Amplifying the Security of Functional Encryption, Unconditionally

Aayush Jain
UCLA

Alexis Korb
UCLA

Nathan Manohar
UCLA

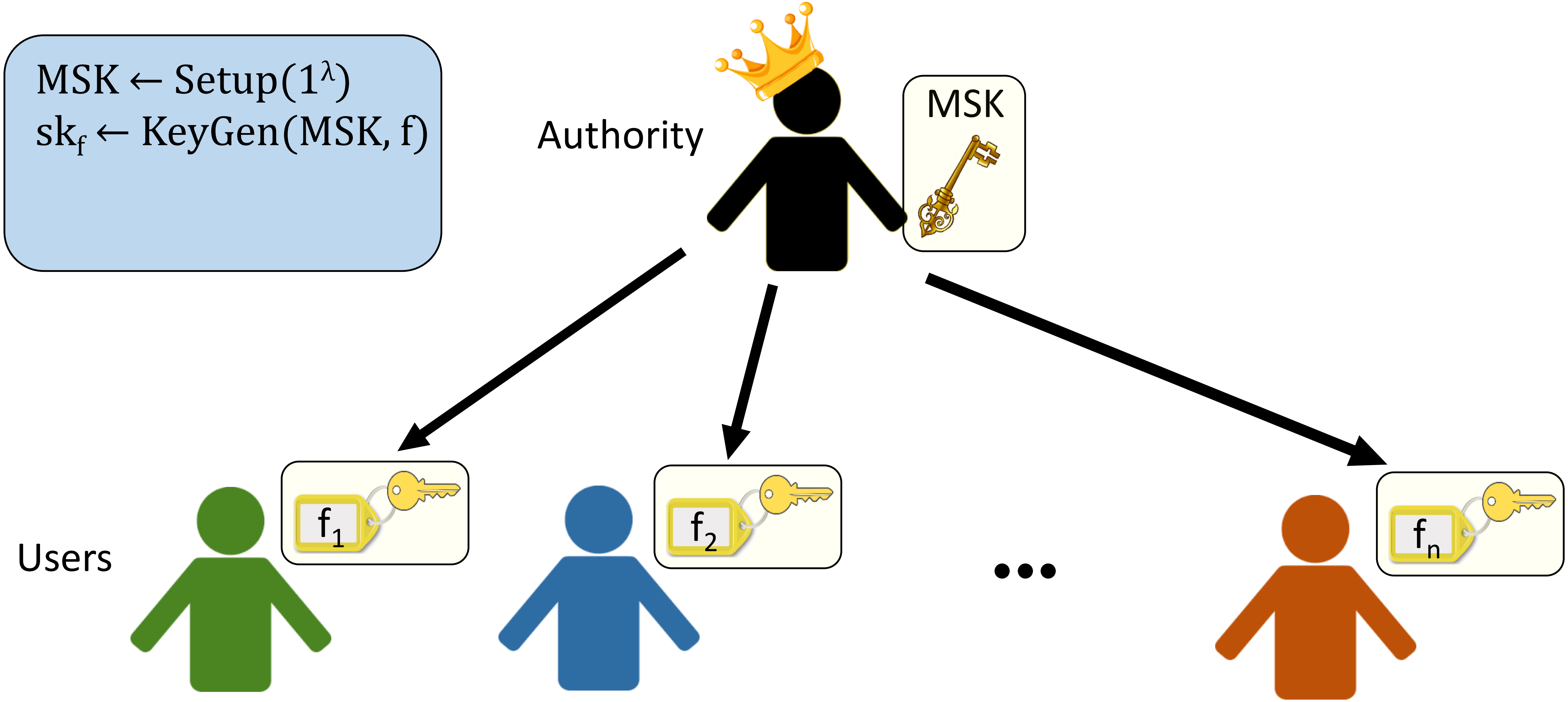
Amit Sahai
UCLA

(Secret Key) Functional Encryption

[SW05, BSW11, O'N10]

$\mathrm{MSK} \leftarrow \mathrm{Setup}(1^\lambda)$

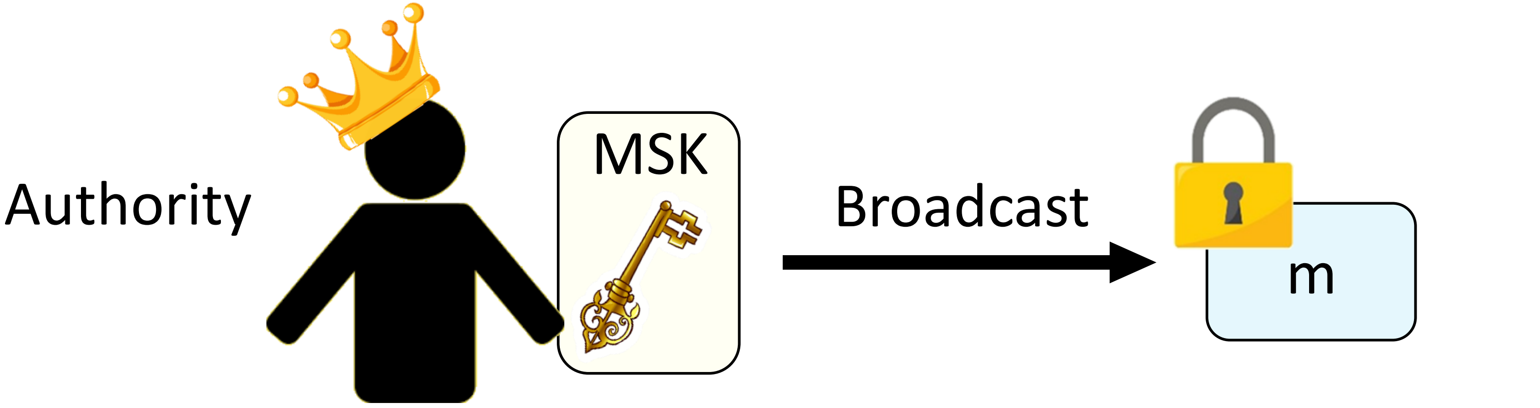
$\mathrm{sk}_f \leftarrow \mathrm{KeyGen}(\mathrm{MSK}, f)$

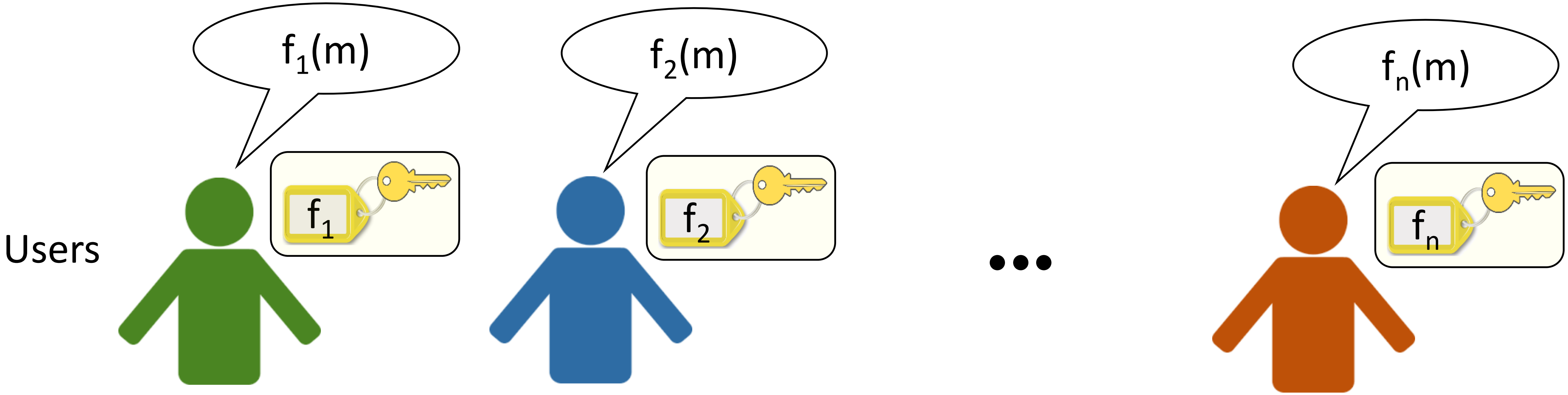


(Secret Key) Functional Encryption

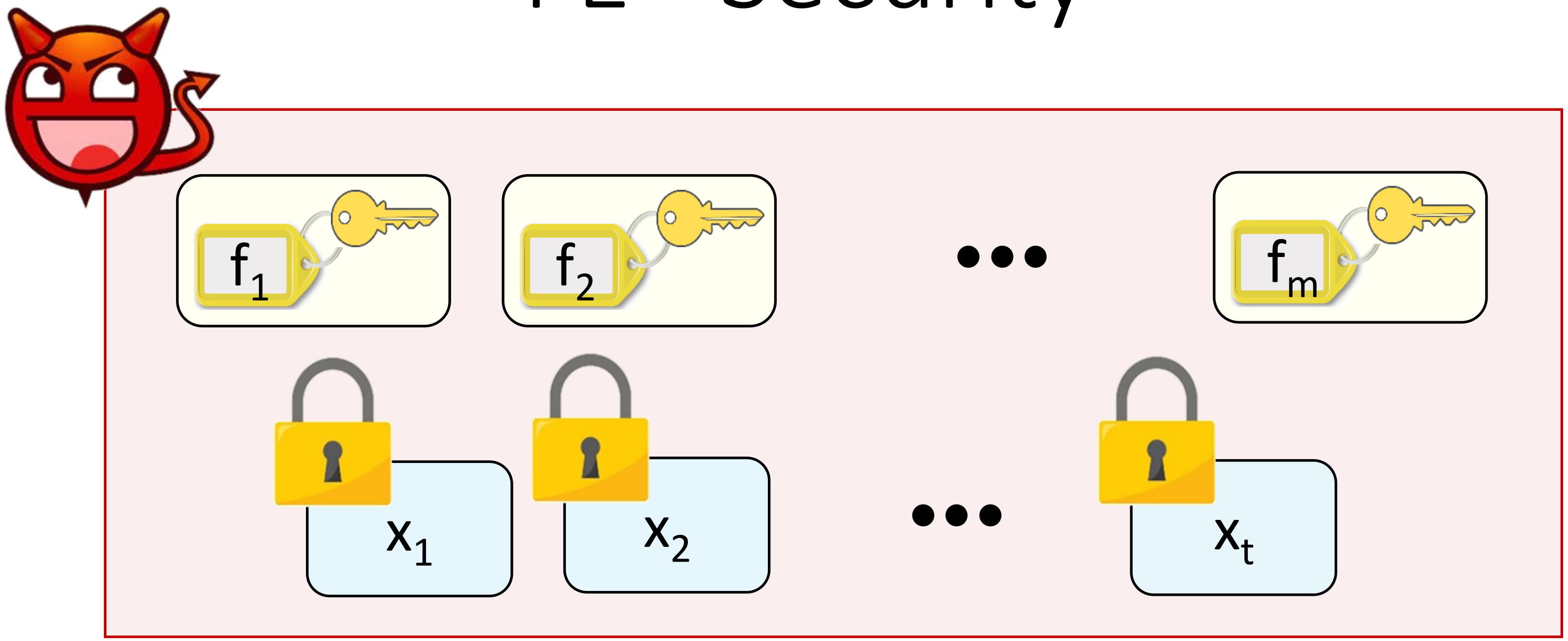
[SW05, BSW11, O'N10]

$\mathrm{MSK} \leftarrow \mathrm{Setup}(1^\lambda)$
$\mathrm{sk}_f \leftarrow \mathrm{KeyGen}(\mathrm{MSK}, f)$
$\mathrm{ct} \leftarrow \mathrm{Enc}(\mathrm{MSK}, m)$
$y \leftarrow \mathrm{Dec}(\mathrm{sk}_f, \mathrm{ct})$



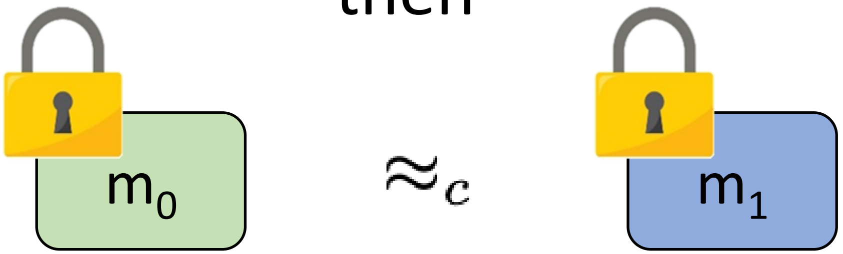


FE - Security



Idea: Given sk_f and Enc(m), adversary should only learn f(m).

If $\forall i$, $f_i(m_0) = f_i(m_1)$, then

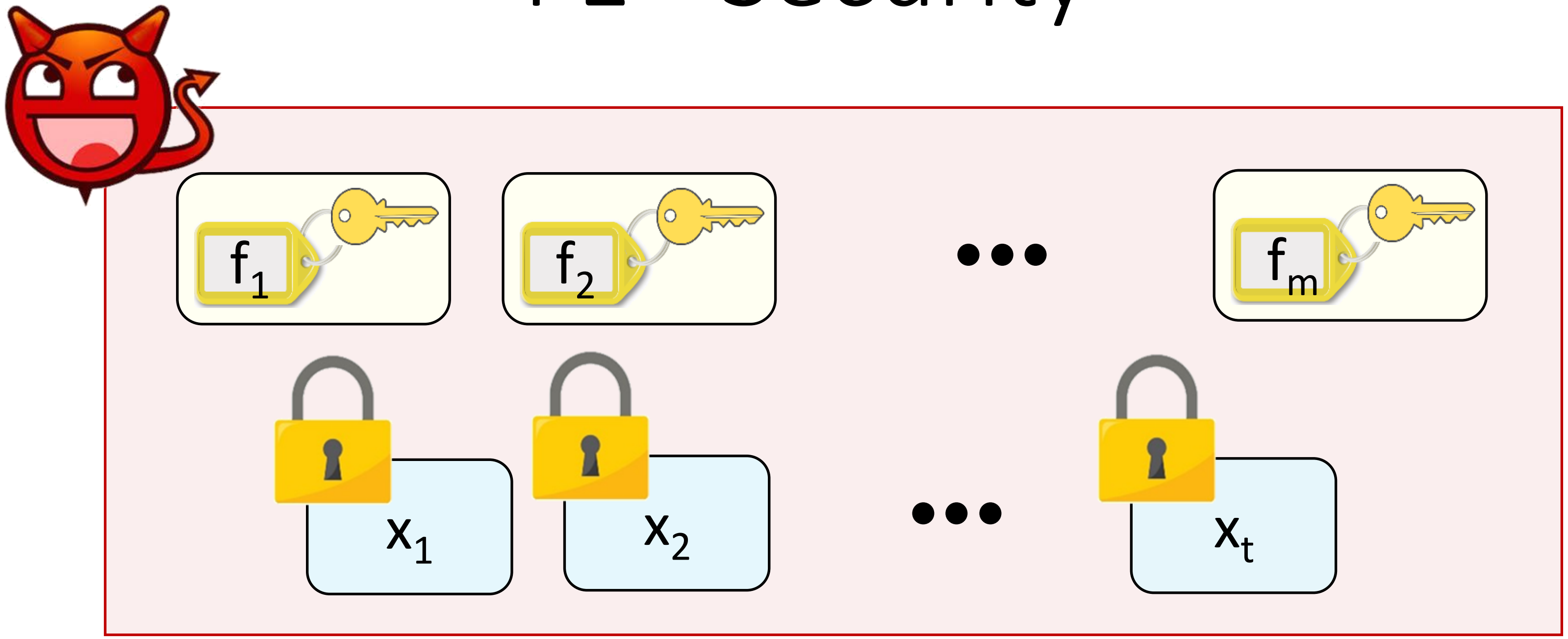


FE Amplification

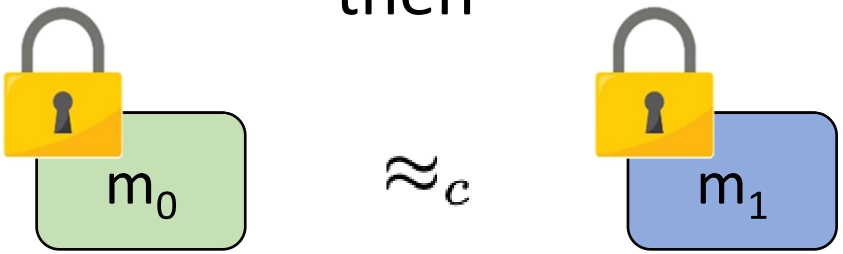
- Fundamental question
- New sources of hardness may lead to weak primitives → amplify to fully secure
- Results can be *unconditional*

FE - Security

Idea: Given sk_f and Enc(m), adversary should only learn f(m).



If $\forall i$, $f_i(m_0) = f_i(m_1)$, then



***p*-secure FE** = Adversary can distinguish between Enc(m_0) and Enc(m_1) with probability at most *p*

FE Amplification

- Fundamental question
- New sources of hardness may lead to weak primitives → amplify to fully secure
- Results can be *unconditional*

Previous Work

- [AJS18, AJL+19] Amplify FE from $(1 - 1/\text{poly}(\lambda))$-security to full security assuming subexponentially secure LWE.
  - Preserves compactness and sublinearity
  - Polynomial and subexponential versions
- No other FE amplification results known

Previous Work

- [AJS18, AJL+19] Amplify FE from (1 - 1/poly(λ))-security to full security assuming subexponentially secure LWE.
  - Preserves compactness and sublinearity
  - Polynomial and subexponential versions
- No other FE amplification results known

Can we get FE amplification from weaker assumptions?

Previous Work

- [AJS18, AJL+19] Amplify FE from (1 - 1/poly(λ))-security to full security assuming subexponentially secure LWE.
  - Preserves compactness and sublinearity
  - Polynomial and subexponential versions
- No other FE amplification results known

Can we get FE amplification from weaker assumptions?

YES!

Previous Work

- [AJS18, AJL+19] Amplify FE from $\underline{(1 - 1/poly(\lambda))}$-security to full security assuming subexponentially secure LWE.
  - Preserves compactness and sublinearity
  - Polynomial and subexponential versions
- No other FE amplification results known

Our Work

- Amplify FE from $\underline{\varepsilon}$-security for any $\underline{\text{constant } \varepsilon \in (0,1)}$ to full security, unconditionally.
  - Preserves compactness
  - Polynomial and subexponential versions

Two Steps in Amplification

ε-secure FE -> fully secure FE

1. Constant ε -> arbitrarily small constant ε'

2. Small constant ε'-> fully secure

Two Steps in Amplification

ε-secure FE -> fully secure FE

1. Constant ε -> arbitrarily small constant ε'
   - Uses nesting technique (NEW!)

Nested PKE Amplification

For any constant $\varepsilon \in (0,1)$ and ε-secure PKE scheme *PKE*, the PKE scheme *PKE** obtained by composing *PKE* with itself is $\varepsilon^2 + negl(\lambda) - \text{secure.}$

2. Small constant ε'-> fully secure

Two Steps in Amplification

ε-secure FE -> fully secure FE

1. Constant ε -> arbitrarily small constant ε'
   - Uses nesting technique (NEW!)

Nested PKE Amplification

For any constant $\varepsilon \in (0,1)$ and ε-secure PKE scheme *PKE*, the PKE scheme *PKE** obtained by composing *PKE* with itself is $\varepsilon^2 + negl(\lambda) - \text{secure}$.

