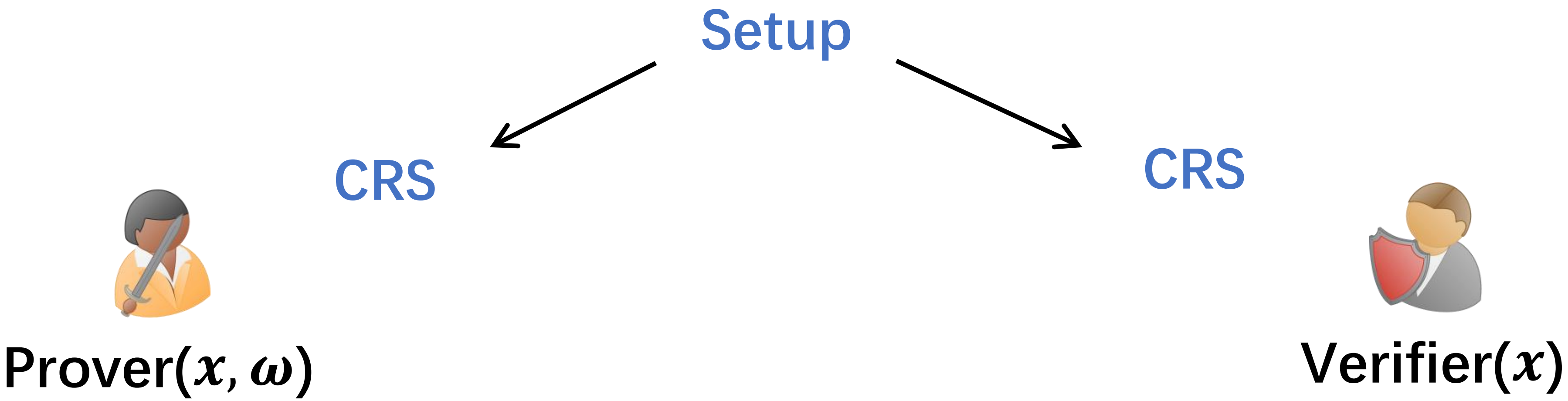
Non-Interactive Zero Knowledge from Sub-exponential DDH

Abhishek Jain **Zhengzhong Jin**

JOHNS HOPKINS UNIVERSITY

Non-Interactive Zero Knowledge (NIZK)



Non-Interactive Zero Knowledge (NIZK)

Setup

CRS

“$x \in L$”

CRS

Prover(x, ω)

Verifier(x)

Non-Interactive Zero Knowledge (NIZK)

Setup

CRS

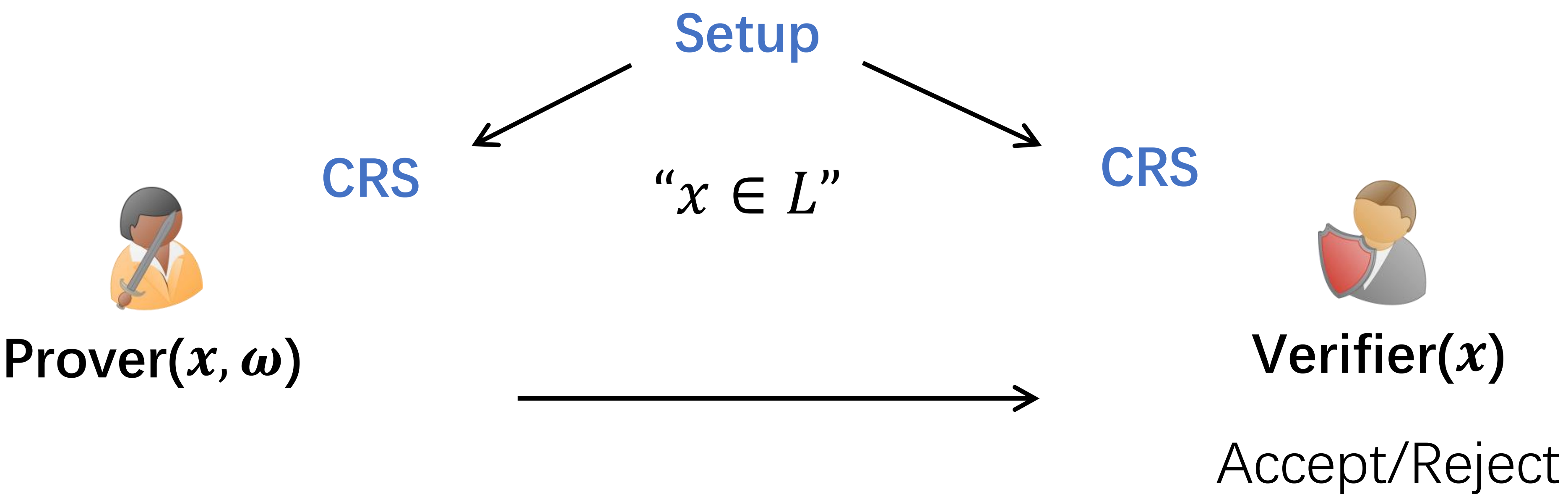
“$x \in L$”

CRS

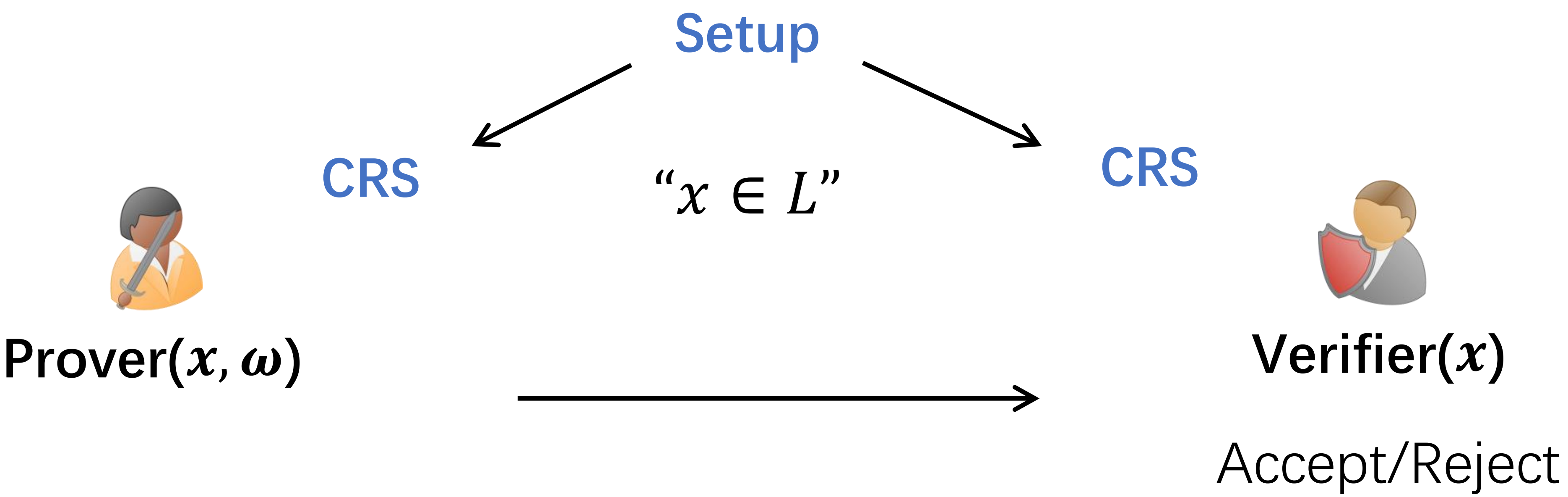
Prover(x, ω)

Verifier(x)

Non-Interactive Zero Knowledge (NIZK)

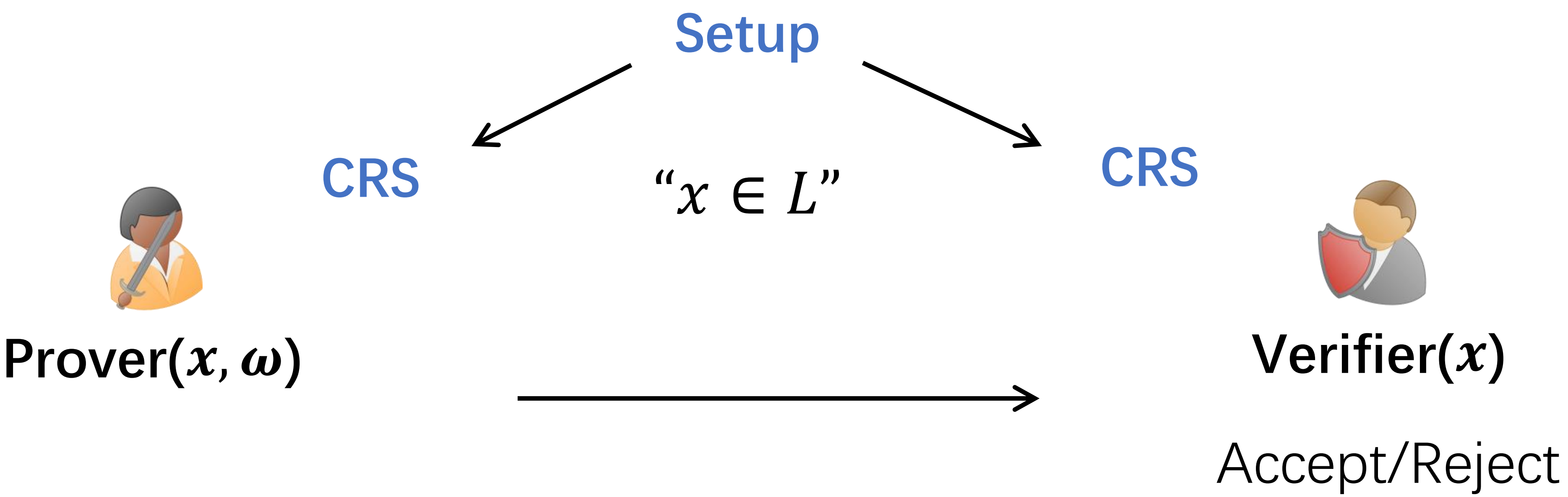


Non-Interactive Zero Knowledge (NIZK)



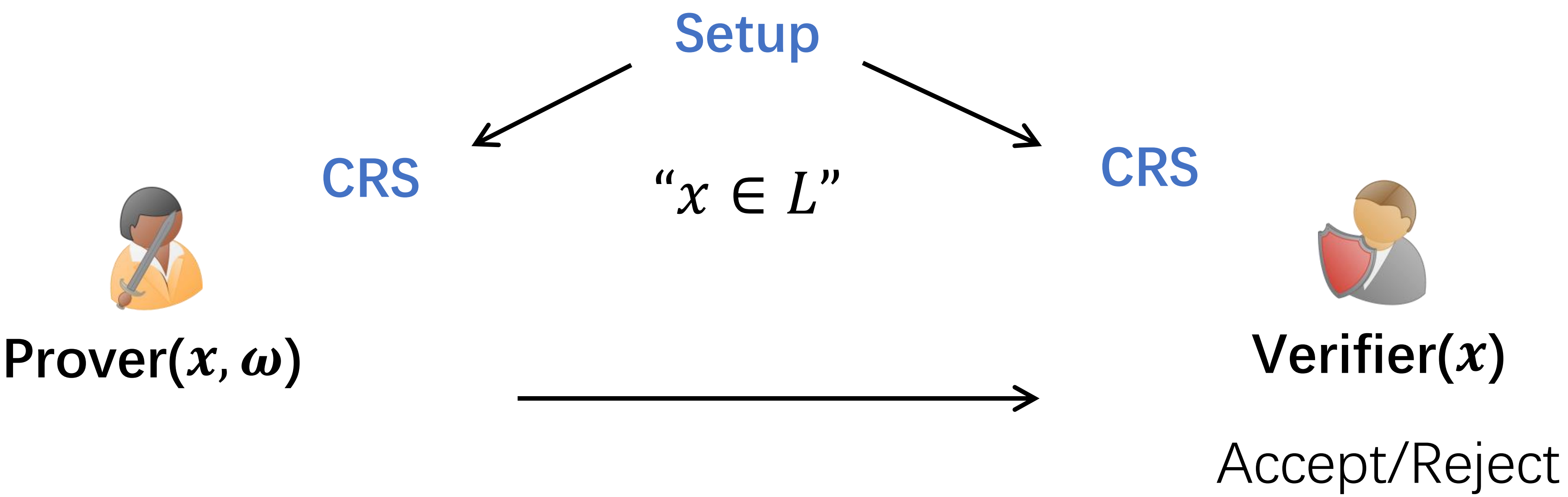
- **Completeness:** If $x \in L$, verifier accepts the honestly generated proof.
- **Soundness:** for any $x \notin L$, the verifier rejects.
- **Zero-Knowledge:** the proof reveals nothing beyond $x \in L$.

Non-Interactive Zero Knowledge (NIZK)



- **Completeness:** If $x \in L$, verifier accepts the honestly generated proof.
- **Soundness:** for any $x \notin L$, the verifier rejects.
- **Zero-Knowledge:** the proof reveals nothing beyond $x \in L$.

Non-Interactive Zero Knowledge (NIZK)



- **Completeness:** If $x \in L$, verifier accepts the honestly generated proof.
- **Soundness:** for any $x \notin L$, the verifier rejects.
- **Zero-Knowledge:** the proof reveals nothing beyond $x \in L$.

What assumptions are sufficient for NIZKs?

Prior Works

Prior Works

- Quadratic Residuosity Assumption (QR) [BFM88]
- Factoring [FLS90]
- Bilinear Maps [CHK03, GOS06, GOS06]
- Learning with Errors (LWE) [CCHLRRW19, PS19]
- Learning Parity with Noise and Trapdoor Hash Function [BKM20] (Trapdoor Hash Function is known from DDH/LWE/QR/DCR)

Prior Works

- Quadratic Residuosity Assumption (QR) [BFM88]
- Factoring [FLS90]
- Bilinear Maps [CHK03, GOS06, GOS06]
- Learning with Errors (LWE) [CCHLRRW19, PS19]
- Learning Parity with Noise and Trapdoor Hash Function [BKM20] (Trapdoor Hash Function is known from DDH/LWE/QR/DCR)

Prior Works

- Quadratic Residuosity Assumption (QR) [BFM88]
- Factoring [FLS90]
- Bilinear Maps [CHK03, GOS06, GOS06]
- Learning with Errors (LWE) [CCHLRRW19, PS19]
- Learning Parity with Noise and Trapdoor Hash Function [BKM20] (Trapdoor Hash Function is known from DDH/LWE/QR/DCR)

Prior Works

- Quadratic Residuosity Assumption (QR) [BFM88]
- Factoring [FLS90]
- Bilinear Maps [CHK03, GOS06, GOS06]
- Learning with Errors (LWE) [CCHLRRW19, PS19]
- Learning Parity with Noise and Trapdoor Hash Function [BKM20]
  (Trapdoor Hash Function is known from DDH/LWE/QR/DCR)

Prior Works

- Quadratic Residuosity Assumption (QR) [BFM88]
- Factoring [FLS90]
- Bilinear Maps [CHK03, GOS06, GOS06]
- Learning with Errors (LWE) [CCHLRRW19, PS19]
- Learning Parity with Noise and Trapdoor Hash Function [BKM20] (Trapdoor Hash Function is known from DDH/LWE/QR/DCR)

Prior Works

- Quadratic Residuosity Assumption (QR) [BFM88]
- Factoring [FLS90]
- Bilinear Maps [CHK03, GOS06, GOS06]
- Learning with Errors (LWE) [CCHLRRW19, PS19]
- Learning Parity with Noise and Trapdoor Hash Function [BKM20] (Trapdoor Hash Function is known from DDH/LWE/QR/DCR)
- NIZKs from discrete-log related assumptions?

Question (1): Do there exist NIZKs from DDH?

Pairing vs Non-pairing Groups

	Pairing	Non-Pairing
Attribute-Based Encryption	[SW04,GPSW06]	?
Identity-Based Encryption	[BF01]	[DG17]
NIZKs	[CHK03,GOS06]	?*

Pairing vs Non-pairing Groups

	Pairing	Non-Pairing
Attribute-Based Encryption	[SW04,GPSW06]	**?**
Identity-Based Encryption	[BF01]	[DG17]
NIZKs	[CHK03,GOS06]	**?***

Are the gaps inherent?

Pairing vs Non-pairing Groups

	Pairing	Non-Pairing
Attribute-Based Encryption	[SW04,GPSW06]	**?**
Identity-Based Encryption	[BF01]	[DG17]
NIZKs	[CHK03,GOS06]	**?***

Are the gaps inherent?

* From non-standard assumptions, NIZKs are known from non-pairing groups [CCRR18,CKU20]

Our Result (1):

Our Result (1):

- NIZK arguments for NP:

Our Result (1):

- NIZK arguments for NP:

	Zero-Knowledge	Soundness	CRS
I	Statistical	Non-adaptive	Random
II	Computational	Adaptive	Random

Our Result (1):

- NIZK arguments for NP:

	Zero-Knowledge	Soundness	CRS
I	Statistical	Non-adaptive	Random
II	Computational	Adaptive	Random

- From ***sub-exponential*** DDH in the standard non-pairing groups.

Sub-exponential DDH

- $\exists\ 0 < c < 1, \forall$ **non-uniform PPT adversary** D, $\forall$ sufficiently large λ,

$$|\Pr[D(1^\lambda, g, g^a, g^b, g^{ab}) = 1] - \Pr[D(1^\lambda, g, g^a, g^b, g^c) = 1]| < \mathbf{2^{-\lambda^c}}$$

$$a \leftarrow Z_p, b \leftarrow Z_p, c \leftarrow Z_p$$

Our Result (2):

Statistical Zap arguments from sub-exponential DDH, with non-adaptive soundness.

Our Result (2):

Statistical Zap arguments from sub-exponential DDH, with non-adaptive soundness.

Statistical Zaps from group-based assumptions were not known.

Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

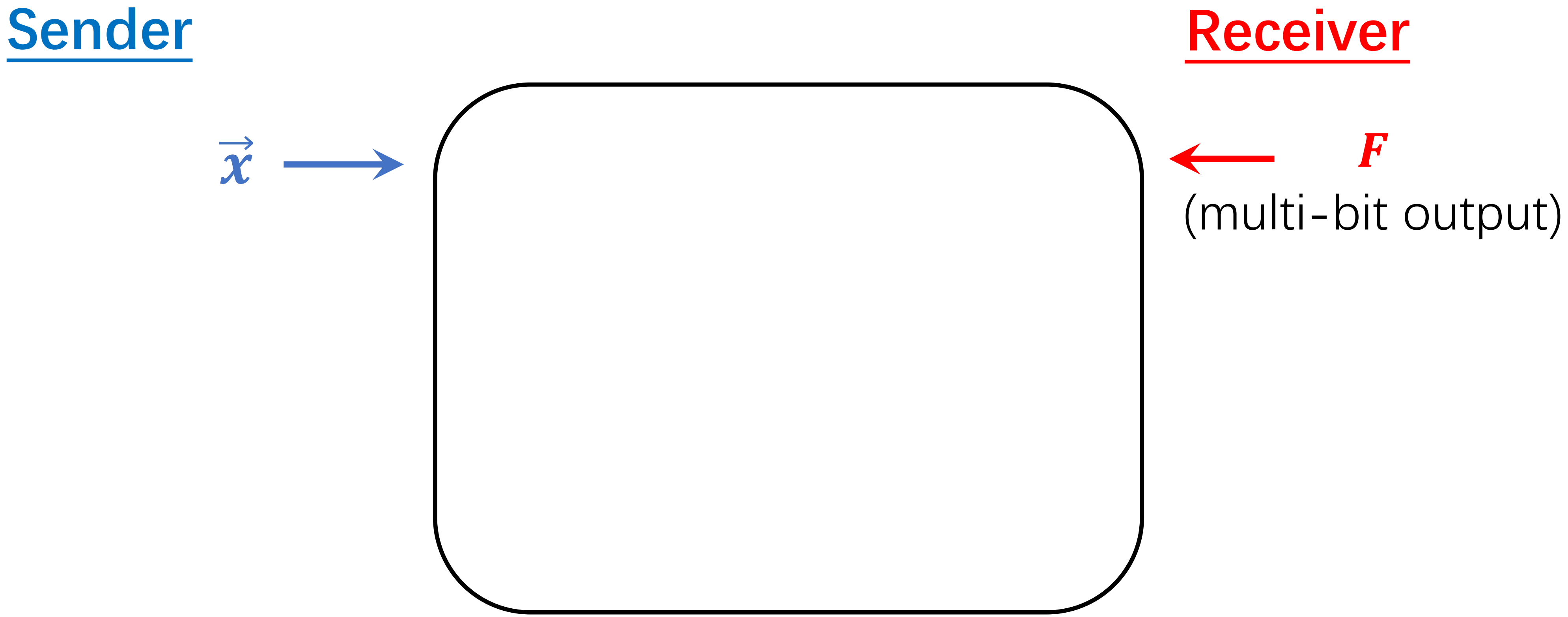
Sender

Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

Sender

Receiver

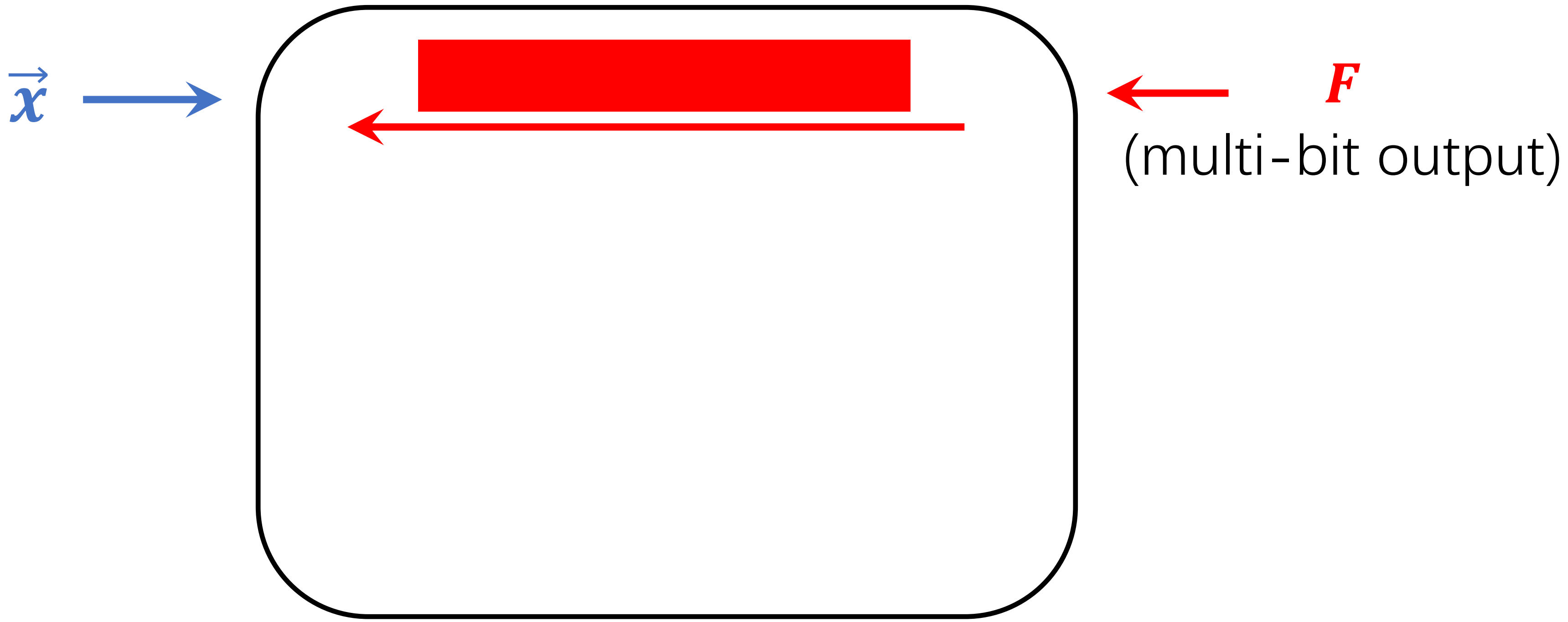
Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)



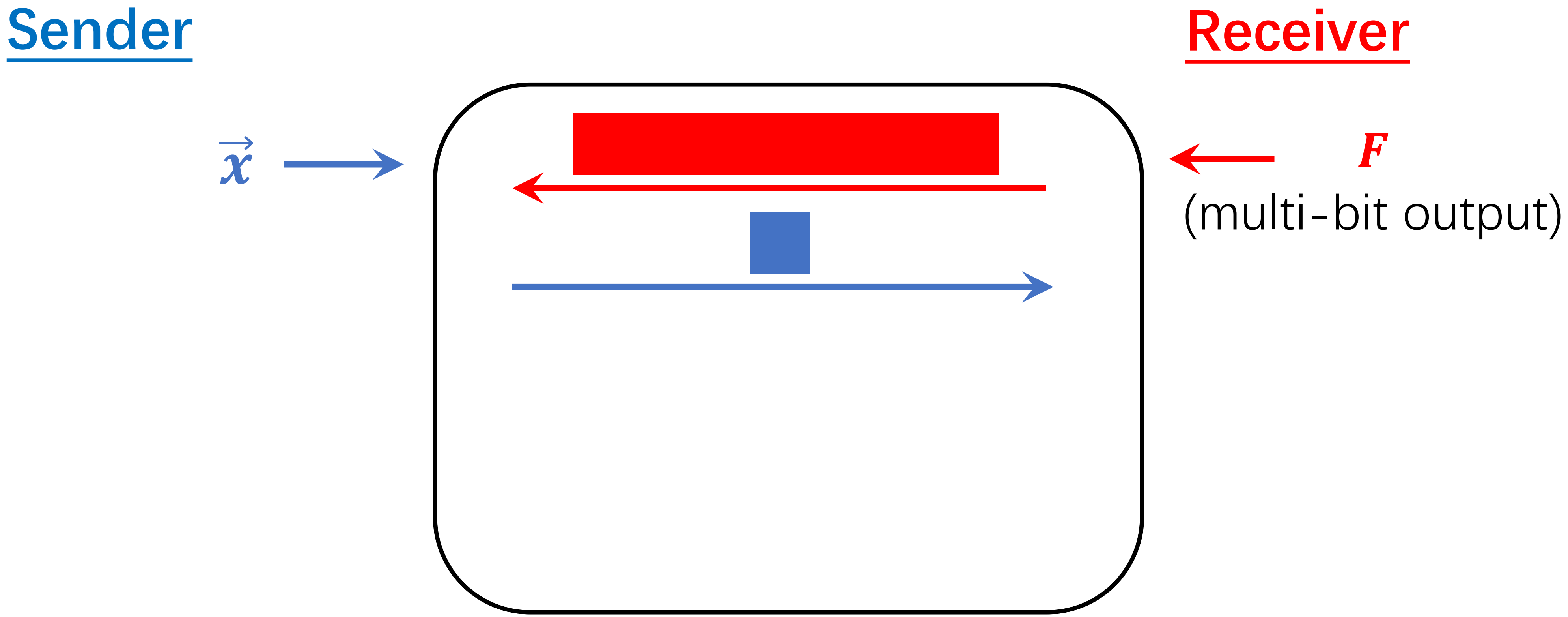
Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

Sender

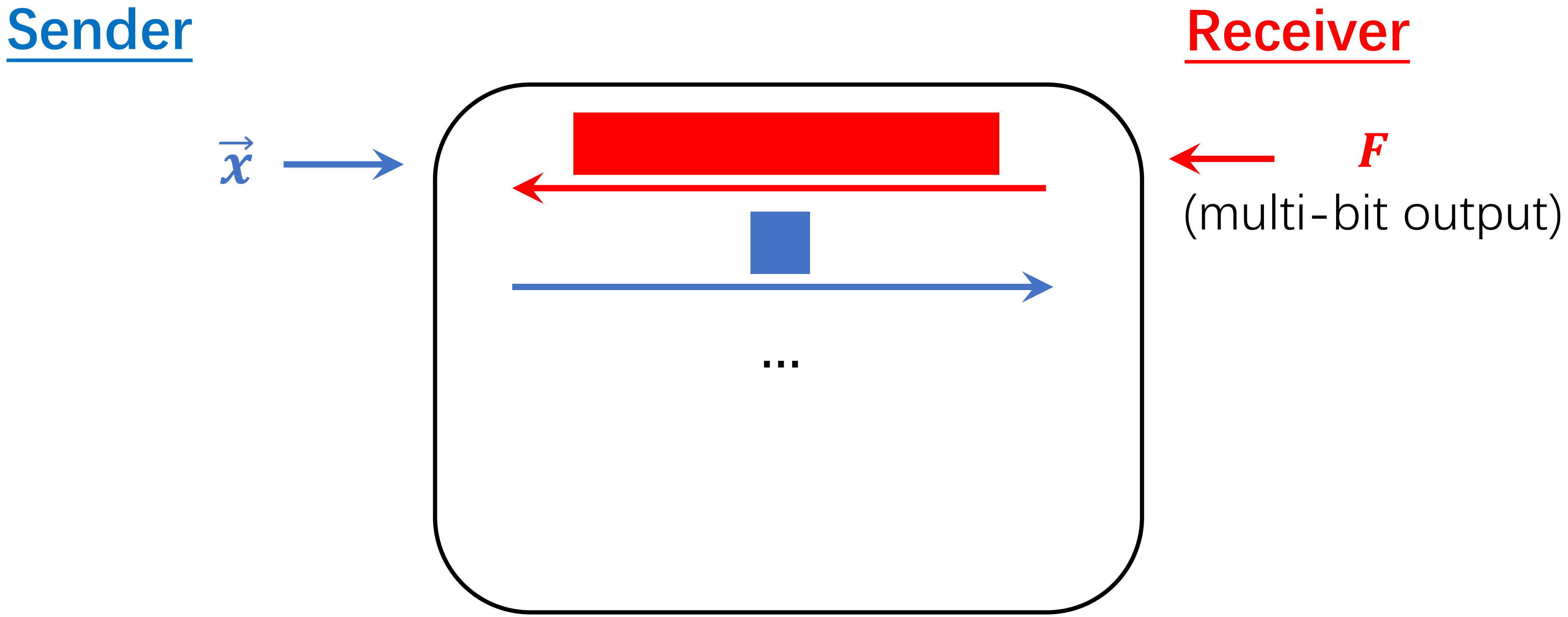
Receiver



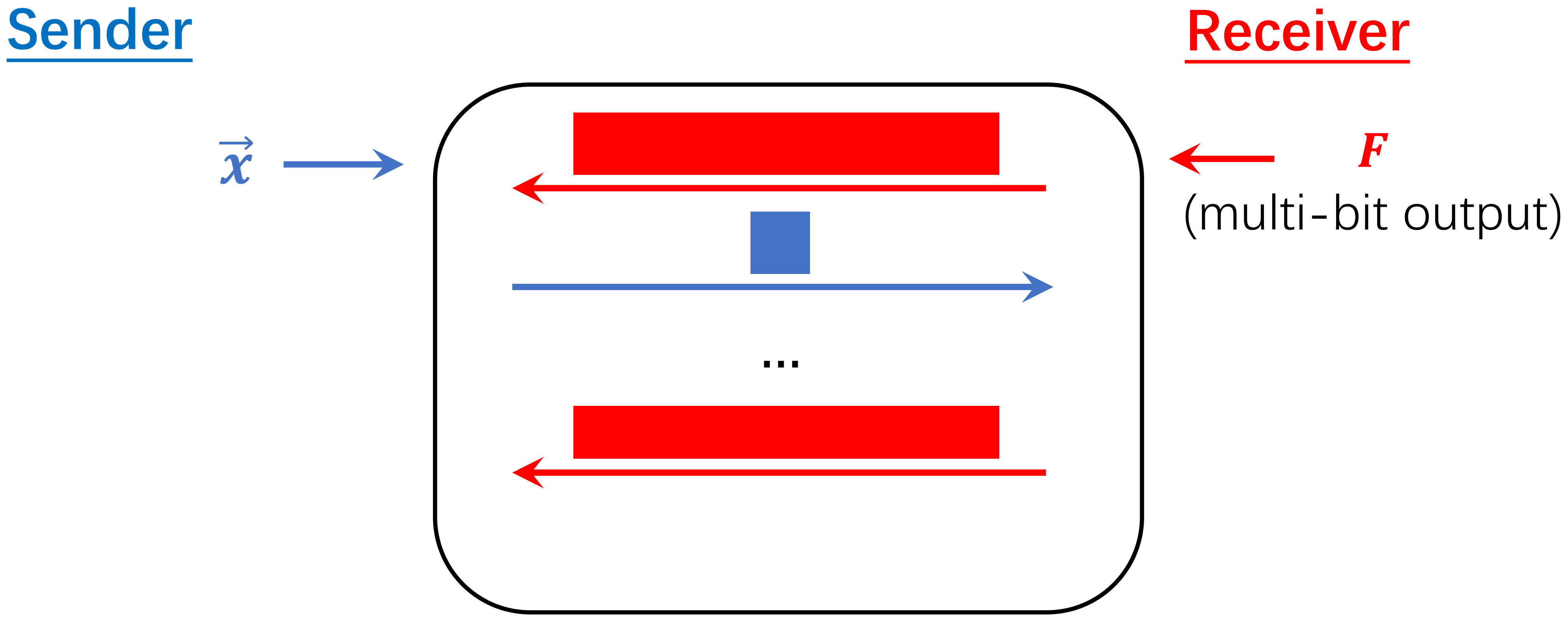
Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)



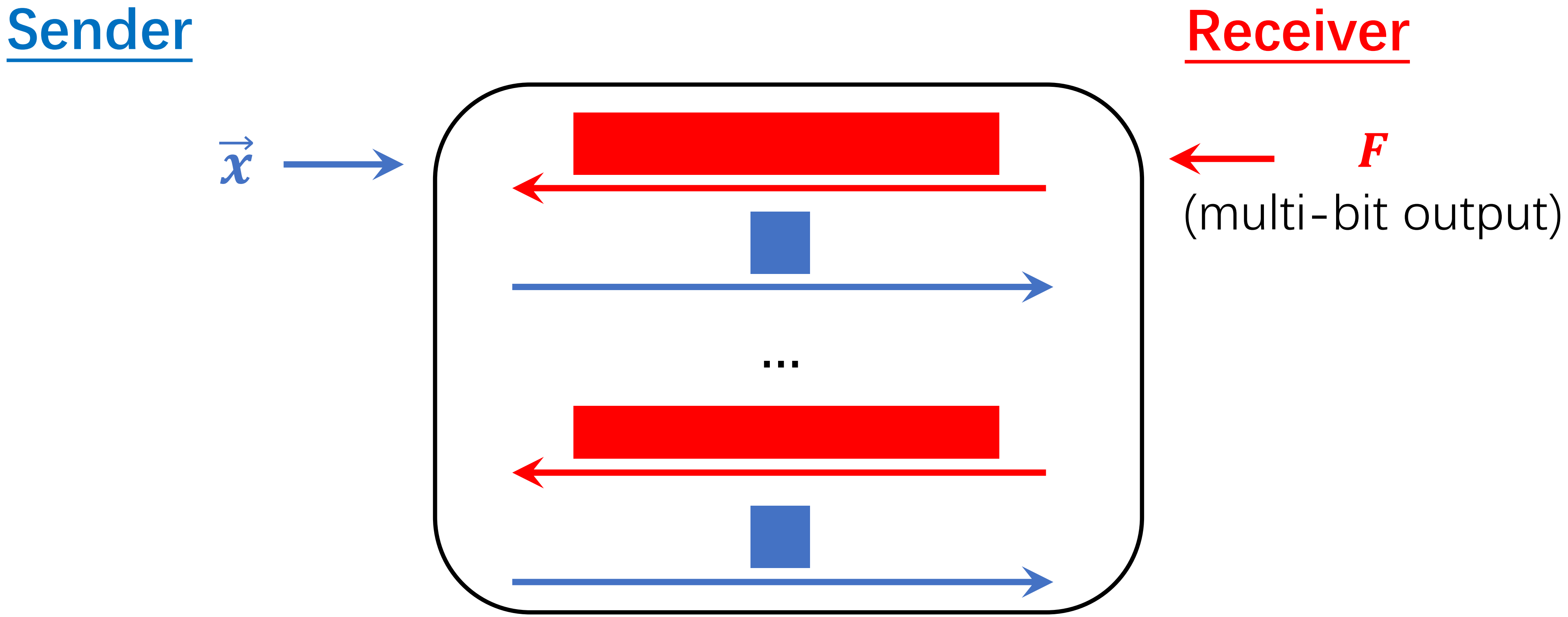
Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)



Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

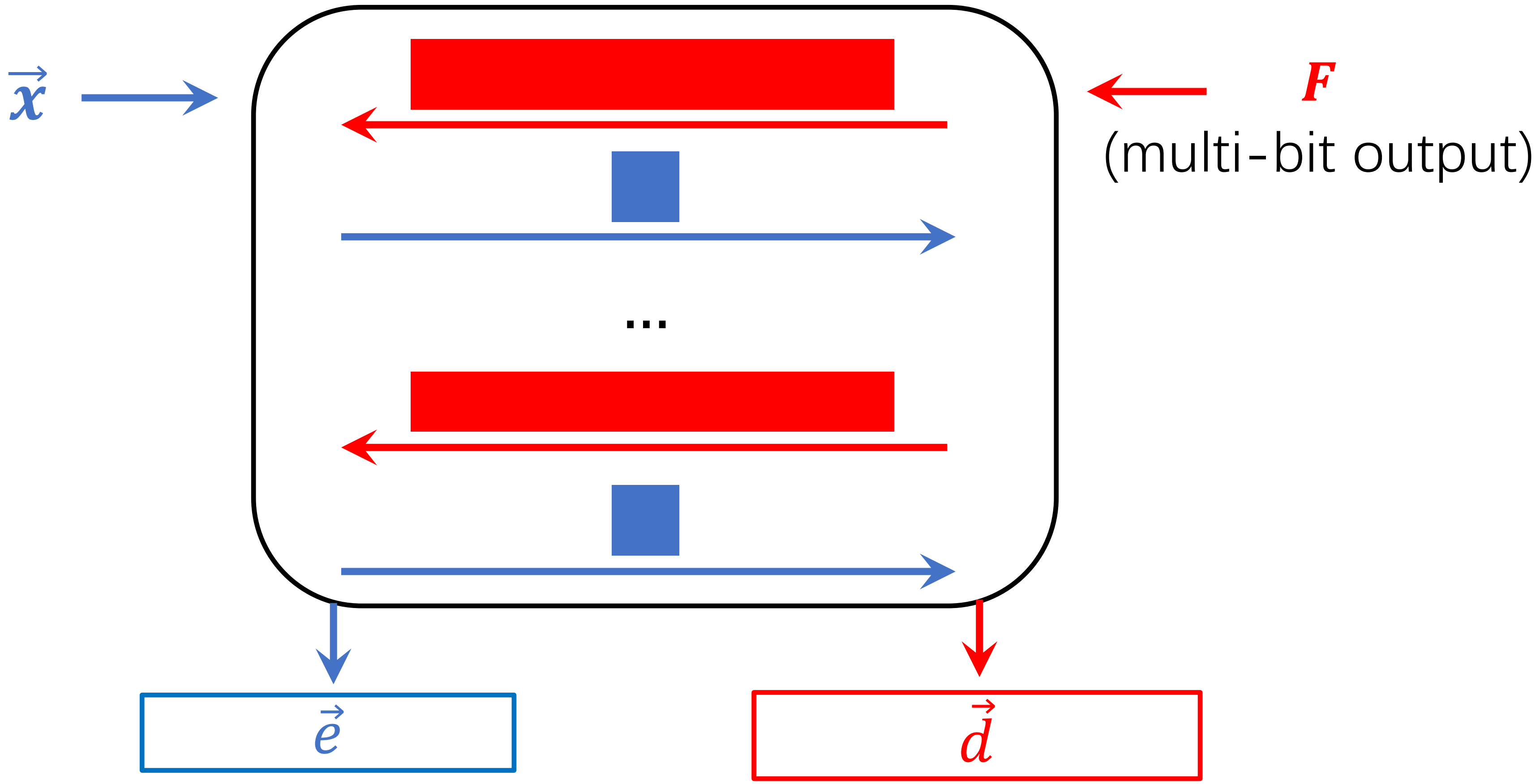


Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

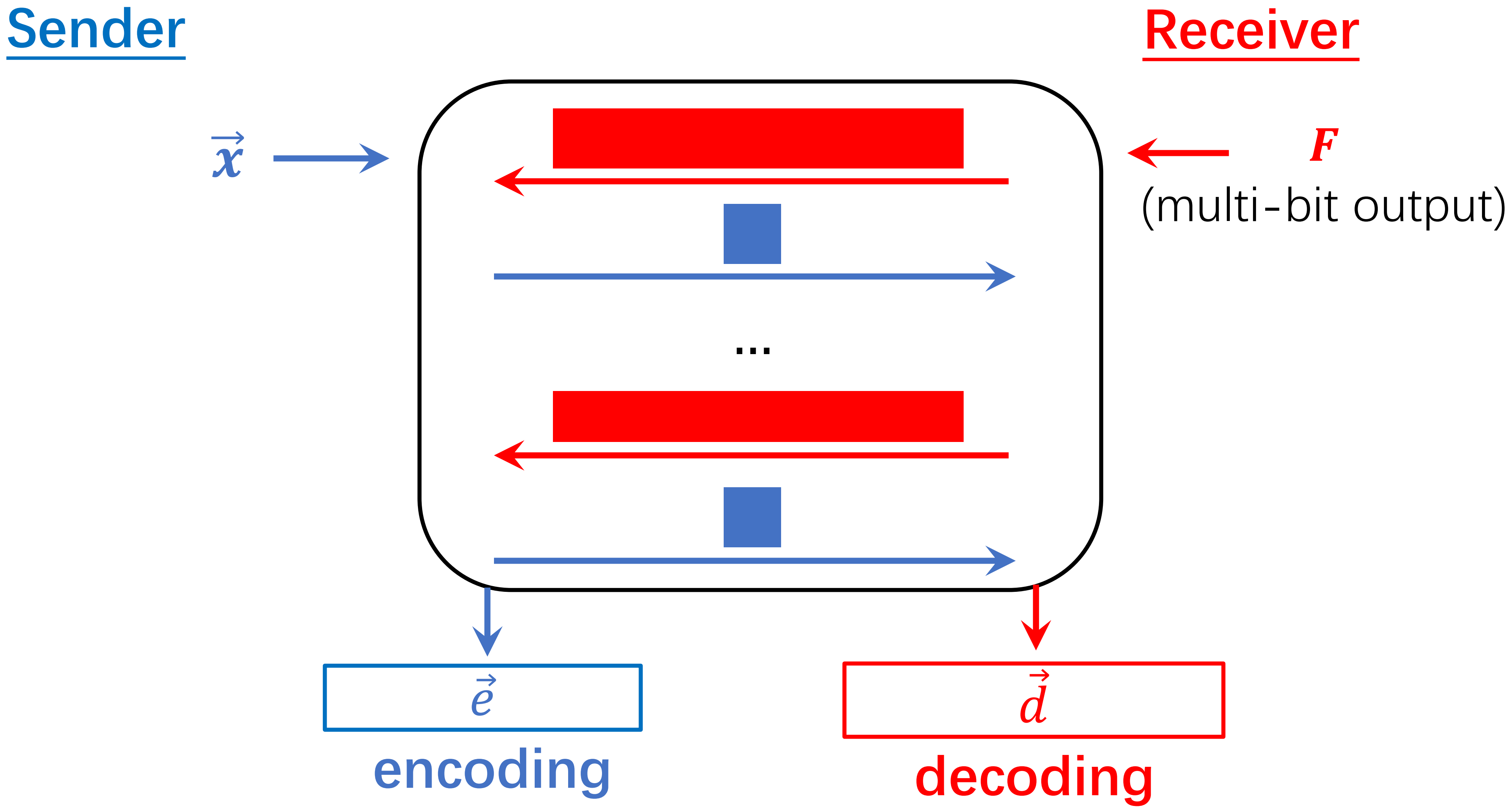


Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

Sender **Receiver**

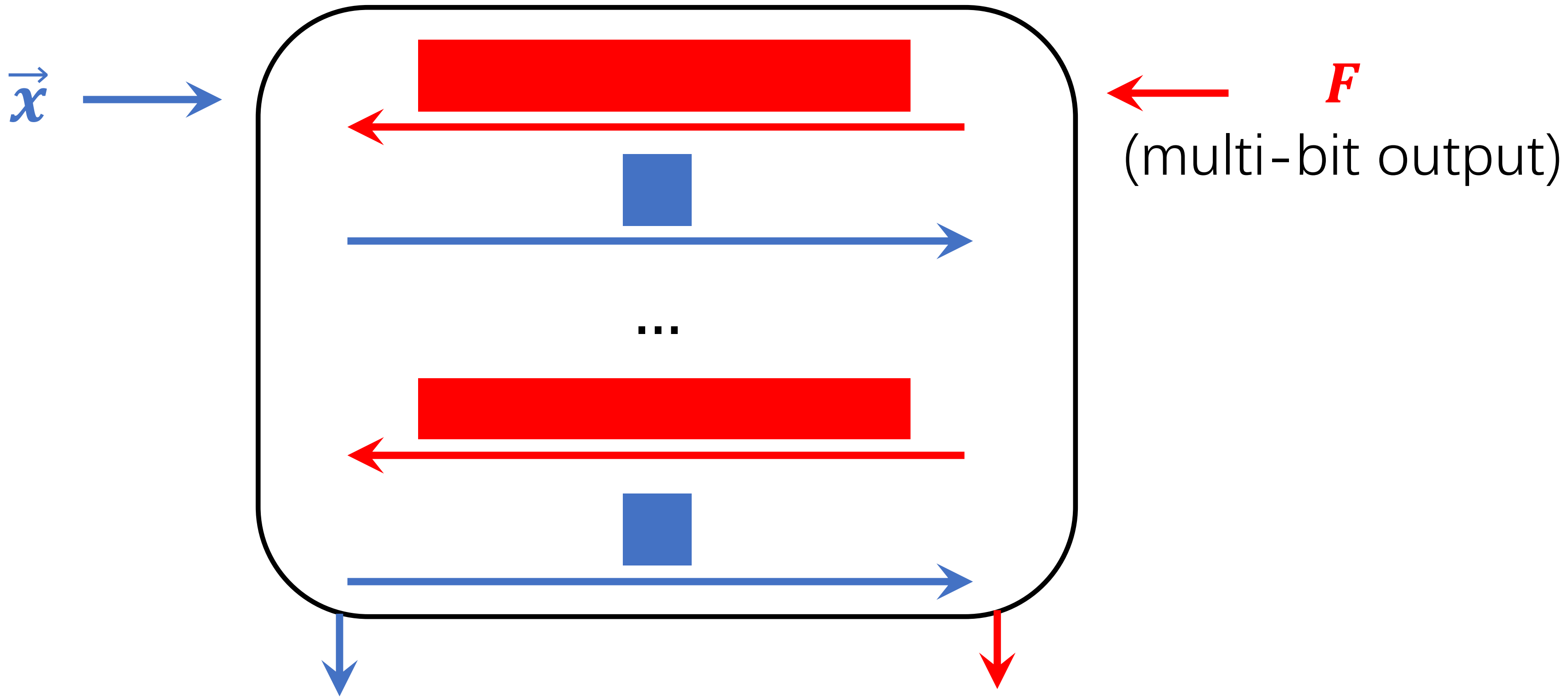


Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)



Main Tool: Interactive Trapdoor Hashing Protocol (ITDH)

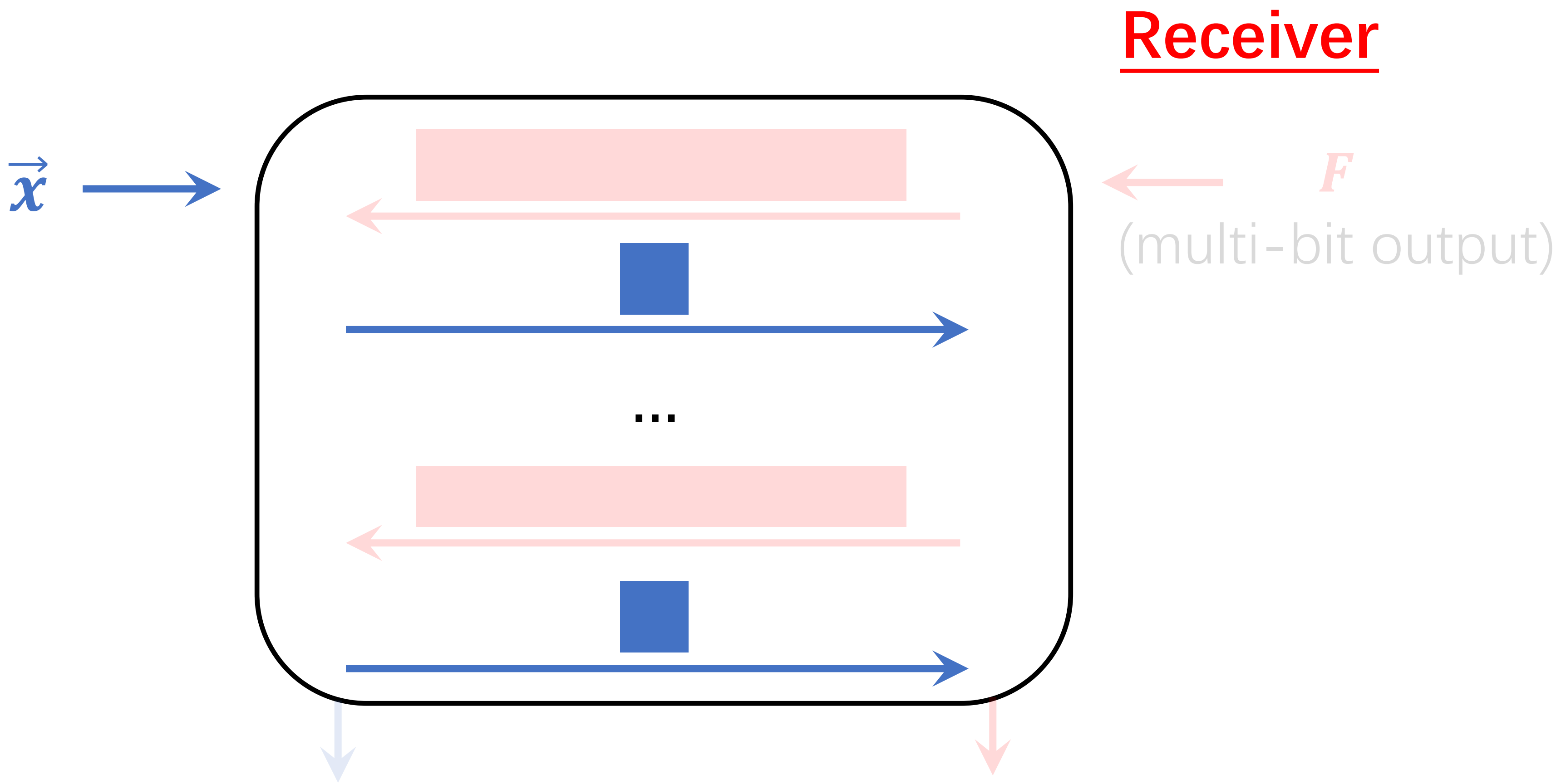
Sender | **Receiver**



- **Additive reconstruction:** $\boxed{\vec{e}} \oplus \boxed{\vec{d}} = F(\vec{x})$

encoding | decoding

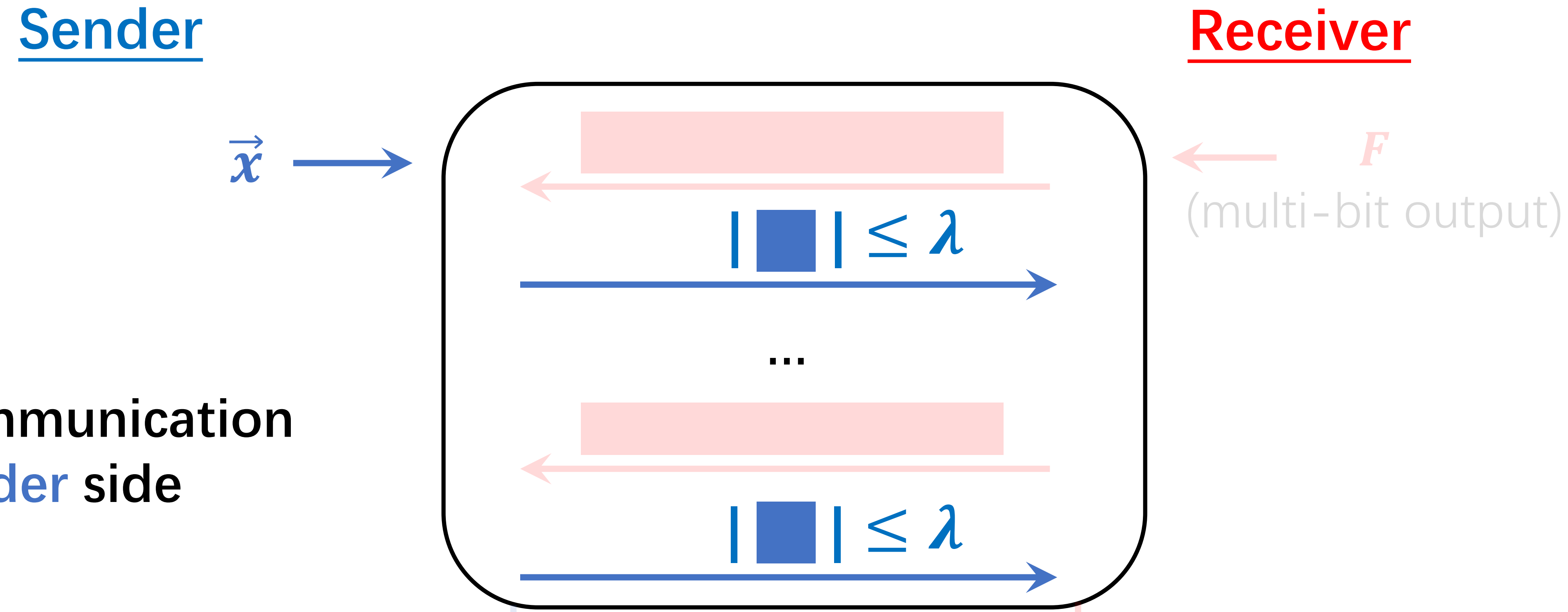
Sender's Side: Laconic Communication



- Additive reconstruction: $\boxed{\vec{e}} \oplus \boxed{\vec{d}} = F(\vec{x})$

encoding decoding

Sender's Side: Laconic Communication

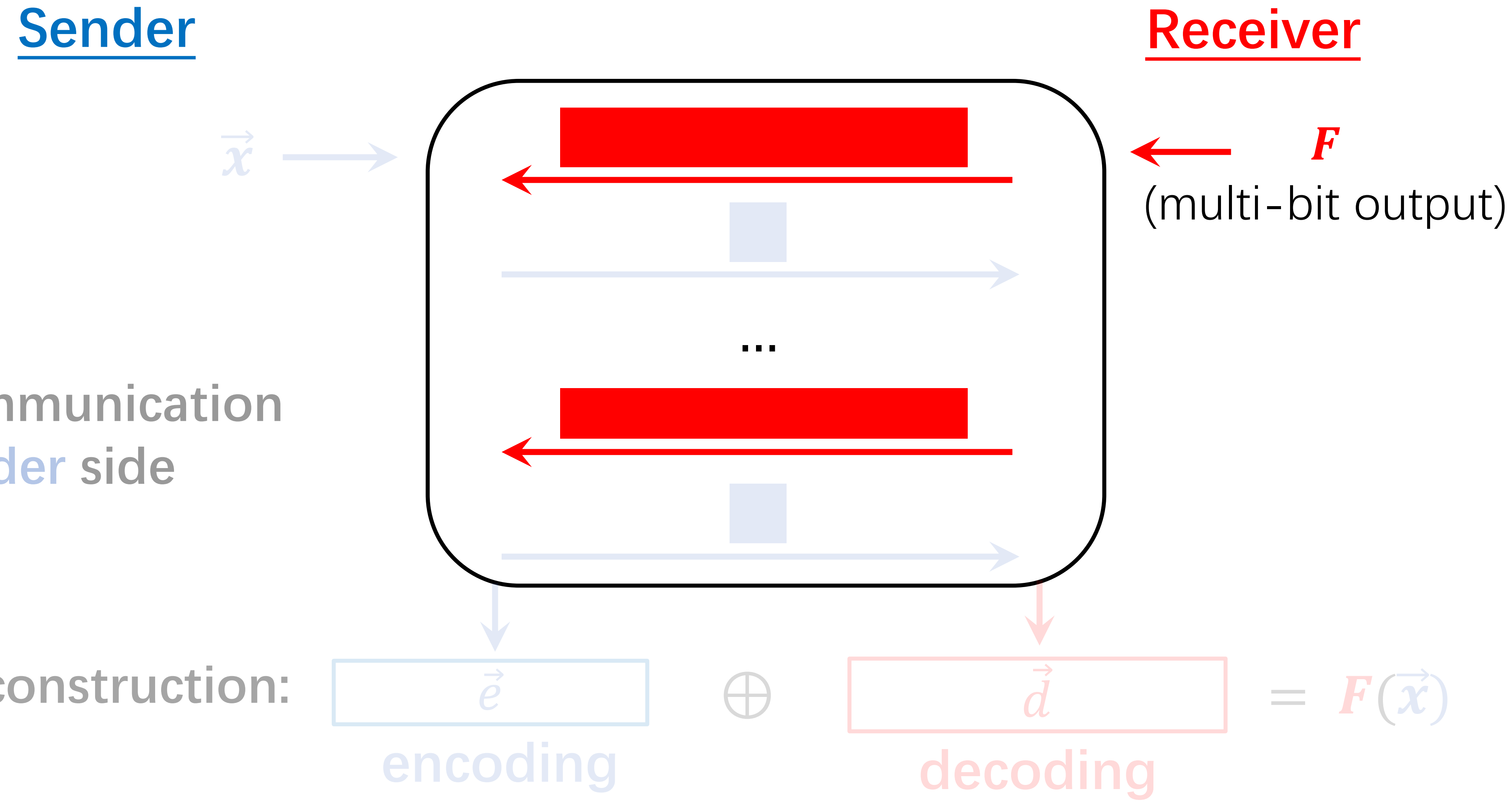


- **Laconic communication on sender side**

- **Additive reconstruction:** $\vec{e} \oplus \vec{d} = F(\vec{x})$

encoding decoding

Receiver's Side: Function Hiding

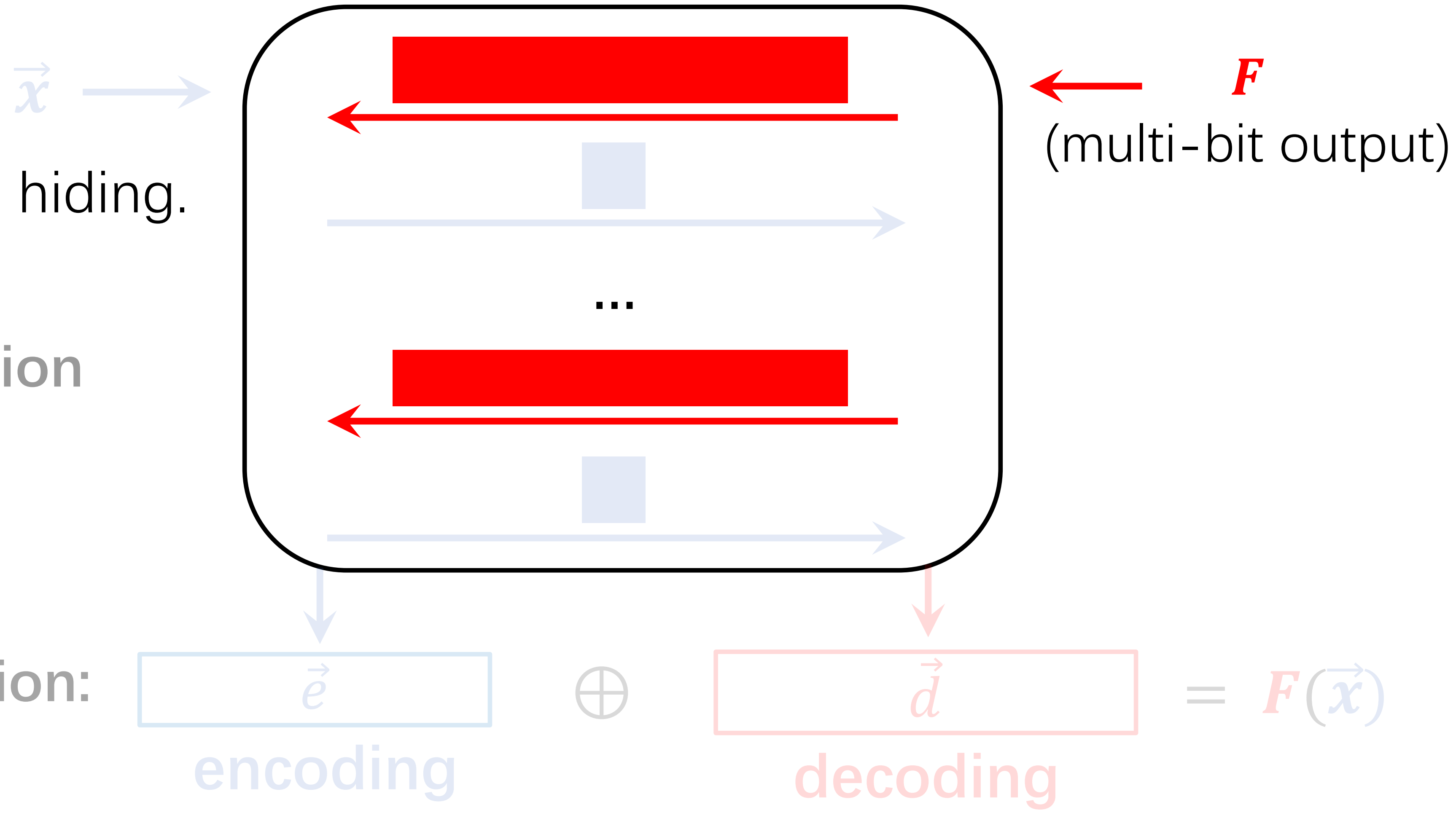


- Laconic communication on sender side
- Additive reconstruction:

Receiver’s Side: Function Hiding

Sender | **Receiver**

- **Function Hiding:** F is hiding.
- **Laconic communication on sender side**
- **Additive reconstruction:**



Interactive TDH vs Trapdoor Hash Function [DGIMMO19]

Sender

Receiver

Interactive TDH vs Trapdoor Hash Function [DGIMMO19]

Sender

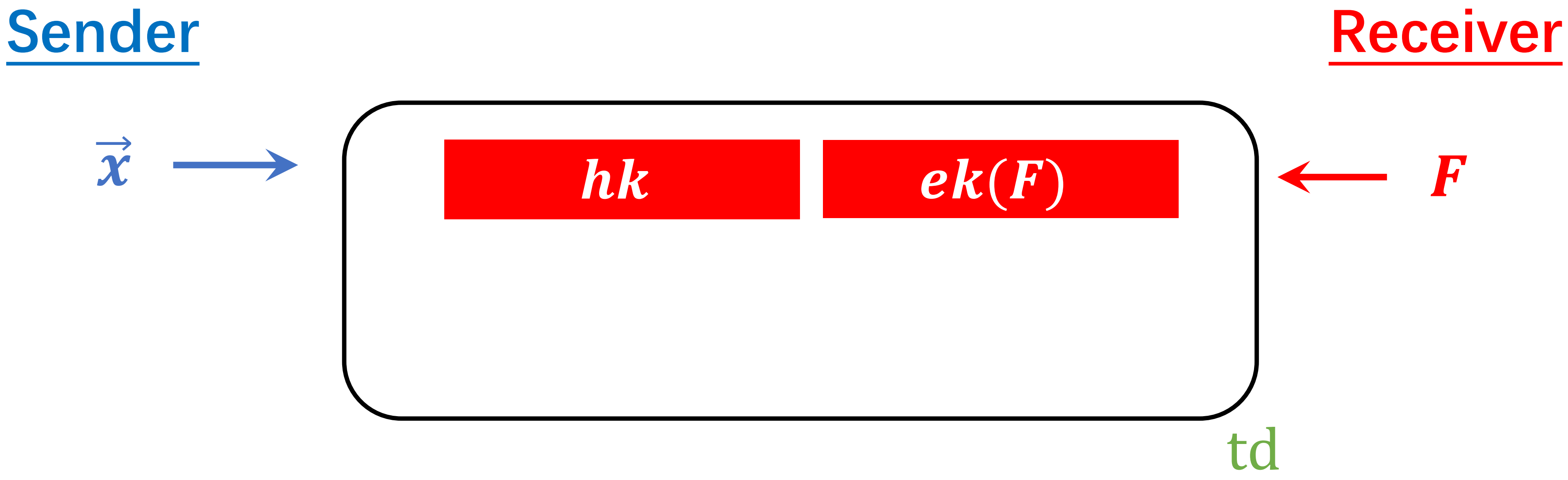
$\vec{x}$

Receiver

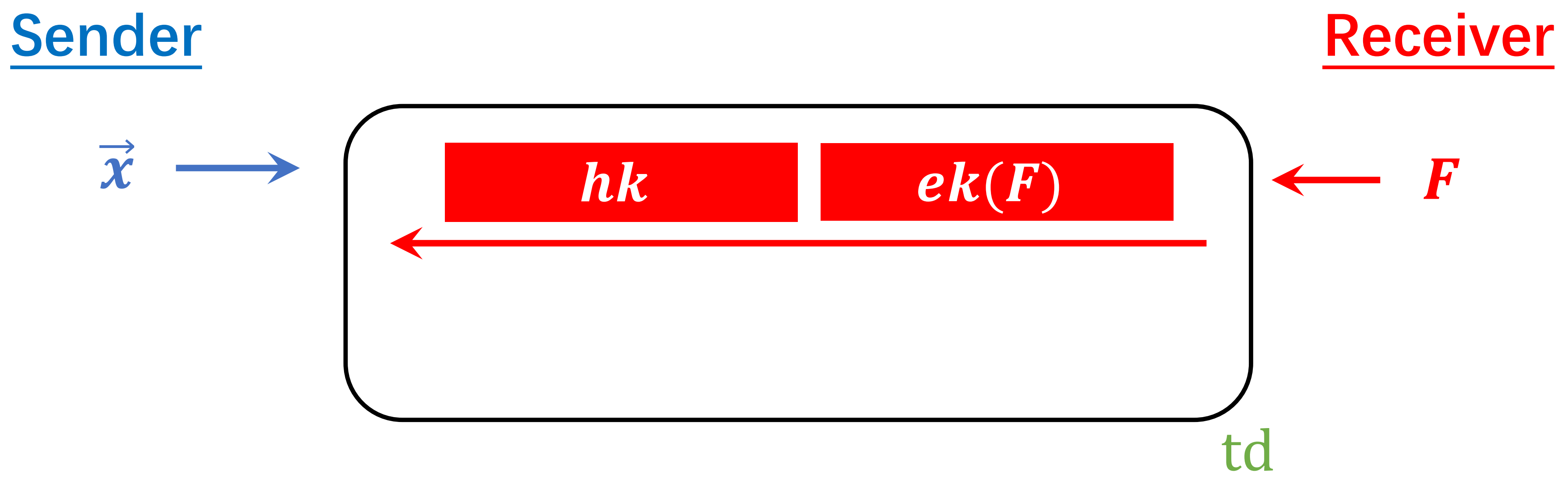
F

Interactive TDH vs Trapdoor Hash Function [DGIMMO19]

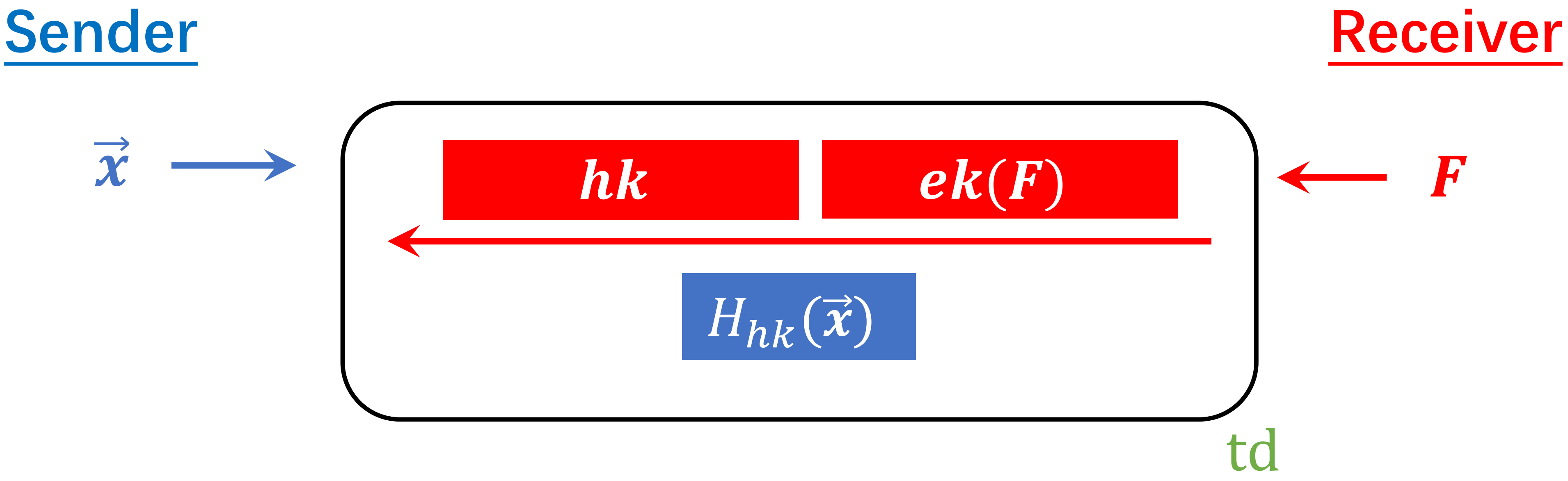
Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



Interactive TDH vs Trapdoor Hash Function [DGIMMO19]

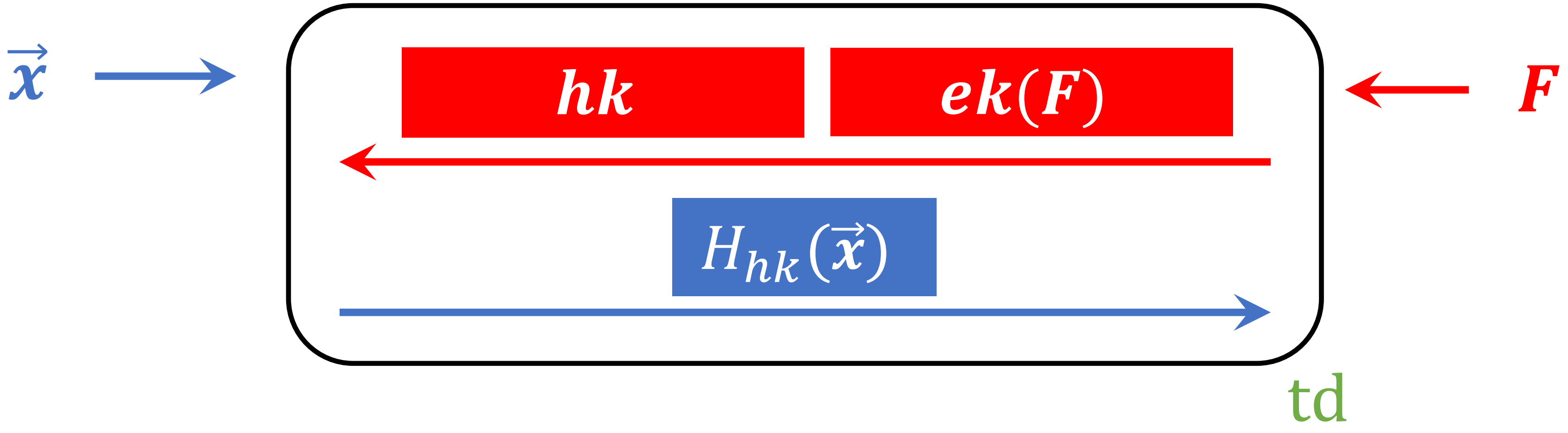


Interactive TDH vs Trapdoor Hash Function [DGIMMO19]

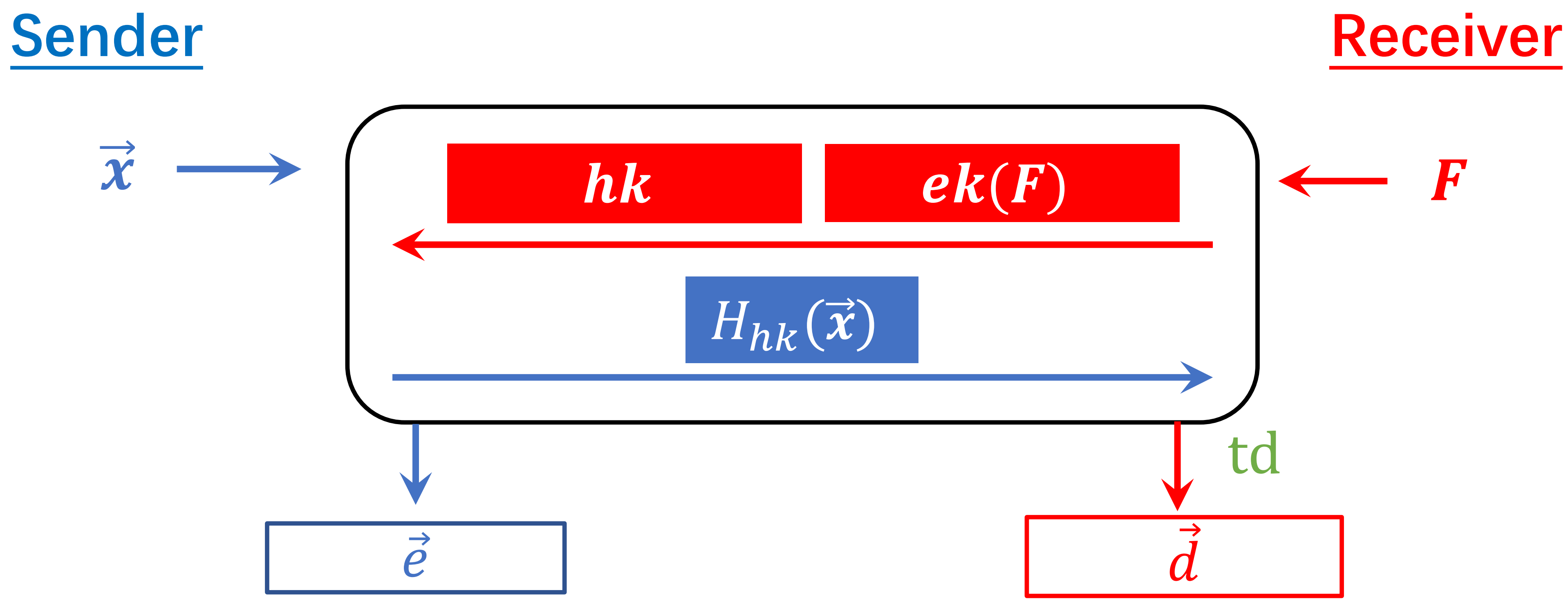


Interactive TDH vs Trapdoor Hash Function [DGIMMO19]

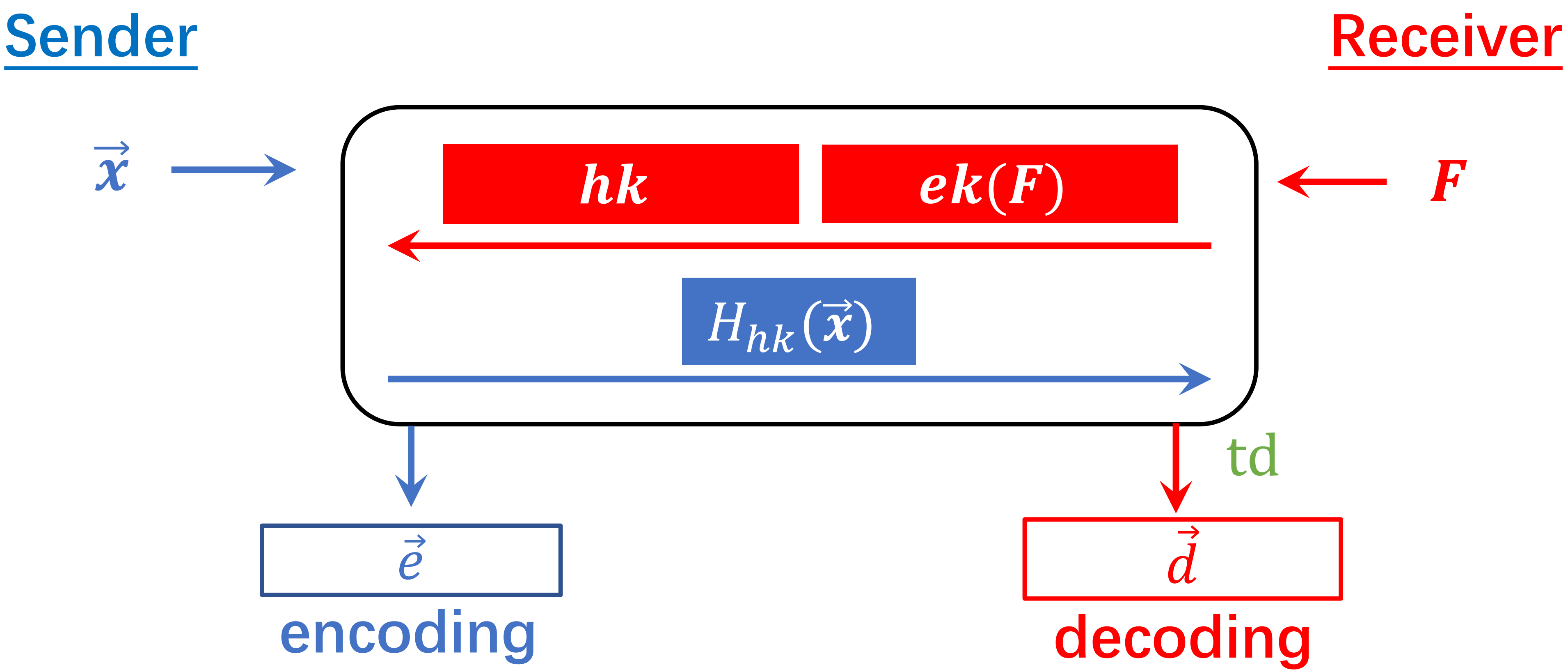
Sender **Receiver**



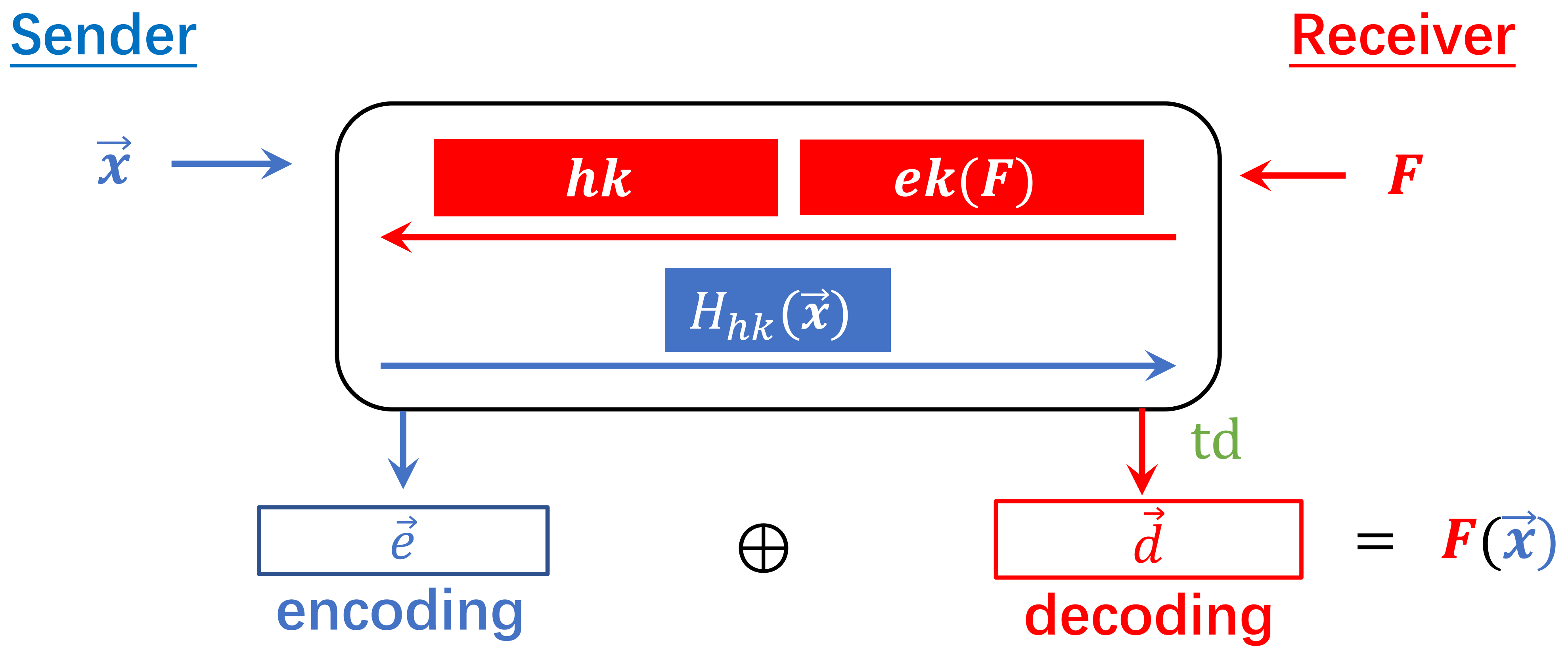
Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



