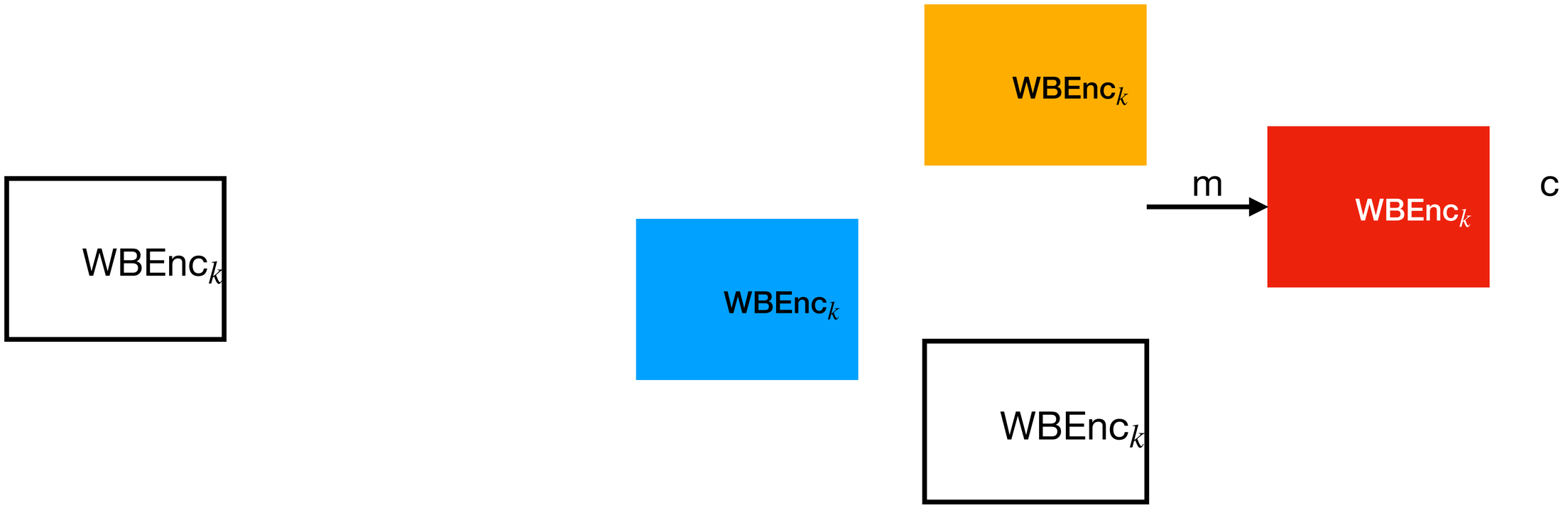
Code-lifting on white-box programs

- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks



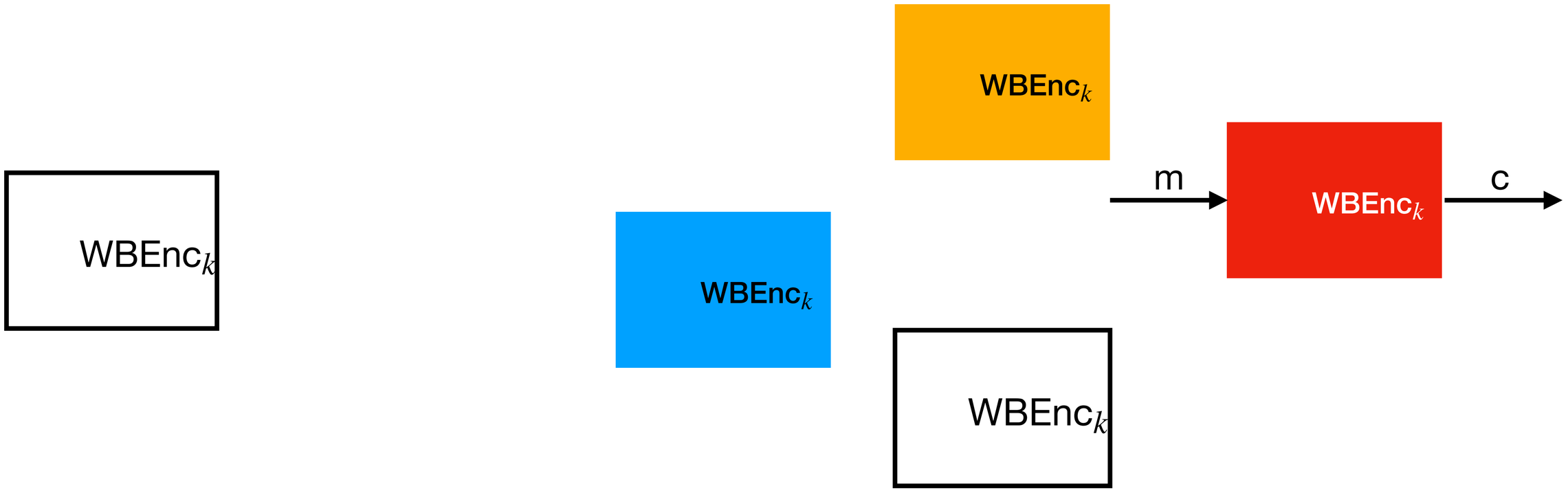
Code-lifting on white-box programs

- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks



Code-lifting on white-box programs

- White-box programs need to achieve security against key extraction
  - But white-box programs are also susceptible to *code-lifting* attacks



Mitigation against code-lifting

The white-box research community has proposed different security properties for mitigating code-lifting attacks:

Mitigation against code-lifting

The white-box research community has proposed different security properties for mitigating code-lifting attacks:

Incompressibility

Mitigation against code-lifting

The white-box research community has proposed different security properties for mitigating code-lifting attacks:

Incompressibility

Traceability

Mitigation against code-lifting

The white-box research community has proposed different security properties for mitigating code-lifting attacks:

Incompressibility

Traceability

Hardware-binding

Mitigation against code-lifting

The white-box research community has proposed different security properties for mitigating code-lifting attacks:

Incompressibility

Traceability

Hardware-binding

Software-binding

Mitigation against code-lifting

The white-box research community has proposed different security properties for mitigating code-lifting attacks:

Incompressibility

Traceability

Hardware-binding

Software-binding

Incompressibility

- Introduced by Delerablee, Lepoint, Pallier and Rivain in [1]
  - Idea: from a symmetric encryption scheme of conventional size - derive a large and incompressible, functional equivalent program
  - Incompressible meaning:
    - If fragments of the program are removed, the program loses its functionality
    - If the program is compressed, it loses its functionality
    - Given the incompressible program, It should be hard to derive the original (short) key of the program

Incompressibility

- Introduced by Delerablee, Lepoint, Pallier and Rivain in [1]
  - Idea: from a symmetric encryption scheme of conventional size - derive a large and incompressible, functional equivalent program
  - Incompressible meaning:
    - If fragments of the program are removed, the program loses its functionality
    - If the program is compressed, it loses its functionality
    - Given the incompressible program, It should be hard to derive the original (short) key of the program

[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - syntax

Incompressibility - syntax

$$SE = (\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Comp})$$

Incompressibility - syntax

SE = (KeyGen,Enc,Dec, Comp)

Comp(k)\$ $\longrightarrow$ WBInc

Incompressibility - syntax

$\mathsf{SE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Comp})$

$\mathsf{Comp}(k)\$ \longrightarrow \mathsf{WBInc}$

$\mathsf{Dec}(k, \mathsf{WBInc}(m)) = m$

Incompressibility - syntax

$\mathsf{SE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Comp})$

$\mathsf{Comp}(k)\$ \longrightarrow \mathsf{WBInc}$

$$\mathsf{Dec}(k, \mathsf{WBInc}(m)) = m$$

$$\mathsf{WBInc} \;>>\; \mathsf{Enc}(k, .)$$

Incompressibility - security

[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security

WBInc

[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

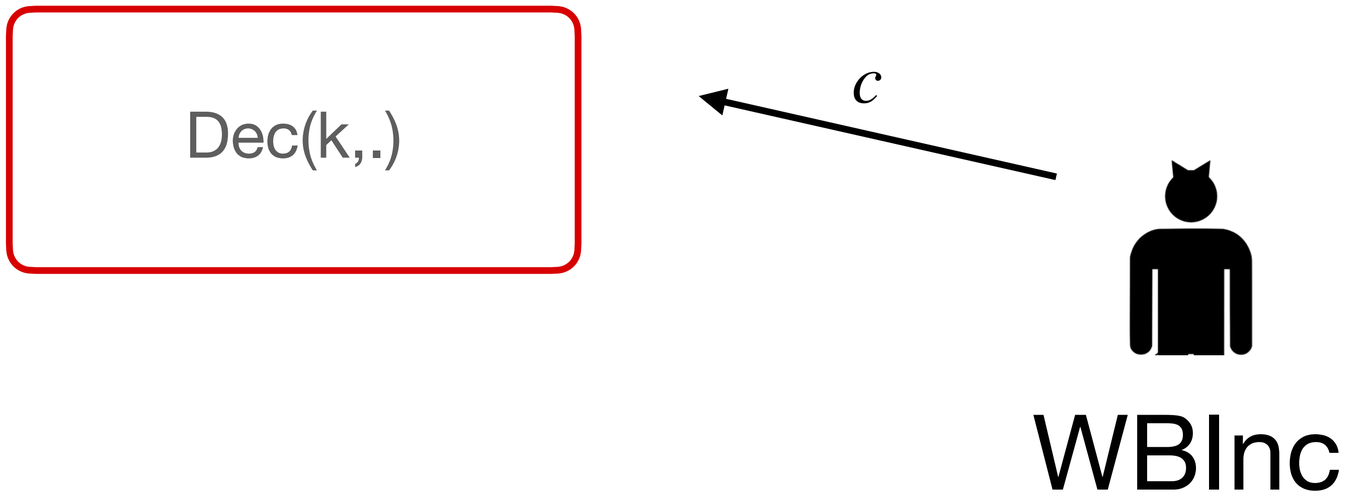
Incompressibility - security

Dec(k,.)

WBInc

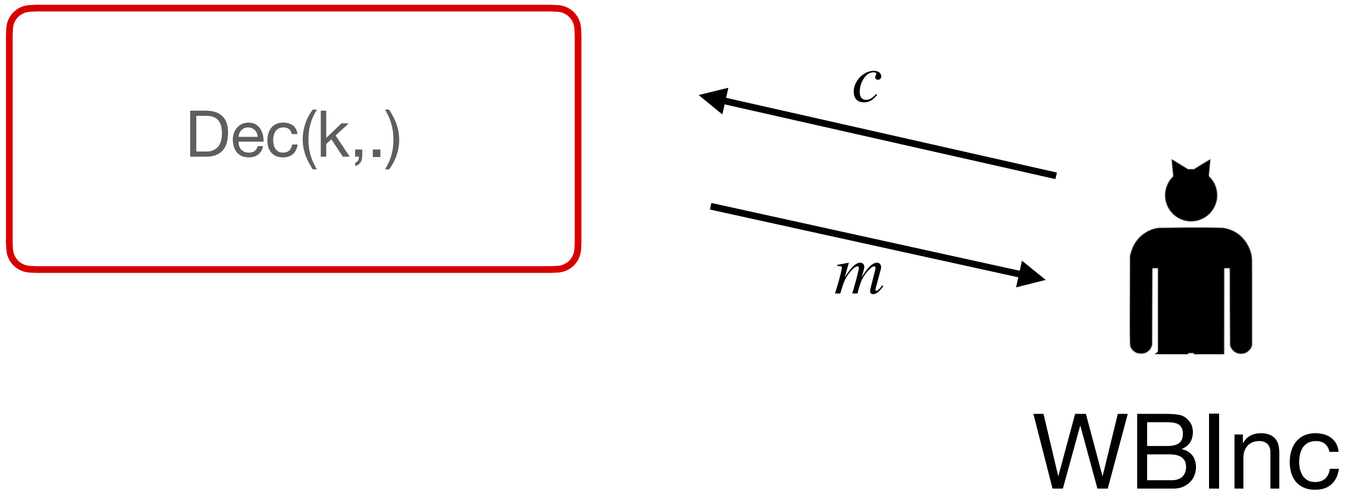
[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security



[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security

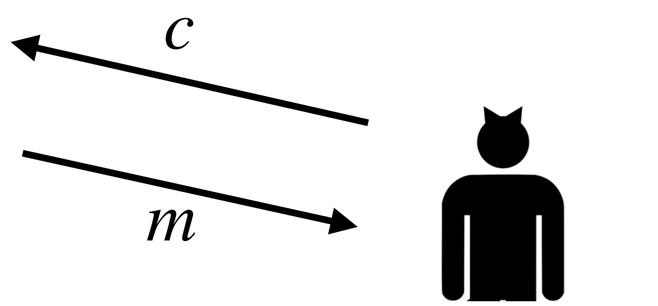


[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security

Dec(k,.)