2. Small constant ε'-> fully secure
   - Parallel repetition
   - Set homomorphic secret sharing (NEW!)

Two Steps in Amplification

ε-secure FE -> fully secure FE

1. Constant ε -> arbitrarily small constant ε'
   - Uses nesting technique (NEW!)

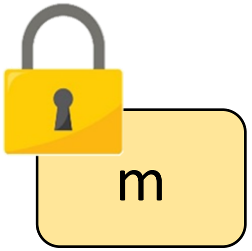
Nested PKE Amplification

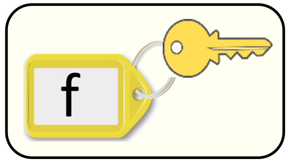
For any constant $\varepsilon \in (0,1)$ and ε-secure PKE scheme *PKE*, the PKE scheme *PKE** obtained by composing *PKE* with itself is $\varepsilon^2 + negl(\lambda) - \text{secure.}$

2. Small constant ε'-> fully secure
   - Parallel repetition
   - Set homomorphic secret sharing (NEW!)

Nested FE

$\mathrm{CT} \leftarrow \mathrm{Enc}(\mathrm{MSK}_1, \mathrm{m})$

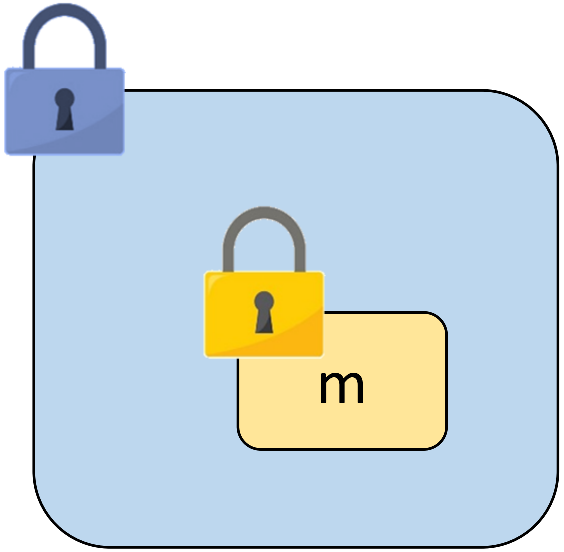


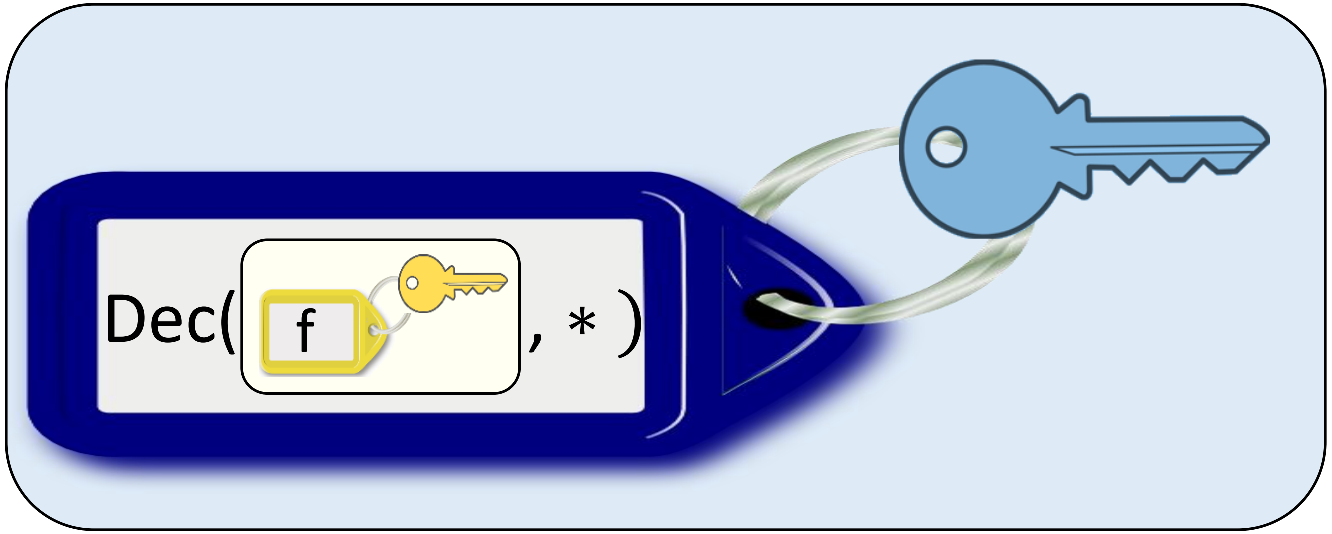


$\mathrm{SK}_\mathrm{f} \leftarrow \mathrm{KeyGen}(\mathrm{MSK}_1, \mathrm{f})$

Nested FE

$\mathrm{CT} \leftarrow \mathrm{Enc}(\mathrm{MSK}_2,\ \mathrm{Enc}(\mathrm{MSK}_1, \mathrm{m}))$

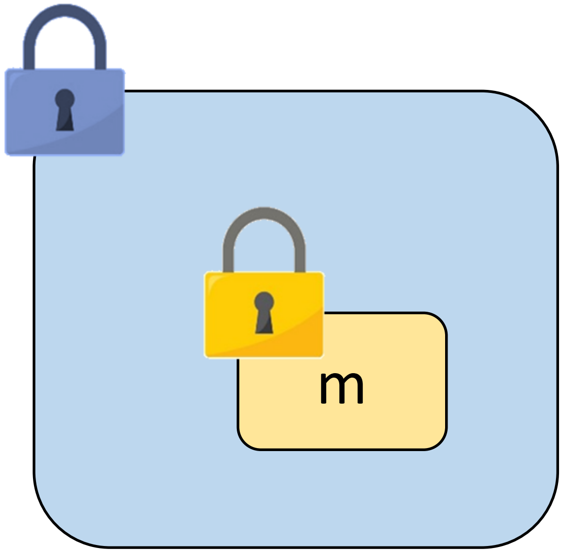


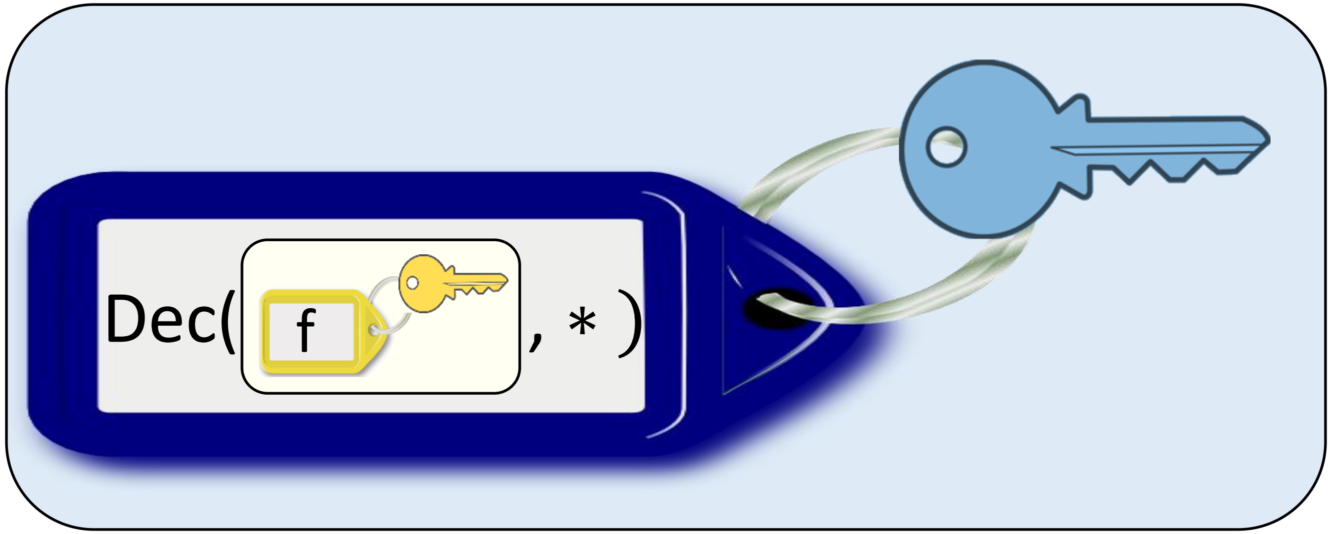


$\mathrm{SK}_f \leftarrow \mathrm{KeyGen}(\mathrm{MSK}_2, \mathrm{Dec}(\mathrm{KeyGen}(\mathrm{MSK}_1, \mathrm{f}), *))$

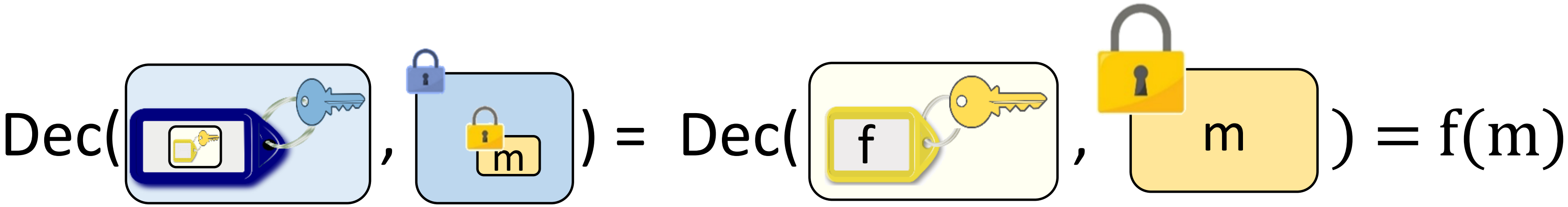
Nested FE

$\mathrm{CT} \leftarrow \mathrm{Enc}(\mathrm{MSK}_2,\ \mathrm{Enc}(\mathrm{MSK}_1, \mathrm{m}))$

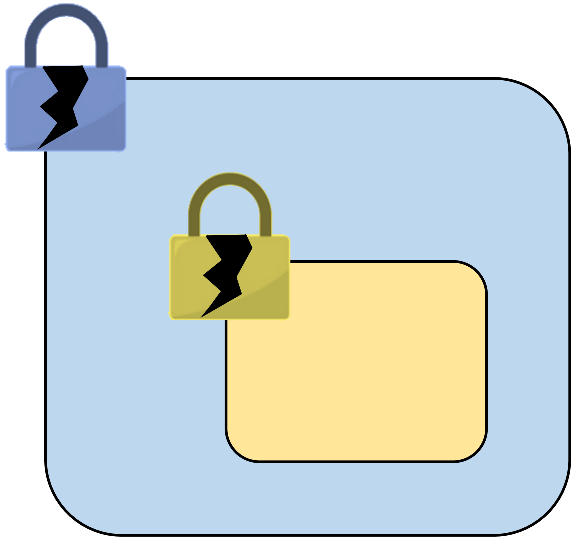




$\mathrm{SK}_f \leftarrow \mathrm{KeyGen}(\mathrm{MSK}_2, \mathrm{Dec}(\mathrm{KeyGen}(\mathrm{MSK}_1, \mathrm{f}), *\ ))$



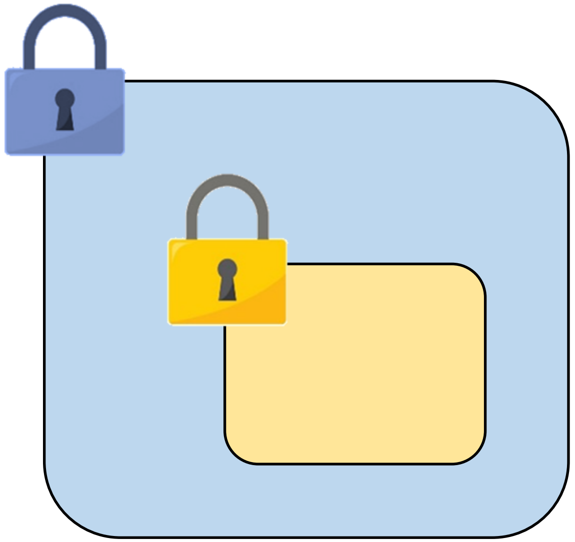
Amplification of Nested Primitives



Intuition: If one layer is secure, then the whole thing is secure

Expectation: Amplify security from $\varepsilon \rightarrow \varepsilon^2$

Nested PKE

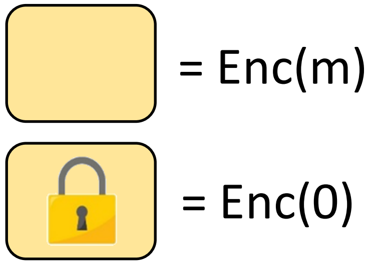


$\text{CT} \leftarrow \mathrm{Enc}(\mathrm{PK}_2,\ \mathrm{Enc}(\mathrm{PK}_1, \mathrm{m}))$

$\text{SK} \leftarrow (\mathrm{SK}_1, \mathrm{SK}_2)$

Security:

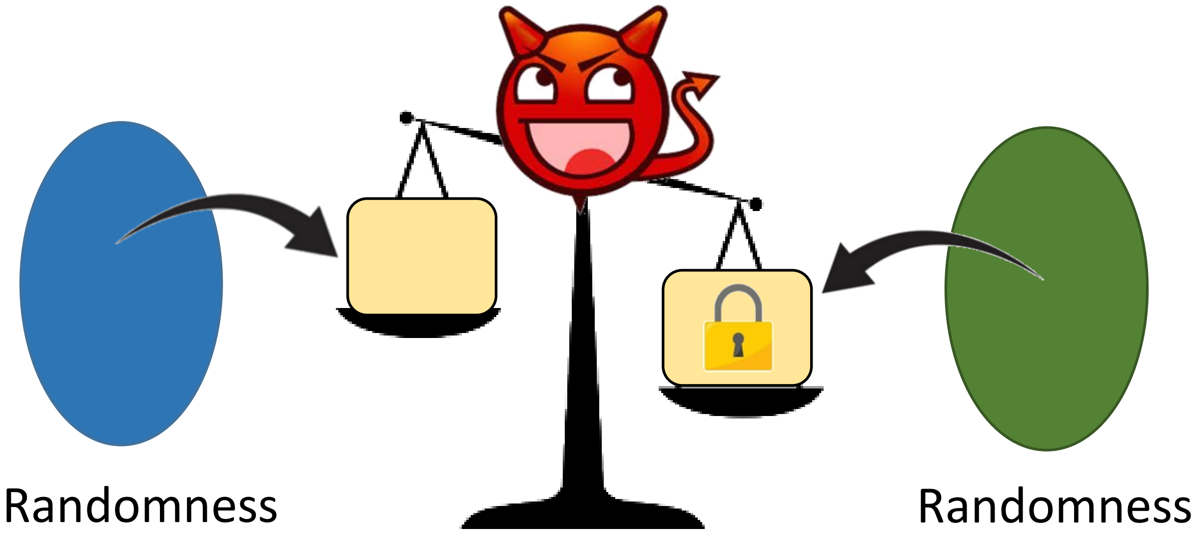
$\text{Enc(m)} \approx_c \text{Enc(0)}$

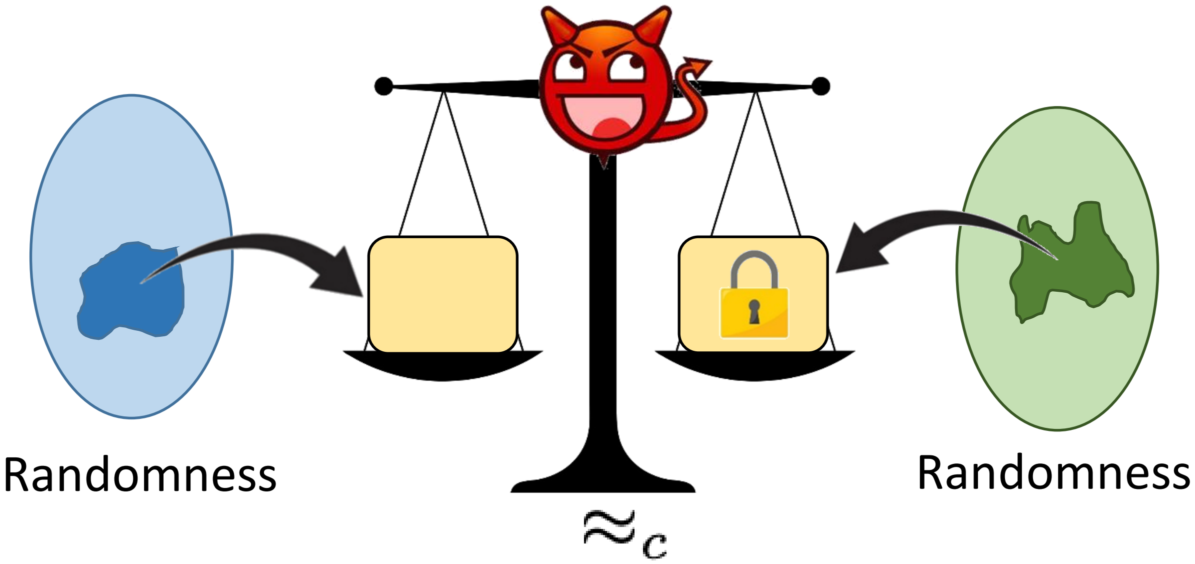


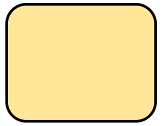
Hardcore Measures

[Imp95, MT10]

Weak indistinguishability over uniform randomness → Strong indistinguishability over hardcore measures





 = Enc(m)

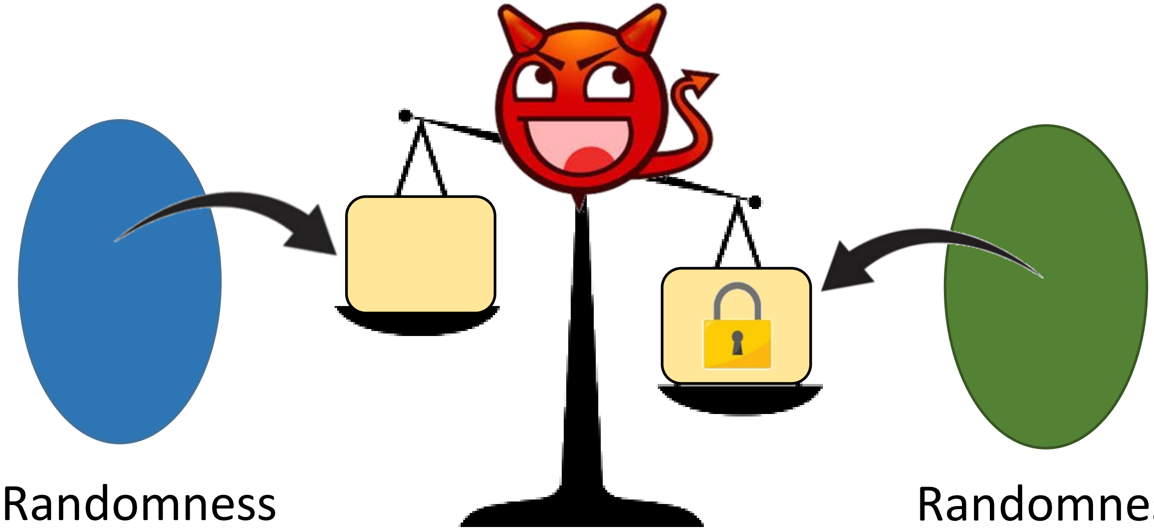
 = Enc(0)

