Limits on the Power of Garbling Techniques for Public-Key Encryption

Sanjam Garg (UC Berkeley)

Mohammad Hajiabadi (UC Berkeley, Univ. of Virginia)

Mohammad Mahmoody (Univ. of Virginia)

Ameer Mohammed (Univ. of Virginia → Kuwait University)

Long-Standing Open Problem

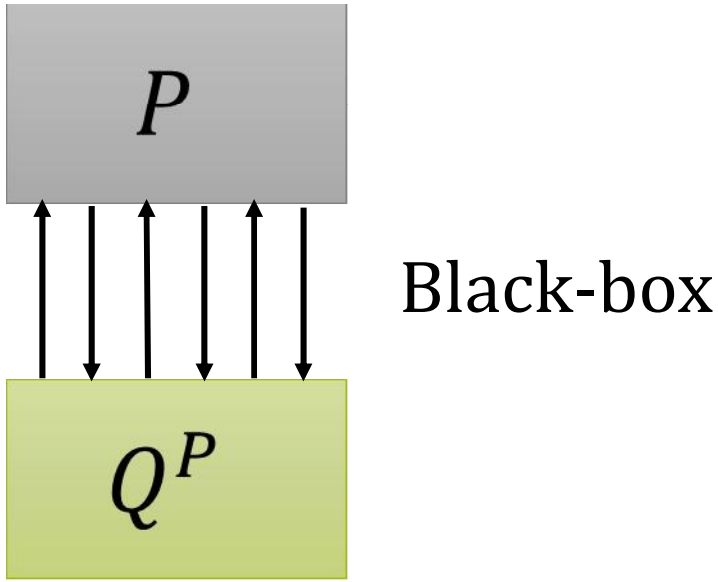
One-Way Functions $\overset{?}{\Rightarrow}$ Public-Key Encryption

Not in a **black-box** way [IR89]

What about **non-black-box** methods?

Black-Box Constructions [IR89, RTV04, BBF13]

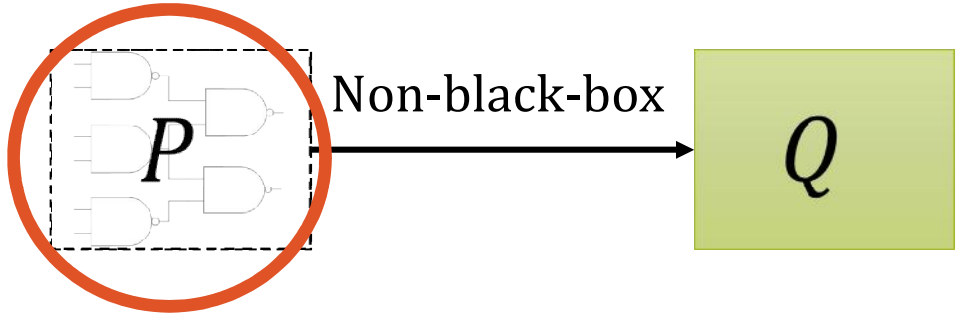
A **black-box construction** of Q from P:



Security: Adv_Q breaking $Q \Longrightarrow \mathrm{Adv}_P$ breaking P

Common Non-Black-Box Techniques

- "Low-Tech" (OWF-realizable):
  - Garbling [Yao86]
  - Zero-knowledge proofs [GMR85]
  - Witness Indistinguishability/Hiding [FFS87, FS90]



- "High-Tech" (based on stronger assumptions):
  - Fully Homomorphic Encryption [Gentry09]
  - Functional Encryption [O'Neill11, BSW11]
  - Witness Encryption [GGSW13]
  - Indistinguishability Obfuscation [BGI+02, GGHRSW13]
  - …..

Garbling Scheme

Decomposable/projective [BHR12]

Security: $\text{Sim}\left(1^{|C|}, 1^{|x|}, C(x)\right) \equiv_c \left(\tilde{C}, \{w_i^{x_i}\}_{i \in n}\right)$

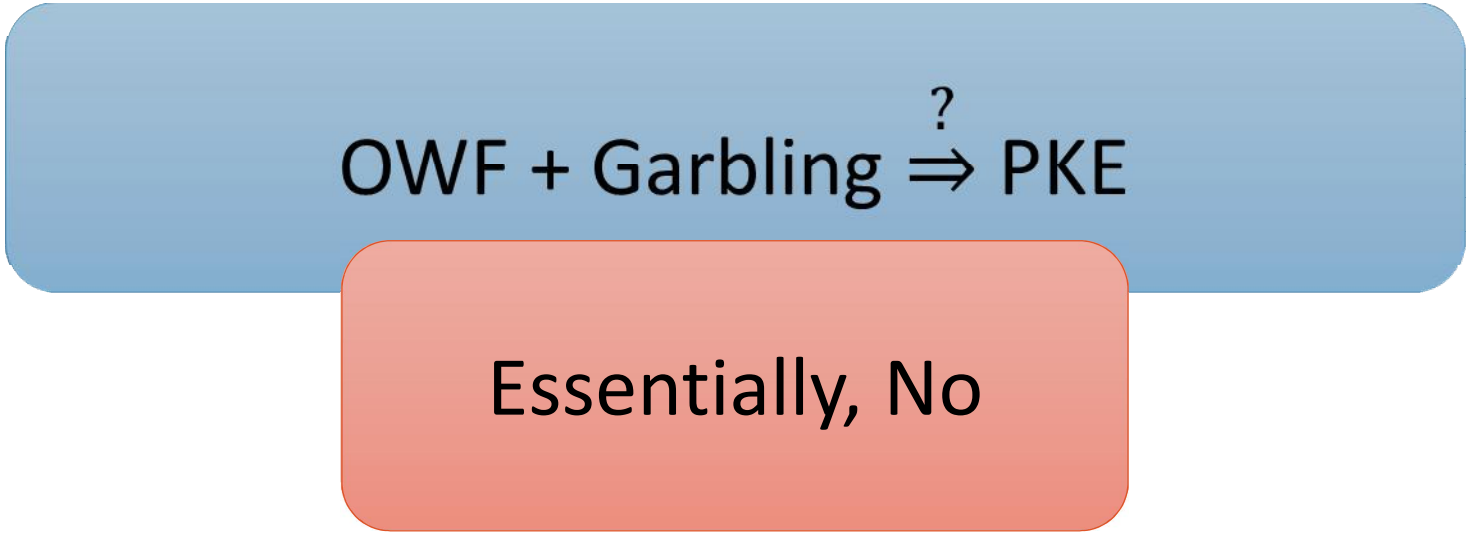
Our Main Question

$$\text{OWF} + \text{Garbling} \overset{?}{\Rightarrow} \text{PKE}$$

- OWF $\Rightarrow$ Garbling [Yao86]

- CDH + Garbling $\Rightarrow$ IBE [DG17] circumventing CDH $\not\Rightarrow$ IBE [PRV12]
  Non-BB BB

Our Main Result (Informal)



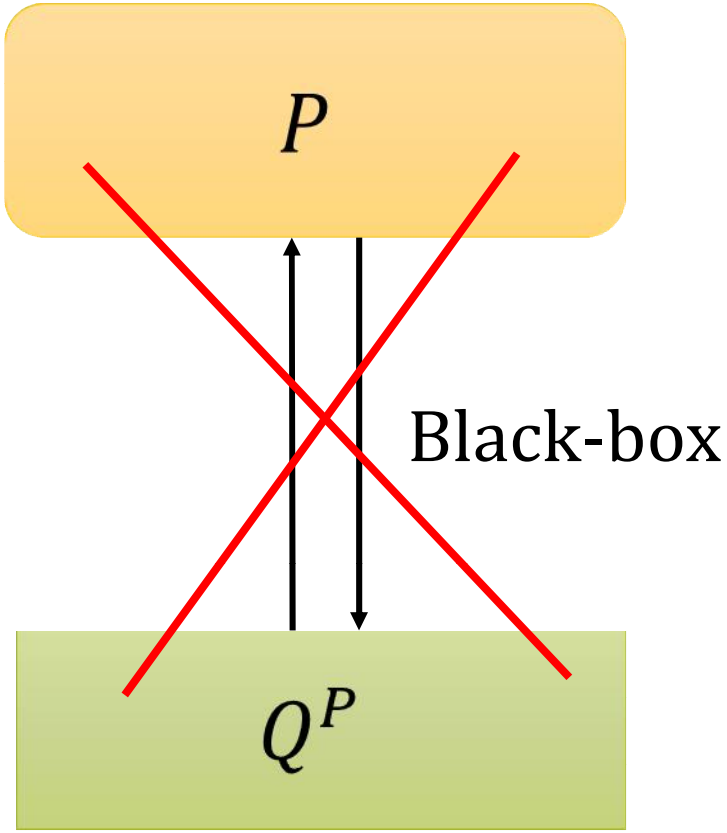
Same model as:

[Brakerski-Katz-Segev-Yerukhimovich'11] and [Asharov-Segev'15]

Outline

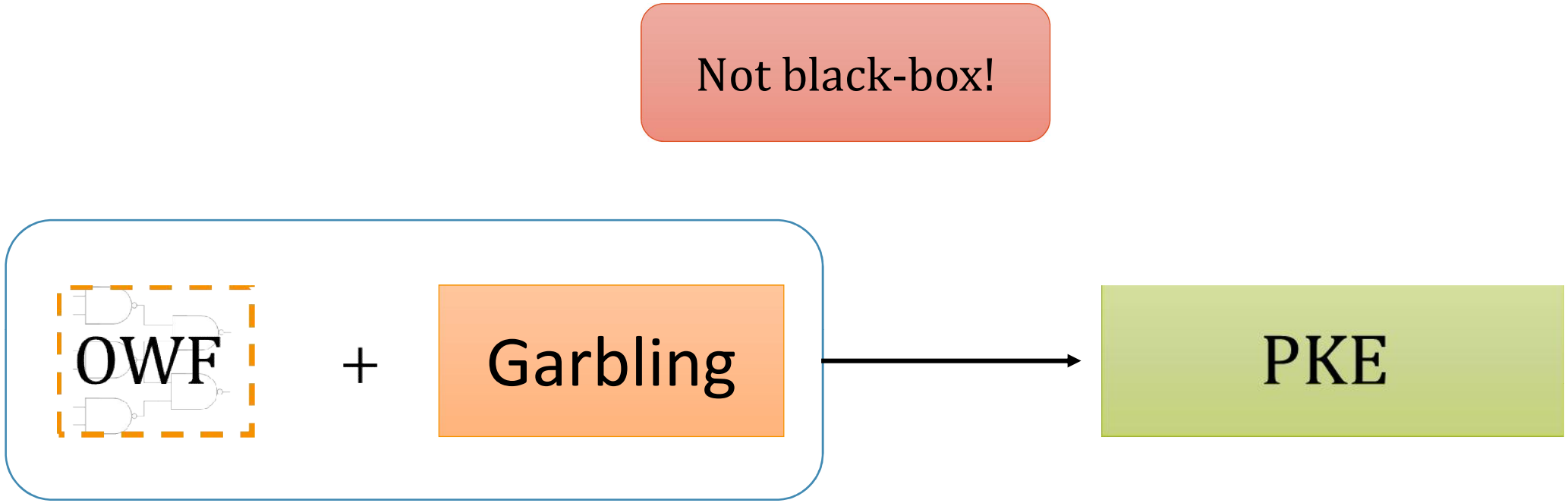
- Problem and Motivation
- Separation Model
- Ideas behind the proof

Black-box Separations: Most Separations in Crypto



$$P \nRightarrow Q$$
BB

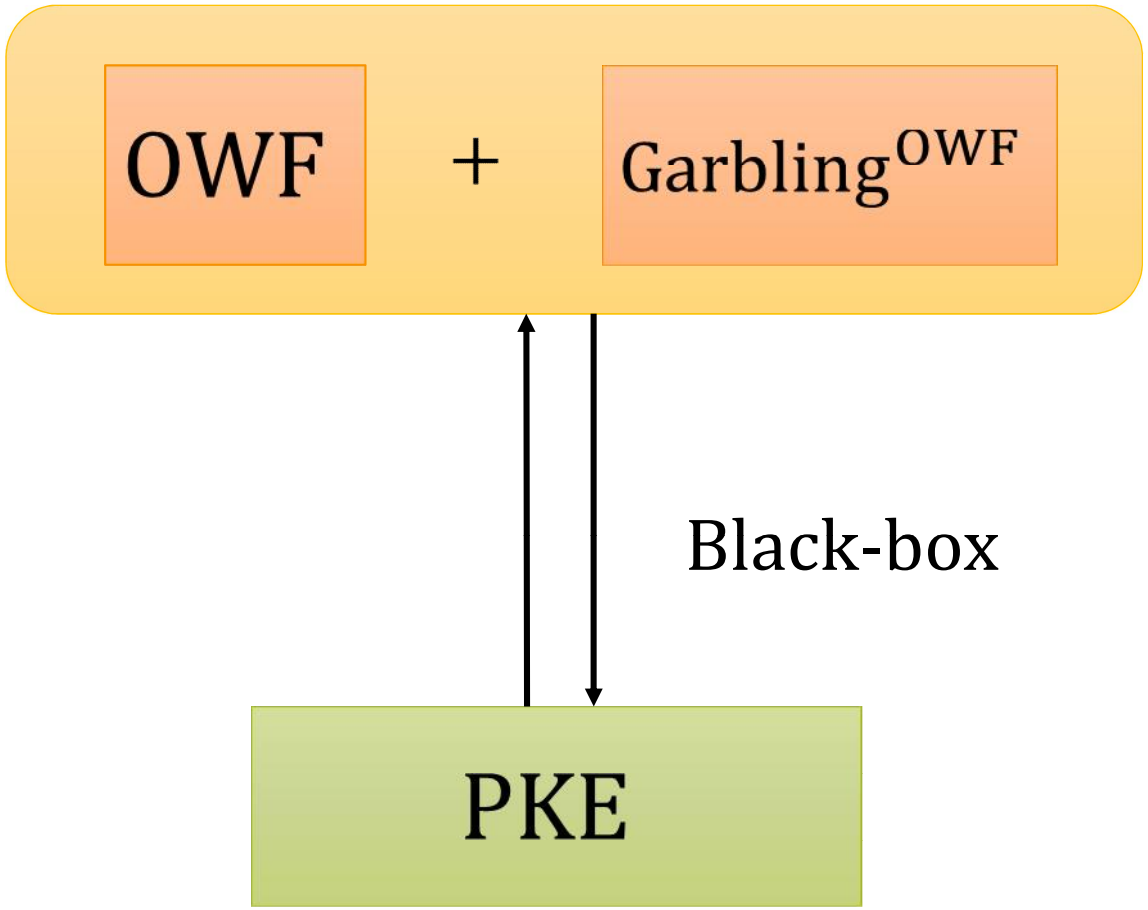
Constructions We Want to Rule Out



How Does Garbling Make Constructions Non-BB?

