EFFICIENT ADAPTIVELY SECURE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

Chaya Ganesh	Aarhus
Yashvanth Kondi	Northeastern
Arpita Patra	Indian Inst. of Science
Pratik Sarkar	

OUR RESULTS

- We start from [Jawurek-Kerschbaum-Orlandi13] ZK protocol: **improve rounds** and upgrade to **adaptive security** (without erasures) at essentially **zero overhead**
- This gives us the first **practical adaptively secure ZK protocol**

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

$x \quad f_x$

P ***V***

w

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $f_x(w)=1$ if $x \in L$

P

w

V

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $\quad f_x(w)=1$ if $x \in L$

P $\qquad\qquad\qquad\qquad$ ***V***

w

$\longrightarrow$

$\longleftarrow$

$\longrightarrow$

$\longleftarrow$

$\longrightarrow$ $\qquad$ accept/reject

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $\quad f_x(w)=1$ if $x \in L$

P $\qquad\qquad\qquad\qquad$ ***V***

w

accept/reject

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $\quad f_x(w)=1$ if $x \in L$

Sim$_P$ $\qquad\qquad$ V

accept/reject

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $f_x(w)=1$ if $x \in L$

Sim$_{\boldsymbol{P}}$ $\boldsymbol{V}$

accept/reject

$\{\langle \boldsymbol{P}(w), \boldsymbol{V} \rangle(x)\}_{x \in L} \approx_c \{\langle \mathrm{Sim}_{\boldsymbol{P}} \rangle(x)\}_{x \in L}$

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $f_x(w)=1$ if $x \in L$

P Sim_V

w

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $f_x(w)=1$ if $x \in L$

P

w

Sim$_V$

accept/reject

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $f_x(w)=1$ if $x \in L$

P

w

$\text{Sim}_{\boldsymbol{V}}$

accept

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $\quad f_x(w)=1$ if $x \in L$

P

w

Sim***V***

accept

w

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x $\quad f_x(w)=1$ if $x \in L$

P

Sim$_{\boldsymbol{V}}$

w

accept

$f_x(w)=1$

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

- Previous definition models static corruptions

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

- Previous definition models static corruptions
- Adaptive corruptions closer to real world; systems 'hacked' dynamically

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

- Previous definition models static corruptions
- Adaptive corruptions closer to real world; systems ‘hacked’ dynamically
- Adaptive ZK composed to build larger protocols resilient to dynamic corruptions

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

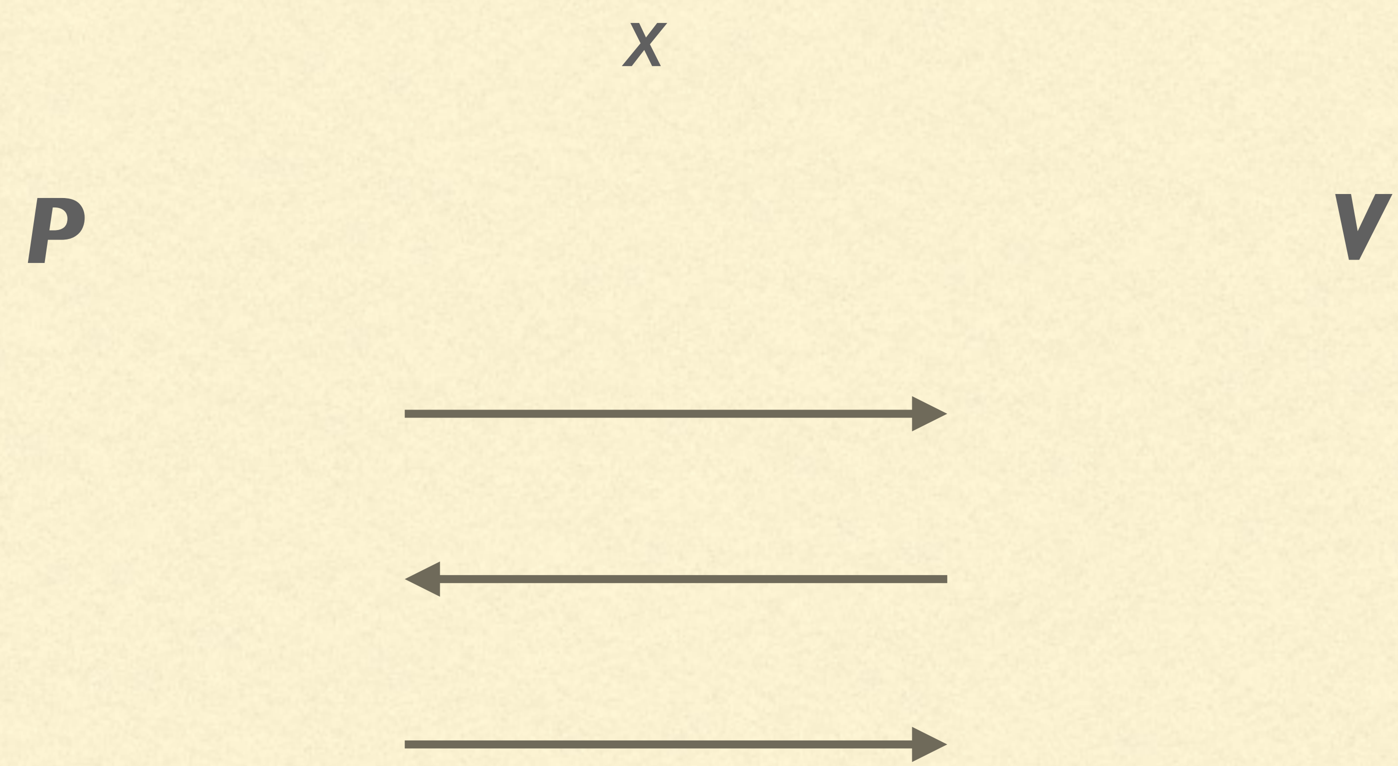
- Previous definition models static corruptions
- Adaptive corruptions closer to real world; systems ‘hacked’ dynamically
- Adaptive ZK composed to build larger protocols resilient to dynamic corruptions
- Adaptive security without erasures: Every simulated transcript has to be completely ‘explainable’ wrt any witness

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x

P V

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS



ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x

P $\qquad\qquad\qquad\qquad$ V

P corrupted here

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

x

Sim_P

V

P corrupted here

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

x

$\mathrm{Sim}_{\boldsymbol{P}}$ w $\qquad\qquad$ $\boldsymbol{V}$

$\longrightarrow$

$\longleftarrow$

$\longrightarrow$

P corrupted here

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x

$\mathrm{Sim}_{\boldsymbol{P}}(w)$: $r_{\boldsymbol{P}}$

$\boldsymbol{V}$

P corrupted here

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x

$\boldsymbol{P}\,(w\,;\,r_P)$ $\boldsymbol{V}$

$\longrightarrow$

$\longleftarrow$

$\longrightarrow$

P corrupted here

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

x

$\boldsymbol{P}\ (w\ ;\ r_P)$ $\qquad\qquad\qquad\qquad$ $\boldsymbol{V}$

$\longrightarrow$

$\longleftarrow$

$\longrightarrow$

P corrupted here

r_P explains messages wrt w

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

x

$\boldsymbol{P}$ (w ; r_P)

$\boldsymbol{V}$

$\boldsymbol{P}$ corrupted here

r_P explains messages wrt w

Execution finishes

accept/reject

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

x

P (w ; r_P)

V

P corrupted here

r_P explains messages wrt w

accept/reject

Execution finishes

V corrupted now

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x

P (w ; $r_{\boldsymbol{P}}$)

V

P corrupted here

$r_{\boldsymbol{P}}$ explains messages wrt w

Execution finishes

V corrupted now

accept/reject

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

x

$\boldsymbol{P}$ (w ; $r_{\boldsymbol{P}}$)

Sim$_{\boldsymbol{V}}$

$\boldsymbol{P}$ corrupted here

$r_{\boldsymbol{P}}$ explains messages wrt w

accept/reject

Execution finishes

$\boldsymbol{V}$ corrupted now

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x

$\boldsymbol{P}\,(w\,;\,r_P)$

Sim$_{\boldsymbol{V}}$ r_V

$\boldsymbol{P}$ corrupted here

r_P explains messages wrt w

accept/reject

Execution finishes

$\boldsymbol{V}$ corrupted now

ADAPTIVE ZERO-KNOWLEDGE FROM GARBLED CIRCUITS

x

$\boldsymbol{P}\,(w\,;\,r_P)$

$\boldsymbol{V}\,(\,r_V\,)$

P corrupted here

r_P explains messages wrt w

accept/reject

Execution finishes

V corrupted now

ADAPTIVE ZERO-KNOWLEDGE
FROM GARBLED CIRCUITS

x

P (w ; r_P)

V (r_V)

r_V explains messages and output

P corrupted here

r_P explains messages wrt w

Execution finishes

V corrupted now

accept/reject

ADAPTIVE ZERO-KNOWLEDGE: DESIDERATA

ADAPTIVE ZERO-KNOWLEDGE: DESIDERATA

- Most useful when **universally composable (UC)**

ADAPTIVE ZERO-KNOWLEDGE: DESIDERATA

- Most useful when **universally composable (UC)**
- Asymptotic: **constant rounds**, **linear communication**

ADAPTIVE ZERO-KNOWLEDGE: DESIDERATA

- Most useful when **universally composable (UC)**
- Asymptotic: **constant rounds**, **linear communication**
- Towards **concrete efficiency**

ADAPTIVE ZERO-KNOWLEDGE: DESIDERATA

- Most useful when **universally composable (UC)**
- Asymptotic: **constant rounds**, **linear communication**
- Towards **concrete efficiency**
- Assumptions: standard public-key (**DDH**, **LWE**, **LPN**, **QR**)

RELATED WORK

RELATED WORK

- [Canetti-Poburinnaya-Venkitasubramaniam 17] Constant-round Adaptive 2PC, Quadratic communication, heavy machinery

RELATED WORK

- [Canetti-Poburinnaya-Venkitasubramaniam 17] Constant-round Adaptive 2PC, Quadratic communication, heavy machinery
- [Hazay-Venkitasubramaniam 16] Adaptive ZK from OWFs, Non-constant rounds, quasi-linear communication, standalone security

RELATED WORK

- [Canetti-Poburinnaya-Venkitasubramaniam 17] Constant-round Adaptive 2PC, Quadratic communication, heavy machinery
- [Hazay-Venkitasubramaniam 16] Adaptive ZK from OWFs, Non-constant rounds, quasi-linear communication, standalone security
- [Jawurek-Kerschbaum-Orlandi 13] ZK from GCs: Static security, **upgraded** to adaptive with greater efficiency **in this work**

TECHNICAL CONTRIBUTIONS

Improving efficiency

Improving security

TECHNICAL CONTRIBUTIONS

Improving efficiency

OT+1 rounds, UC ZK, linear comm, GRO model

Improving security

TECHNICAL CONTRIBUTIONS

Improving efficiency	OT+1 rounds, UC ZK, linear comm, GRO model
Improving security	Adaptive UC ZK from Recv-Equiv-OT

TECHNICAL CONTRIBUTIONS

Improving efficiency	OT+1 rounds, UC ZK, linear comm, GRO model
Improving security	Adaptive UC ZK from Recv-Equiv-OT

- Any OT can be transformed to Recv-Equiv OT [Canetti-Lindell-Ostrovsky-Sahai 02]
- Two-round UC static ZK from garbled circuits with CRS

BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]

BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]

1^k

BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]

1^k

f

BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]

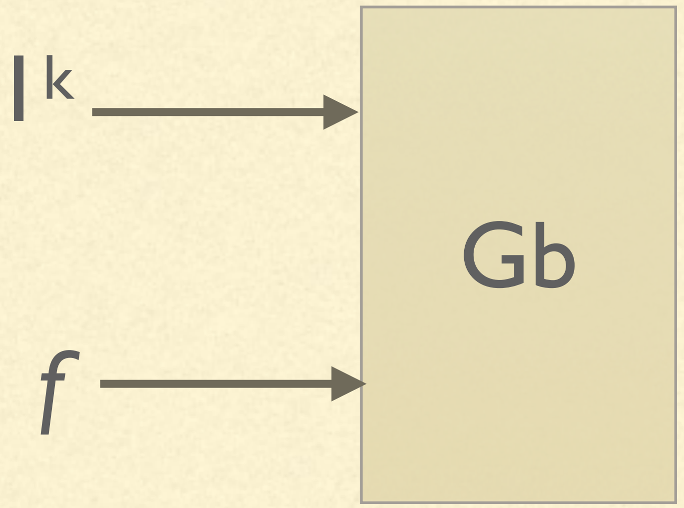
1^k

f

x

BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



x

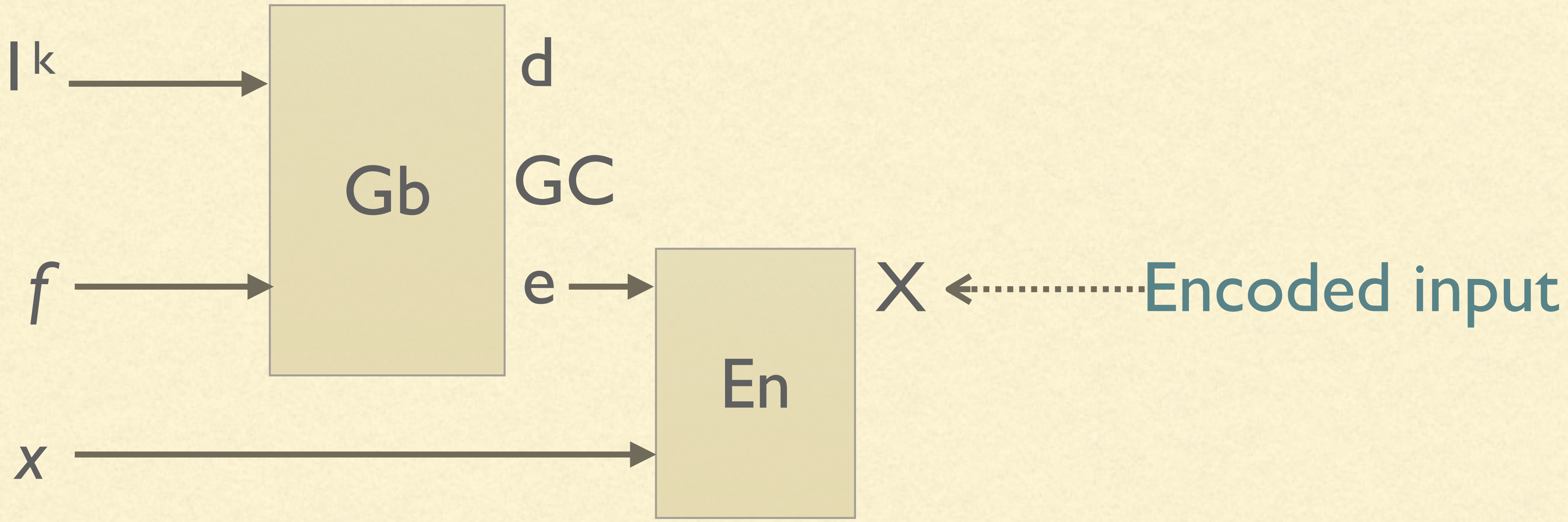
BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]

x

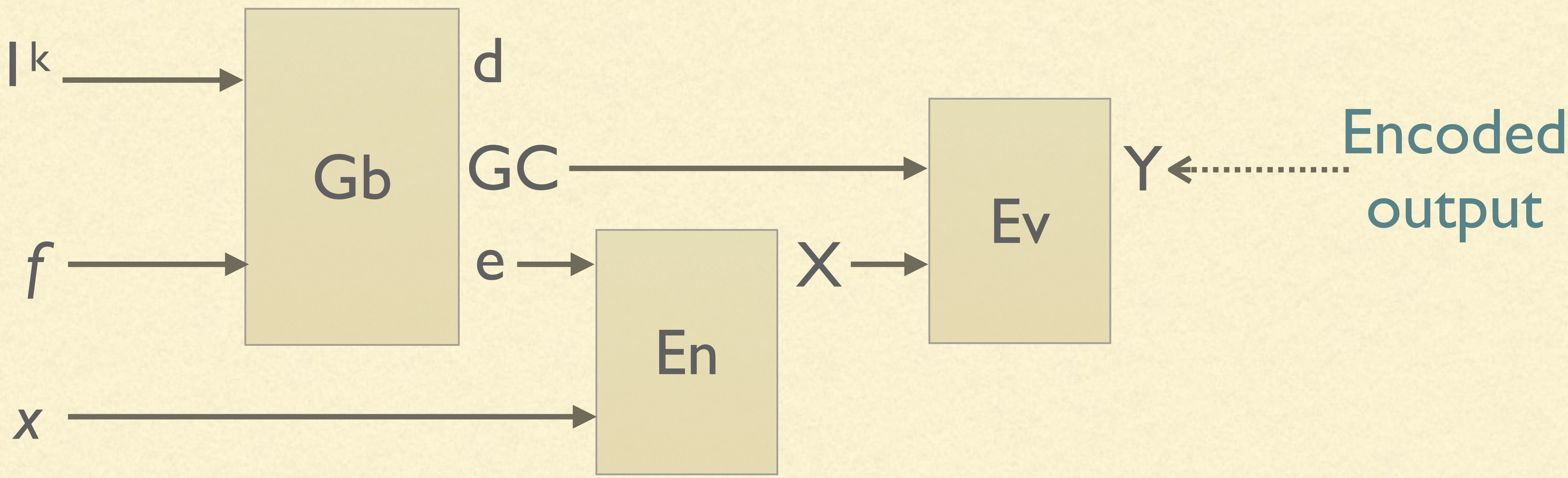
BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



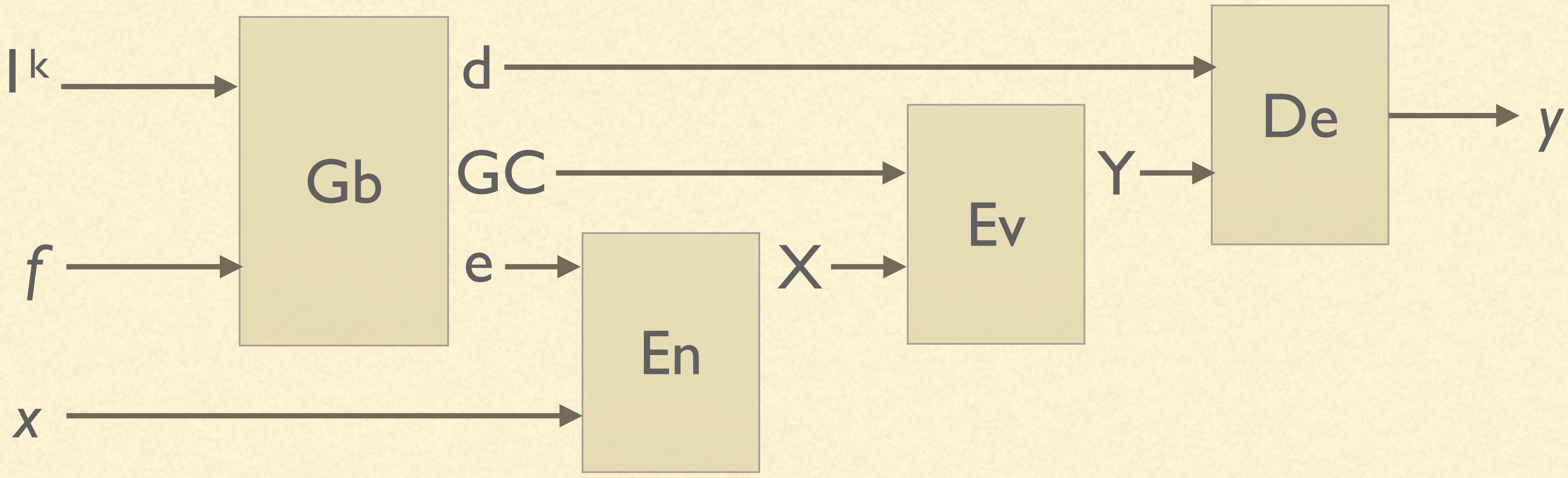
BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