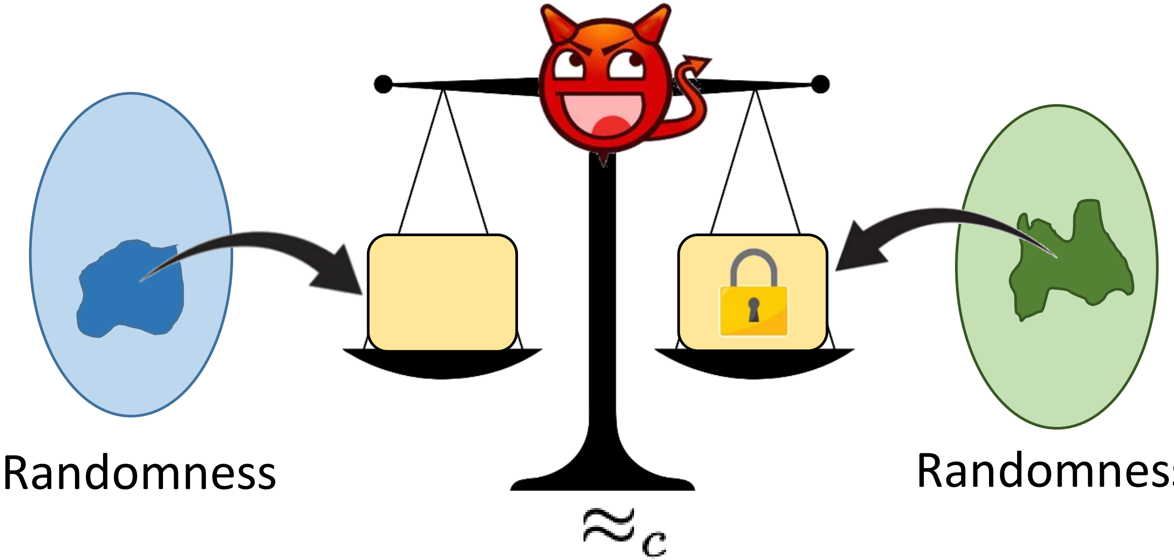
Hardcore Measures

[Imp95, MT10]

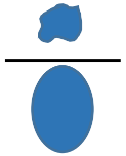
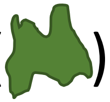
Weak indistinguishability over uniform randomness

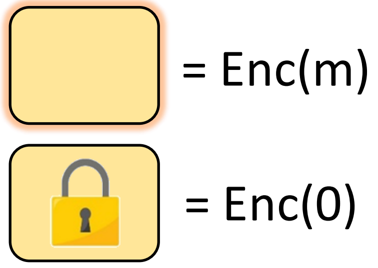
Strong indistinguishability over hardcore measures





ε - distinguishable

density() =  = density() = 1 - ε

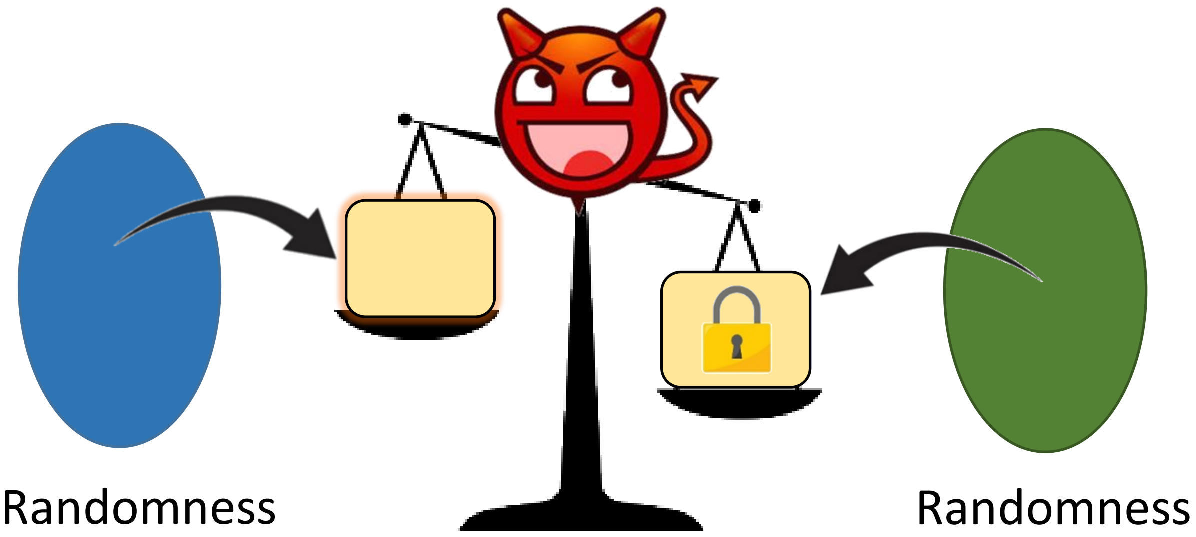


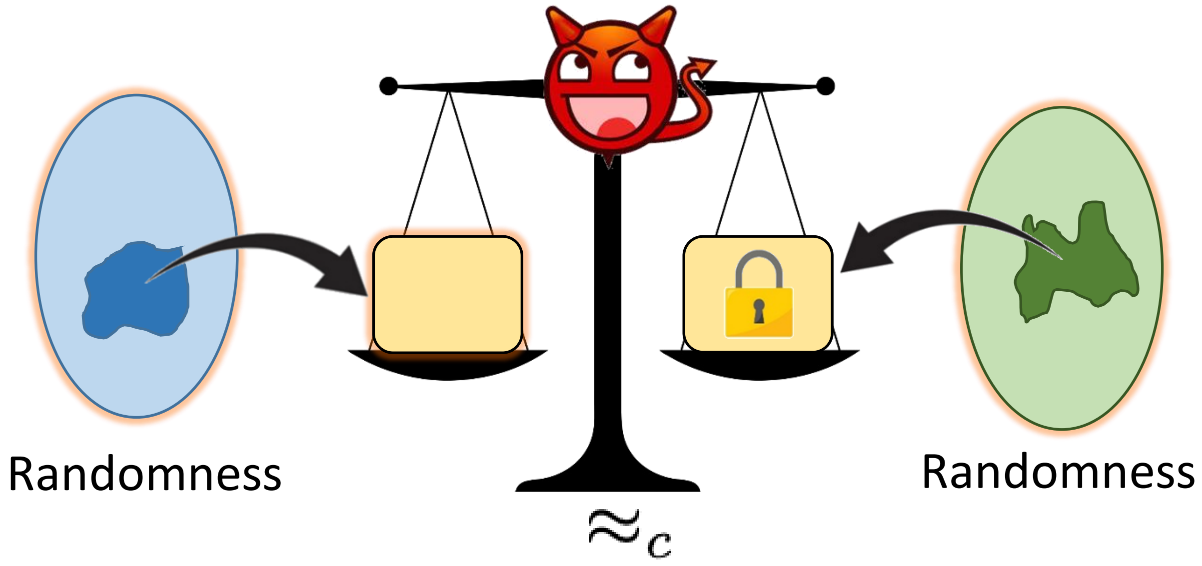
Hardcore Measures

[Imp95, MT10]

Weak indistinguishability over uniform randomness

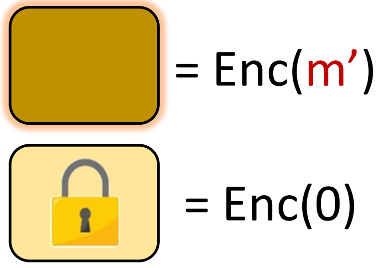
Hardcore measures depend on the input to the encryption.





ε - distinguishable

density( ) = — = density( ) = 1 - ε

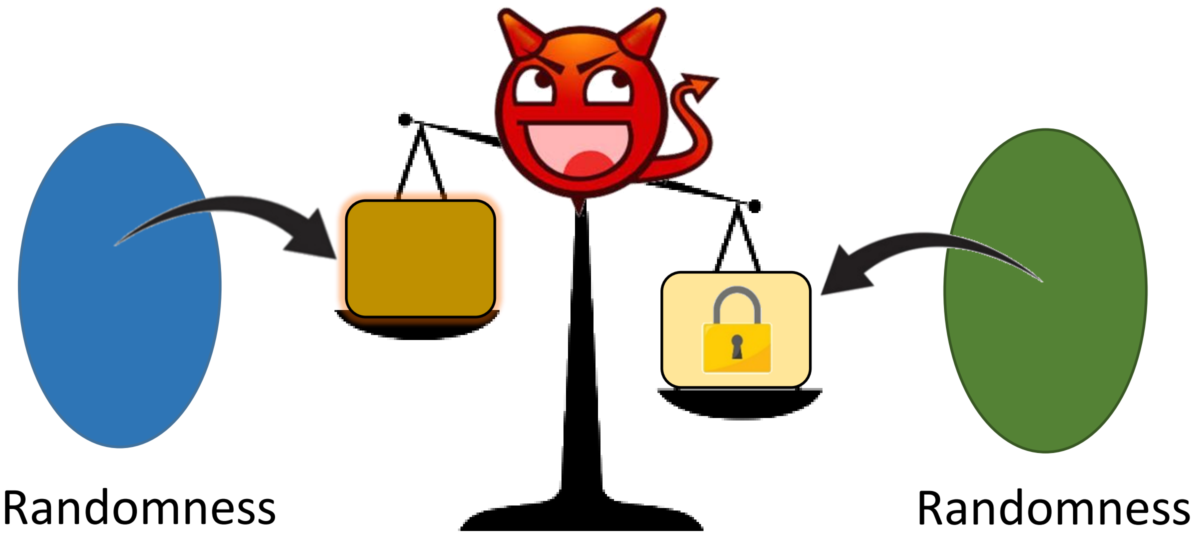


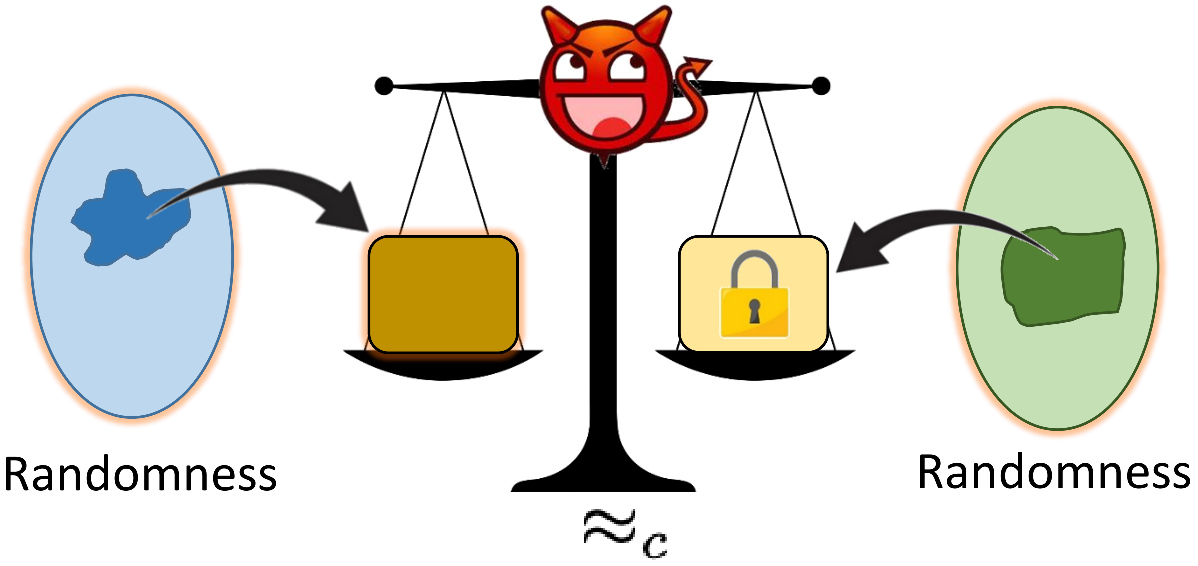
Hardcore Measures

[Imp95, MT10]

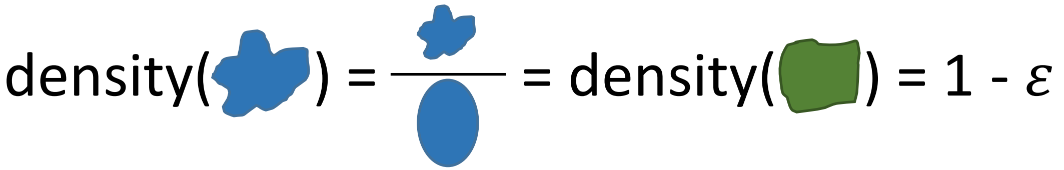
Weak indistinguishability over uniform randomness

Hardcore measures depend on the input to the encryption.





ε - distinguishable



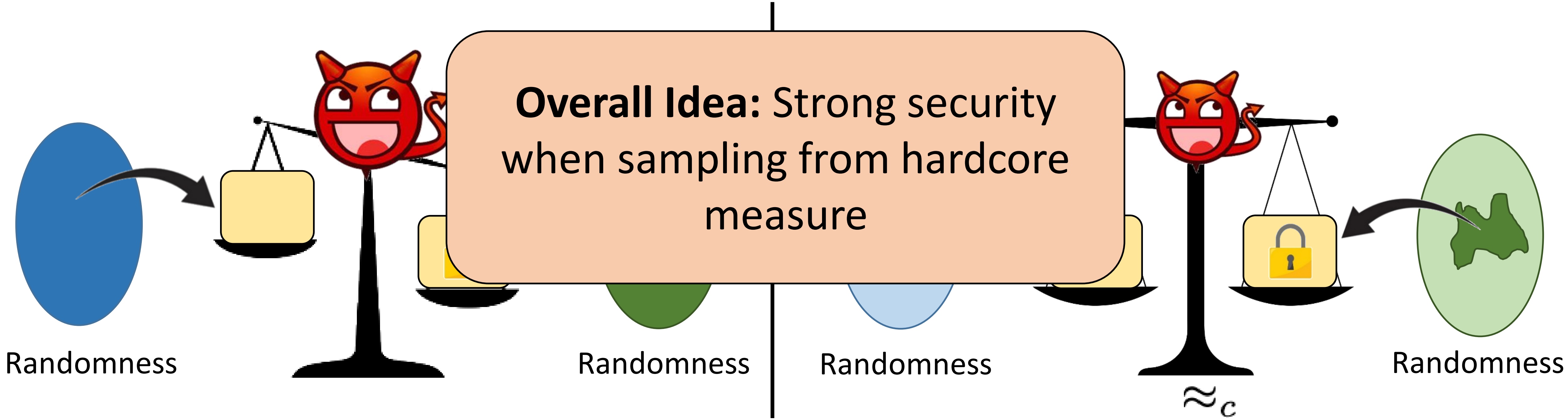
= Enc(m)

= Enc(0)

Hardcore Measures

[Imp95, MT10]

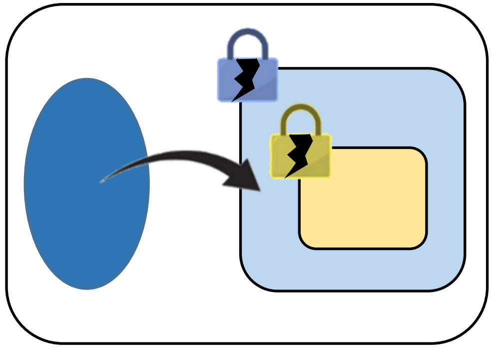
Weak indistinguishability over uniform randomness → Strong indistinguishability over hardcore measures

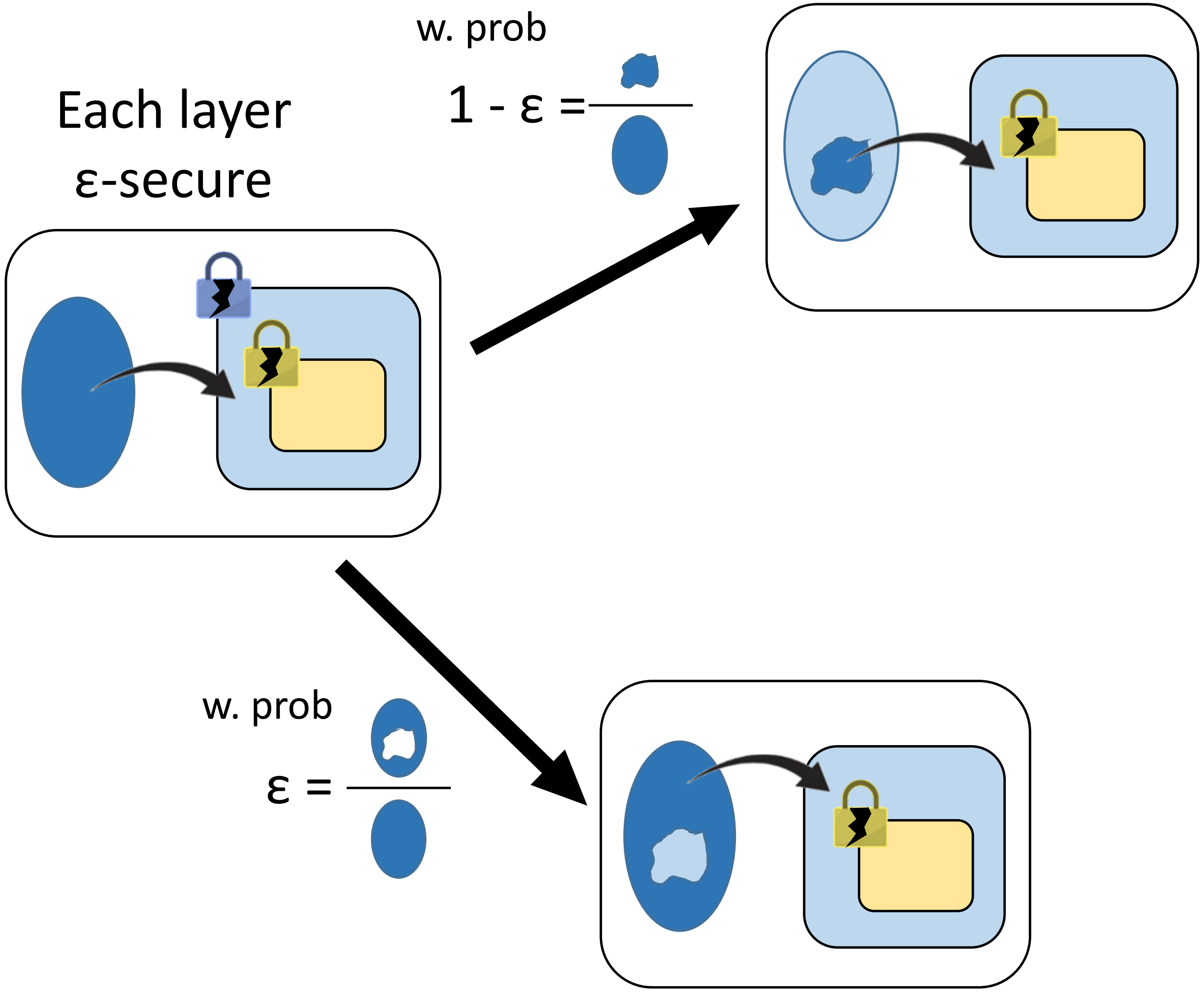


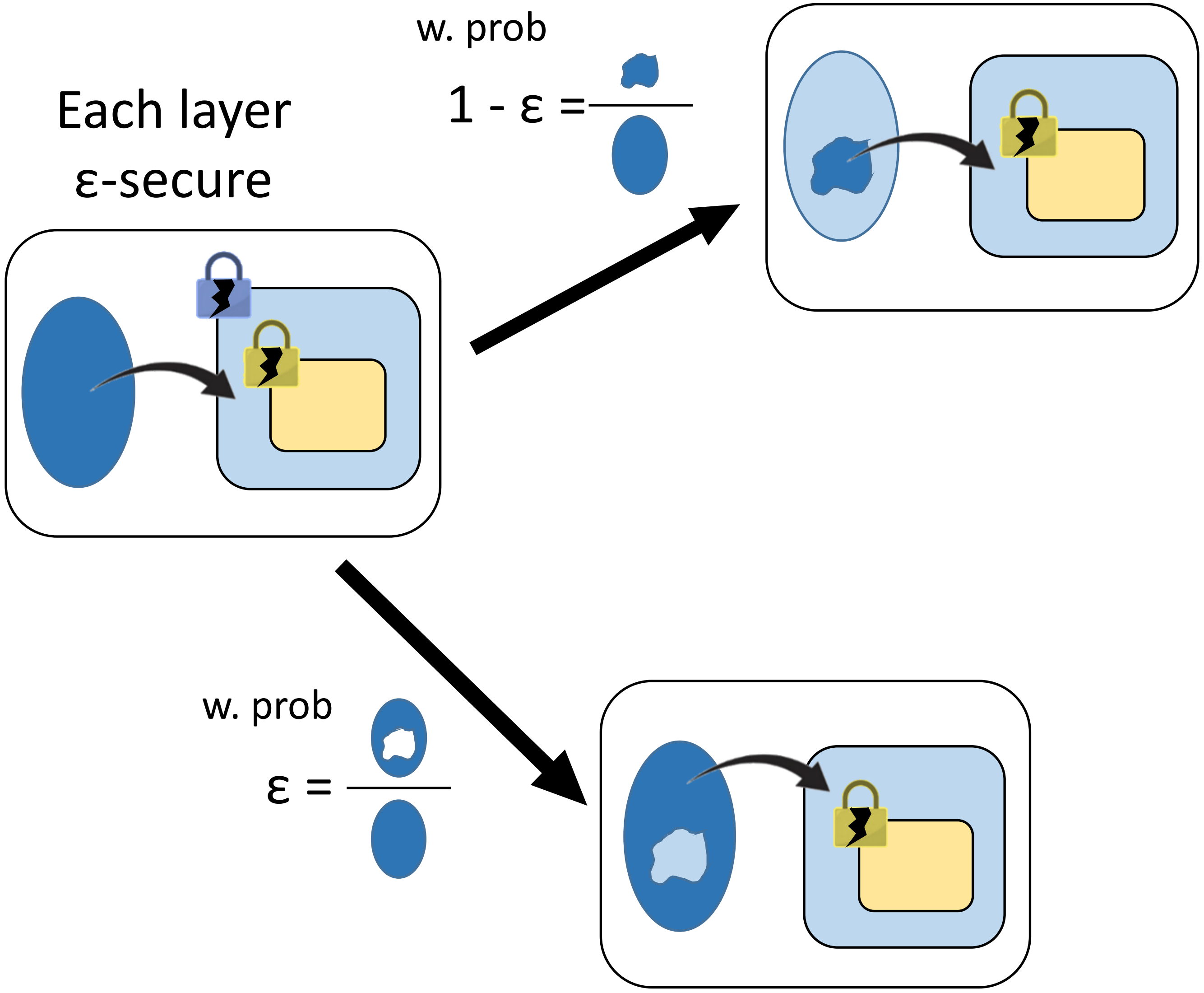
Overall Idea: Strong security when sampling from hardcore measure