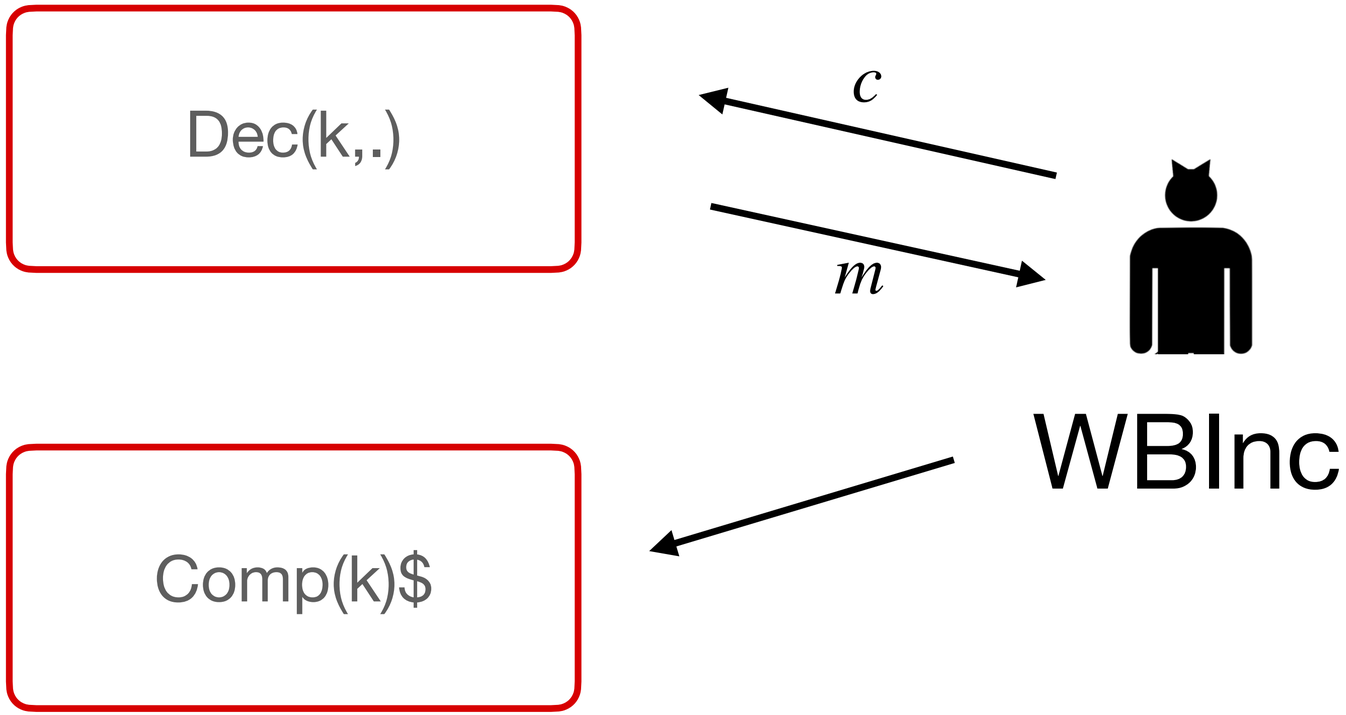
Comp(k)$



WBInc

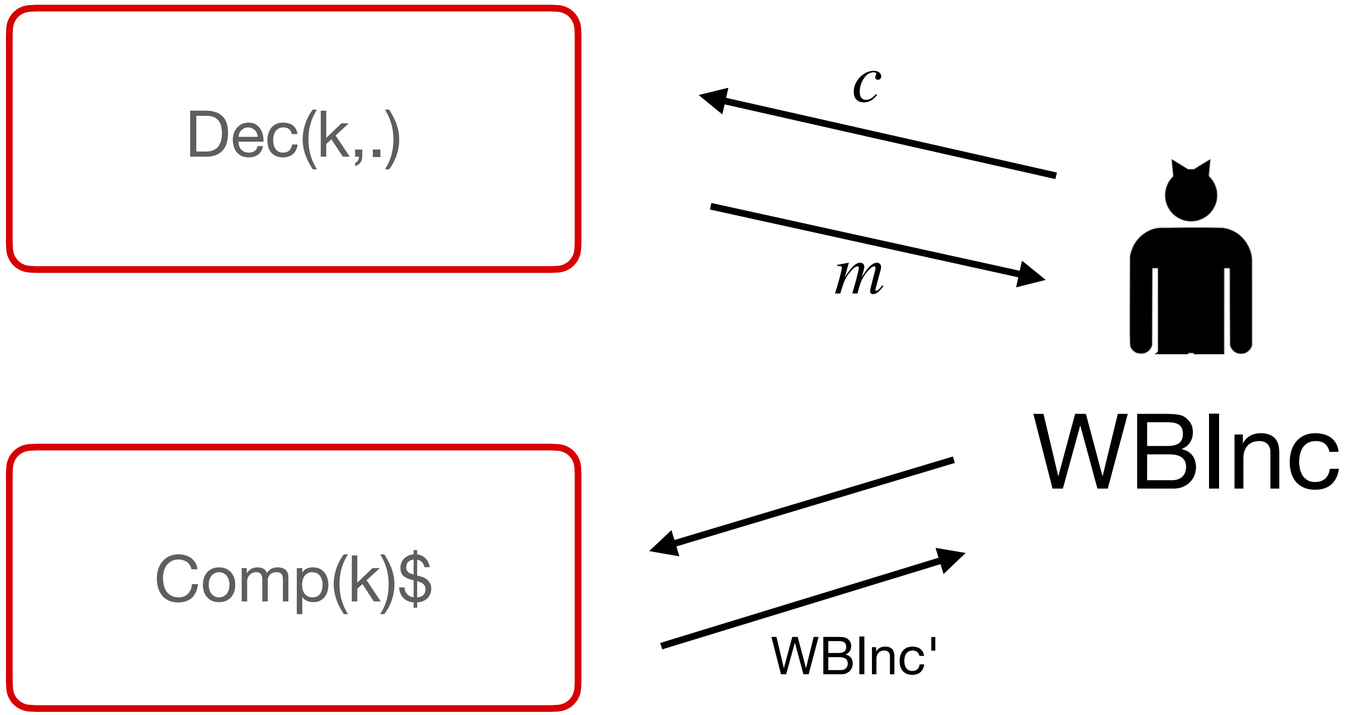
[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security



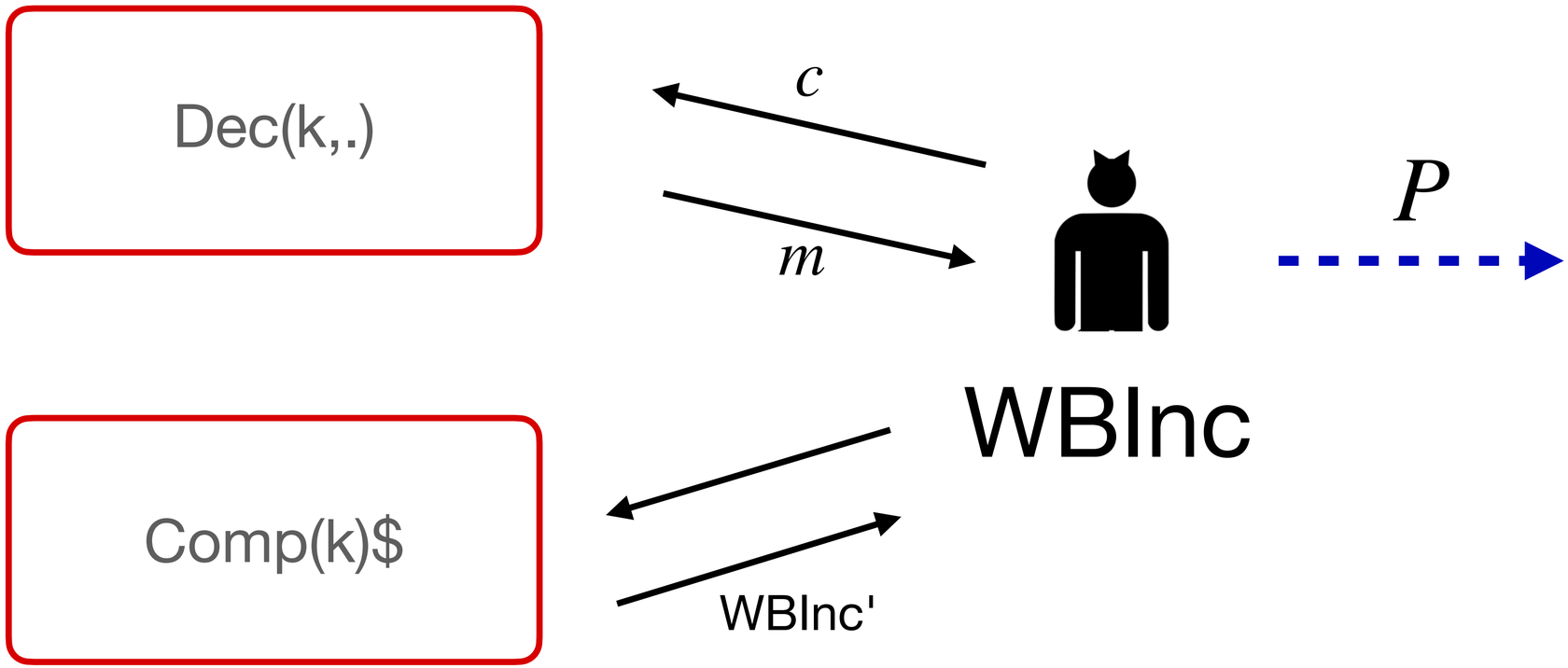
[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security



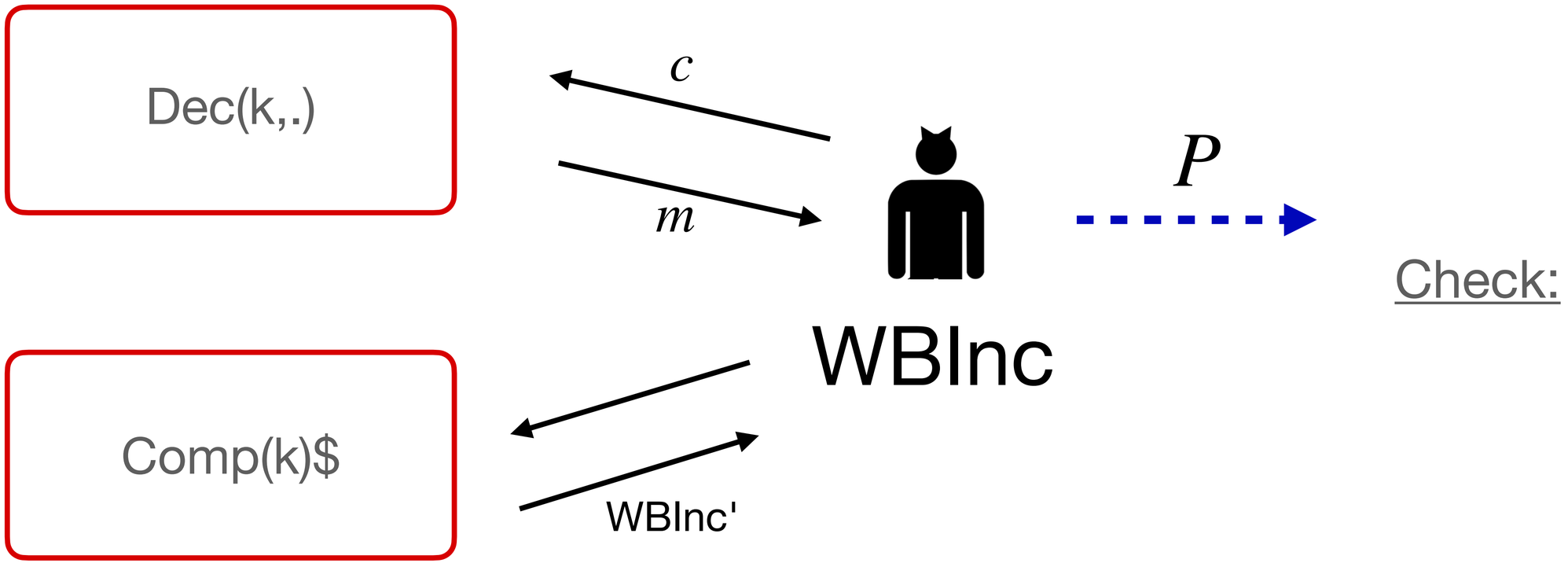
[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security



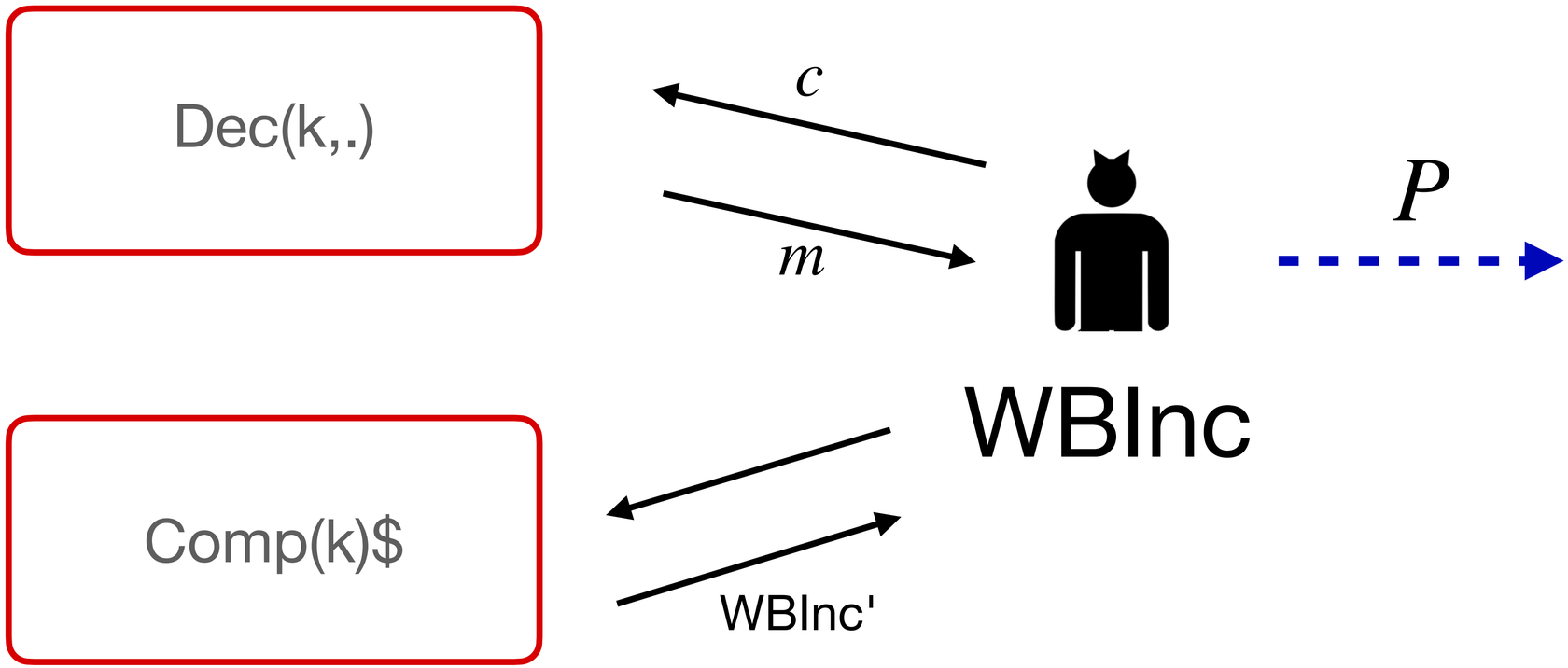
[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security