Treating OWF + Garbling as a Black-Box

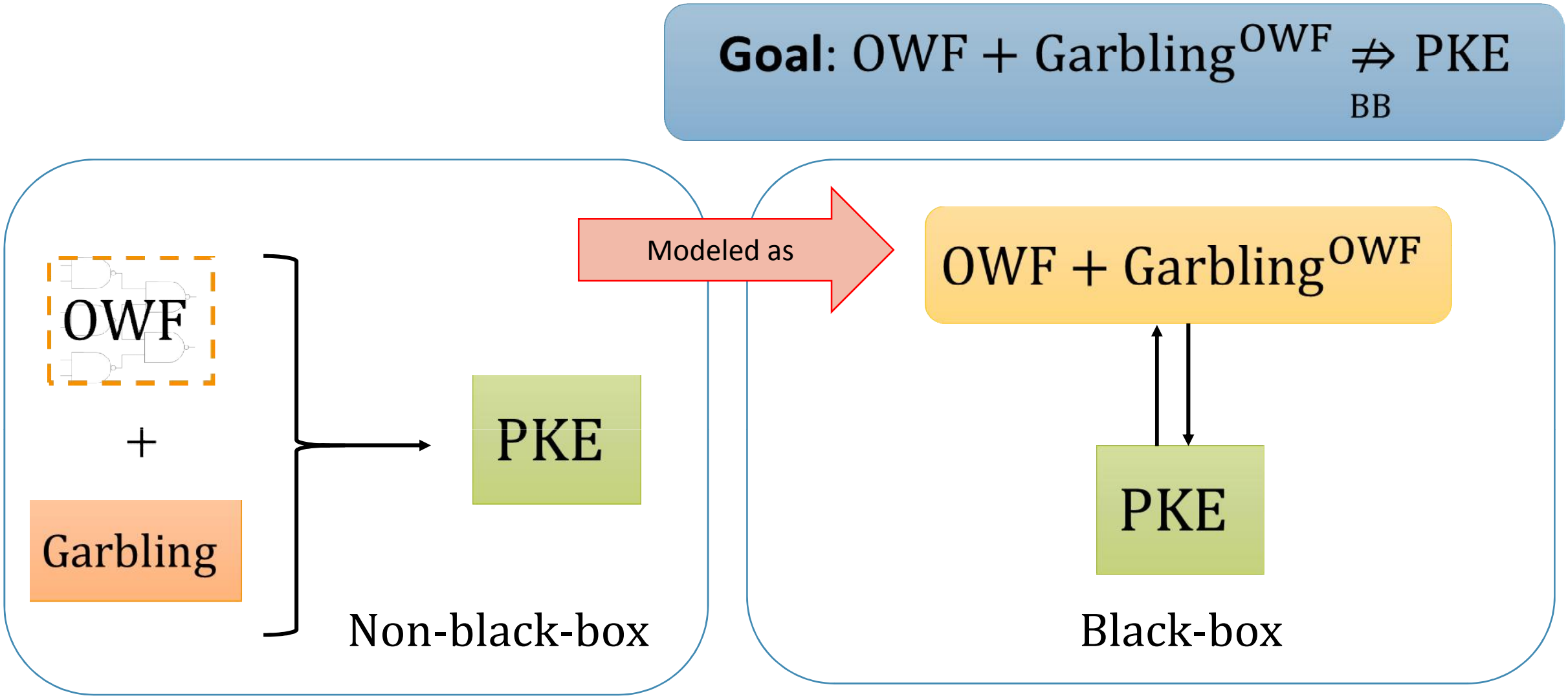
Treating OWF + Garbling as a Black-Box



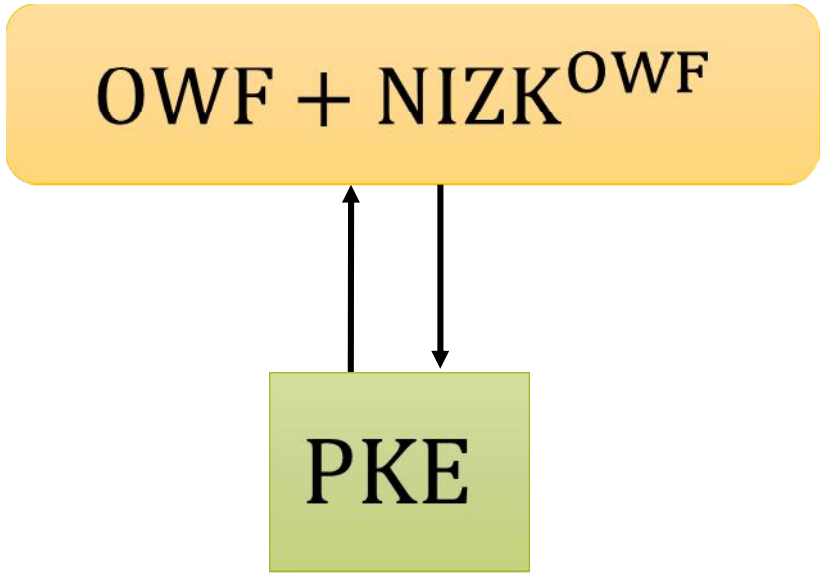
Main Theorem (Formal)

There exists no **black-box** construction of PKE from $(\mathrm{OWF}, \mathrm{Garbling}^{\mathrm{OWF}})$

Big Picture of our Approach

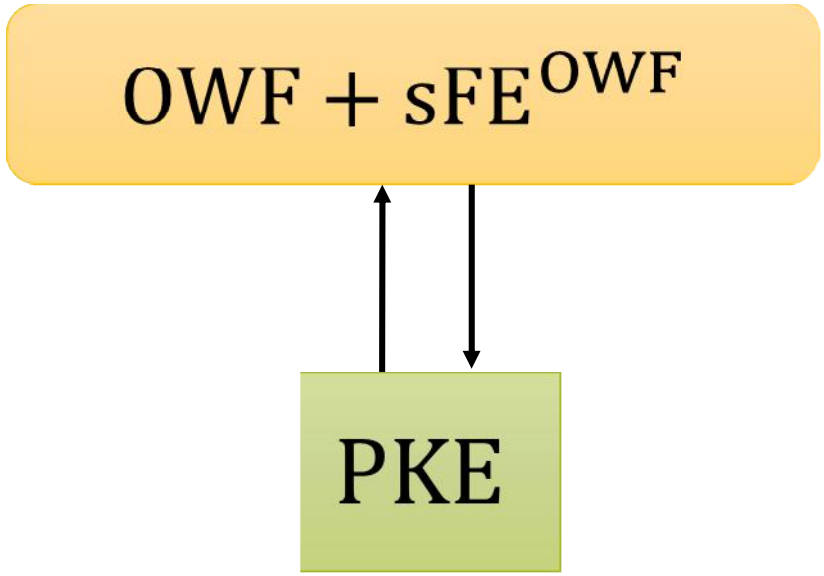


Previous Separation Results That Use This Model



$\mathrm{OWF} + \mathrm{NIZK}^{\mathrm{OWF}} \not\Rightarrow \mathrm{PKE}$
[BKSY11]

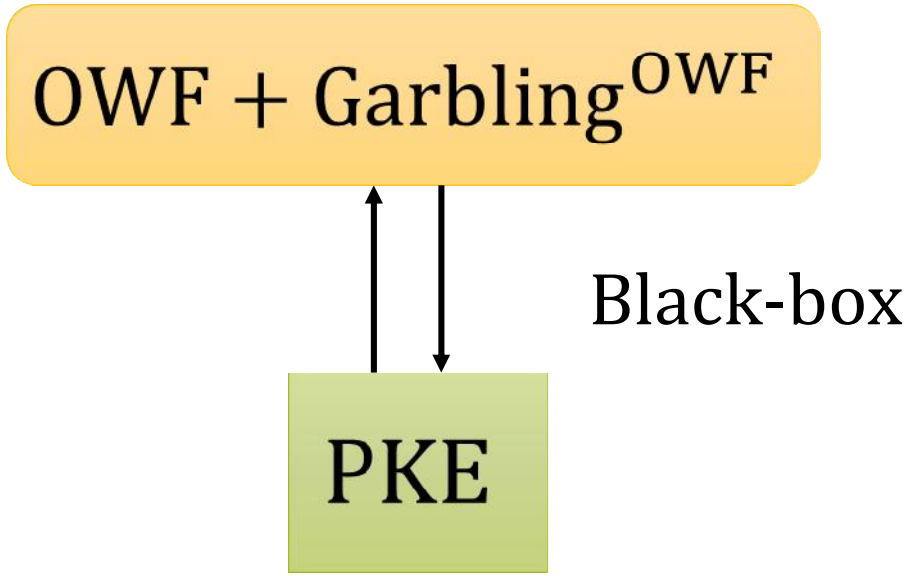
[BKSY11]: Perfectly complete PKE
vs. this work: Imperfectly complete PKE



$\mathrm{OWF} + \mathrm{sFE}^{\mathrm{OWF}} \not\Rightarrow \mathrm{PKE}$
[AS15]

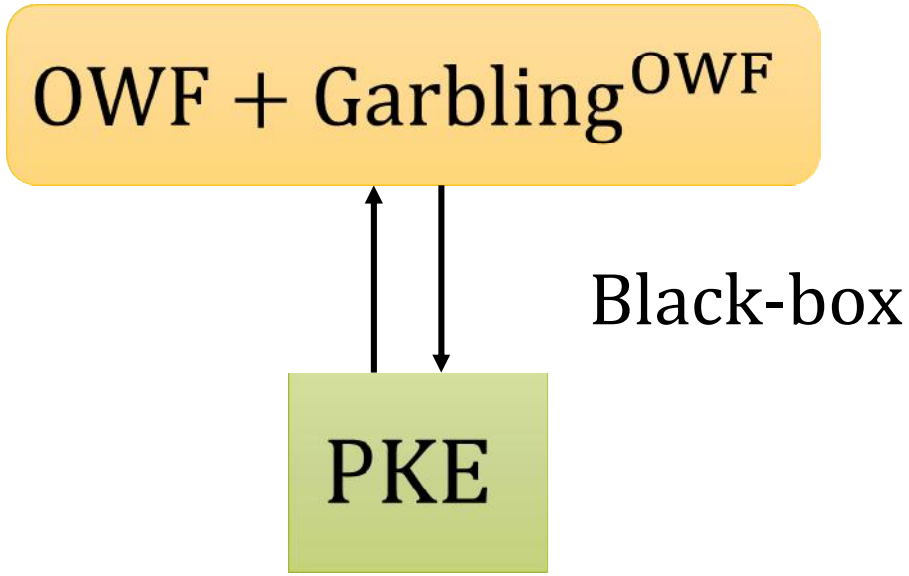
[AS15]: Non-projective but reusable
vs. this work: Projective Garbling

Constructions Captured in This Model



- Garbling circuits with OWF gates
  - [Beaver96, LO13, GLOS15]

Constructions NOT Captured by This Model

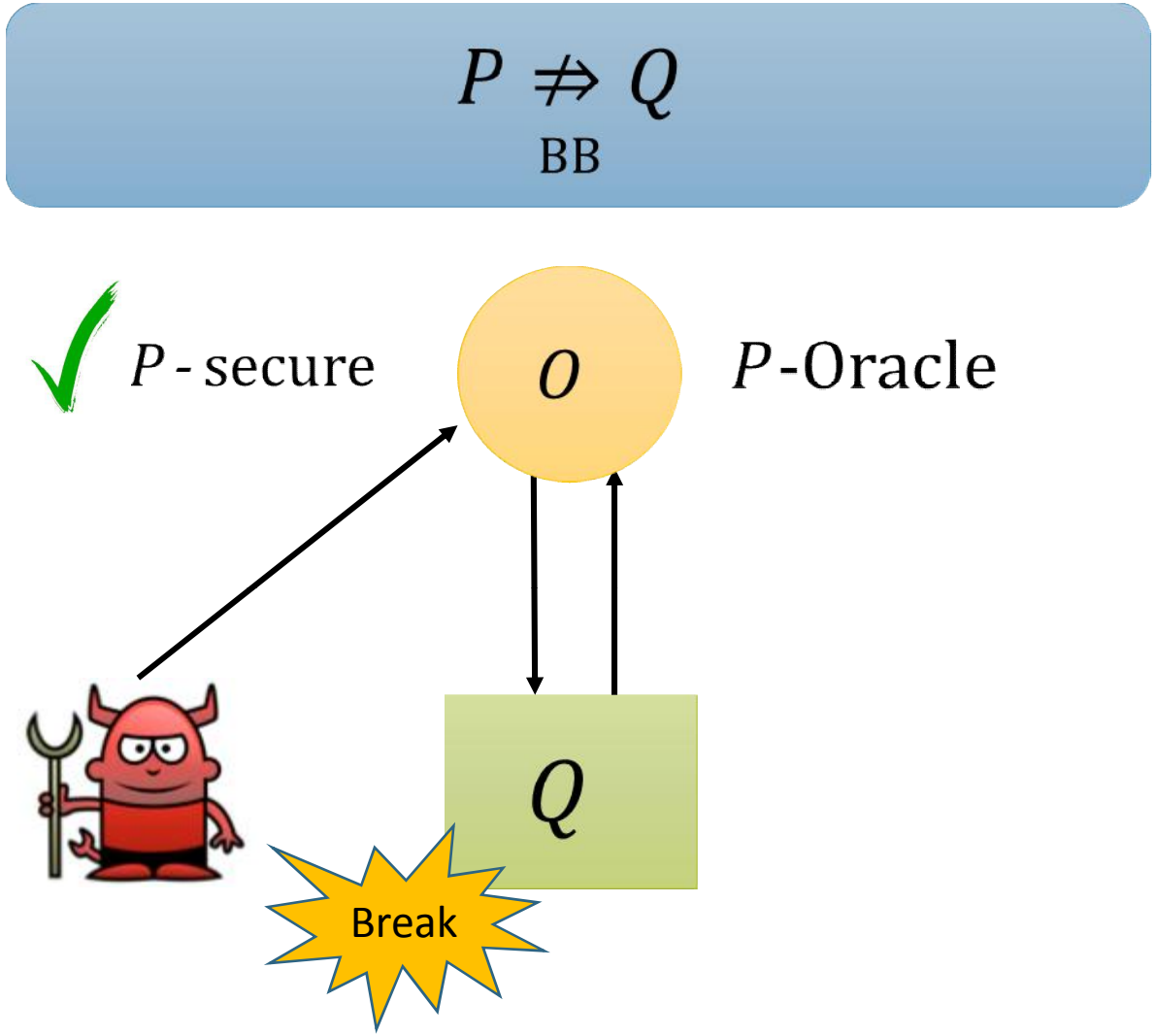


- Garbling circuits with Garb/Eval gates
  - Falls under the monolithic framework of [Garg-Mahmoody-M17]