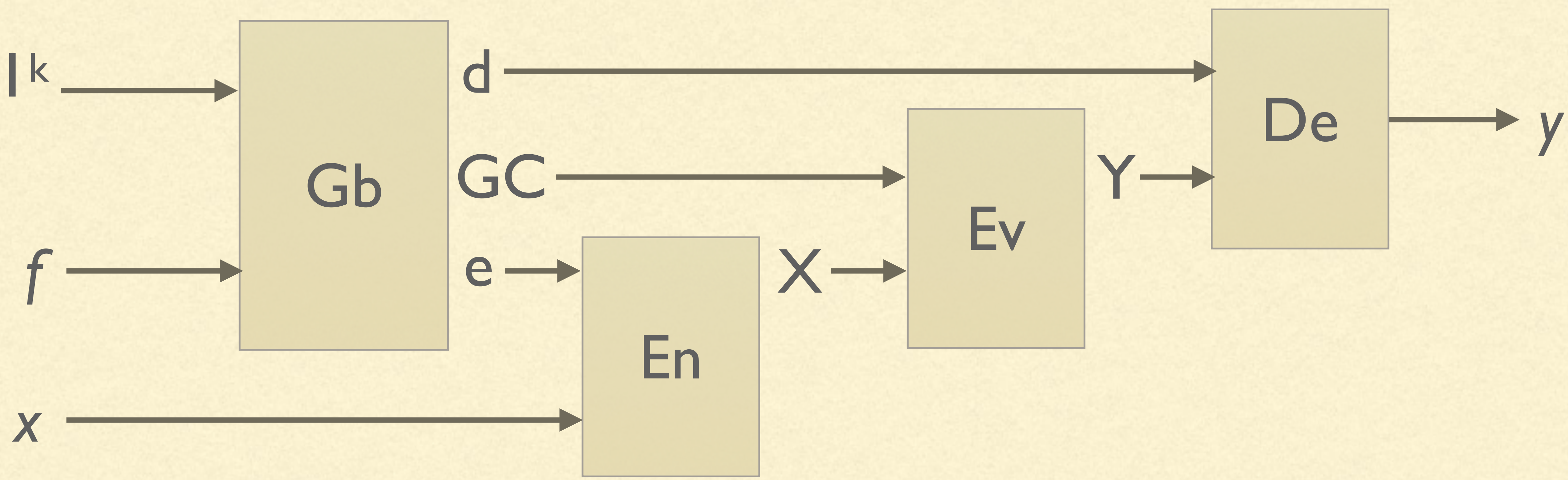
BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



BUILDING BLOCK: GARBLED CIRCUITS

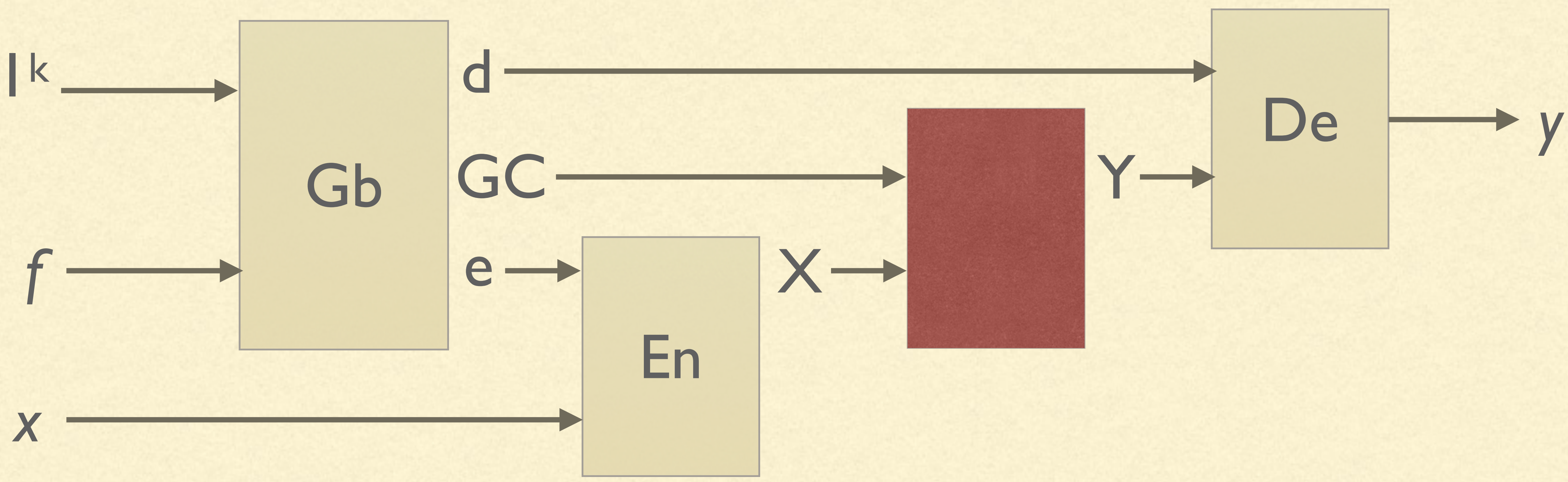
- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



- Correctness: *f(x) = y*

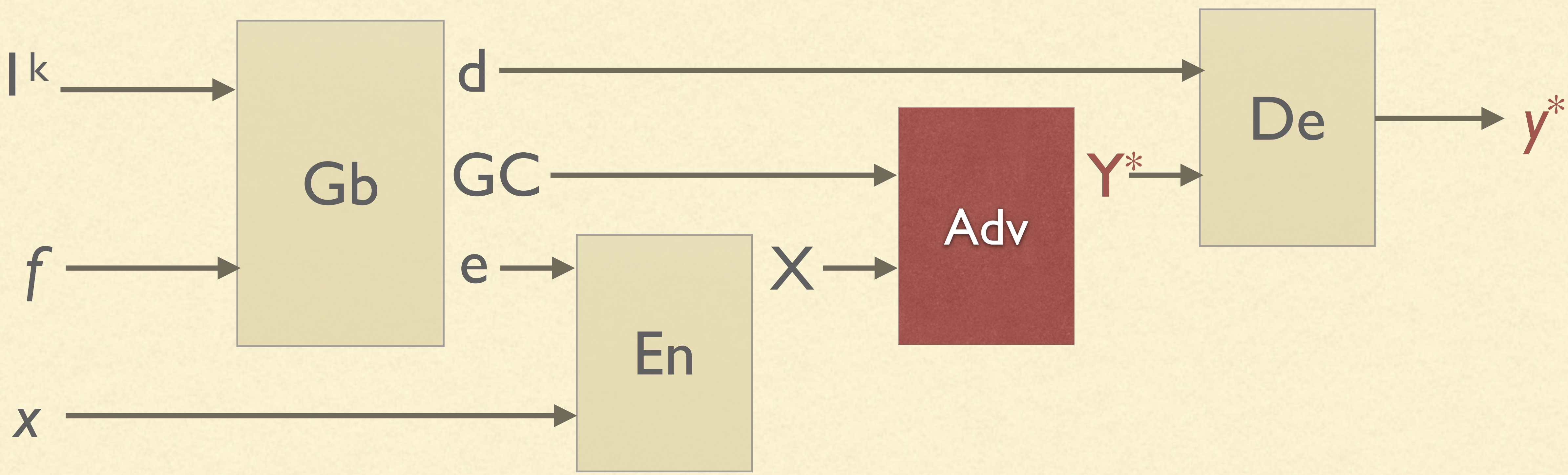
BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



BUILDING BLOCK: GARBLED CIRCUITS

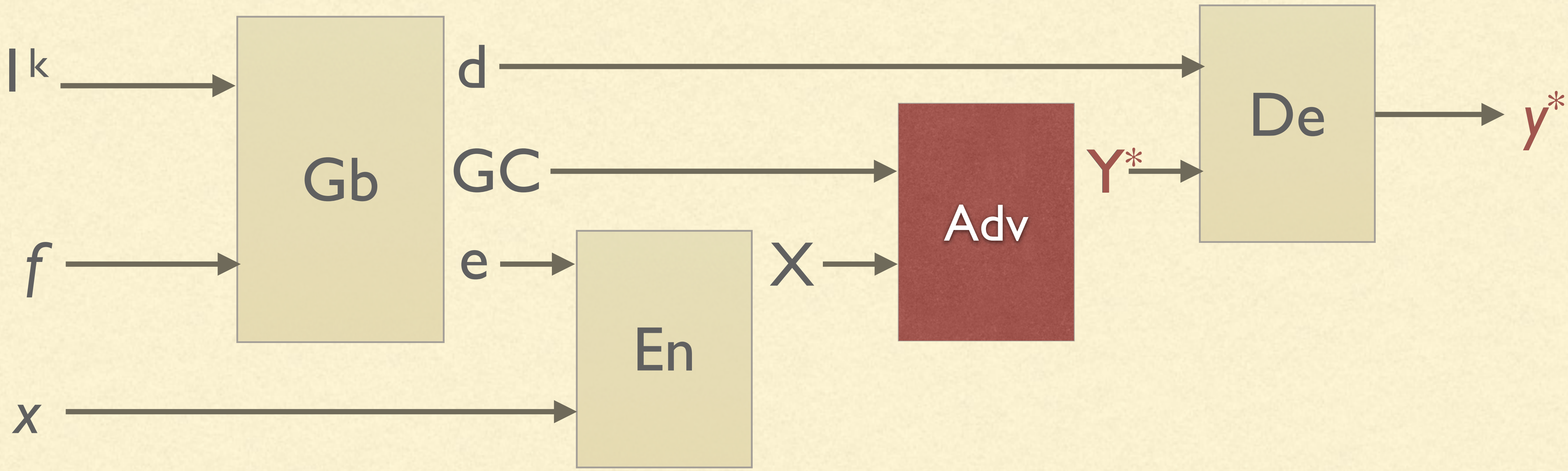
- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



- Authenticity: $Y^* = \mathrm{Ev}(GC, X)$

BUILDING BLOCK: GARBLED CIRCUITS

- Introduced by [Yao 86], formalization of [Bellare-Hoang-Rogaway 12]



- Authenticity: Y* = Ev(GC, X) … or De fails

BUILDING BLOCK: 1-2 OBLIVIOUS TRANSFER

S

R

BUILDING BLOCK: 1-2 OBLIVIOUS TRANSFER

S

m_0

m_1

R

BUILDING BLOCK: 1-2 OBLIVIOUS TRANSFER

S

m_0

m_1

R

b

BUILDING BLOCK: 1-2 OBLIVIOUS TRANSFER

S

m_0

m_1

OT

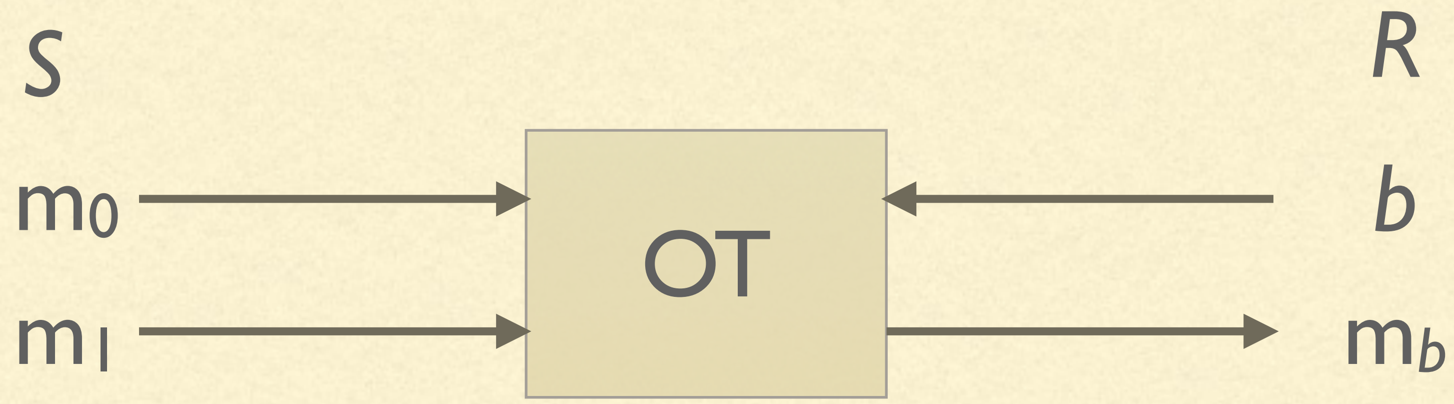
R

b

BUILDING BLOCK: 1-2 OBLIVIOUS TRANSFER

BUILDING BLOCK: 1-2 OBLIVIOUS TRANSFER

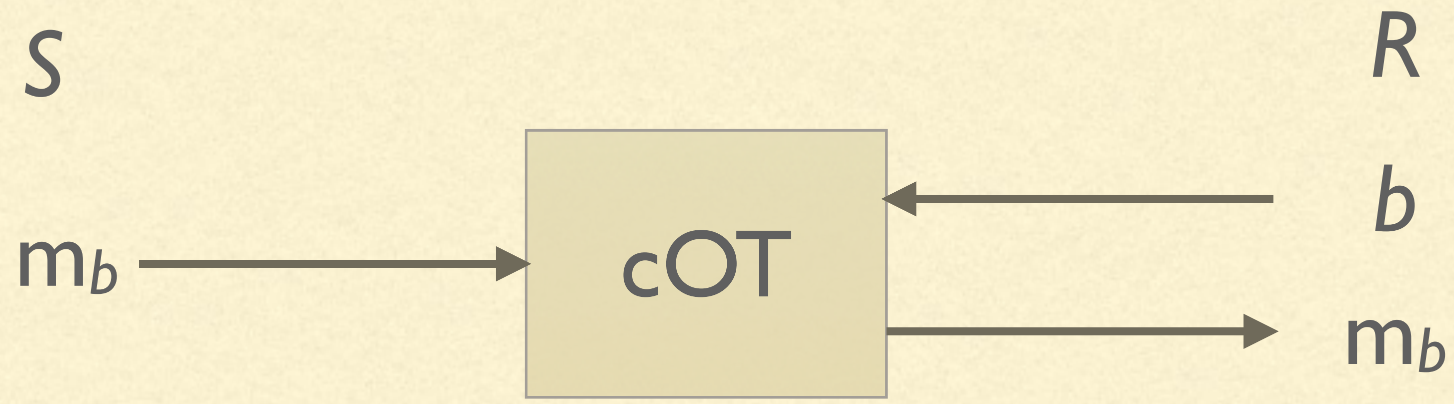
BUILDING BLOCK: 1-2 OBLIVIOUS TRANSFER



BUILDING BLOCK: COMMITTED OBLIVIOUS TRANSFER

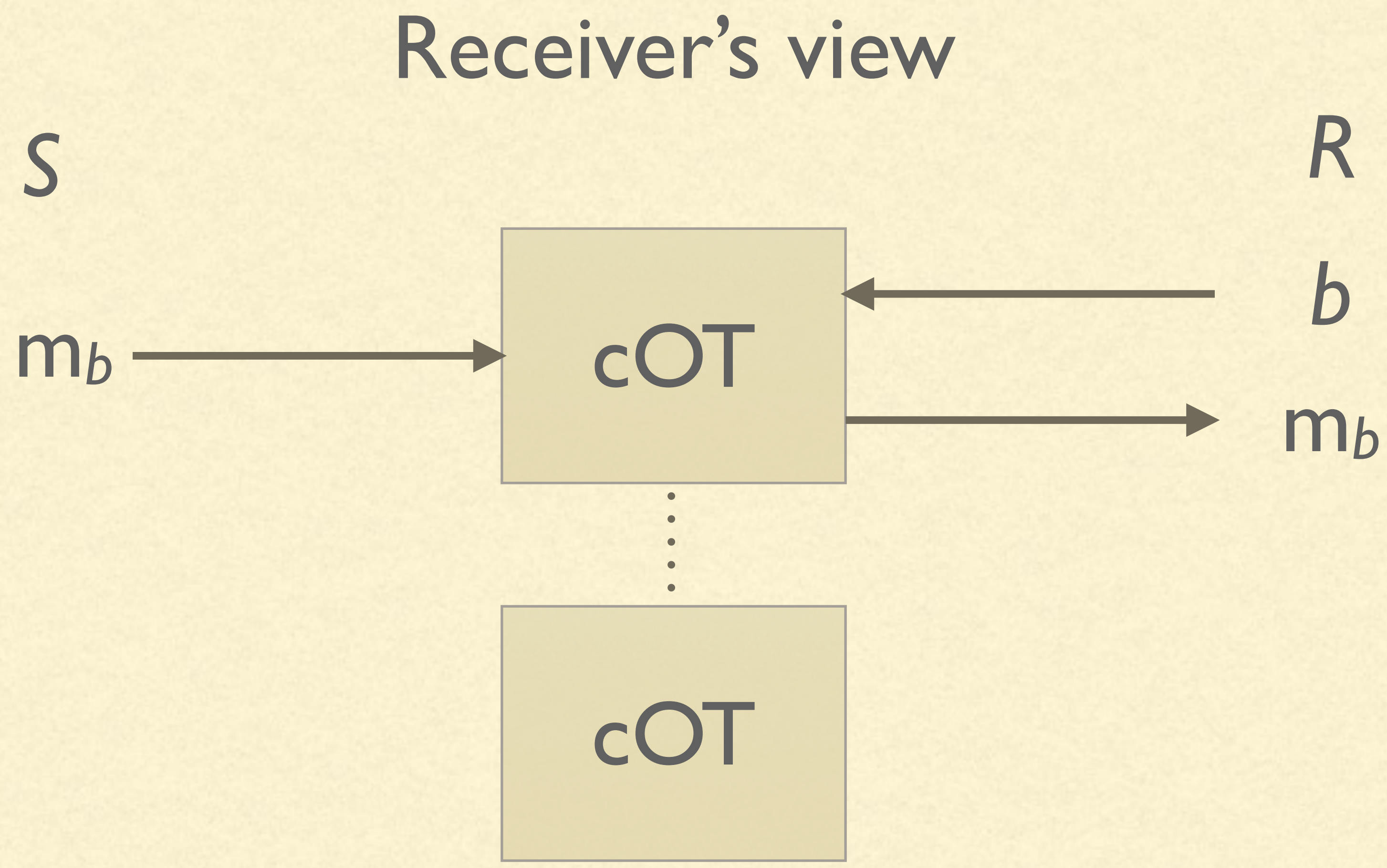
- For this protocol: committed cOT

Receiver's view



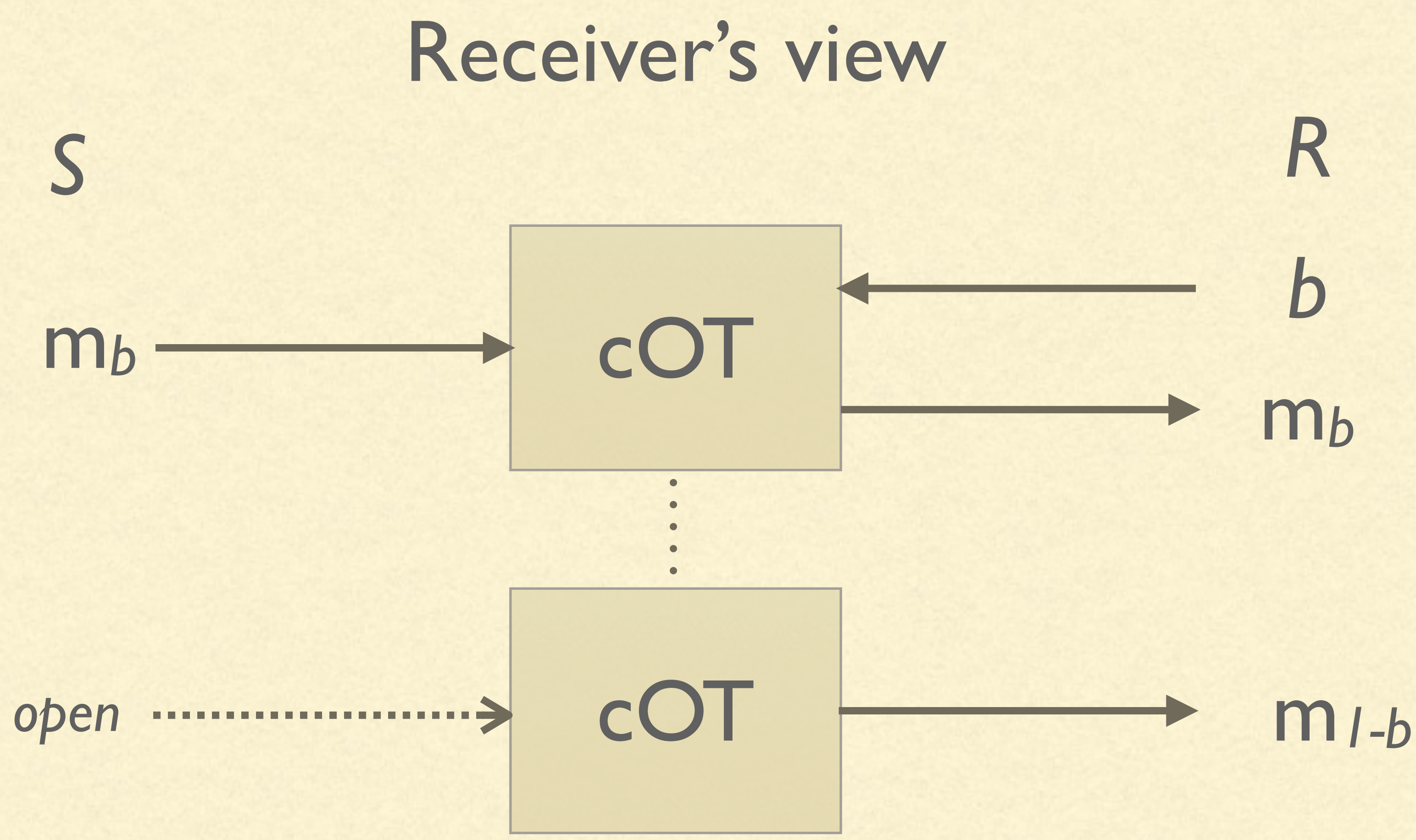
BUILDING BLOCK: COMMITTED OBLIVIOUS TRANSFER