[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security

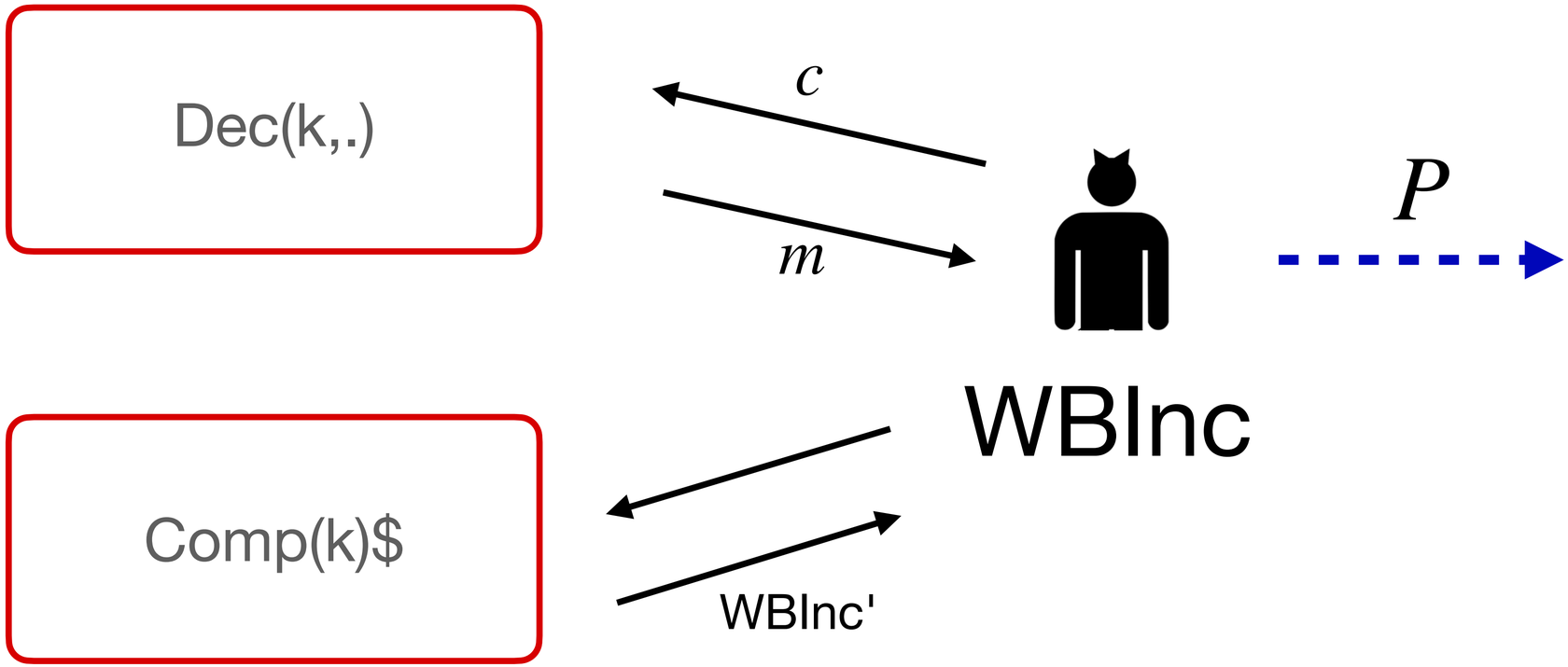


Check:

$$\mathrm{Dec}(k, P(m)) = m$$

[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Incompressibility - security



Check:

$\mathrm{Dec}(k, P(m)) = m$

$P << \mathrm{WBInc}$

[1] Delerablee et al.: White-box security notions for symmetric encryption schemes, SAC 2013

Strong Incompressibility

- Introduced by Fouque, Karpman, Kirchner and Minaud in [2]
  - Captures more security properties than the standard incompressibility
  - Essentially corresponds to an IND-CPA under leakage

[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility

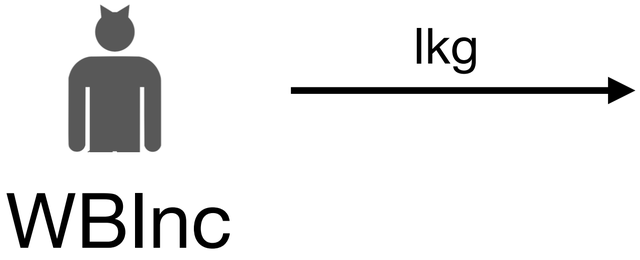
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility

WBInc

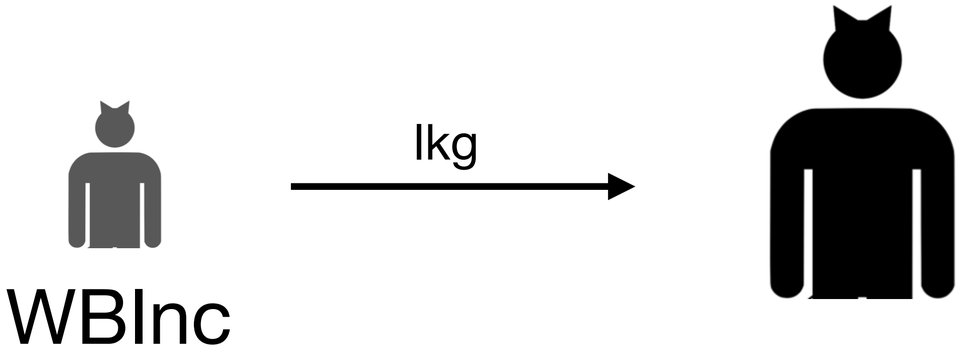
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



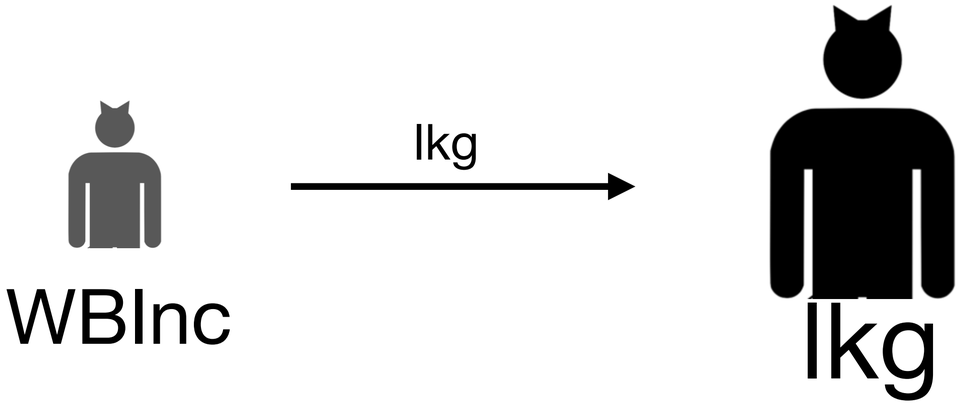
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



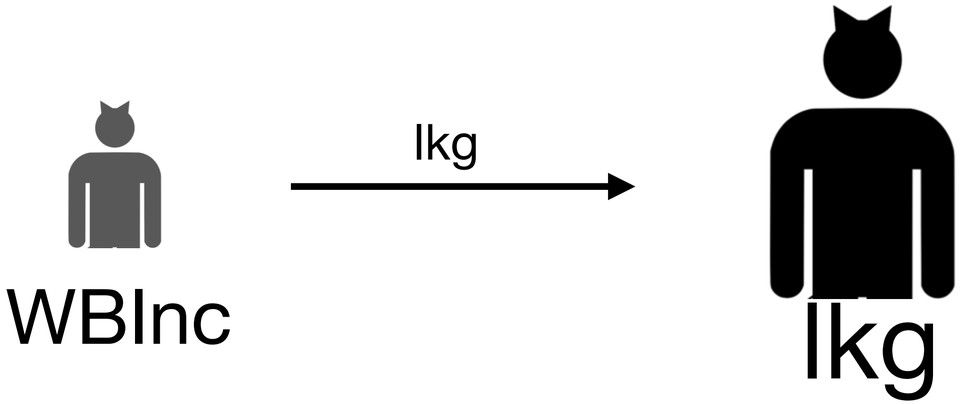
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