Outline

- Problem and Motivation
- Separation Model
- Ideas behind the proof

How to Prove Black-Box Separations



The $\mathbf{OWF} + \mathbf{Garbling}^{\mathbf{OWF}}$ Oracle O (first attempt)

The oracle O consists of:

Oracle O

$$f, \mathrm{Garb}^f, \mathrm{Eval}^f$$

1. To realize OWF: Random oracle

$$f: \{0,1\}^n \xrightarrow{\$} \{0,1\}^n$$

2. To realize $\mathrm{Garbling}^{\mathrm{OWF}}$: "Ideal" Garbling Scheme for circuits with f-gates

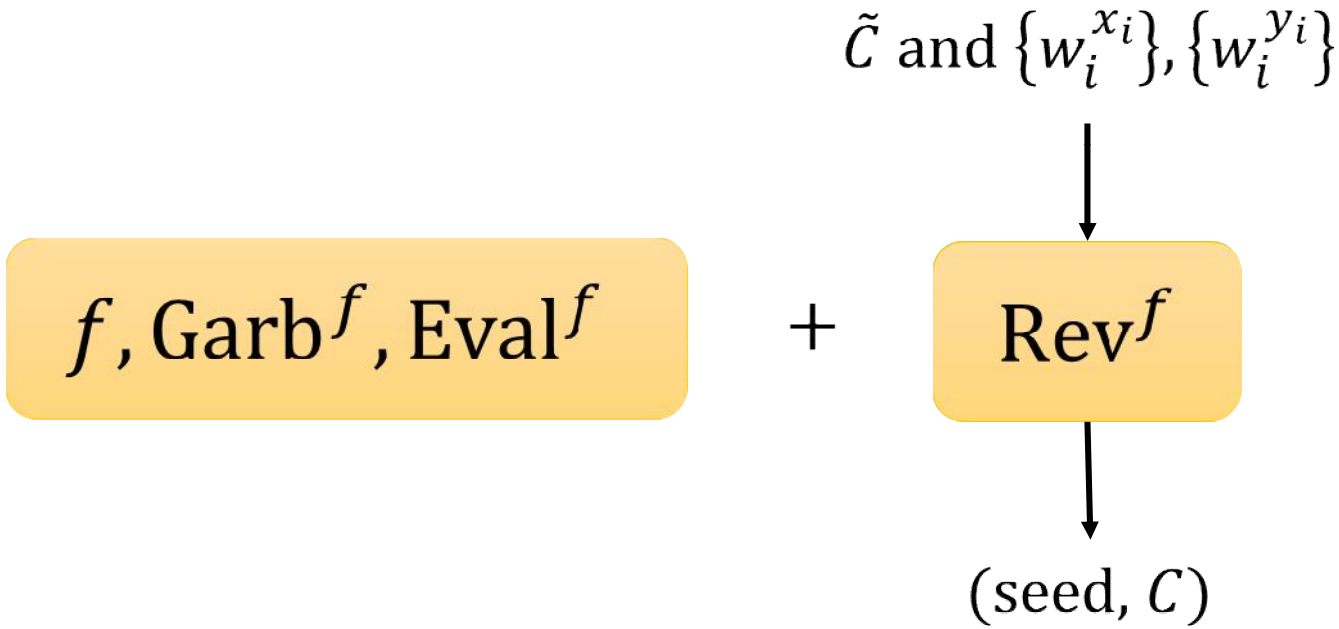
$$(\mathrm{Garb}^f, \mathrm{Eval}^f)$$

Where $\mathrm{Garb}(\mathrm{seed}, C^f) \xrightarrow{\$} \left(\tilde{C}, \{w_i^0, w_i^1\}_{i \in n}\right)$

And $\mathrm{Eval}^f\left(\tilde{C}, \left(w_1^{x_1}, \ldots, w_n^{x_n}\right)\right)$ outputs $C(x)$

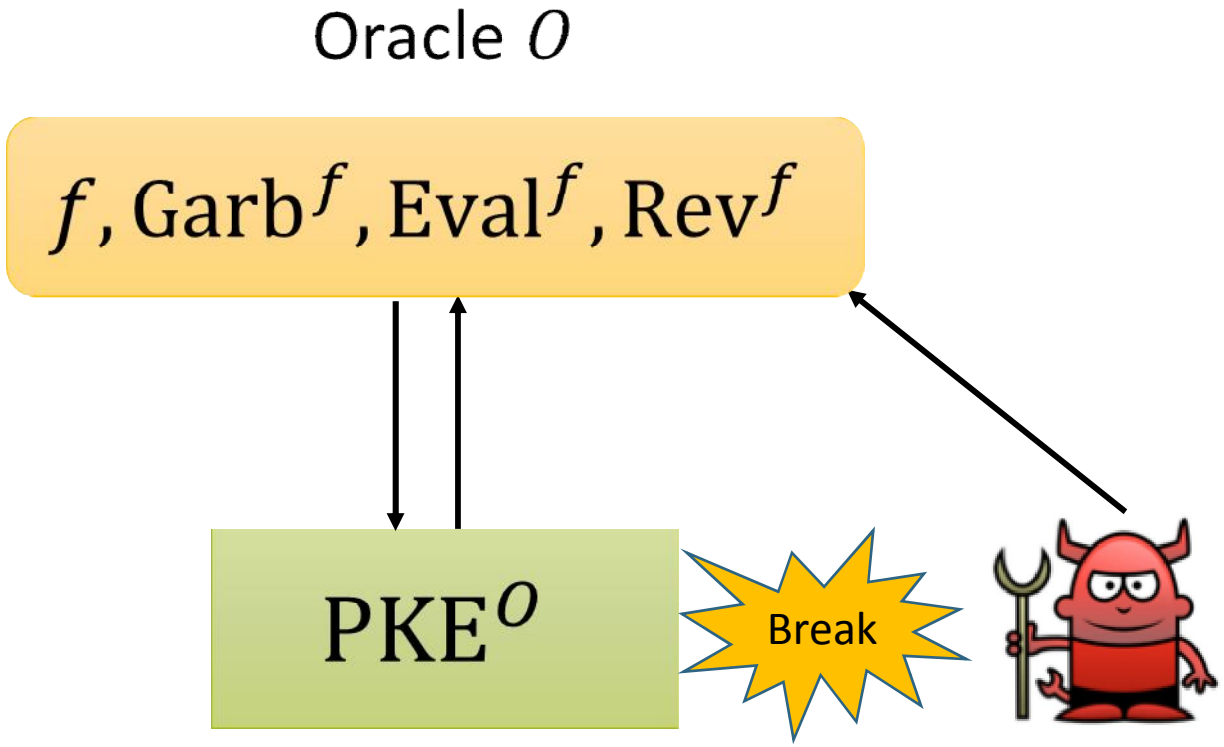
Problem: O is too strong!
We can realize VBB obf. using it

The $\mathbf{OWF} + \mathbf{Garbling}^{\mathbf{OWF}}$ Oracle O (right version)



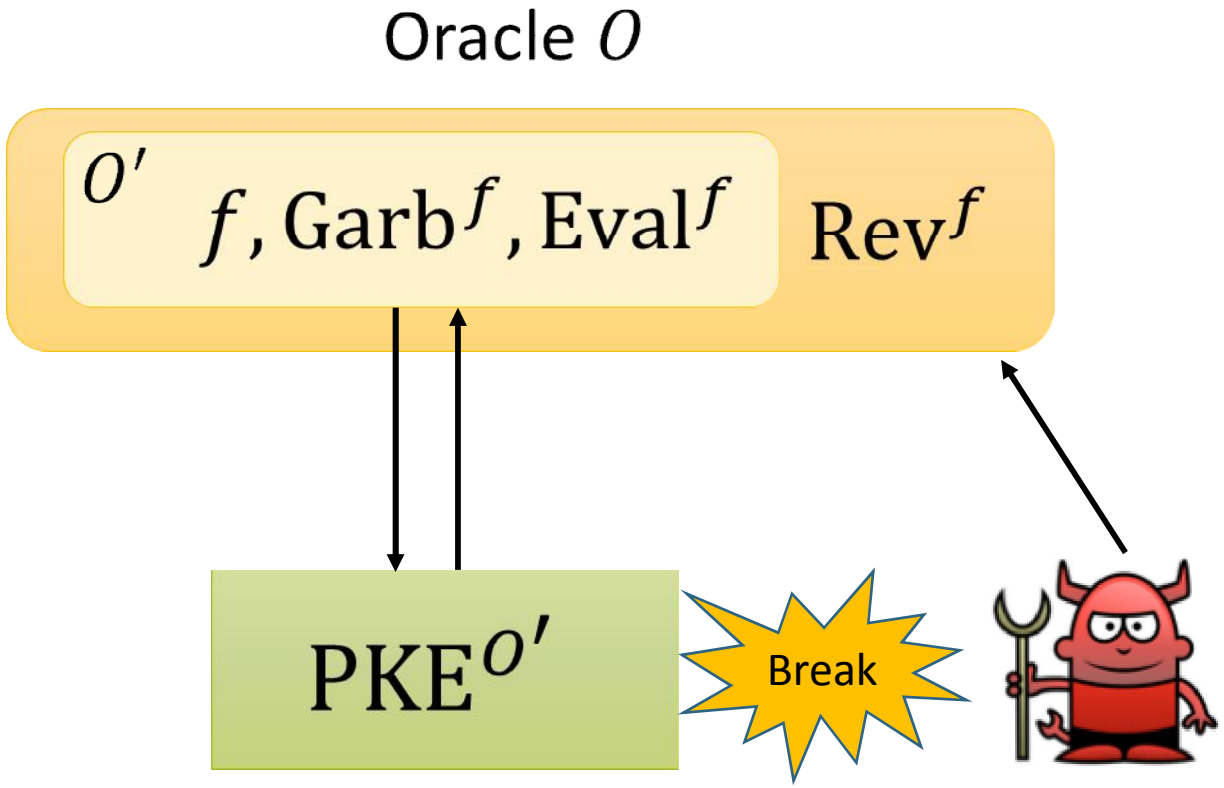
Oracle O $\quad f, \mathrm{Garb}^f, \mathrm{Eval}^f, \mathrm{Rev}^f$

Our Goal



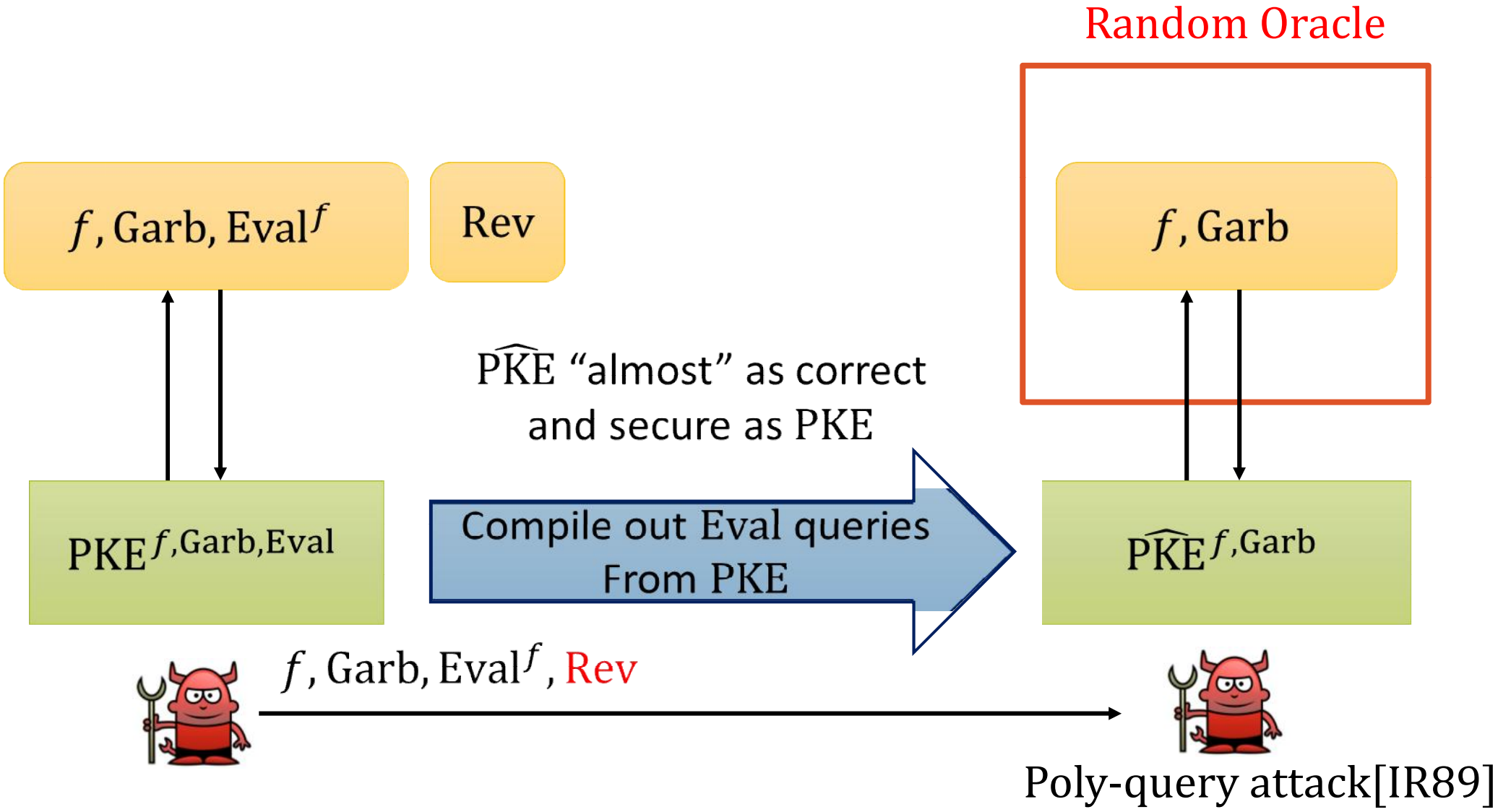
- Any PKE can be broken with only poly-queries to O