ε - distinguishable → density( ) = — = density( ) = 1 - ε

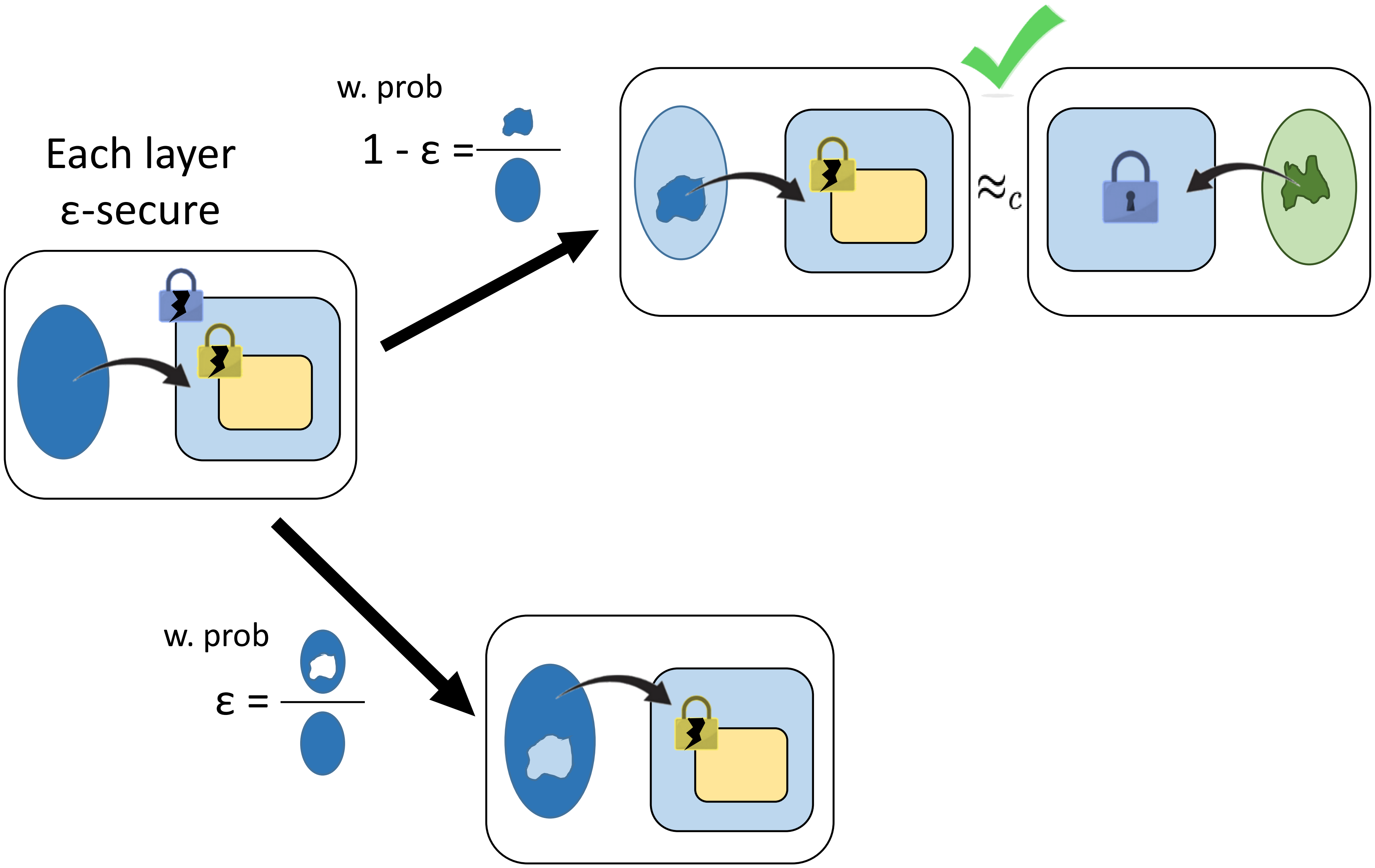
Each layer
ε-secure



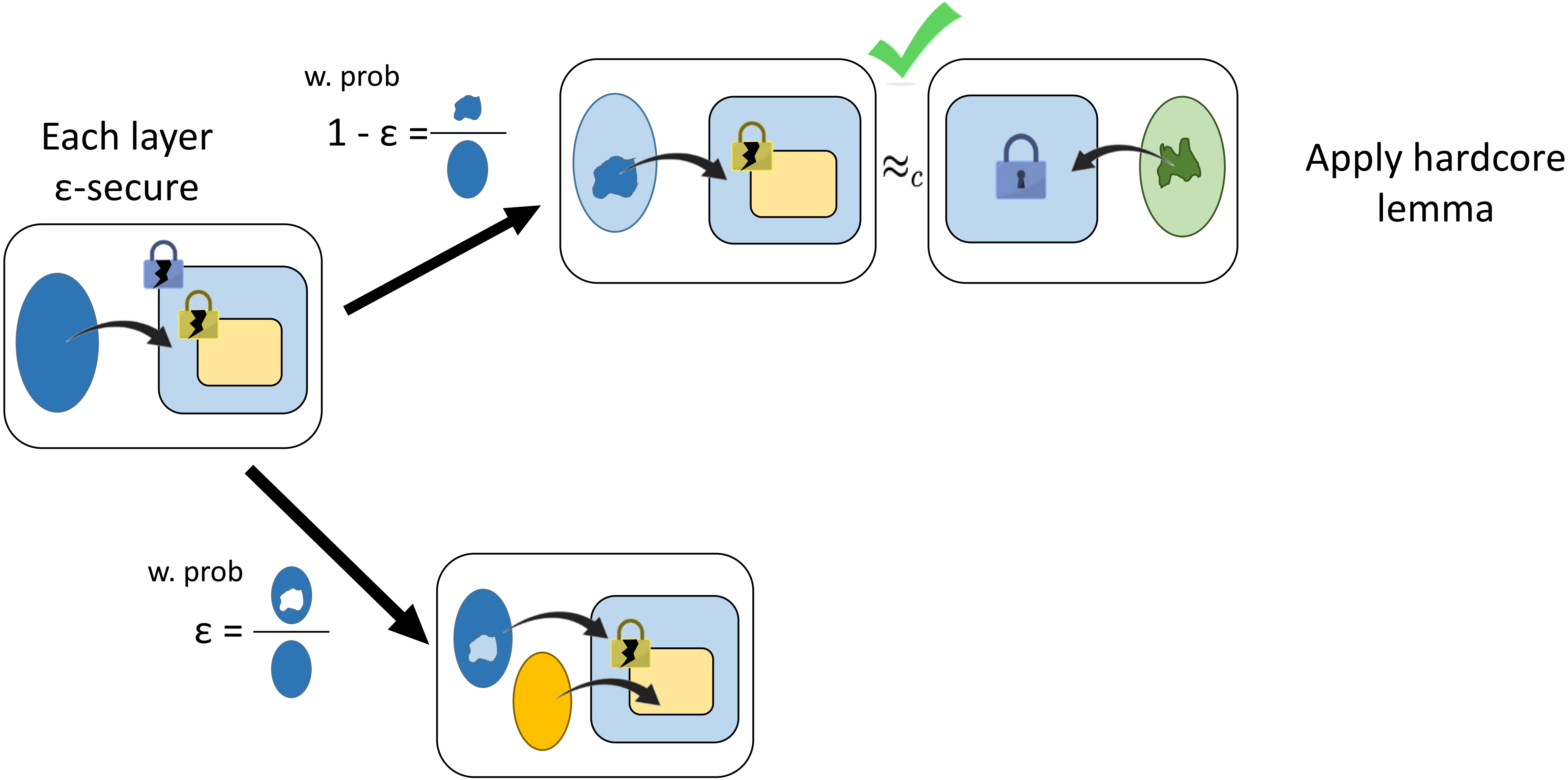

Each layer
ε-secure
w. prob
1 - ε =
w. prob
ε =



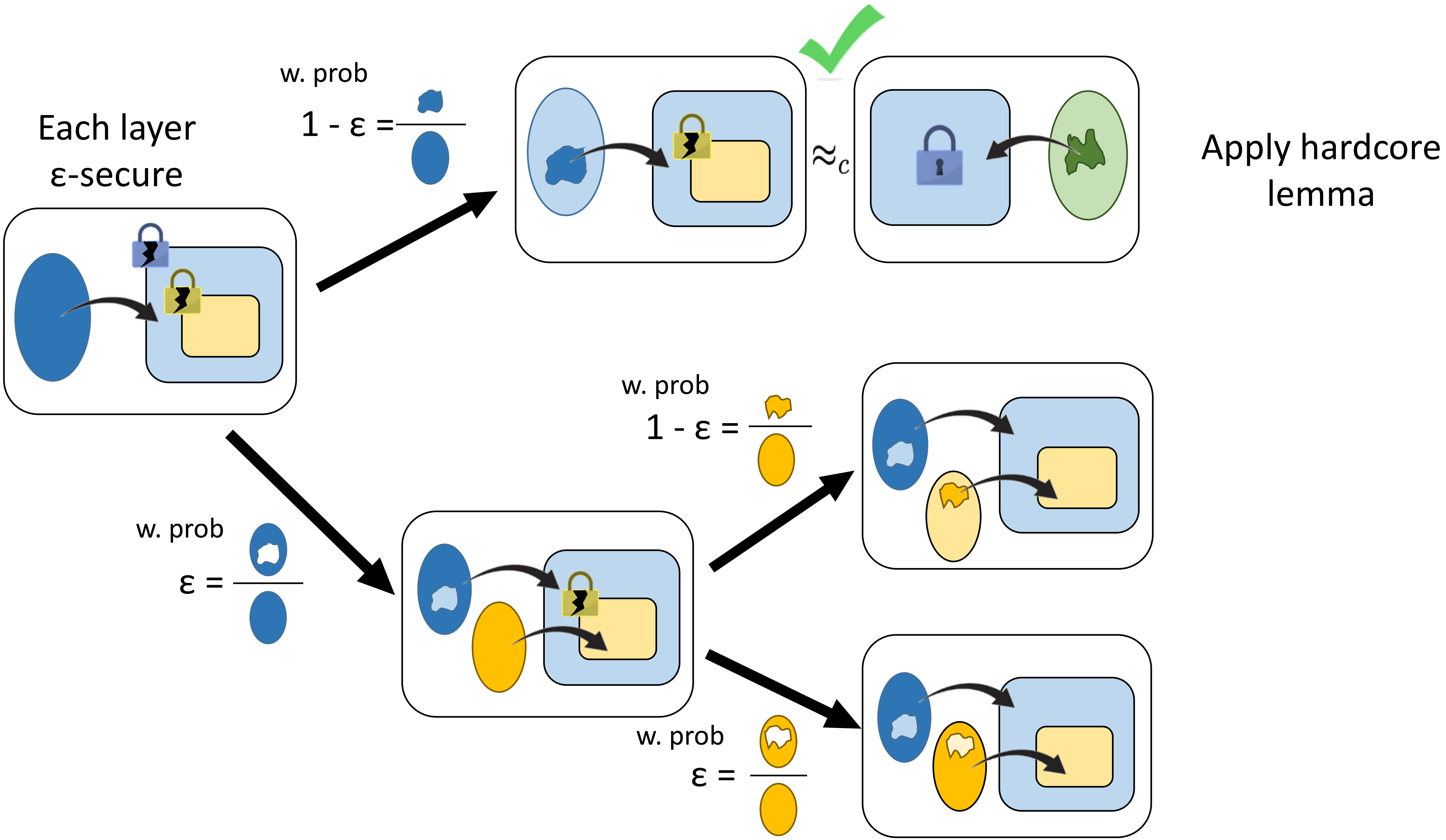
Sample from hardcore measure
→ Expect strong security