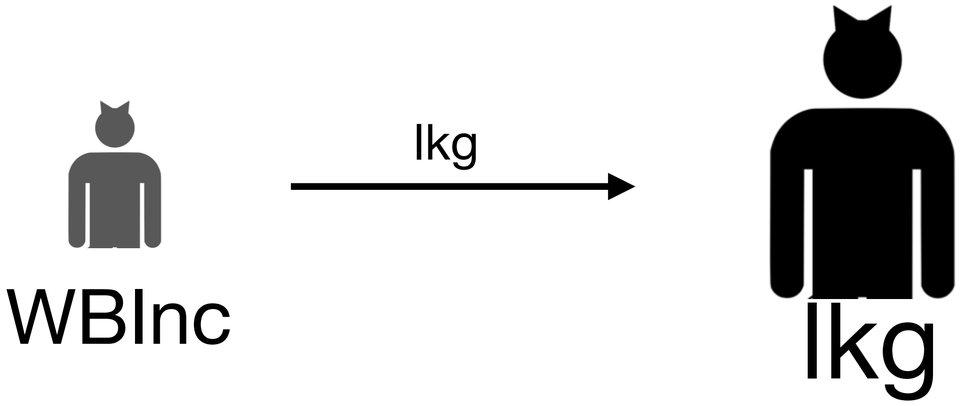
Strong Incompressibility



$\text{lkg} < \beta$

[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility

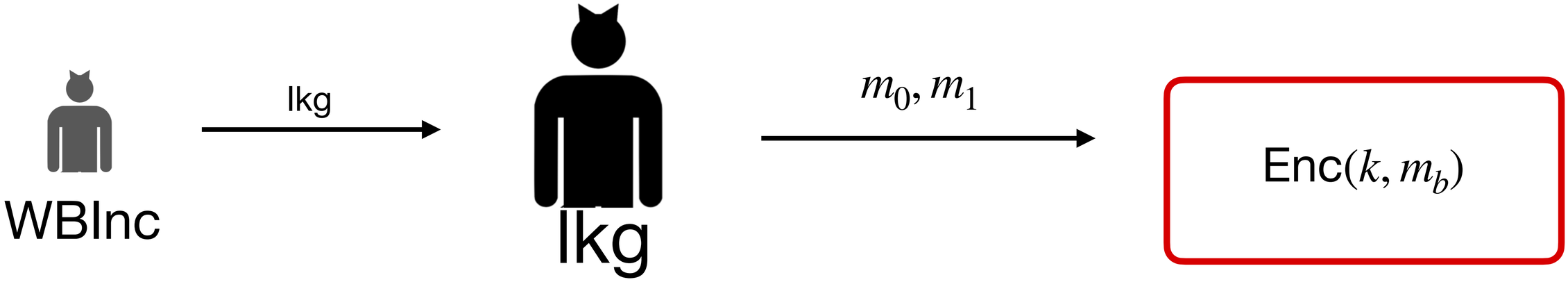


$\text{lkg} < \beta$

$\text{Enc}(k, m_b)$

[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

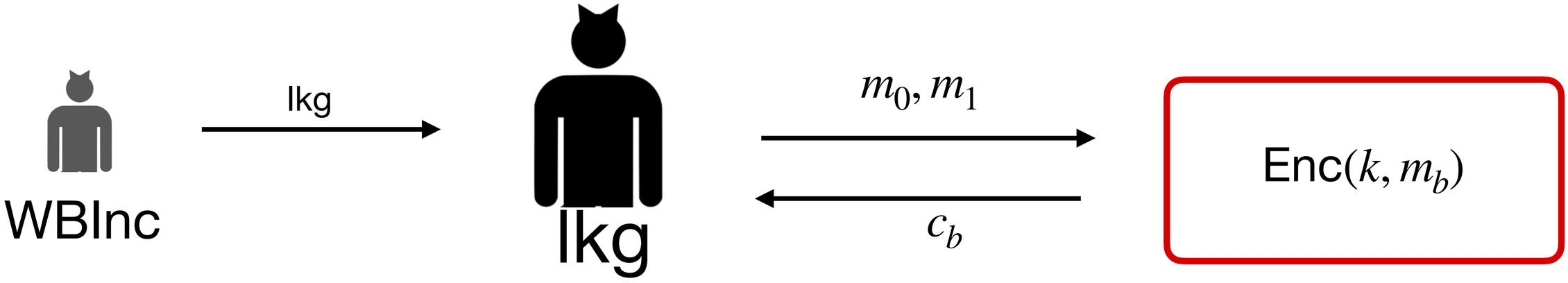
Strong Incompressibility



lkg $< \beta$

[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

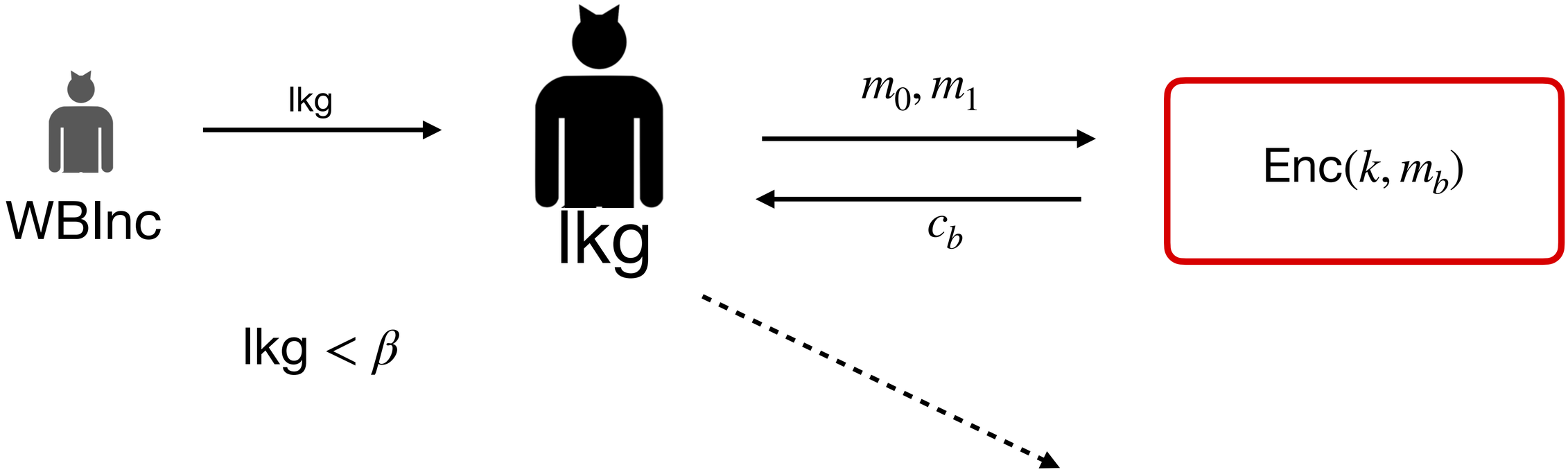
Strong Incompressibility



lkg $< \beta$

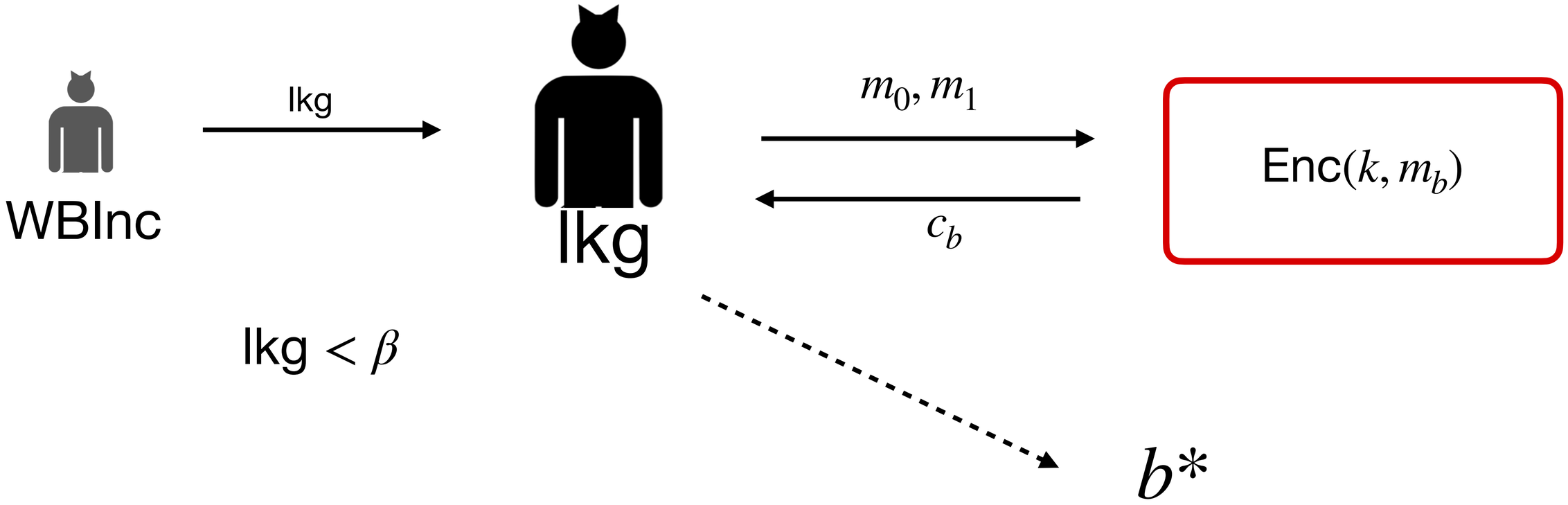
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



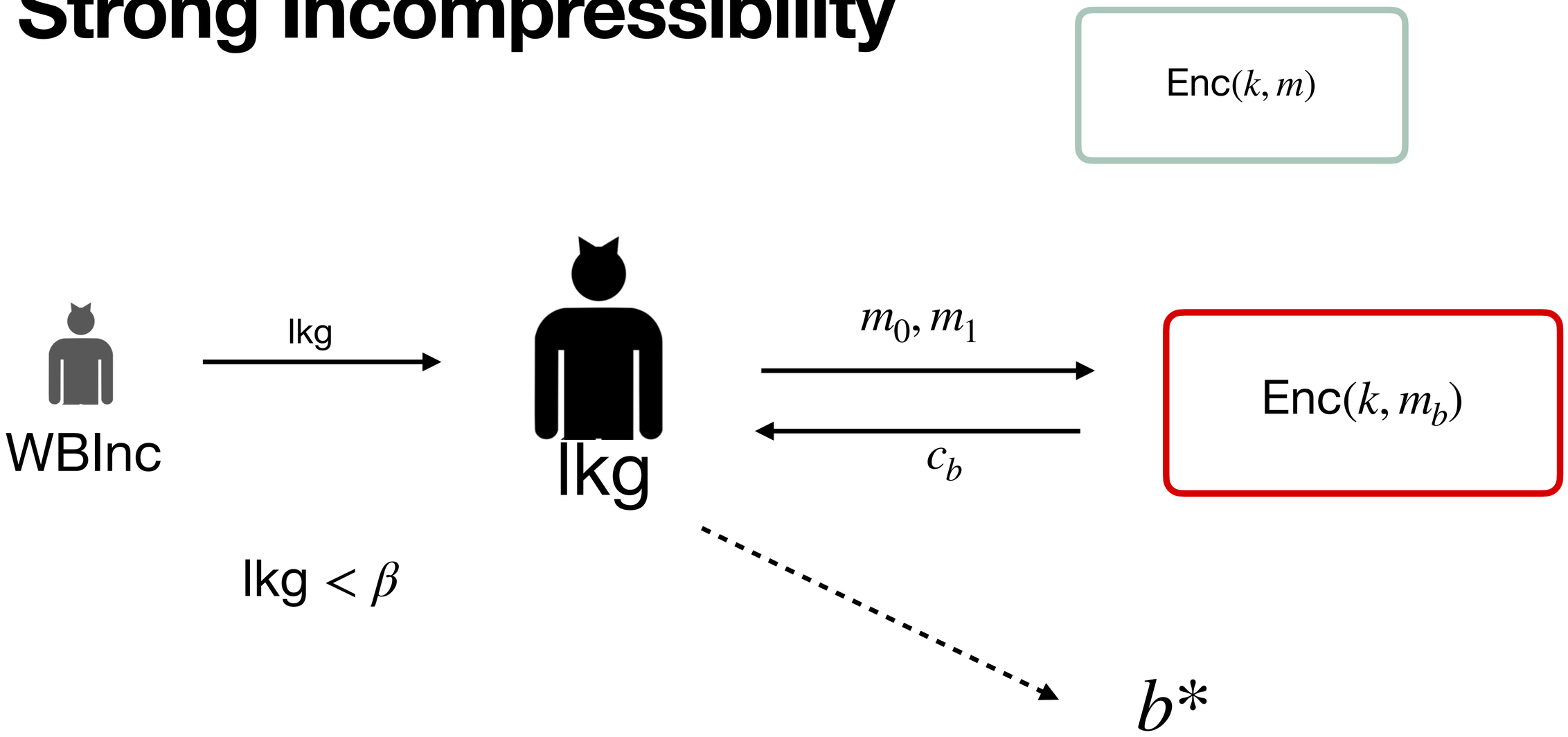
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



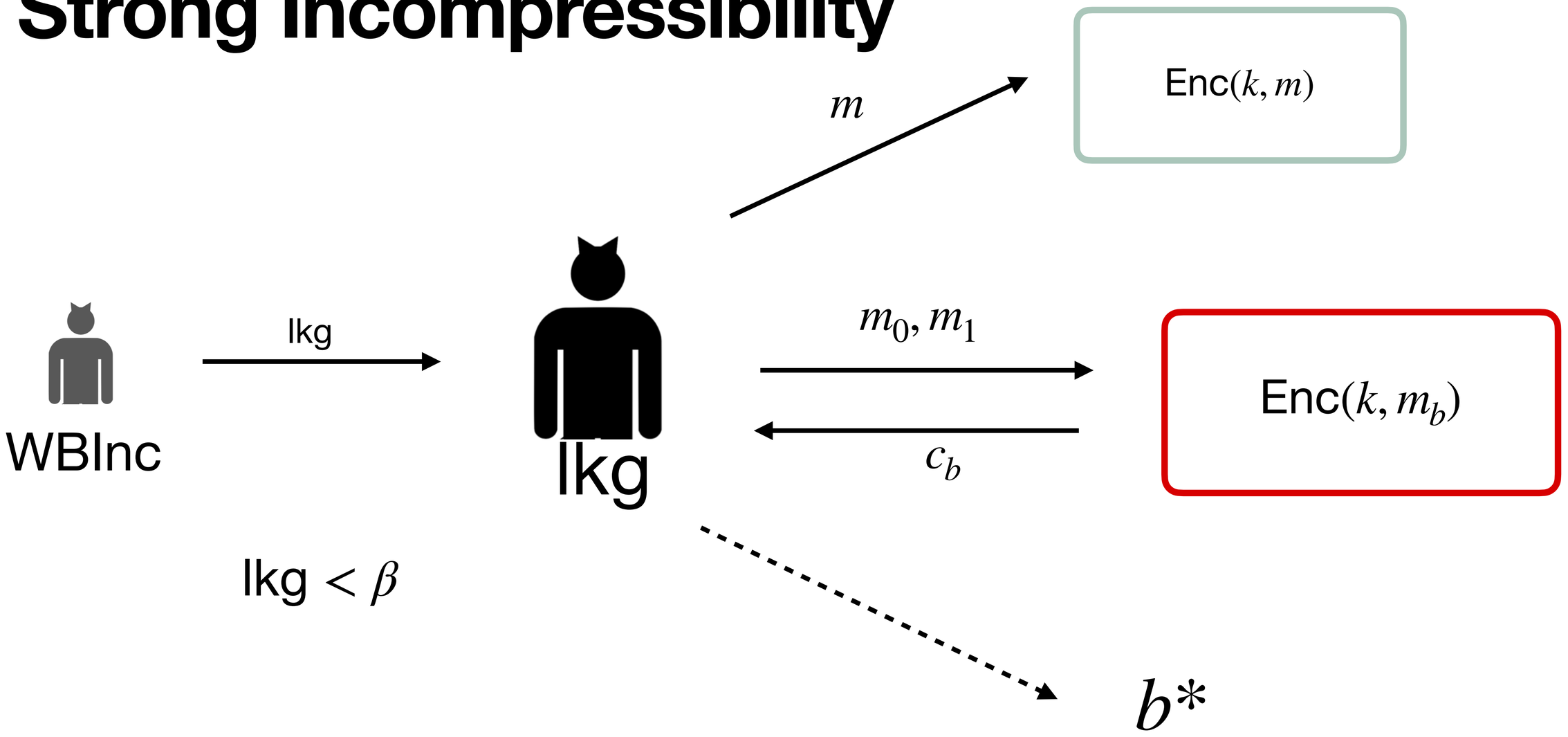
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



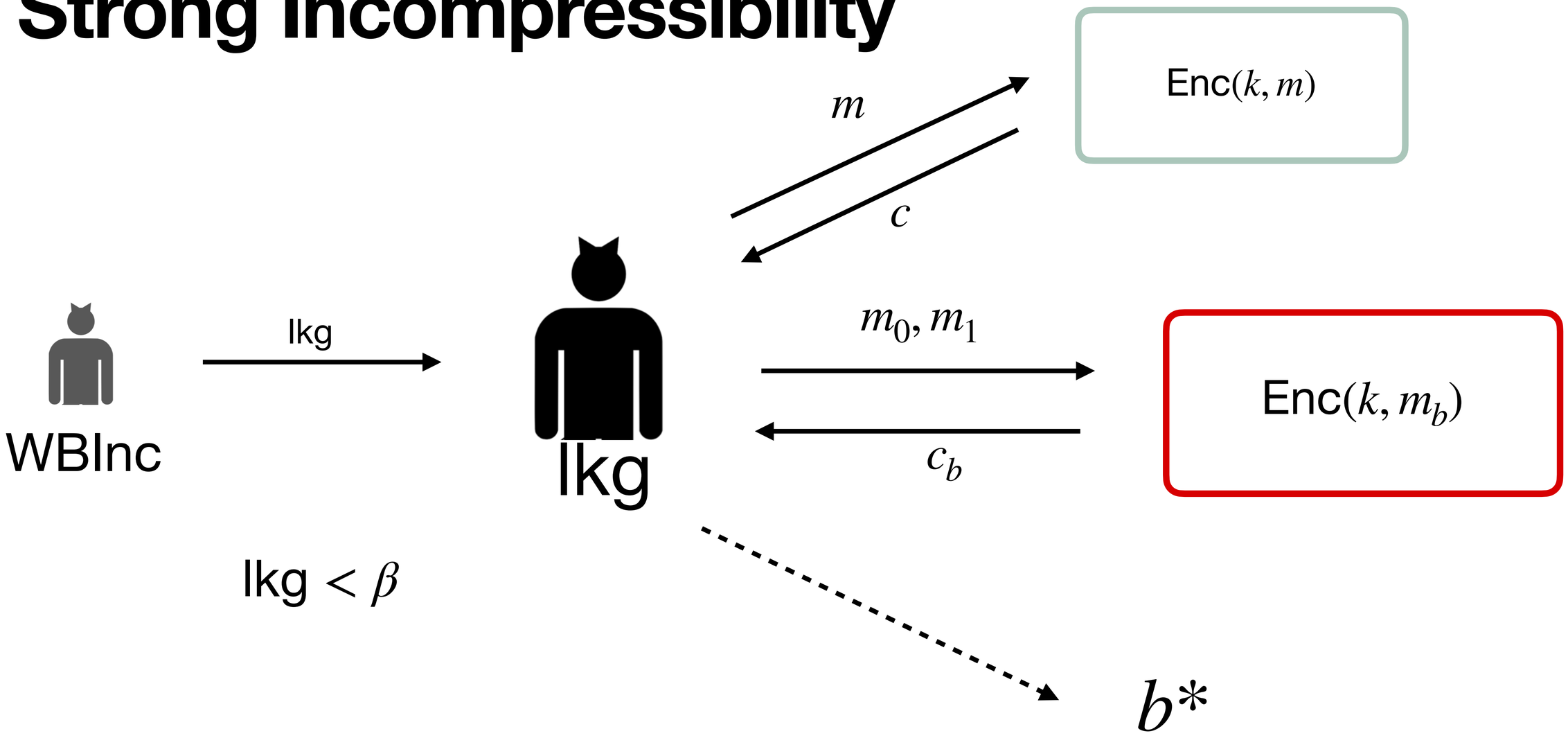
[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong Incompressibility