Our Goal

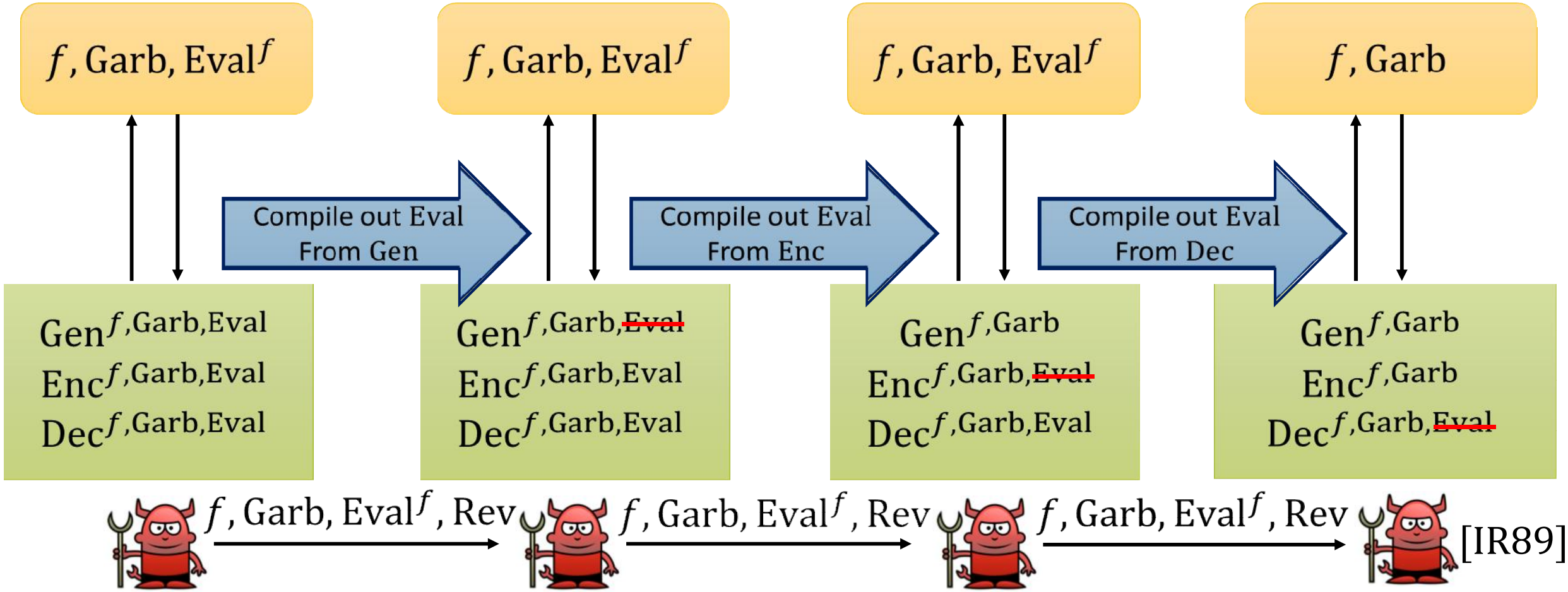


- Any PKE can be broken with only poly-queries to O
- Similar to techniques used in [GMR01,HR04]

Our Approach (at a high level)



Our Approach (closer look)



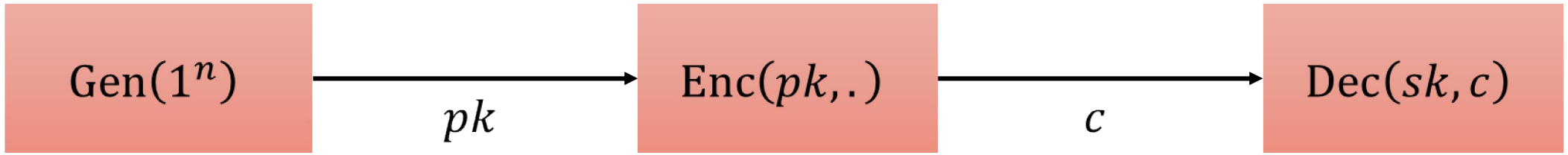
Compiling out Eval queries from Gen

Gen knows C_1

$\mathrm{Garb}(C_1) = \tilde{C}_1$

~~$\mathrm{Eval}^f(\tilde{C}_1, .)$~~

Key property of Garb: $|\tilde{C}| \gg |C|$
So hard to find any $\tilde{C}$ without calling Garb



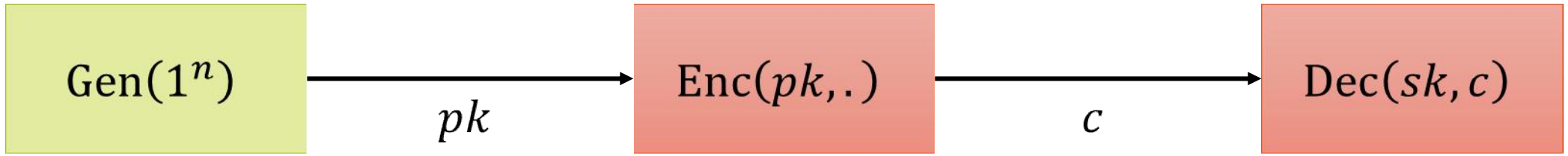
Compiling out Eval queries from Enc

Challenge: Enc does not know C_1

Idea: Let Gen help Enc

$\mathrm{Garb}(C_1) = \tilde{C}_1$

$\mathrm{Garb}(C_2) = \tilde{C}_2$
$\mathrm{Eval}^f(\tilde{C}_1, .)$



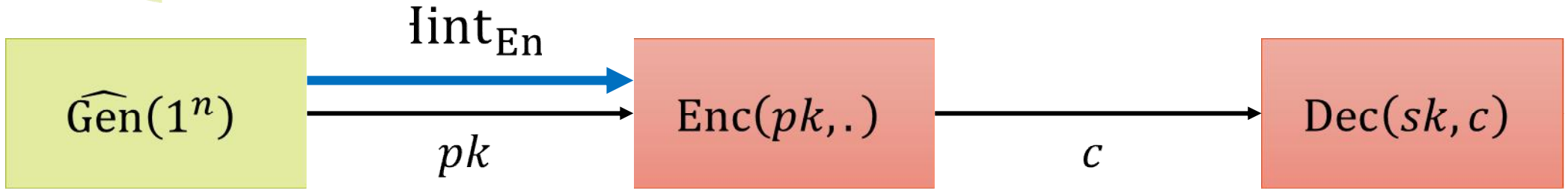
Compiling out Eval queries from Enc

How about
$\text{Hint}_{\text{Enc}} = (\text{Garb}(C_1), \tilde{C}_1)$?

Breaks Security

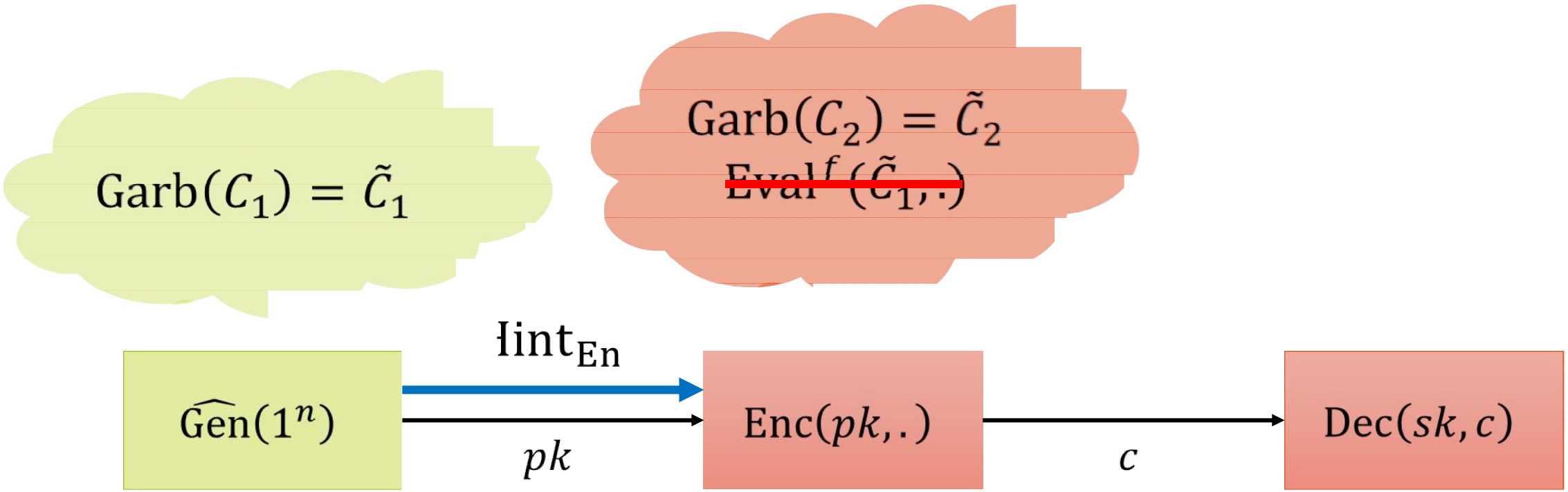
$\text{Garb}(C_1) = \tilde{C}_1$

$\text{Garb}(C_2) = \tilde{C}_2$
$\text{Eval}^f(\tilde{C}_1, .)$



Compiling out Eval queries from Enc

Solution: $\widehat{\mathrm{Gen}}$ runs $\mathrm{Enc}(pk, .)$ "many times" then adds answers of Eval queries to $\mathrm{Hint}_{\mathrm{Enc}}$



Compiling out Eval queries from Dec

Challenge: Dec does not know C_2

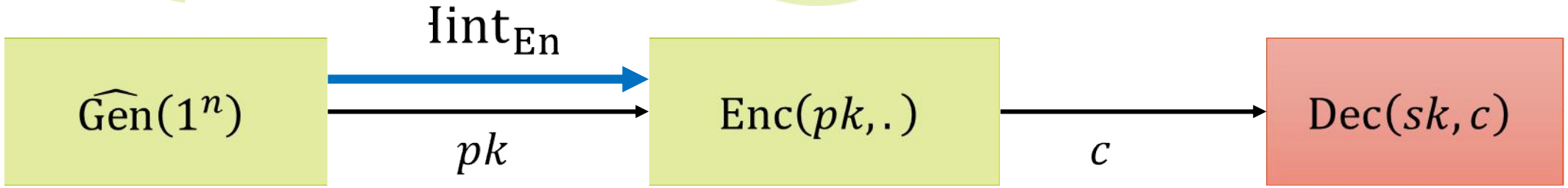
Idea: Let Enc help Dec

$\text{Garb}(C_1) = \tilde{C}_1$

$\text{Garb}(C_2) = \tilde{C}_2$

$\text{Garb}(C_3) = \tilde{C}_3$

~~$\text{Eval}^f(\tilde{C}_2, .)$~~



Compiling out Eval queries from Dec

Enc does not know sk!

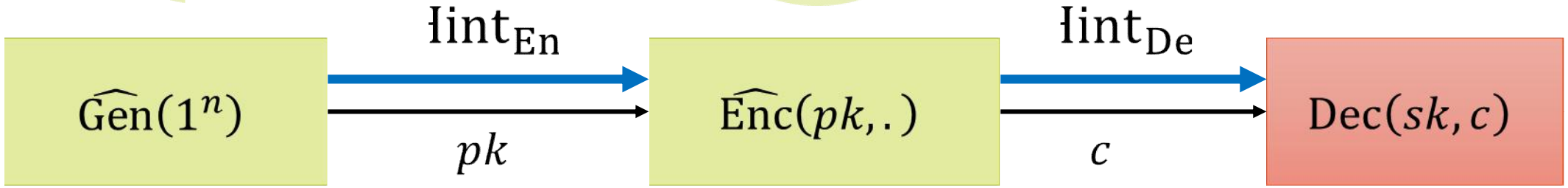
How about $\widehat{\mathrm{Enc}}$ runs $\mathrm{Dec}(sk, c)$ "many times" then adds answers of Eval queries to $\mathrm{Hint}_{\mathrm{Dec}}$?

$\mathrm{Garb}(C_1) = \tilde{C}_1$

$\mathrm{Garb}(C_2) = \tilde{C}_2$

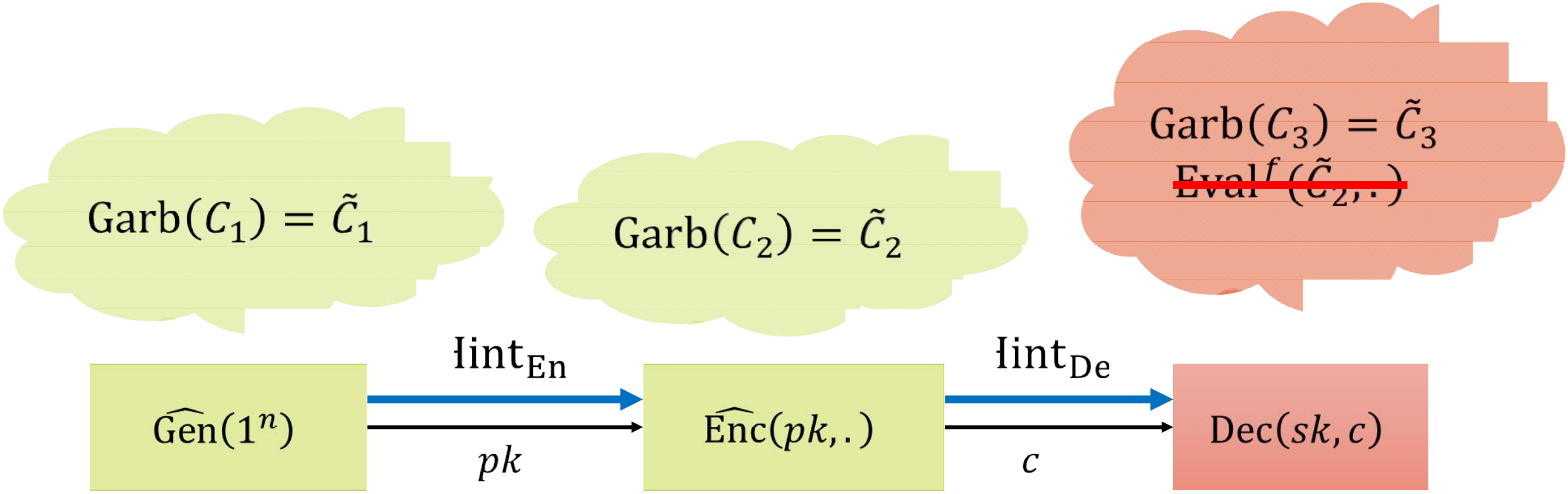
$\mathrm{Garb}(C_3) = \tilde{C}_3$

~~$\mathrm{Eval}^f(\tilde{C}_2, .)$~~



Compiling out Eval queries from Dec

Solution: $\widehat{\mathrm{Enc}}$ runs $\mathrm{Dec}(\widehat{sk},.)$ "many times" then adds answers of Eval queries to $\mathrm{Hint}_{\mathrm{Dec}}$



Summary

- Main Result
  - OWF + garbling for circuits with OWF gates are insufficient for constructing PKE in a black-box way.