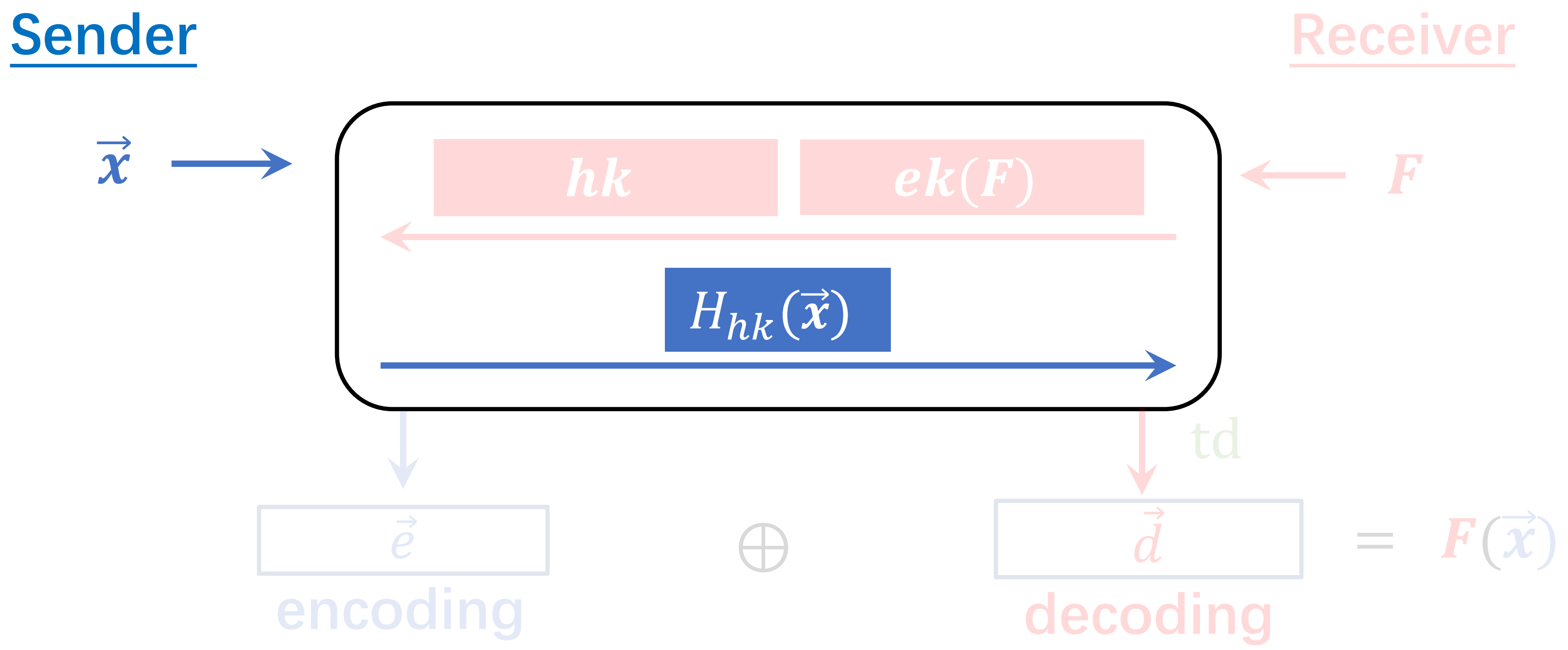
Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



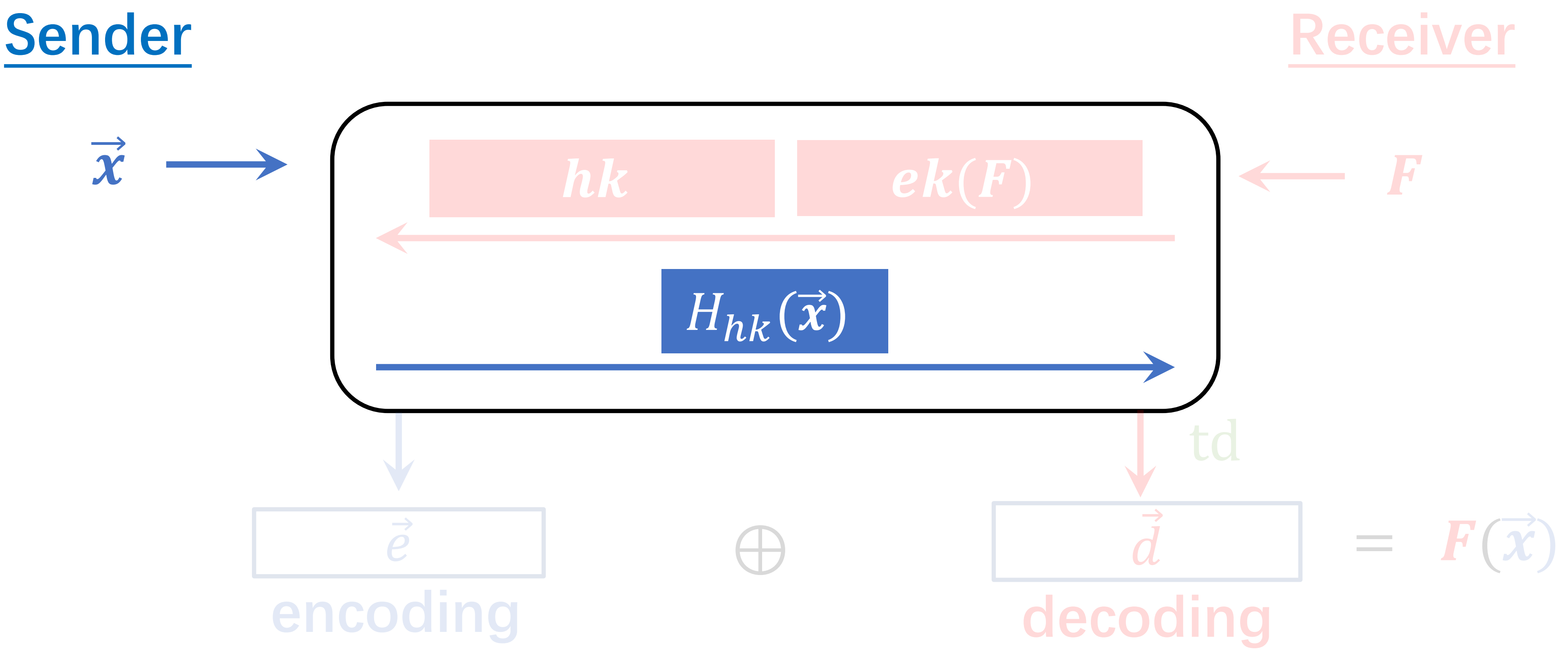
Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



Interactive TDH vs Trapdoor Hash Function [DGIMMO19]

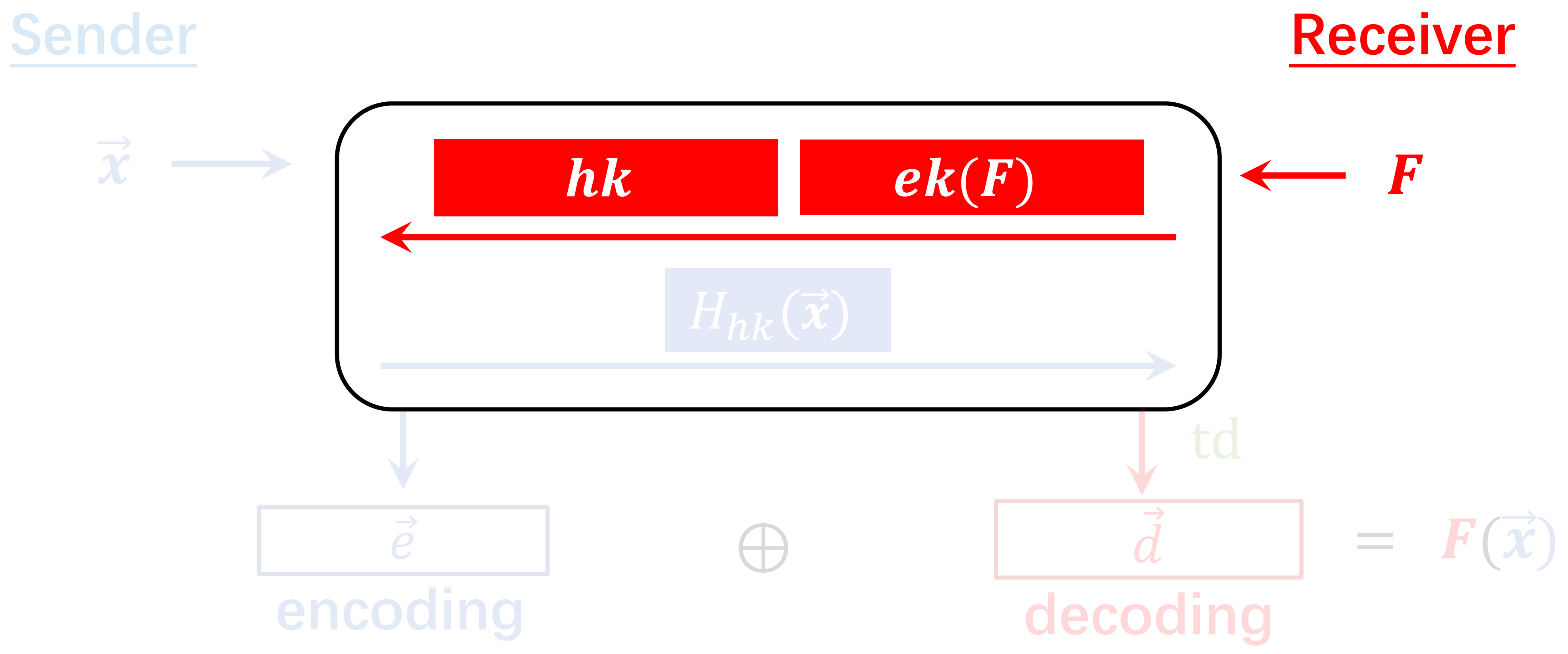


Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



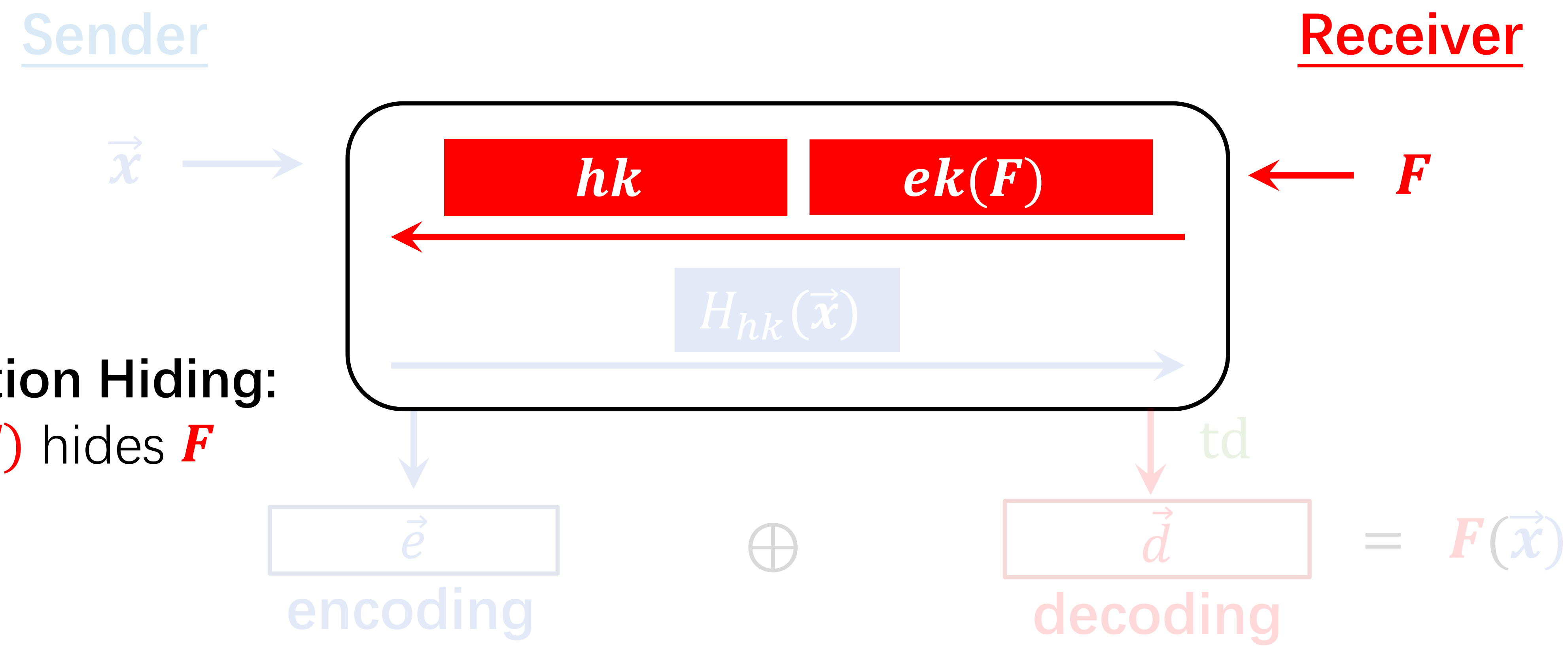
- **Laconic Communication on the Sender's Side: $H_{hk}(\vec{x})$ is small**

Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



- Laconic Communication on the Sender's Side: $H_{hk}(\vec{x})$ is small

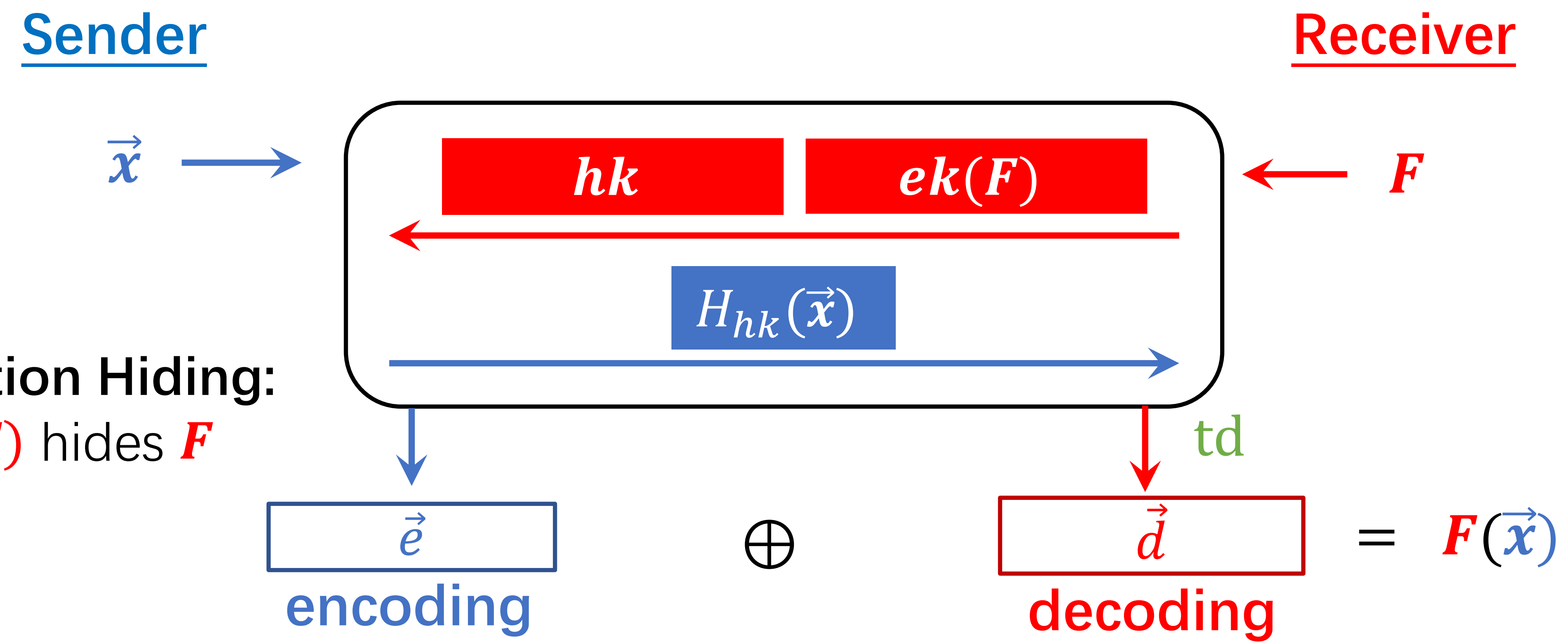
Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



- **Function Hiding:**
  $ek(F)$ hides F

- **Laconic Communication on the Sender's Side:** $H_{hk}(\vec{x})$ is small

Interactive TDH vs Trapdoor Hash Function [DGIMMO19]



- **Function Hiding:**
  $ek(F)$ hides F

- **Laconic Communication on the Sender's Side:** $H_{hk}(\vec{x})$ **is small**

Trapdoor Hash Functions

Previous Works:

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

Applications:

- Secure computation, rate-1 oblivious transfer, private information retrieval etc. [DGIMMO19]
- Correlation intractable hash and NIZKs [BKM20]

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

Applications:

- Secure computation, rate-1 oblivious transfer, private information retrieval etc. [DGIMMO19]
- Correlation intractable hash and NIZKs [BKM20]

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

Applications:

- Secure computation, rate-1 oblivious transfer, private information retrieval etc. [DGIMMO19]
- Correlation intractable hash and NIZKs [BKM20]

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

Applications:

- Secure computation, rate-1 oblivious transfer, private information retrieval etc. [DGIMMO19]
- Correlation intractable hash and NIZKs [BKM20]

Trapdoor Hash Functions

Previous Works:

- [DGIMMO19] TDH for index predicate & linear functions from DDH/LWE/QR/DCR
- [BKM20] TDH for constant-degree polynomials from DDH/LWE/QR/DCR

By leveraging the power of interaction, can we handle a larger class of circuits?

Applications:

- Secure computation, rate-1 oblivious transfer, private information retrieval etc. [DGIMMO19]
- Correlation intractable hash and NIZKs [BKM20]

Intermediate Result (1):

$\boldsymbol{O}(\mathbf{1})$-round Interactive TDH
for $\mathbf{TC^0}$ from DDH.

Intermediate Result (1):

$\boldsymbol{O(1)}$-round Interactive TDH
for $\mathbf{TC^0}$ from DDH.

(TC^0: constant-depth threshold circuits.)

Intermediate Result (1):

$\boldsymbol{O(1)}$-round Interactive TDH for $\mathbf{TC^0}$ from DDH.

(TC^0: constant-depth threshold circuits.)

(Can be generalized to poly-round for $\mathrm{P}/poly$ circuits)

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- **A Family of Hash:** $\{H_k(\vec{x})\}_k$

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- **A Family of Hash:** $\{\boldsymbol{H}_{\boldsymbol{k}}(\vec{\boldsymbol{x}})\}_{\boldsymbol{k}}$
- **Key Generation:** $\boldsymbol{k} \leftarrow \mathbf{Gen}(\mathbf{1}^{\lambda})$

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- A Family of Hash: $\{H_k(\vec{x})\}_k$
- Key Generation: $k \leftarrow \mathbf{Gen}(1^\lambda)$

<u>**Correlation Intractable for a Circuit Class $\mathcal{F}$:**</u>

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- A Family of Hash: $\{H_k(\vec{x})\}_k$
- Key Generation: $k \leftarrow \mathbf{Gen}(1^\lambda)$

Correlation Intractable for a Circuit Class $\mathcal{F}$:

$\forall$ fixed $F \in \mathcal{F}$

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- **A Family of Hash:** $\{\boldsymbol{H_k(\vec{x})}\}_{\boldsymbol{k}}$
- **Key Generation:** $\boldsymbol{k \leftarrow \mathrm{Gen}(1^\lambda)}$

Correlation Intractable for a Circuit Class $\boldsymbol{\mathcal{F}}$:

$\forall$ fixed $\boldsymbol{F} \in \boldsymbol{\mathcal{F}}$

PPT. Adversary

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- A Family of Hash: $\{H_k(\vec{x})\}_k$
- Key Generation: $k \leftarrow \mathbf{Gen}(1^\lambda)$

<u>Correlation Intractable for a Circuit Class $\mathcal{F}$:</u>

$\forall$ fixed $F \in \mathcal{F}$

$k \leftarrow \mathbf{Gen}(1^\lambda)$

PPT. Adversary

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- A Family of Hash: $\{H_k(\vec{x})\}_k$
- Key Generation: $k \leftarrow \mathsf{Gen}(1^\lambda)$

<u>Correlation Intractable for a Circuit Class $\mathcal{F}$:</u>

$\forall$ fixed $F \in \mathcal{F}$

PPT. Adversary

$k \leftarrow \mathsf{Gen}(1^\lambda)$

$\vec{x}$

Correlation Intractable Hash (CIH)

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

- A Family of Hash: $\{H_k(\vec{x})\}_k$
- Key Generation: $k \leftarrow \mathbf{Gen}(1^\lambda)$

<u>Correlation Intractable for a Circuit Class $\mathcal{F}$:</u>

$\forall$ fixed $F \in \mathcal{F}$

PPT. Adversary

$k \leftarrow \mathbf{Gen}(1^\lambda)$

$\vec{x}$

$\Pr_{\cdot,k}[H_k(\vec{x}) = F(\vec{x})] \leq \mathrm{negl}$

Correlation Intractable Hash (CIH)

Previous Works:

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Applications:

- NIZKs [CCHLRRW19,PS19,BKM20]
- SNARGs [CCHLRRW19,JKKZ20]
- Verifiable Delay Functions [LV20],
- PPAD-hardness [CHKPRR19,LV20,JKKZ20].

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Applications:

- NIZKs [CCHLRRW19,PS19,BKM20]
- SNARGs [CCHLRRW19,JKKZ20]
- Verifiable Delay Functions [LV20],
- PPAD-hardness [CHKPRR19,LV20,JKKZ20].

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Applications:

- NIZKs [CCHLRRW19,PS19,BKM20]
- SNARGs [CCHLRRW19,JKKZ20]
- Verifiable Delay Functions [LV20],
- PPAD-hardness [CHKPRR19,LV20,JKKZ20].

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Applications:

- NIZKs [CCHLRRW19,PS19,BKM20]
- SNARGs [CCHLRRW19,JKKZ20]
- Verifiable Delay Functions [LV20],
- PPAD-hardness [CHKPRR19,LV20,JKKZ20].

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Applications:

- NIZKs [CCHLRRW19,PS19,BKM20]
- SNARGs [CCHLRRW19,JKKZ20]
- Verifiable Delay Functions [LV20],
- PPAD-hardness [CHKPRR19,LV20,JKKZ20].

Correlation Intractable Hash (CIH)

Previous Works:

- [PS19,CCHLRRW19] CIH from LWE for polynomial size circuits
- [BKM20] CIH from TDH for approximate constant-degree polynomials.

Can we build CIH for a larger class of circuits from assumptions other than LWE?

Applications:

- NIZKs [CCHLRRW19,PS19,BKM20]
- SNARGs [CCHLRRW19,JKKZ20]
- Verifiable Delay Functions [LV20],
- PPAD-hardness [CHKPRR19,LV20,JKKZ20].

Intermediate Result (2):

Correlation Intractable Hash for $\mathbf{TC}^0$
from sub-exponential DDH.

Intermediate Result (2):

Correlation Intractable Hash for $\mathbf{TC}^0$ from sub-exponential DDH.

Assuming DDH is hard for sub-exponential time adversary, we can also obtain CIH for $O(\log\log\lambda)$-depth threshold circuits.

Technical Detail

Technical Detail

- Recap of Fiat-Shamir
- Main Challenges
- ITDH for $\mathbf{TC}^{\mathbf{0}}$ → CIH for $\mathbf{TC}^{\mathbf{0}}$
- Construction of ITDH

Technical Detail

- Recap of Fiat-Shamir
- Main Challenges
- ITDH for $\mathbf{TC}^{\mathbf{0}}$ → CIH for $\mathbf{TC}^{\mathbf{0}}$
- Construction of ITDH

Technical Detail

- Recap of Fiat-Shamir
- Main Challenges
- ITDH for $\mathbf{TC^0}$ → CIH for $\mathbf{TC^0}$
- Construction of ITDH

Technical Detail

- Recap of Fiat-Shamir
- Main Challenges
- ITDH for $\mathbf{TC^0}$ → CIH for $\mathbf{TC^0}$
- Construction of ITDH

Technical Detail

- **Recap of Fiat-Shamir**
- Main Challenges
- ITDH for $\mathbf{TC}^0$ → CIH for $\mathbf{TC}^0$
- Construction of ITDH

Fiat-Shamir via CIH

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

Σ-protocol

$x \in L$

P

V

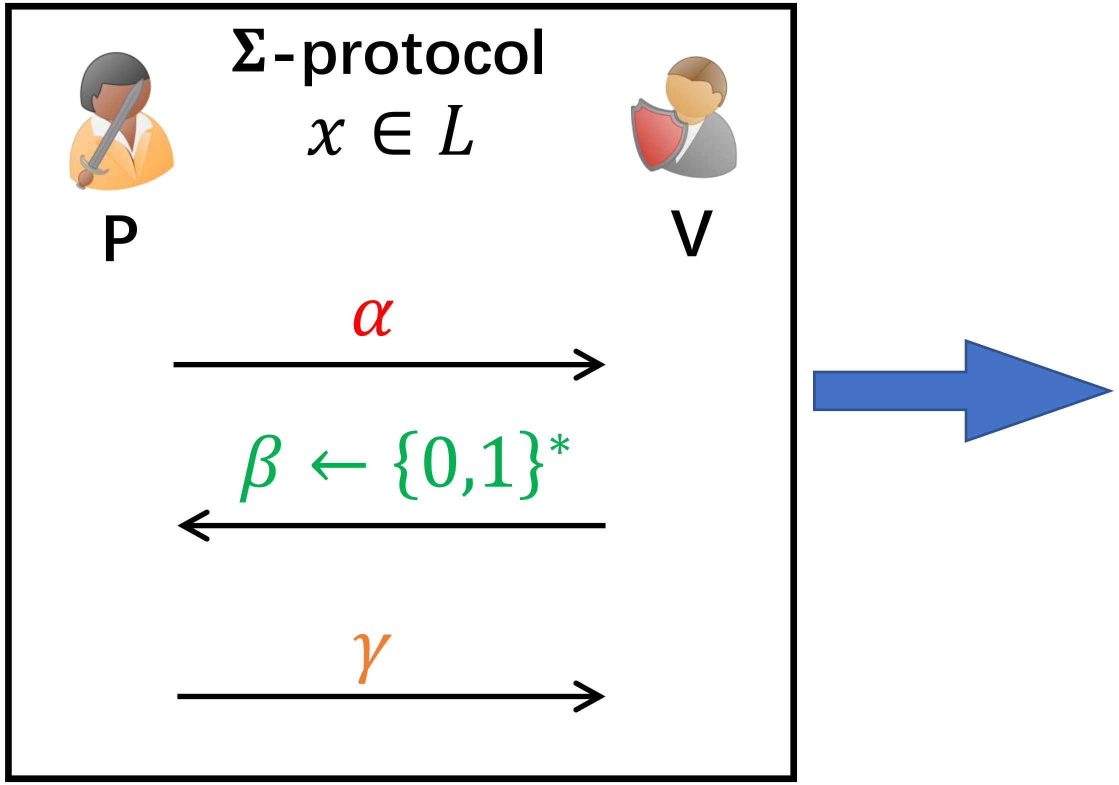
$$P \xrightarrow{\alpha} V$$

$$P \xleftarrow{\beta \leftarrow \{0,1\}^*} V$$

$$P \xrightarrow{\gamma} V$$

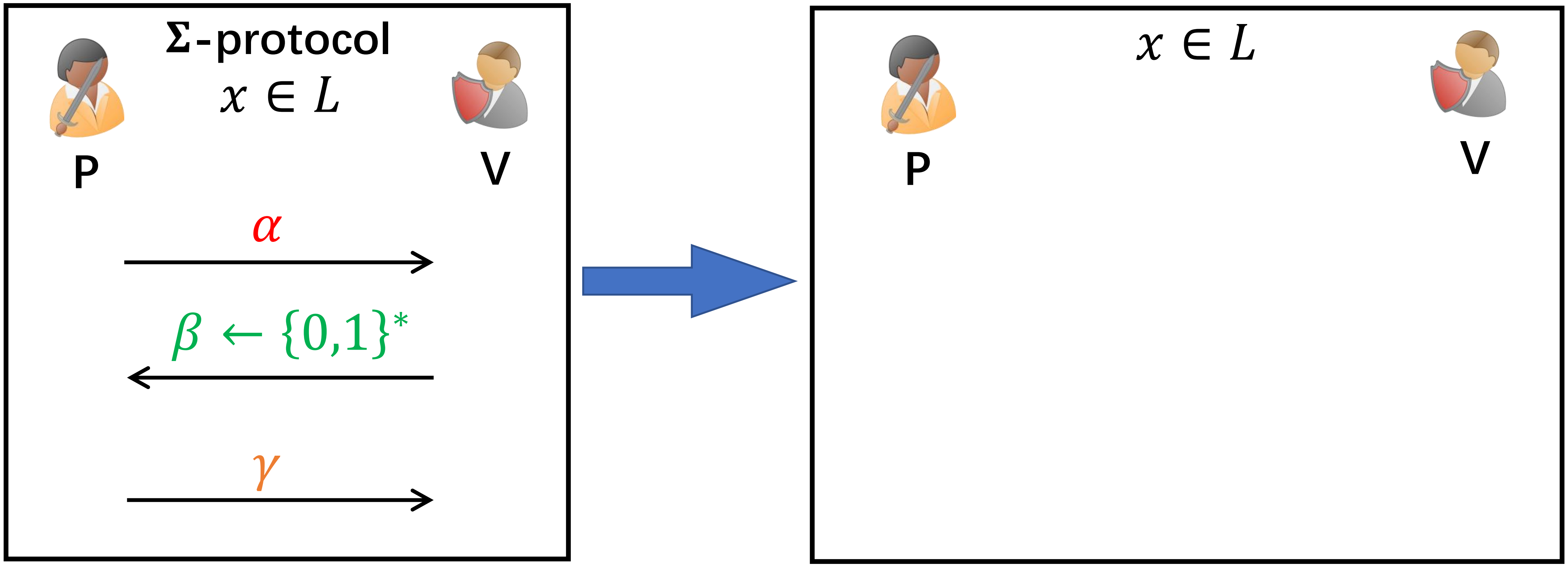
Fiat-Shamir via CIH

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



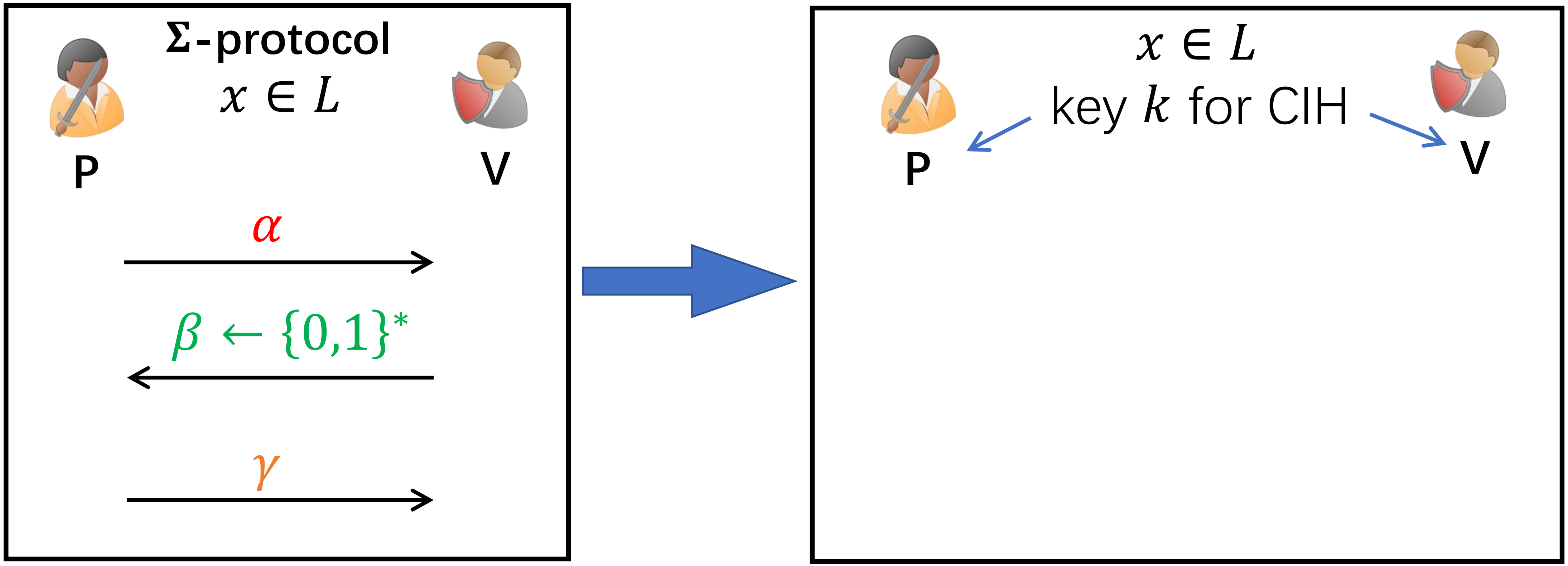
Fiat-Shamir via CIH

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



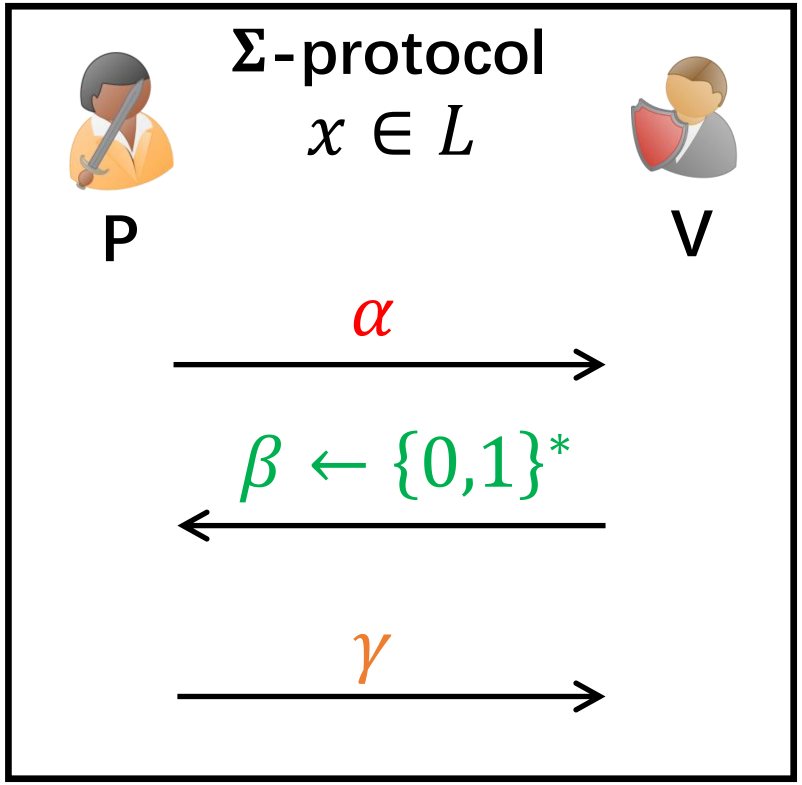
Fiat-Shamir via CIH

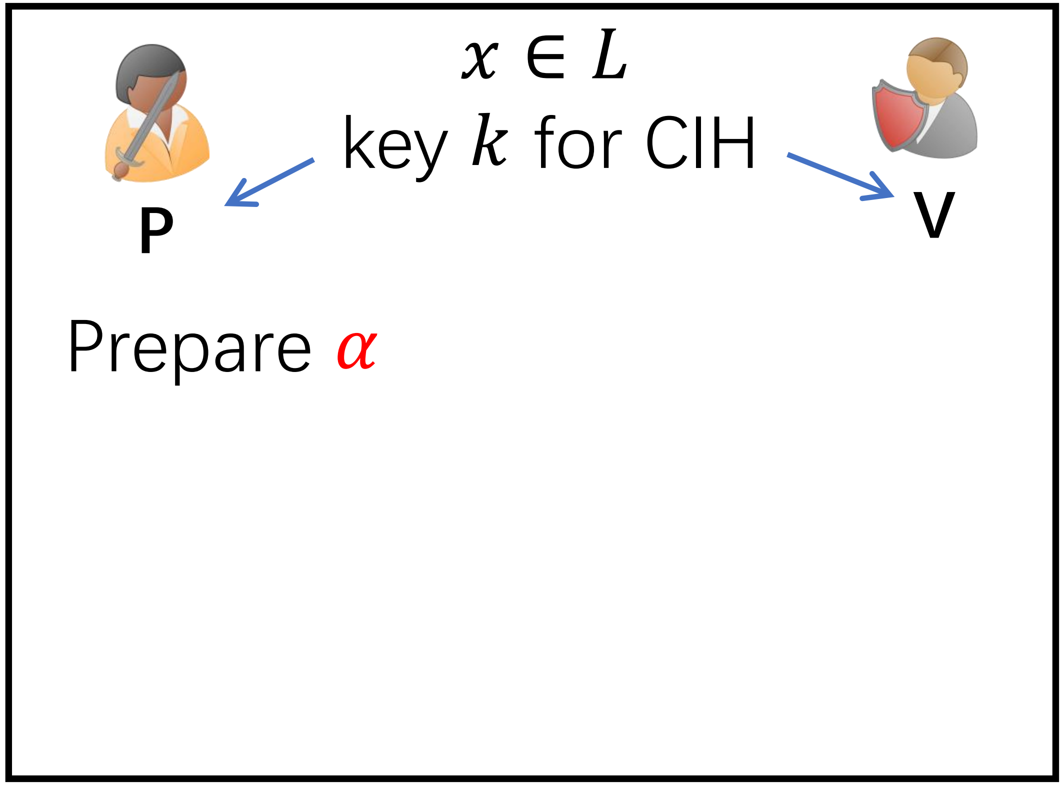
[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



Fiat-Shamir via CIH

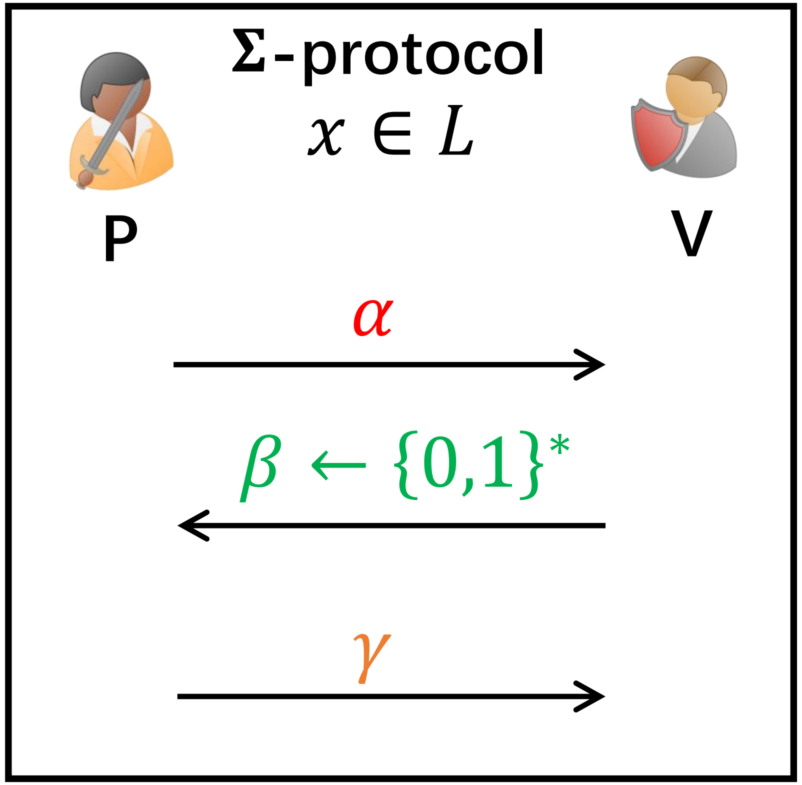
[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