[2] Fouque et al.: Efficient and provable white-box primitives, ASIACRYPT 2016

Strong incompressibility & big key symmetric encryption

- The strong incompressibility model is essentially the same as IND-CPA under leakage
- This model was also considered by Bellare, Kane and Rogaway in [3] in the context of Big-key symmetric encryption
  - Aims to achieve security via large keys, but also considers methods for using those keys efficiently
- BKE [3] presents constructions and proves security in the random oracle model

[3] Bellare et al.: Big-key symmetric encryption: resisting key exfiltration. CRYPTO 2016

Construction (idea)

And motivation for this paper

Construction in the standard model

- We had an idea for a (strong) incompressible scheme in the standard model
- The construction followed an (only) incompressible secure construction based on one-way permutations and published [4]

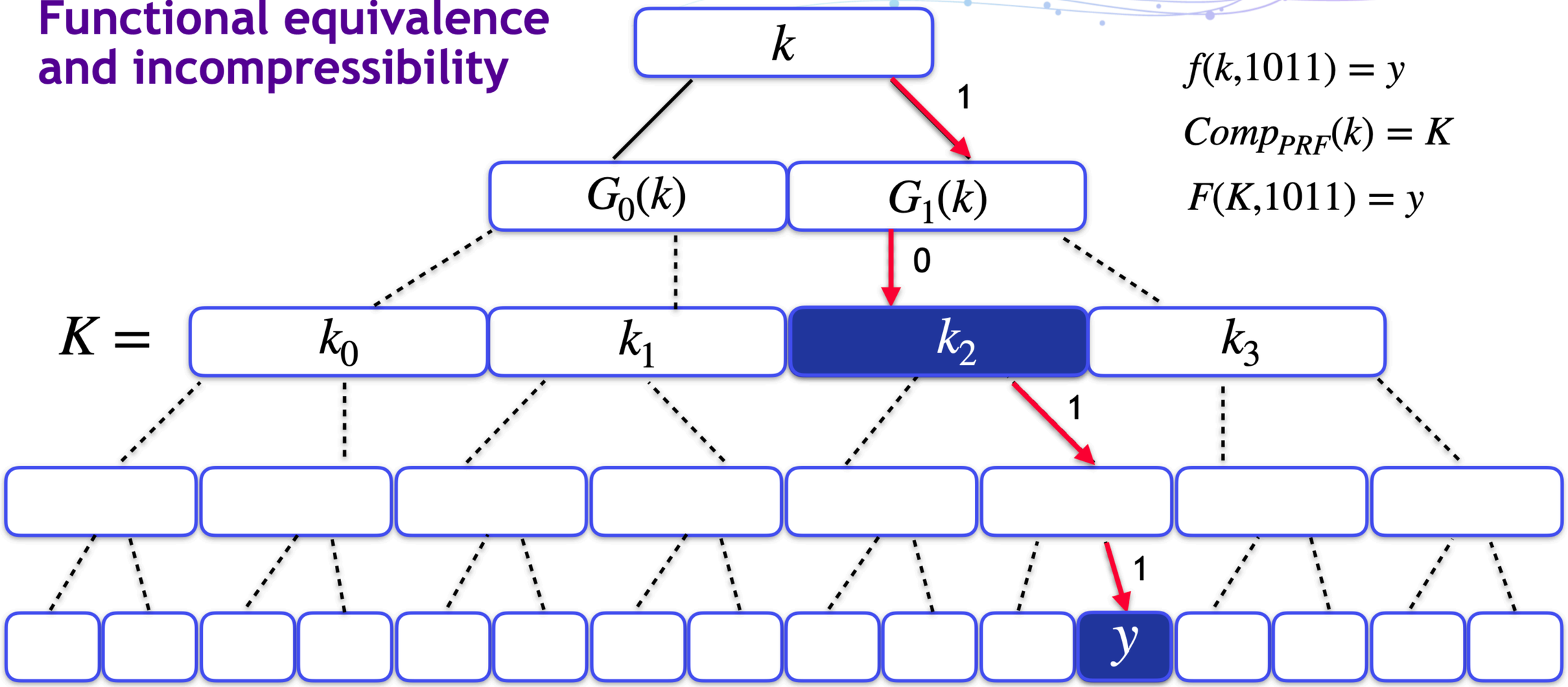
[4] Alpirez Bock et al.: Doubly-half injective PRGs for incompressible white-box cryptography, CT-RSA 2019

Construction in the standard model

- We had an idea for a (strong) incompressible scheme in the standard model
- The construction followed an (only) incompressible secure construction based on one-way permutations and published [4]

[4] Alpirez Bock et al.: Doubly-half injective PRGs for incompressible white-box cryptography, CT-RSA 2019

Functional equivalence and incompressibility

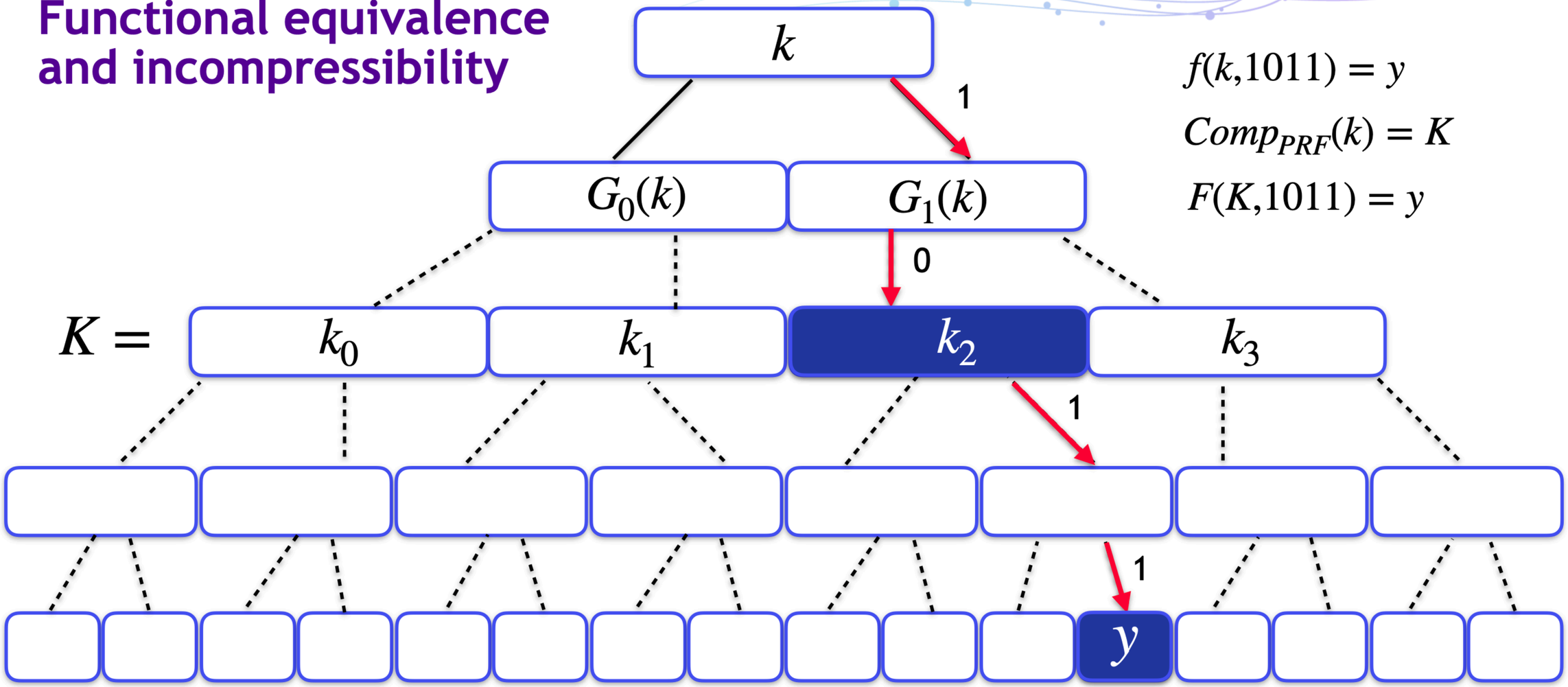


$f(k,1011) = y$

$Comp_{PRF}(k) = K$

$F(K,1011) = y$

Functional equivalence and incompressibility



$f(k,1011) = y$

$Comp_{PRF}(k) = K$

$F(K,1011) = y$

Construction in the standard model

- Idea: use the same construction for generating PRF values and use them for encryption

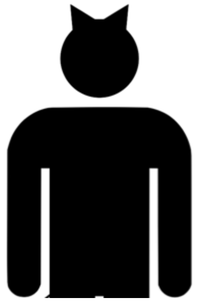
$$x \longleftarrow \$; \qquad c := (\mathrm{PRF}(K, x) \oplus m, \quad x)$$

- Or: just use the PRF as a key generator and then use a conventional encryption scheme
  - Given only some leakage of the large key K, it *should* be difficult to win the IND-CPA game

[4] Alpirez Bock et al.: Doubly-half injective PRGs for incompressible white-box cryptography, CT-RSA 2019

WBInc

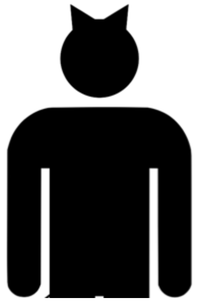
$\mathrm{lkg} = k_2$



$\mathrm{lkg} = k_2$

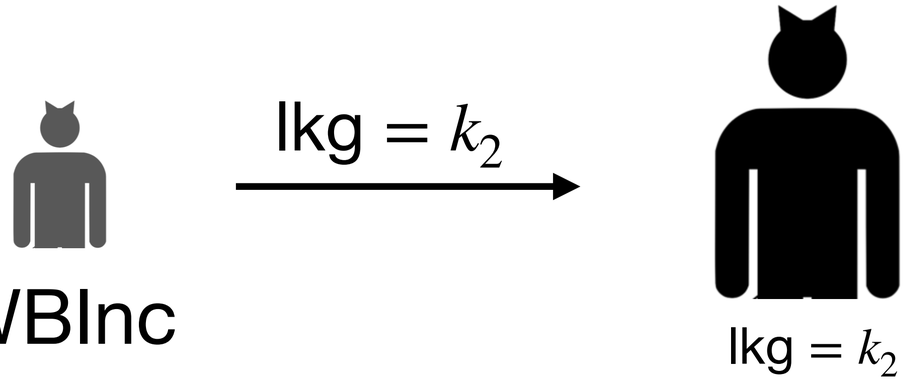
$\mathrm{lkg} < \beta$

WBInc



lkg = k_2

lkg < β



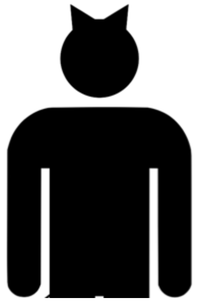
lkg $< \beta$

$m_1, m_2, \ldots, m_i$

Enc(k, m)