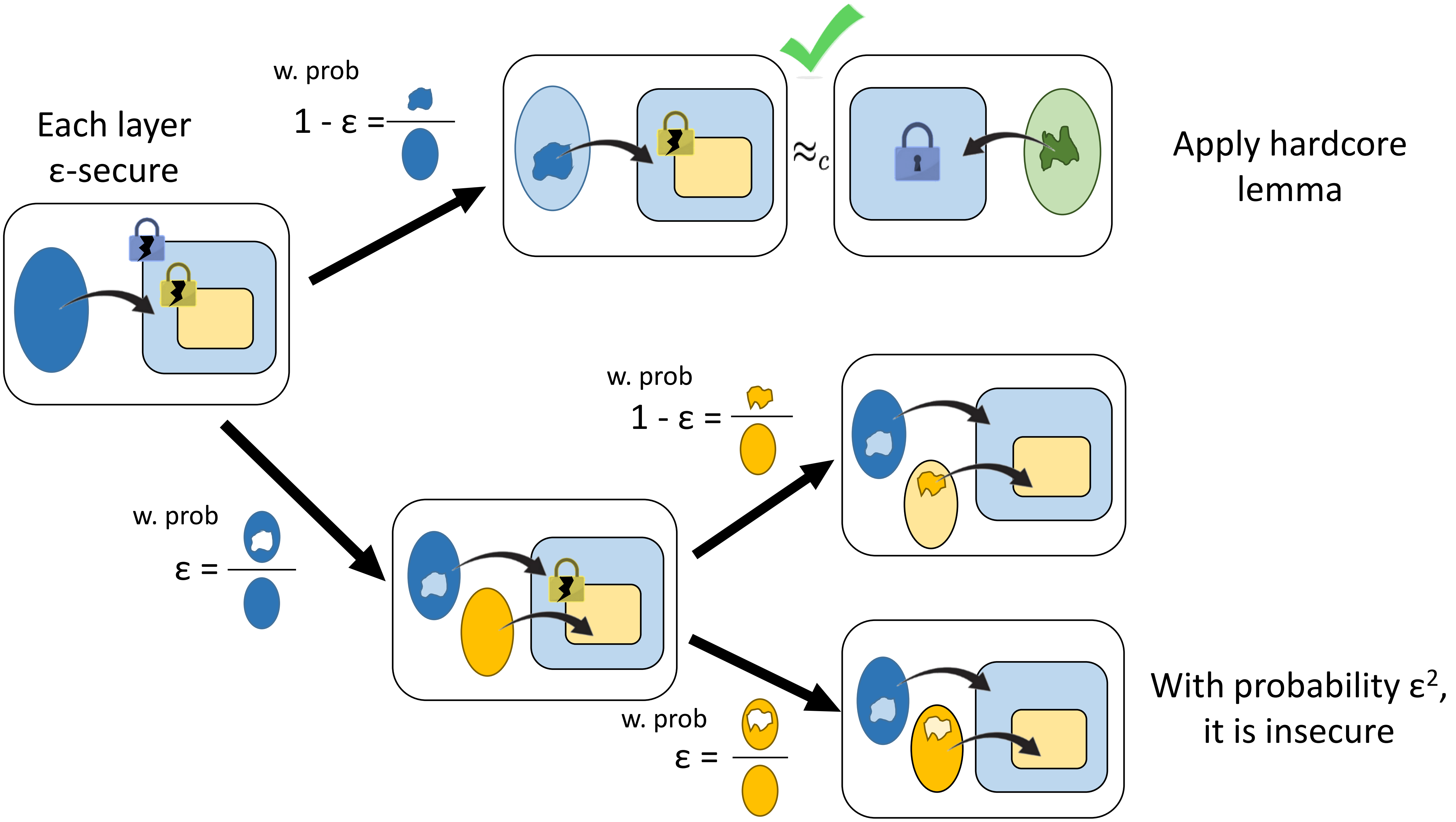
Apply hardcore lemma

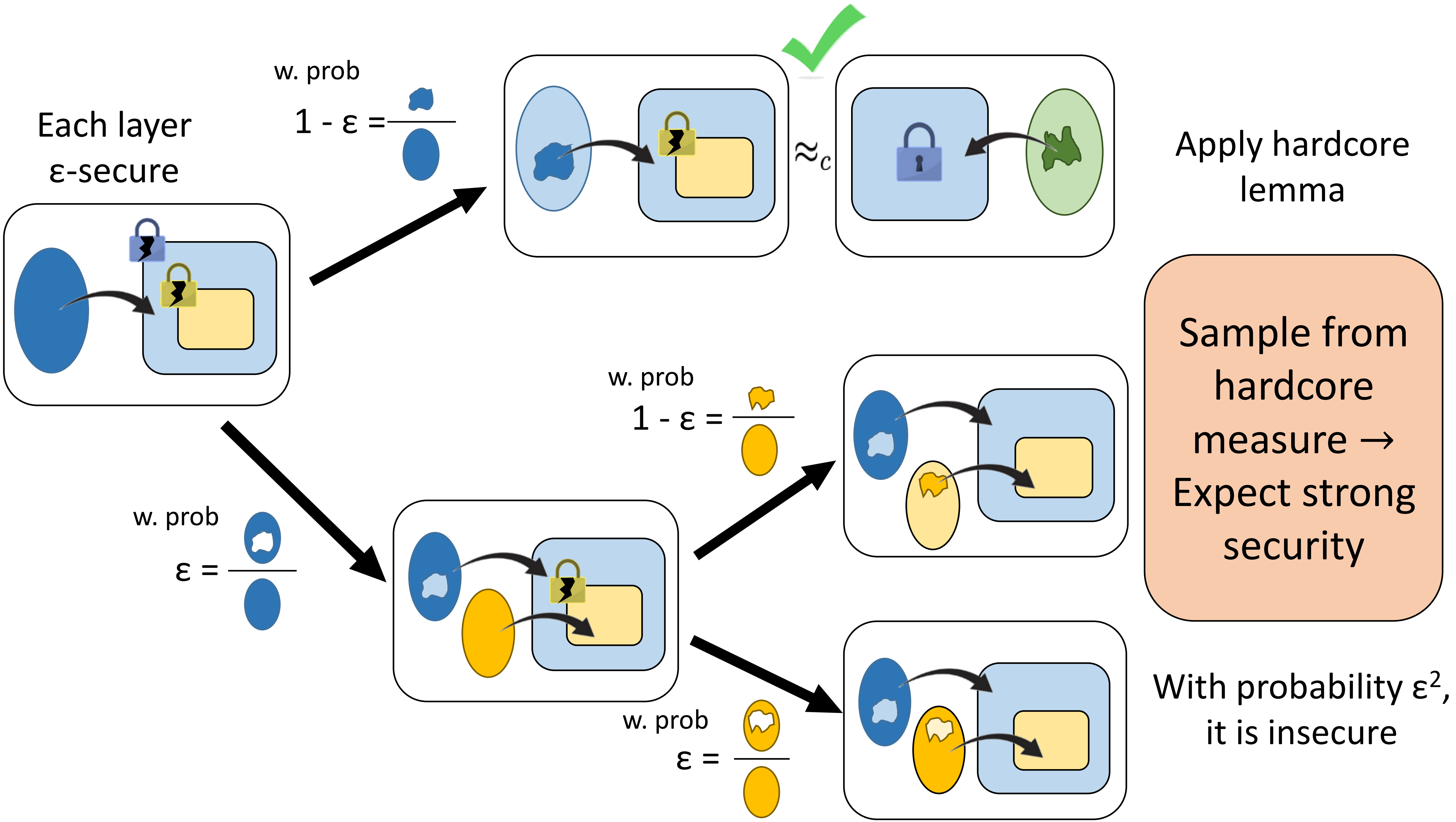


Each layer
ε-secure
w. prob
1 - ε =
≈c
Apply hardcore
lemma
w. prob
ε =

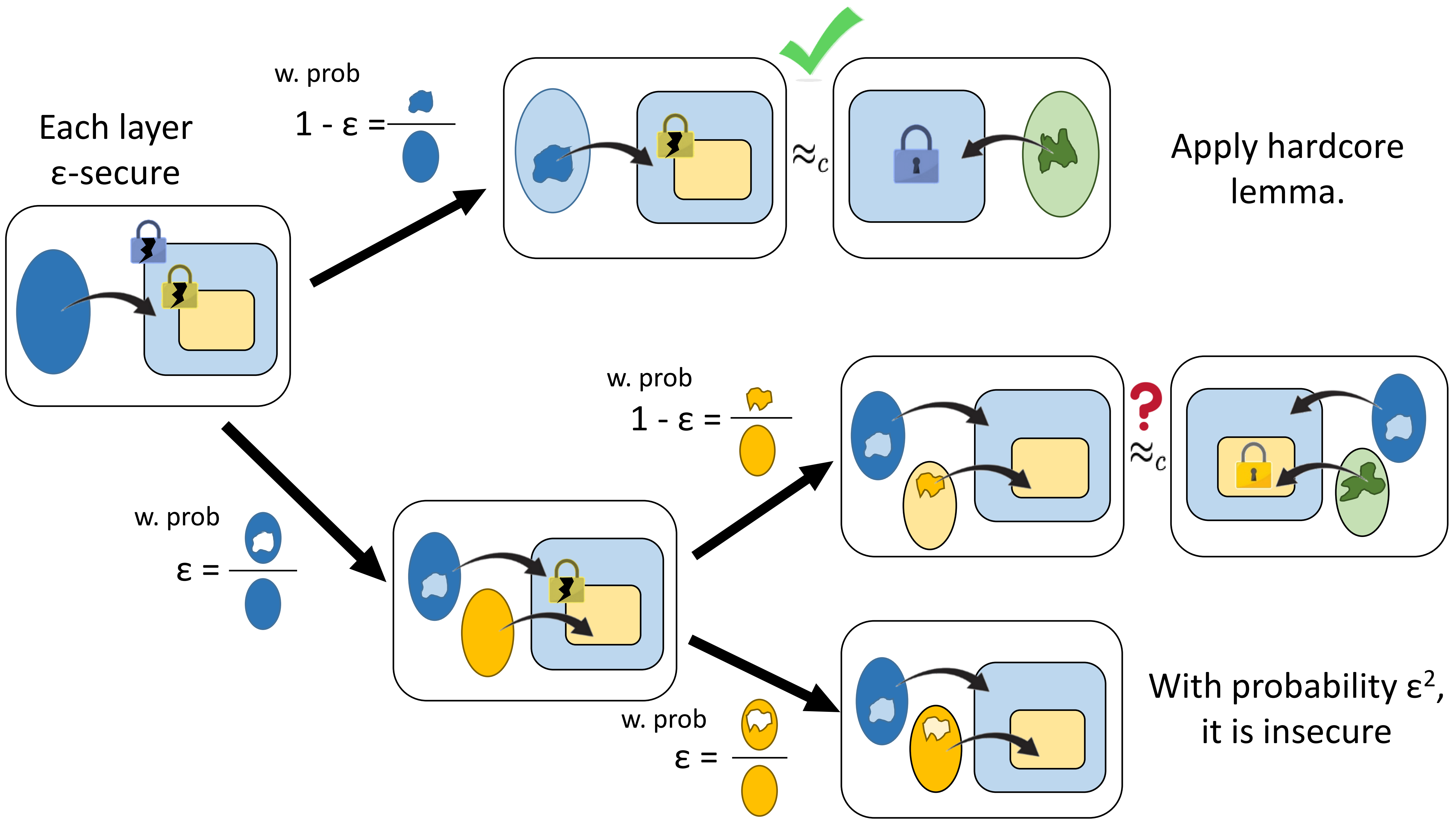


Each layer
ε-secure
w. prob
1 - ε =
≈c
Apply hardcore
lemma
w. prob
1 - ε =
w. prob
ε =
w. prob
ε =

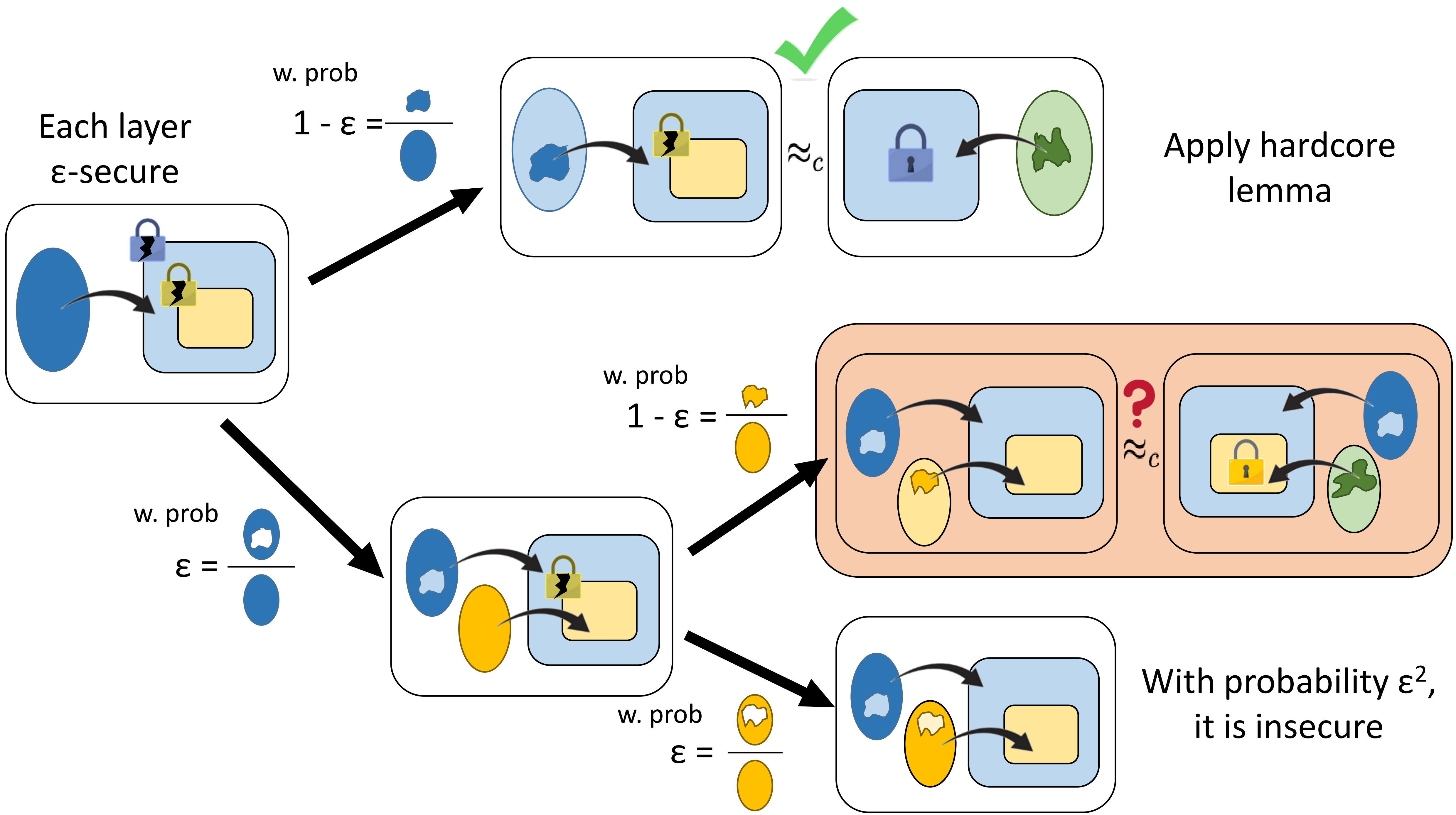

Each layer
ε-secure
w. prob
1 - ε =
$\approx_c$
Apply hardcore
lemma
w. prob
1 - ε =
w. prob
ε =
w. prob
ε =
With probability ε^2,
it is insecure



Each layer
ε-secure
w. prob
1 - ε =
≈c
Apply hardcore
lemma
Sample from
hardcore
measure →
Expect strong
security
w. prob
1 - ε =
w. prob
ε =
w. prob
ε =
With probability ε²,
it is insecure

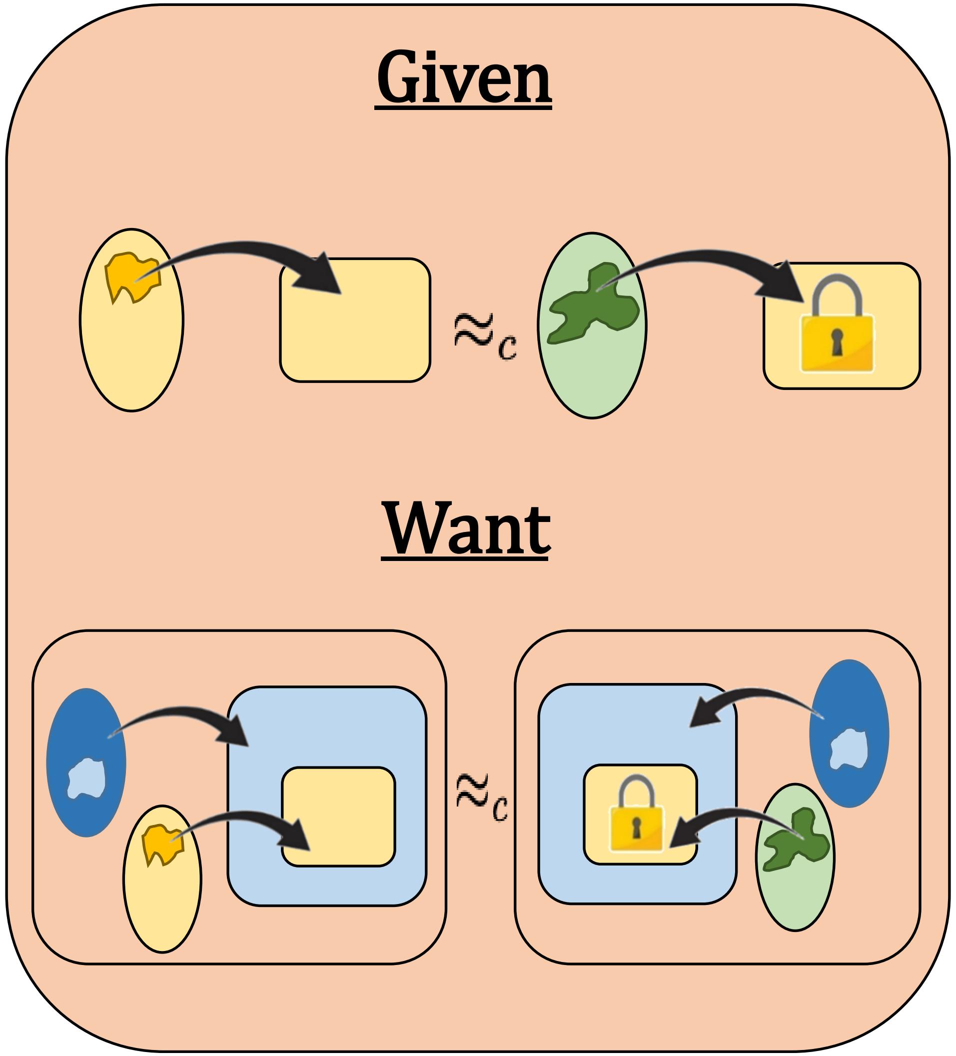


Each layer
ε-secure
w. prob
1 - ε =
Apply hardcore
lemma.
w. prob
1 - ε =
w. prob
ε =
w. prob
ε =
With probability ε²,
it is insecure



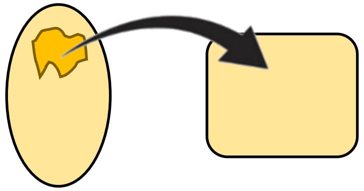
Each layer
ε-secure
w. prob
1 - ε =
≈c
Apply hardcore
lemma
w. prob
1 - ε =
≈c
w. prob
ε =
w. prob
ε =
With probability ε2,
it is insecure

Reduction First Attempt

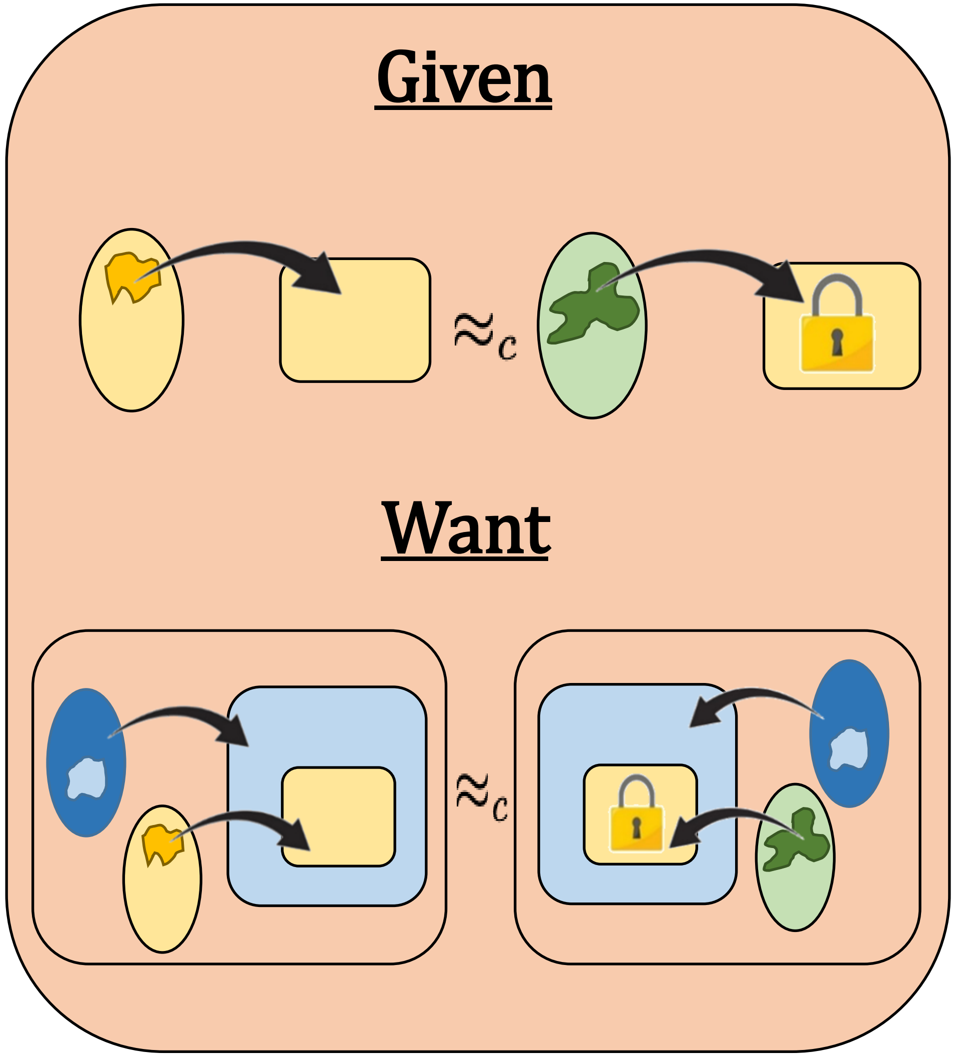


Reduction First Attempt

1. Receive  which is either

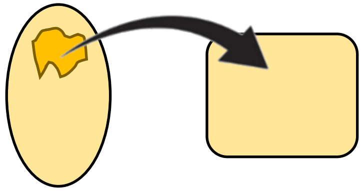


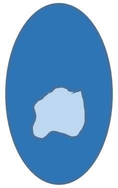
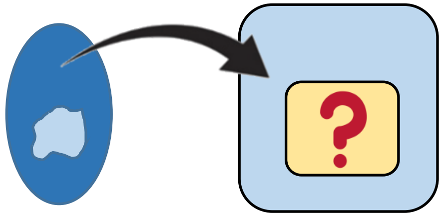
or



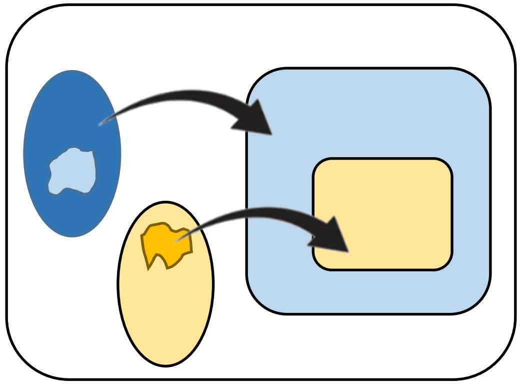
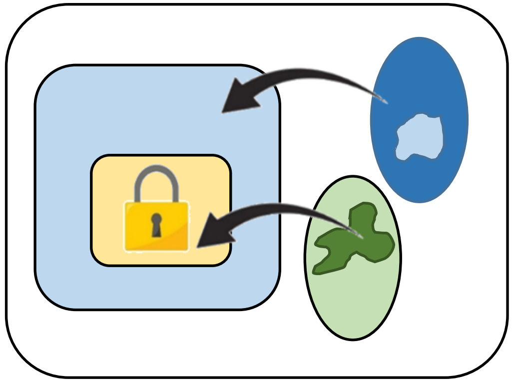
Reduction First Attempt