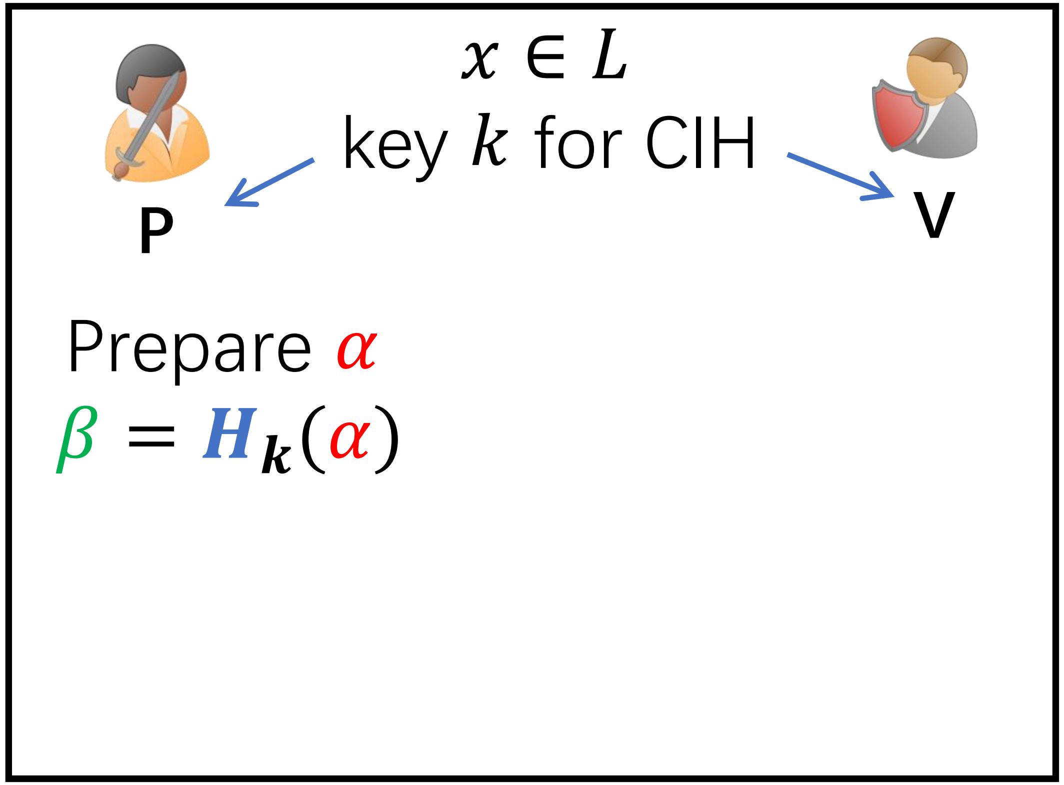




Fiat-Shamir via CIH

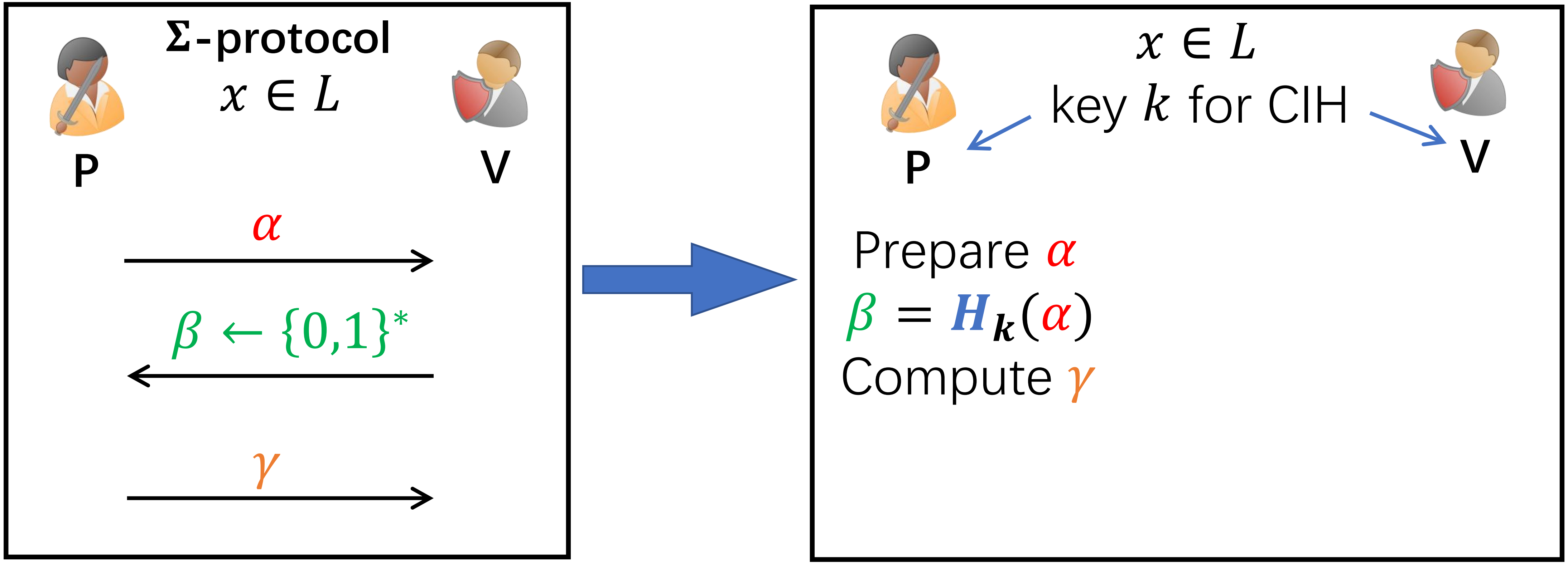
[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]





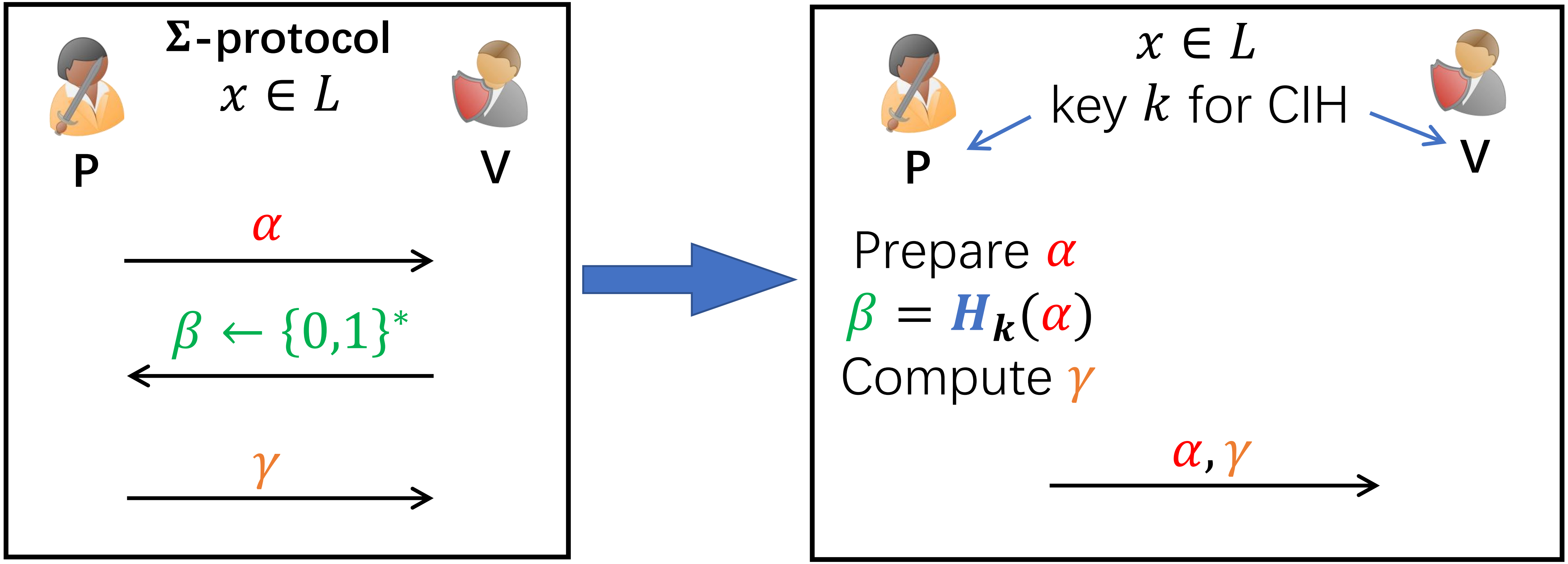
Fiat-Shamir via CIH

[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

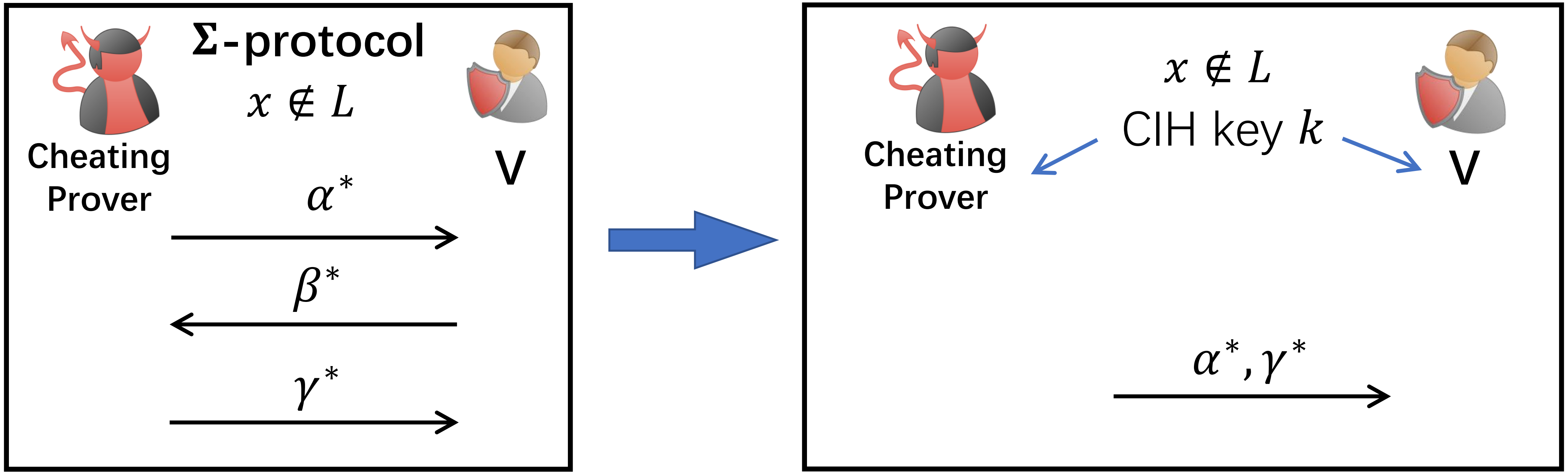


Fiat-Shamir via CIH

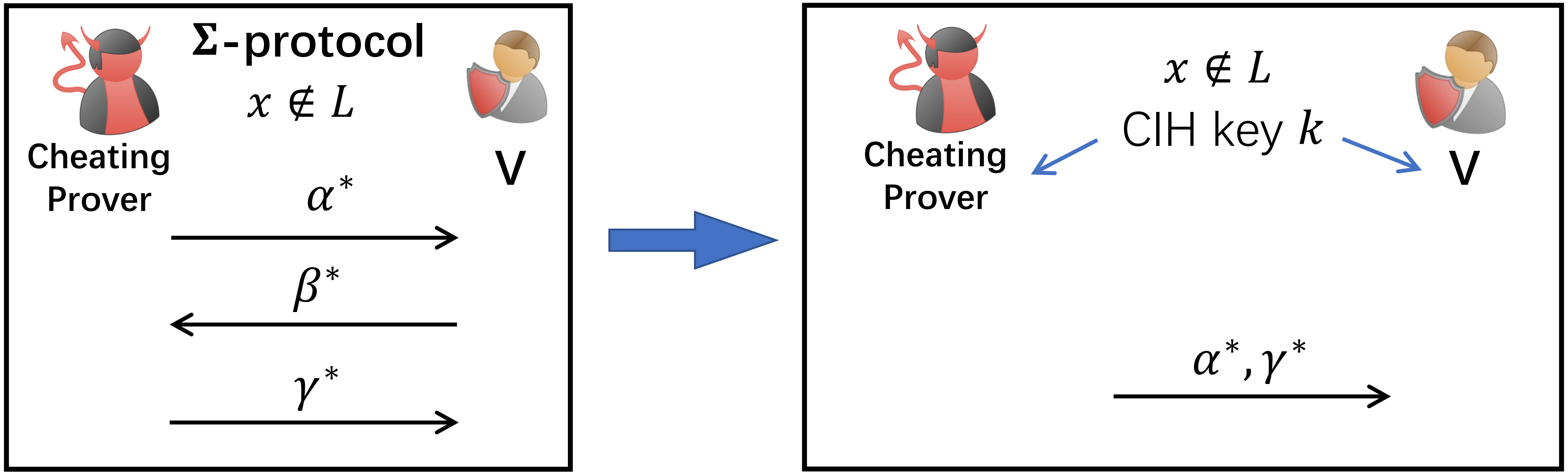
[CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

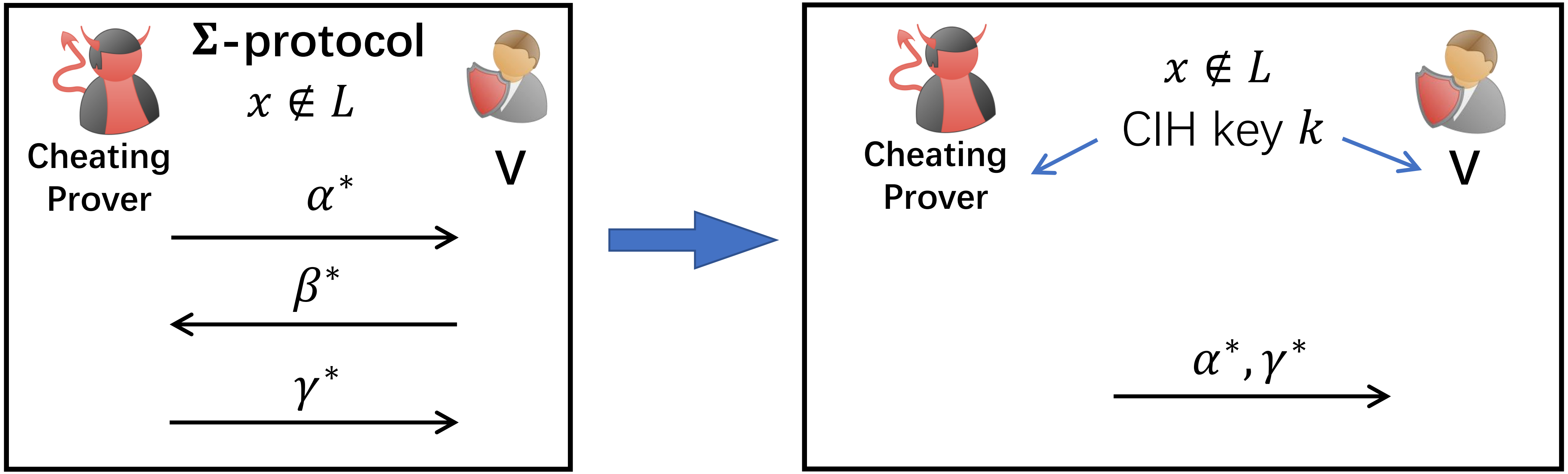


Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



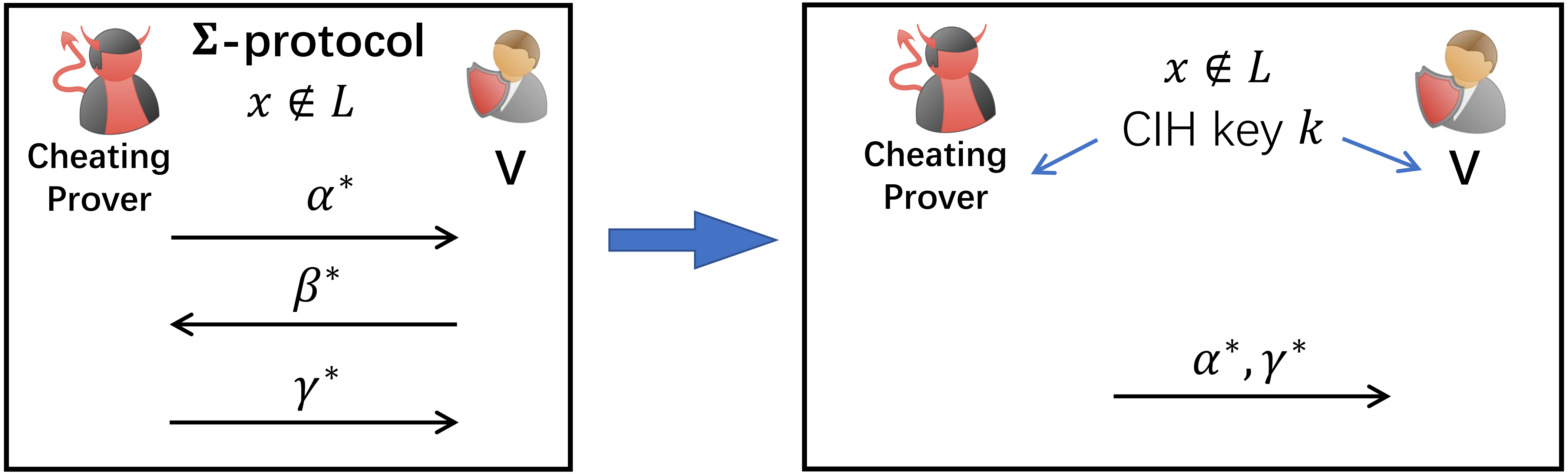
Special Soundness: A witness can be extracted from two accepting transcripts $(\alpha^*, \beta_0^*, \gamma_0^*)$, $(\alpha^*, \beta_1^*, \gamma_1^*)$, if $\beta_0^* \neq \beta_1^*$.

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



Special Soundness: A witness can be extracted from two accepting transcripts $(\alpha^*, \beta_0^*, \gamma_0^*)$, $(\alpha^*, \beta_1^*, \gamma_1^*)$, if $\beta_0^* \neq \beta_1^*$.

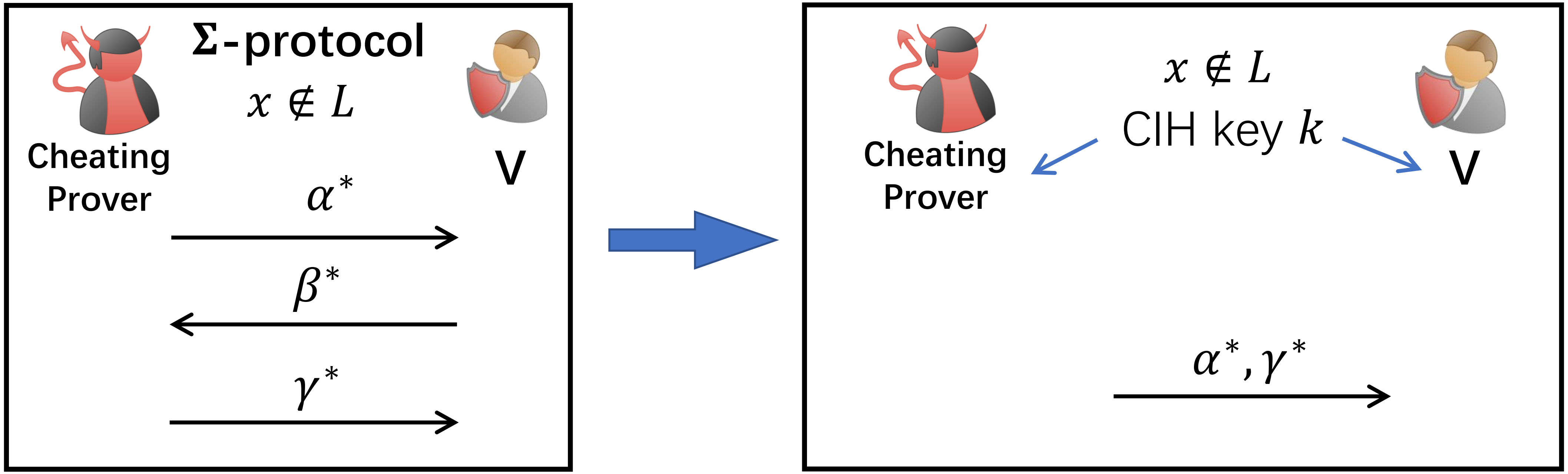
Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



Special Soundness: A witness can be extracted from two accepting transcripts $(\alpha^*, \beta_0^*, \gamma_0^*)$, $(\alpha^*, \beta_1^*, \gamma_1^*)$, if $\beta_0^* \neq \beta_1^*$.

If $x \notin L$, for any α^*, ∃ unique β^* such that $(\alpha^*, \beta^*, \cdot)$ can be accepted.

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

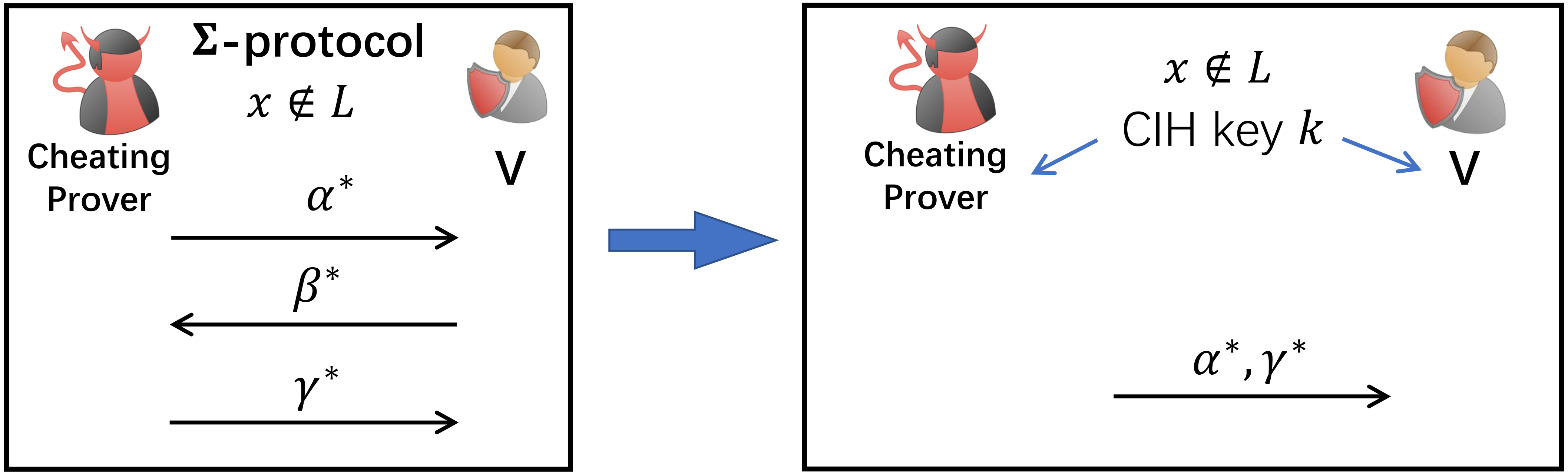


Special Soundness: A witness can be extracted from two accepting transcripts $(\alpha^*, \beta_0^*, \gamma_0^*)$, $(\alpha^*, \beta_1^*, \gamma_1^*)$, if $\beta_0^* \neq \beta_1^*$.

If $x \notin L$, for any α^*, $\exists$ unique β^* such that $(\alpha^*, \beta^*, \cdot)$ can be accepted.

$\alpha^* \longrightarrow$ the unique β^*

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

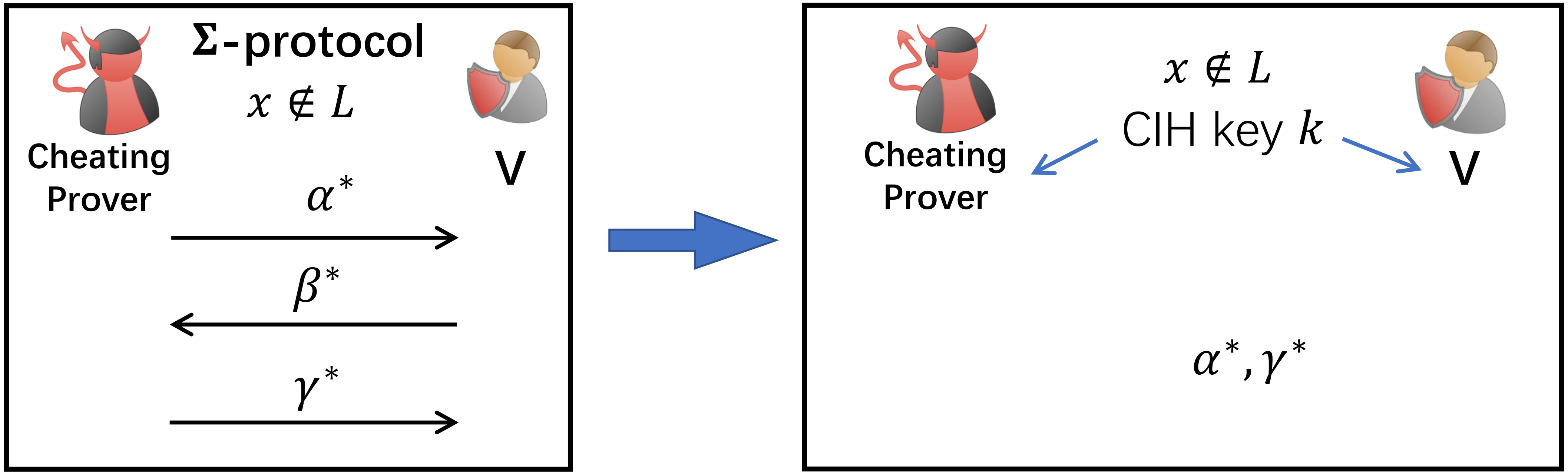


Special Soundness: A witness can be extracted from two accepting transcripts $(\alpha^*, \beta_0^*, \gamma_0^*)$, $(\alpha^*, \beta_1^*, \gamma_1^*)$, if $\beta_0^* \neq \beta_1^*$.

If $x \notin L$, for any α^*, ∃ unique β^* such that $(\alpha^*, \beta^*, \cdot)$ can be accepted.

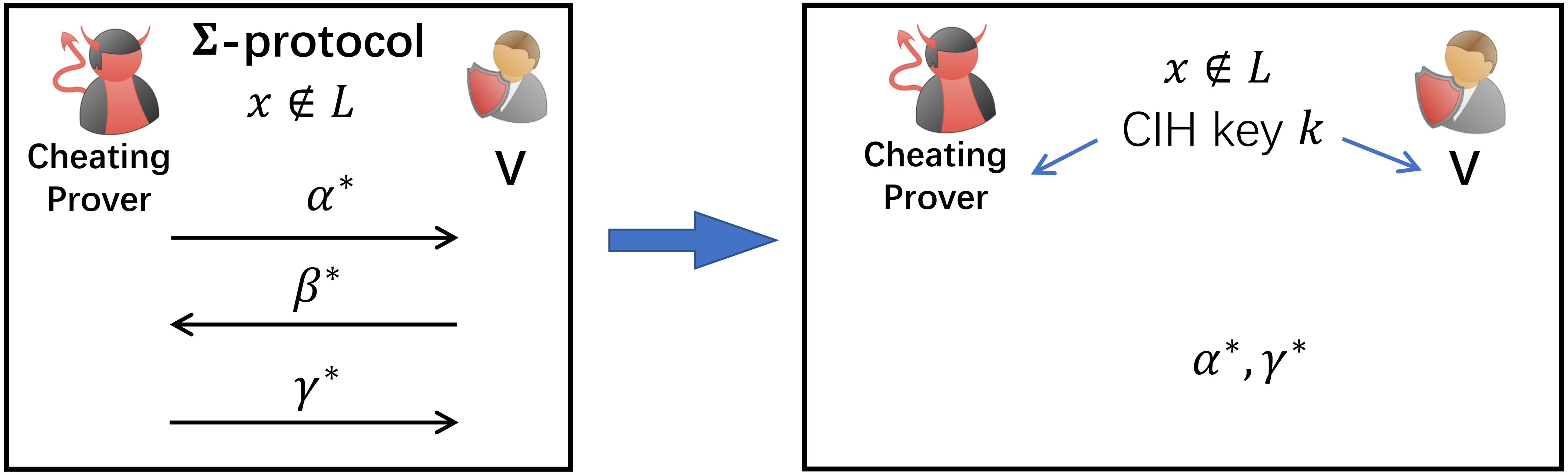
BAD: $\alpha^* \longrightarrow$ the unique β^*

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



BAD: α^* ⟶ the unique β^*

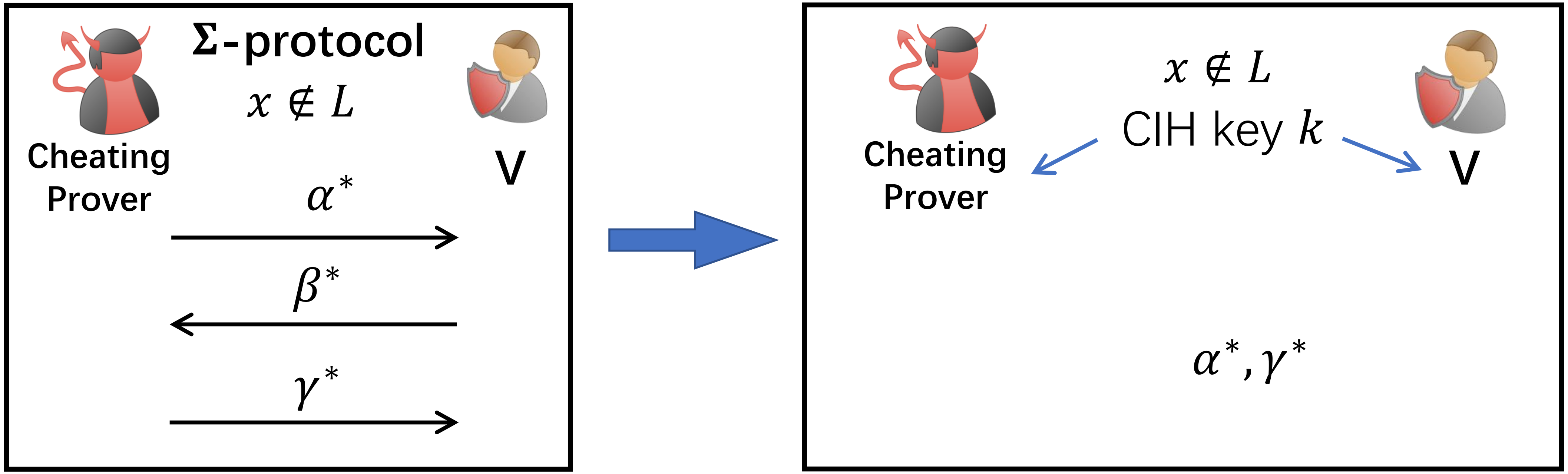
Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



BAD: α^* $\longrightarrow$ the unique β^*

Verifier accepts $\Rightarrow \beta^* = \mathrm{CIH}_k(\alpha^*) = \mathbf{BAD}(\alpha^*)$: **Contradiction to Correlation Intractability**

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

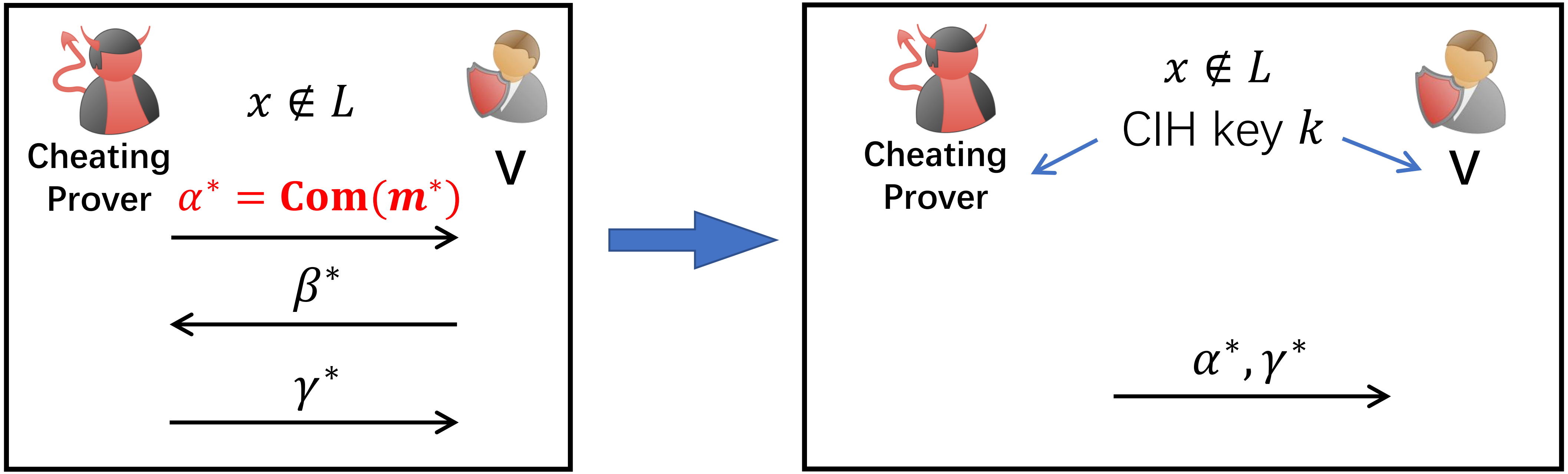


Known constructions of CIH can only handle efficiently computable BAD

$\mathbf{BAD}$: α^* → the unique β^*

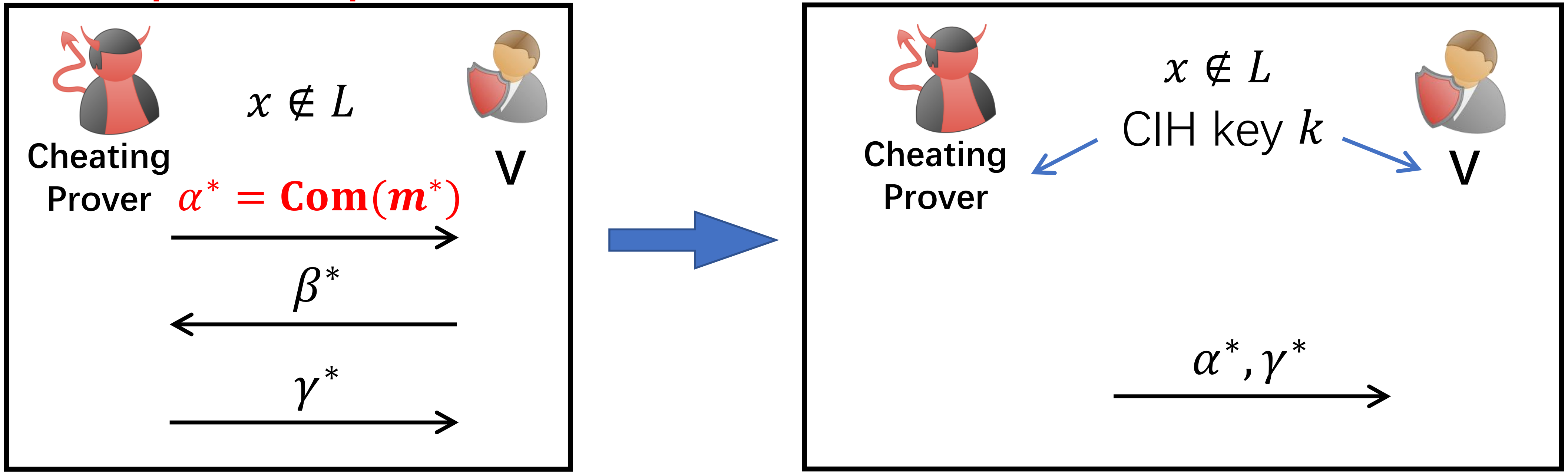
Verifier accepts $\Rightarrow \beta^* = \mathrm{CIH}_k(\alpha^*) = \mathbf{BAD}(\alpha^*)$: Contradiction to Correlation Intractability

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

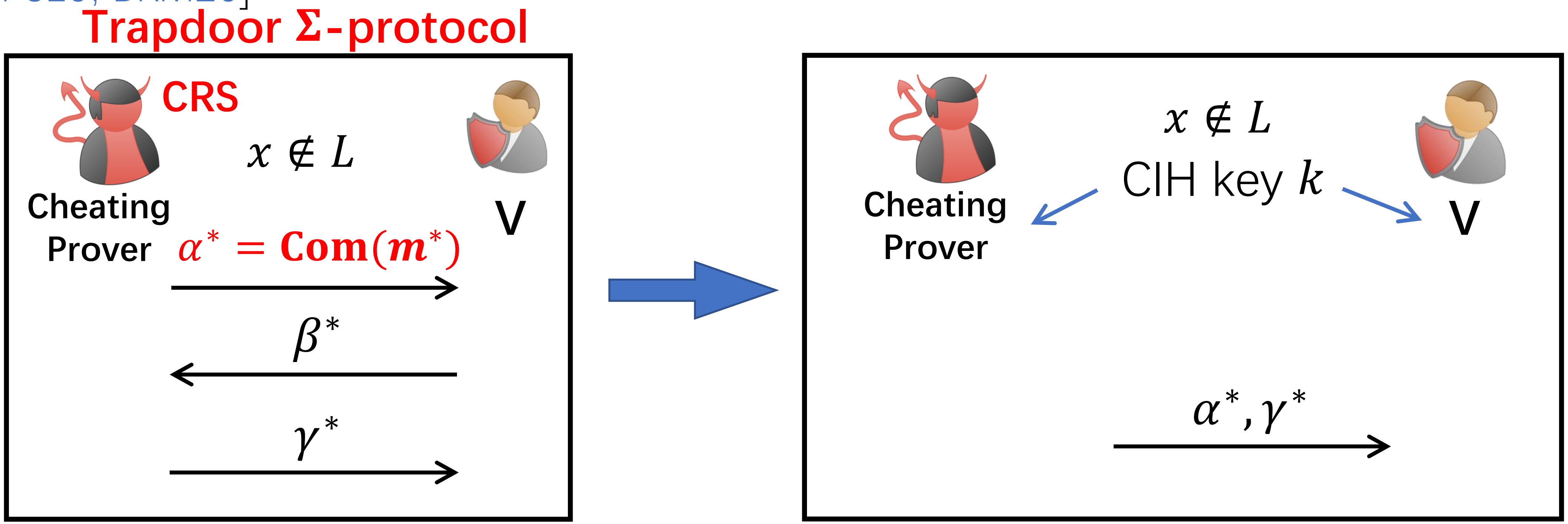


Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

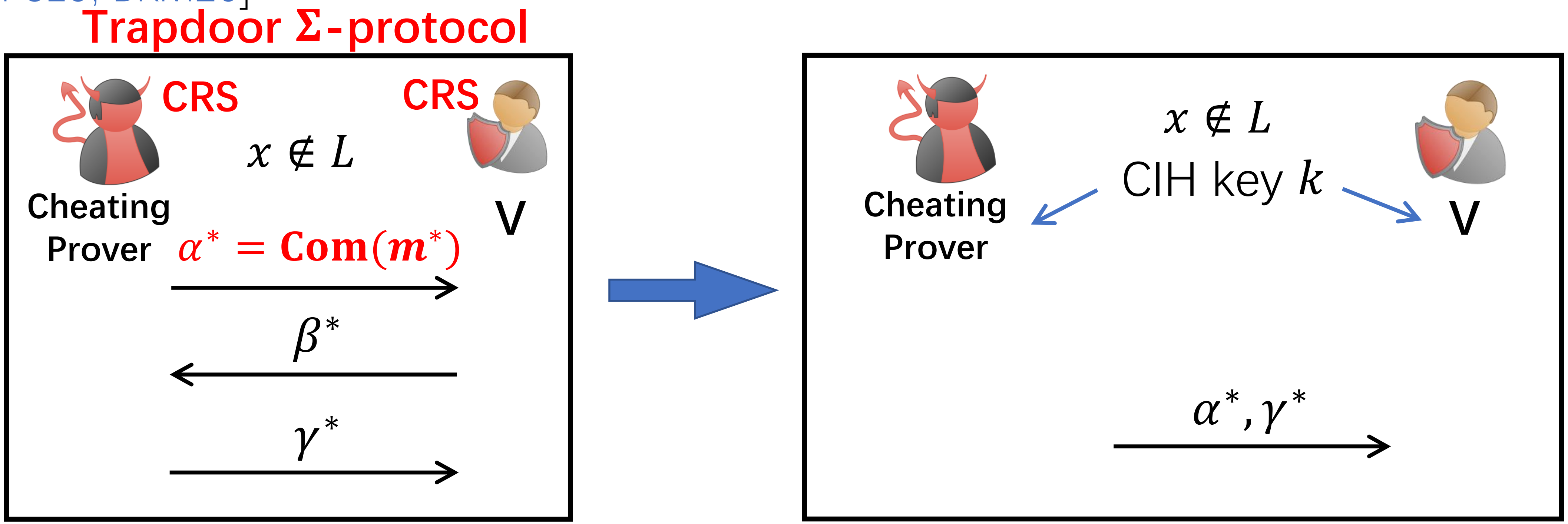
Trapdoor Σ-protocol



Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

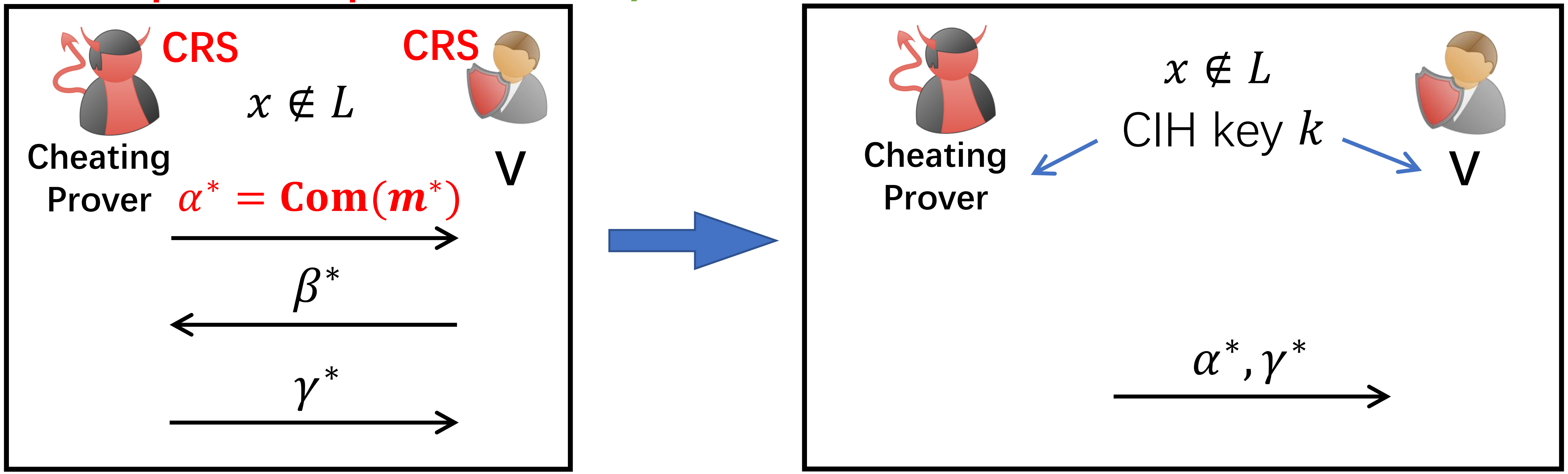


Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



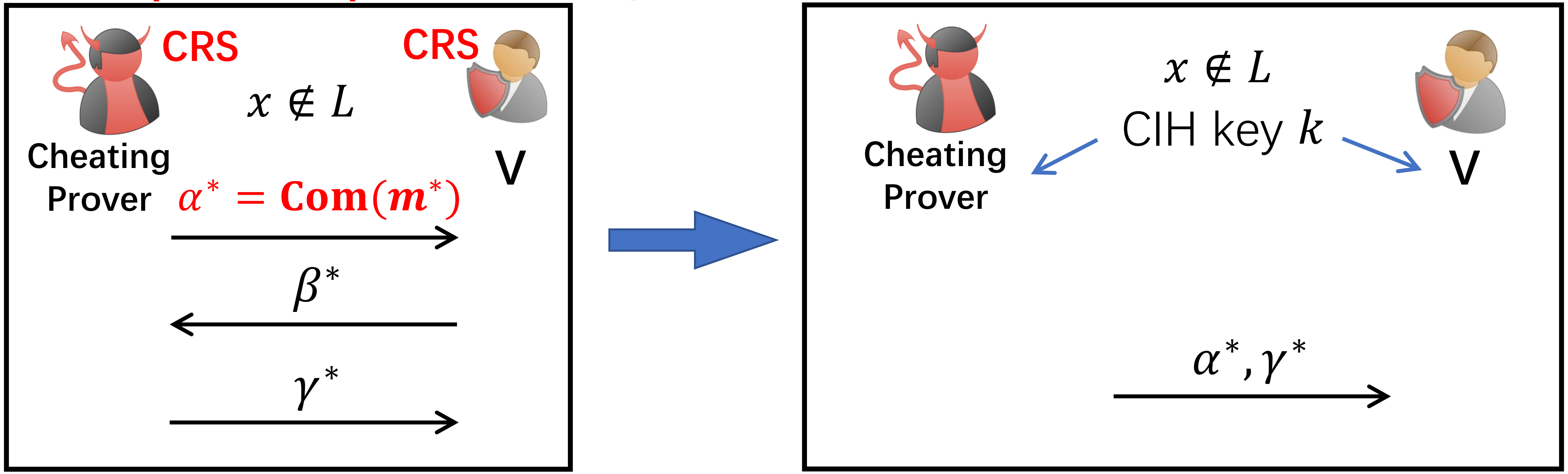
Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