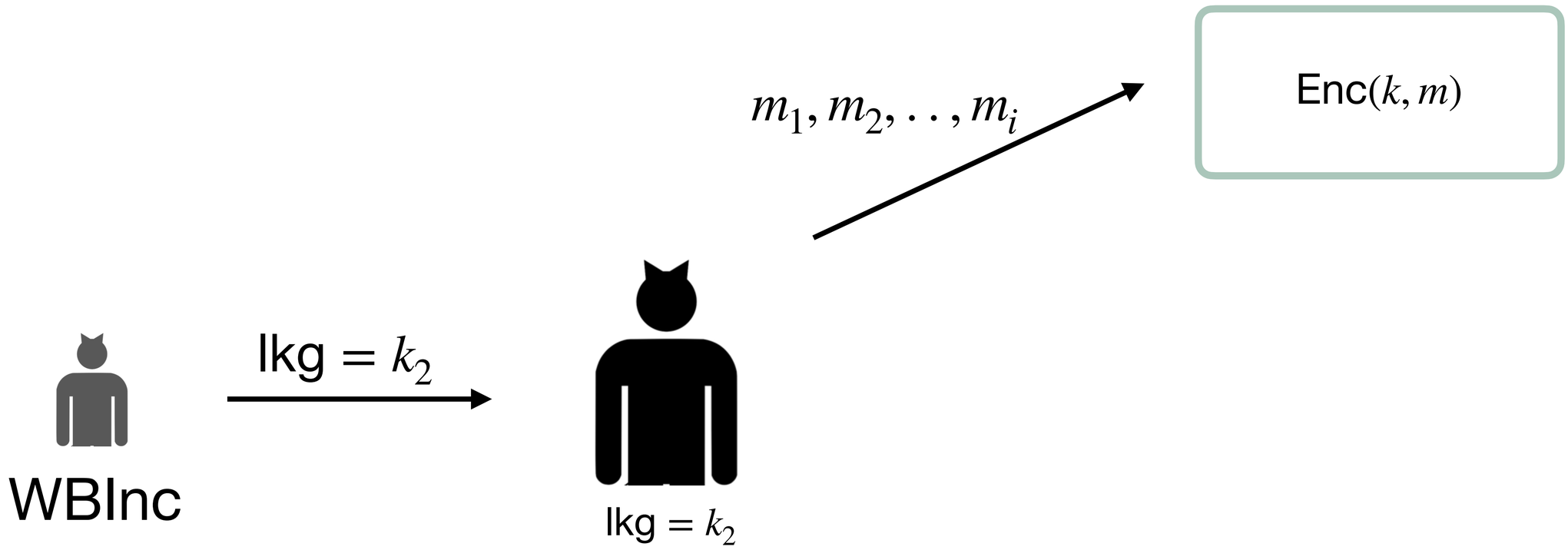
WBInc

lkg $= k_2$

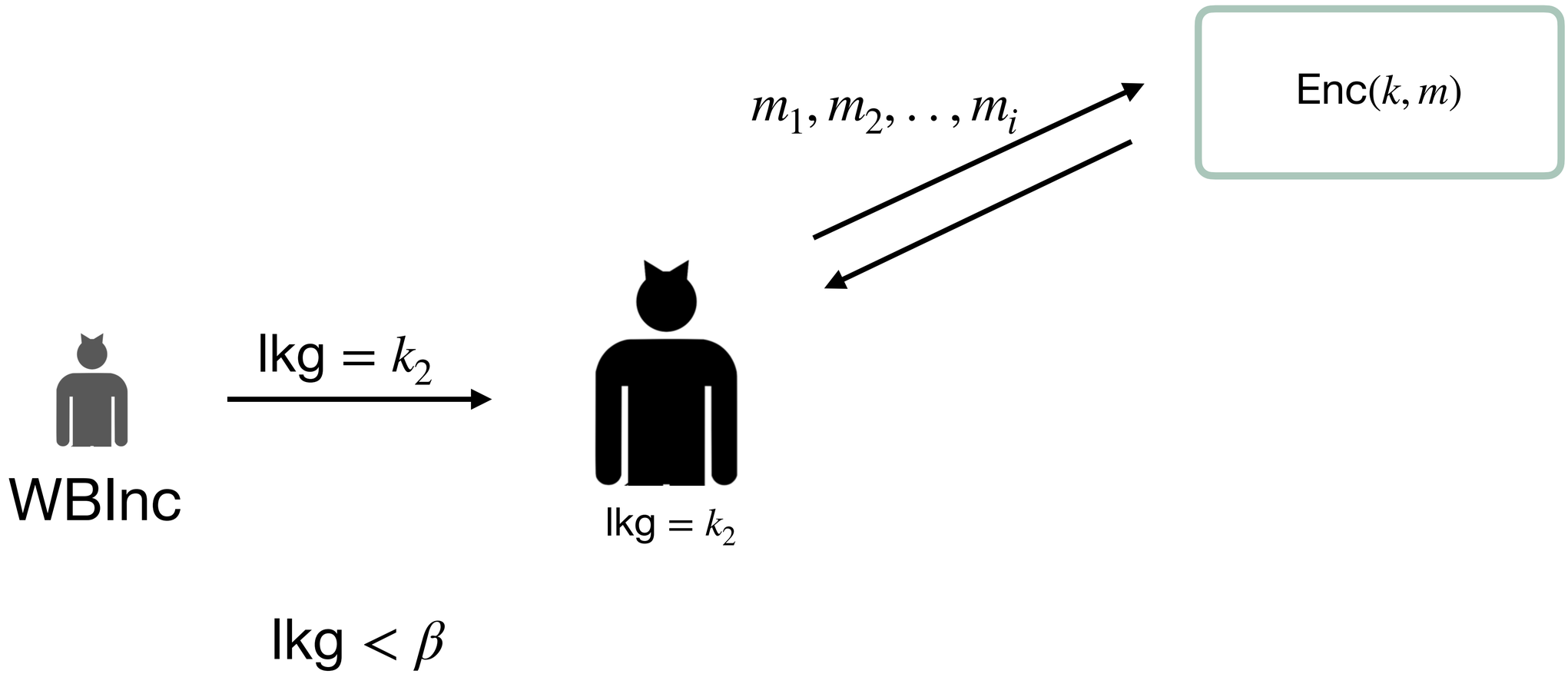


lkg $= k_2$

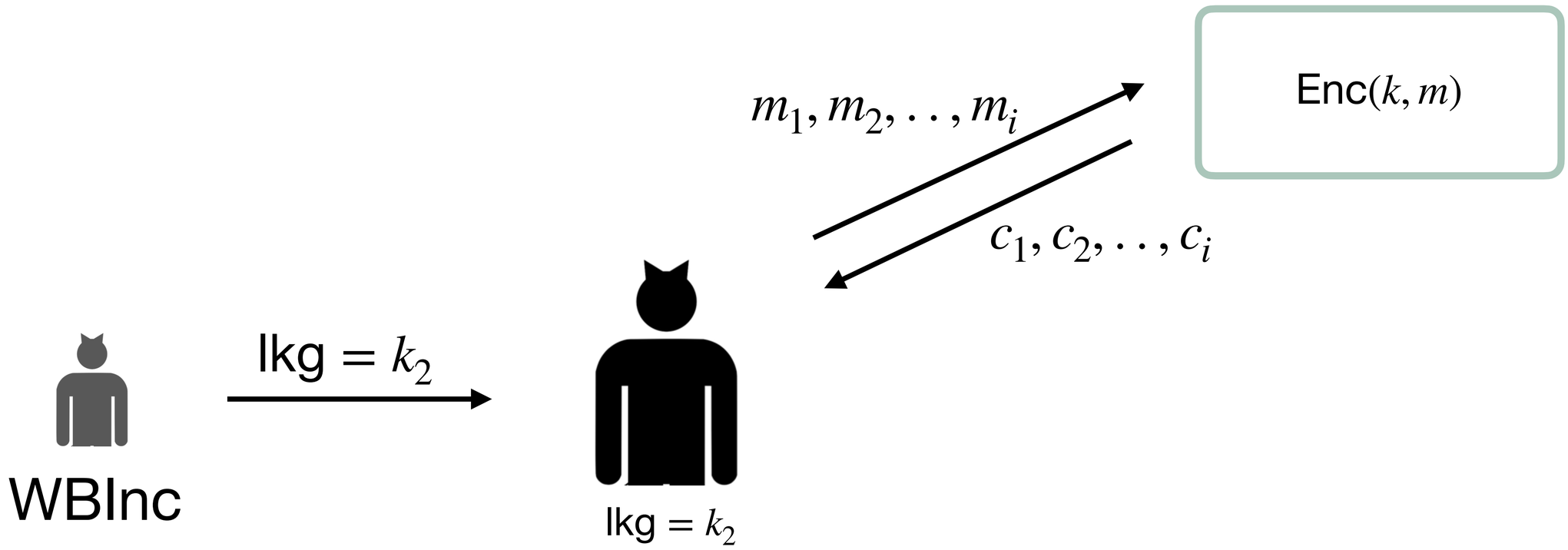
lkg $< \beta$



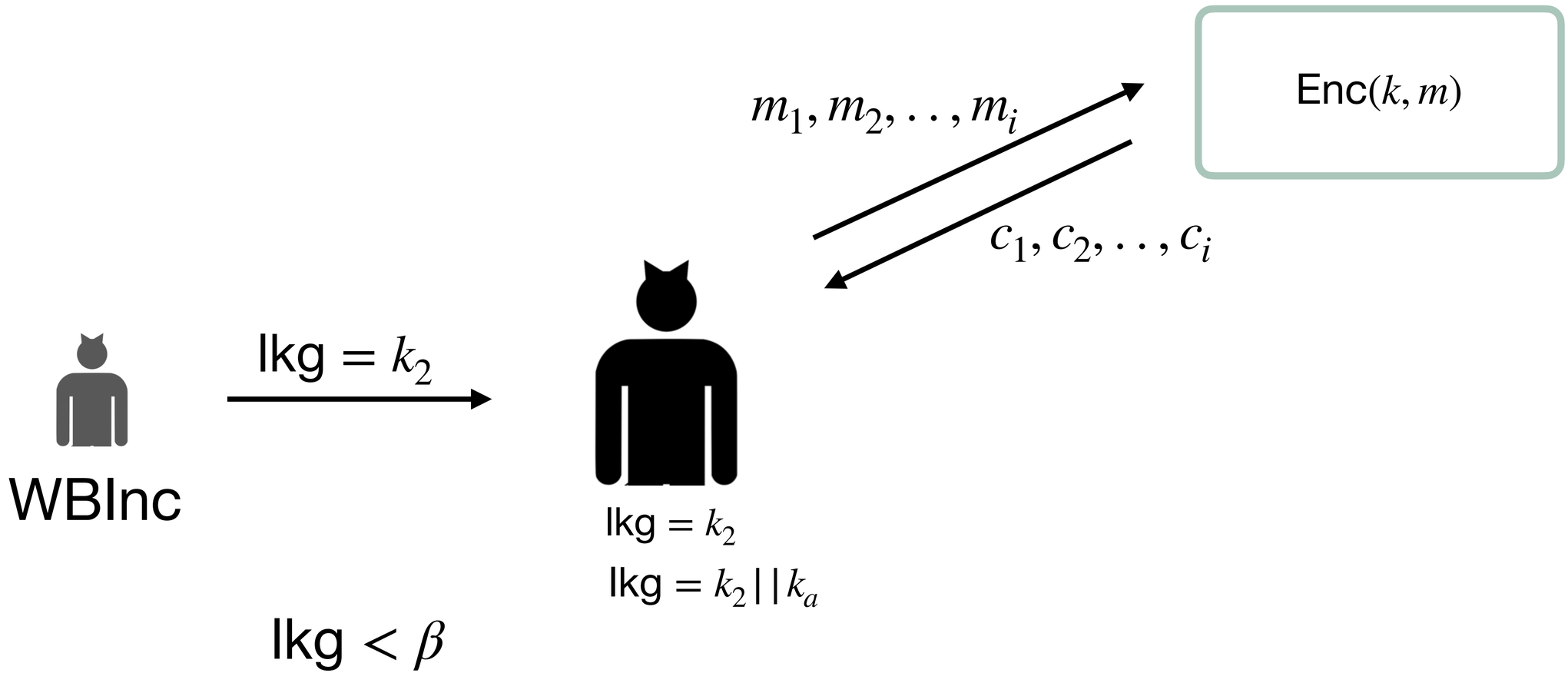
lkg $< \beta$

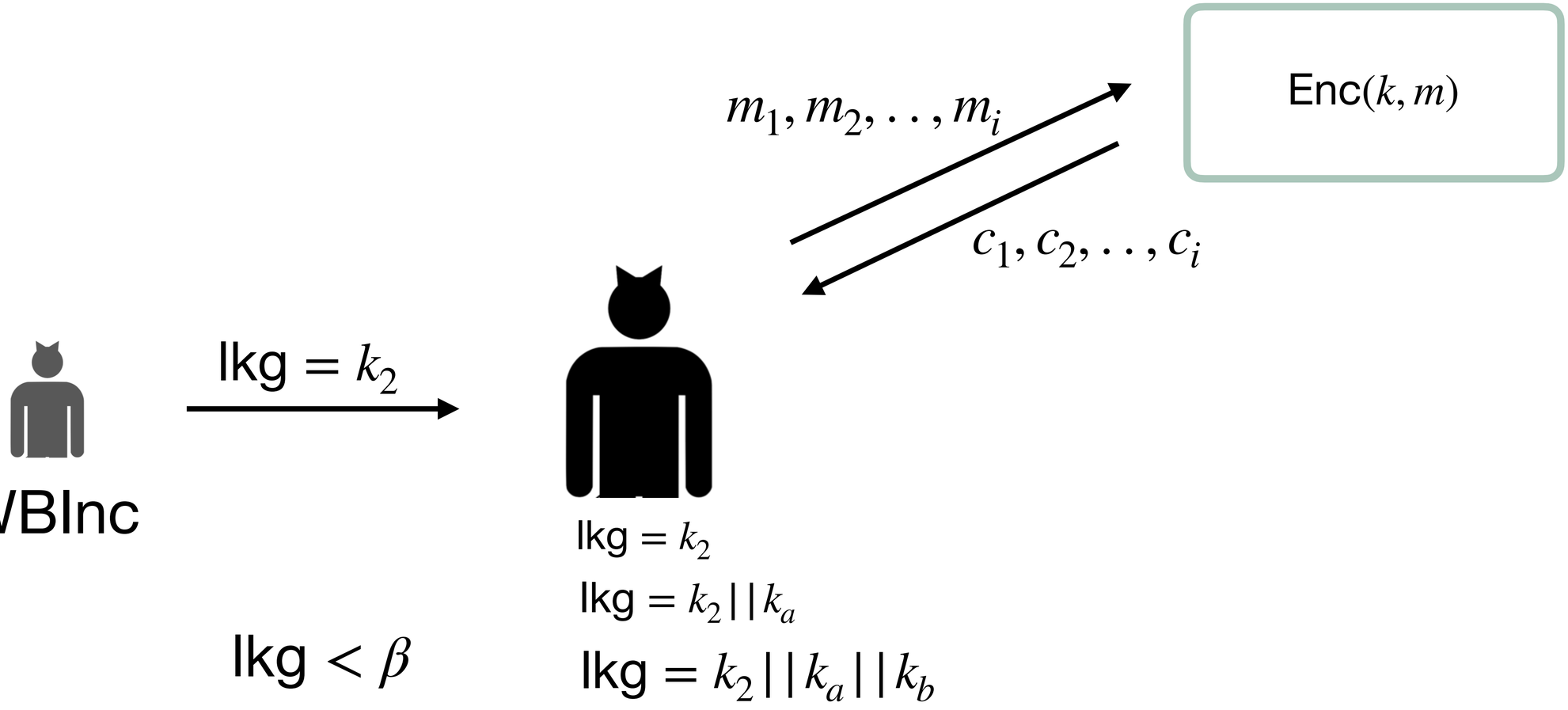


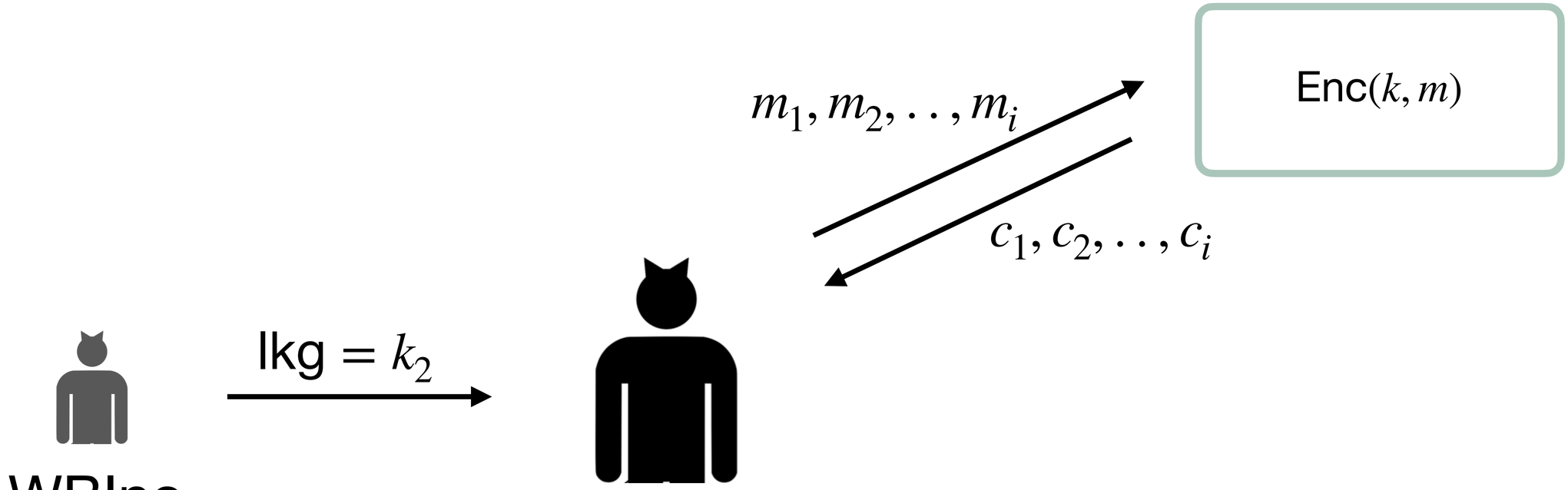
lkg $< \beta$

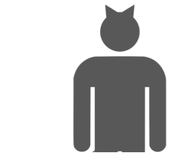


lkg $< \beta$


$m_1, m_2, \ldots, m_i$
Enc(k, m)
$c_1, c_2, \ldots, c_i$
lkg $= k_2$
WBInc
lkg $= k_2$
lkg $= k_2 || k_a$
lkg $< \beta$

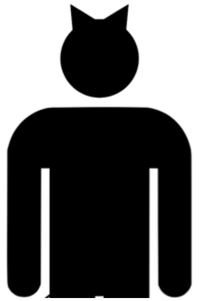

Enc(k, m)
$m_1, m_2, \ldots, m_i$
$c_1, c_2, \ldots, c_i$
lkg $= k_2$
WBInc
lkg $= k_2$
lkg $= k_2 || k_a$
lkg $= k_2 || k_a || k_b$
lkg $< \beta$

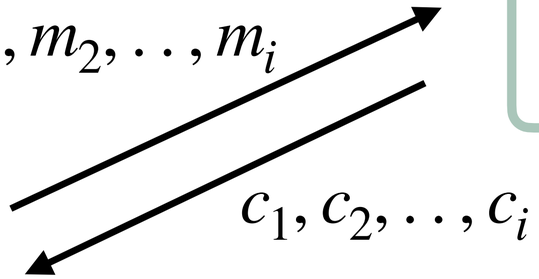

Enc(k, m)
$m_1, m_2, \ldots, m_i$
$c_1, c_2, \ldots, c_i$
lkg $= k_2$
WBInc
lkg $= k_2$
lkg $= k_2 || k_a$
lkg $< \beta$