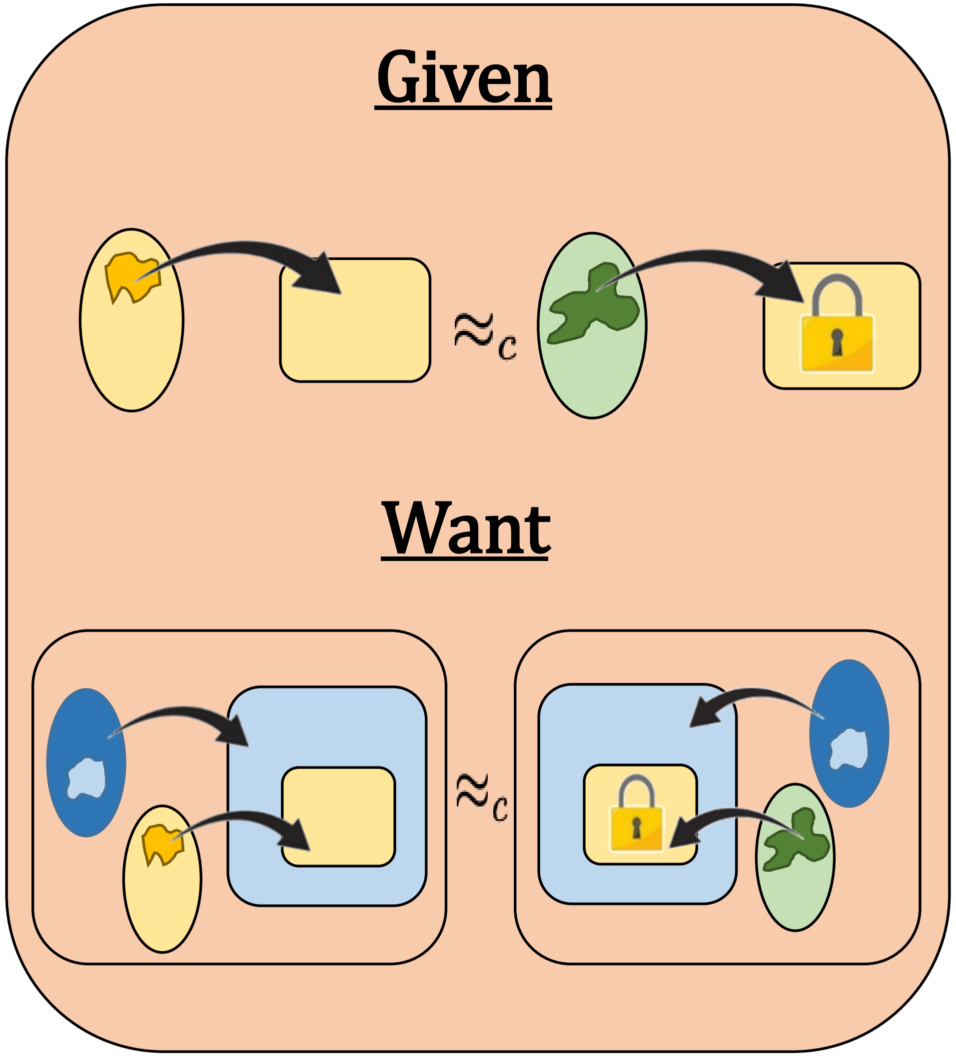
1. Receive  which is either

 or 

2. Sample from  to compute 

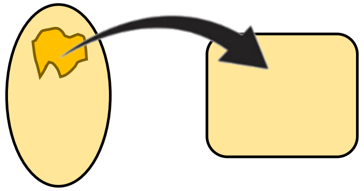
to get either

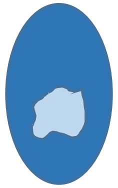
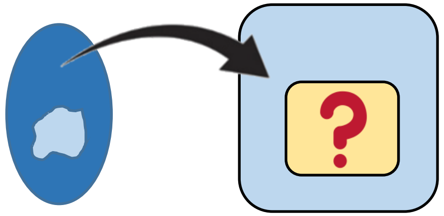
 or 



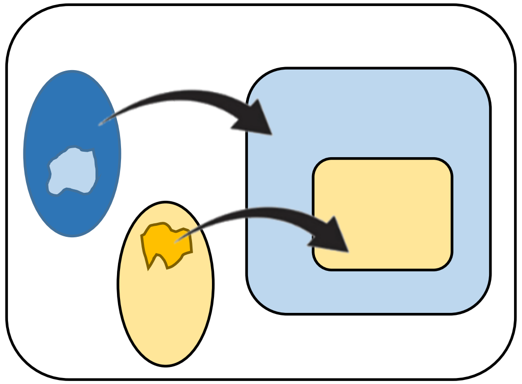
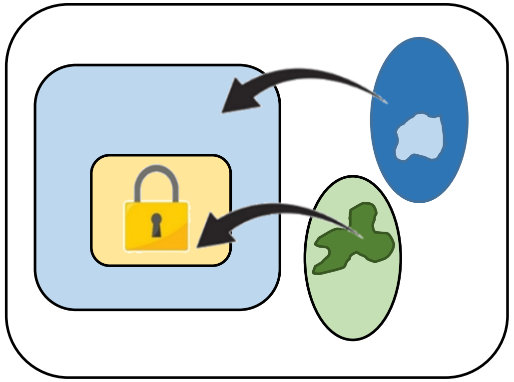
Reduction First Attempt

1. Receive  which is either

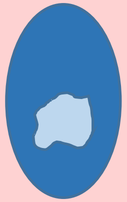
 or 

2. <u>Sample from</u>  to compute 

to get either

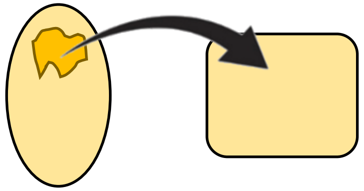
 or 

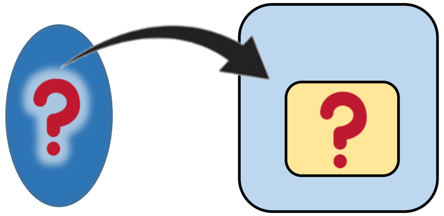
<u>Problems</u>

1.  might not be efficiently samplable or computable.

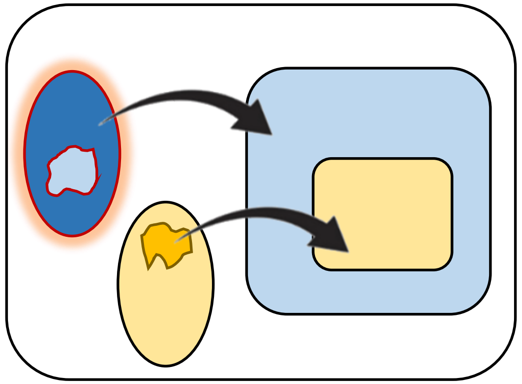
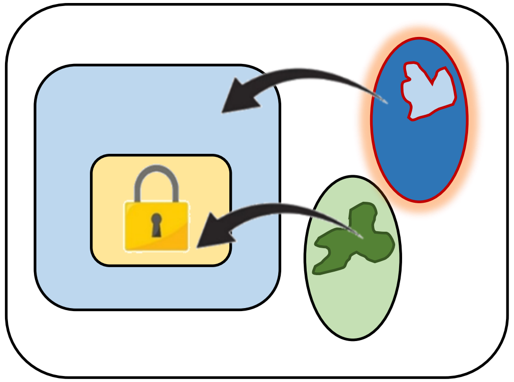
Reduction First Attempt

1. Receive  which is either

 or 

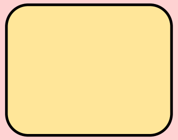
2. Sample from  to compute 

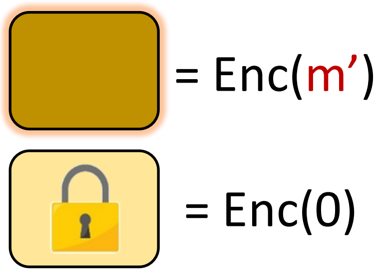
to get either

 or 

Problems

1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have  or 

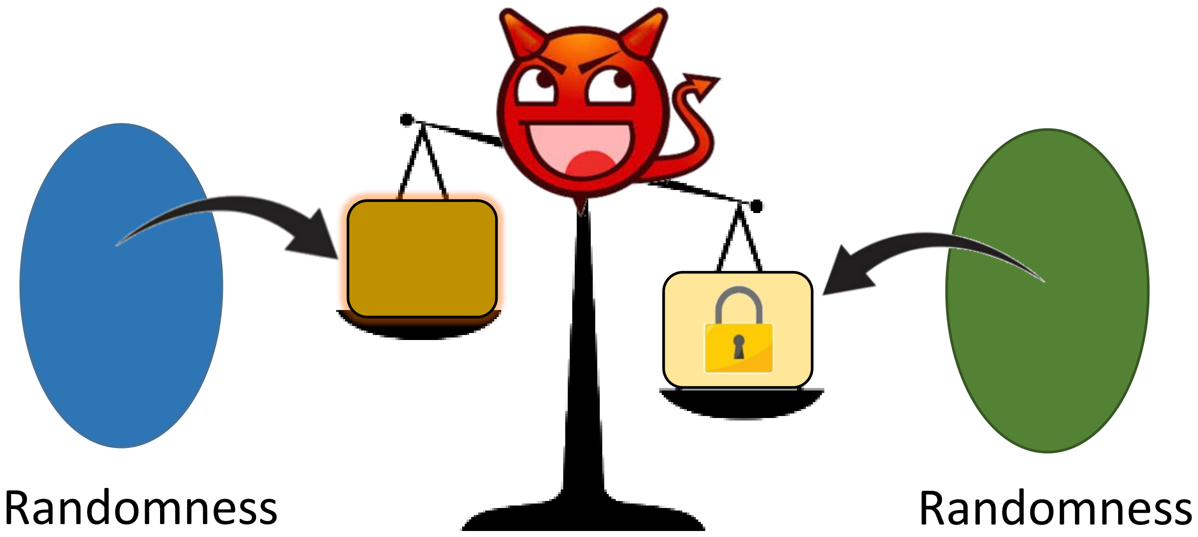


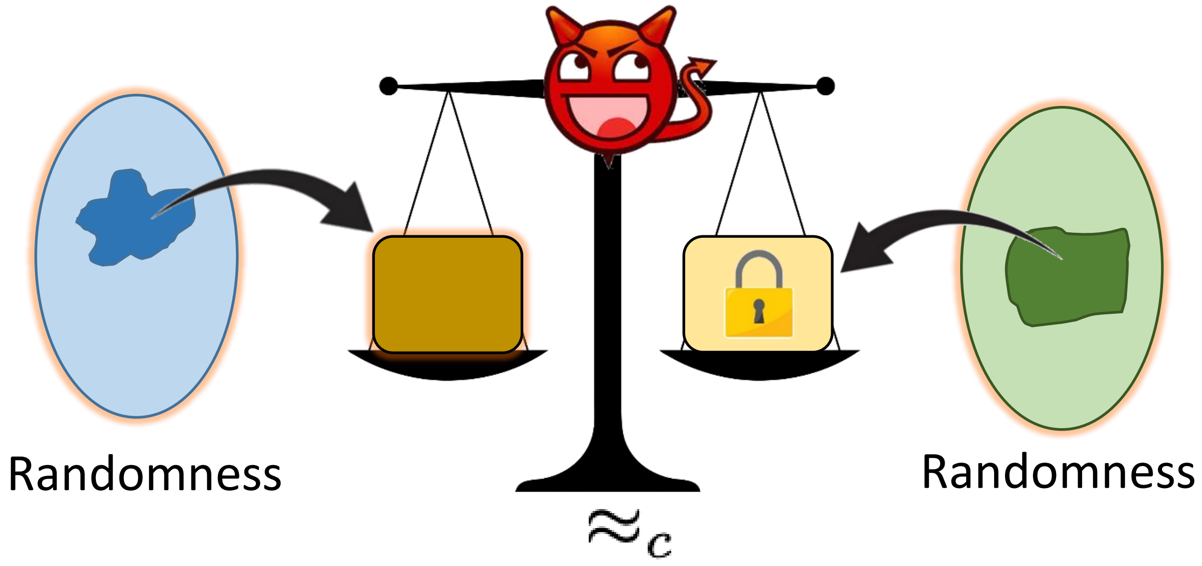
Hardcore Measures

[Imp95, MT10]

Weak indistinguishability over uniform randomness

Hardcore measures depend on the input to the encryption.



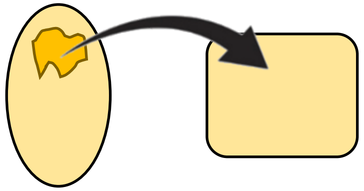


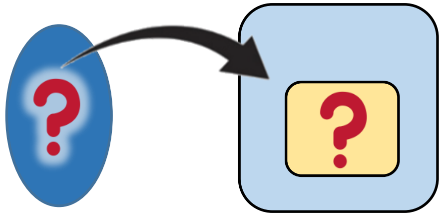
ε - distinguishable

density( ) = —— = density( ) = 1 - ε

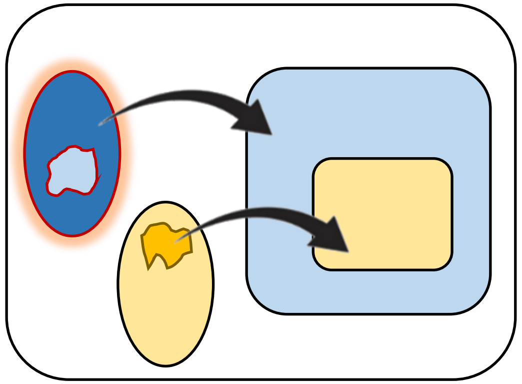
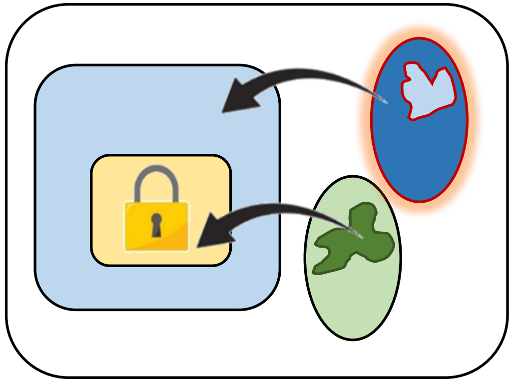
Reduction First Attempt

1. Receive  which is either

 or 

2. <u>Sample from</u>  to compute 

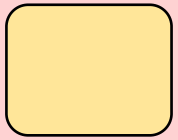
to get either

 or 

<u>Problems</u>

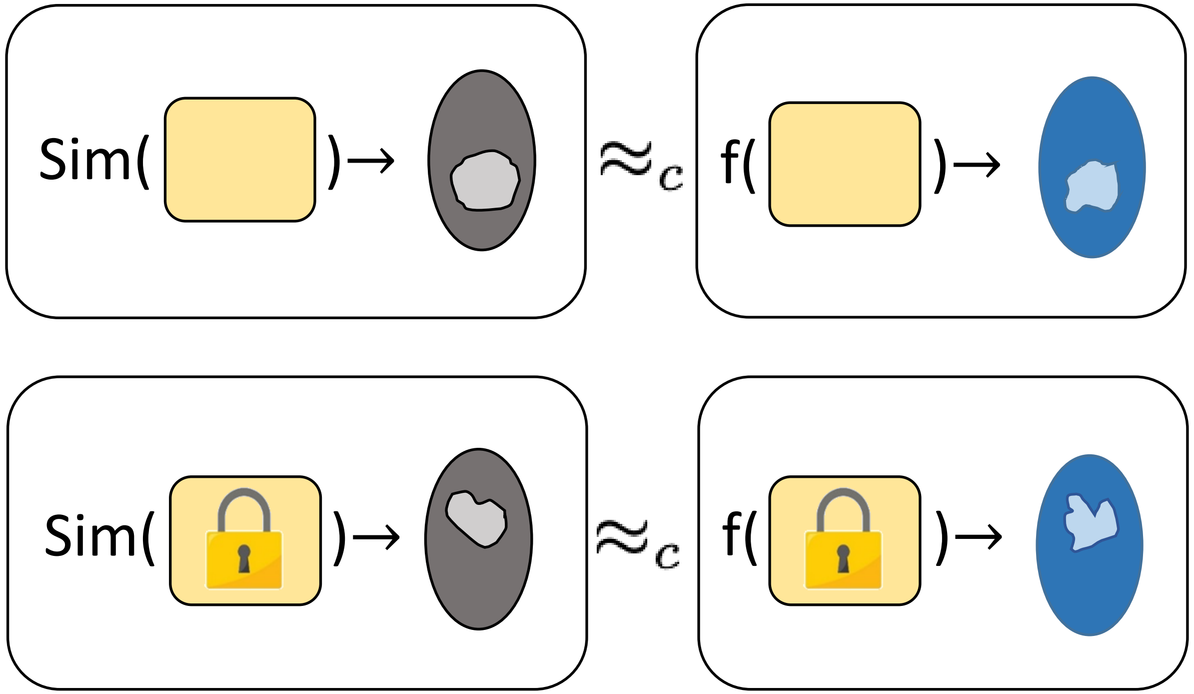
1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have

 or 

Fixing Problem 1: Efficient Simulation

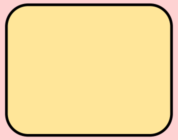
1. [TTV09, Skó15] (informal) Every high density measure can be "efficiently" simulated