- Extensions in this work (not discussed in this talk):
  - OWF + garbling mechanisms for circuits with OWF gates are insufficient for constructing ***constant-round*** key-agreement protocols.
  - OWF + NIWI/NIZK for statements with OWF gates are insufficient for constructing PKE with ***without assuming perfect correctness*** (extending [BKSY11])

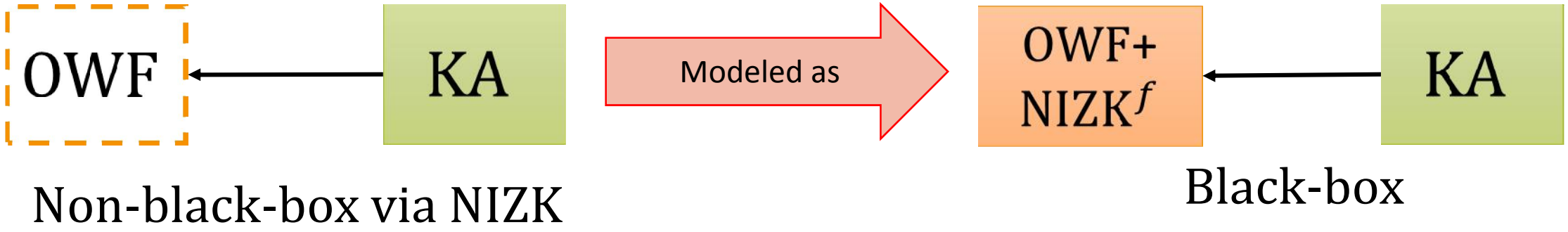
Open Problems

- Extension to ruling out PKE from if we allow garbling of the ***garbling scheme itself***.

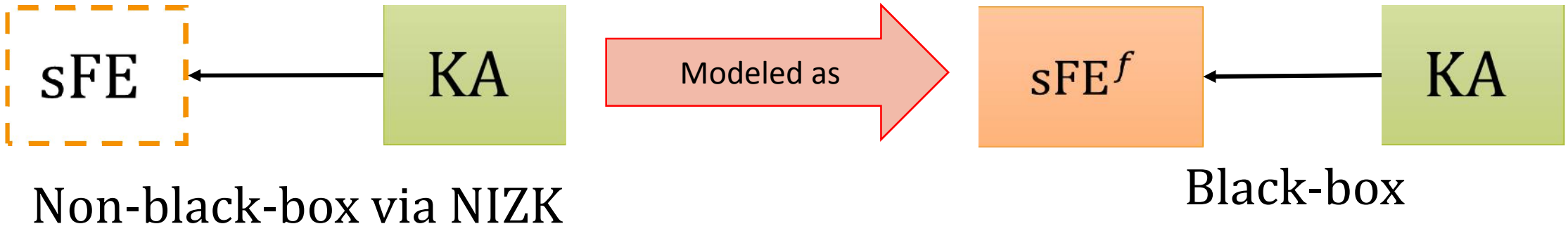
- Extension to ruling out key exchange with ***polynomial*** number of rounds from OWF + garbling.

Related results

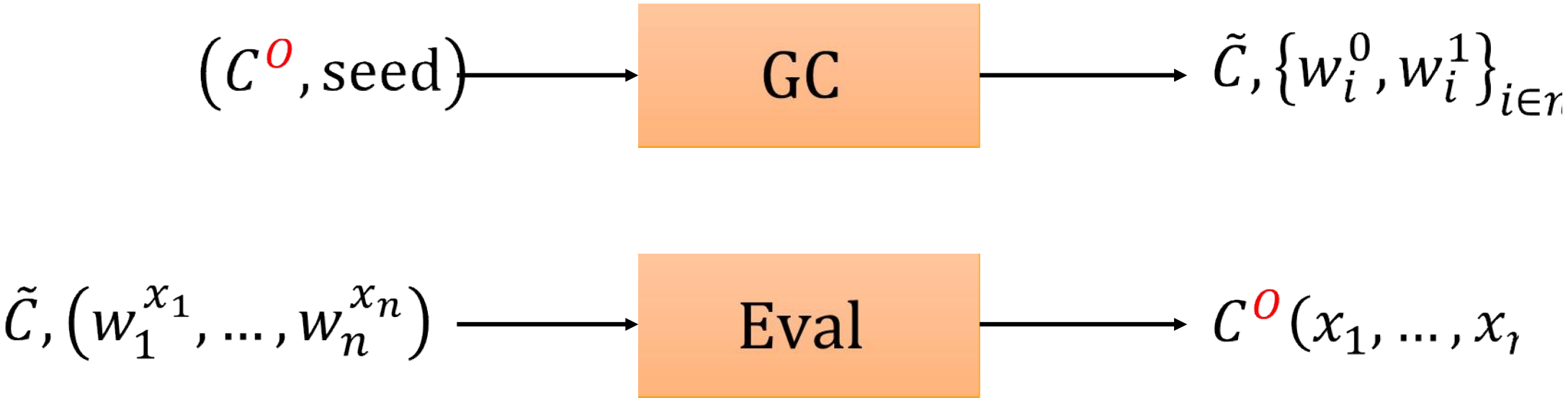
- [BKSY11]: OWF + NIZK/NIWI ⇏ (perfectly-complete) key agreement



- [AS15]: secret-key FE ⇏ key agreement



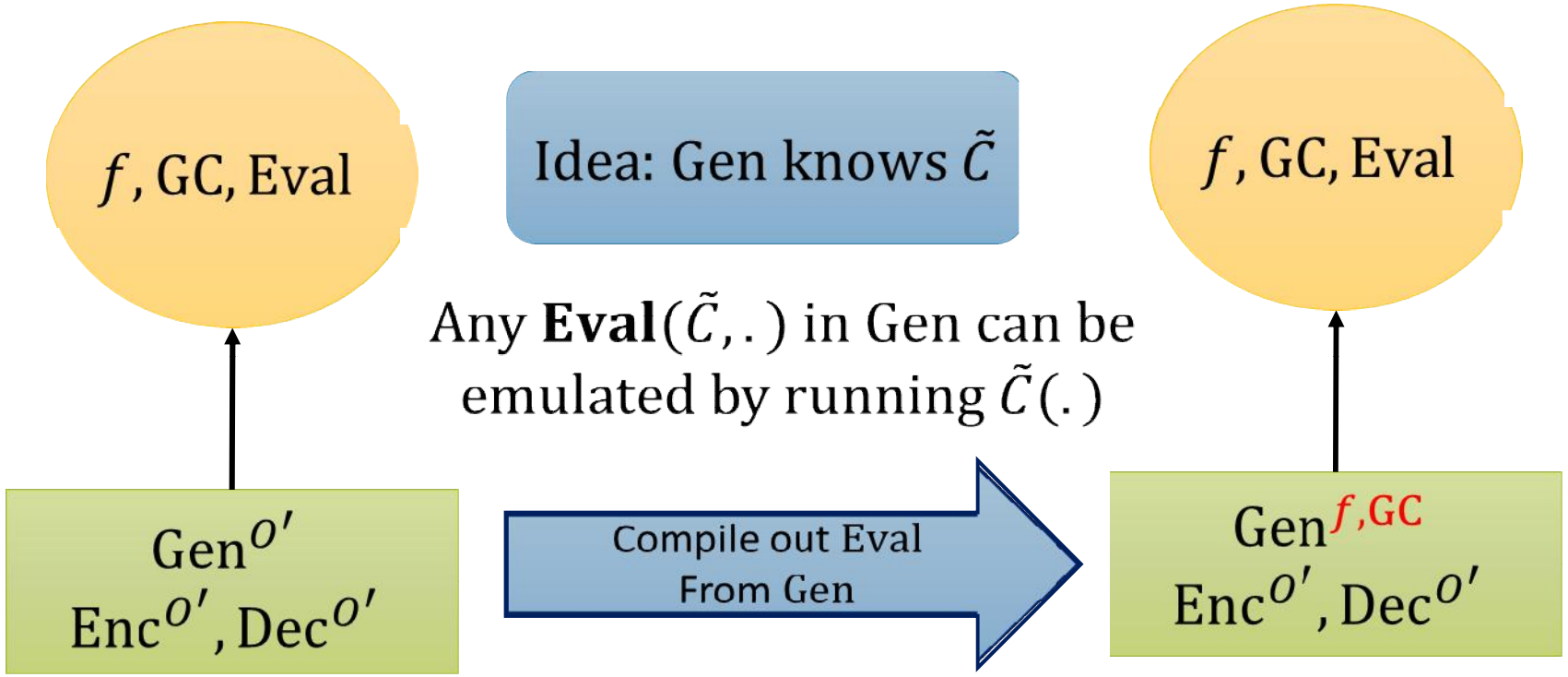
Garbling Scheme for oracle-aided circuits



Security: $\exists$ PPT Sim : $\text{Sim}\left(1^{|C|}, 1^{|x|}, C^O(x)\right) \equiv_c \left(\tilde{C}, \{w_i^0, w_i^1\}_{i \in n}\right)$

Compiling out Eval queries from Gen

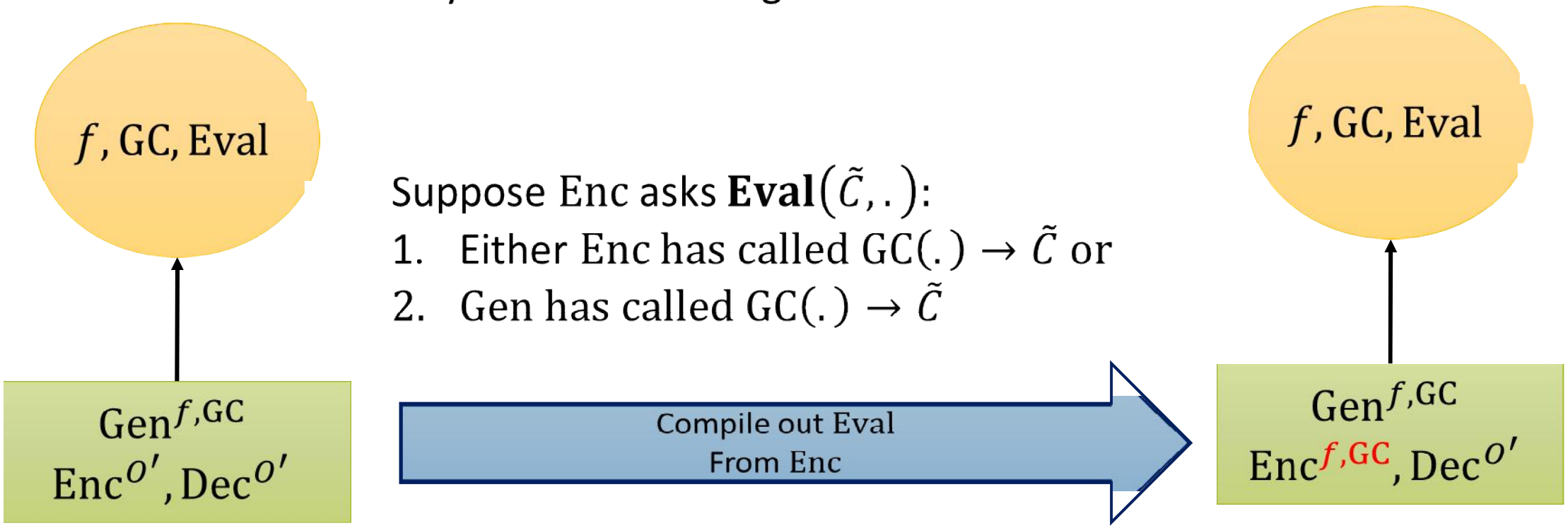
- Key property of GC used: $|\tilde{C}| \gg |C|$
  - So hard to find any $\tilde{C}$ without calling GC



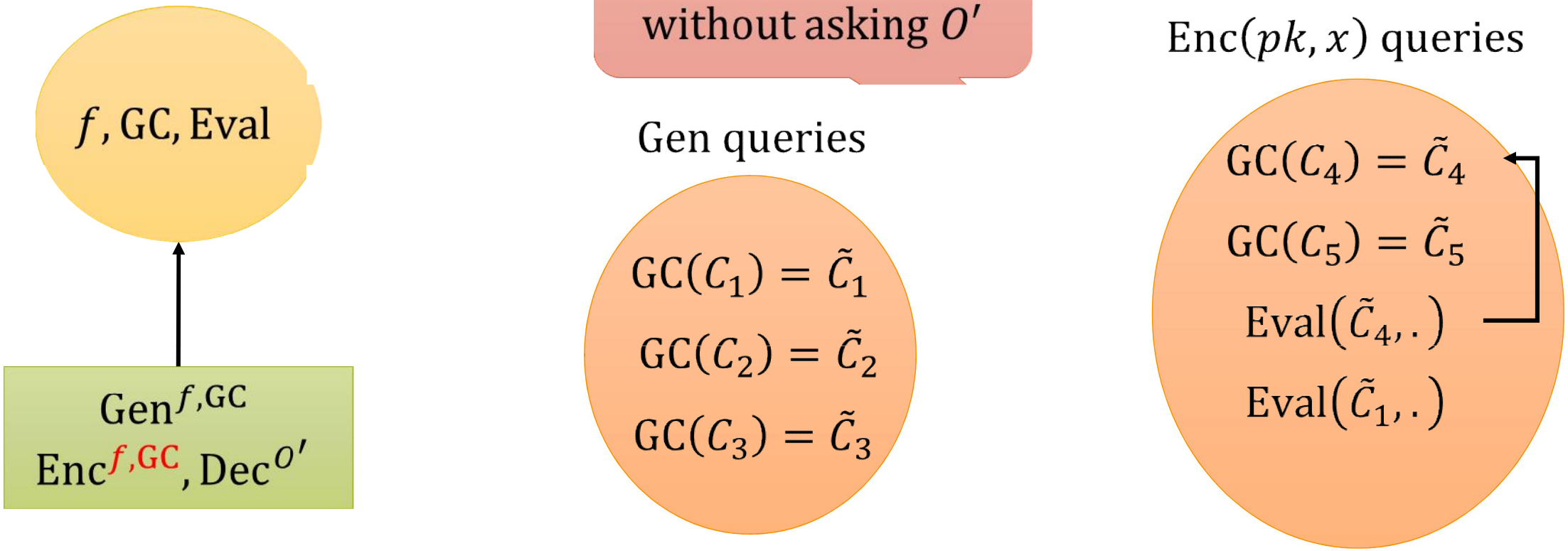
Compiling out Eval queries from Enc

- Key property of GC used: $|\tilde{C}| \gg |C|$
  - So hard to find any $\tilde{C}$ without calling GC