lkg $= k_2 || k_a || k_b$
lkg $= k_2 || k_a || k_b || \ldots || k_i$



WBInc

$\text{lkg} = k_2$





$m_1, m_2, \ldots, m_i$

$c_1, c_2, \ldots, c_i$

Enc(k, m)

$\text{lkg} = k_2$

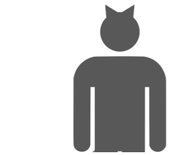
$\text{lkg} = k_2 || k_a$

$\text{lkg} < \beta$

$\text{lkg} = k_2 || k_a || k_b$

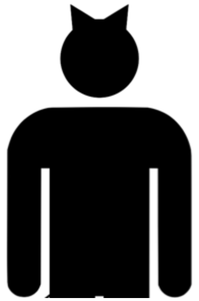
$\text{lkg} = k_2 || k_a || k_b || \ldots || k_i$

$\text{lkg} > \beta$



WBInc

$\text{lkg} = k_2$



$\text{lkg} = k_2$

$\text{lkg} = k_2 || k_a$

$\text{lkg} < \beta$

$\text{lkg} = k_2 || k_a || k_b$

$\text{lkg} = k_2 || k_a || k_b || .. || k_i$

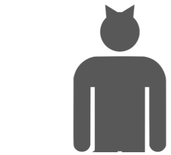
$\text{lkg} > \beta$

$m_1, m_2, \ldots, m_i$

$c_1, c_2, \ldots, c_i$

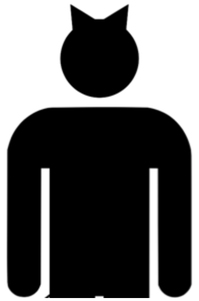
m_0, m_1

Enc(k, m)



WBInc

$\text{lkg} = k_2$



$m_1, m_2, \ldots, m_i$

$c_1, c_2, \ldots, c_i$

Enc(k, m)