- For this protocol: committed cOT



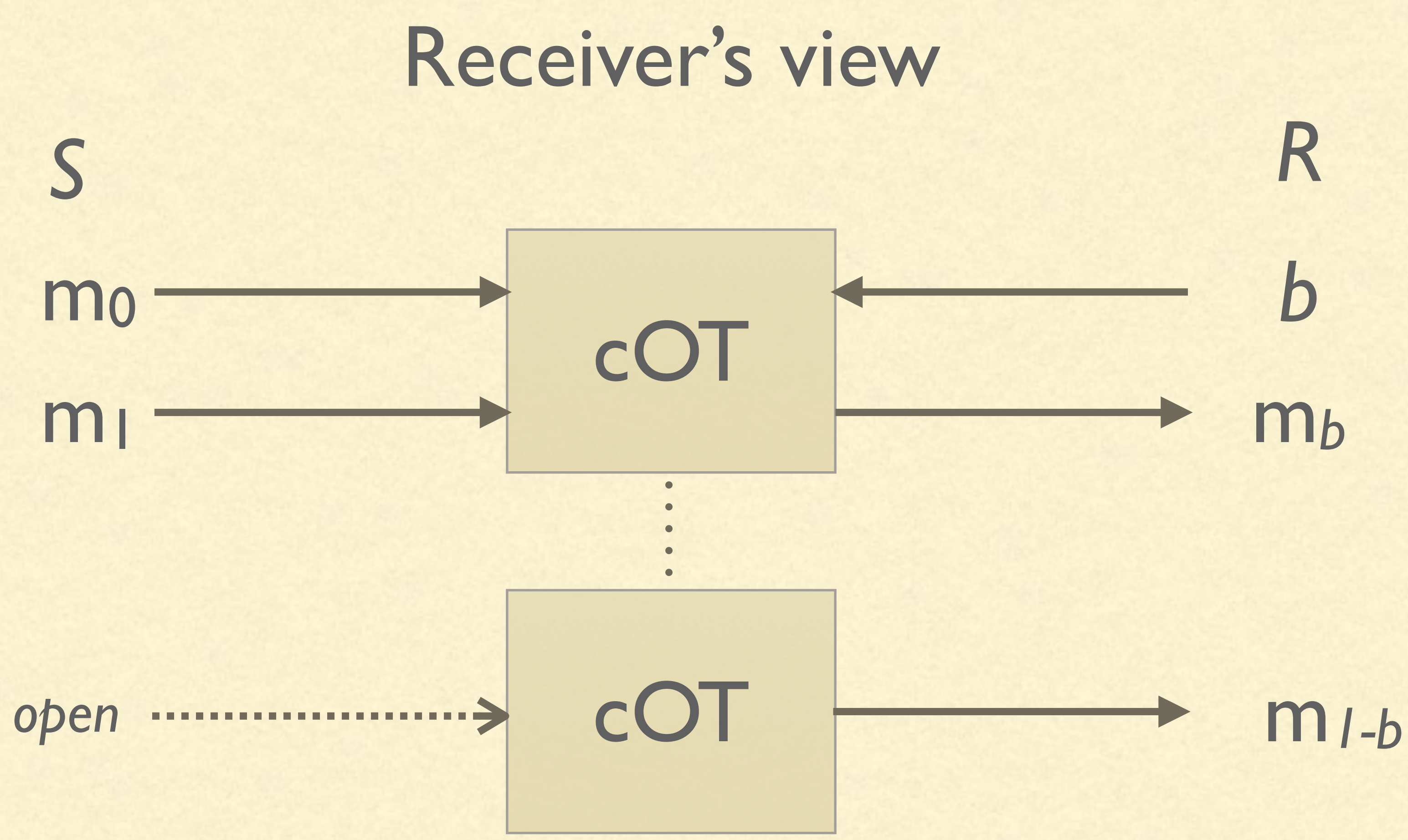
BUILDING BLOCK: COMMITTED OBLIVIOUS TRANSFER

- For this protocol: committed cOT



BUILDING BLOCK: COMMITTED OBLIVIOUS TRANSFER

- For this protocol: committed cOT



ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- ‘***P***’ wishes to prove knowledge of witness ‘w’ for NP statement ‘$x \in L$’ to ‘***V***’

P ***V***

ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- ‘***P***’ wishes to prove knowledge of witness ‘w’ for NP statement ‘x∈L’ to ‘***V***’

P ***V***

w

ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- ‘***P***’ wishes to prove knowledge of witness ‘w’ for NP statement ‘x∈L’ to ‘***V***’

P ***V***

Gb(f_x, 1^k)

w

ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement '$x \in L$' to '***V***'

P	***V***
	GC $\longleftarrow \mathrm{Gb}(f_x, 1^k)$
w	

ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement '$x \in L$' to '***V***'

P ***V***

GC , e $\longleftarrow Gb(f_x, 1^k)$

w

ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement '$x \in L$' to '***V***'

P	***V***
	$GC, e, Z \leftarrow Gb(f_x, 1^k)$
w	

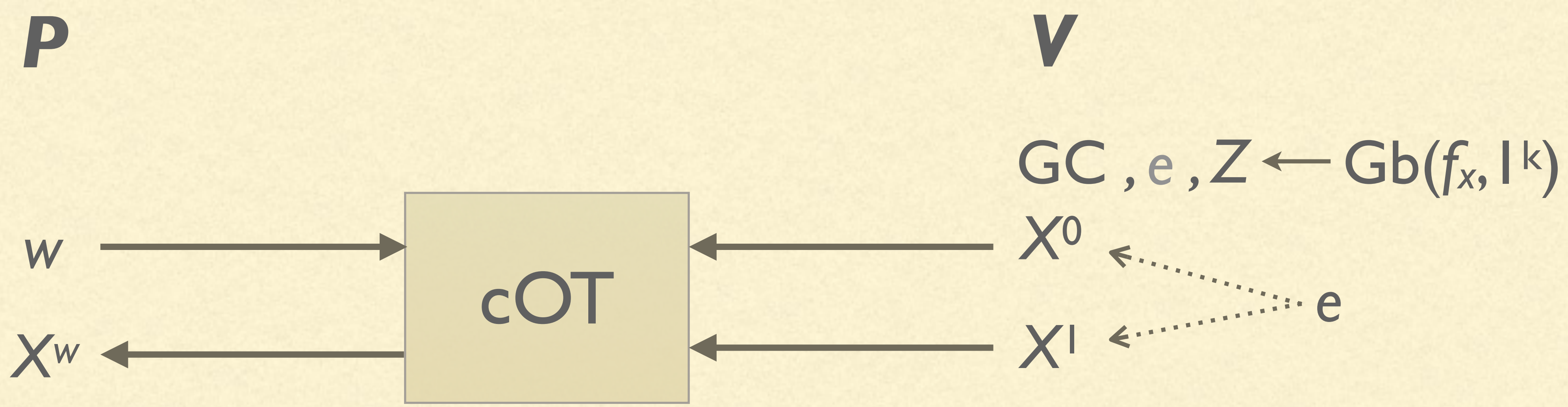
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- ‘***P***’ wishes to prove knowledge of witness ‘w’ for NP statement ‘x∈L’ to ‘***V***’

P	***V***
	$GC, e, Z \longleftarrow Gb(f_x, 1^k)$
w	X^0
	e
	X^1

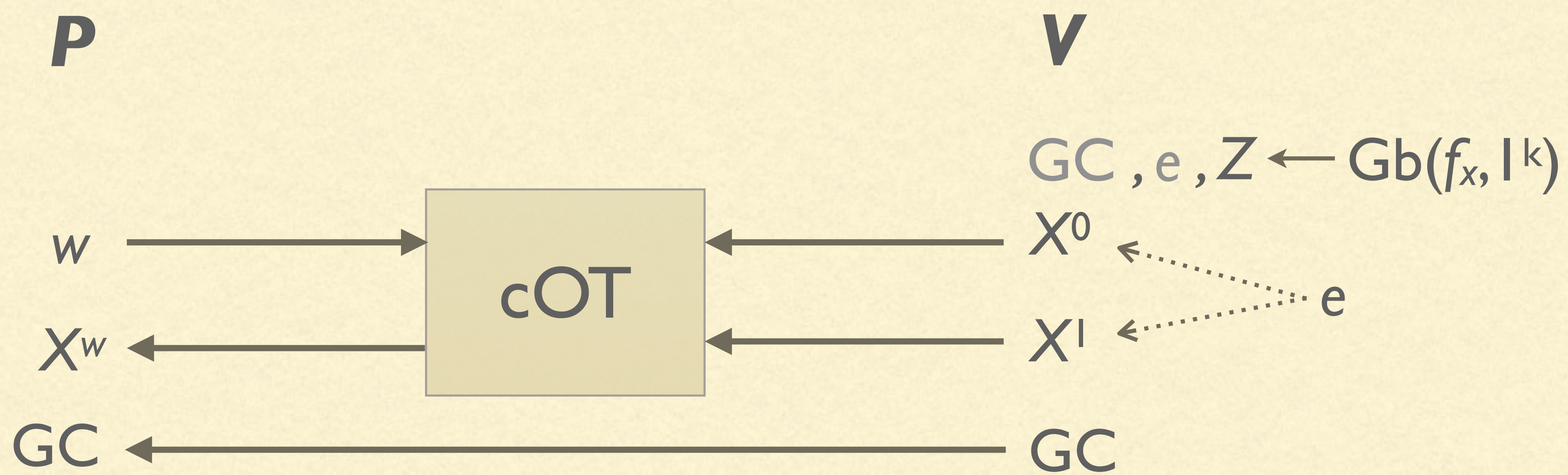
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement 'x∈L' to '***V***'



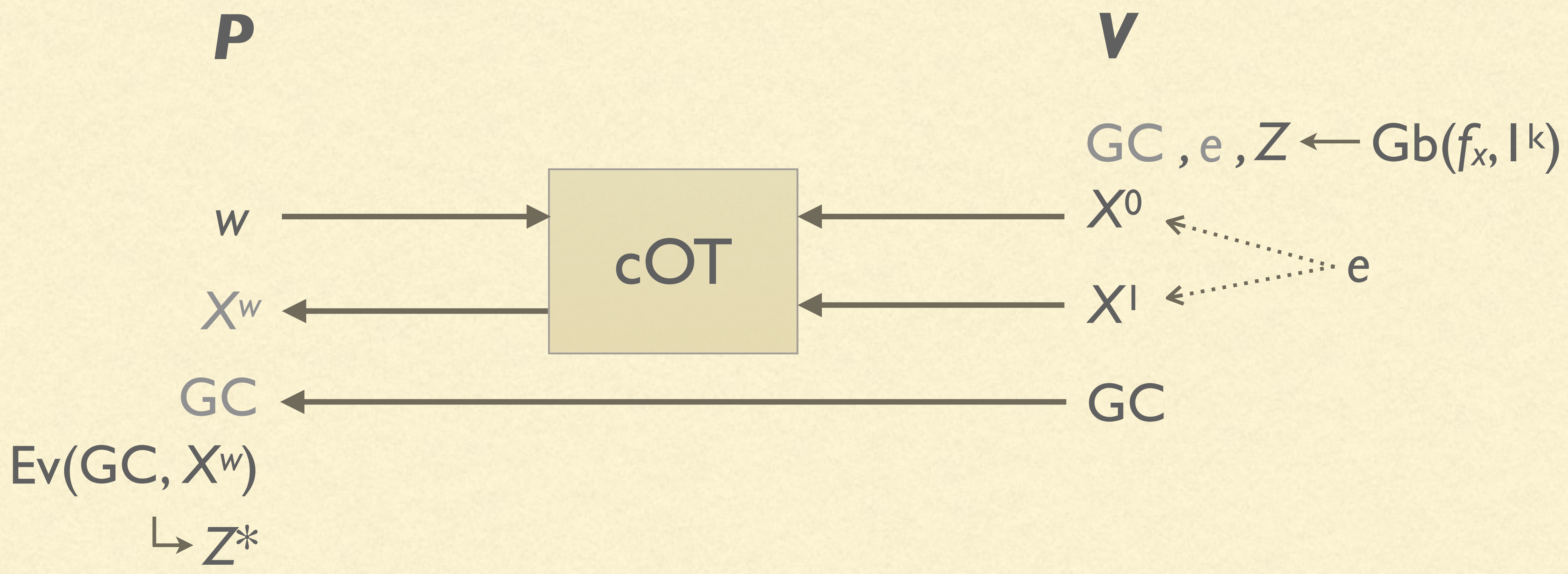
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement 'x∈L' to '***V***'



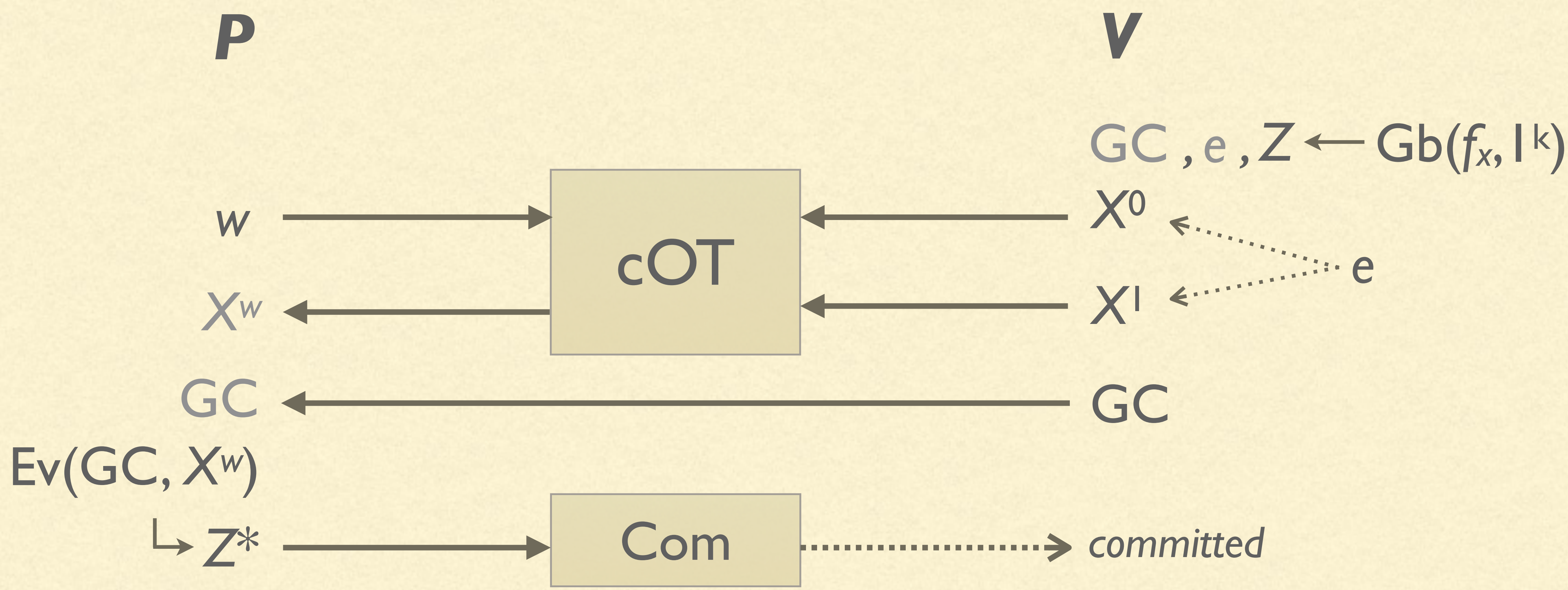
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement 'x∈L' to '***V***'



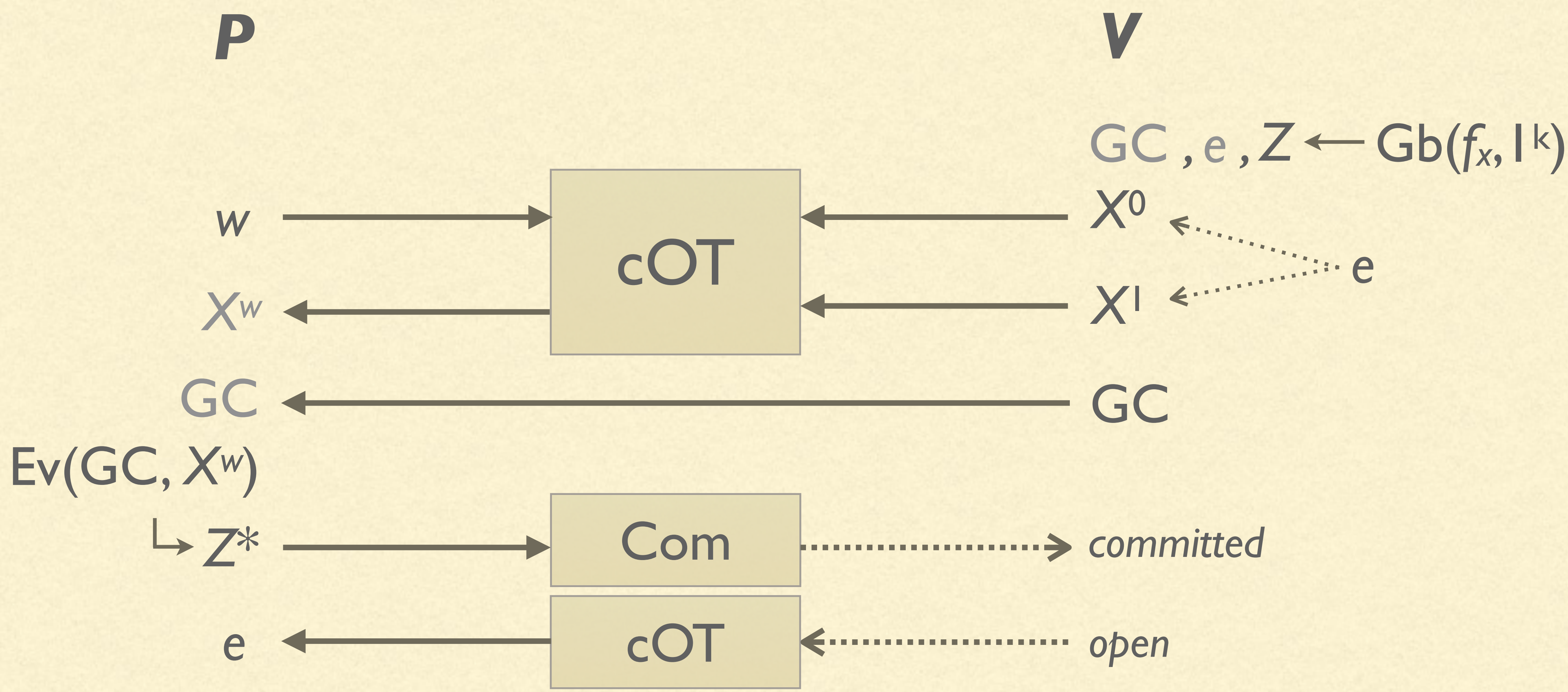
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement '$x \in L$' to '***V***'