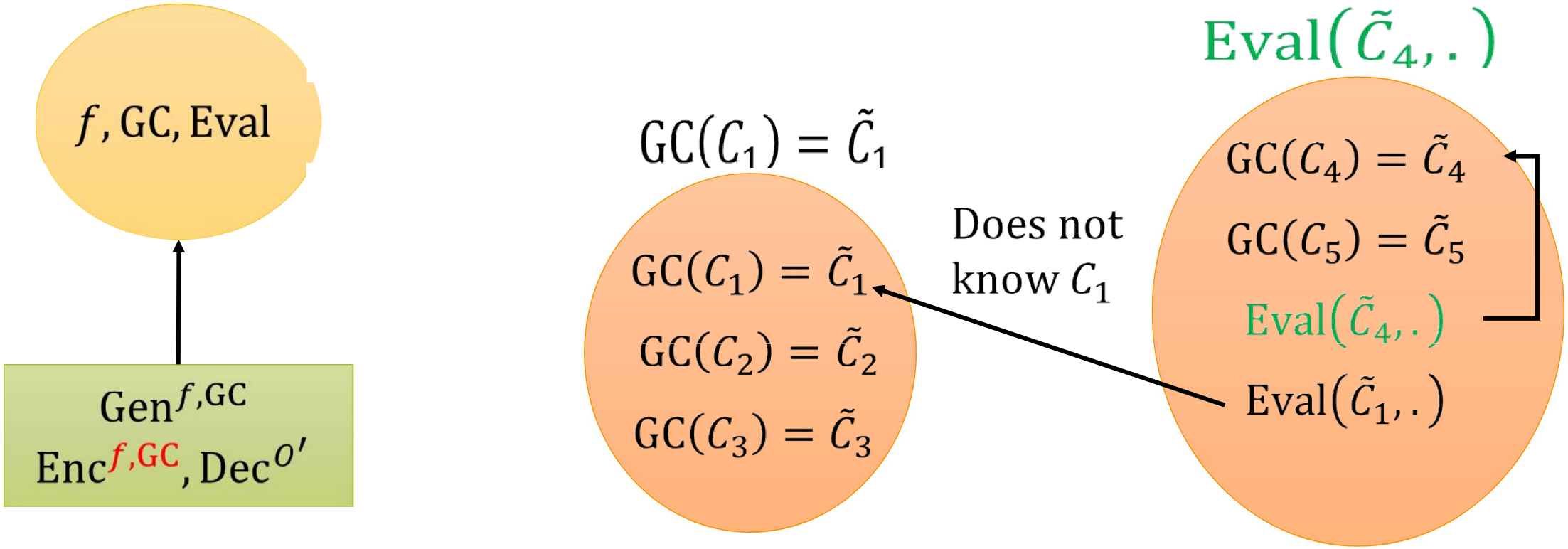
Problem for 2: Enc does not know $\tilde{C}$
Idea: Let Gen help Enc