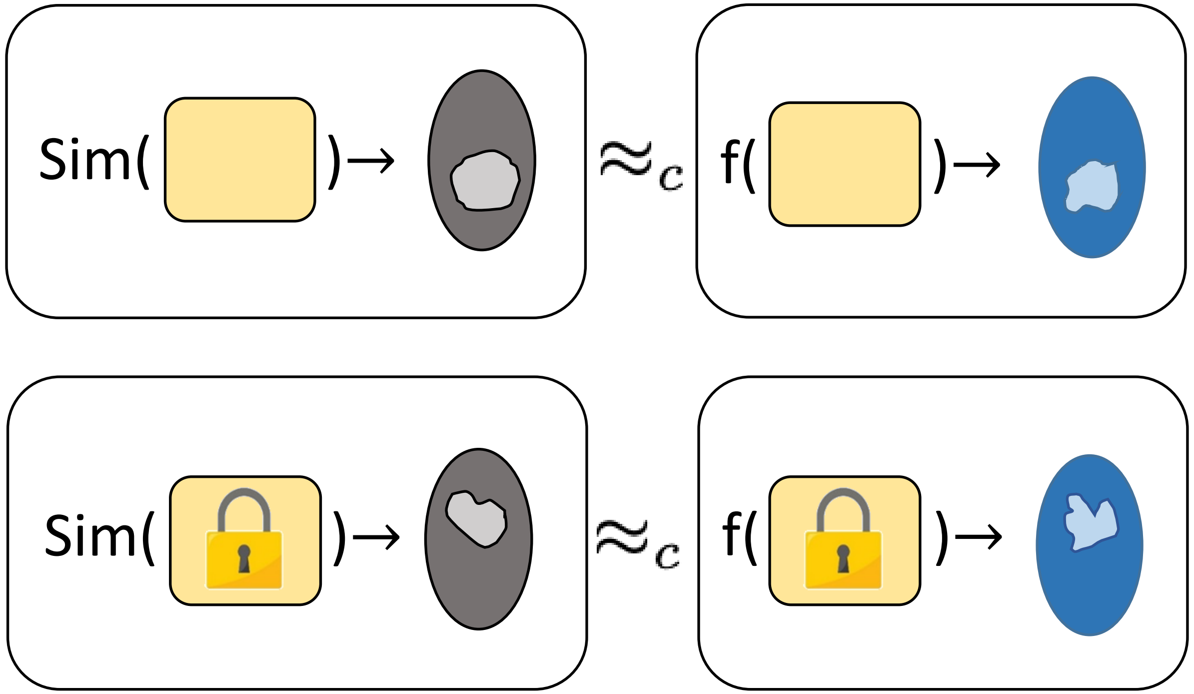
<u>Problems</u>

1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have

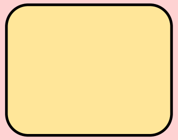
 or 

Fixing Problem 2: Independence from Input



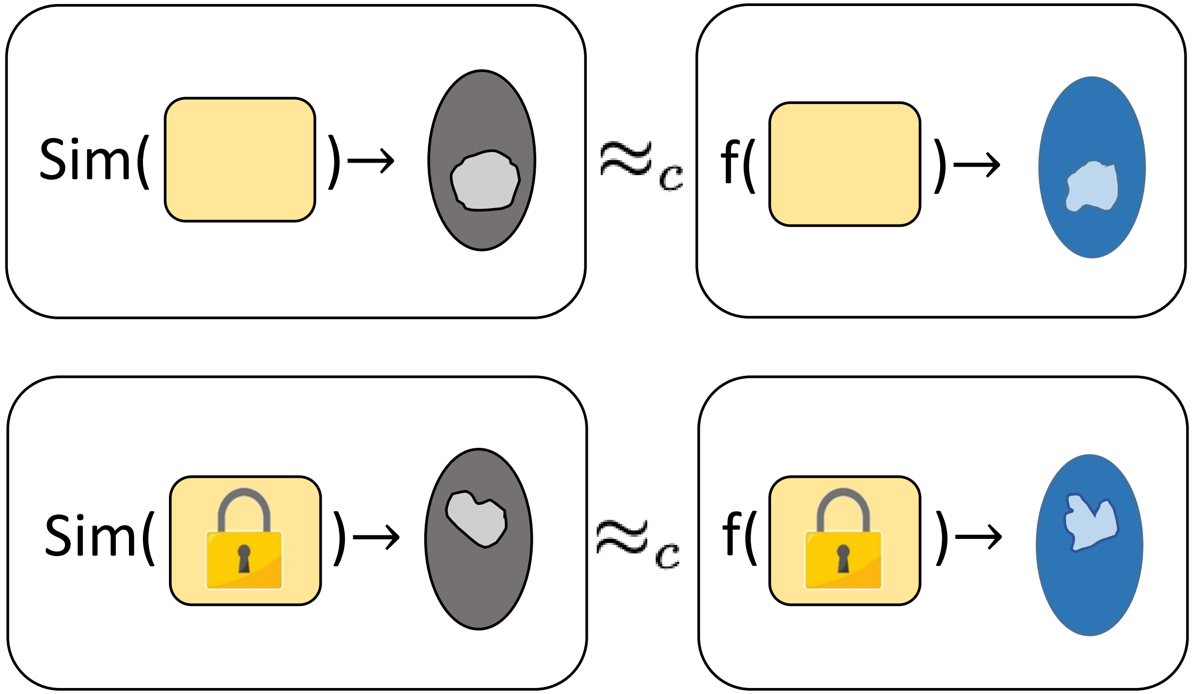
Problems

1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have  or 

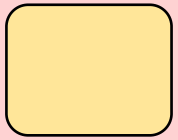
Fixing Problem 2: Independence from Input

Key Observation: Efficiency of simulator is only dependent on the output of f



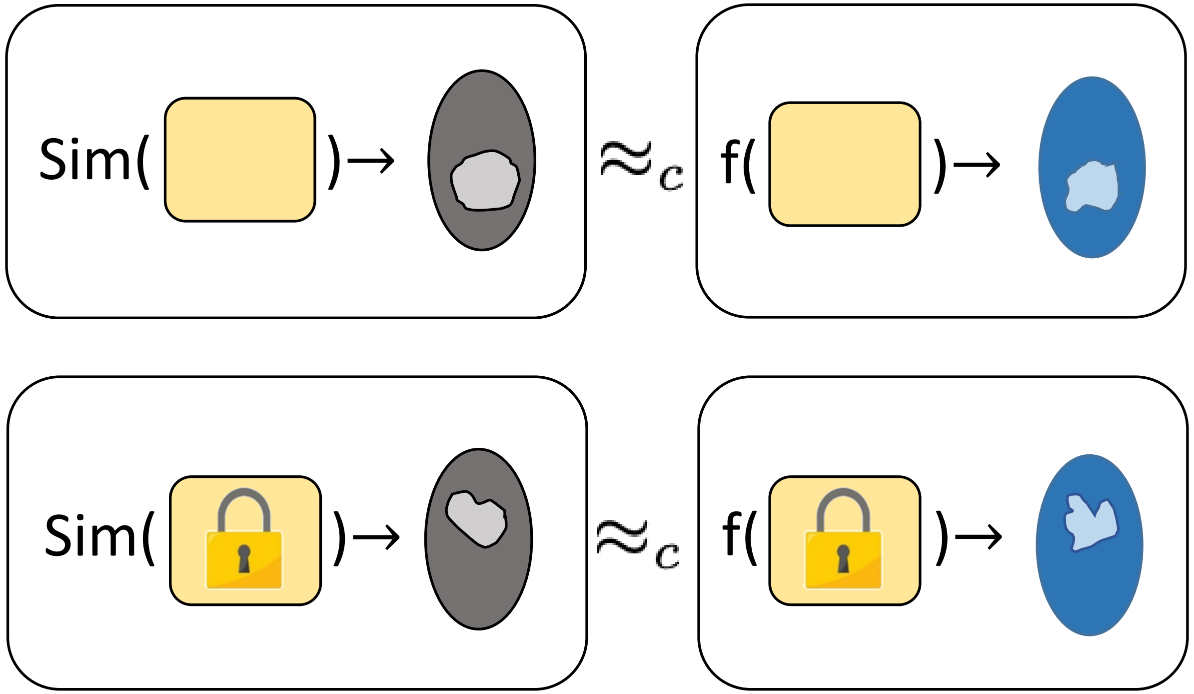
Problems

1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have  or 

Fixing Problem 2: Independence from Input

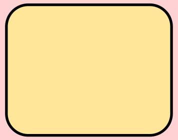
2. Use commitment of hidden information.



<u>Problems</u>

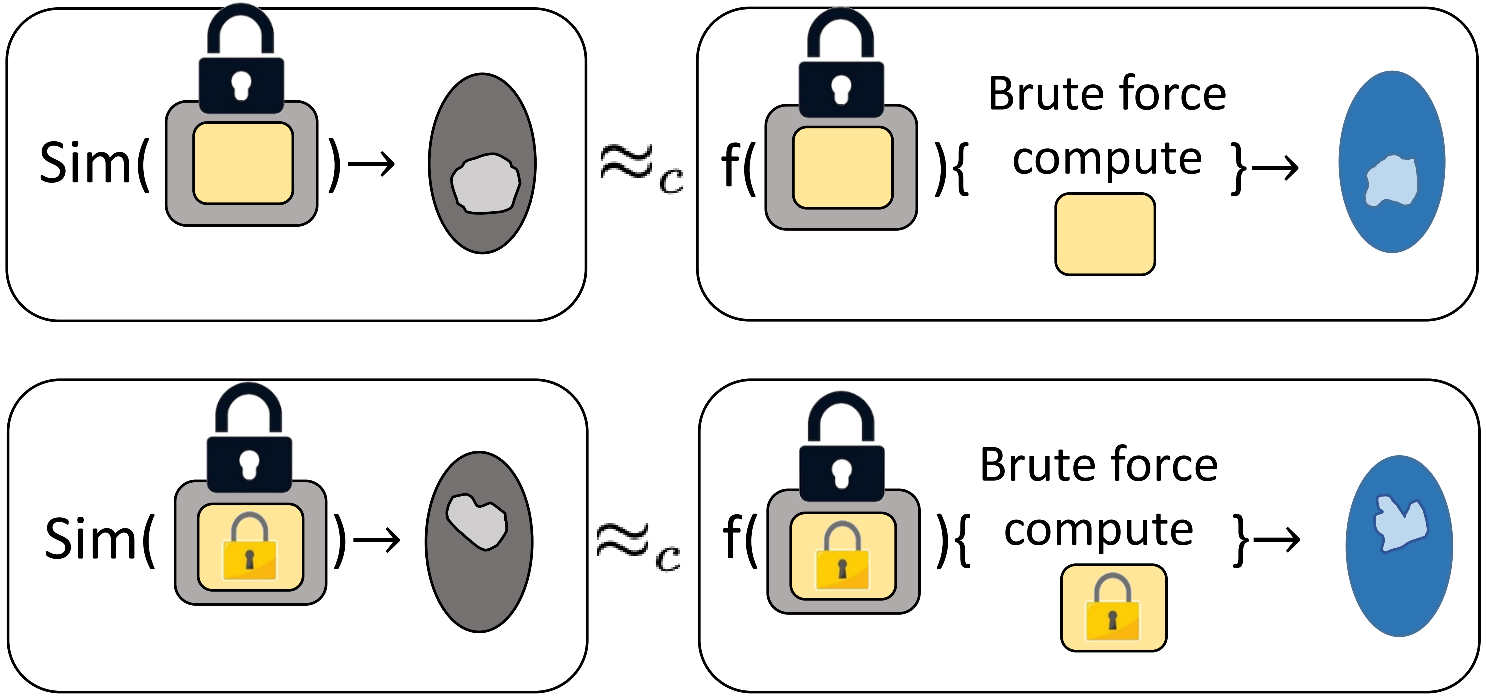
1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have

 or 

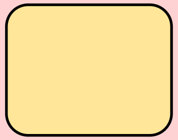
Fixing Problem 2: Independence from Input

2. Use commitment of hidden information.



Problems

1.  might not be efficiently samplable or computable.

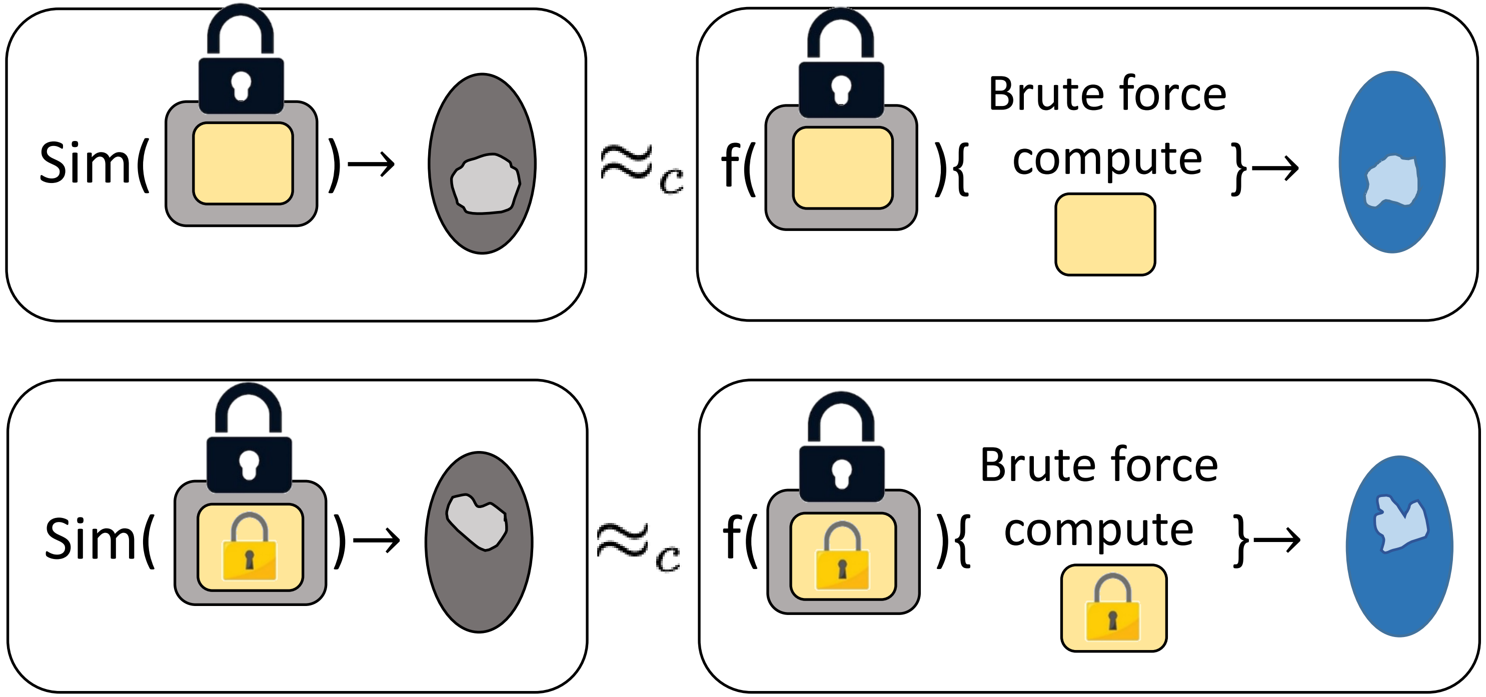
2. Hardcore measure depends on whether we have  or 

Fixing Problem 2: Independence from Input

Problems

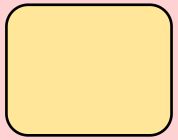
2. Use commitment of hidden information.

Sim just as efficient!



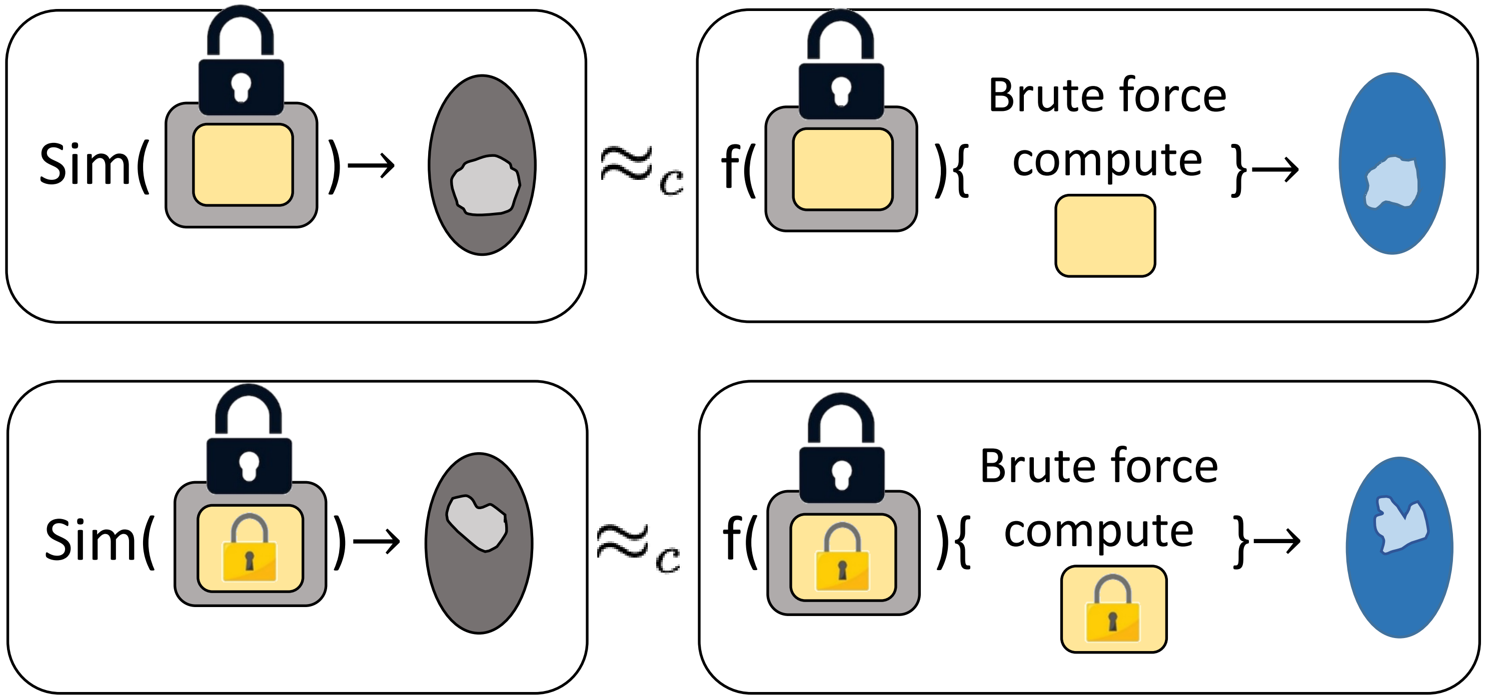
1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have

 or 

Fixing Problem 2: Independence from Input

2. Use commitment of hidden information.



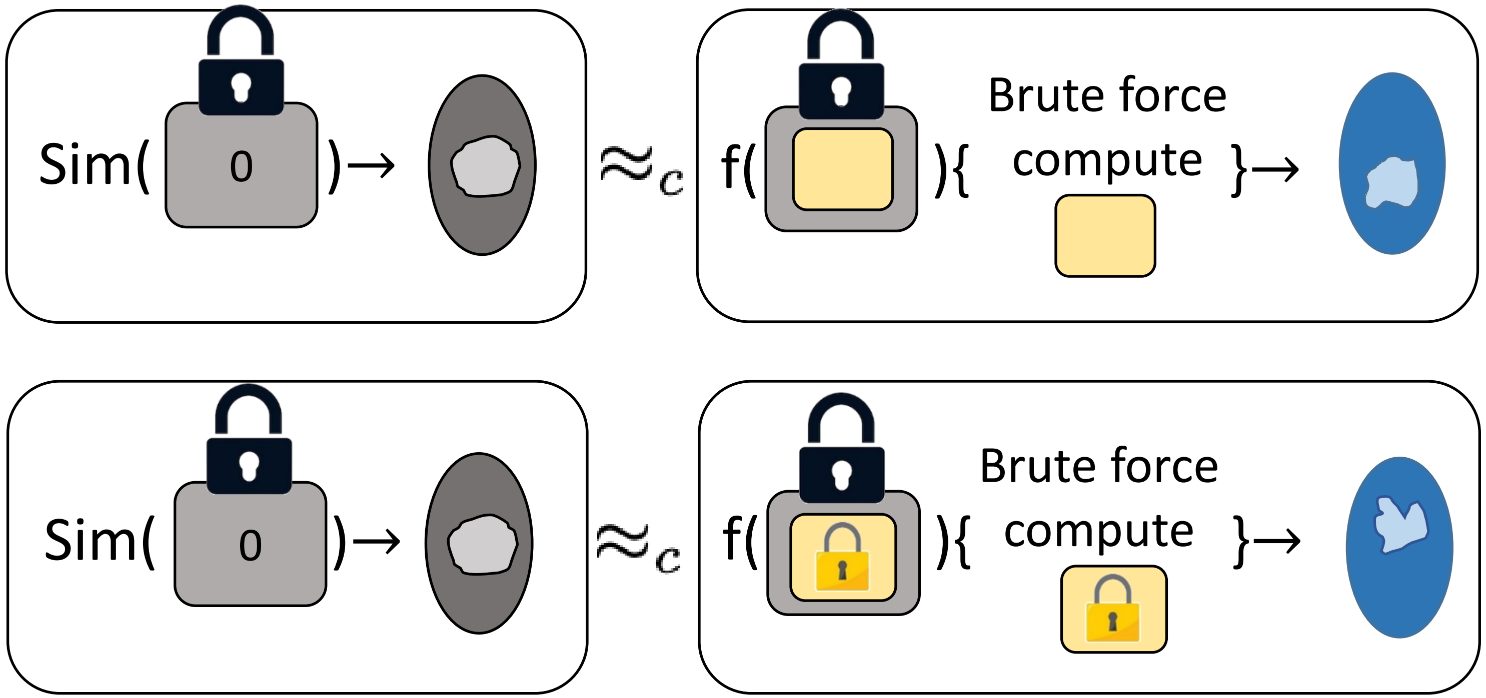
Problems

1. might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have or

Fixing Problem 2: Independence from Input

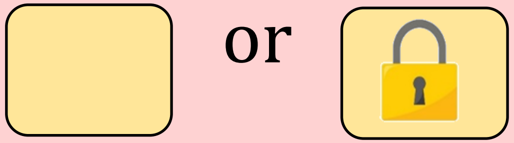
2. Use commitment of hidden information.
   Change commitment to zero.