Trapdoor Σ-protocol **trapdoor: td**



Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

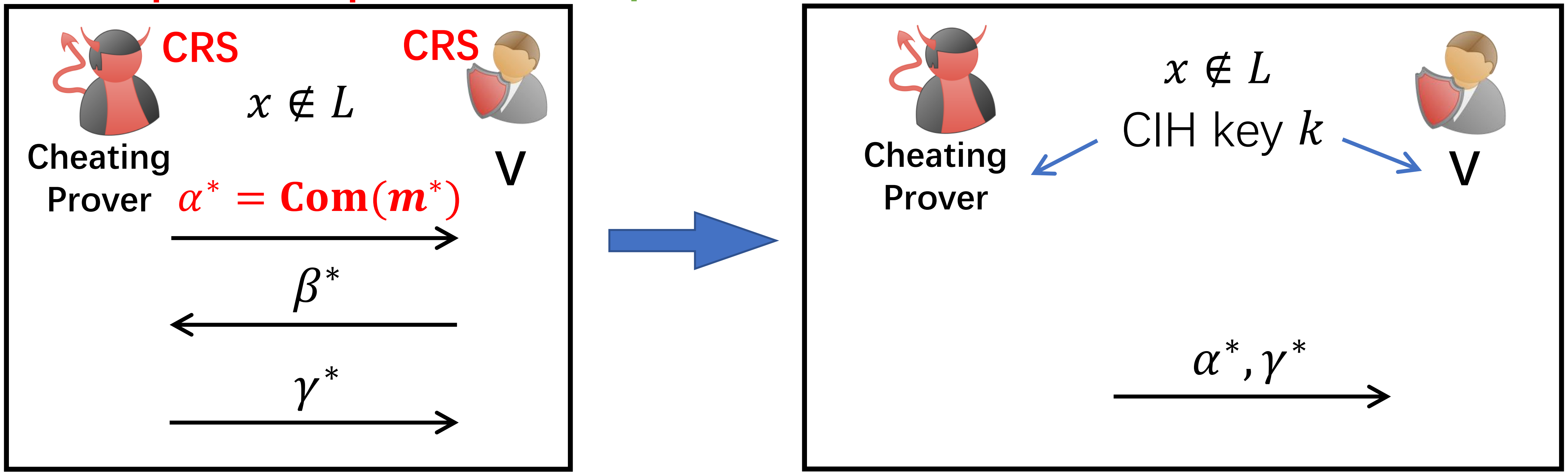
Trapdoor Σ-protocol **trapdoor: td**



BAD:

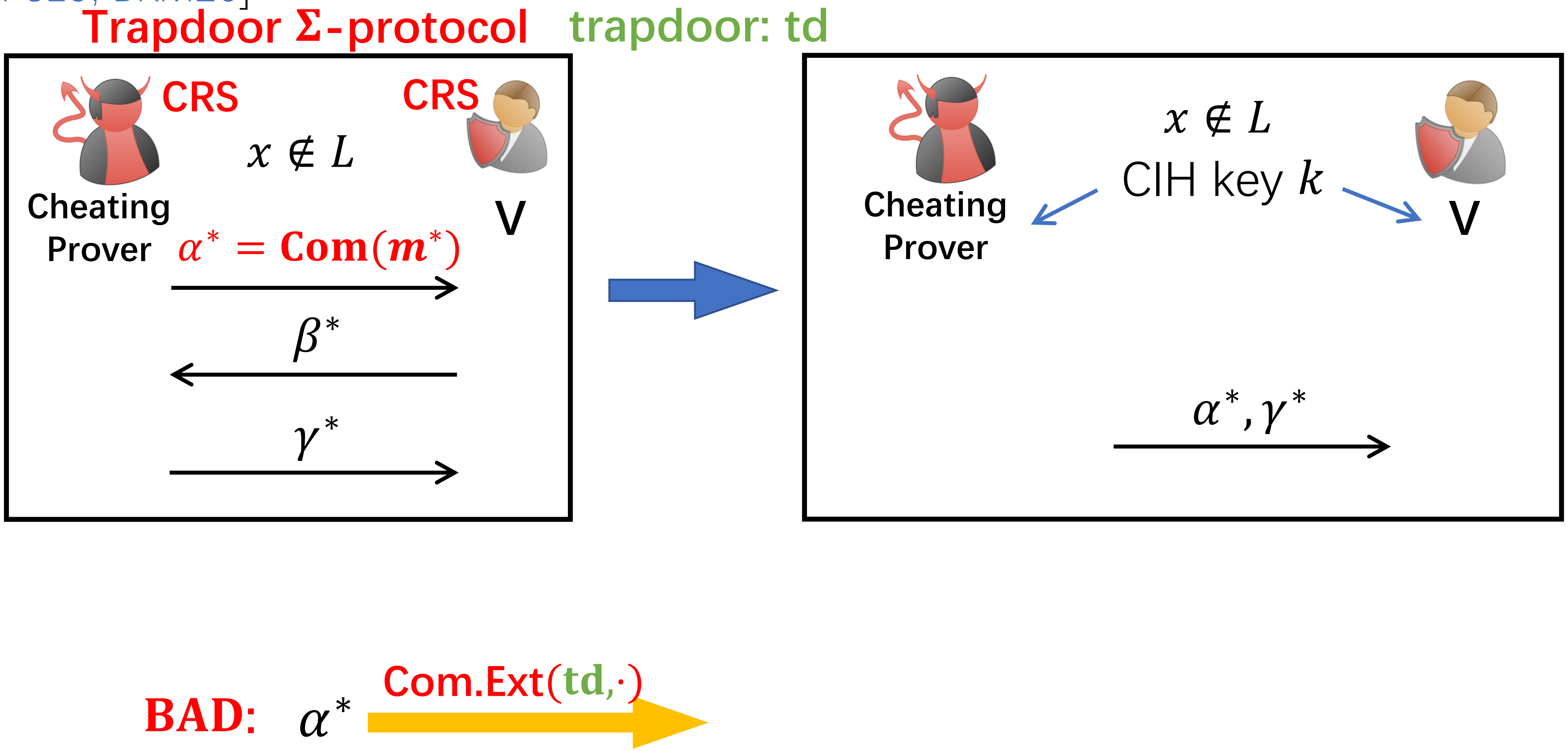
Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

Trapdoor Σ-protocol trapdoor: td



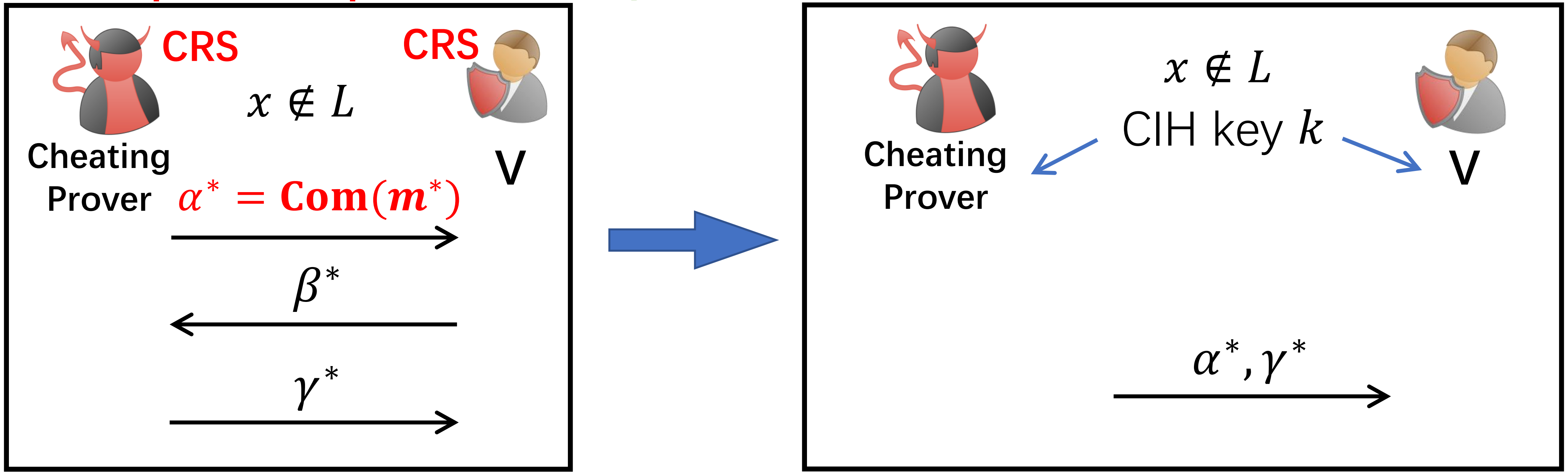
BAD: α^*

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]



Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

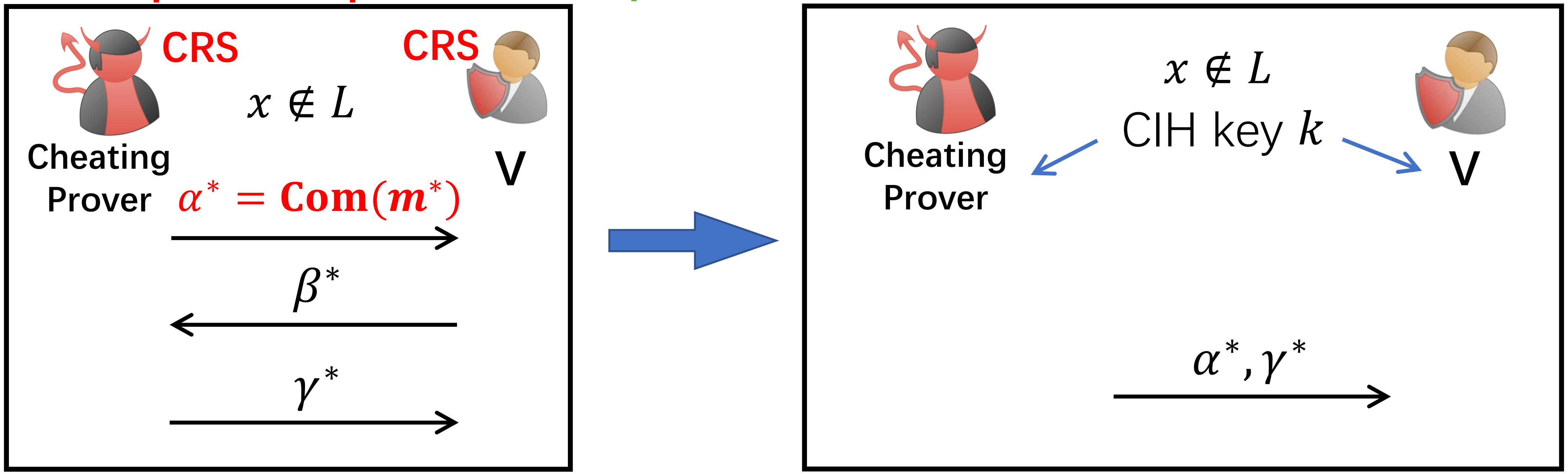
Trapdoor Σ-protocol   **trapdoor: td**



BAD: $\alpha^* \xrightarrow{\text{Com.Ext}(\mathbf{td},\cdot)} m^*$

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

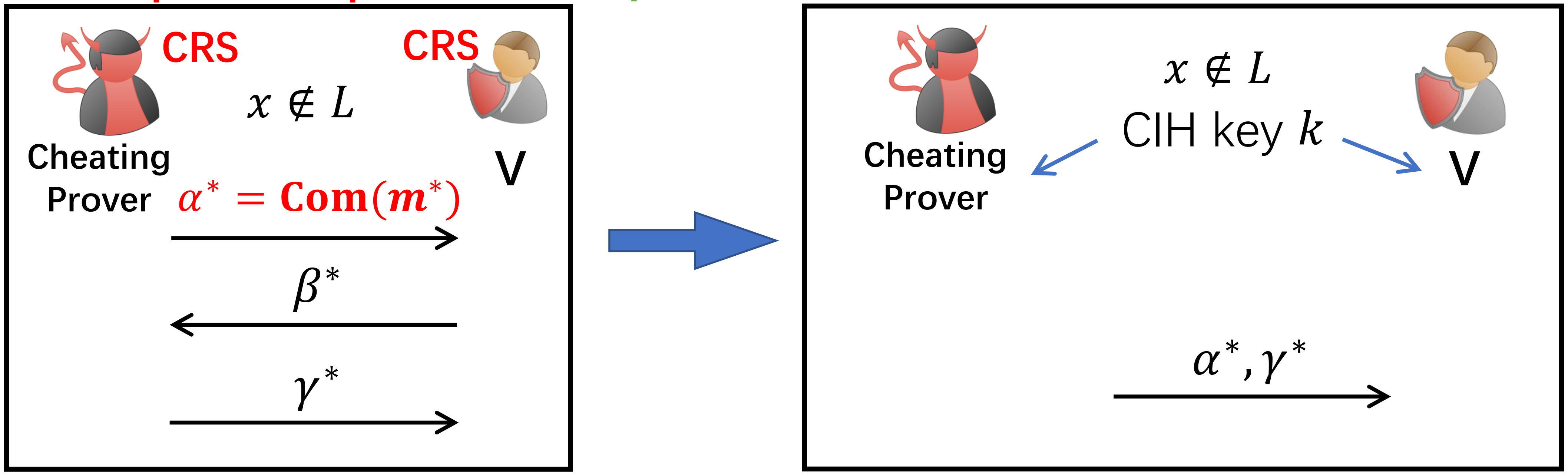
Trapdoor Σ-protocol **trapdoor: td**



BAD: $\alpha^* \xrightarrow{\text{Com.Ext}(\mathbf{td},\cdot)} m^* \longrightarrow$ the unique bad β^*

Fiat-Shamir: Soundness [CGH98, KRR17, CCRR18, HL18, CCHLRRW19, PS19, BKM20]

Trapdoor Σ-protocol **trapdoor: td**



Correlation Intractability needs to at least capture the Com.Ext(td,·) circuit

BAD: $\alpha^* \xrightarrow{\text{Com.Ext}(\mathbf{td},\cdot)} m^* \longrightarrow$ the unique bad β^*

Towards Instantiation from DDH: Main Challenges

- Recap of Fiat-Shamir
- **Main Challenges**
- ITDH for $\mathbf{TC}^0$ → CIH for $\mathbf{TC}^0$
- Construction of ITDH

Instantiate Fiat-Shamir from DDH

Instantiate Fiat-Shamir from DDH

- **Instantiate Trapdoor Commitment from DDH**

Instantiate Fiat-Shamir from DDH

- **Instantiate Trapdoor Commitment from DDH**

  Commitment → ElGamal Encryption
  Extraction → ElGamal Decryption

Instantiate Fiat-Shamir from DDH

- **Instantiate Trapdoor Commitment from DDH**

Commitment $\rightarrow$ ElGamal Encryption
Extraction $\rightarrow$ ElGamal Decryption

Instantiate Fiat-Shamir from DDH

- **Instantiate Trapdoor Commitment from DDH**

Commitment → ElGamal Encryption
Extraction → ElGamal Decryption

If we have CIH for *ElGamal Decryption circuit* from DDH,
then we can hope to construct NIZKs from DDH.

Previous CIH from DDH [BKM20]

Previous CIH from DDH [BKM20]

$\boldsymbol{H}_{\boldsymbol{k}}(\cdot)$

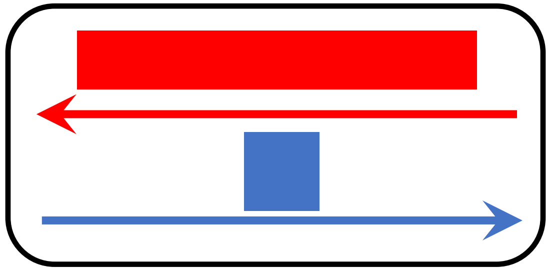
CIH for approximable relations of $\boldsymbol{O(1)}$-degree poly.

Previous CIH from DDH [BKM20]

$\Longrightarrow$ $\boldsymbol{H_k}(\cdot)$

CIH for approximable relations of $\boldsymbol{O(1)}$-degree poly.

Previous CIH from DDH [BKM20]

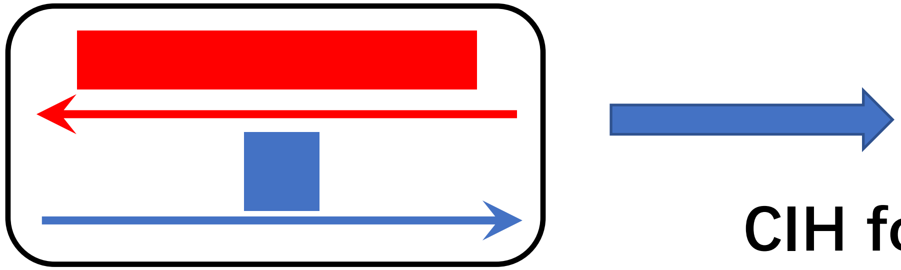


TDH for $\boldsymbol{O(1)}$-degree poly.

$\boldsymbol{H_k(\cdot)}$

CIH for approximable relations of $\boldsymbol{O(1)}$-degree poly.

Previous CIH from DDH [BKM20]



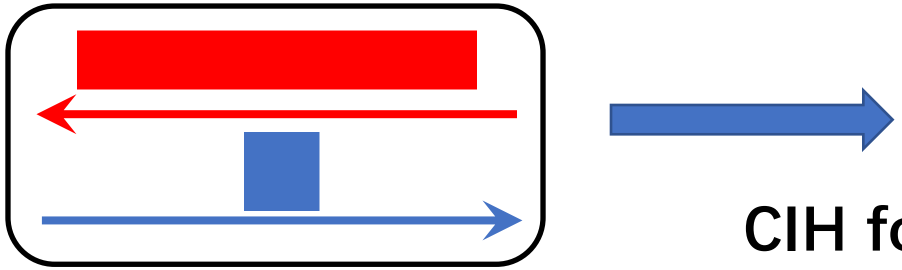
TDH for **$O(1)$-degree poly.**

$H_k(\cdot)$

CIH for approximable relations of $O(1)$-degree poly.

- [BKM20] used trapdoor commitment from LPN, where
  $\mathrm{Com.Extraction}(\mathrm{td},\cdot) \in$ { approximate $O(1)$-degree poly. }

Previous CIH from DDH [BKM20]



TDH for $\boldsymbol{O(1)}$-degree poly.

$\boldsymbol{H_k(\cdot)}$

CIH for approximable relations of $\boldsymbol{O(1)}$-degree poly.

Approximating the ElGamal Decryption by $\boldsymbol{O(1)}$-degree poly is not known

- [BKM20] used trapdoor commitment from LPN, where $\mathrm{Com.Extraction}(\mathrm{td},\cdot) \in \{$ approximate $O(1)$-degree poly. $\}$

What circuit class of CIH is sufficient to instantiate Fiat-Shamir from DDH?

What circuit class of CIH is sufficient to instantiate Fiat-Shamir from DDH?

How to build CIH for such a circuit class?

Our Approach

Our Approach

CIH for $\mathbf{TC^0}$ suffices for building NIZKs from DDH

Our Approach

CIH for $\mathbf{TC^0}$ suffices for building NIZKs from DDH

Our Approach

CIH for $\mathbf{TC^0}$ suffices for building NIZKs from DDH

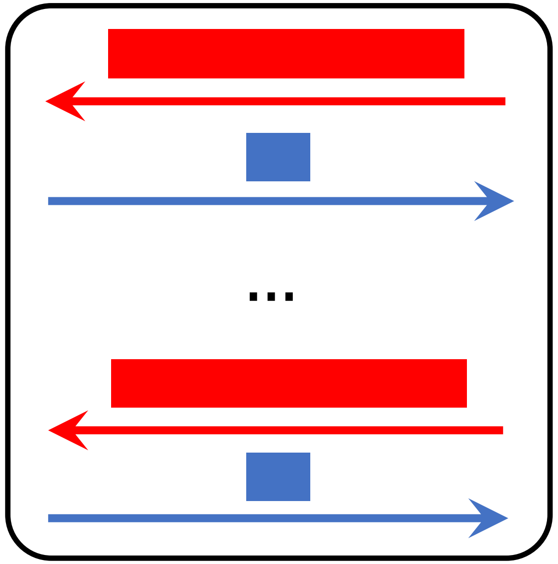
<u>Construct CIH for $\mathbf{TC^0}$</u>

Our Approach

CIH for $\mathbf{TC^0}$ suffices for building NIZKs from DDH

<u>**Construct CIH for $\mathbf{TC^0}$**</u>

$\boldsymbol{O(1)}$**-round ITDH for $\mathbf{TC^0}$**

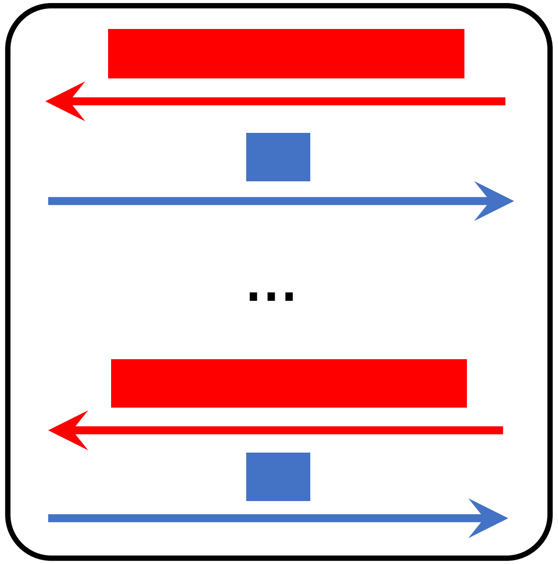


Our Approach

CIH for $\mathbf{TC^0}$ suffices for building NIZKs from DDH

<u>Construct CIH for $\mathbf{TC^0}$</u>

$\boldsymbol{O}(\mathbf{1})$-round ITDH for $\mathbf{TC^0}$



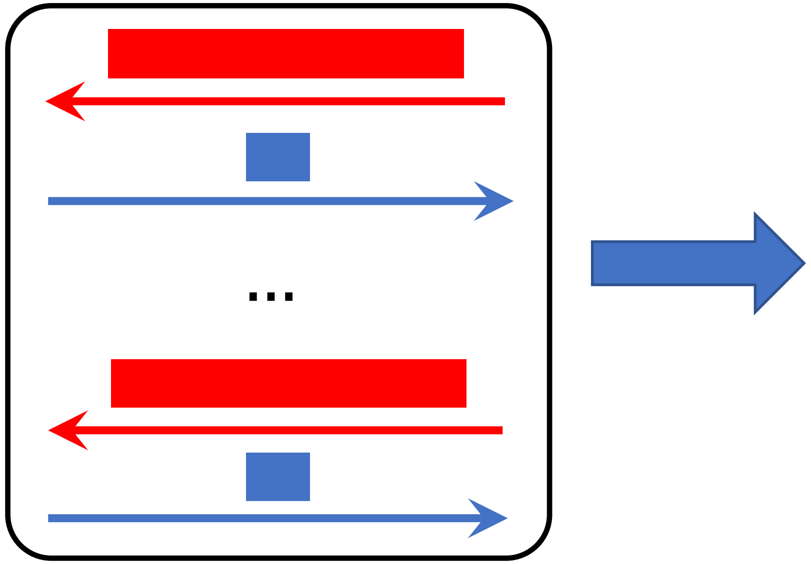
Compute beyond $\boldsymbol{O}(\mathbf{1})$-degree poly by leveraging interaction

Our Approach

CIH for $\mathbf{TC^0}$ suffices for building NIZKs from DDH

Construct CIH for $\mathbf{TC^0}$

$\boldsymbol{O(1)}$**-round ITDH for $\mathbf{TC^0}$**



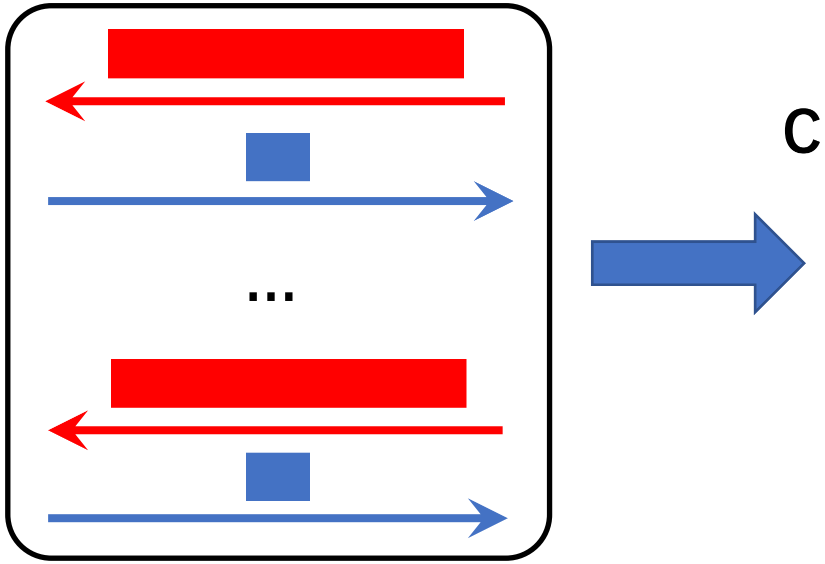
Compute beyond $\boldsymbol{O(1)}$-degree poly by leveraging interaction

Our Approach

CIH for $\mathbf{TC}^0$ suffices for building NIZKs from DDH

Construct CIH for $\mathbf{TC}^0$

$O(1)$-round ITDH for $\mathbf{TC}^0$



CIH for $\mathbf{TC}^0$

$H_k(\cdot)$

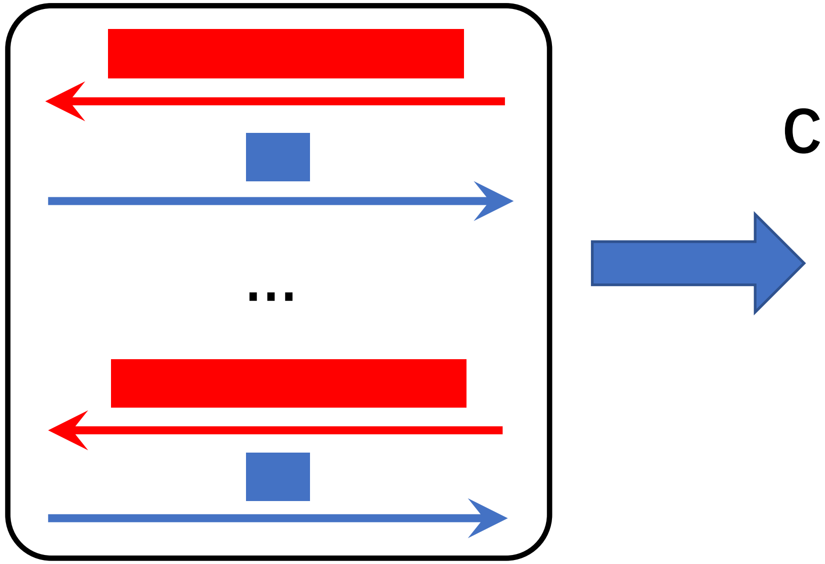
Compute beyond $O(1)$-degree poly by leveraging interaction

Our Approach

CIH for $\mathbf{TC^0}$ suffices for building NIZKs from DDH

Construct CIH for $\mathbf{TC^0}$

$\boldsymbol{O(1)}$-round ITDH for $\mathbf{TC^0}$



CIH for $\mathbf{TC^0}$

$H_k(\cdot)$

Compute beyond $\boldsymbol{O(1)}$-degree poly
by leveraging interaction

Interactive TDH → CIH

- Recap of Fiat-Shamir
- Main Challenges
- **ITDH for $\mathbf{TC^0}$ → CIH for $\mathbf{T}\boldsymbol{C}^{\mathbf{0}}$**
- Construction of ITDH

Recall: Interactive TDH

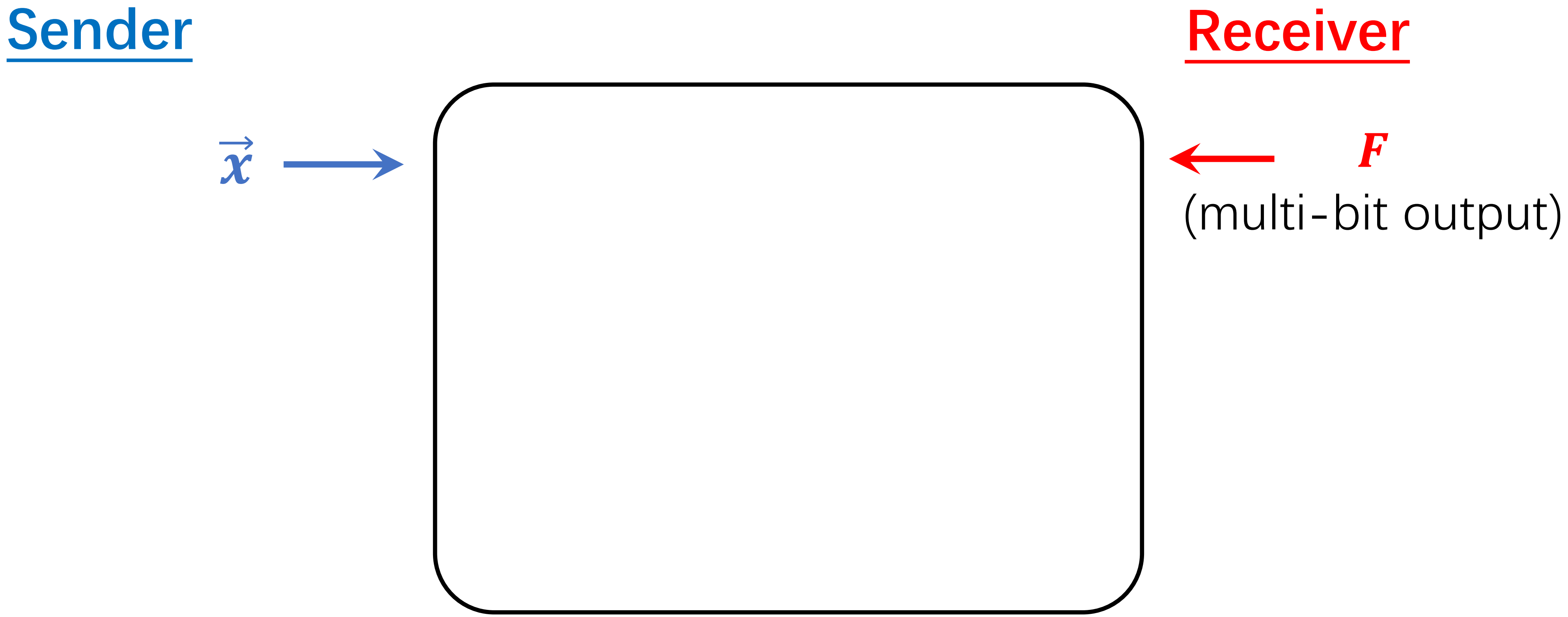
Recall: Interactive TDH

Sender

Recall: Interactive TDH

Sender **Receiver**

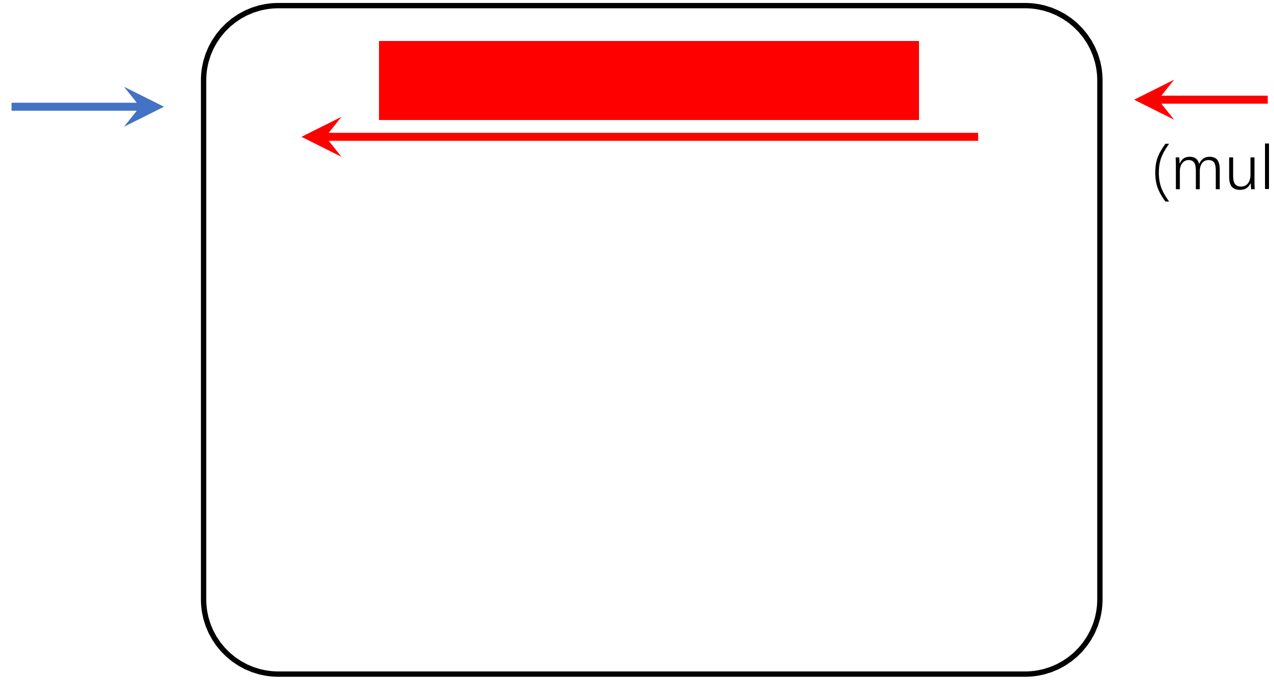
Recall: Interactive TDH



Recall: Interactive TDH

Sender | **Receiver**

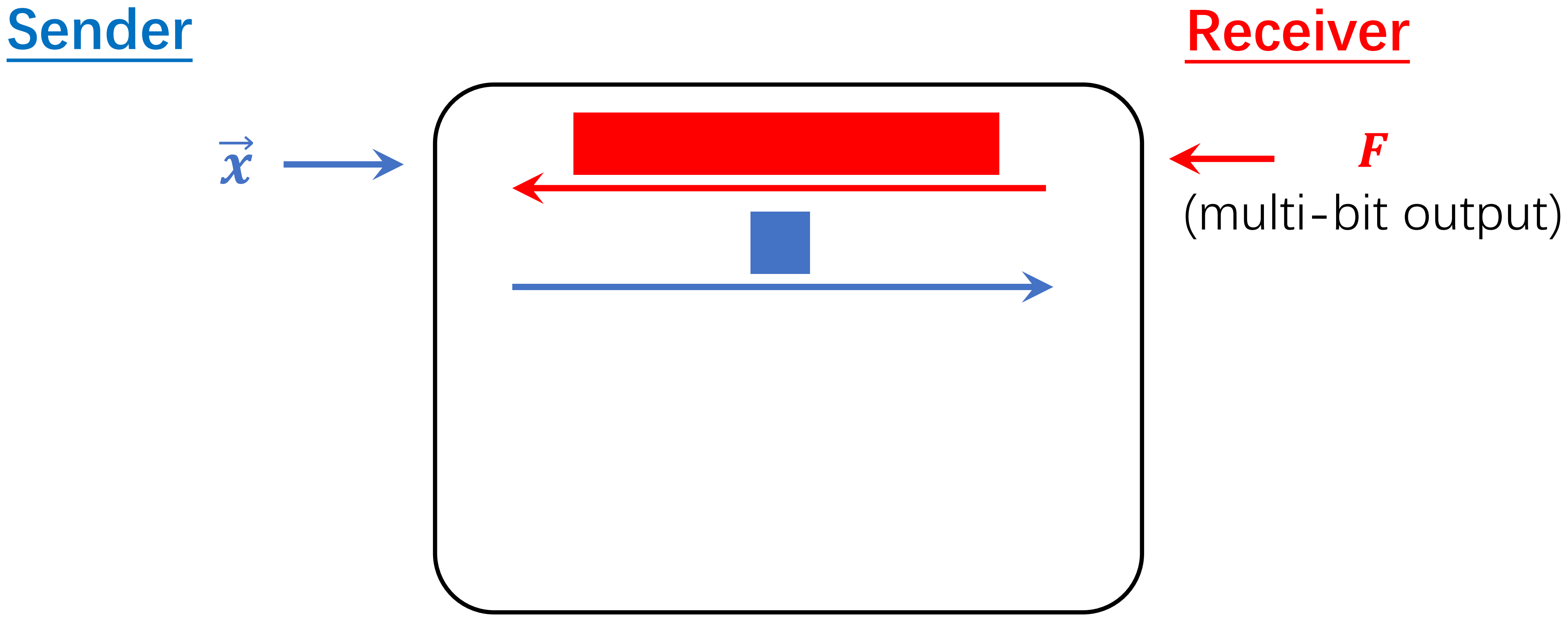
$\vec{x}$



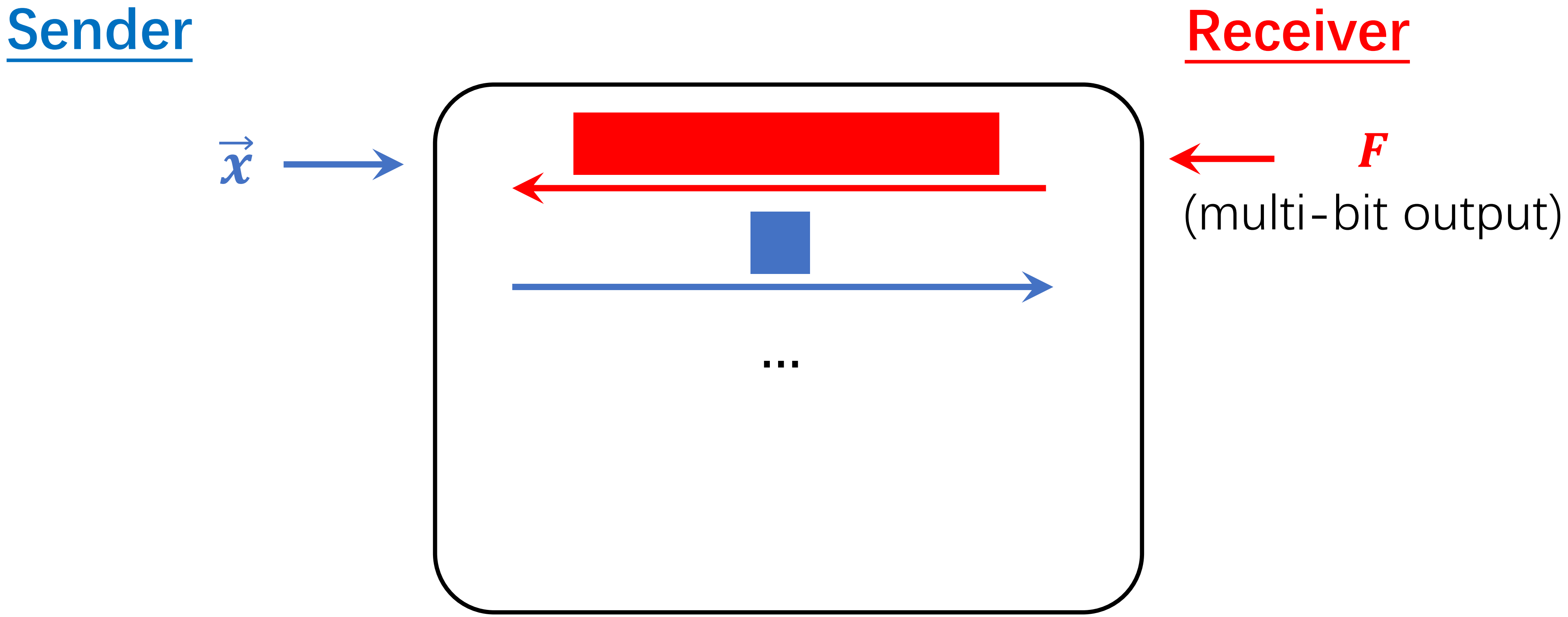
F

(multi-bit output)