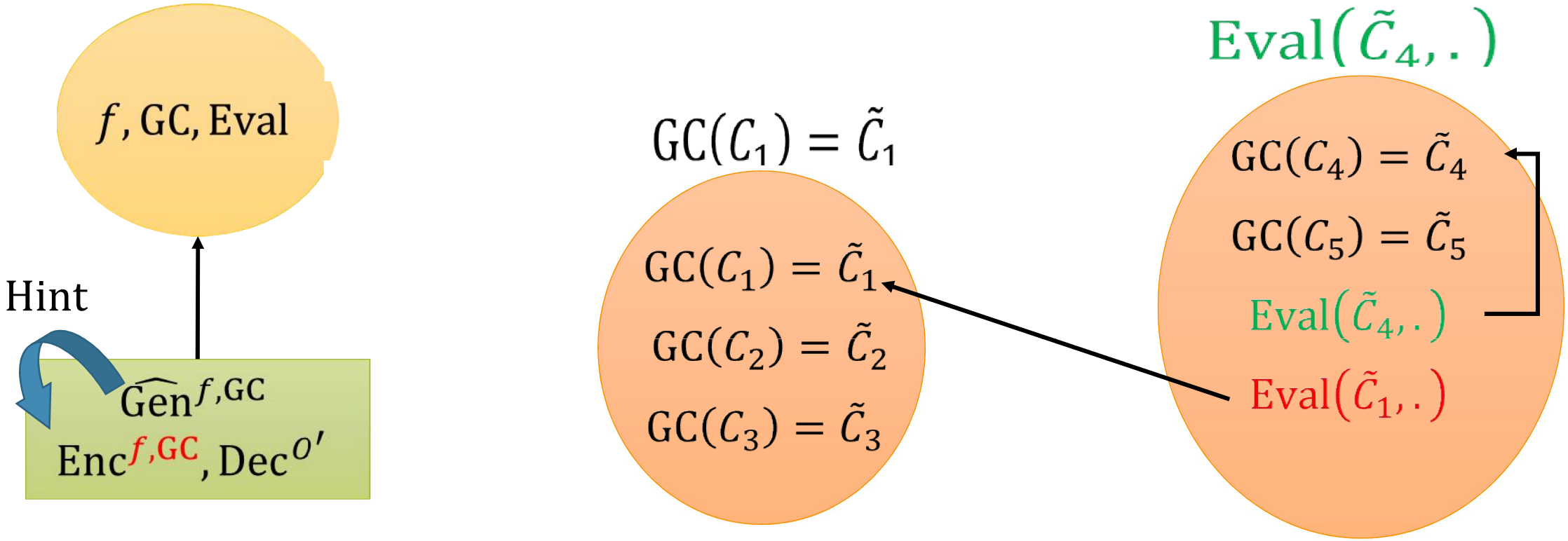
Compiling out Eval queries from Enc



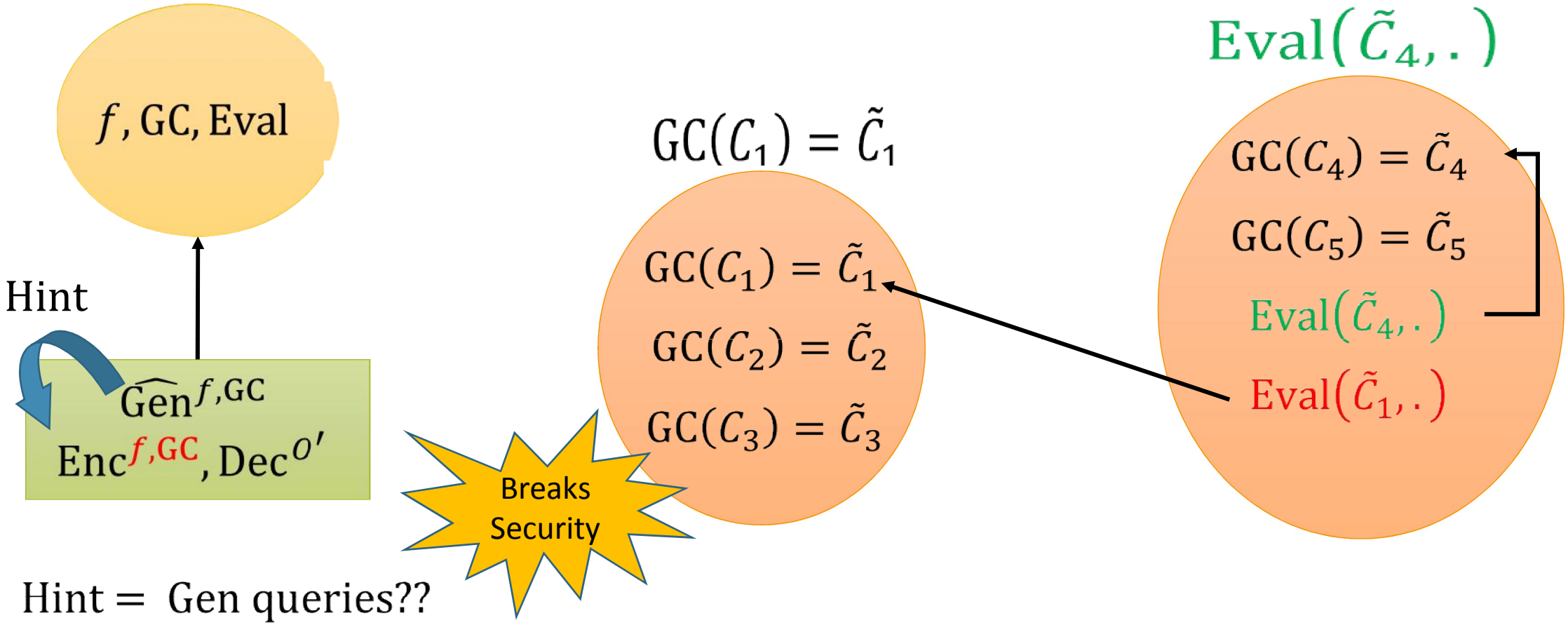
Compiling out Eval queries from Enc



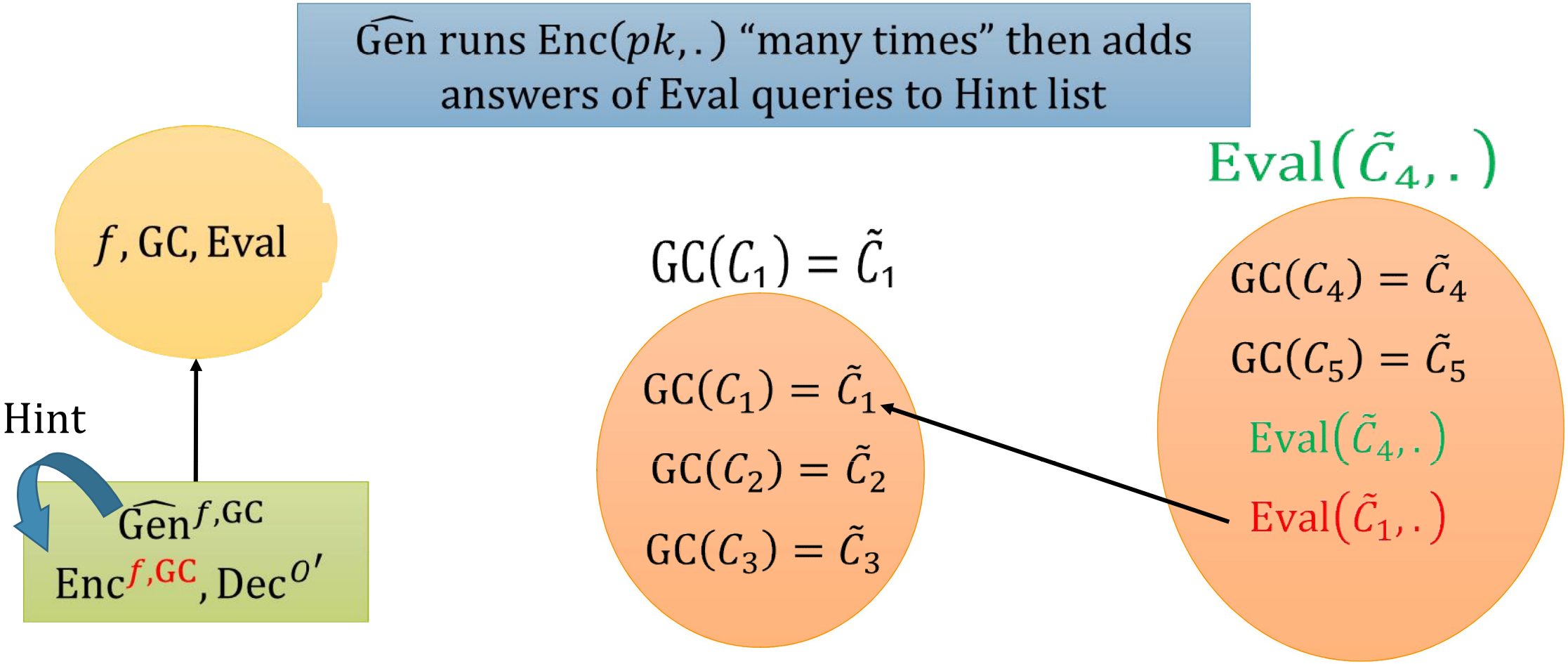
Compiling out Eval queries from Enc



Compiling out Eval queries from Enc

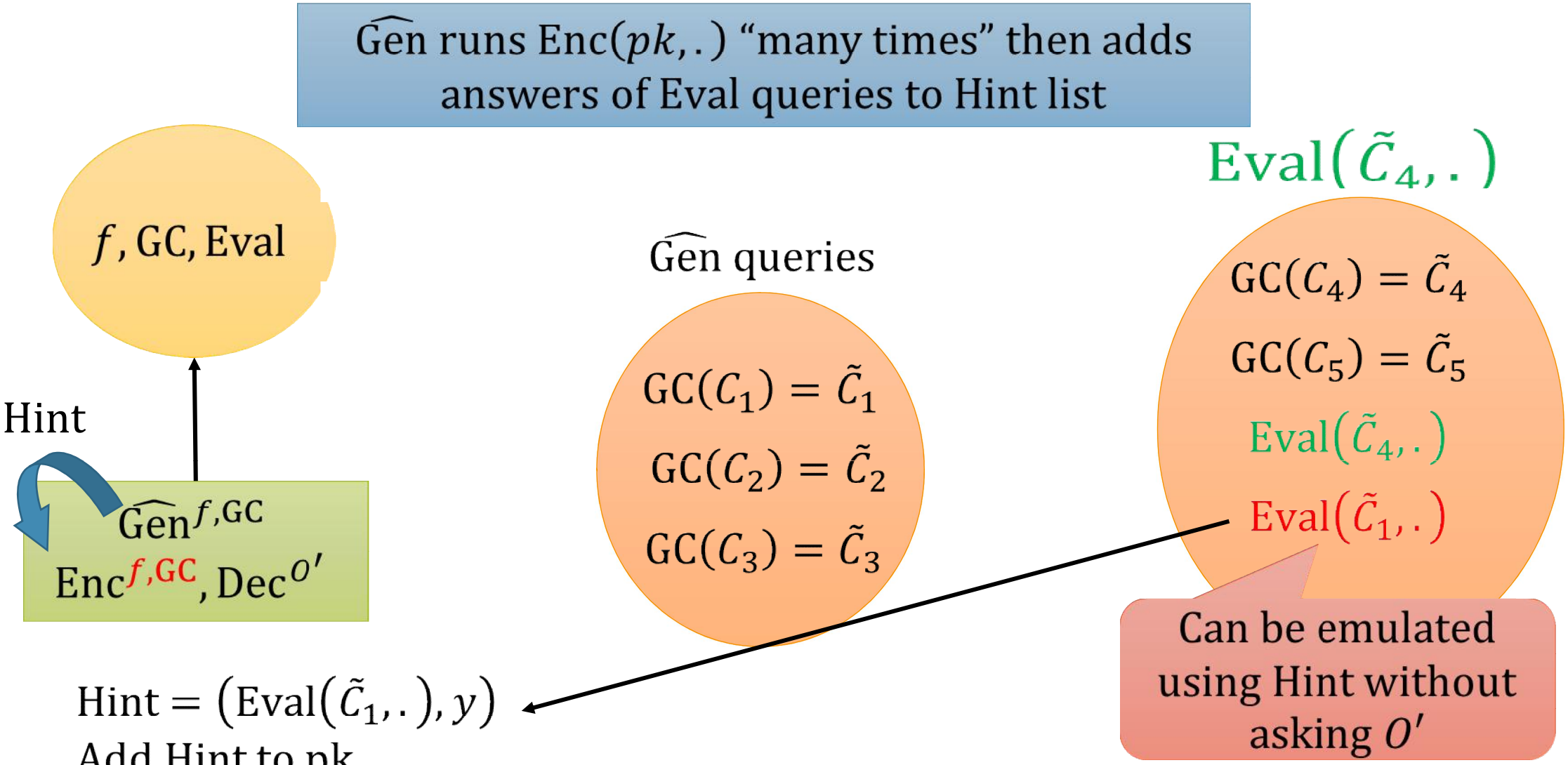


Compiling out Eval queries from Enc



Hint: Contains "heavy learnable" Eval queries of E

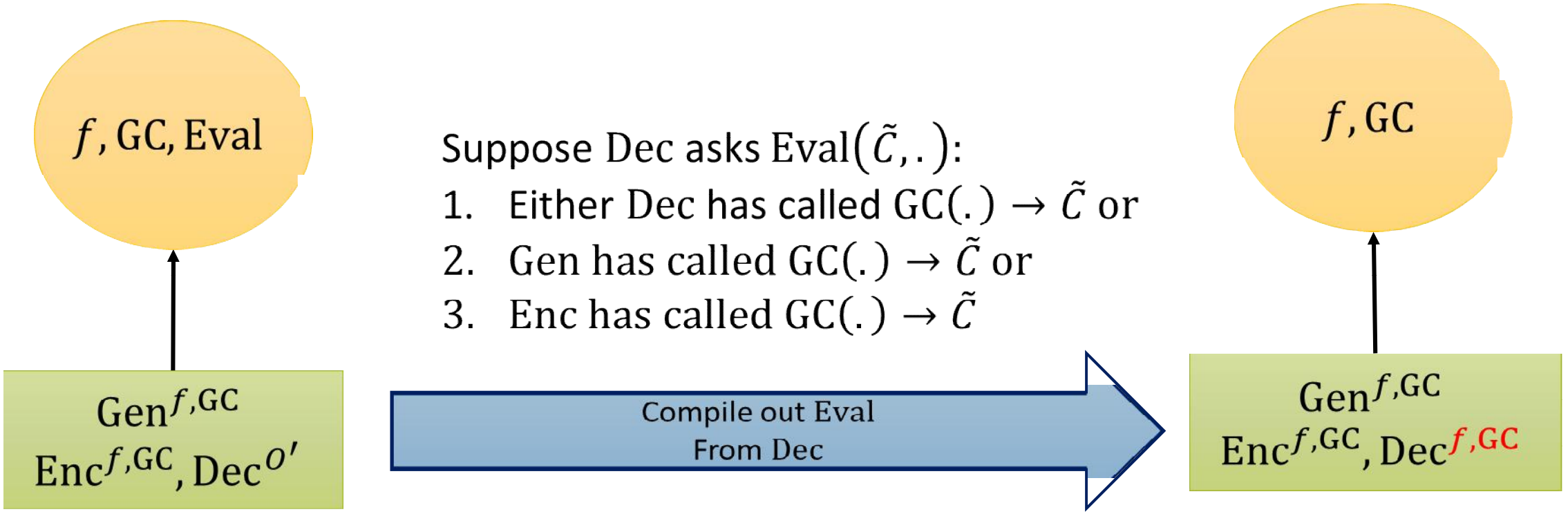
Compiling out Eval queries from Enc



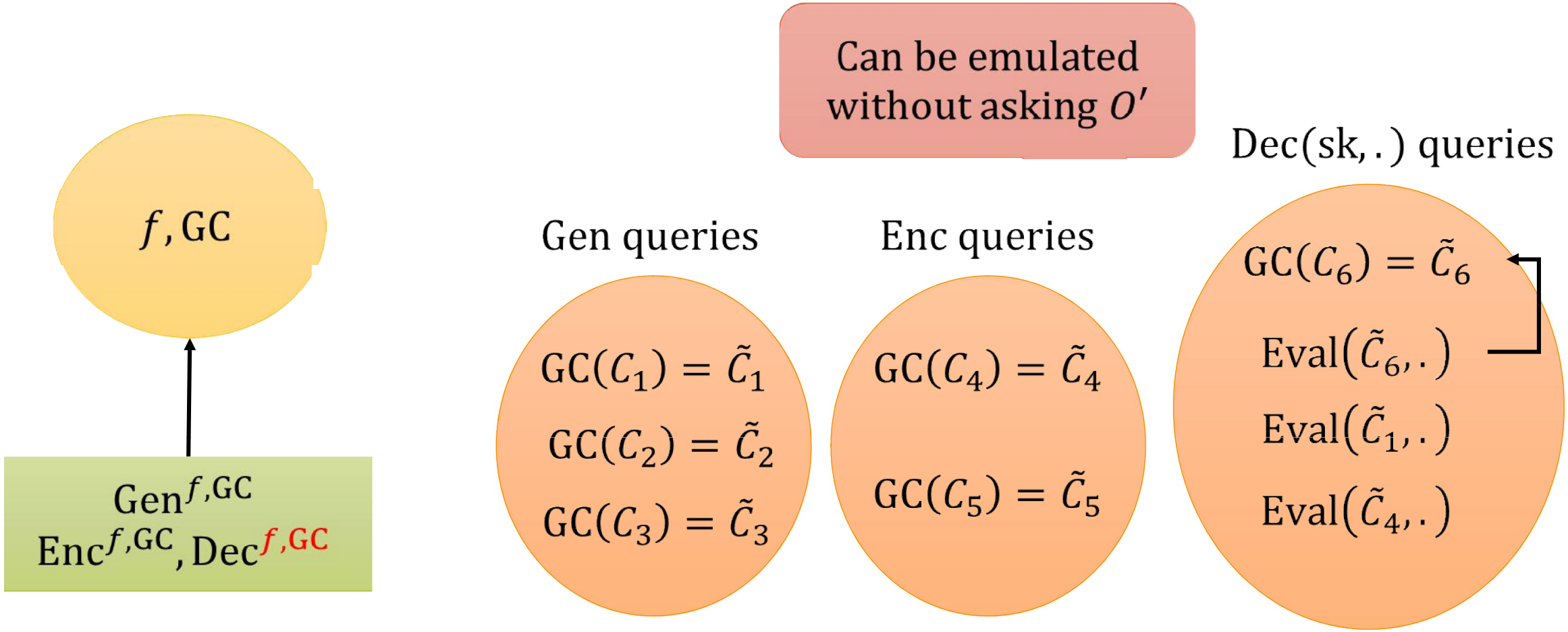
Compiling out Eval queries from Dec

- Key property of GC used: $|\tilde{C}| \gg |C|$

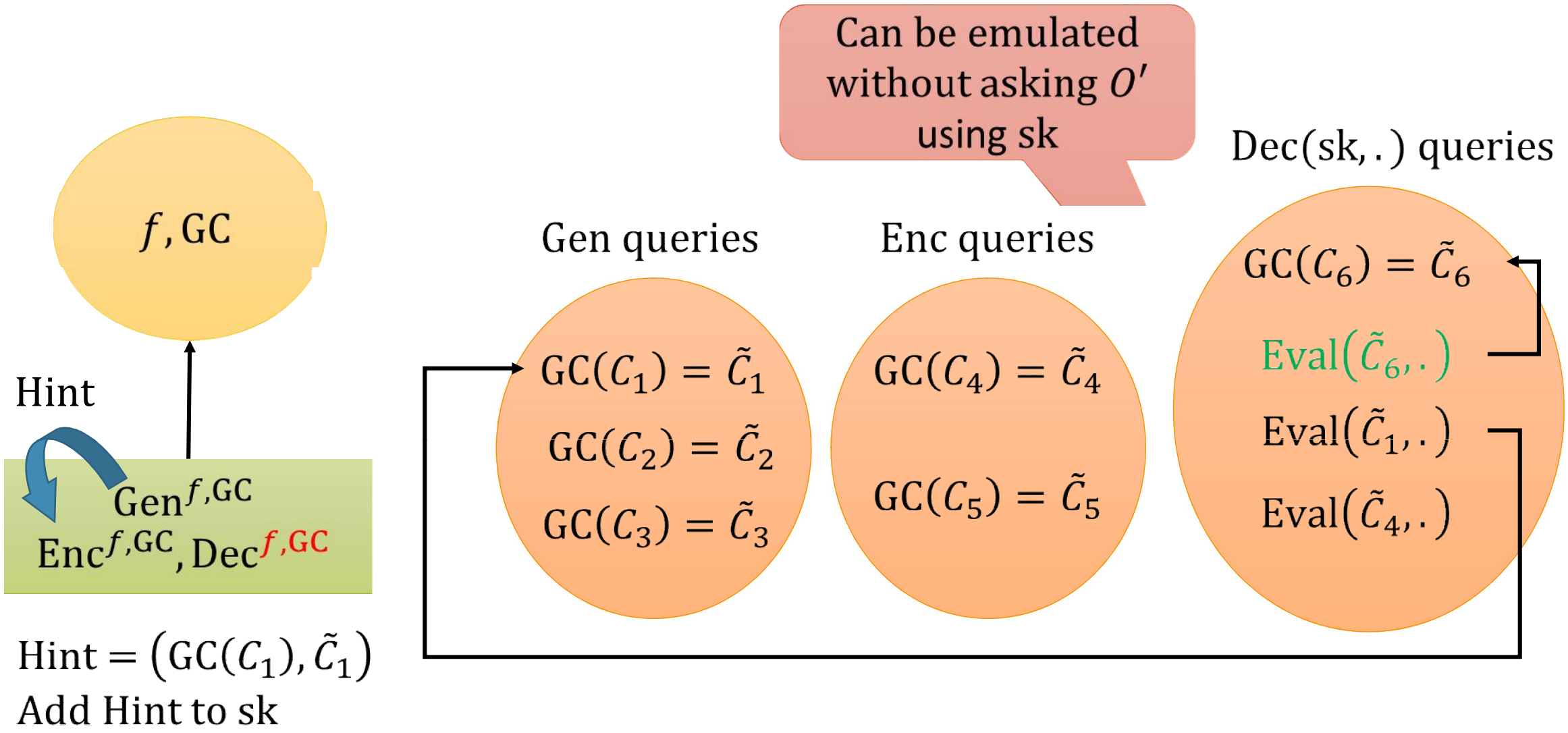
Problem for 2/3: Dec does not know $\tilde{C}$
Idea: Let Gen and Enc help Dec