Problems

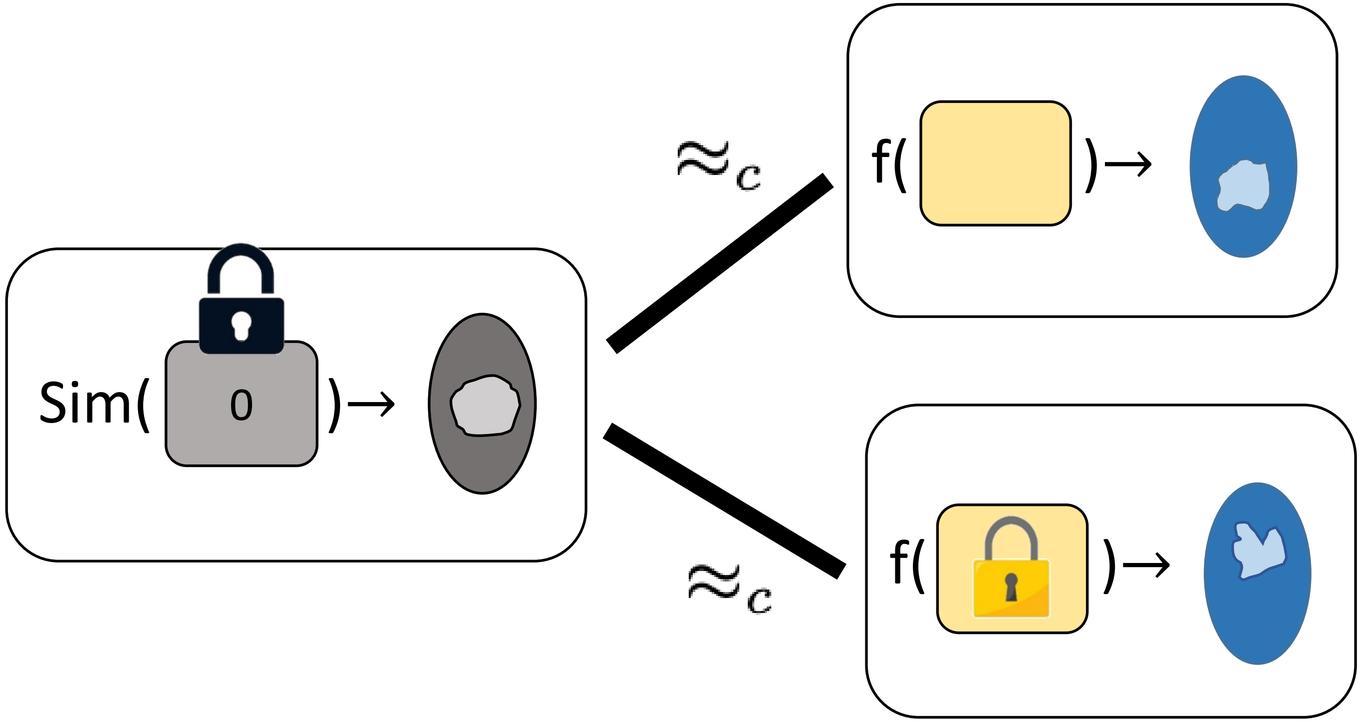
1.  might not be efficiently samplable or computable.

2. Hardcore measure depends on whether we have



Fixing Problem 2: Independence from Input

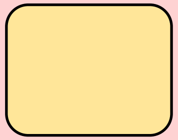
Result: Simulate hardcore measures.



Problems

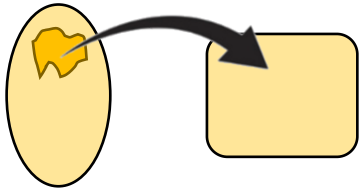
1.  might not be efficiently samplable or computable.

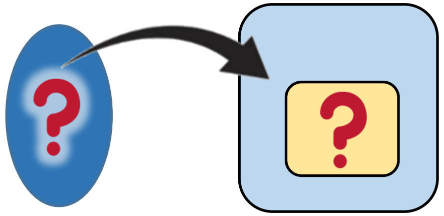
2. Hardcore measure depends on whether we have

 or 

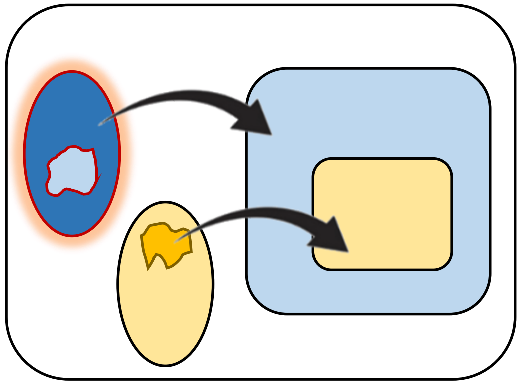
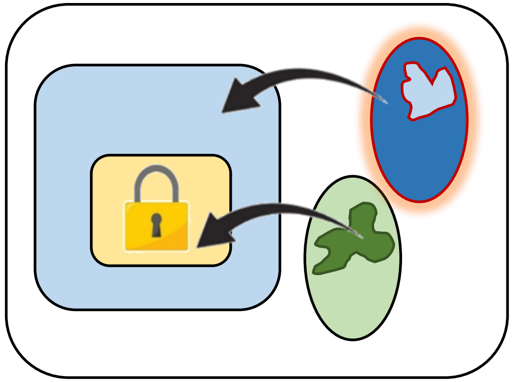
Reduction First Attempt

1. Receive  which is either

 or 

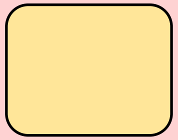
2. Sample from  to compute 

to get either

 or 

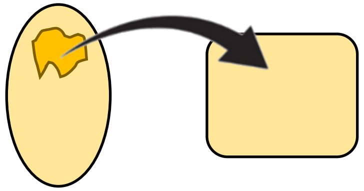
Problems

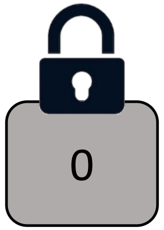
1.  might not be efficiently samplable or computable.

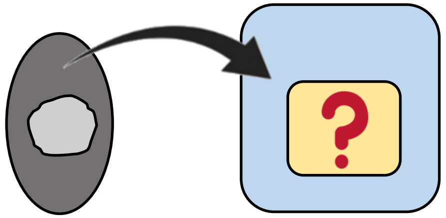
2. Hardcore measure depends on whether we have  or 

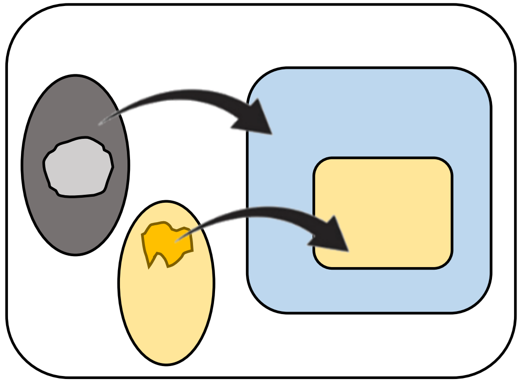
Fixed Reduction

1. Receive  which is either

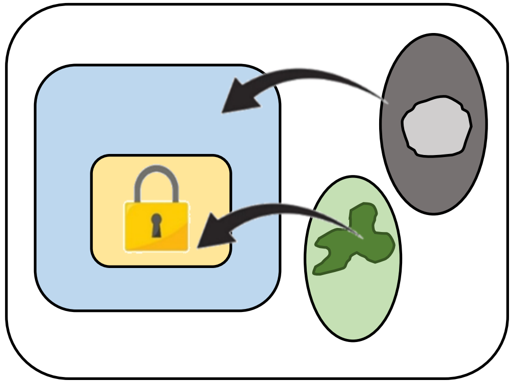
 or 

2. Compute Sim(

)→  to get

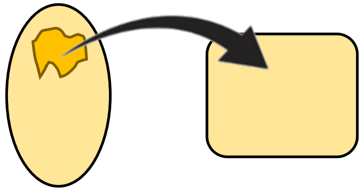


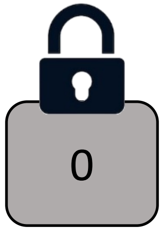
or

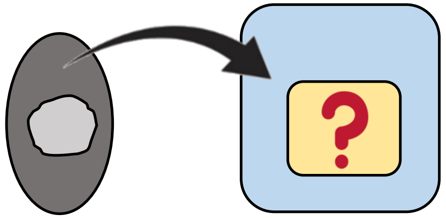


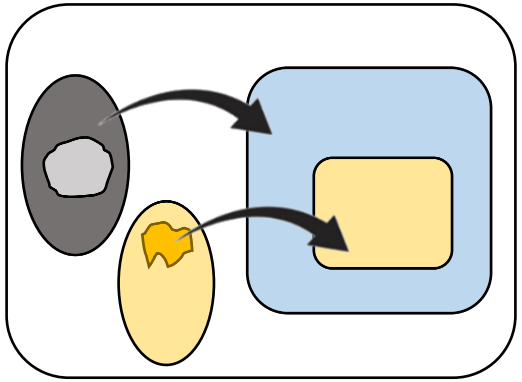
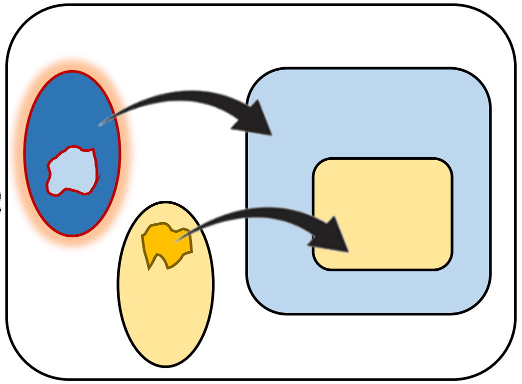
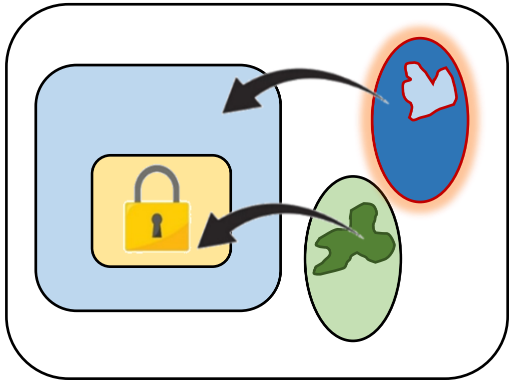
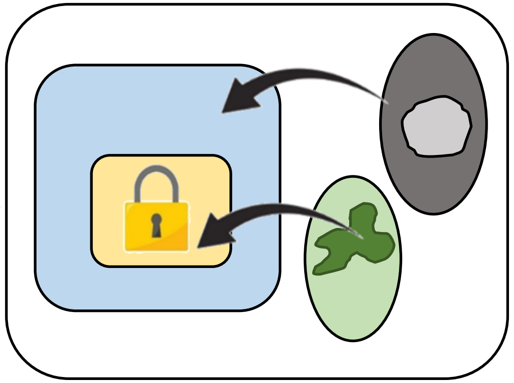
Fixed Reduction

1. Receive  which is either

 or 

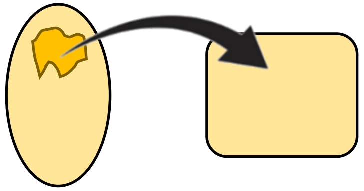
2. Compute Sim(

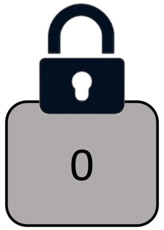
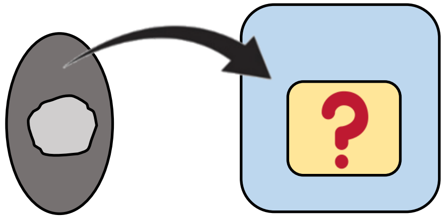
)→  to get

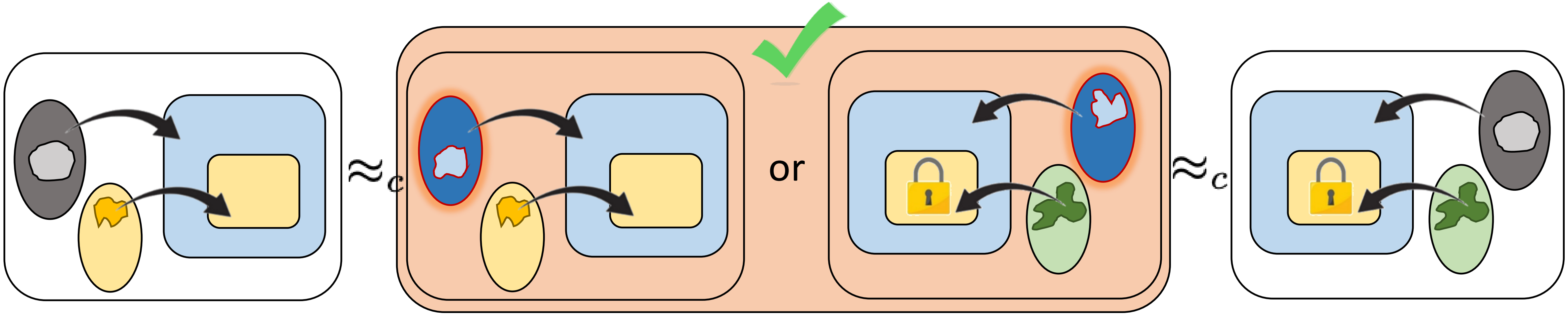
 $\approx_c$  or  $\approx_c$ 

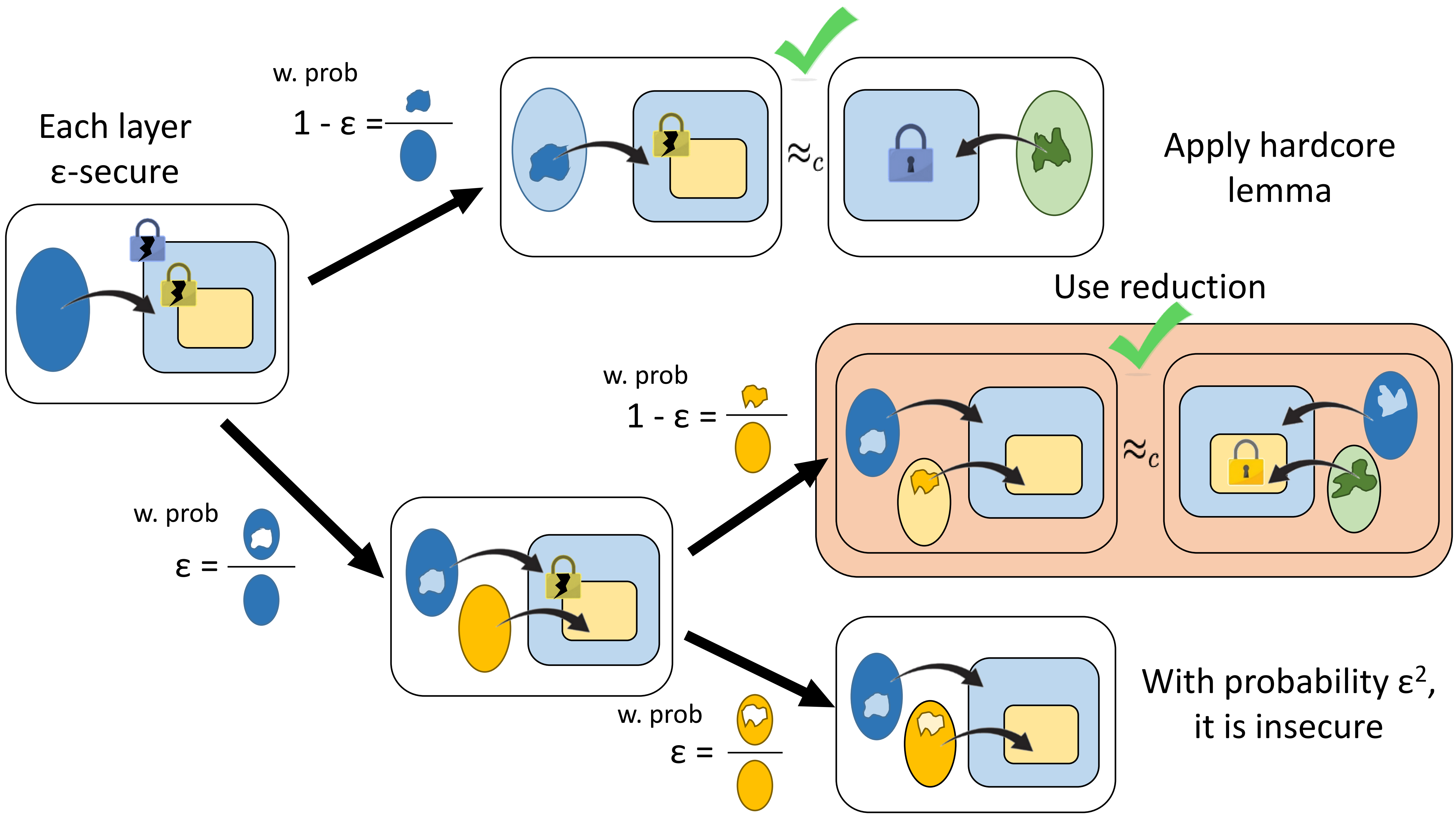
Fixed Reduction

1. Receive  which is either

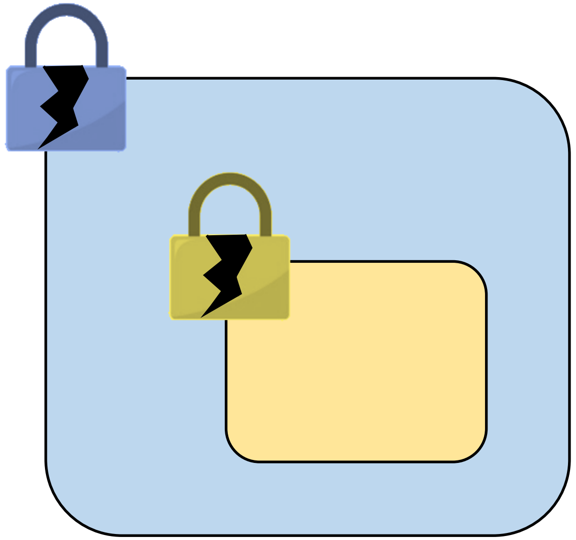
 or 

2. Compute Sim()→  to get




Each layer
ε-secure
w. prob
1 - ε =
Apply hardcore
lemma
Use reduction
w. prob
1 - ε =
w. prob
ε =
w. prob
ε =
With probability ε^2,
it is insecure

Amplification of Nested Primitives



Intuition: If one layer is secure, then the whole thing is secure

Result: Amplify security from $\varepsilon \rightarrow \varepsilon^2 + negl(\lambda)$

Summary

- Amplify FE from ε-security for any constant $\varepsilon \in (0,1)$ to full security, unconditionally.
  - Preserves compactness
- New technique for amplification of nested primitives.
- Introduce set homomorphic secret sharing.

Thank you!