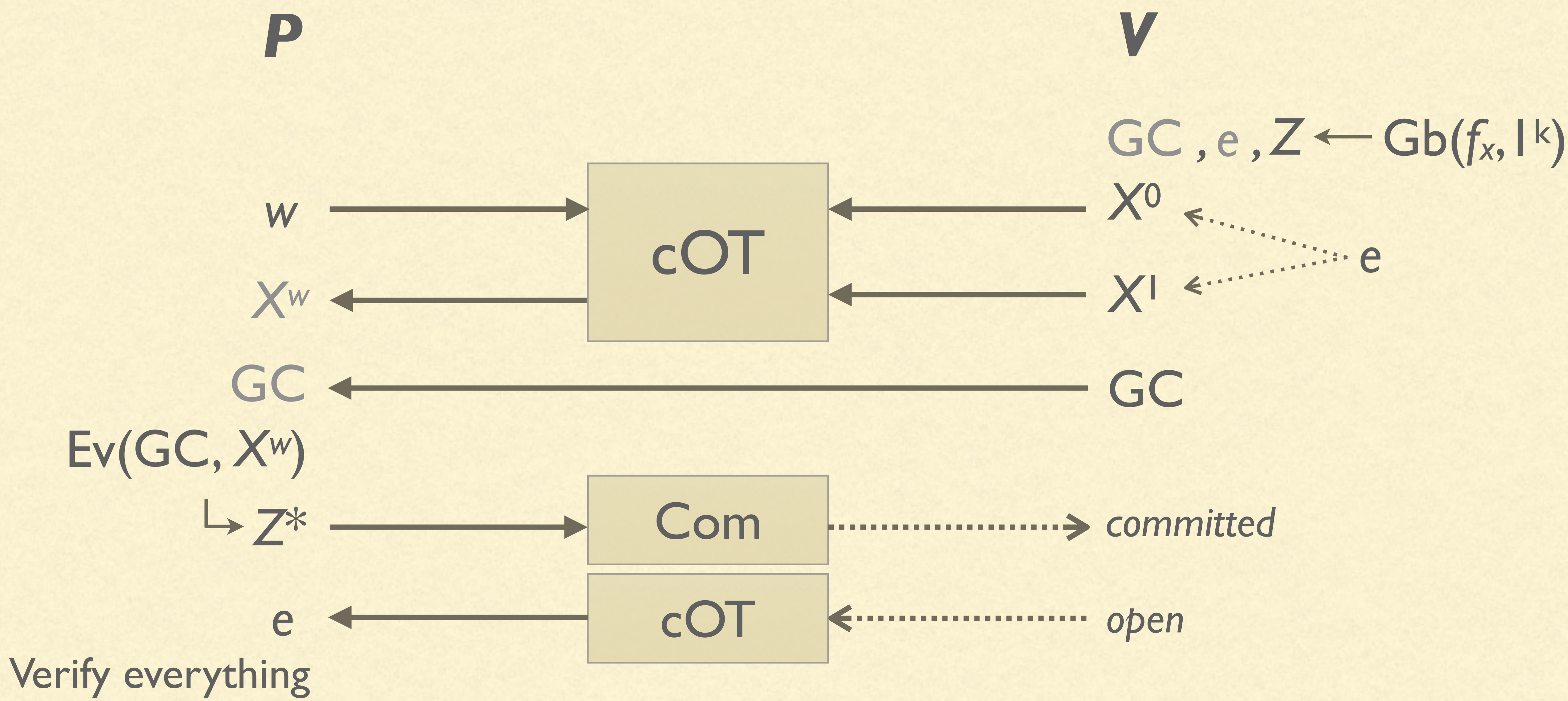
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement '$x \in L$' to '***V***'



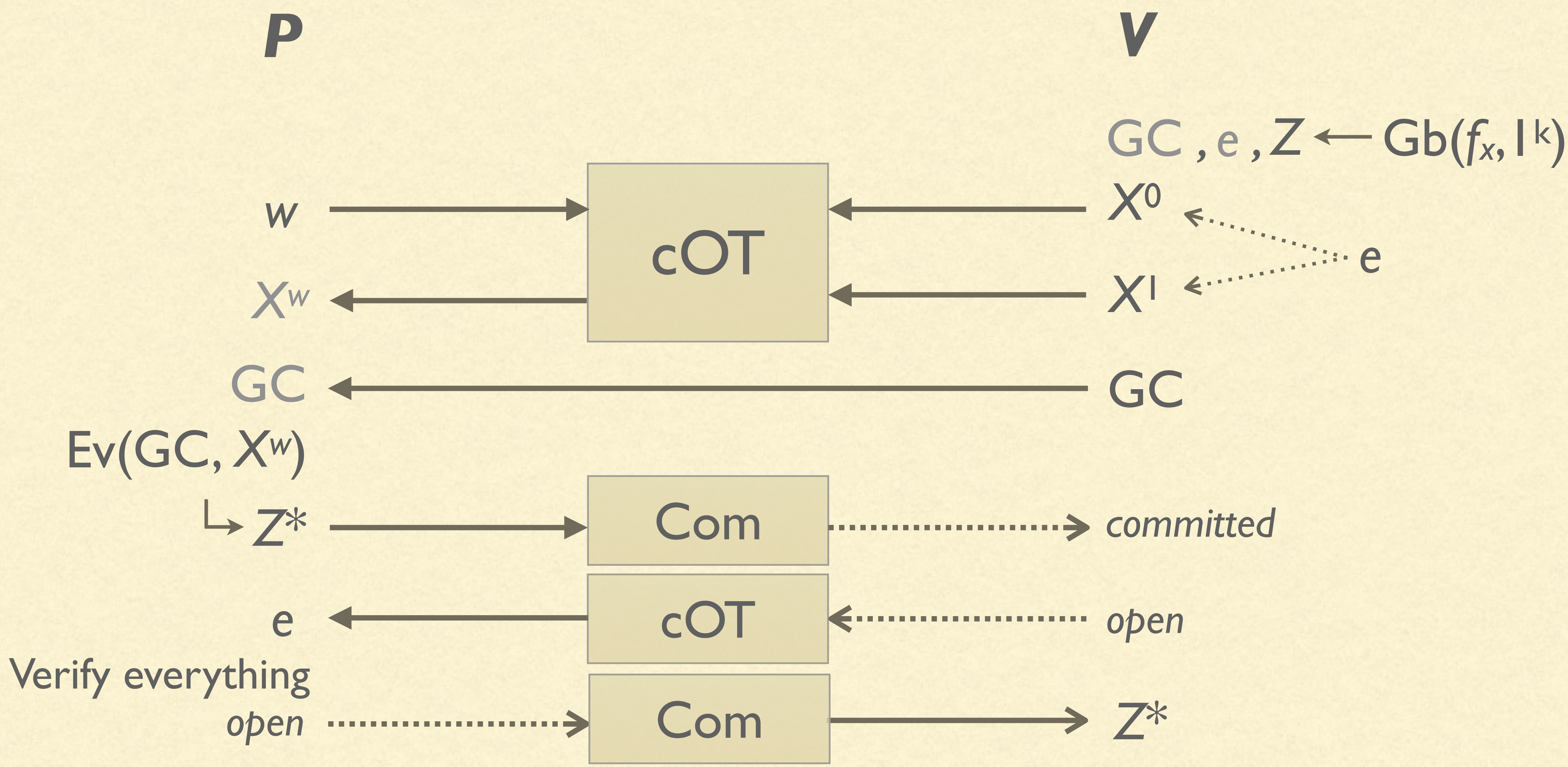
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement 'x∈L' to '***V***'



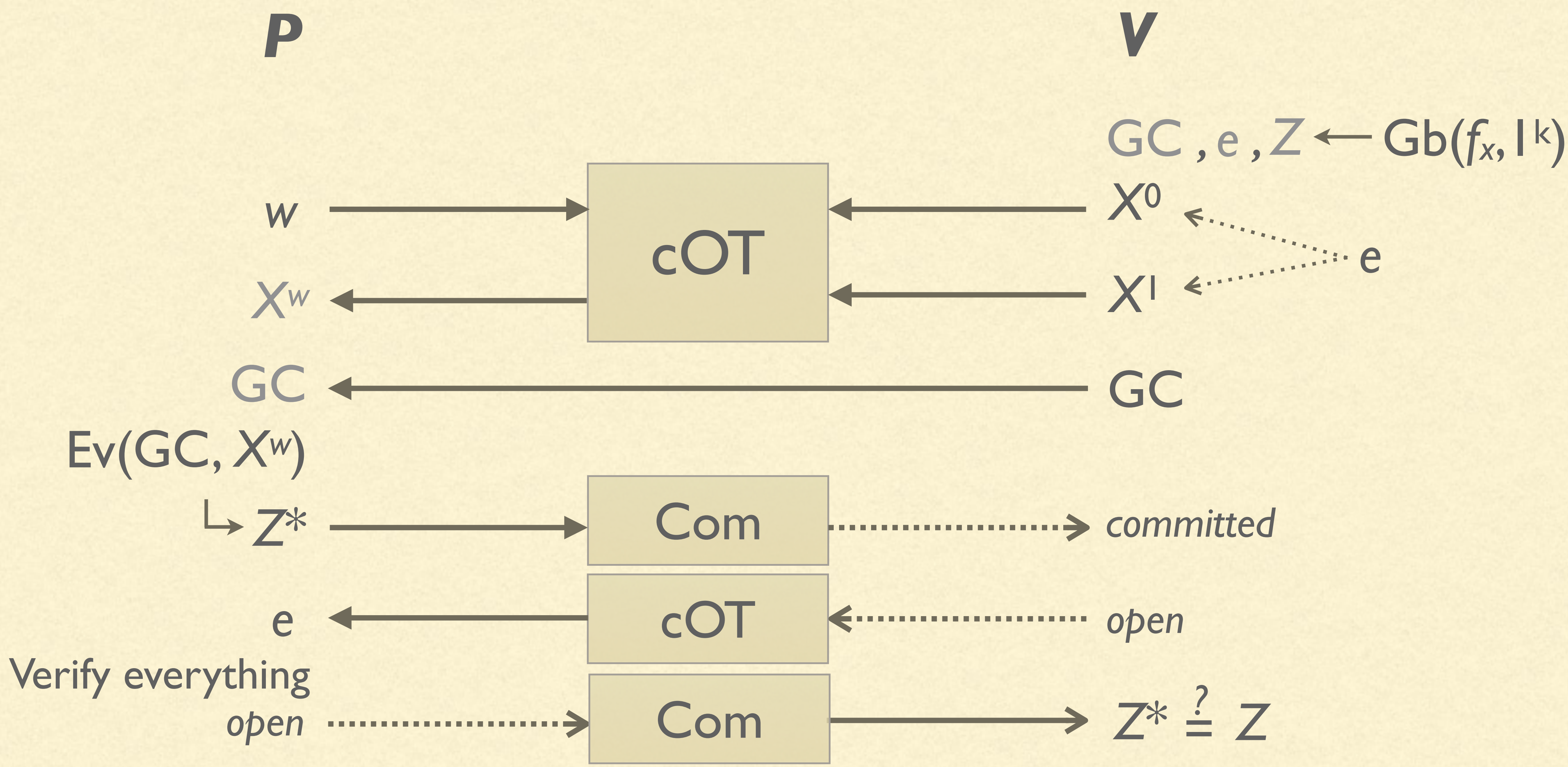
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- '***P***' wishes to prove knowledge of witness 'w' for NP statement 'x∈L' to '***V***'



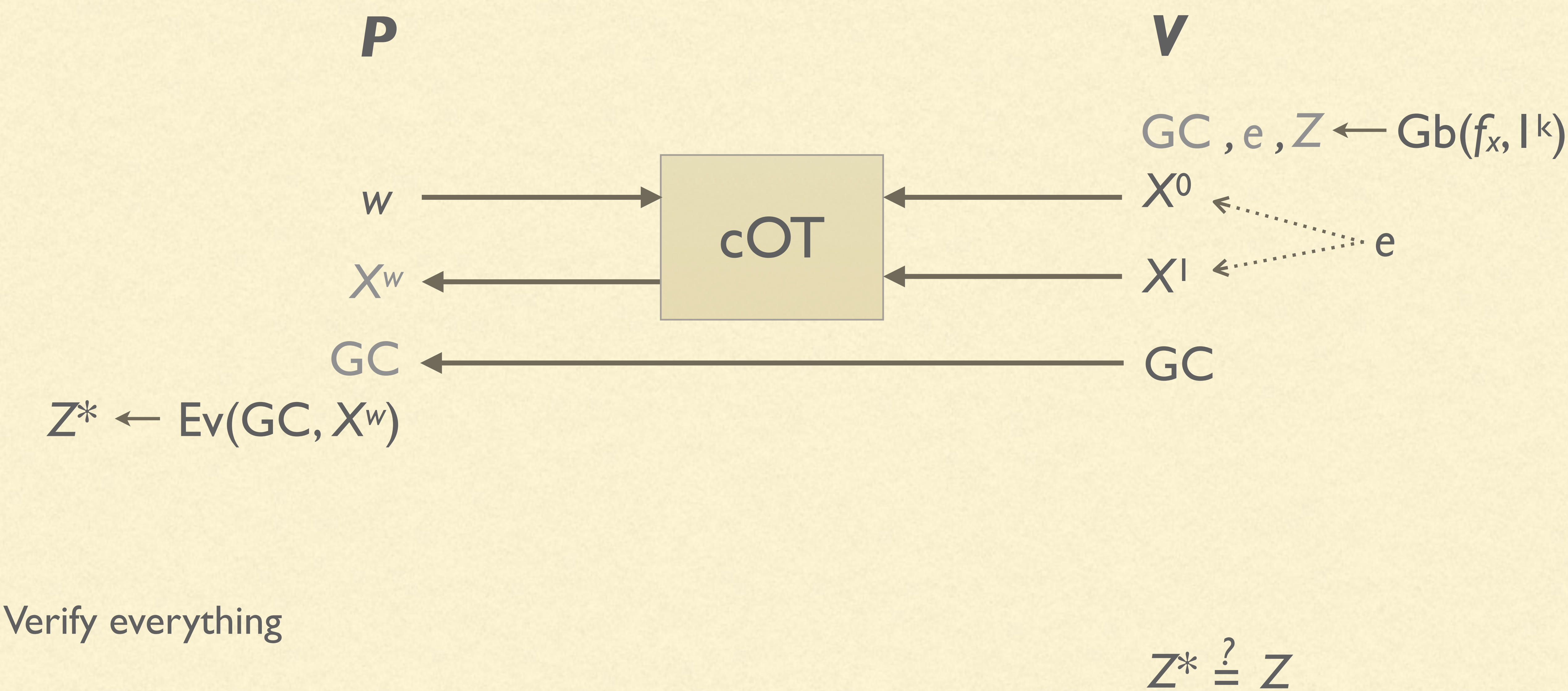
ZERO-KNOWLEDGE FROM GARBLED CIRCUITS [JKO13]

- ‘***P***’ wishes to prove knowledge of witness ‘w’ for NP statement ‘x∈L’ to ‘***V***’



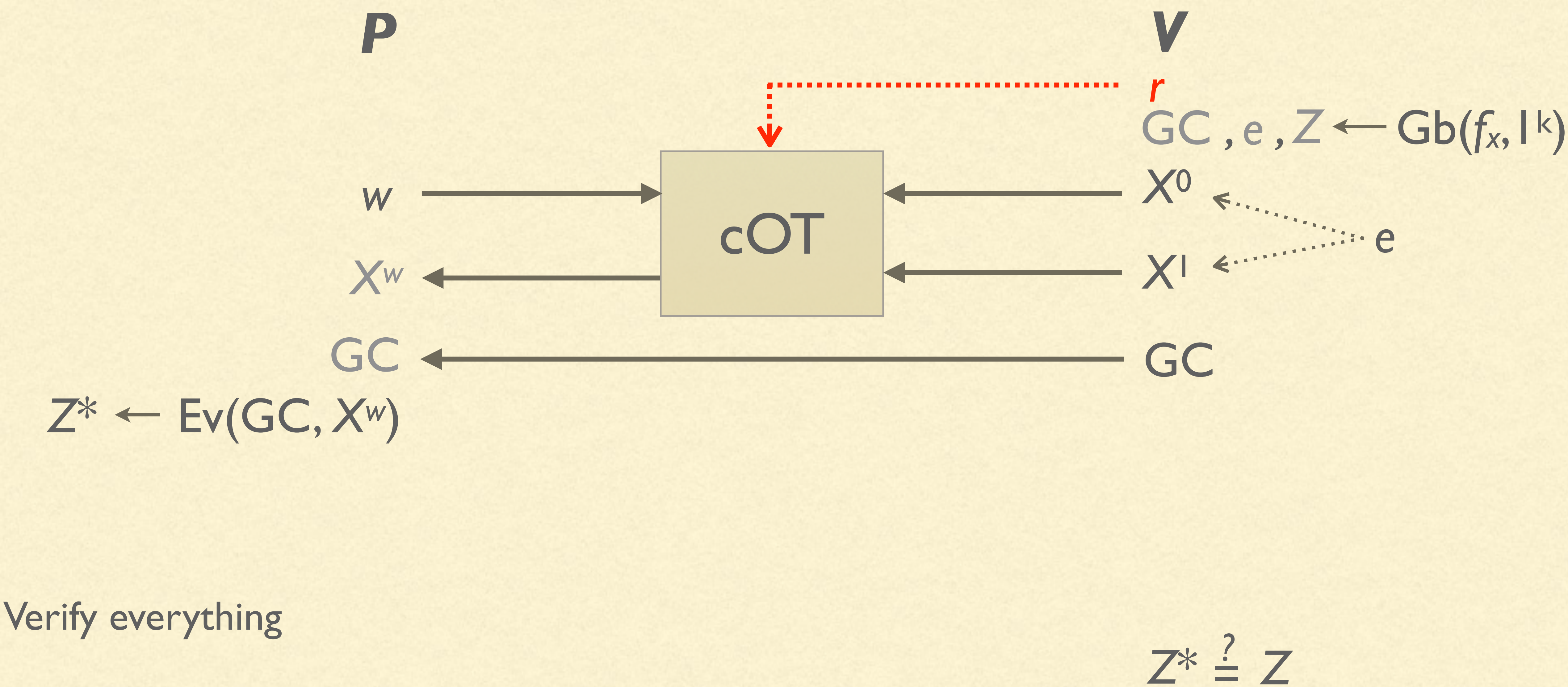
REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