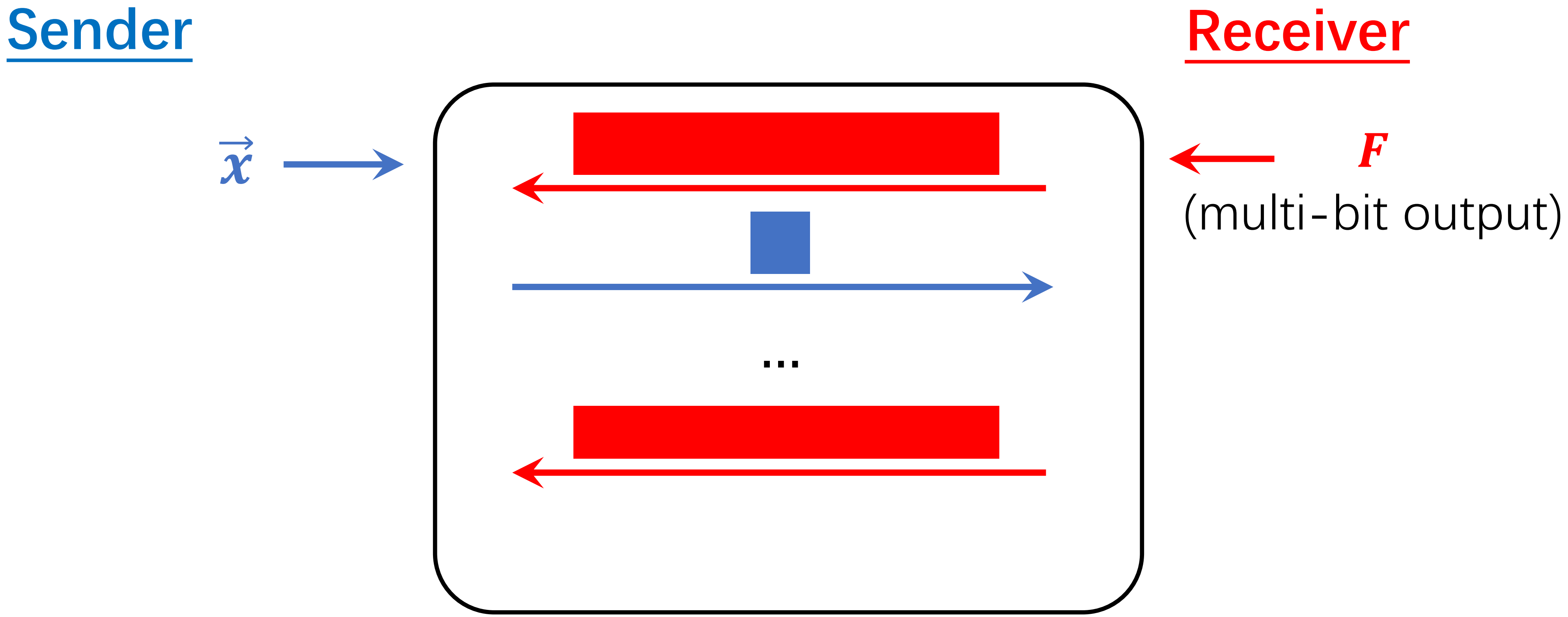
Recall: Interactive TDH



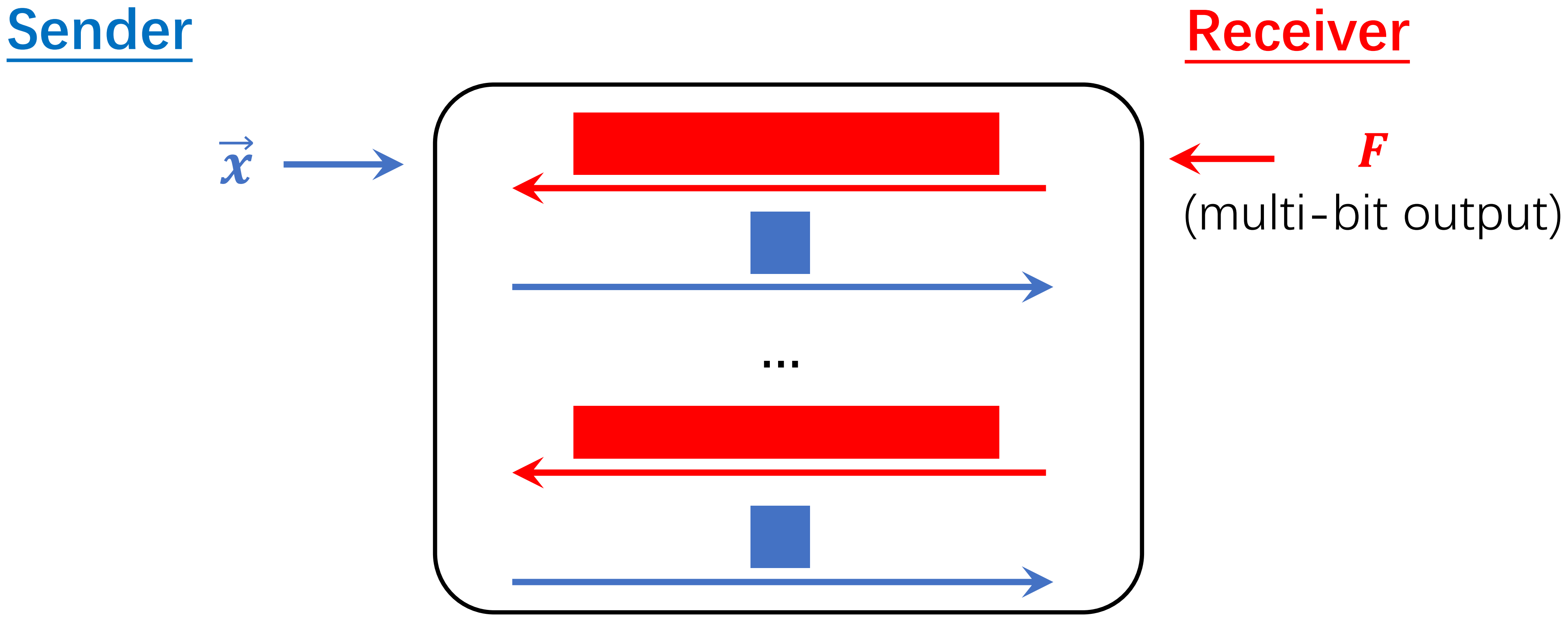
Recall: Interactive TDH



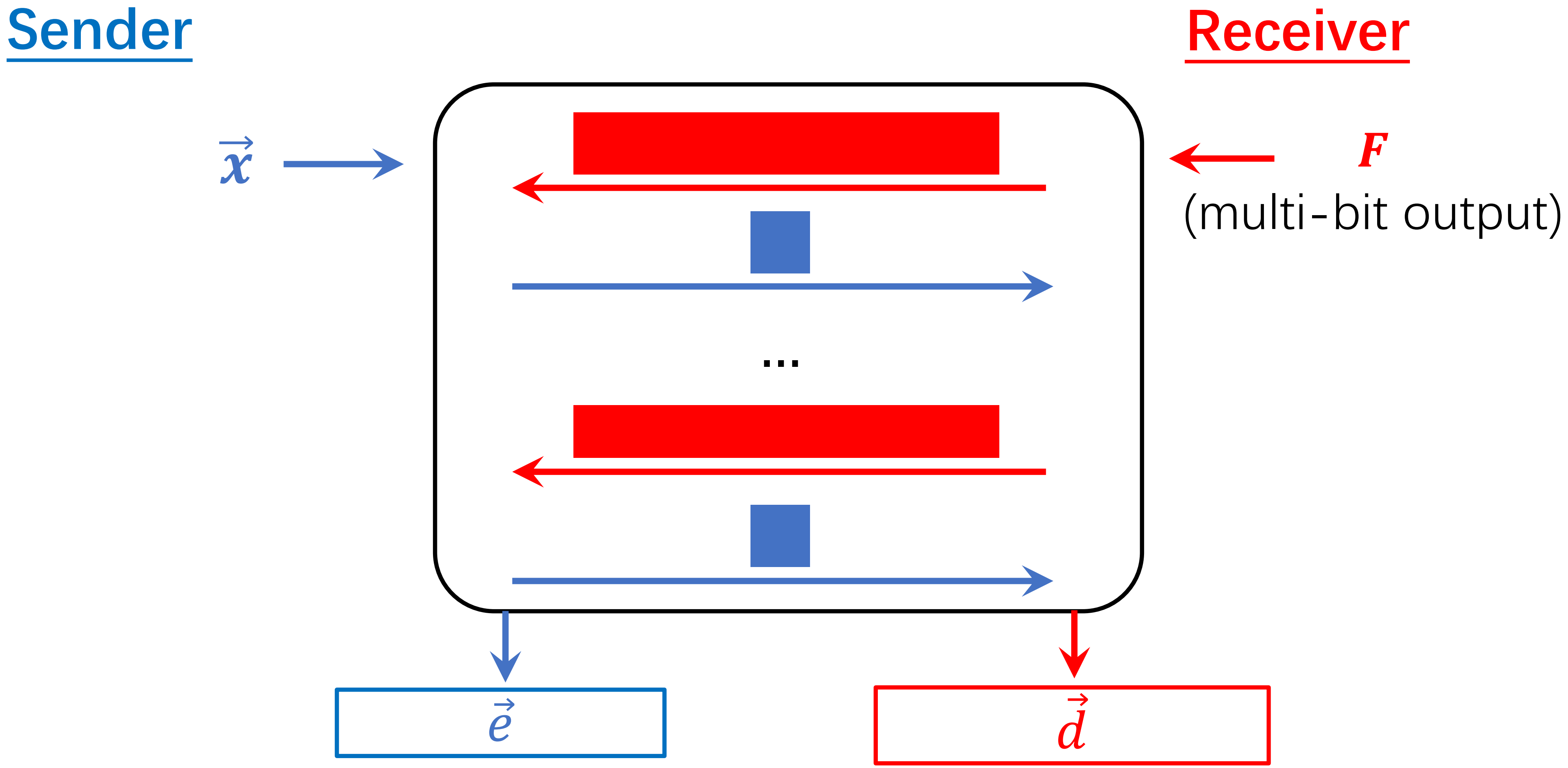
Recall: Interactive TDH



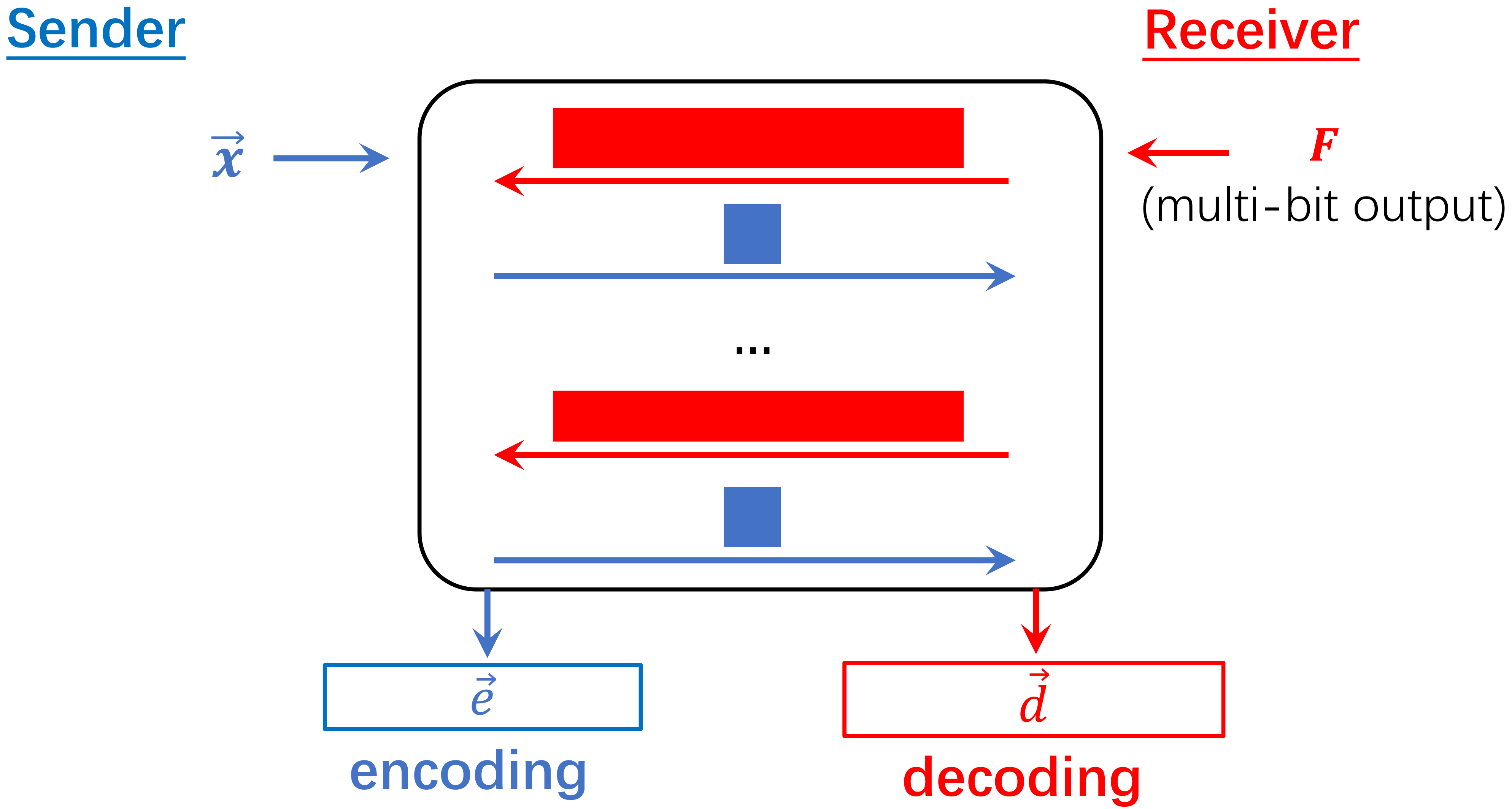
Recall: Interactive TDH



Recall: Interactive TDH

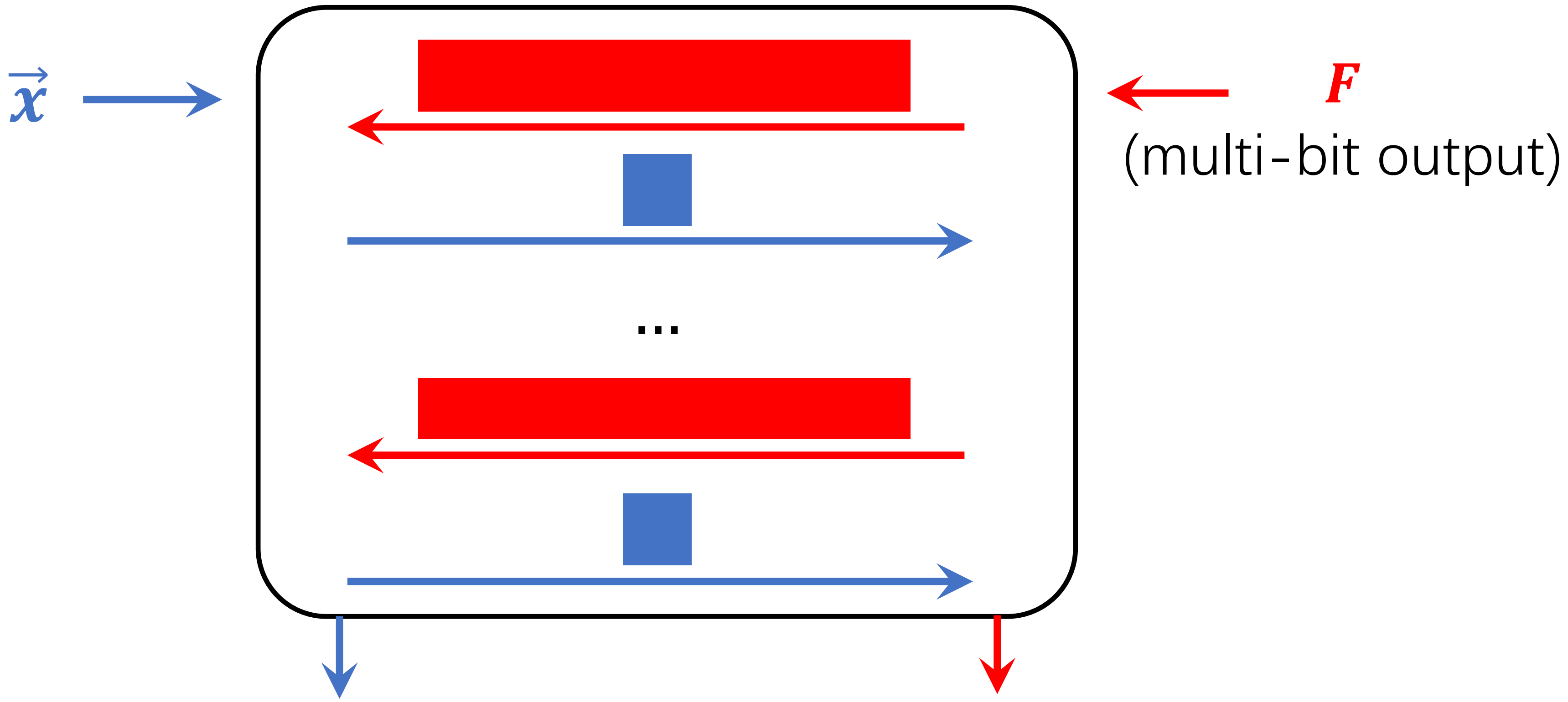


Recall: Interactive TDH



Recall: Interactive TDH

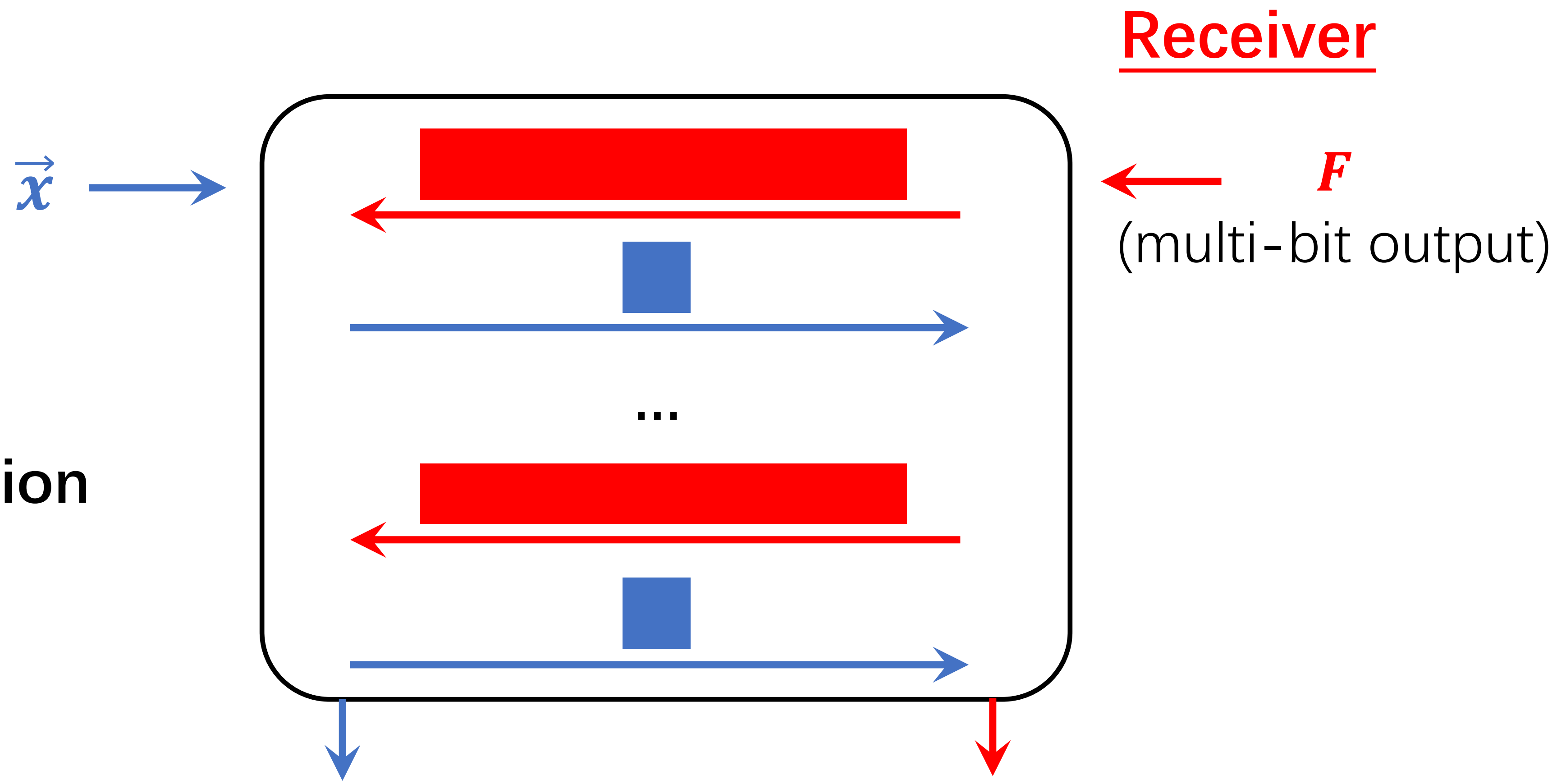
<u>Sender</u> <u>Receiver</u>



- **Additive reconstruction:** $\boxed{\vec{e}} \oplus \boxed{\vec{d}} = F(\vec{x})$

encoding decoding

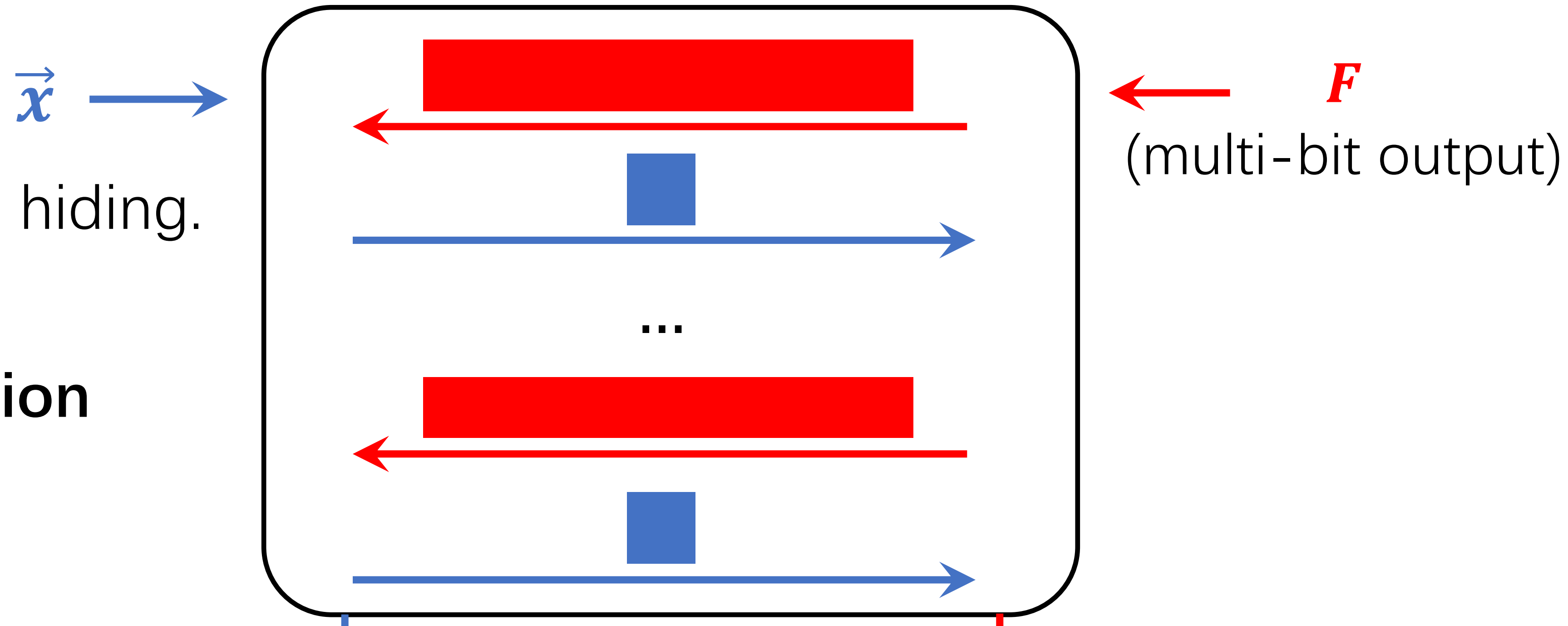
Recall: Interactive TDH



- **Laconic communication on sender side:** $|\blacksquare| \leq \lambda$
- **Additive reconstruction:** $\underbrace{\boxed{\vec{e}}}_{\text{encoding}} \oplus \underbrace{\boxed{\vec{d}}}_{\text{decoding}} = F(\vec{x})$

Recall: Interactive TDH

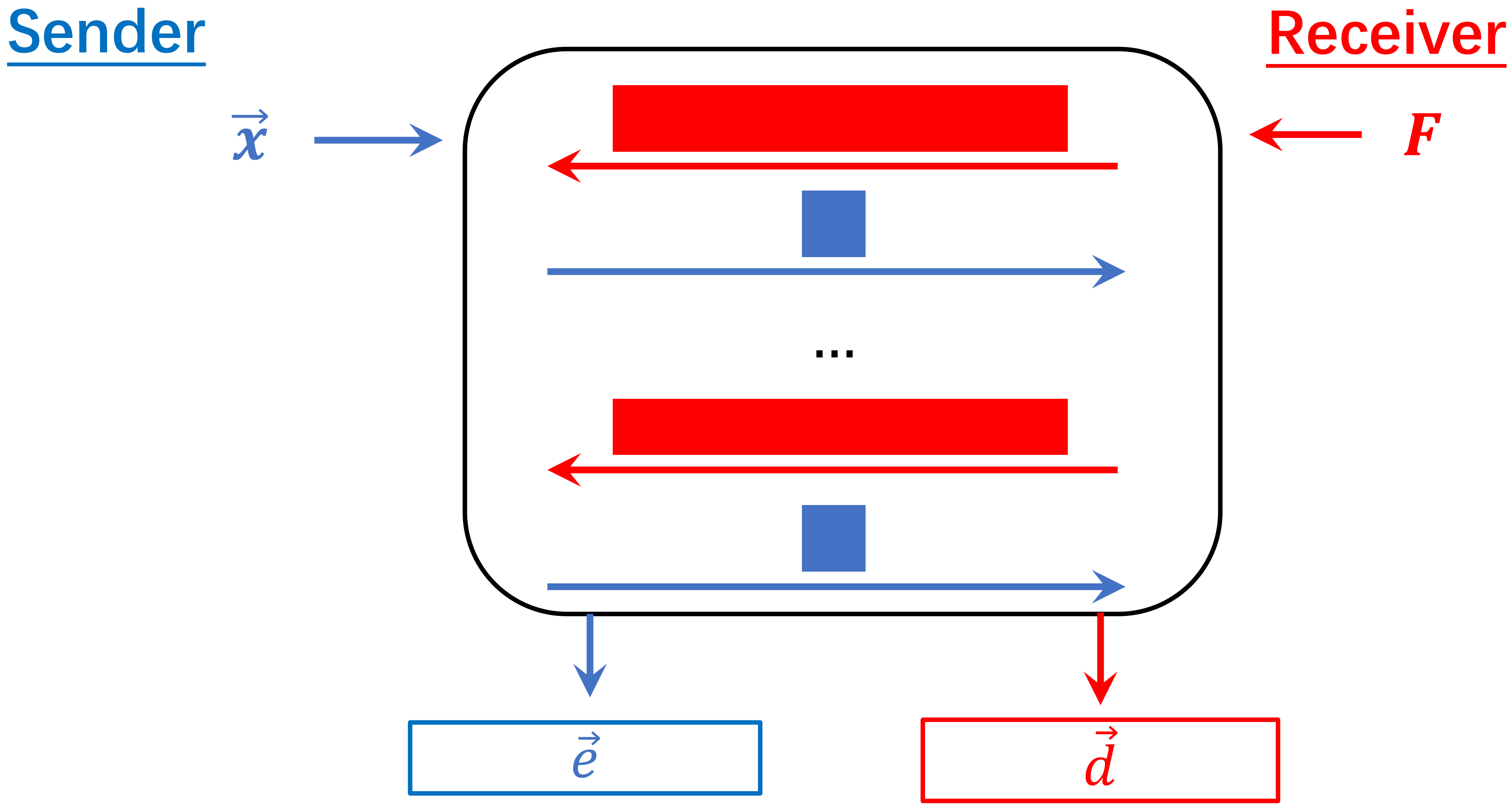
Sender **Receiver**



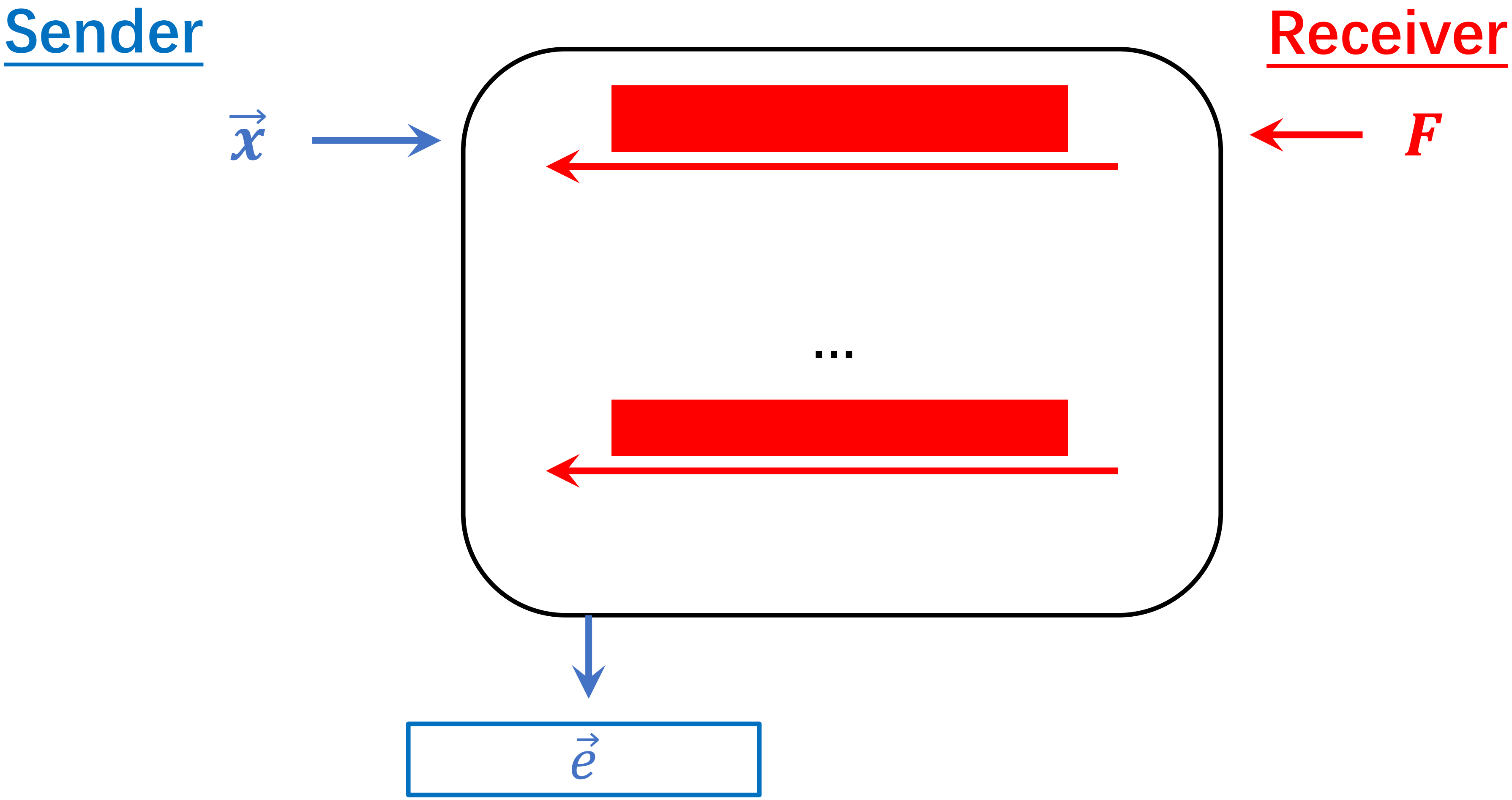
- **Function Hiding:** F is hiding.
- **Laconic communication on sender side:** $|\blacksquare| \leq \lambda$
- **Additive reconstruction:** $\boxed{\vec{e}} \oplus \boxed{\vec{d}} = F(\vec{x})$

encoding decoding

CIH from Interactive TDH

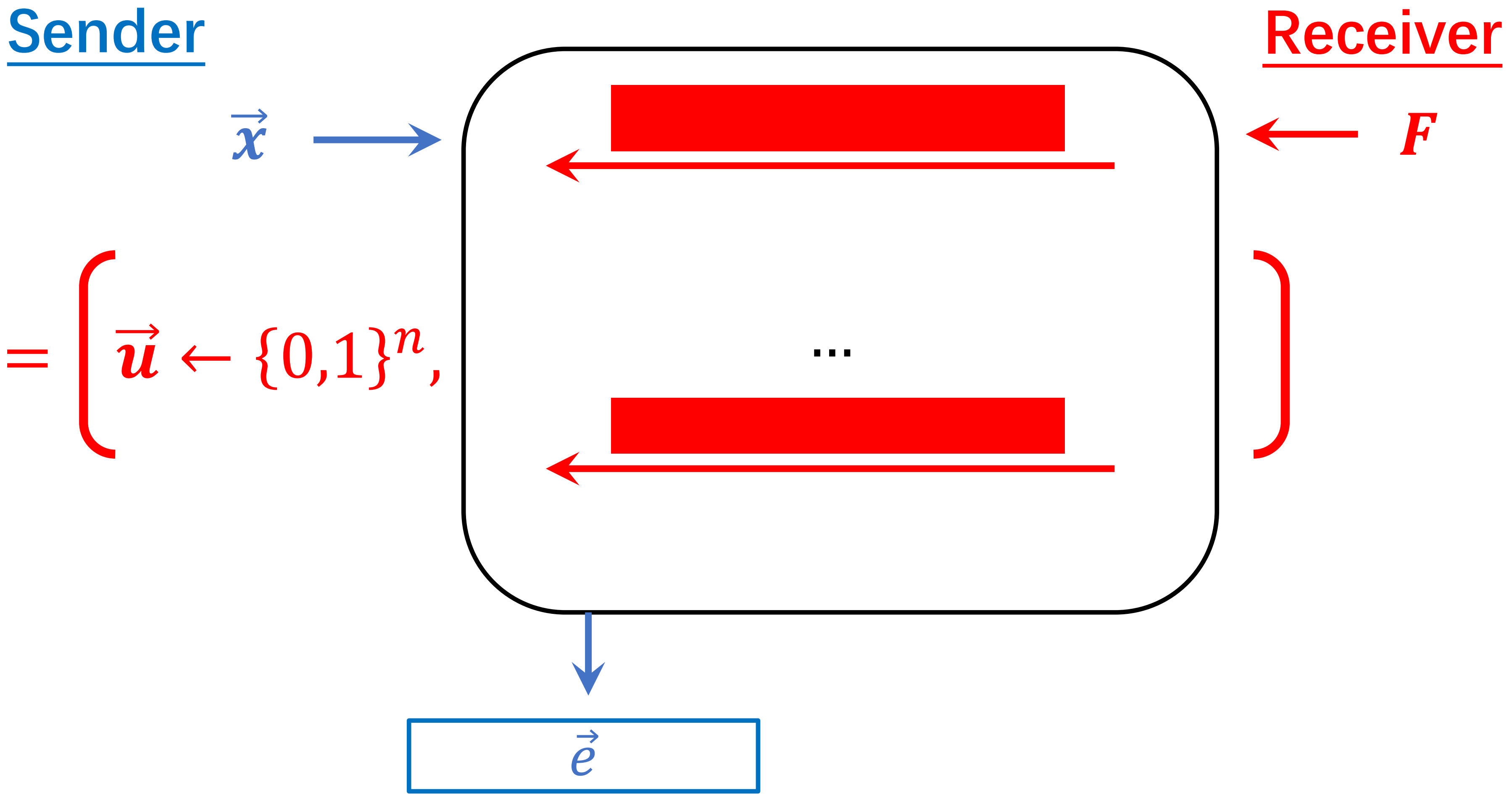


CIH from Interactive TDH

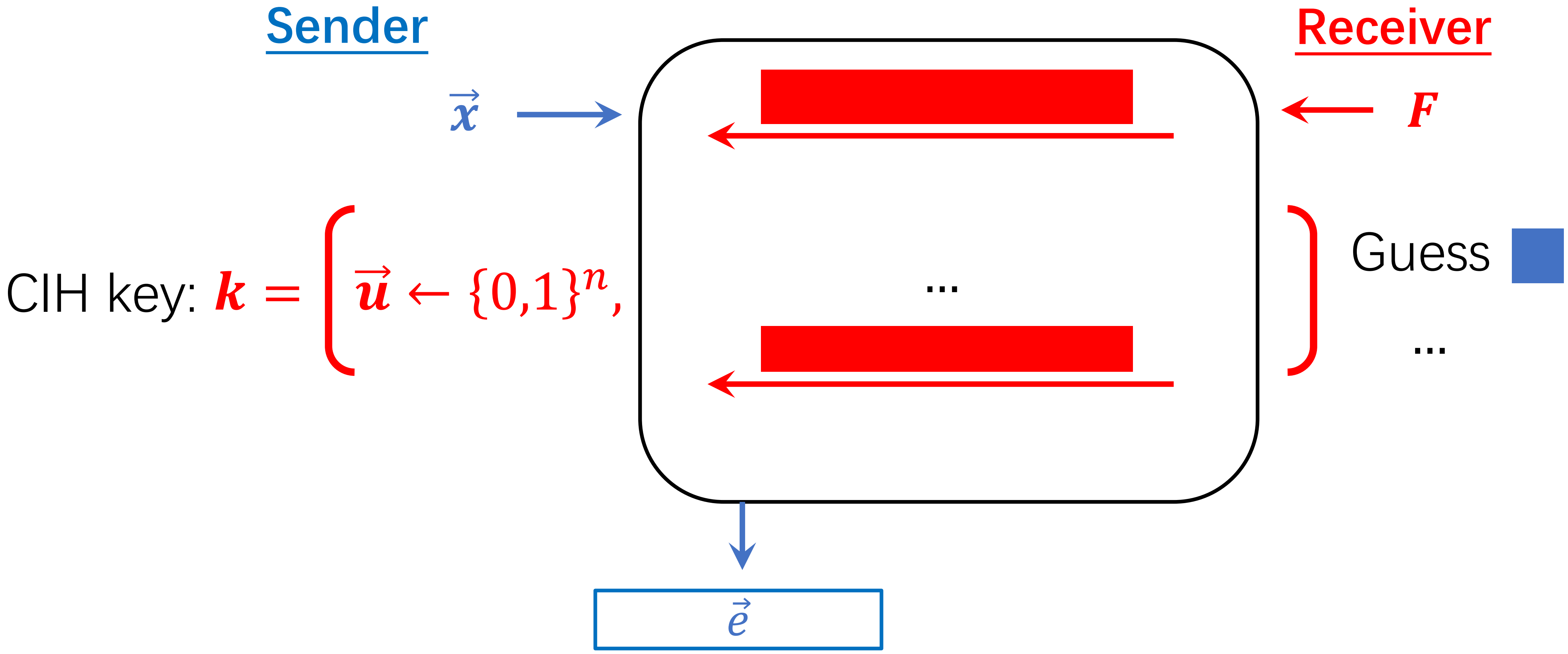


CIH from Interactive TDH

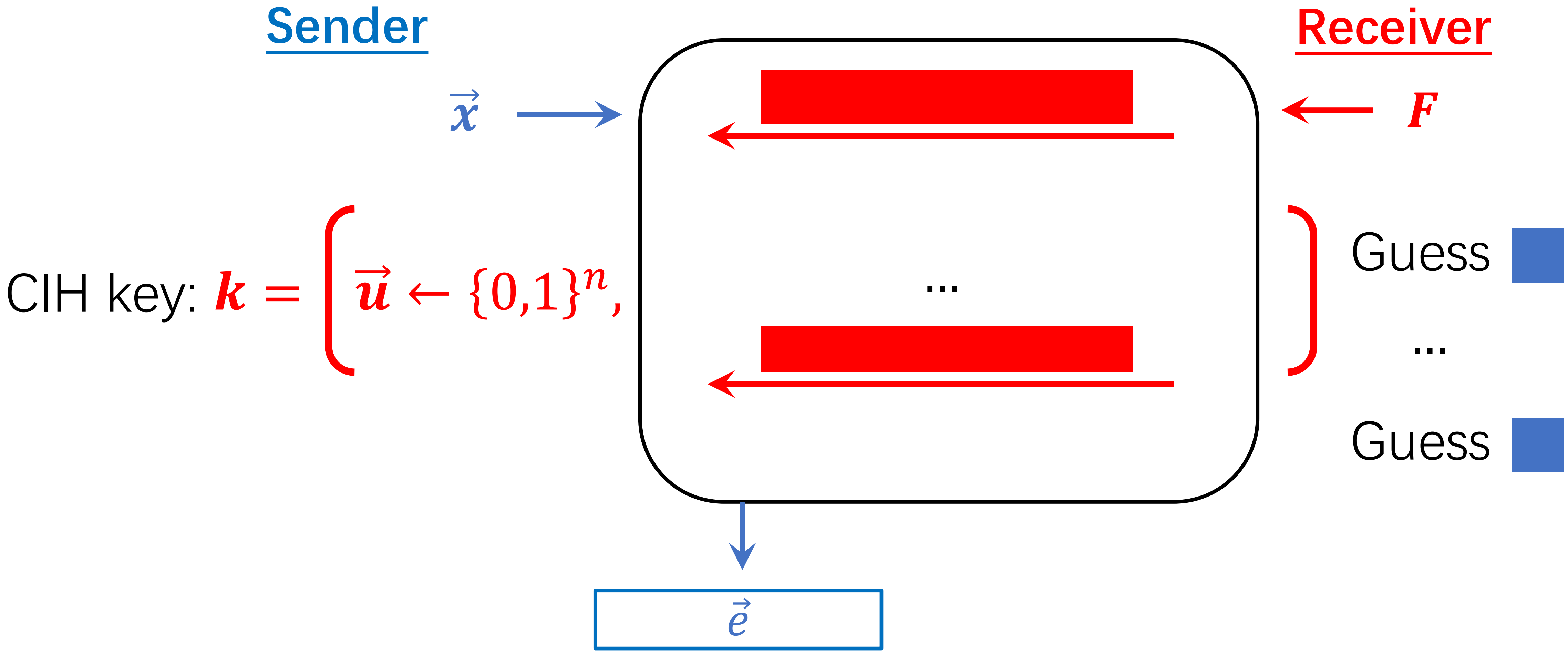
CIH key: $\boldsymbol{k} = \left(\vec{\boldsymbol{u}} \leftarrow \{0,1\}^n, \right.$ … $\left. \right)$