m_0, m_1

$\text{lkg} = k_2$

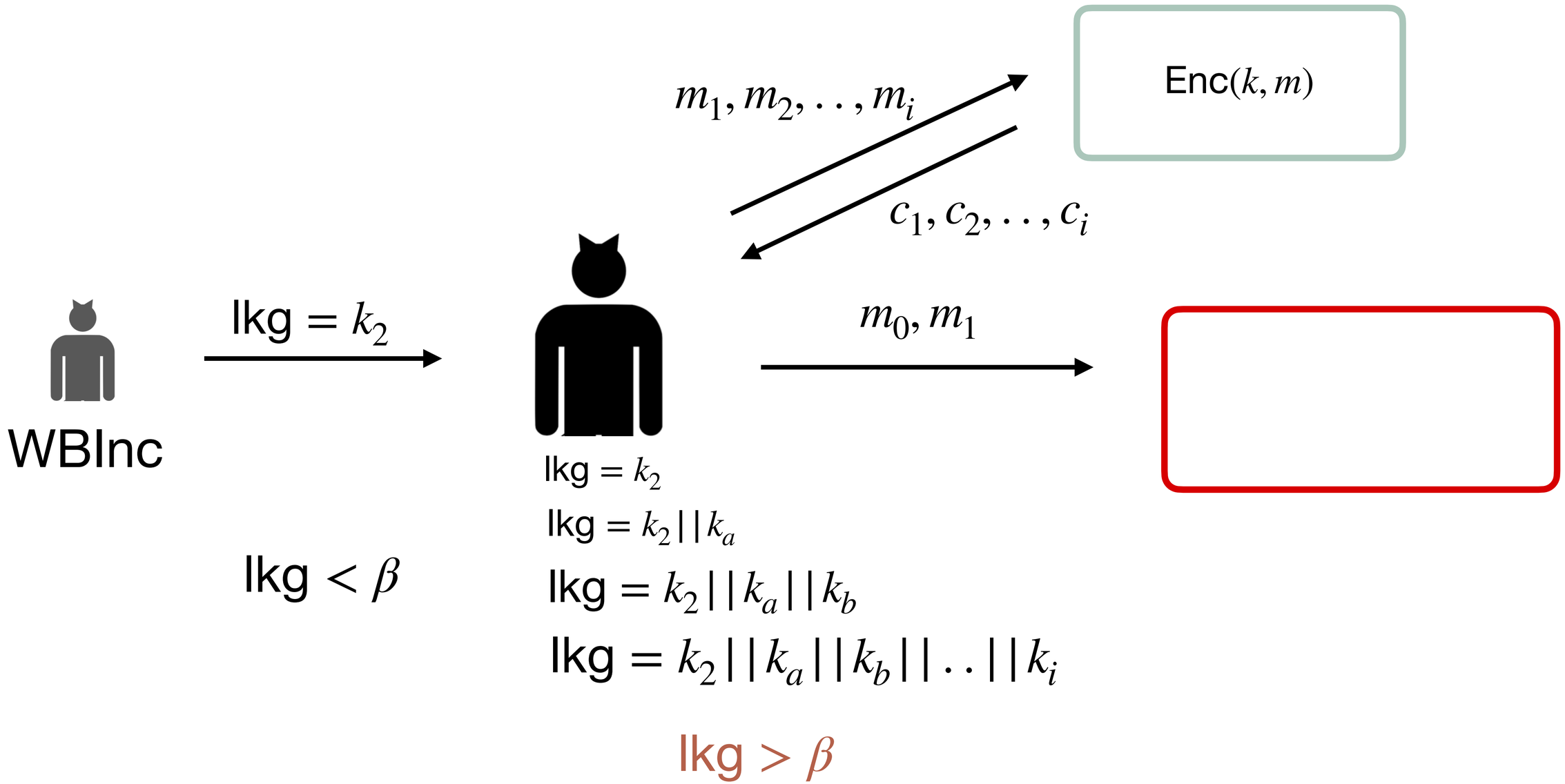
$\text{lkg} = k_2 || k_a$

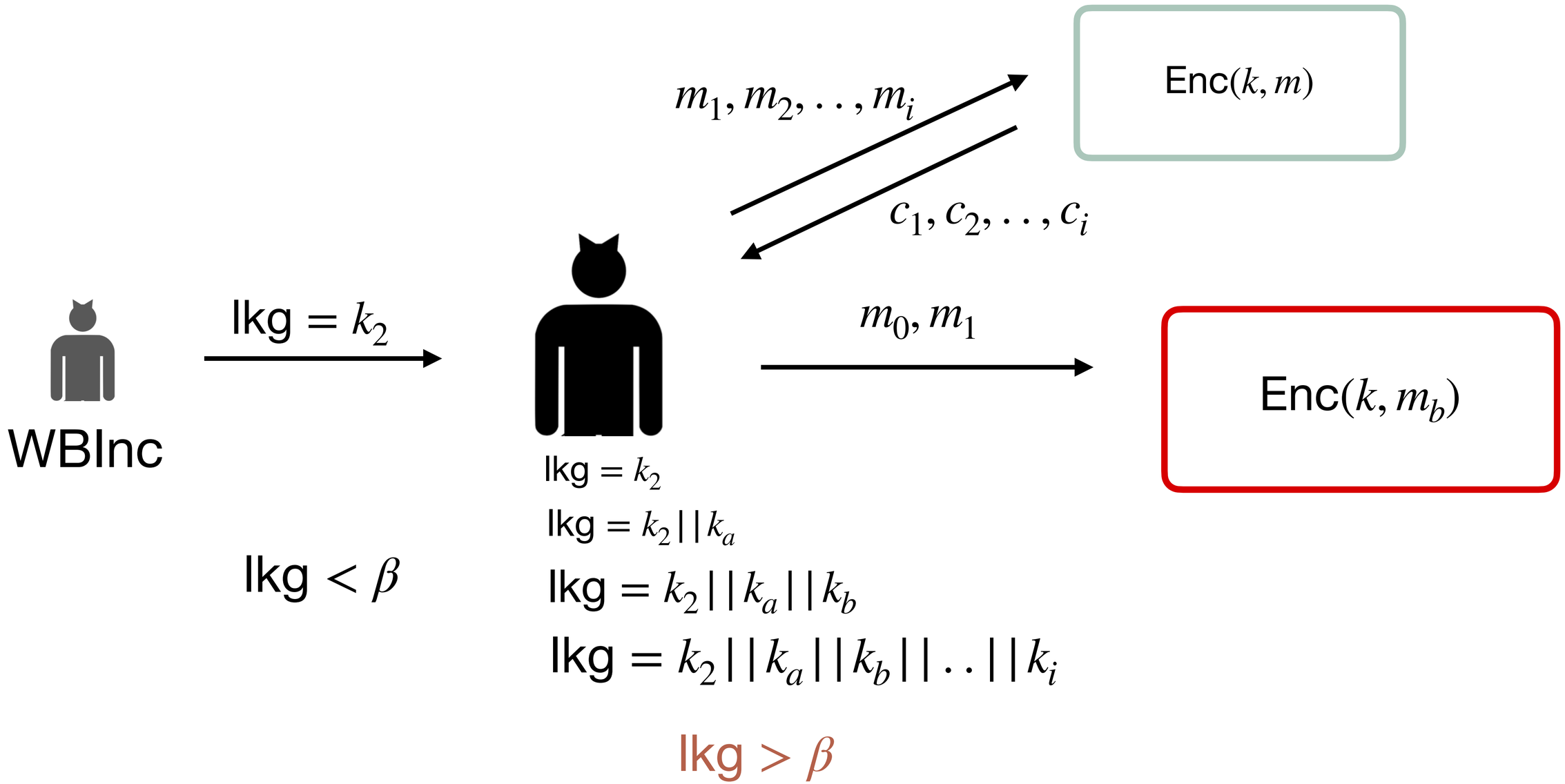
$\text{lkg} < \beta$

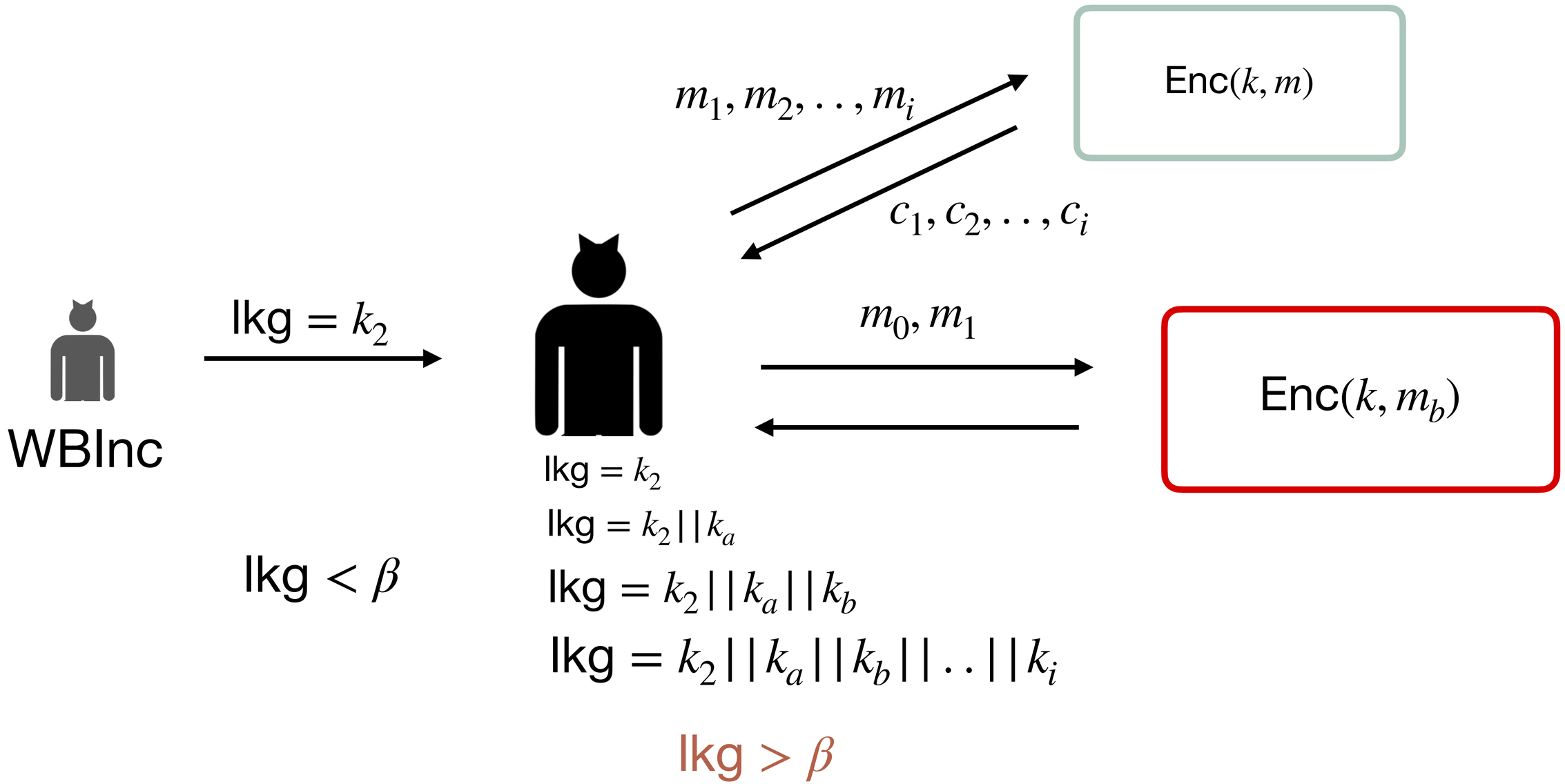
$\text{lkg} = k_2 || k_a || k_b$

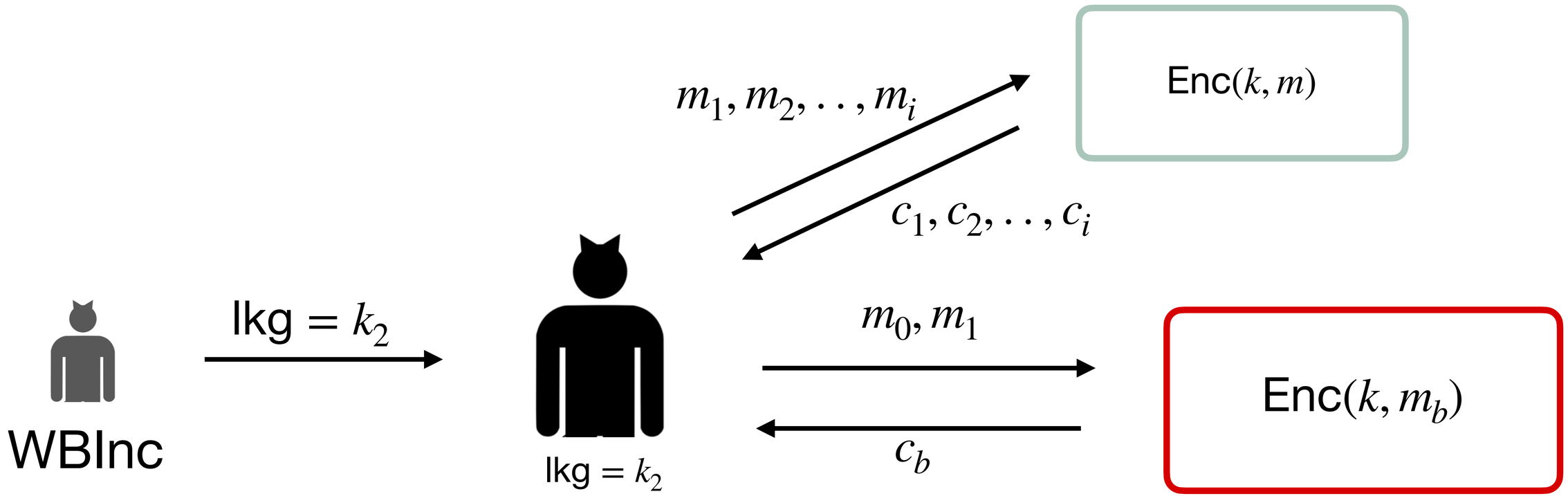
$\text{lkg} = k_2 || k_a || k_b || \ldots || k_i$

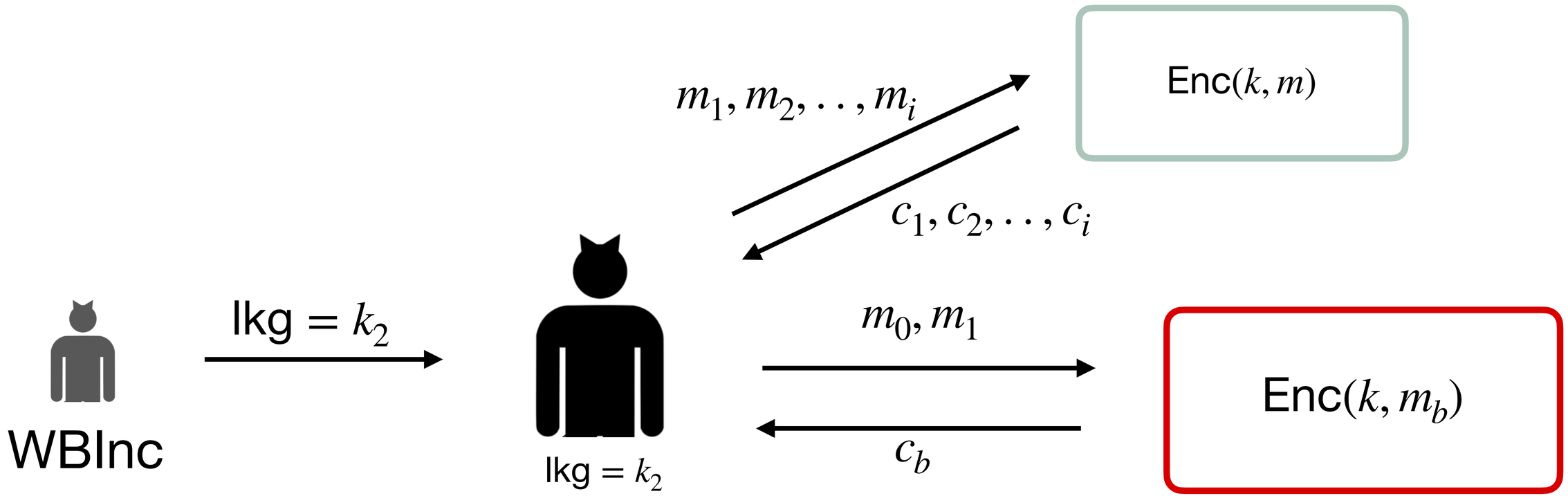
$\text{lkg} > \beta$


Enc(k, m)
$m_1, m_2, \ldots, m_i$
$c_1, c_2, \ldots, c_i$
WBInc
lkg $= k_2$
m_0, m_1
lkg $= k_2$
lkg $= k_2 || k_a$
lkg $< \beta$
lkg $= k_2 || k_a || k_b$
lkg $= k_2 || k_a || k_b || \ldots || k_i$
lkg $> \beta$


Enc(k, m)
m1, m2, . . , mi
c1, c2, . . , ci
lkg = k2
WBInc
m0, m1
Enc(k, mb)
lkg = k2
lkg = k2||ka
lkg < β
lkg = k2||ka||kb
lkg = k2||ka||kb||..||ki
lkg > β


$m_1, m_2, \ldots, m_i$
Enc(k, m)
$c_1, c_2, \ldots, c_i$
lkg $= k_2$
m_0, m_1
Enc(k, m_b)
WBInc
lkg $= k_2$
lkg $= k_2 || k_a$
lkg $< \beta$
lkg $= k_2 || k_a || k_b$
lkg $= k_2 || k_a || k_b || \ldots || k_i$
lkg $> \beta$


Enc(k, m)
m1, m2, . . , mi
c1, c2, . . , ci
lkg = k2
WBInc
m0, m1
Enc(k, mb)
cb
lkg = k2
lkg = k2||ka
lkg < β
lkg = k2||ka||kb
lkg = k2||ka||kb||..||ki
lkg > β



lkg $= k_2 || k_a$

lkg $< \beta$

lkg $= k_2 || k_a || k_b$

lkg $= k_2 || k_a || k_b || \ldots || k_i$

lkg $> \beta$

No reduction, no information theoretic argument :(

The problem

- Wichs [5] explained why proving security in leakage models is difficult
- We apply the same *meta-reduction* argument to the strong incompressibility model, where we:
  - Build a pair of inefficient adversaries who determine the value of the secret key via oracle queries
  - Build a PPT simulation which models the adversaries, but with a shared state
  - Show that in the eyes of a reduction, the simulation is indistinguishable from the pair of adversaries

[5] Wichs: Barriers in Cryptography with Weak, Correlated and Leaky Sources, ITCS 2013

The problem

- Wichs [5] explained why proving security in leakage models is difficult
- We apply the same *meta-reduction* argument to the strong incompressibility model, where we:
  - Build a pair of inefficient adversaries who determine the value of the secret key via oracle queries
  - Build a PPT simulation which models the adversaries, but with a shared state
  - Show that in the eyes of a reduction, the simulation is indistinguishable from the pair of adversaries

——> Catch: the inefficient adversaries determine the key, as long as the scheme allows for this

[5] Wichs: Barriers in Cryptography with Weak, Correlated and Leaky Sources, ITCS 2013

The problem

——> Catch: the inefficient adversaries determine the key, as long as the scheme allows for this

- The meta reduction works as long as the scheme is *key-fixing:*
  - An adversary is able to determine the value of a secret key, given a set of input/output pairs of the scheme
  - Hazay et al. [6] show how to build a non-key fixing scheme - and they prove security under leakage

[6] Hazay et al.: Leakage-resilient cryptography from minimal assumptions, Eurocrypt 2013

The problem

——> Catch: the inefficient adversaries determine the key, as long as the scheme allows for this

- The meta reduction works as long as the scheme is *key-fixing:*
  - An adversary is able to determine the value of a secret key, given a set of input/output pairs of the scheme
  - Hazay et al. [6] show how to build a non-key fixing scheme - and they prove security under leakage
  - However: the construction is not very practical, as it includes huge sections of the key which are never used

[6] Hazay et al.: Leakage-resilient cryptography from minimal assumptions, Eurocrypt 2013

AES in CBC mode

- In the paper, we show that conventional ciphers such as AES are actually key-fixing
  - We prove this by modelling the cipher as a truly random permutation
  - Thus, we cannot have provable secure constructions of incompressible AES
- However: it may be worth exploring candidate designs, such as the one discussed in this presentation, and analyse it from a cryptanalysis perspective

Conclusions

- It is a challenge to prove white-box security in the strong incompressibility model, given large amounts of leakage, if the incompressible program is key-fixing
- We show that AES is key-fixing
  - —> we cannot expect to build a provable secure (strong) incompressible white-box program based on AES in the standard model
- However there exist nice construction ideas which may be worth exploring from a cryptanalysis point of view and which may give us what we need in practice

Děkuji!

Estuardo Alpirez Bock, Chris Brzuska and Russell W. F. Lai