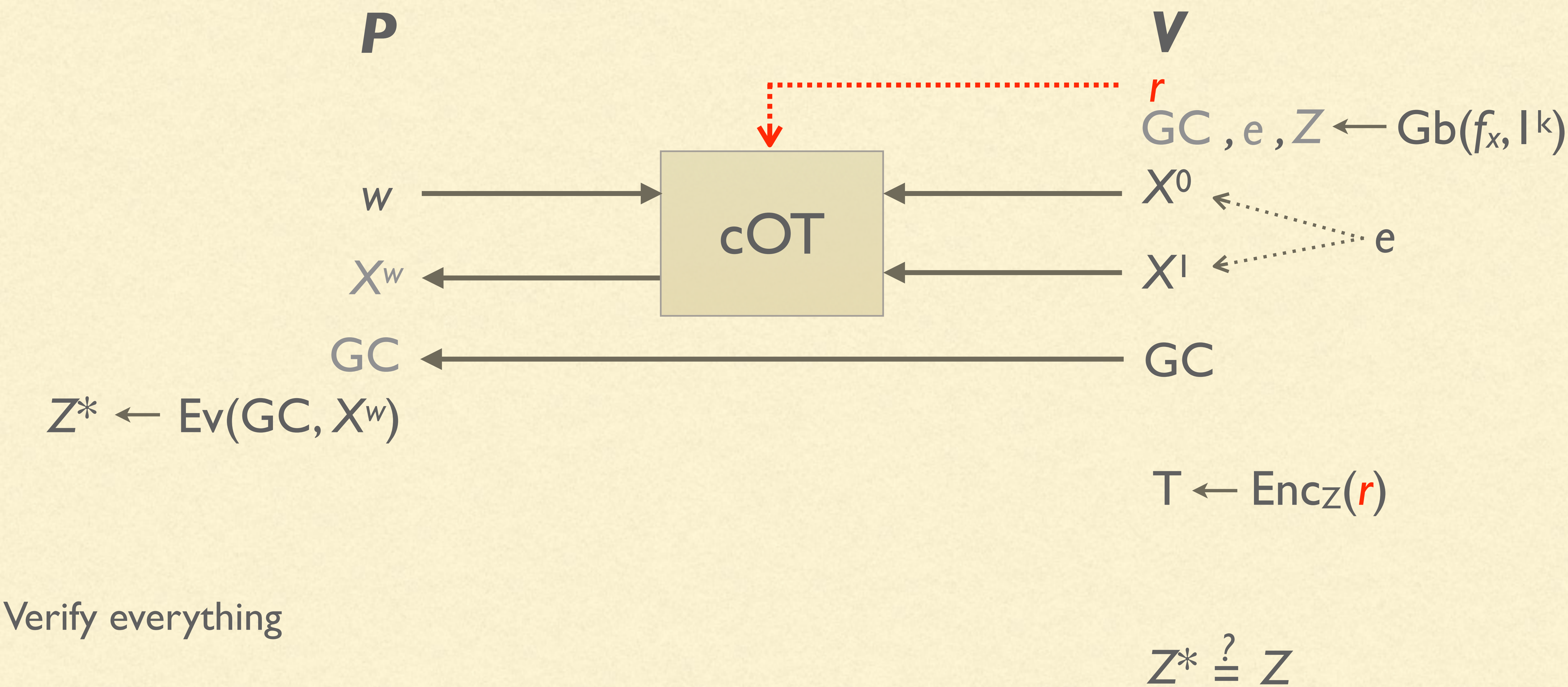
REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



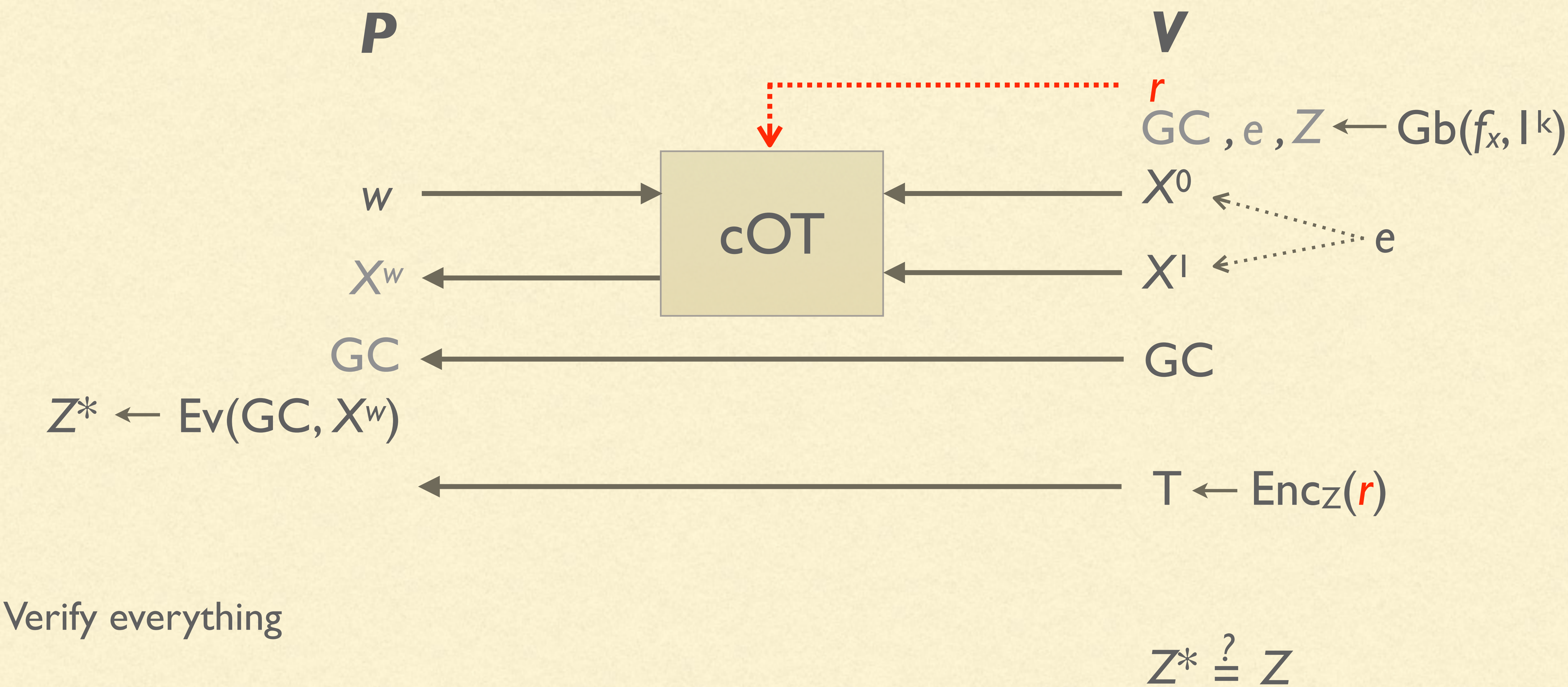
REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



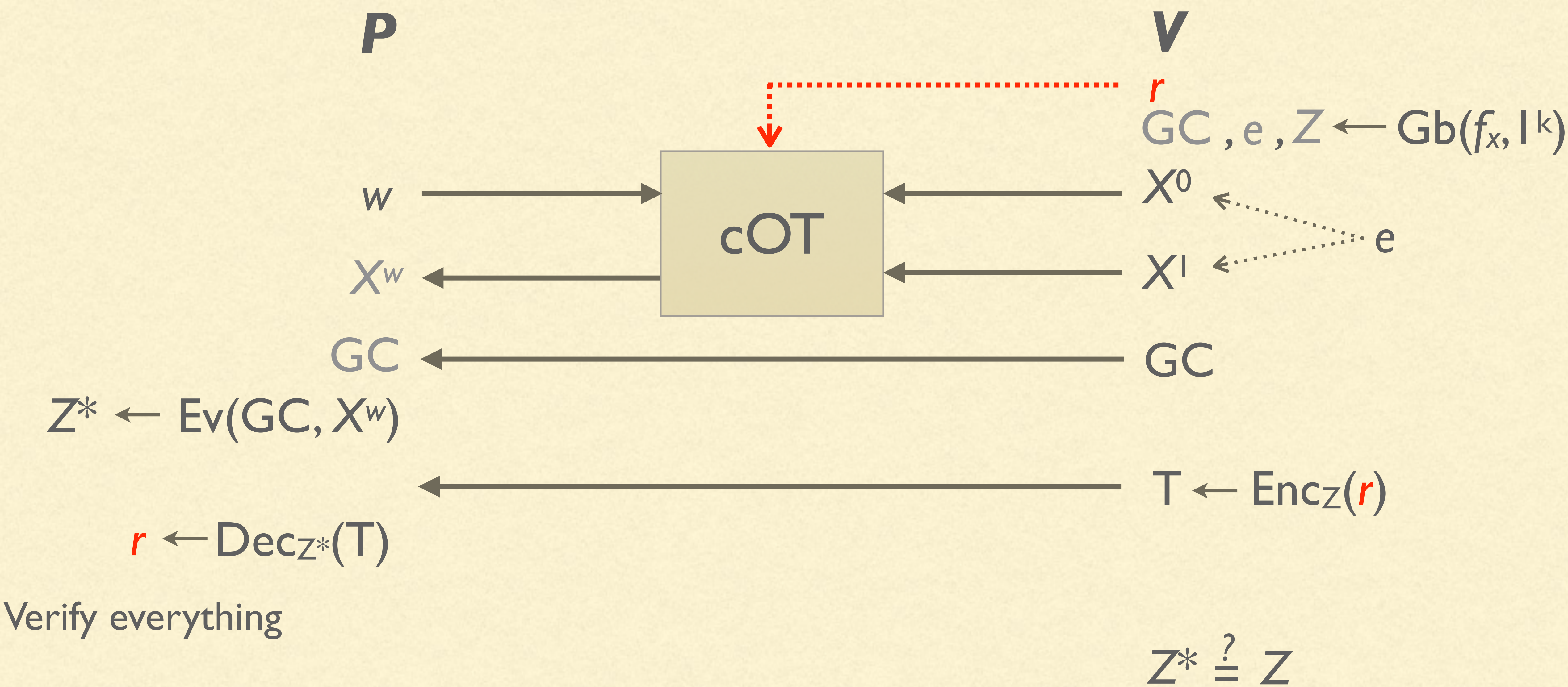
REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



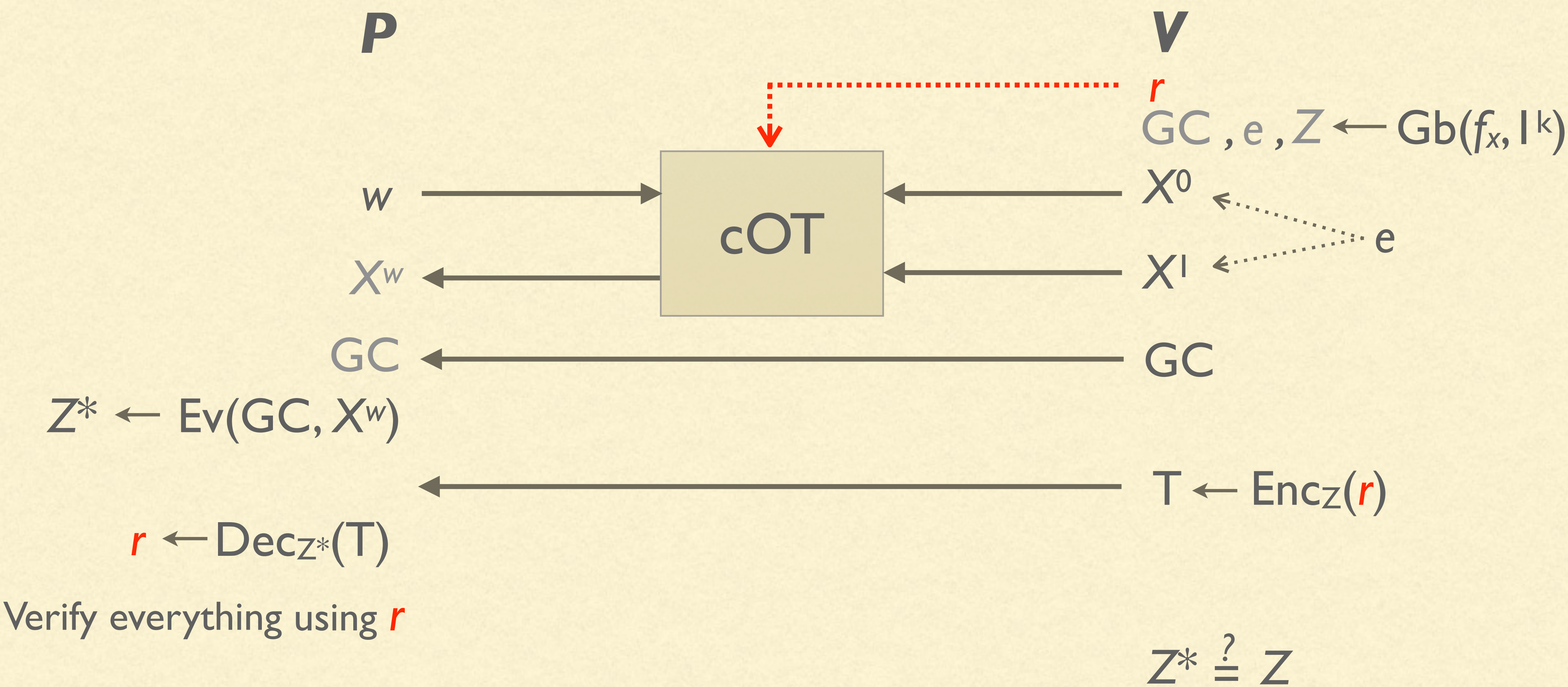
REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



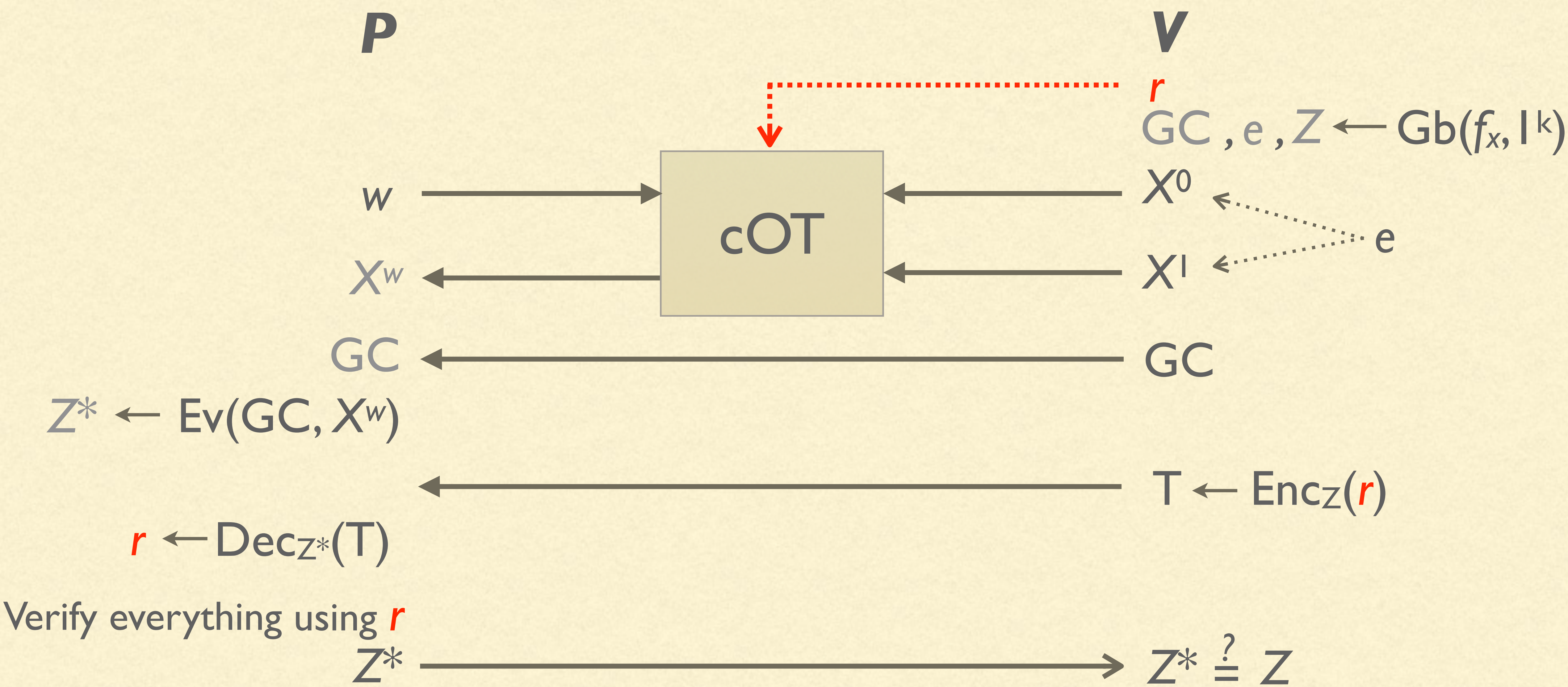
REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



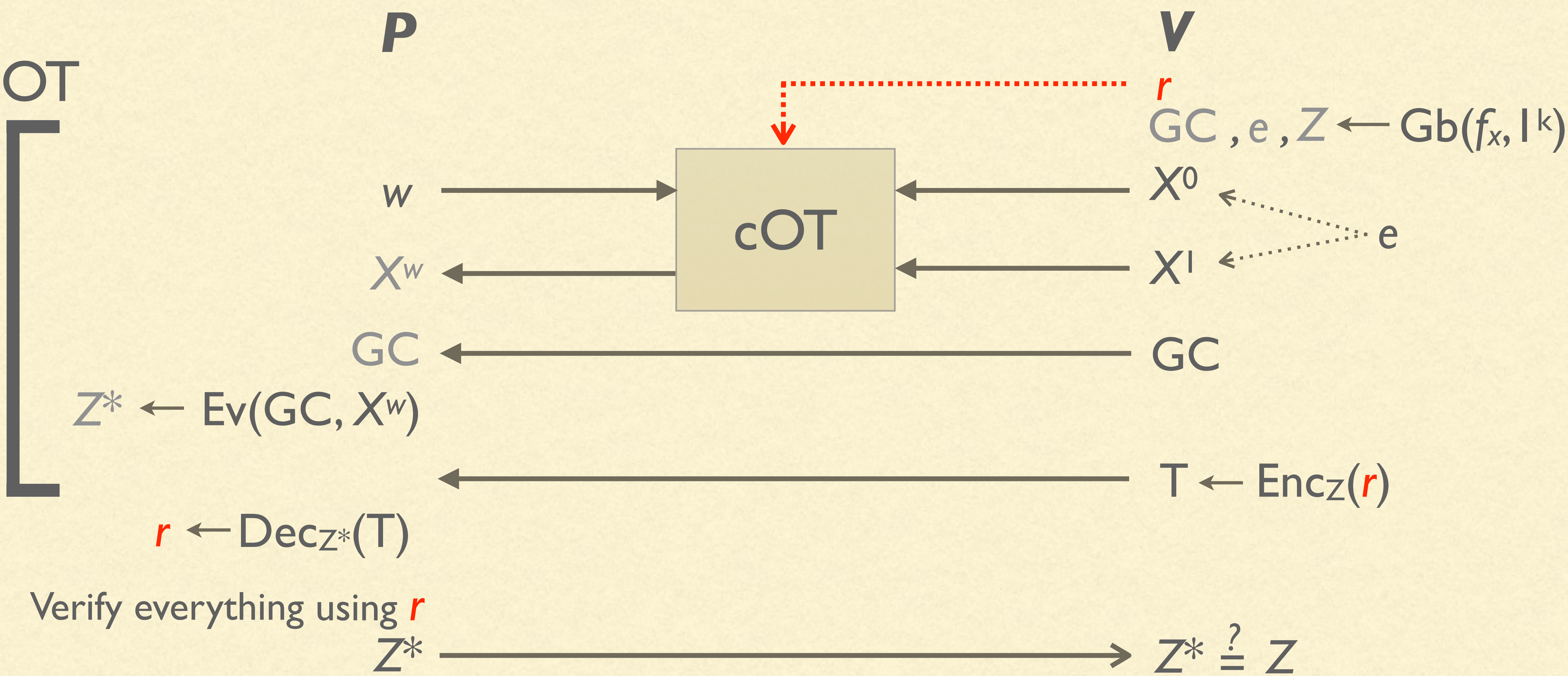
REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



REDUCING TO (OT+1) ROUNDS

- ***P***'s verification phase made non-interactive (still not adaptive)



REDUCING TO (OT+1) ROUNDS

REDUCING TO (OT+1) ROUNDS

- Encryption scheme has to be “committing”

REDUCING TO (OT+1) ROUNDS

- Encryption scheme has to be “committing”
- Circular security and observability of decryption key: solved by random oracle (non-programmable)