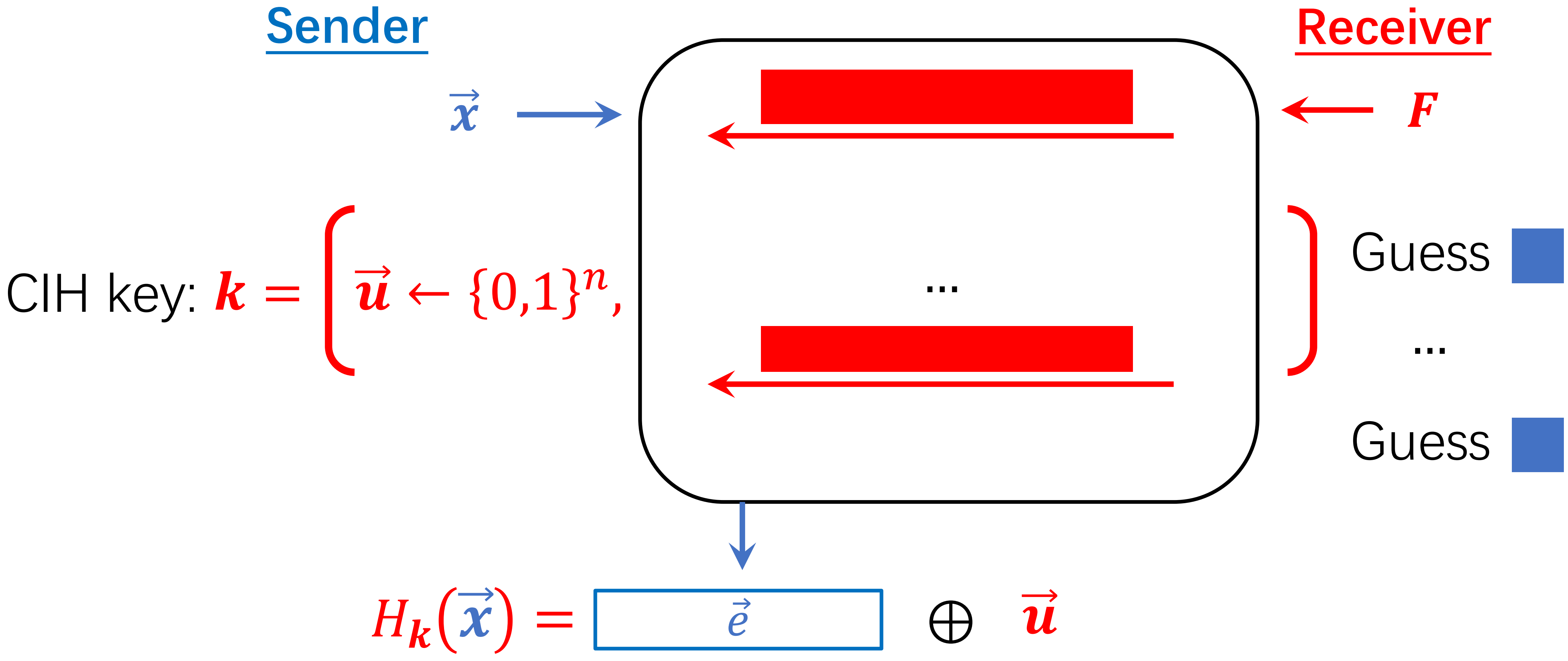
CIH from Interactive TDH



CIH from Interactive TDH



CIH from Interactive TDH



Recall: Correlation Intractable for $\mathcal{F}$

Recall: Correlation Intractable for $\boldsymbol{\mathcal{F}}$

$\forall$ fixed $\boldsymbol{F} \in \boldsymbol{\mathcal{F}}$

Recall: Correlation Intractable for $\boldsymbol{\mathcal{F}}$

$\forall$ fixed $\boldsymbol{F} \in \boldsymbol{\mathcal{F}}$

Recall: Correlation Intractable for $\mathcal{F}$

$\forall$ fixed $F \in \mathcal{F}$

$k \leftarrow \mathbf{Gen}(1^\lambda)$

Recall: Correlation Intractable for $\mathcal{F}$

$\forall$ fixed $F \in \mathcal{F}$

$k \leftarrow \mathbf{Gen}(1^\lambda)$

$\vec{x}$

Recall: Correlation Intractable for $\mathcal{F}$

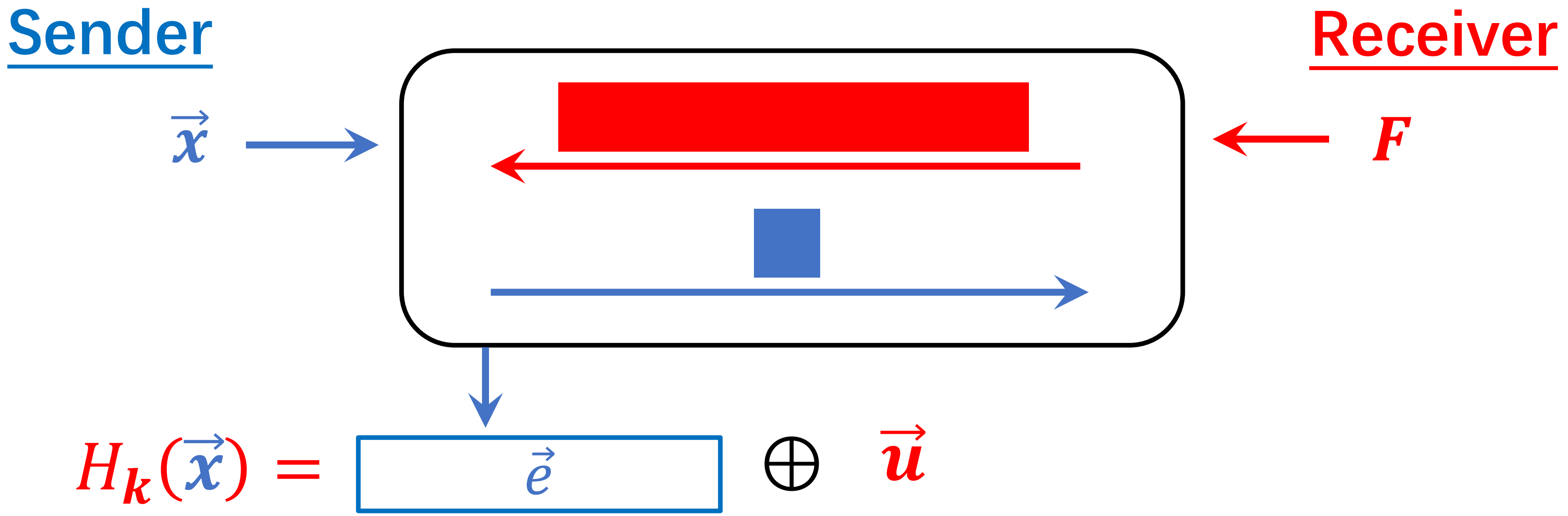
$\forall$ fixed $F \in \mathcal{F}$

$k \leftarrow \mathbf{Gen}(1^\lambda)$

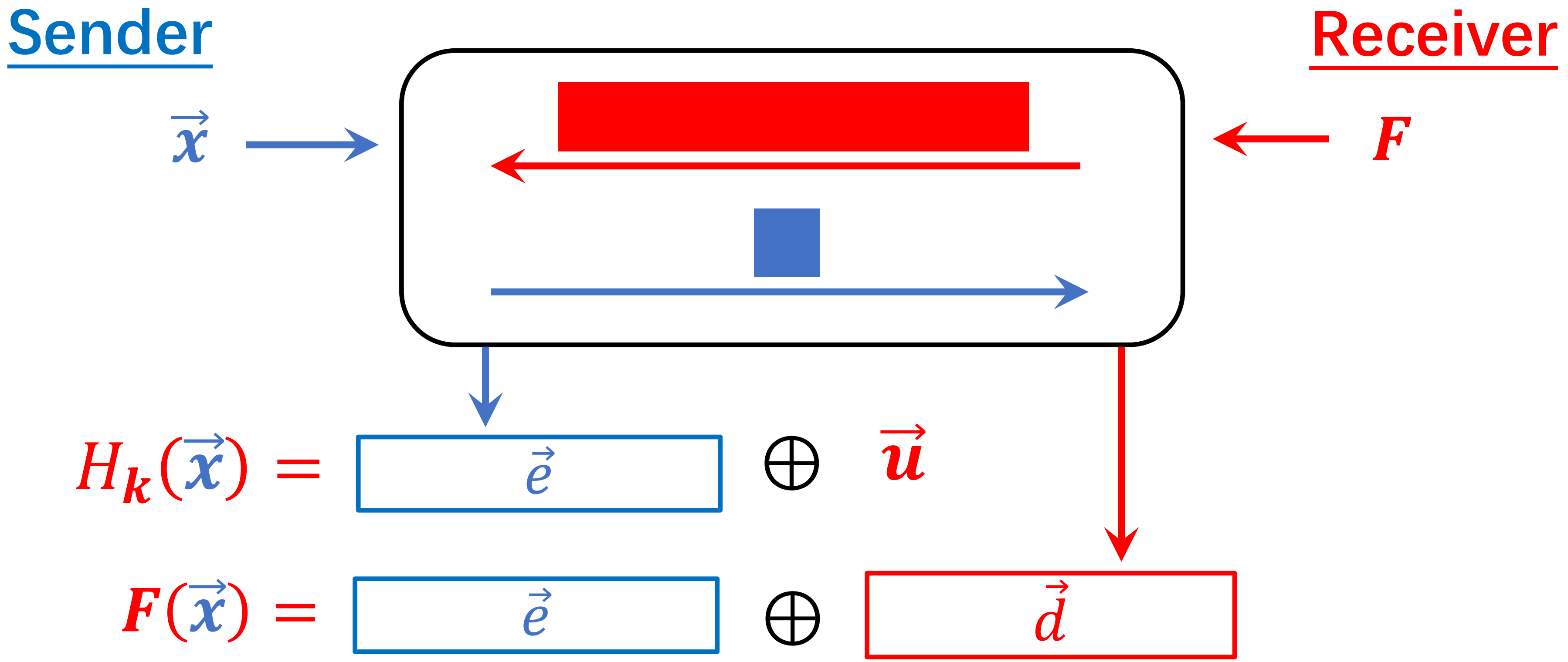
$\vec{x}$

$\Pr_{,k}[H_k(\vec{x}) = F(\vec{x})] \le \mathrm{negl}$

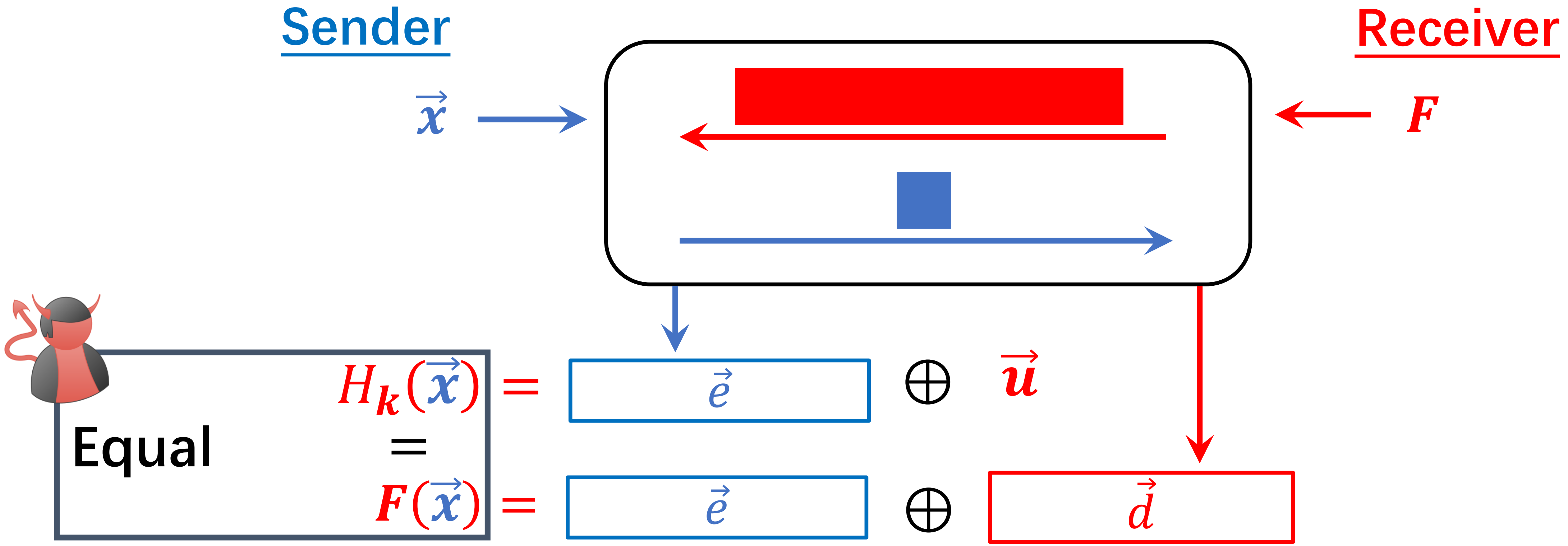
Proof of Correlation Intractability [BKM20]



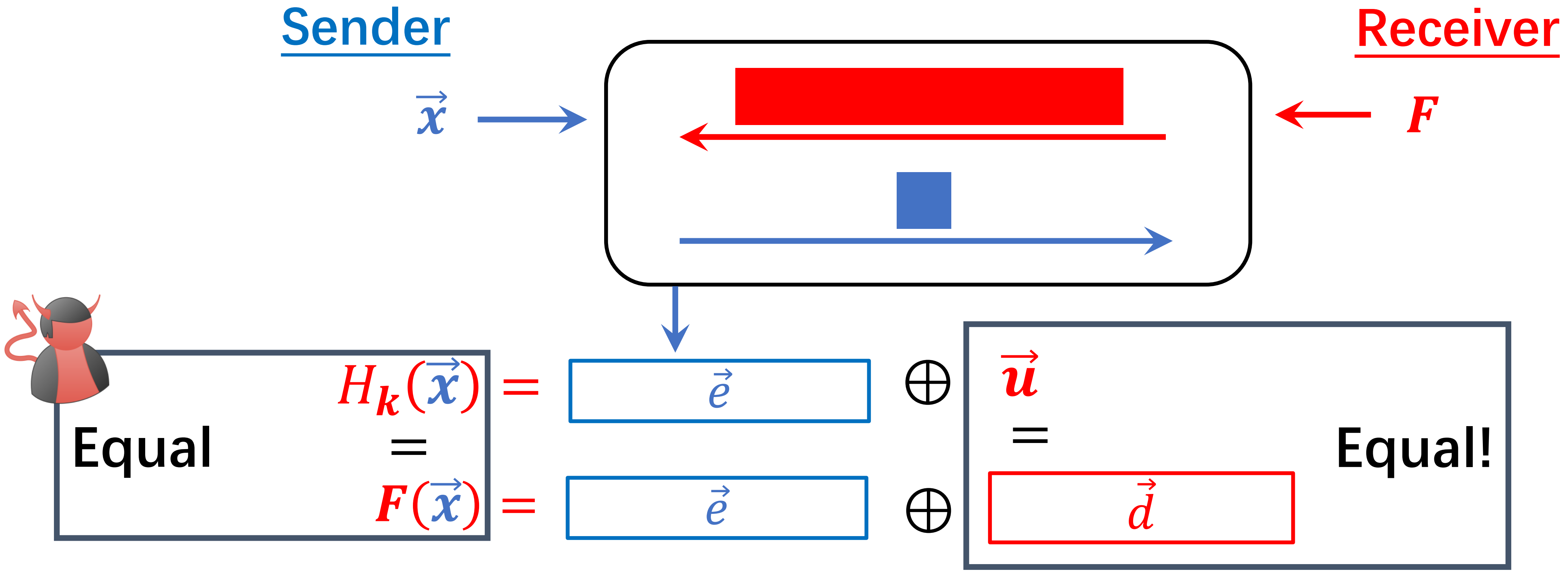
Proof of Correlation Intractability [BKM20]



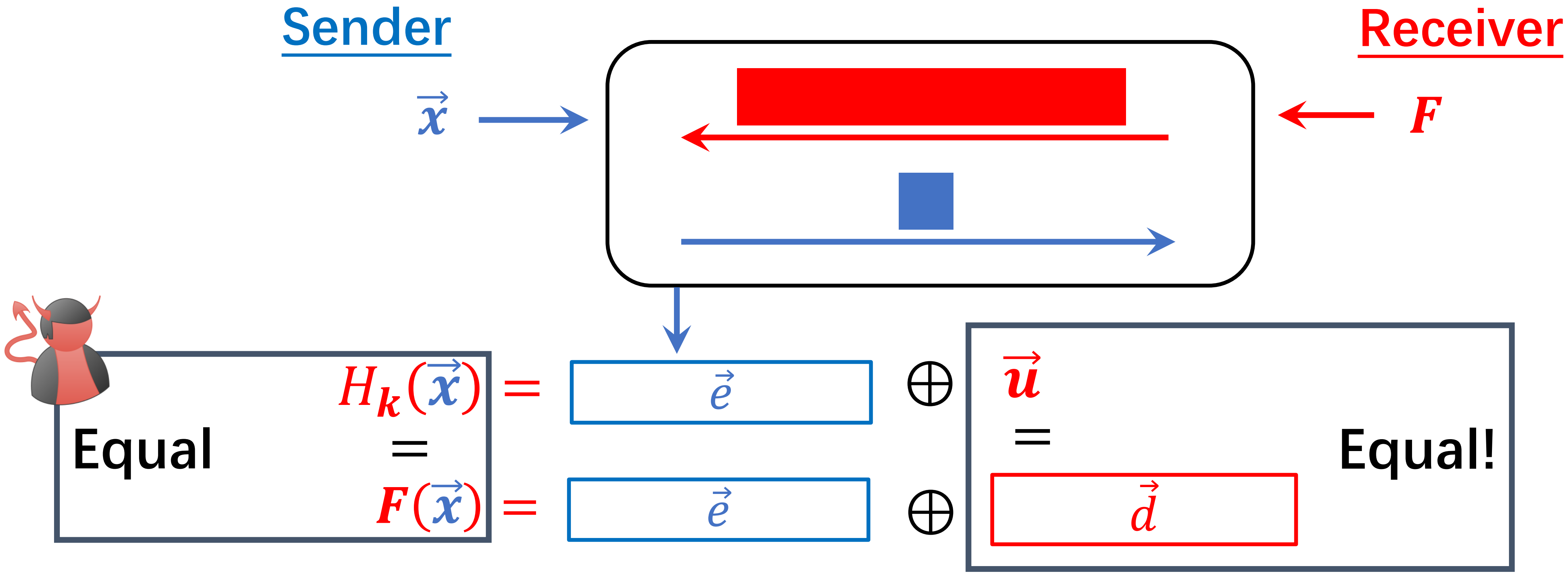
Proof of Correlation Intractability [BKM20]



Proof of Correlation Intractability [BKM20]



Proof of Correlation Intractability [BKM20]

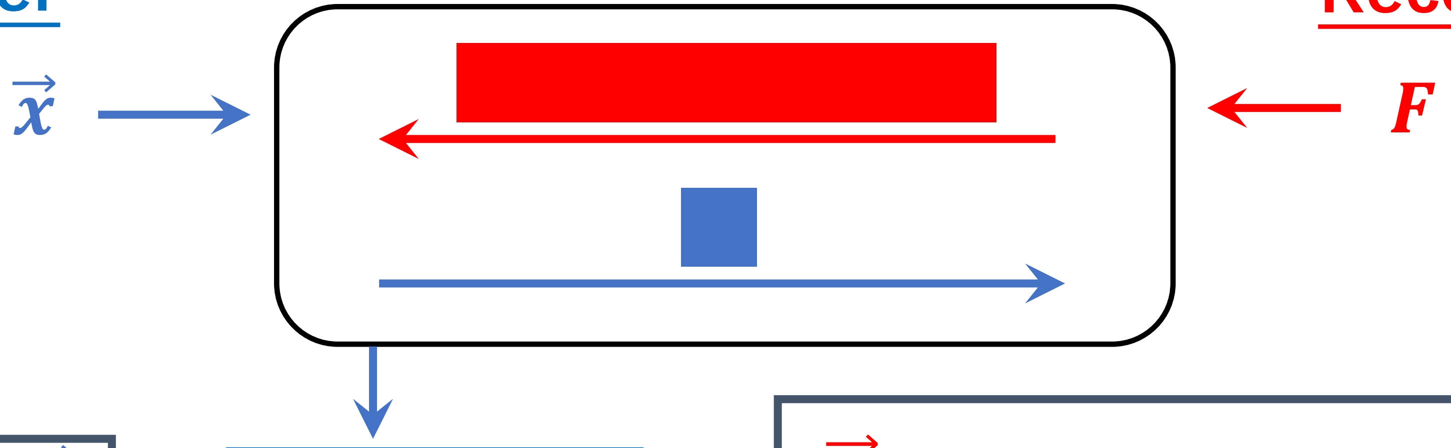


$\boxed{\vec{d}}$ is "sparse",

Since it only depends on

Proof of Correlation Intractability [BKM20]

Sender **Receiver**



$H_k(\vec{x}) = \boxed{\vec{e}} \oplus \vec{u}$

Equal $=$ $=$ **Equal!**

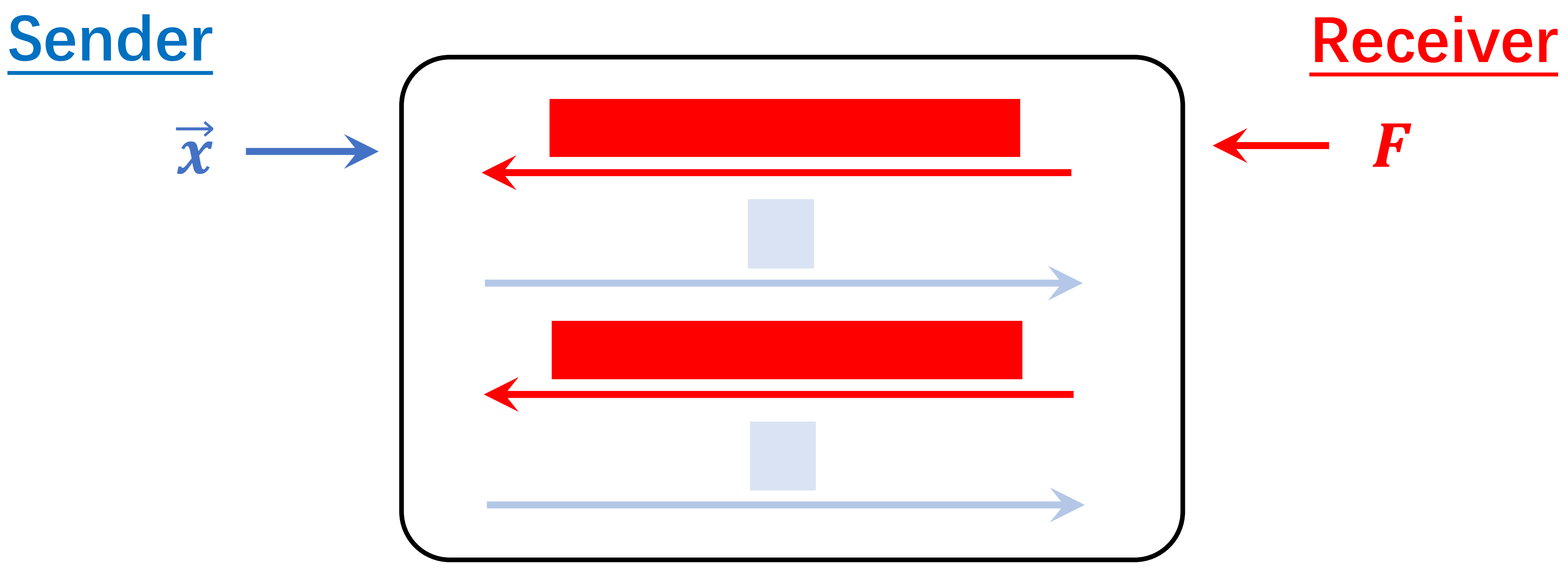
$F(\vec{x}) = \boxed{\vec{e}} \oplus \boxed{\vec{d}}$

$\boxed{\vec{d}}$ is "sparse",

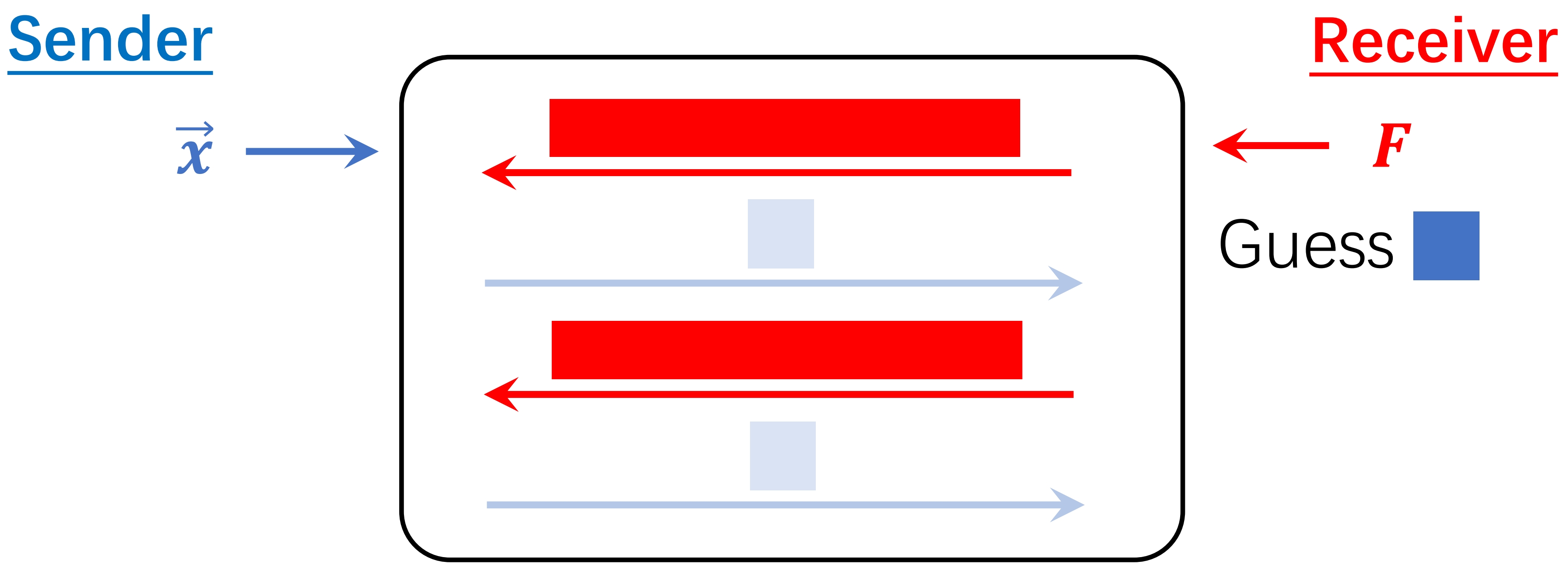
Since it only depends on ■

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x} : \vec{d} = \vec{u}] = \mathbf{negl}$$

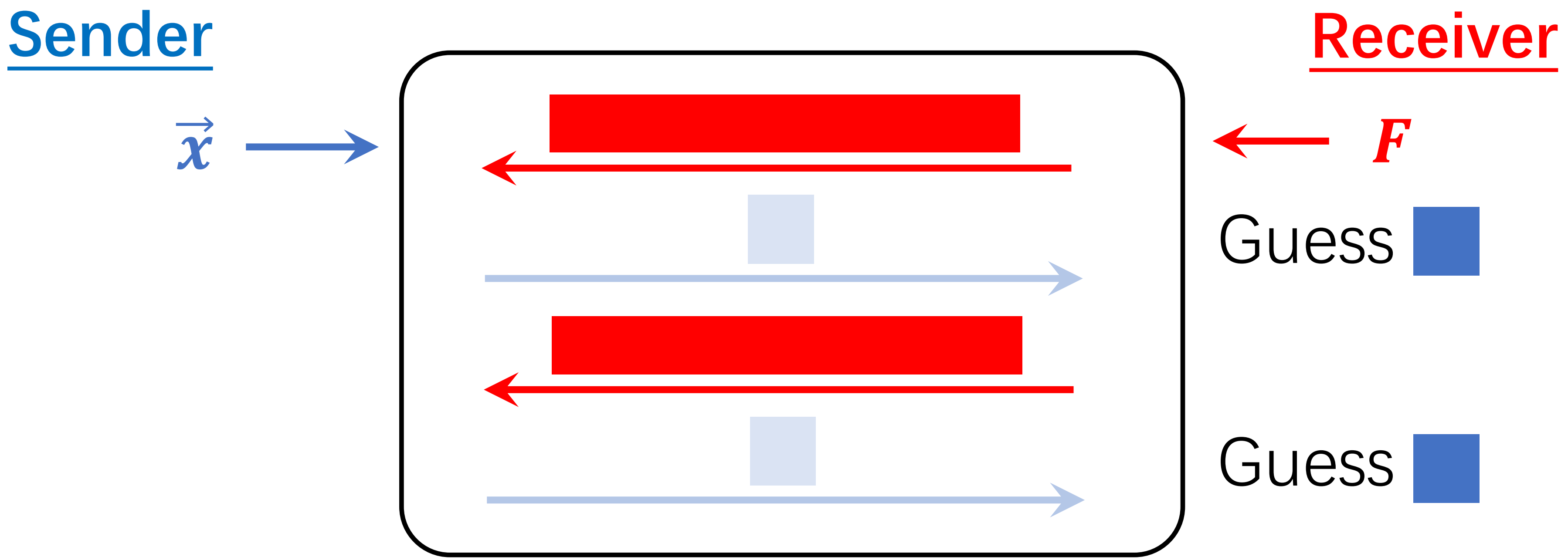
Proof of Correlation Intractability [This work]



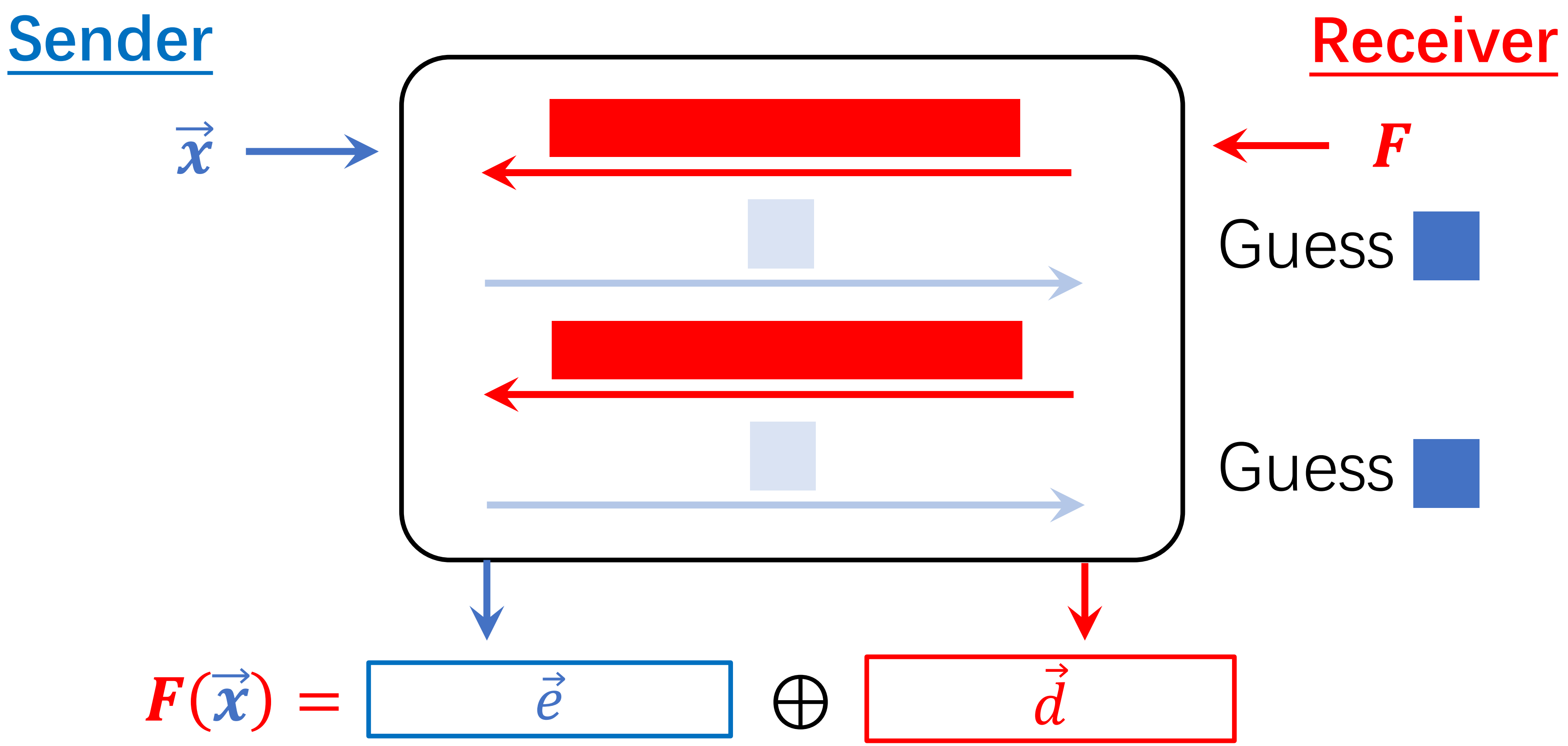
Proof of Correlation Intractability [This work]



Proof of Correlation Intractability [This work]

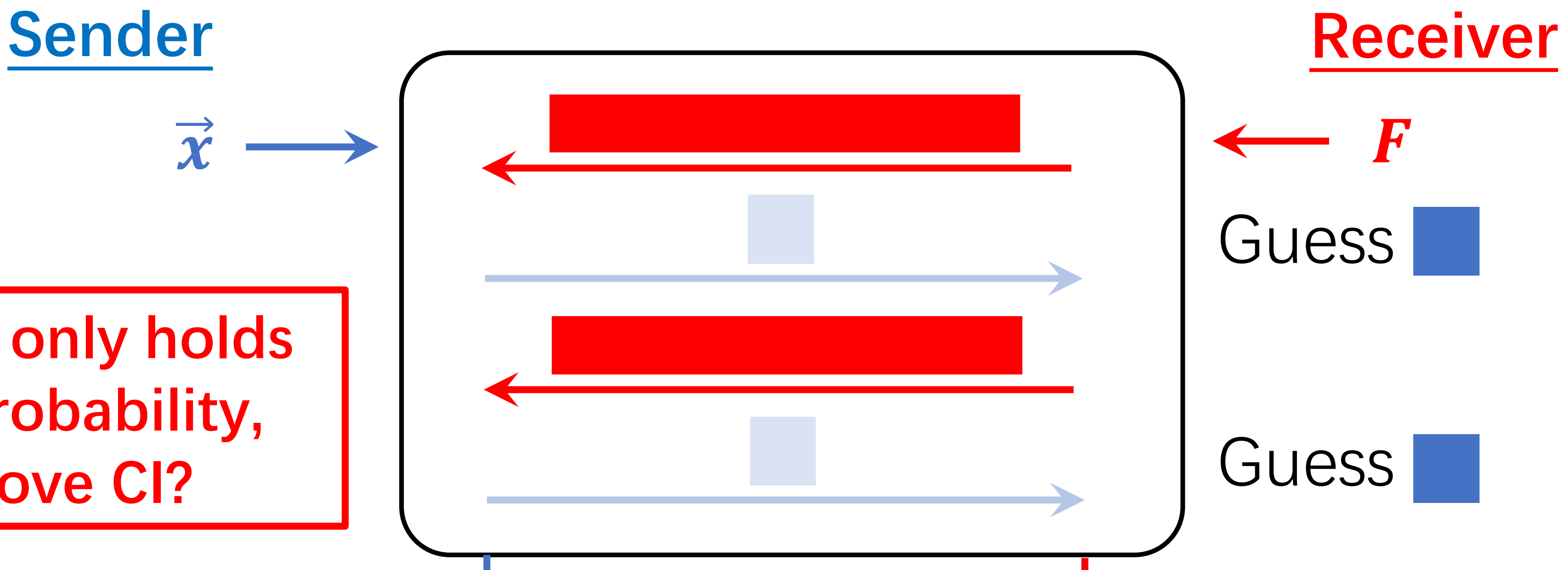


Proof of Correlation Intractability [This work]



Additive reconstruction correctness only holds with Pr[Guessing Correct]

Proof of Correlation Intractability [This work]



If correctness only holds with small probability, how to prove CI?

$$F(\vec{x}) = \boxed{\vec{e}} \oplus \boxed{\vec{d}}$$

Additive reconstruction correctness only holds with Pr[Guessing Correct]

An Oversimplified Case: Guessing is independent of $\vec{x}$

Equal

$$F(\vec{x}) = \vec{e} \oplus \vec{d}$$

$$H_k(\vec{x}) = \vec{e} \oplus \vec{u}$$

Equal!

An Oversimplified Case: Guessing is independent of $\vec{\boldsymbol{x}}$

$\forall \vec{\boldsymbol{x}} \leftarrow$ Pr[Guessing ■ Correct]$= 2^{-O(\lambda)}$,

Equal

$\boldsymbol{F}(\vec{\boldsymbol{x}}) = \boxed{\vec{e}} \oplus \boxed{\vec{d}}$

$H_{\boldsymbol{k}}(\vec{\boldsymbol{x}}) = \boxed{\vec{e}} \oplus \vec{\boldsymbol{u}}$

Equal!

An Oversimplified Case: Guessing is independent of $\vec{x}$

$\forall \vec{x} \leftarrow$ 

Pr[Guessing ■ Correct]$= 2^{-O(\lambda)}$,

Equal

$\boldsymbol{F}(\vec{x}) = \boxed{\vec{e}} \oplus \boxed{\vec{d}}$

$H_{\boldsymbol{k}}(\vec{x}) = \boxed{\vec{e}} \oplus \vec{\boldsymbol{u}}$

Equal!

$$\Pr_{\vec{\boldsymbol{u}} \leftarrow \{0,1\}^n}[\exists \vec{x} : \vec{d} = \vec{\boldsymbol{u}}] \geq 2^{-\boldsymbol{O}(\lambda)}$$

(Not too small)

An Oversimplified Case: Guessing is independent of $\vec{x}$

$\forall \vec{x} \leftarrow$ 

$\Pr[\text{Guessing} \;\blacksquare\; \text{Correct}] = 2^{-O(\lambda)}$,

Equal

$F(\vec{x}) = \boxed{\vec{e}} \oplus \boxed{\vec{d}}$

$H_k(\vec{x}) = \boxed{\vec{e}} \oplus \vec{u}$

Equal!