Compiling out Eval queries from Dec



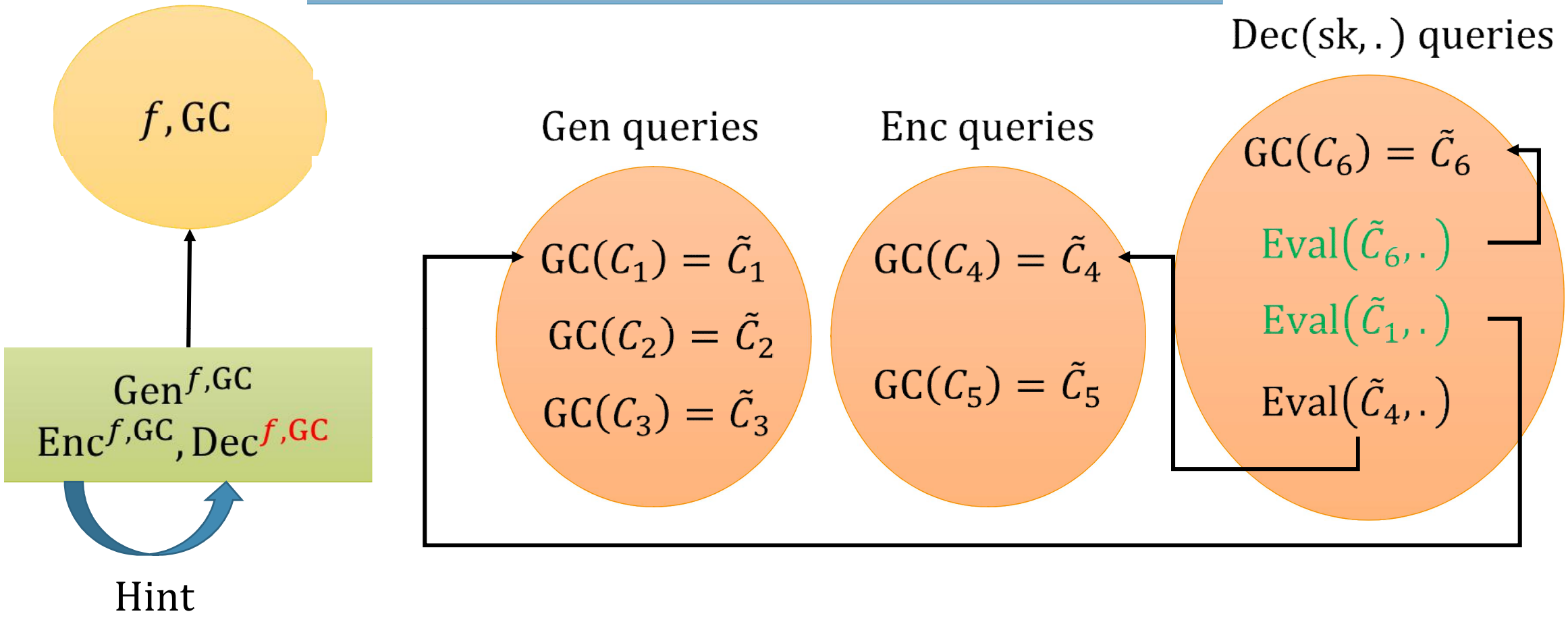
Compiling out Eval queries from Dec



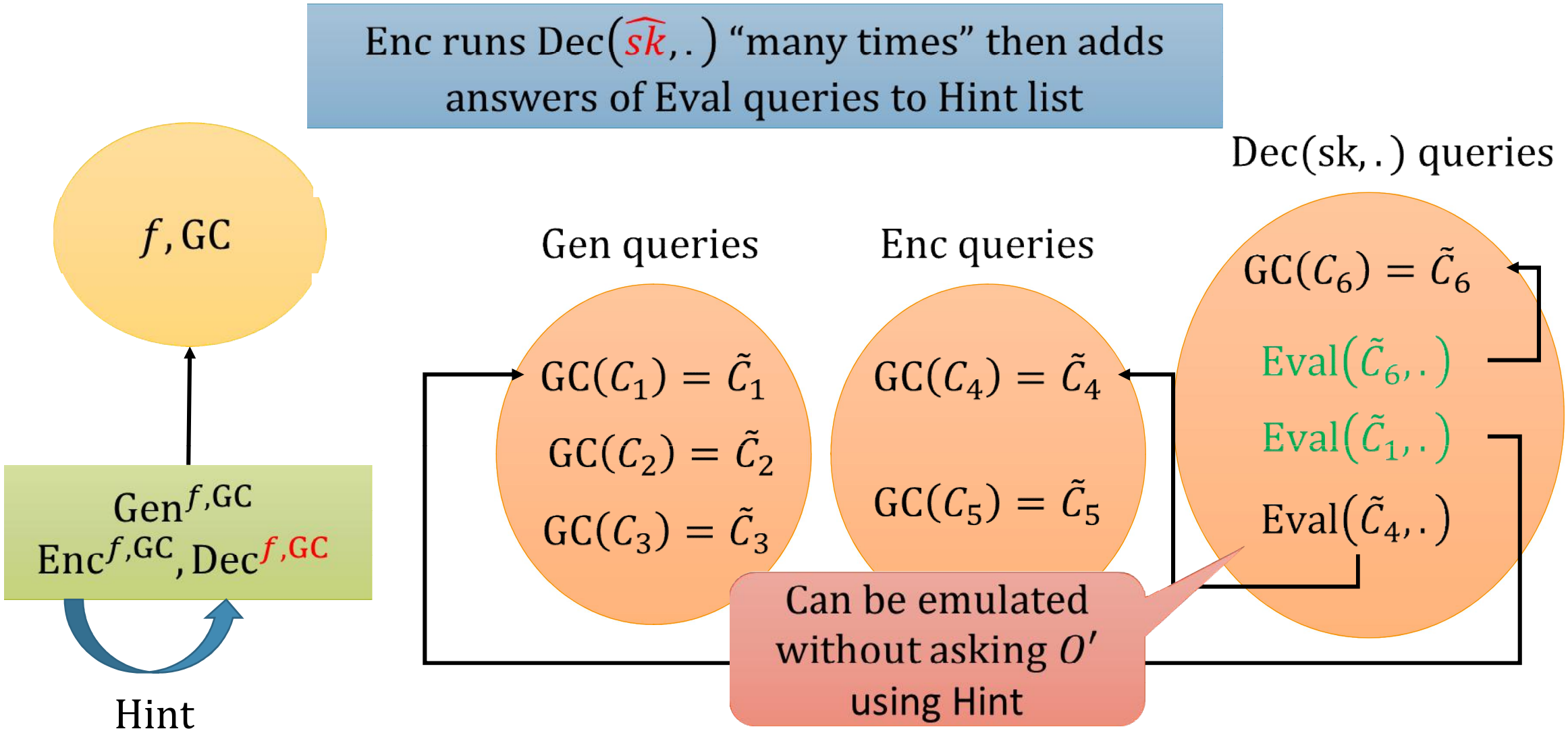
Compiling out Eval queries from Dec

Enc does not know sk

Enc runs Dec$(sk, .)$ "many times" then adds answers of Eval queries to Hint list???



Compiling out Eval queries from Dec



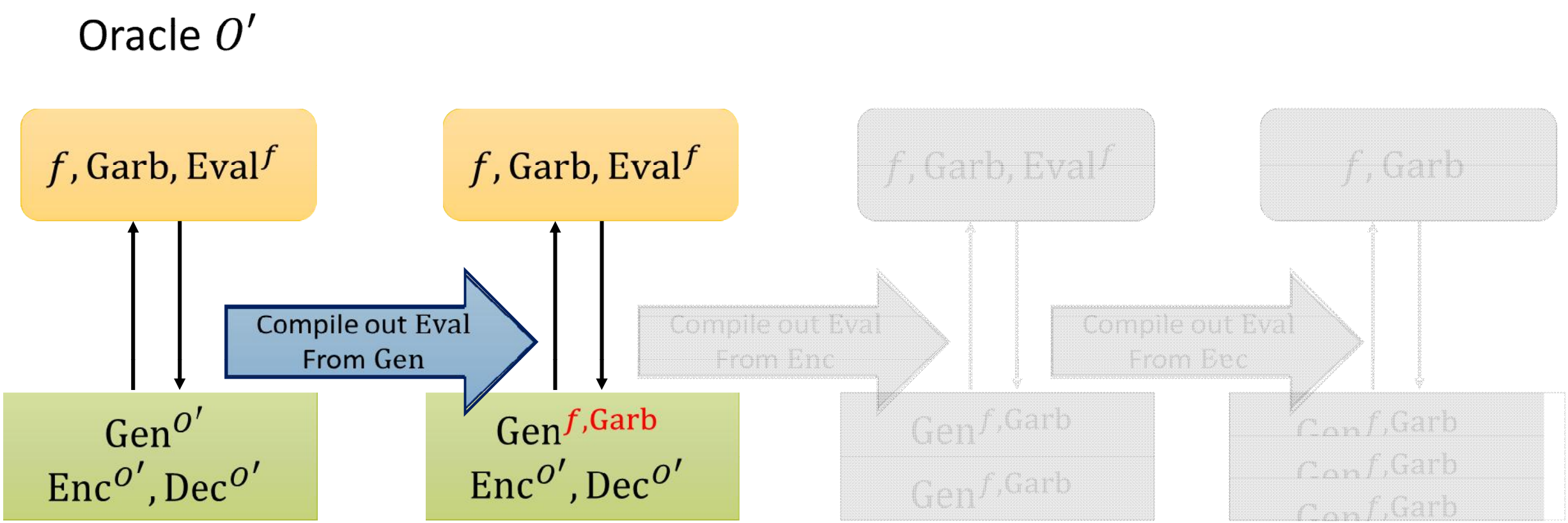
The Idealized Oracle

O $\quad \mathrm{OWF} + \mathrm{Garbling}^{\mathrm{OWF}}$

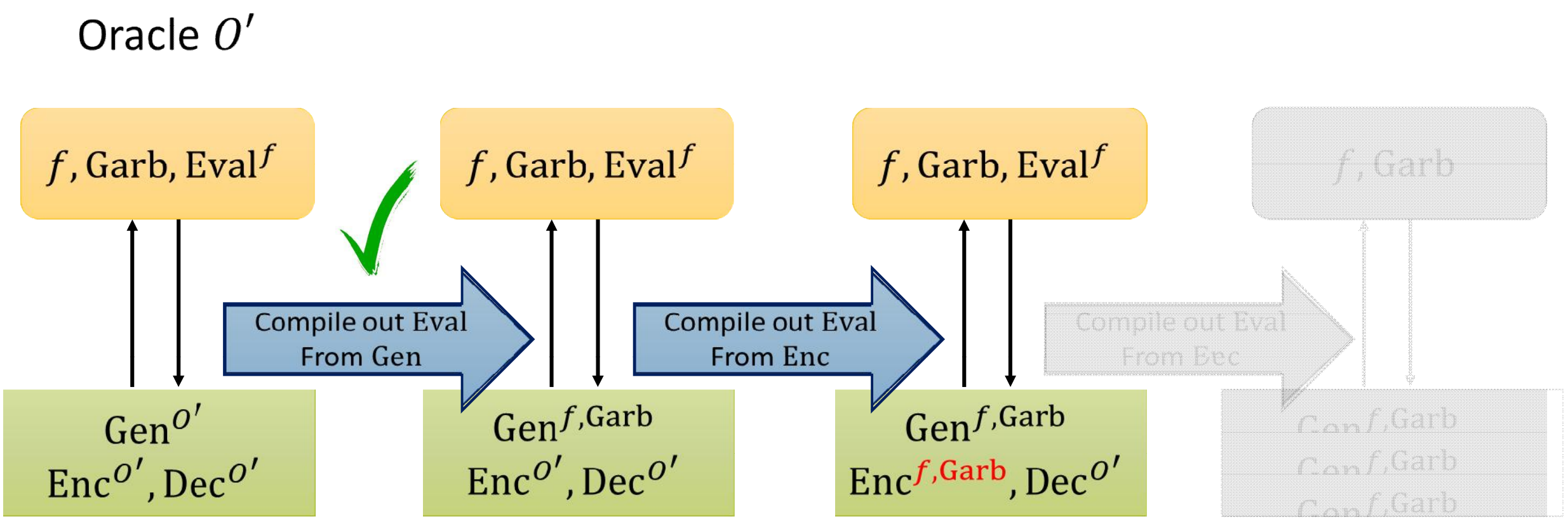
- We want oracle O to realize OWF + Garbling of C^{OWF}
- How to realize a OWF?
  - **Standard way**: Use a random oracle f
- How to realize garbling of circuits with OWF gates?
  - This models non-black-box use of OWF
  - **Solution:** Use a garbling oracle $(\mathrm{Garb}^f, \mathrm{Eval}^f)$ that garbles circuit C^f

O $\quad f, \mathrm{Garb}^f, \mathrm{Eval}^f$

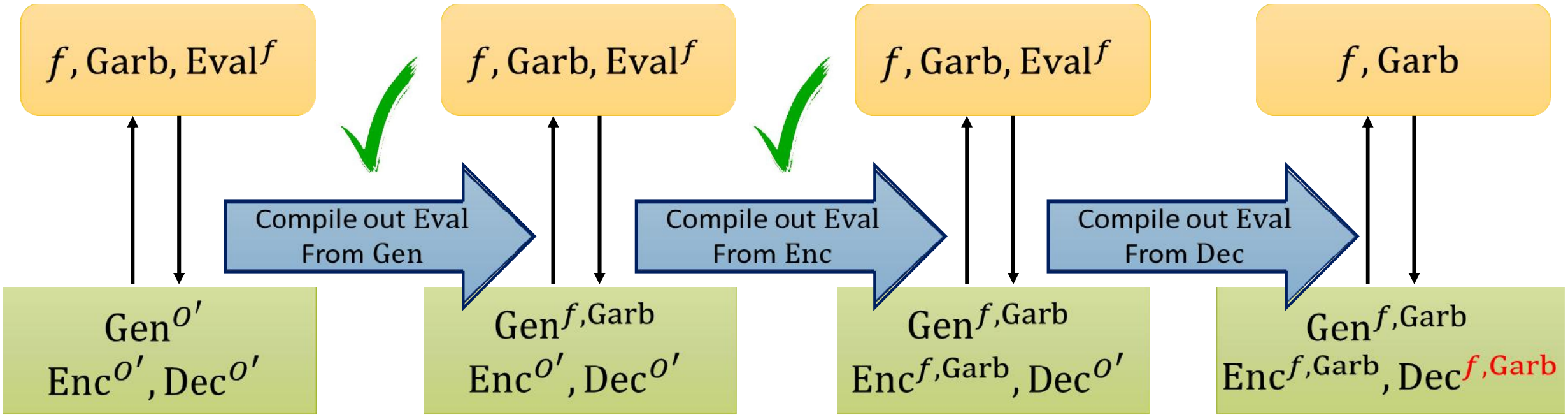
Compiling out Eval queries from Gen



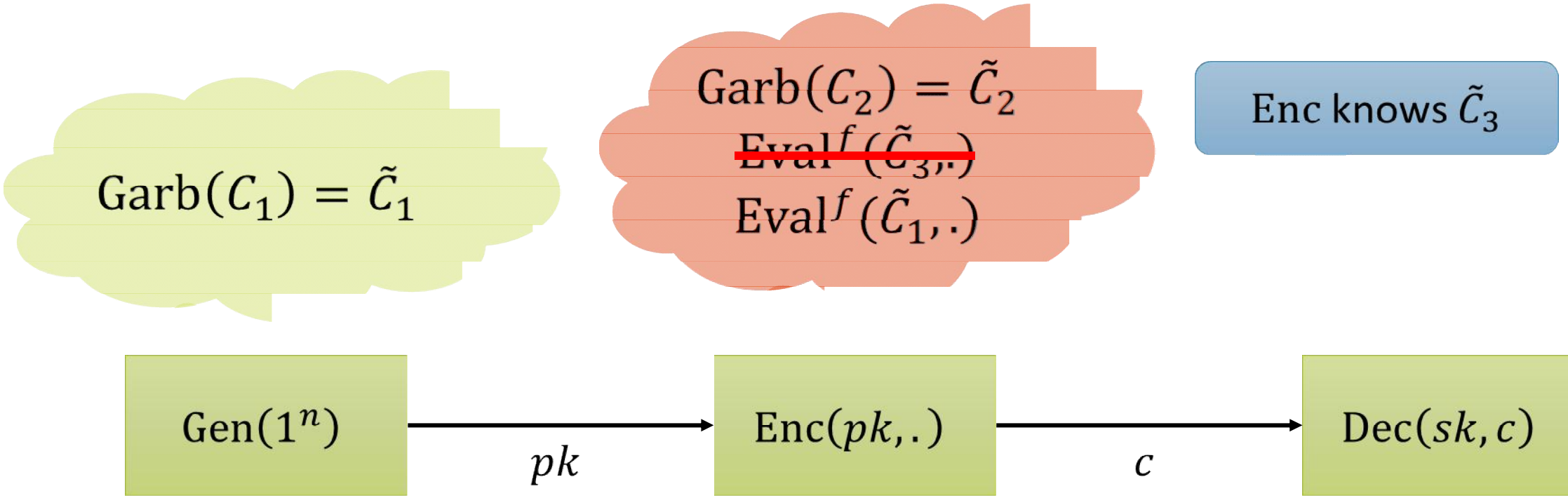
Compiling out Eval queries from Enc



Compiling out Eval queries from Dec



Compiling out Eval queries from Enc



Compiling out Eval queries from Dec

Challenge: Dec does not know C_1

Solution: $\widehat{\text{Gen}}$ adds $\left(\text{Garb}(C_1), \tilde{C}_1\right)$ to sk

$\text{Garb}(C_1) = \tilde{C}_1$

$\text{Garb}(C_2) = \tilde{C}_2$

$\text{Garb}(C_4) = \tilde{C}_4$

~~$\text{Eval}^{f}(\tilde{C}_3, .)$~~