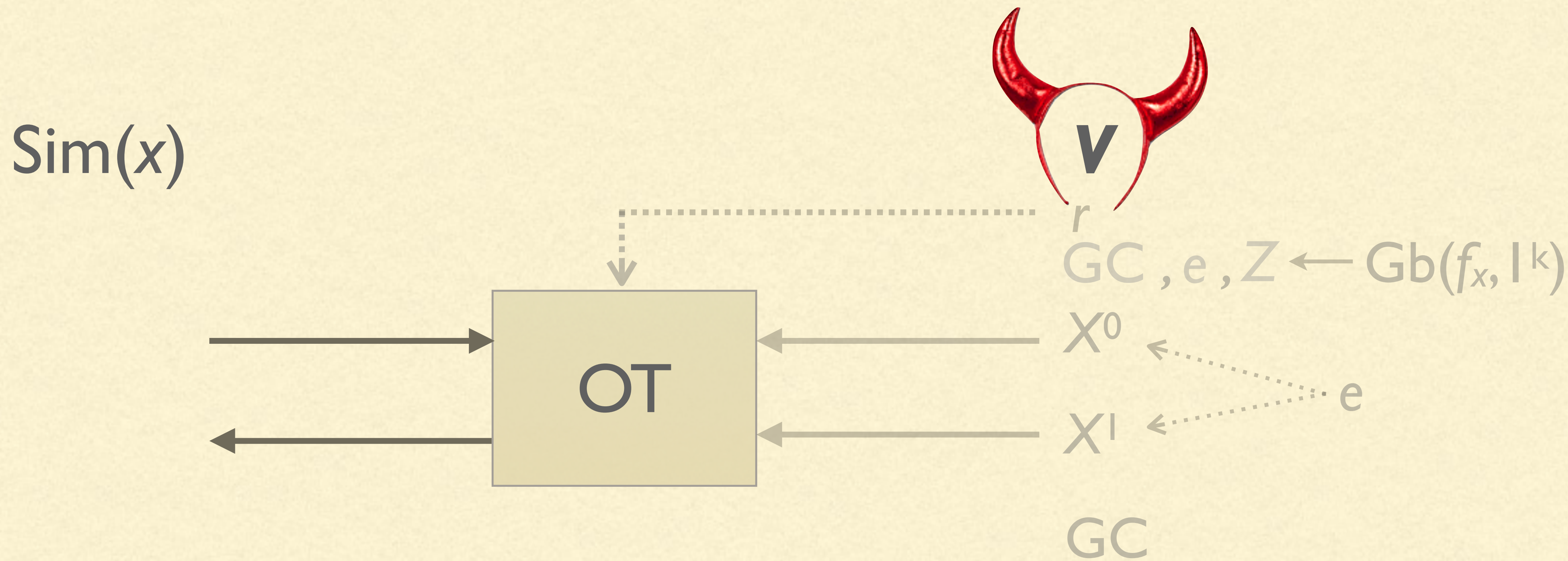
REDUCING TO (OT+1) ROUNDS

- Encryption scheme has to be "committing"
- Circular security and observability of decryption key: solved by random oracle (non-programmable)
- Still simulatable; less work for prover, verifier is the same

ADAPTIVE SECURITY

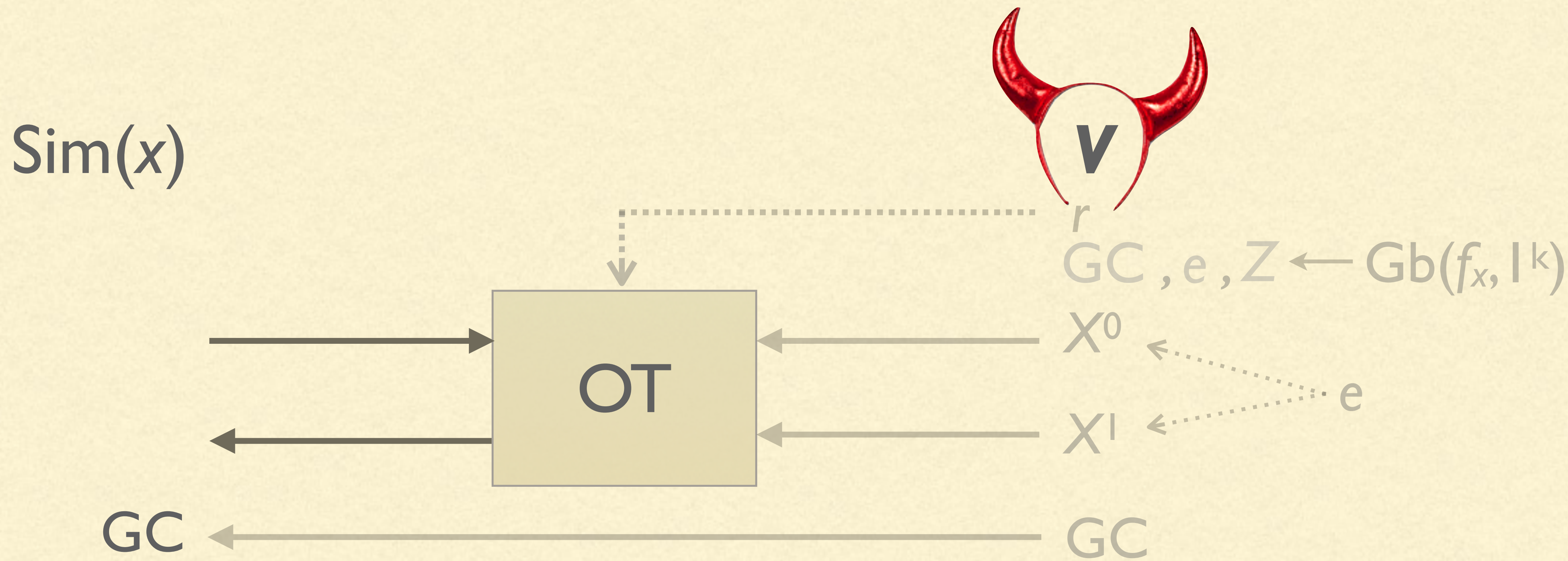
- Many cases of corruption order to consider
- Hardest case is when Verifier is initially corrupted, and Prover is corrupted post-execution. We focus on this case.
- For simplicity, we focus on adaptivity of OT+1 version

HARDEST CASE: ***V*** , ***P***



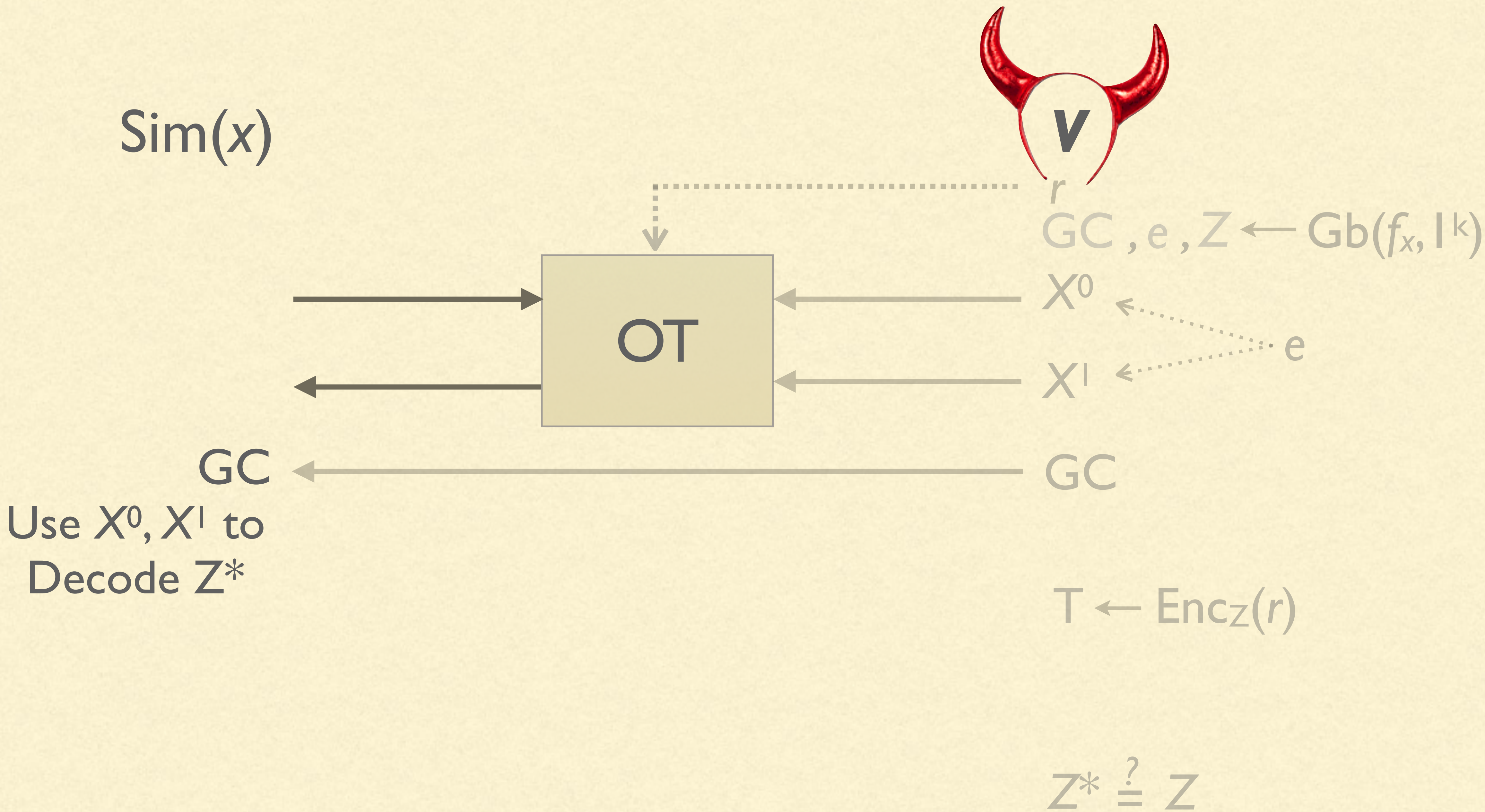
As with [JKO13], simulator extracts X^0, X^1 from OT.

HARDEST CASE: ***V***, ***P***

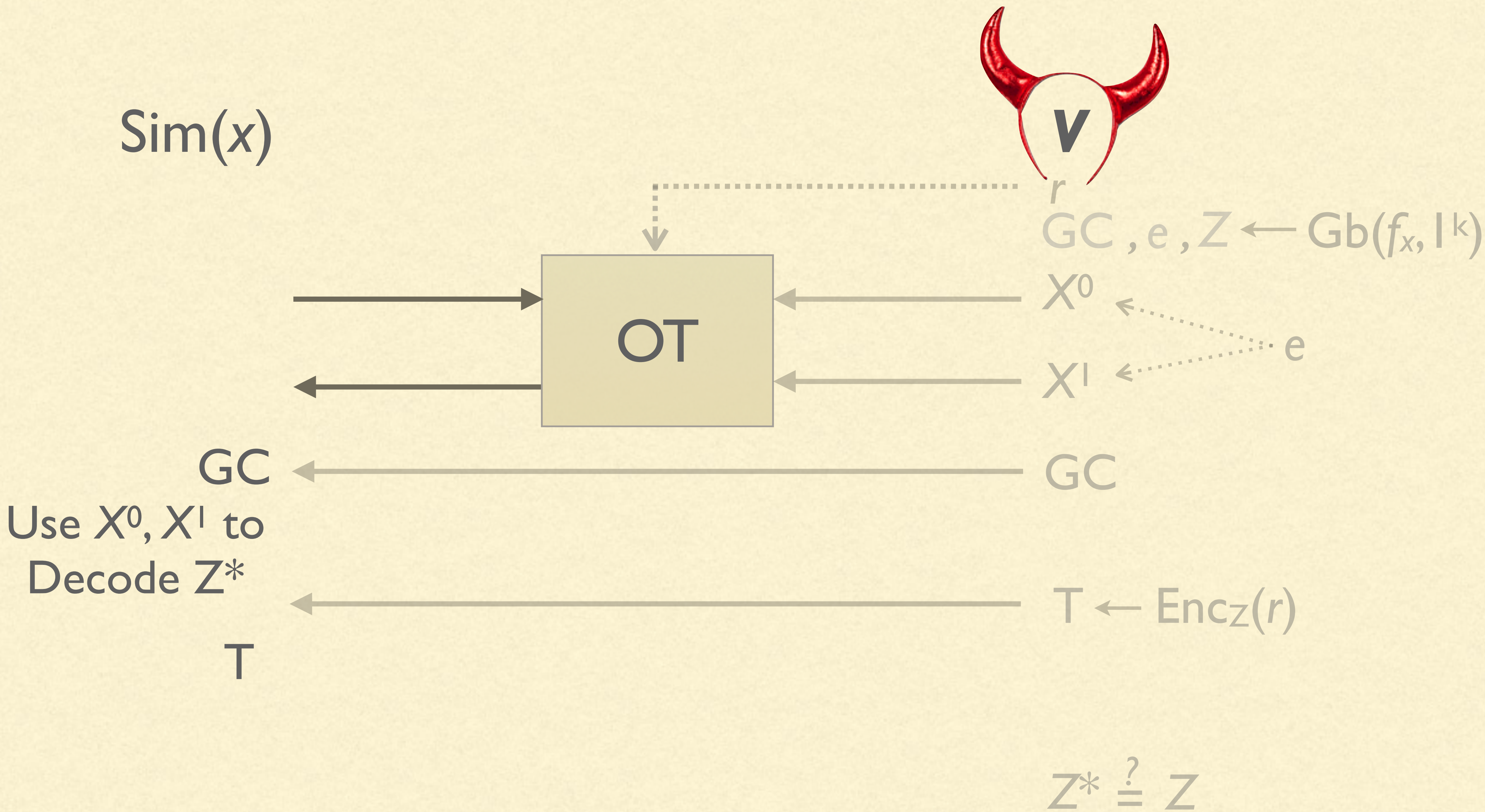


As with [JKO13], simulator extracts X^0, X^1 from OT.

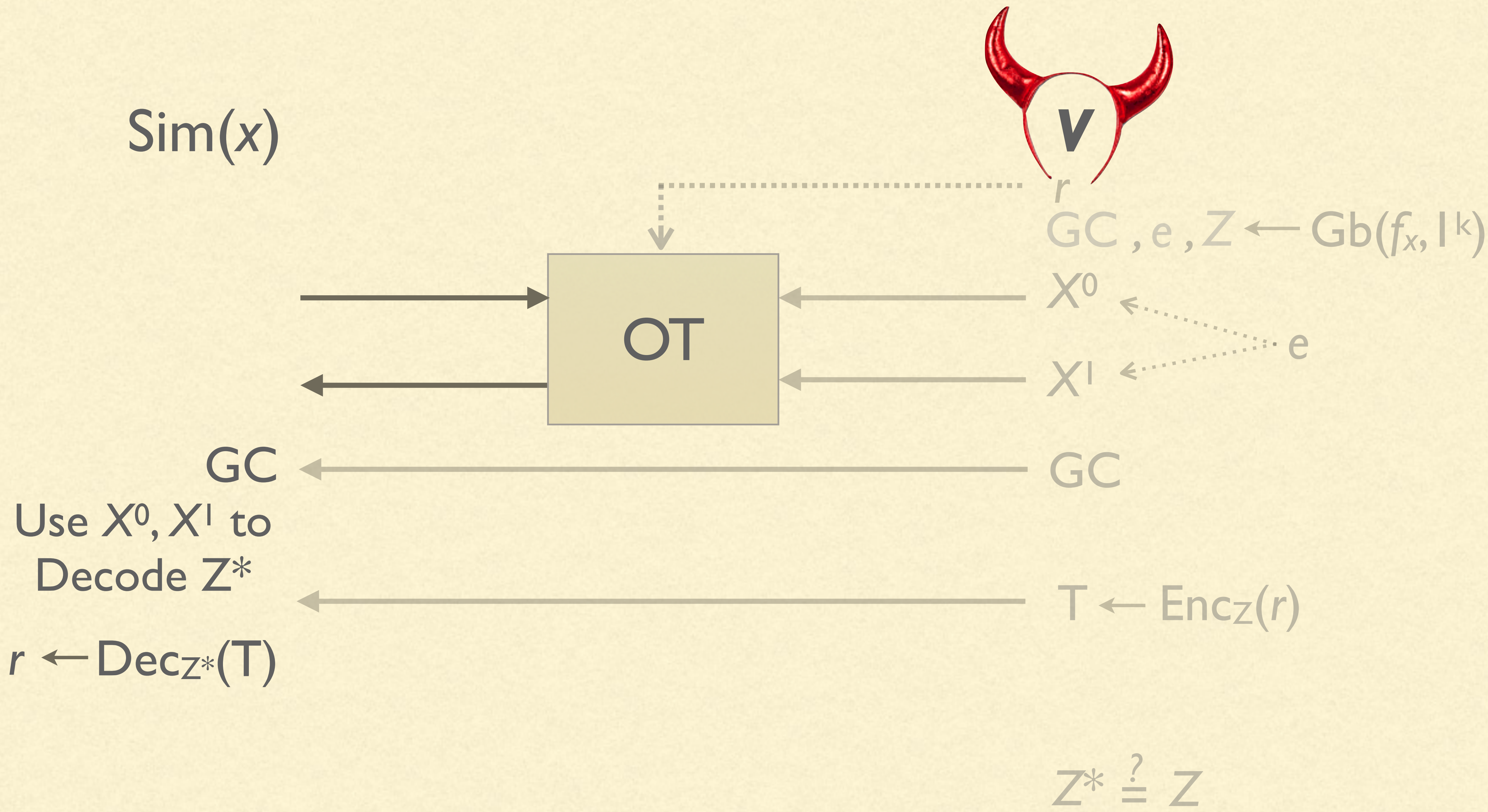
HARDEST CASE: *V*, *P*



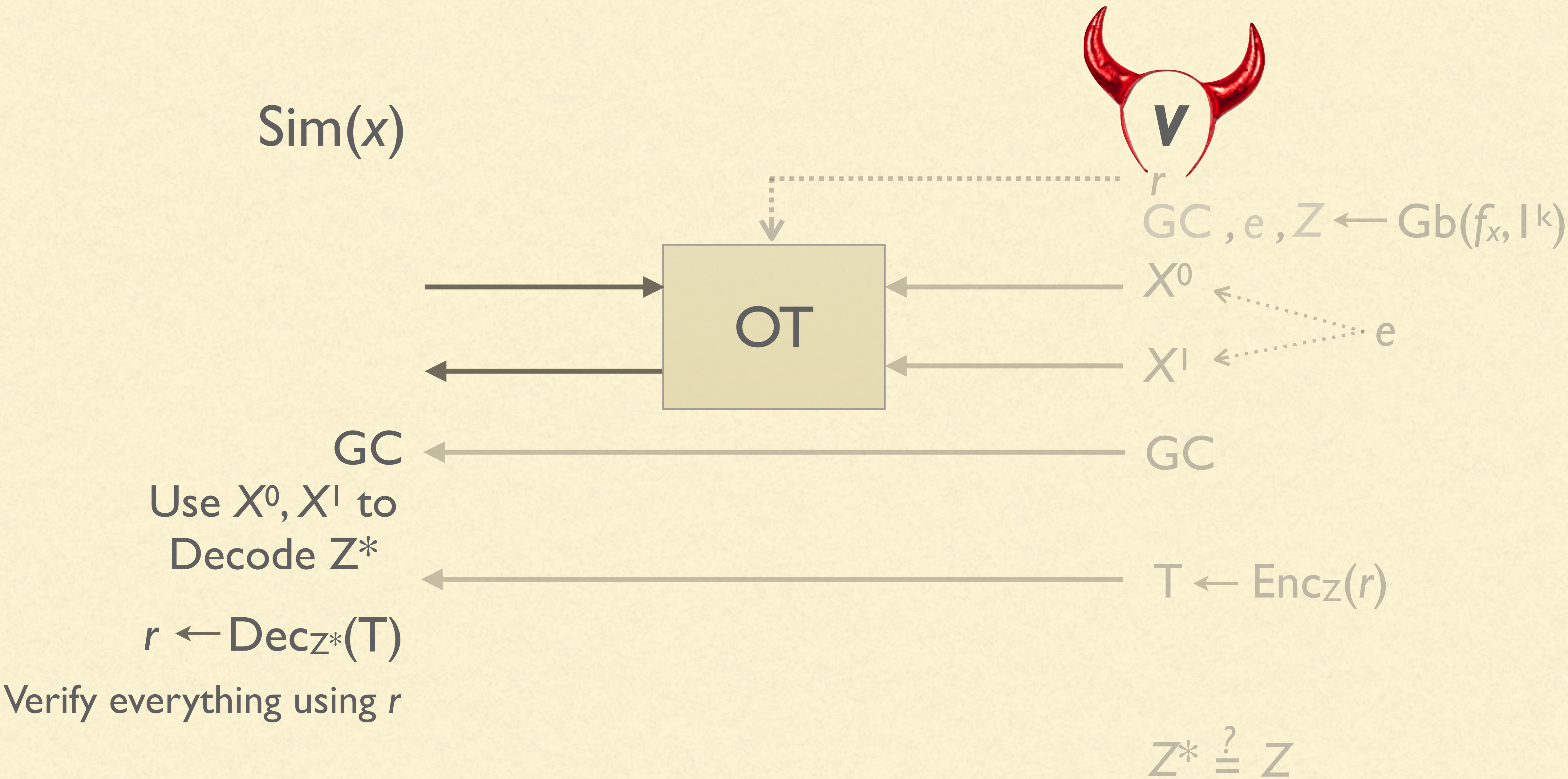
HARDEST CASE: ***V*** , ***P***



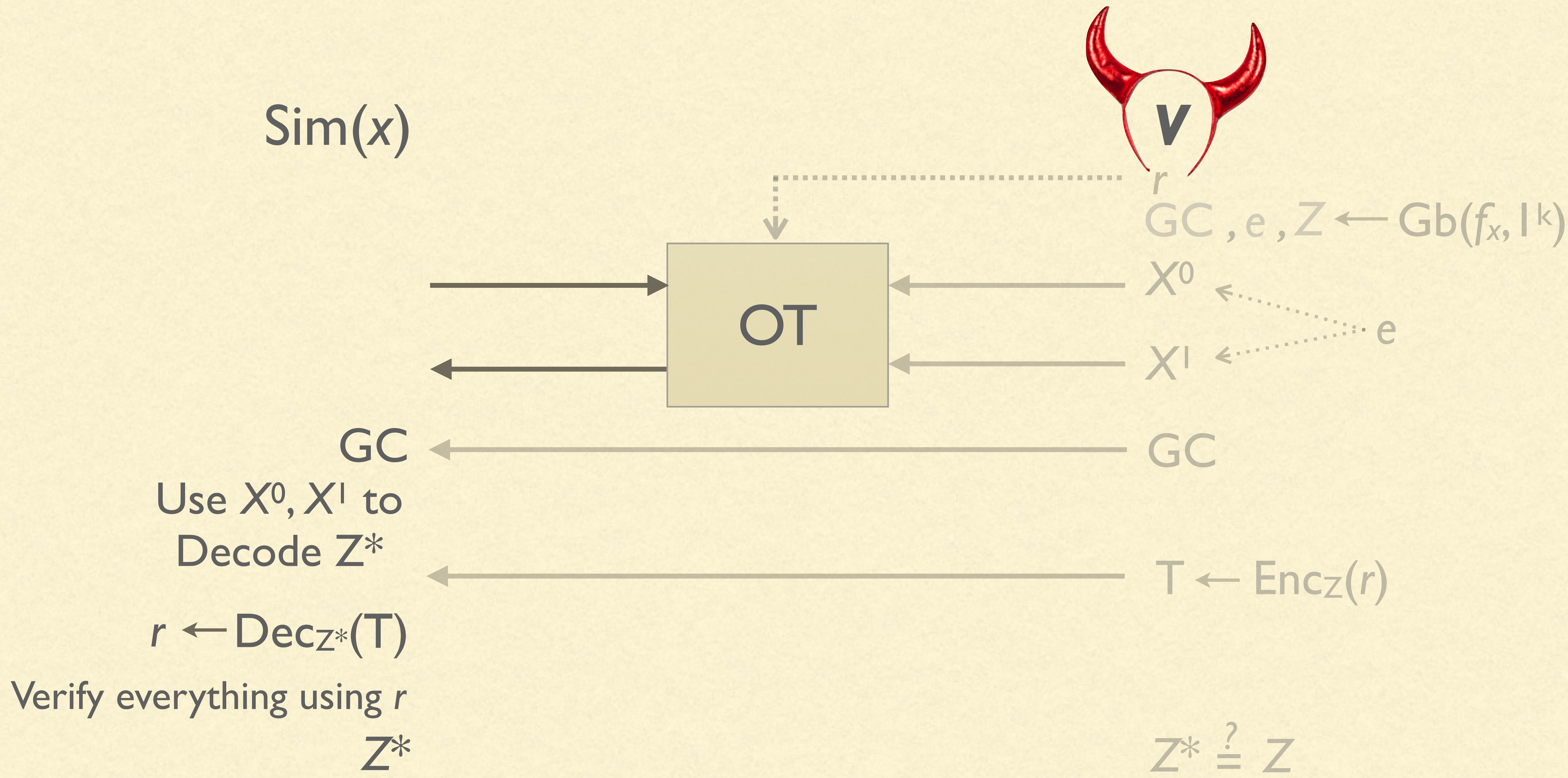
HARDEST CASE: *V*, *P*



HARDEST CASE: *V*, *P*



HARDEST CASE: *V* , *P*



HARDEST CASE: *V*, *P*

Sim(x)

V

r

GC, e, $Z \leftarrow \mathrm{Gb}(f_x, 1^k)$

OT

X^0

e

X^1

GC

GC

Use X^0, X^1 to Decode Z*

$T \leftarrow \mathrm{Enc}_Z(r)$

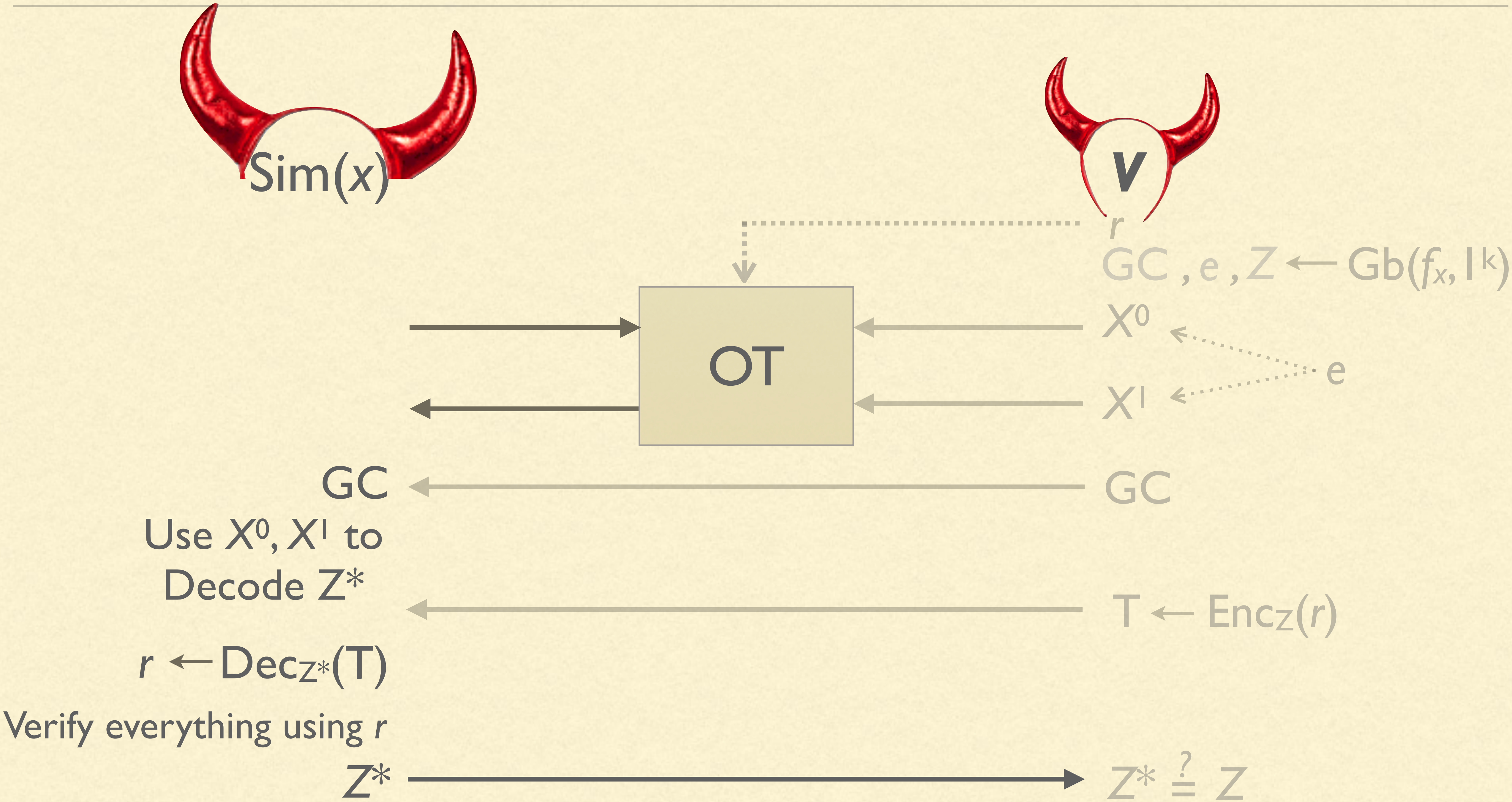
$r \leftarrow \mathrm{Dec}_{Z*}(T)$

Verify everything using r

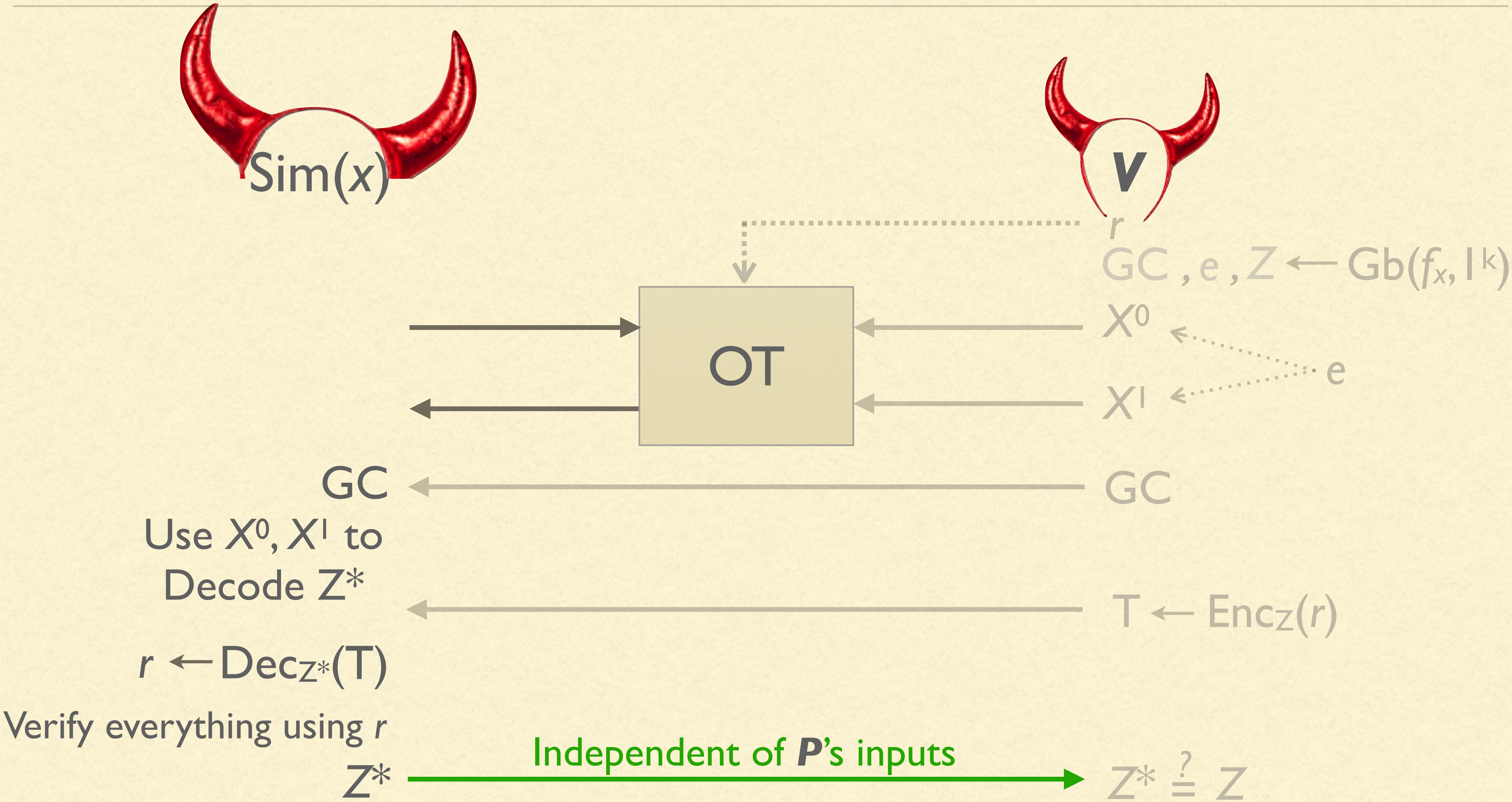
Z^*

$Z^* \stackrel{?}{=} Z$

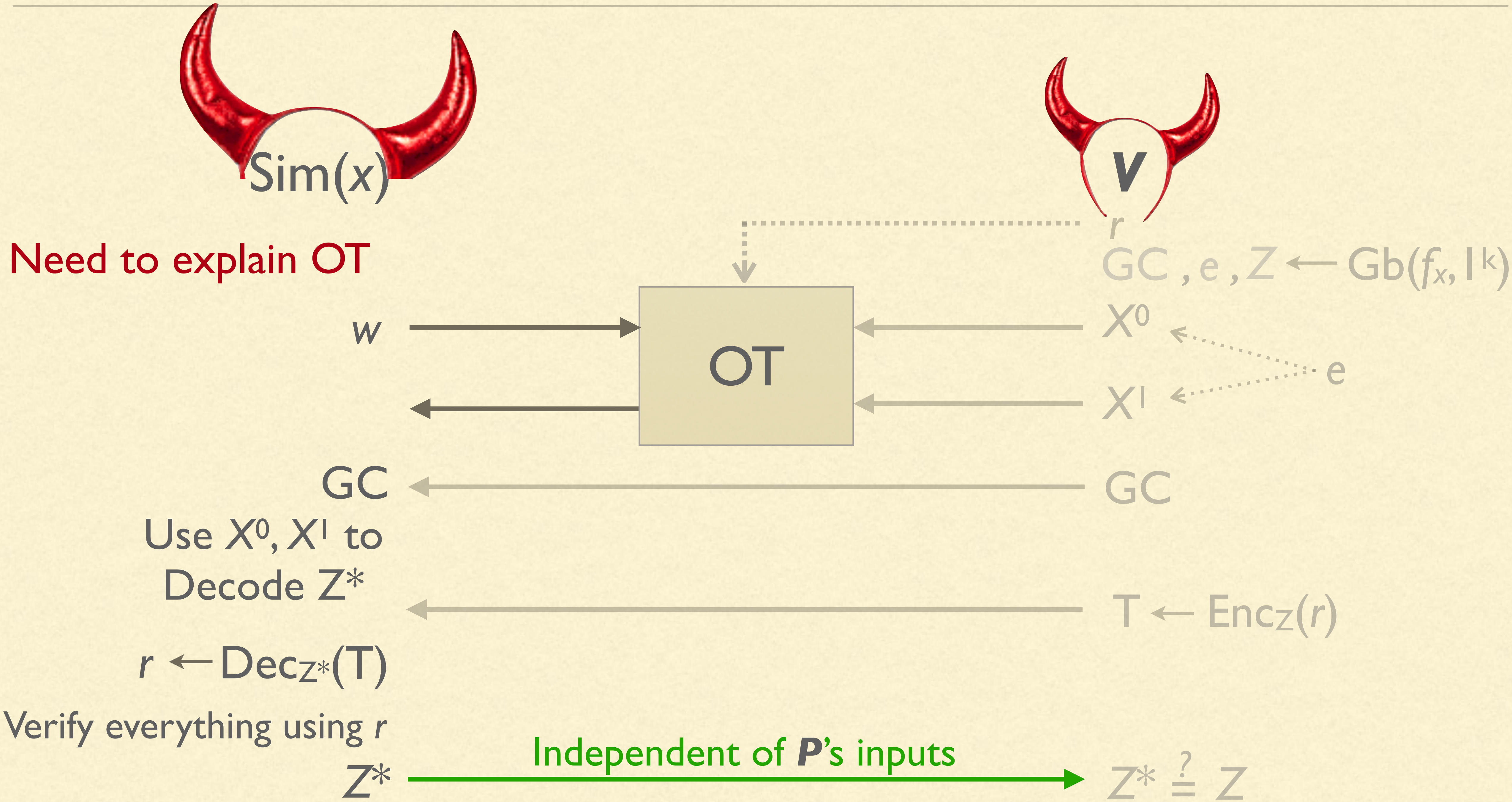
HARDEST CASE: ***V***, ***P***



HARDEST CASE: *V* , *P*

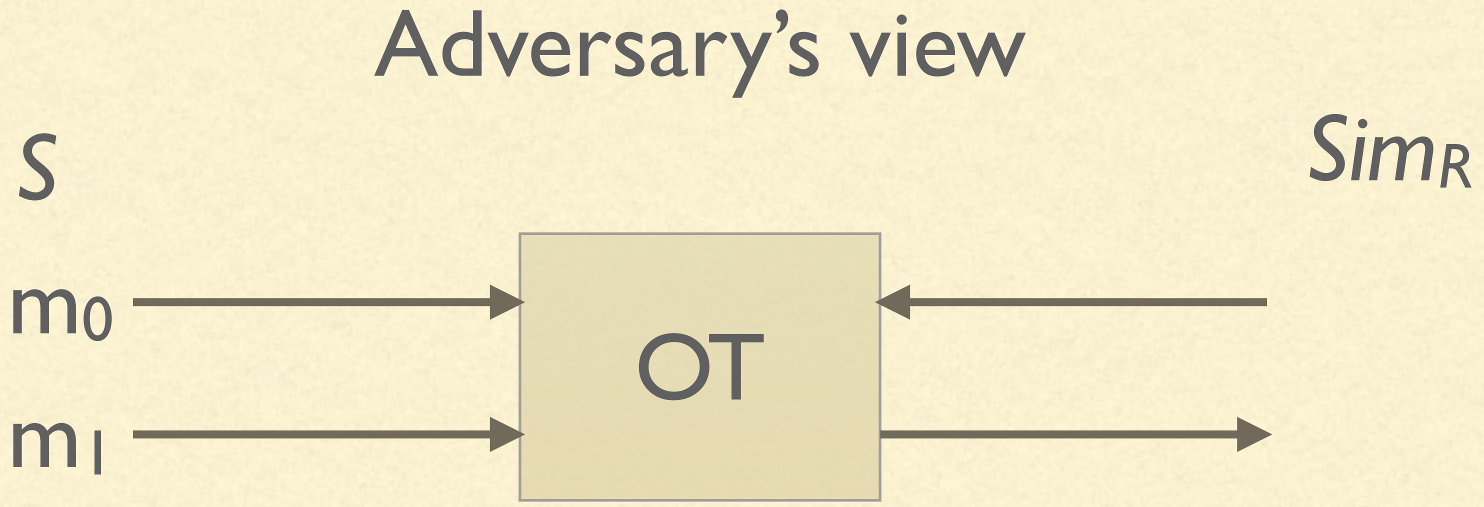


HARDEST CASE: V, P



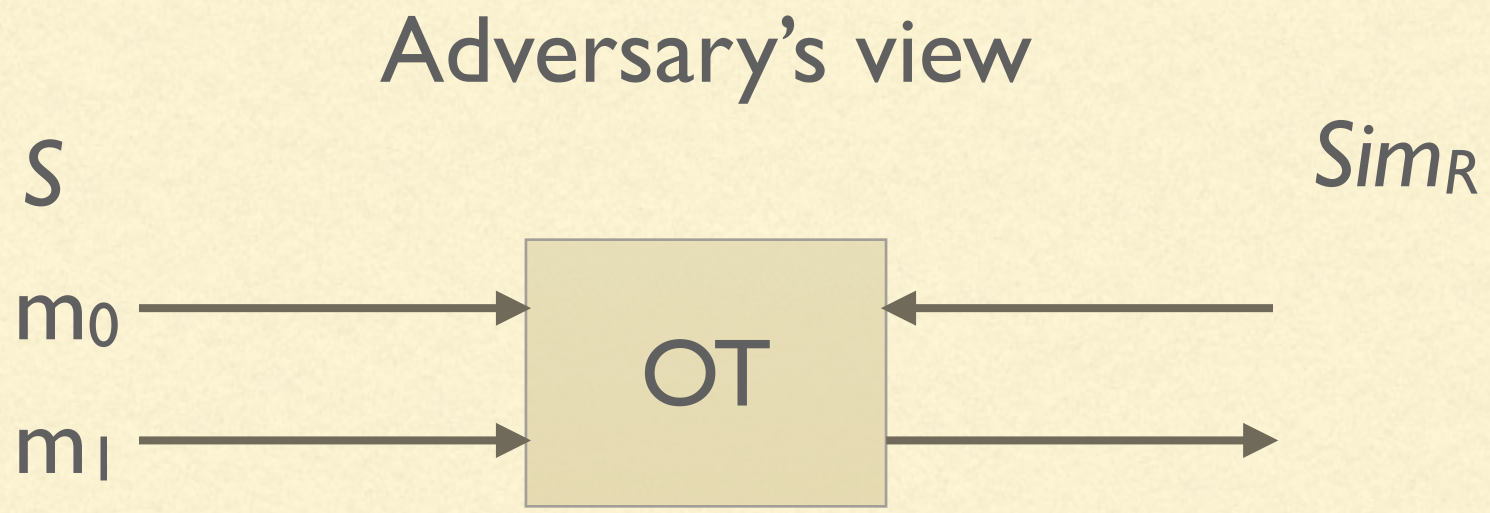
RECEIVER EQUIVOCAL OT

- Notion introduced in [Beaver 96]



RECEIVER EQUIVOCAL OT

- Notion introduced in [Beaver 96]

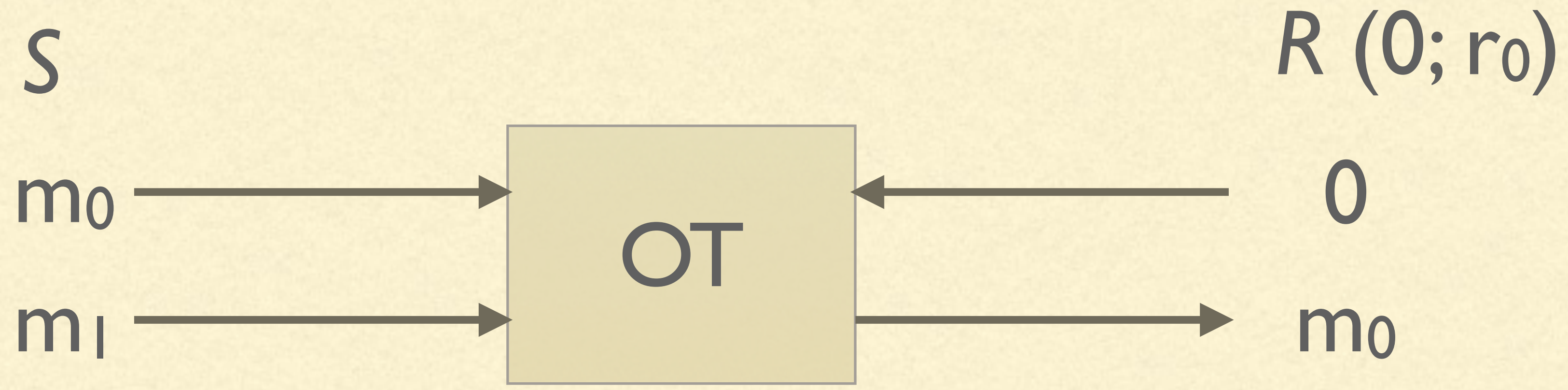


- Sim_R is able to produce r_0 and r_1

RECEIVER EQUIVOCAL OT

- Notion introduced in [Beaver 96]

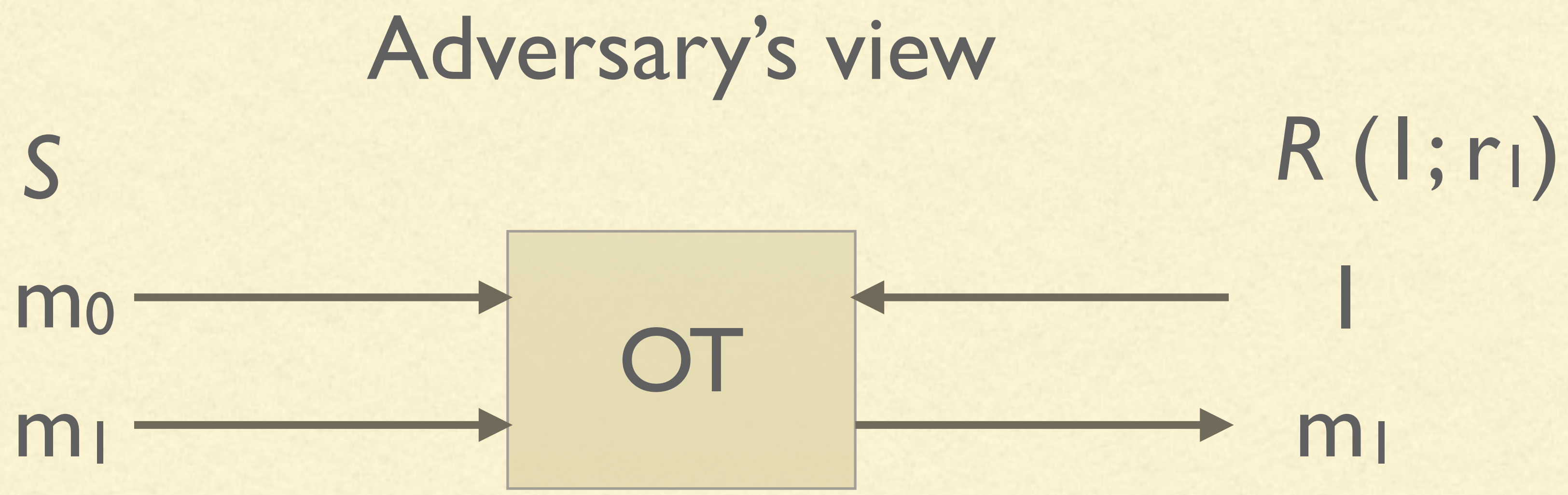
Adversary's view



- Sim_R is able to produce r_0 and r_1

RECEIVER EQUIVOCAL OT

- Notion introduced in [Beaver 96]



- Sim_R is able to produce r_0 and r_1

RECEIVER EQUIVOCAL OT

RECEIVER EQUIVOCAL OT

- The OT framework of [Peikert-Vaikuntanathan-Waters 08] satisfies one-sided adaptivity (but not full adaptivity)

RECEIVER EQUIVOCAL OT

- The OT framework of [Peikert-Vaikuntanathan-Waters 08] satisfies one-sided adaptivity (but not full adaptivity)
- Instantiable under any of DDH, LWE, LPN, QR assumptions

RECEIVER EQUIVOCAL OT

- The OT framework of [Peikert-Vaikuntanathan-Waters 08] satisfies one-sided adaptivity (but not full adaptivity)
- Instantiable under any of DDH, LWE, LPN, QR assumptions
- UC-secure with local CRS as trusted setup

HARDEST CASE: ***V*** , ***P***

Sim(x , w):

V

r

$GC, e, Z \leftarrow Gb(f_x, 1^k)$

RE-OT

X^0

e

X^1

GC

GC

Use X^0, X^1 to Decode Z^*

$T \leftarrow Enc_Z(r)$

$r \leftarrow Dec_{Z^*}(T)$

Verify everything using r

Independent of ***P***'s inputs

Z^*

$Z^* \stackrel{?}{=} Z$

HARDEST CASE: ***V*** , ***P***

Sim(x , w): ***V***

r

$GC , e , Z \leftarrow Gb(f_x, 1^k)$

RE-OT

X^0

X^1

e

GC ← GC

Use X^0, X^1 to
Decode Z^*

← $T \leftarrow Enc_Z(r)$

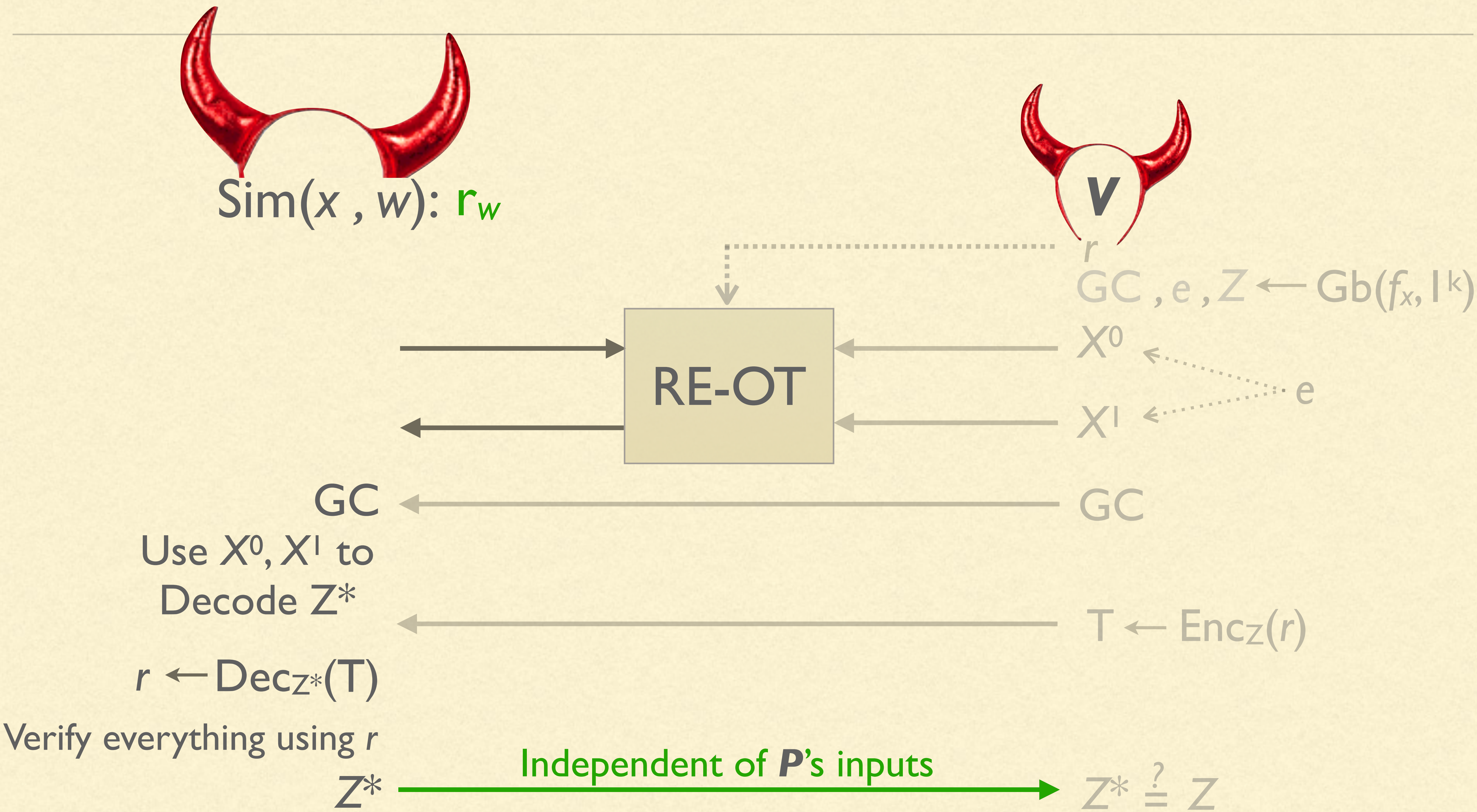
$r \leftarrow Dec_{Z^*}(T)$

Verify everything using r

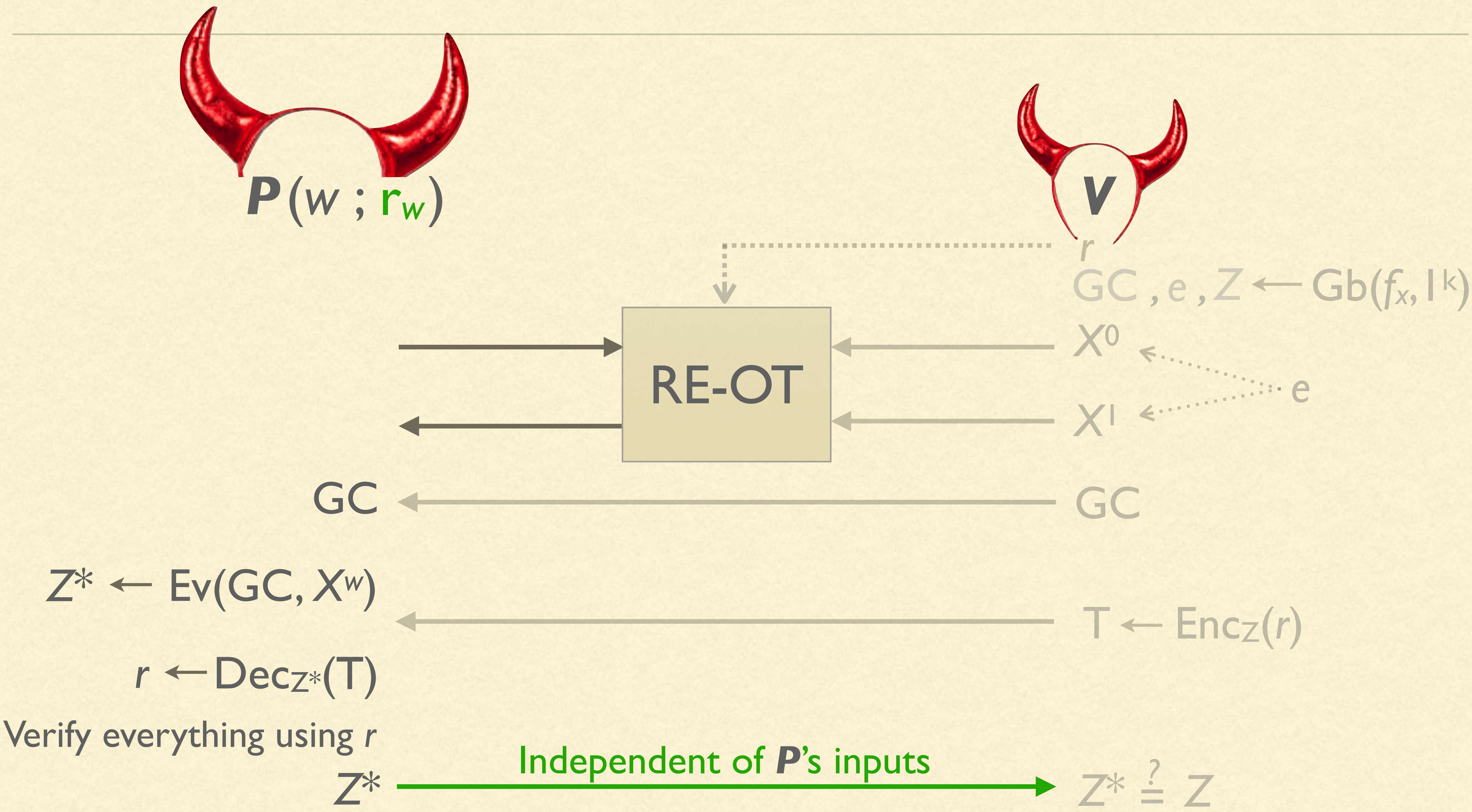
Independent of ***P***'s inputs

$Z^* \longrightarrow Z^* \stackrel{?}{=} Z$

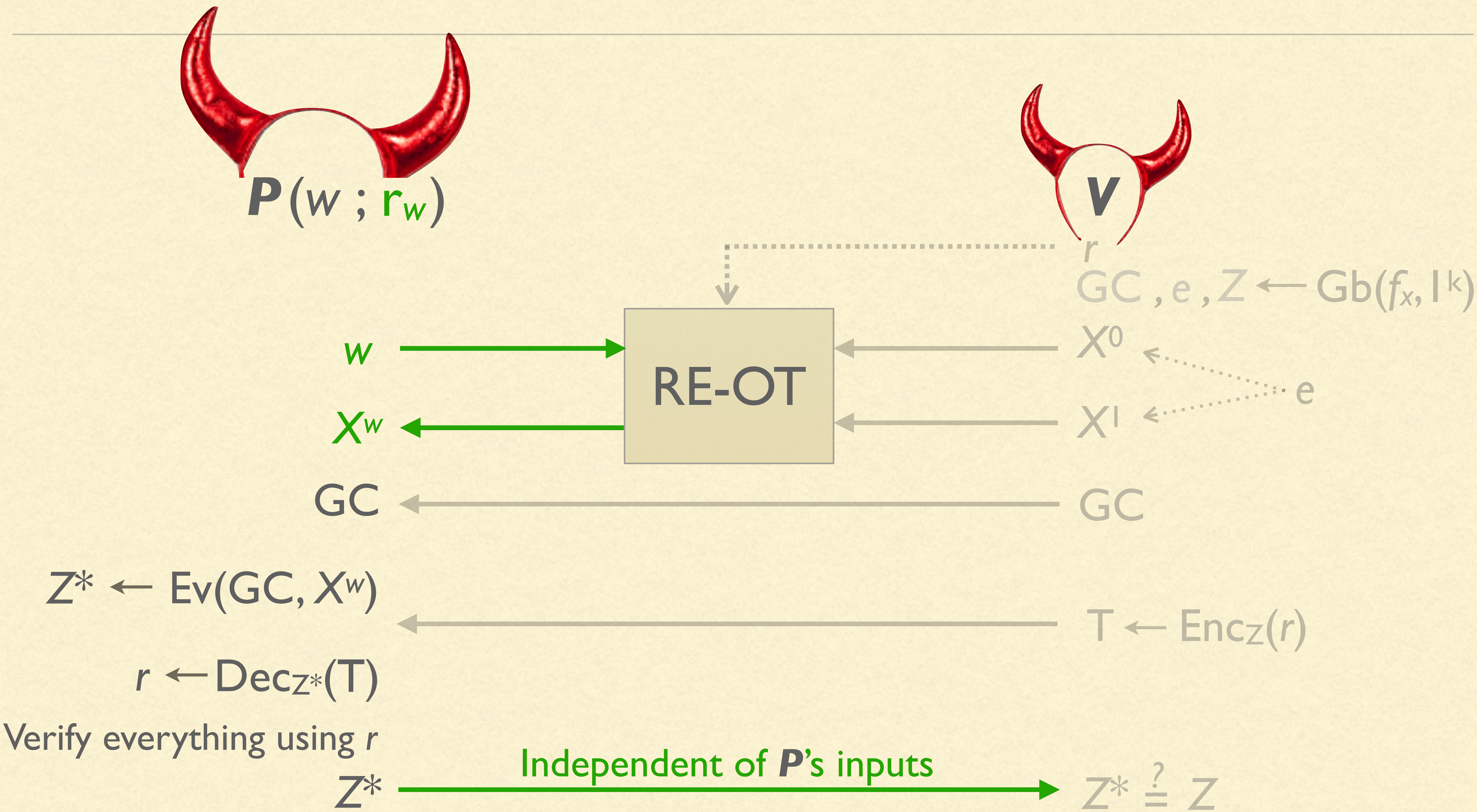
HARDEST CASE: ***V*** , ***P***



HARDEST CASE: *V* , *P*



HARDEST CASE: ***V*** , ***P***



SUMMARY

SUMMARY

- We construct ZK secure against adaptive corruptions

SUMMARY

- We construct ZK secure against adaptive corruptions
- Universally composable (in RE-OT hybrid)

SUMMARY

- We construct ZK secure against adaptive corruptions
- Universally composable (in RE-OT hybrid)
- Instantiable under DDH, LWE, LPN, QR

SUMMARY

- We construct ZK secure against adaptive corruptions
- Universally composable (in RE-OT hybrid)
- Instantiable under DDH, LWE, LPN, QR
- Three rounds (RE-OT+1) with global RO, linear communication

SUMMARY

- We construct ZK secure against adaptive corruptions
- Universally composable (in RE-OT hybrid)
- Instantiable under DDH, LWE, LPN, QR
- Three rounds (RE-OT+1) with global RO, linear communication
- Concretely efficient: $|w|$ OTs + GC with $|f_x|$ gates

SUMMARY

- We construct ZK secure against adaptive corruptions
- Universally composable (in RE-OT hybrid)
- Instantiable under DDH, LWE, LPN, QR
- Three rounds (RE-OT+1) with global RO, linear communication
- Concretely efficient: $|w|$ OTs + GC with $|f_x|$ gates
- Also in paper: 2-round ZK proofs from GCs (cut and choose), authenticity-free garbling

THANK YOU