Sparsity of $\vec{d}$:

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x}: \vec{d} = \vec{u}] \le 2^{-\Omega(n)}$$

(Very small!)

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x}: \vec{d} = \vec{u}] \ge 2^{-O(\lambda)}$$

(Not too small)

An Oversimplified Case: Guessing is independent of $\vec{x}$

$\forall \vec{x} \leftarrow$ 

Pr[Guessing ■ Correct]$= 2^{-O(\lambda)}$,

Equal

$F(\vec{x}) = \vec{e} \oplus \vec{d}$

$H_k(\vec{x}) = \vec{e} \oplus \vec{u}$

Equal!

Sparsity of $\vec{d}$:

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x}: \vec{d} = \vec{u}] \leq 2^{-\Omega(n)}$$

(Very small!)

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x}: \vec{d} = \vec{u}] \geq 2^{-O(\lambda)}$$

(Not too small)

If $n \gg \lambda$, contradiction!

An Oversimplified Case: Guessing is independent of $\vec{\boldsymbol{x}}$

$\forall \vec{x} \leftarrow$ 

Pr[Guessing ■ Correct]$= 2^{-O(\lambda)}$,

Equal

$\boldsymbol{F}(\vec{\boldsymbol{x}}) = \boxed{\vec{e}} \oplus \boxed{\vec{d}}$

$H_{\boldsymbol{k}}(\vec{\boldsymbol{x}}) = \boxed{\vec{e}} \oplus \vec{\boldsymbol{u}}$

Equal!

Sparsity of $\vec{d}$:

$$\Pr_{\vec{\boldsymbol{u}} \leftarrow \{\mathbf{0,1}\}^{\boldsymbol{n}}}[\exists \vec{x}: \vec{d} = \vec{\boldsymbol{u}}] \leq 2^{-\boldsymbol{\Omega}(\boldsymbol{n})}$$

(Very small!)

$\ll$

$$\Pr_{\vec{\boldsymbol{u}} \leftarrow \{\mathbf{0,1}\}^{\boldsymbol{n}}}[\exists \vec{x}: \vec{d} = \vec{\boldsymbol{u}}] \geq 2^{-\boldsymbol{O}(\boldsymbol{\lambda})}$$

(Not too small)

If $n \gg \lambda$, contradiction!

Is Guessing independent of $\vec{\boldsymbol{x}}$?

Is Guessing independent of $\vec{x}$?

$k \leftarrow \mathbf{Gen}(1^\lambda)$

$\longleftarrow$

$\vec{x}$

$\longrightarrow$

$$\Pr_{,k}[H_k(\vec{x}) = F(\vec{x})] \le \text{negl}$$

Is Guessing independent of $\vec{x}$?

ITDH

CIH key: $k = \left(\vec{u}, \quad \cdots \quad \right)$

F

Guess

...

Guess

$k \leftarrow \mathbf{Gen}(1^\lambda)$

$\vec{x}$

$\Pr_{,k}[H_k(\vec{x}) = F(\vec{x})] \leq \mathrm{negl}$

Is Guessing independent of $\vec{x}$?

ITDH

CIH key: $k = \big(\vec{u},$ … $\big)$ $\leftarrow F$

Guess

…

Guess

$k \leftarrow \mathbf{Gen}(1^\lambda)$

$\vec{x}$

 chooses $\vec{x}$ depending on k, which depends on the guessing

Is Guessing independent of $\vec{x}$?

ITDH

CIH key: $k = \big(\vec{u},$ … $\big)$ ← F

Guess

…

Guess

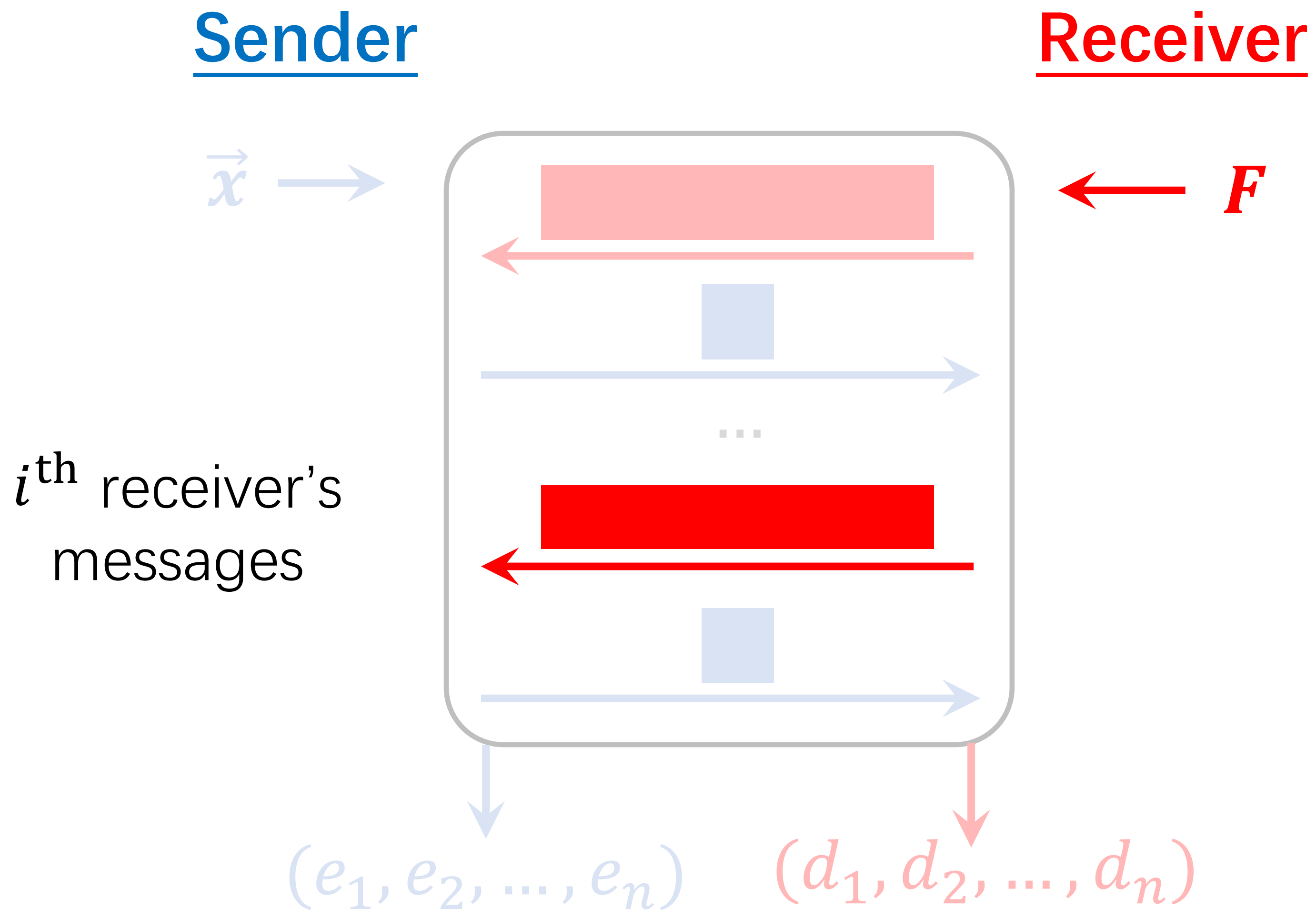
$k \leftarrow \mathbf{Gen}(1^\lambda)$

$\vec{x}$

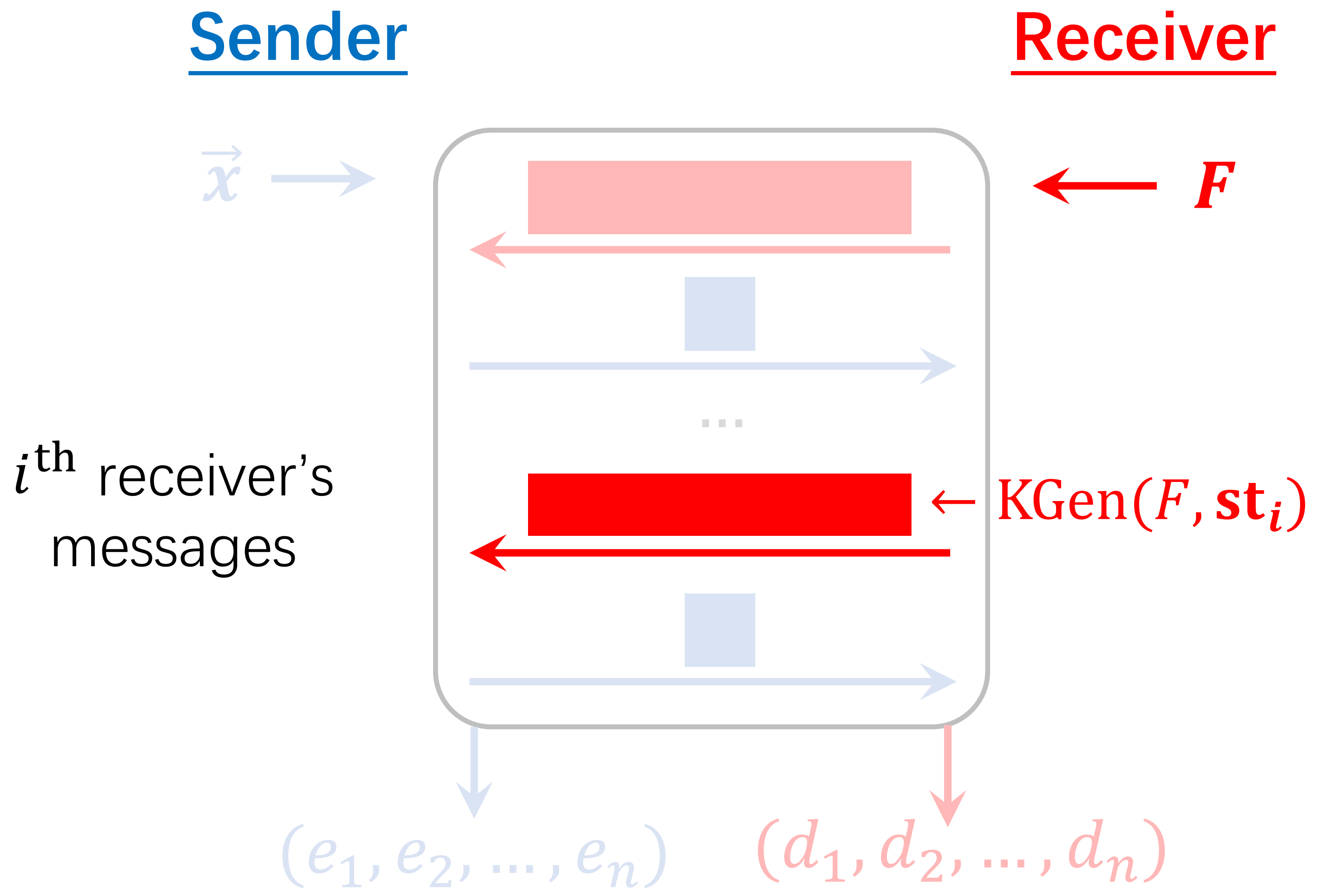
chooses $\vec{x}$ depending on k, which depends on the guessing

Function Hiding: also hides

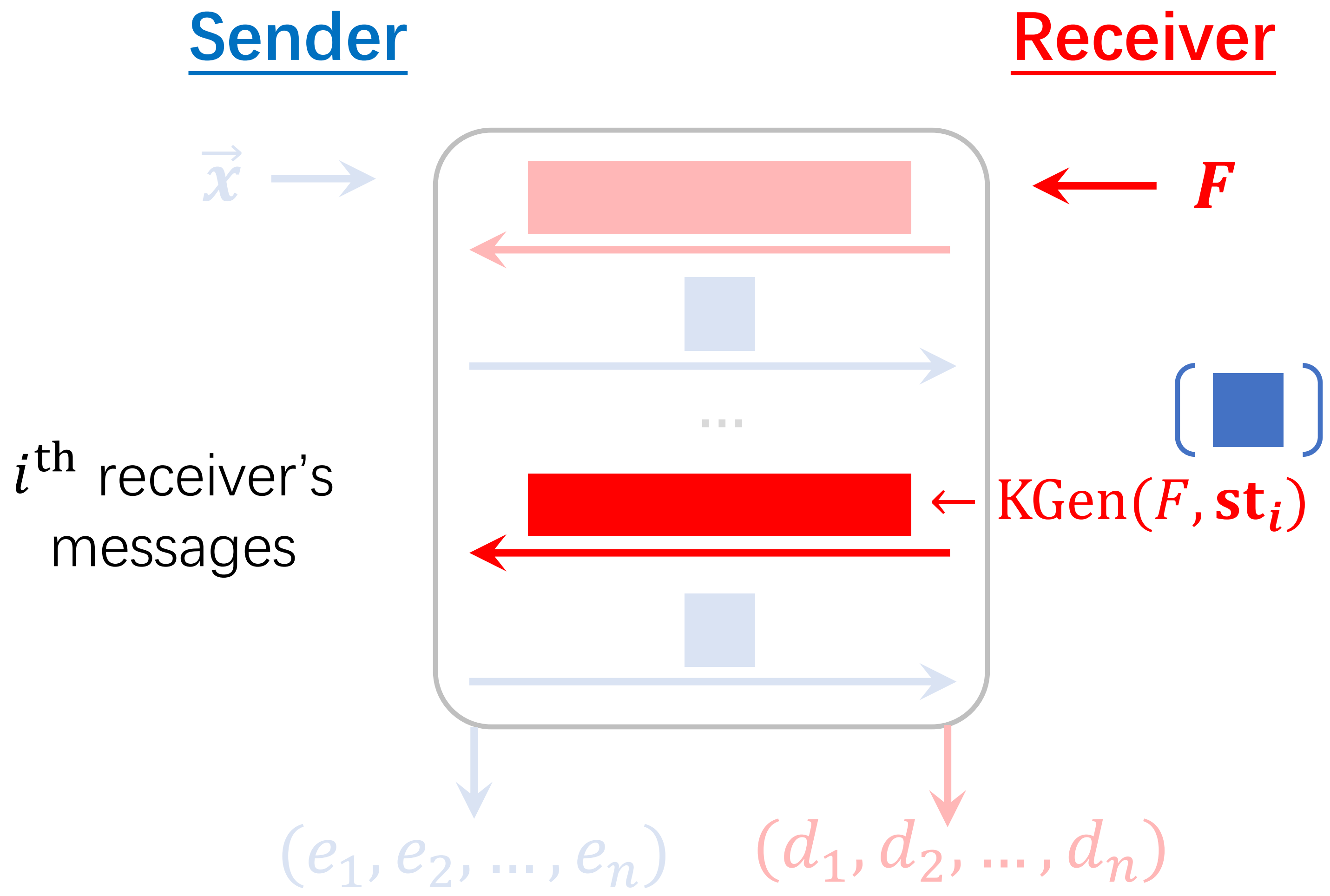
Function Hiding in Detail



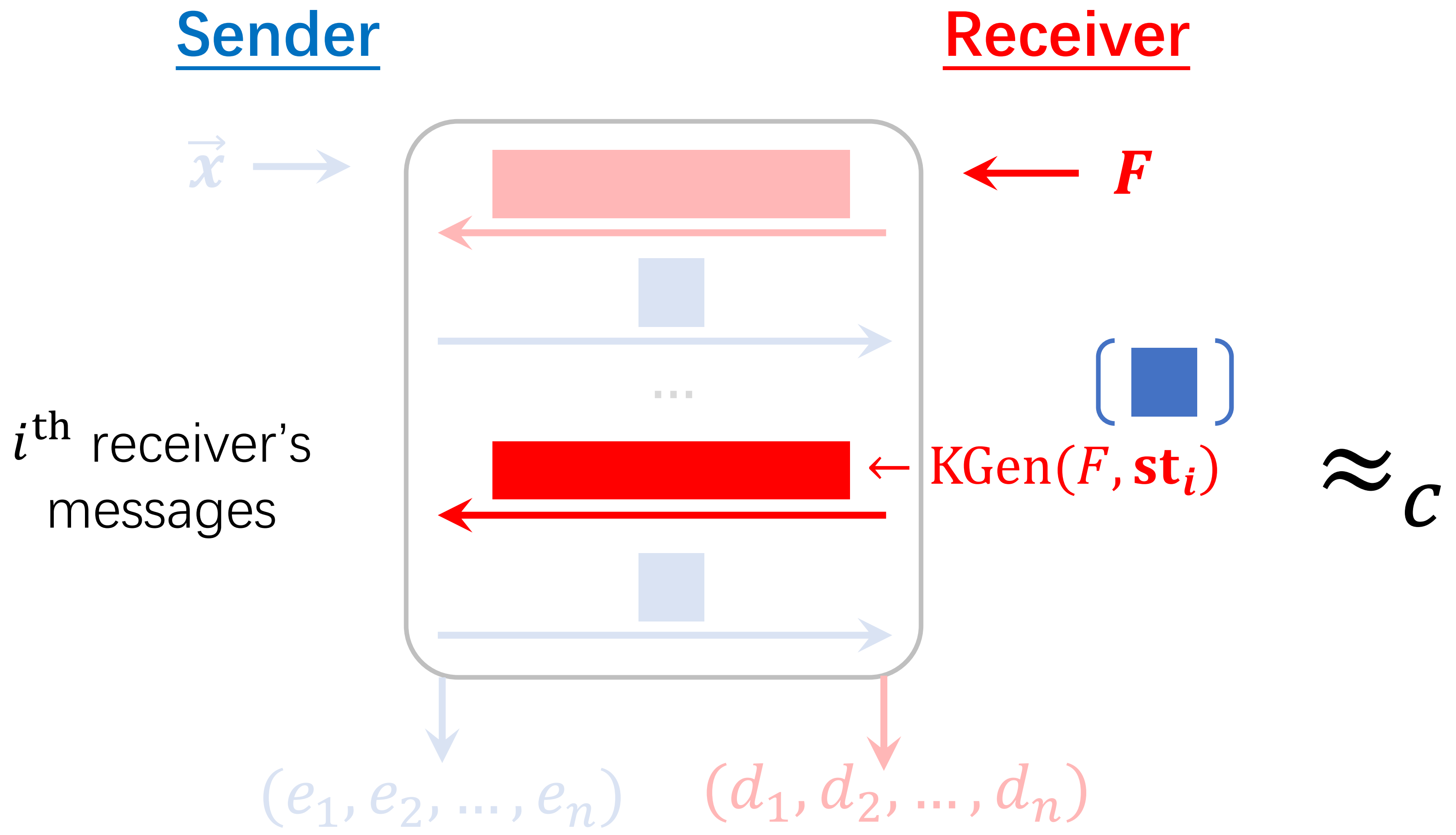
Function Hiding in Detail



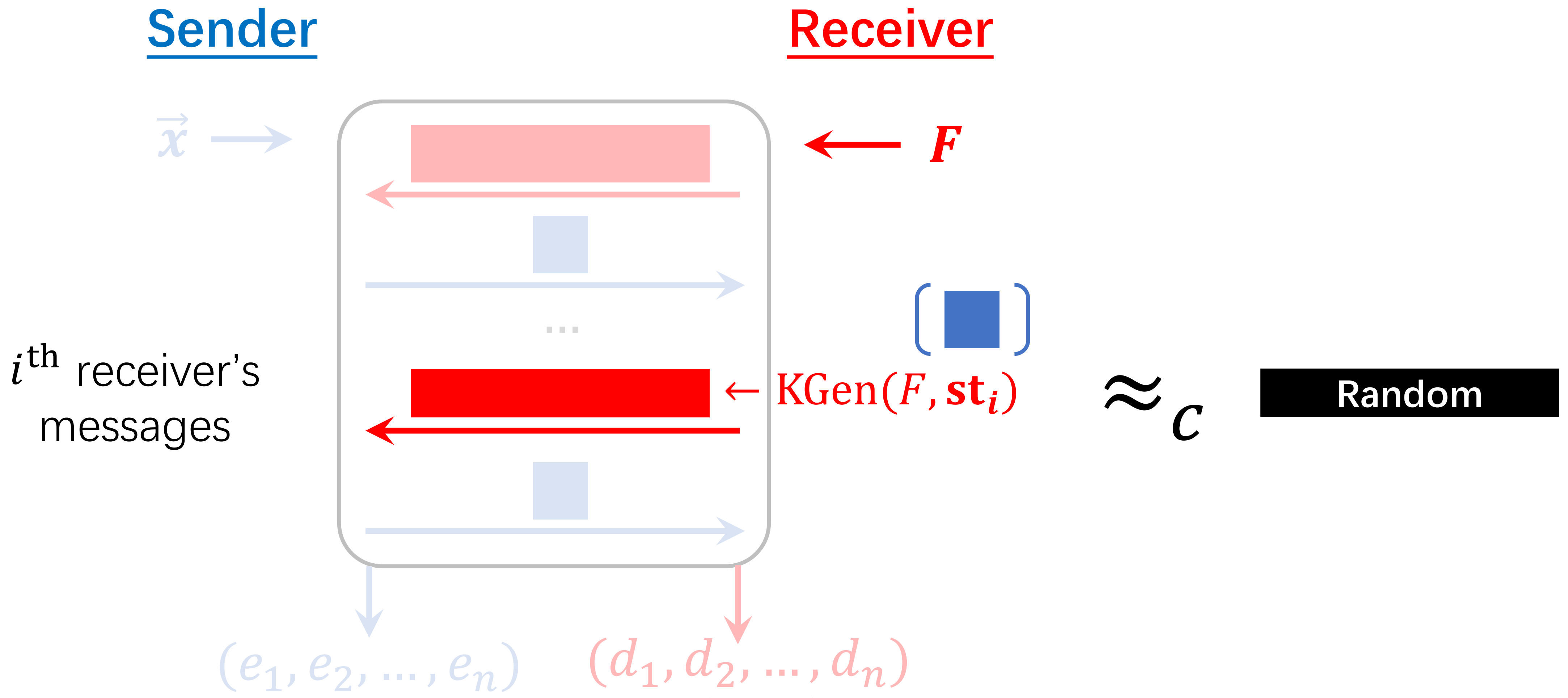
Function Hiding in Detail



Function Hiding in Detail

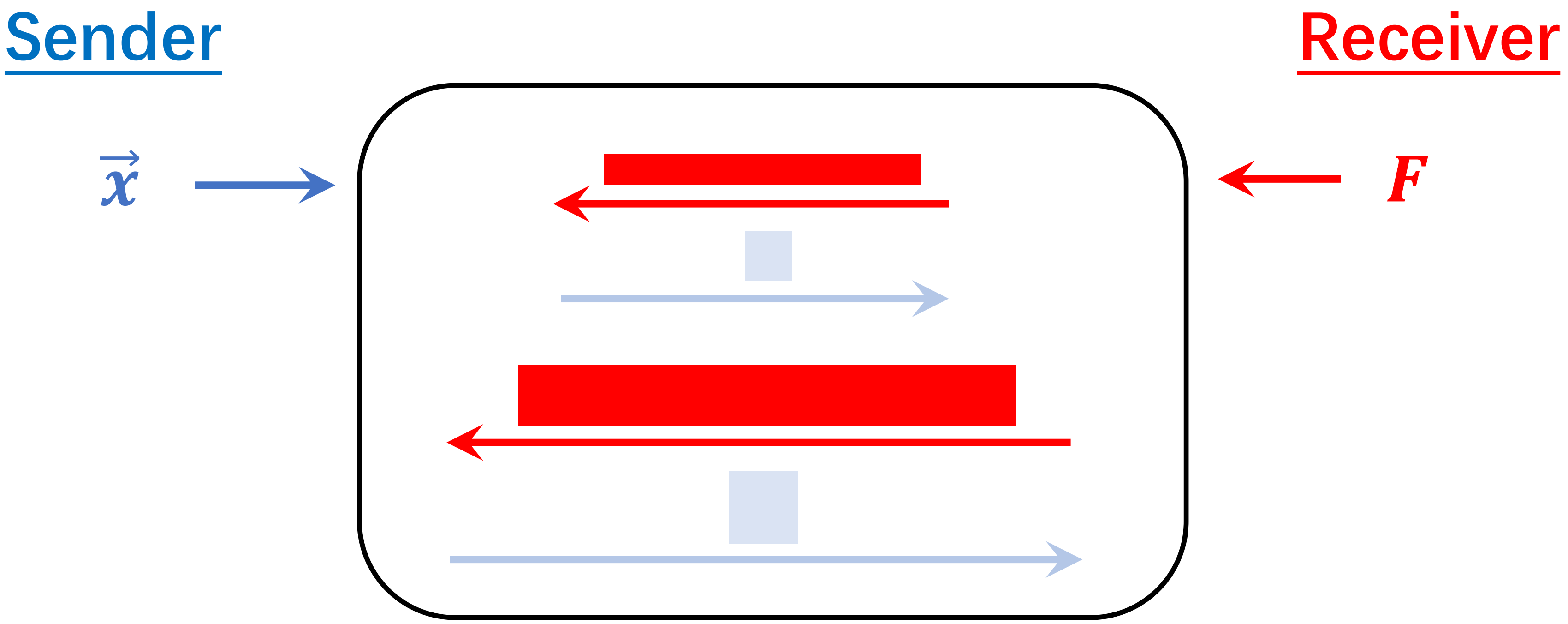


Function Hiding in Detail

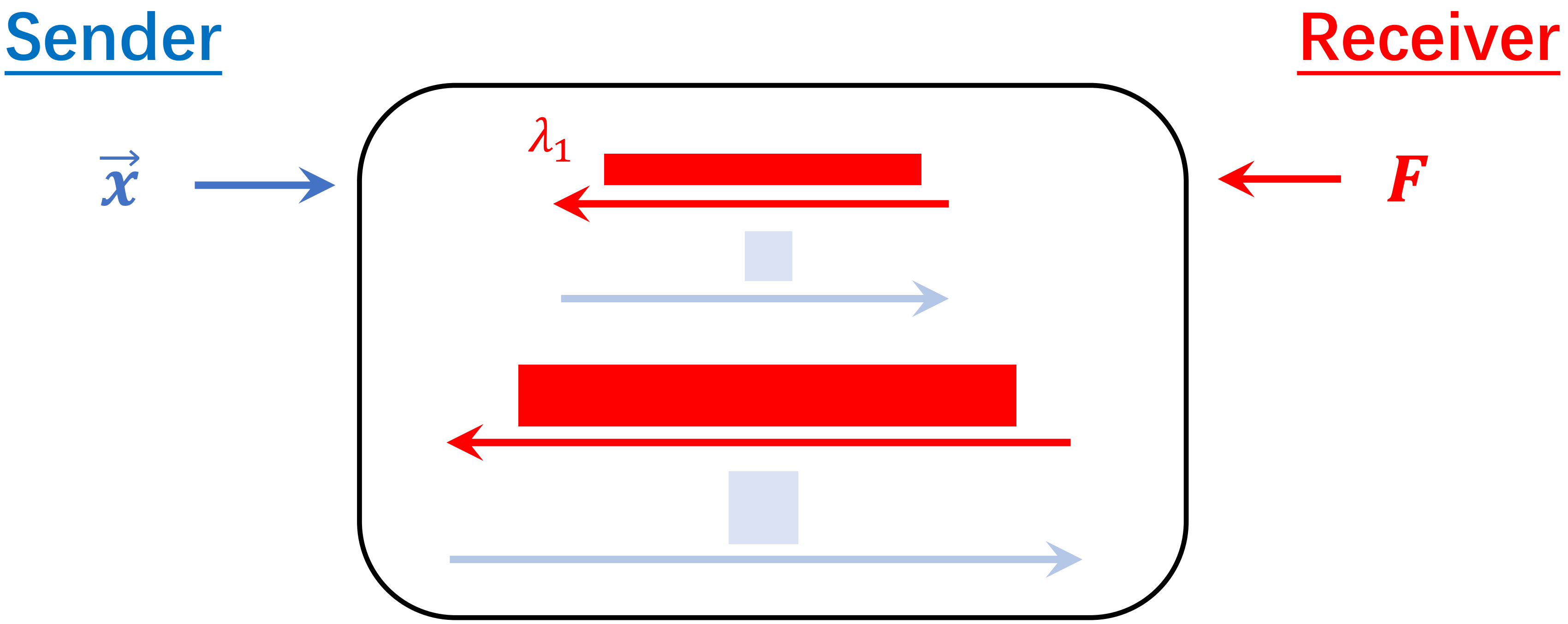


- **Function Hiding:** $\forall\ F, \mathbf{st}_i,\ \mathrm{KGen}(F, \mathrm{st}_i) \approx_c$ **Uniformly Random String**

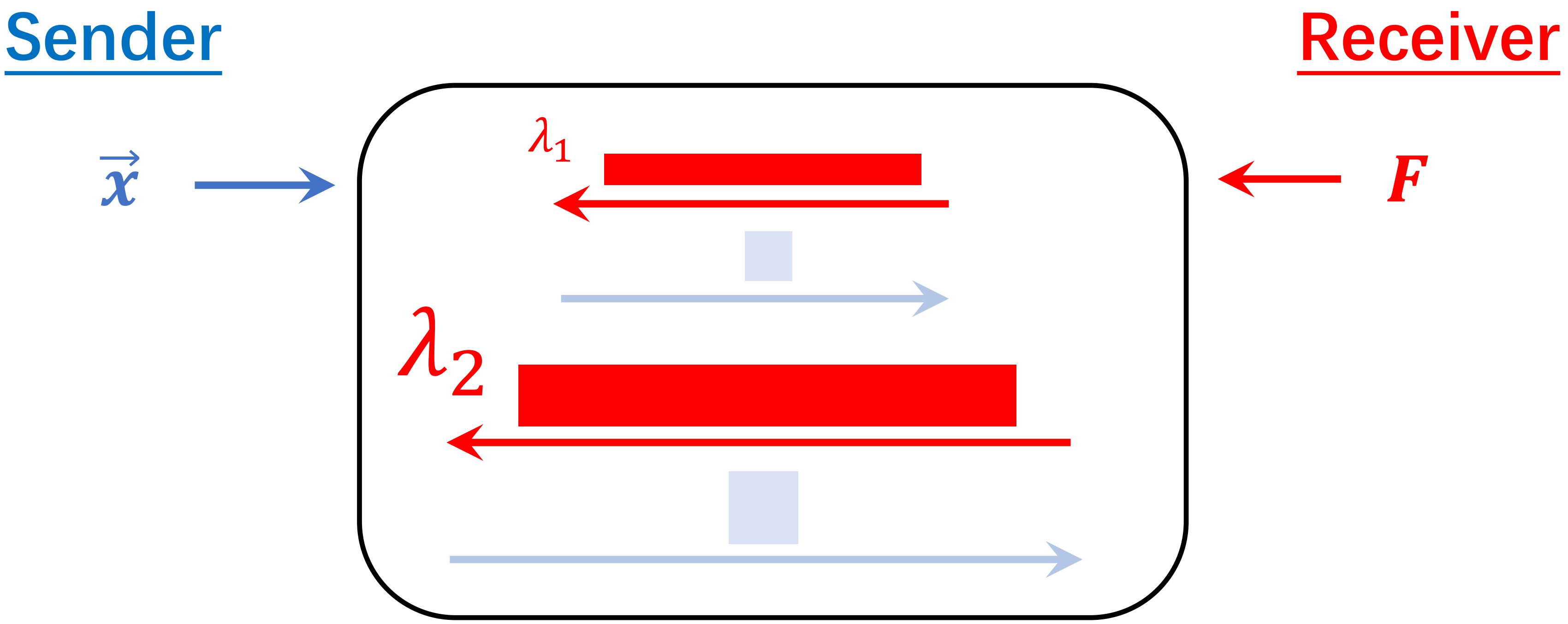
Leverage Function Hiding



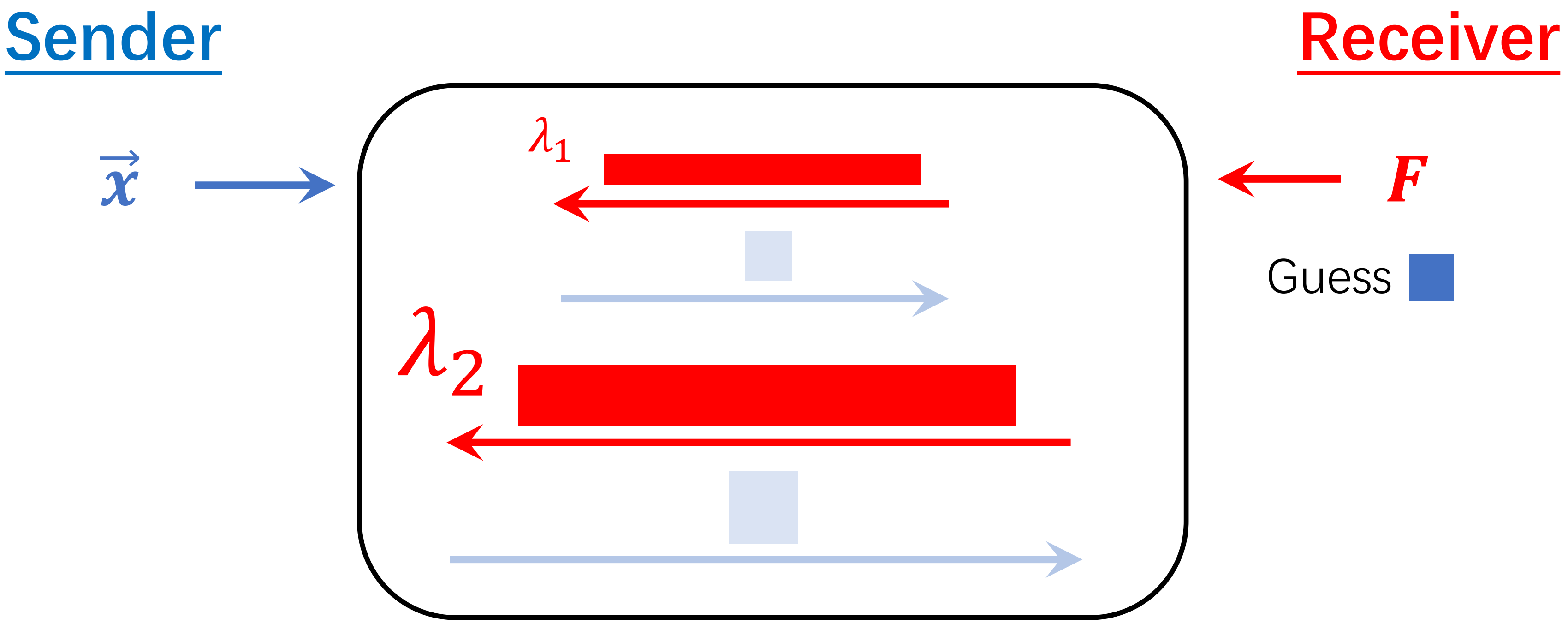
Leverage Function Hiding



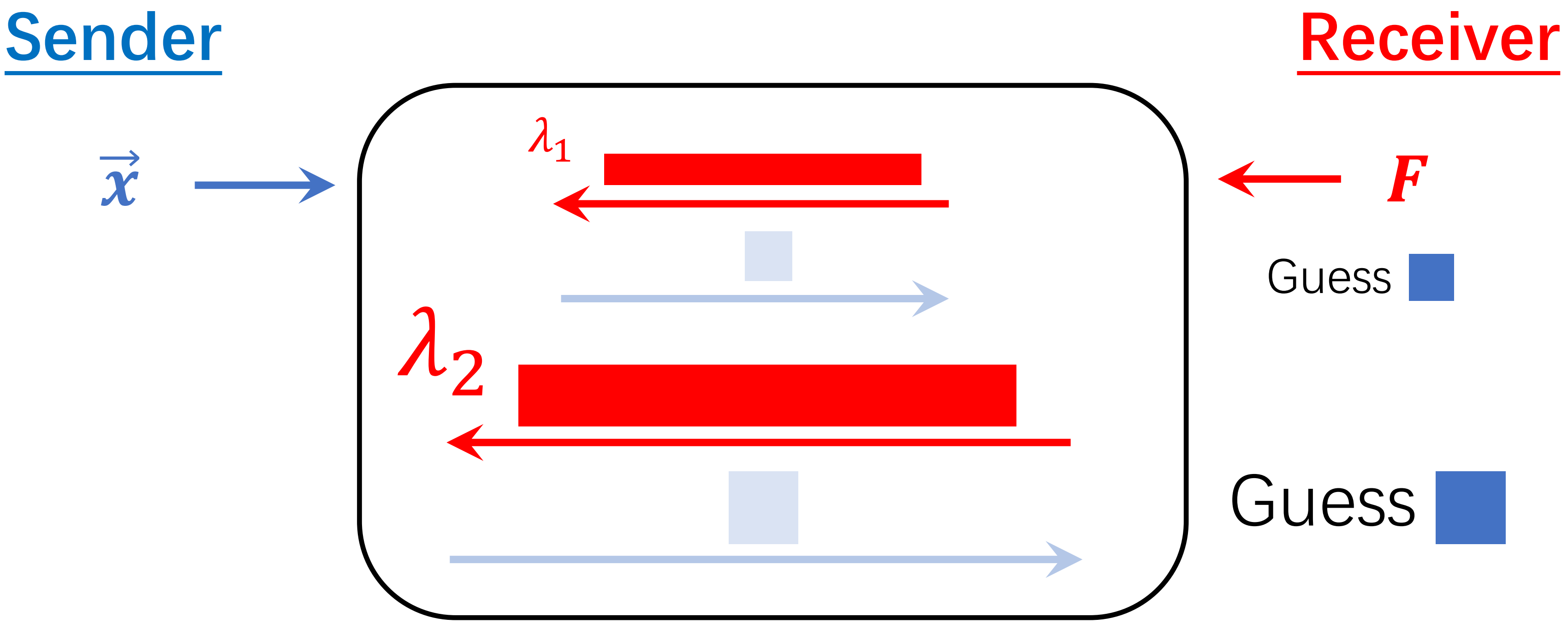
Leverage Function Hiding



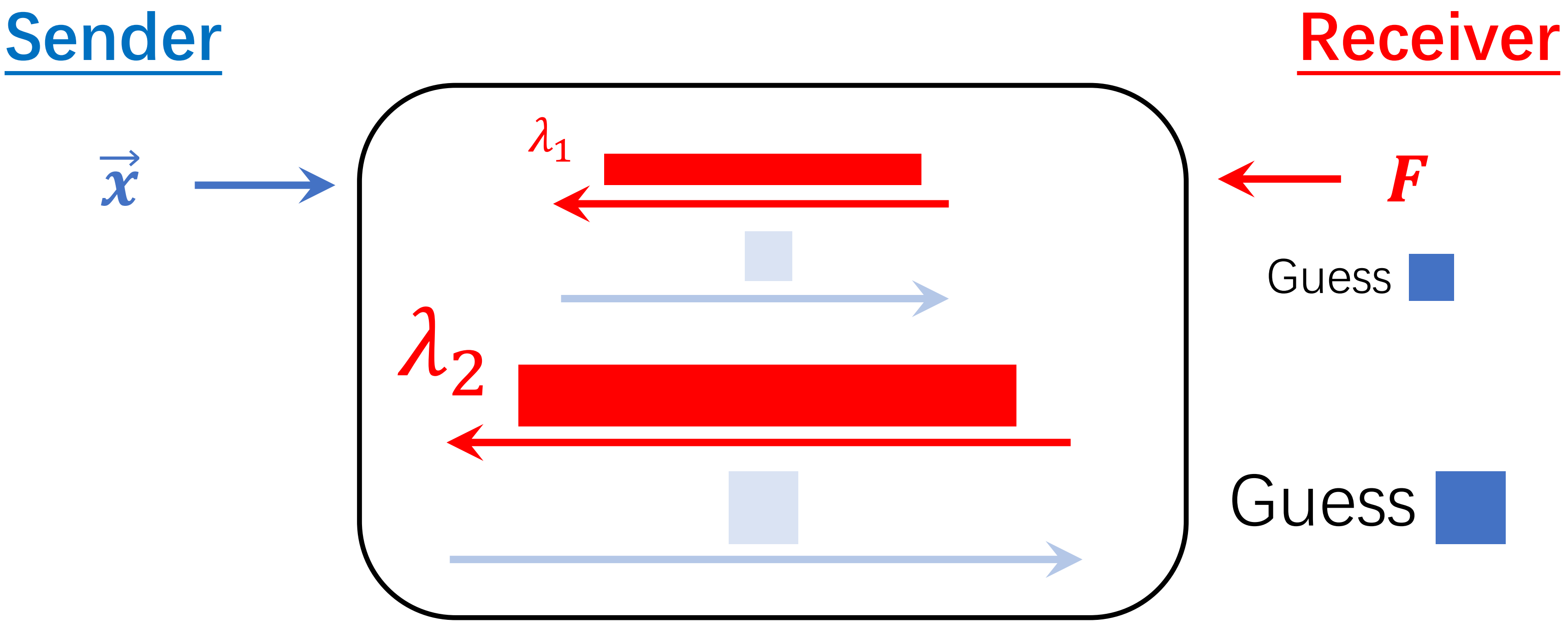
Leverage Function Hiding



Leverage Function Hiding



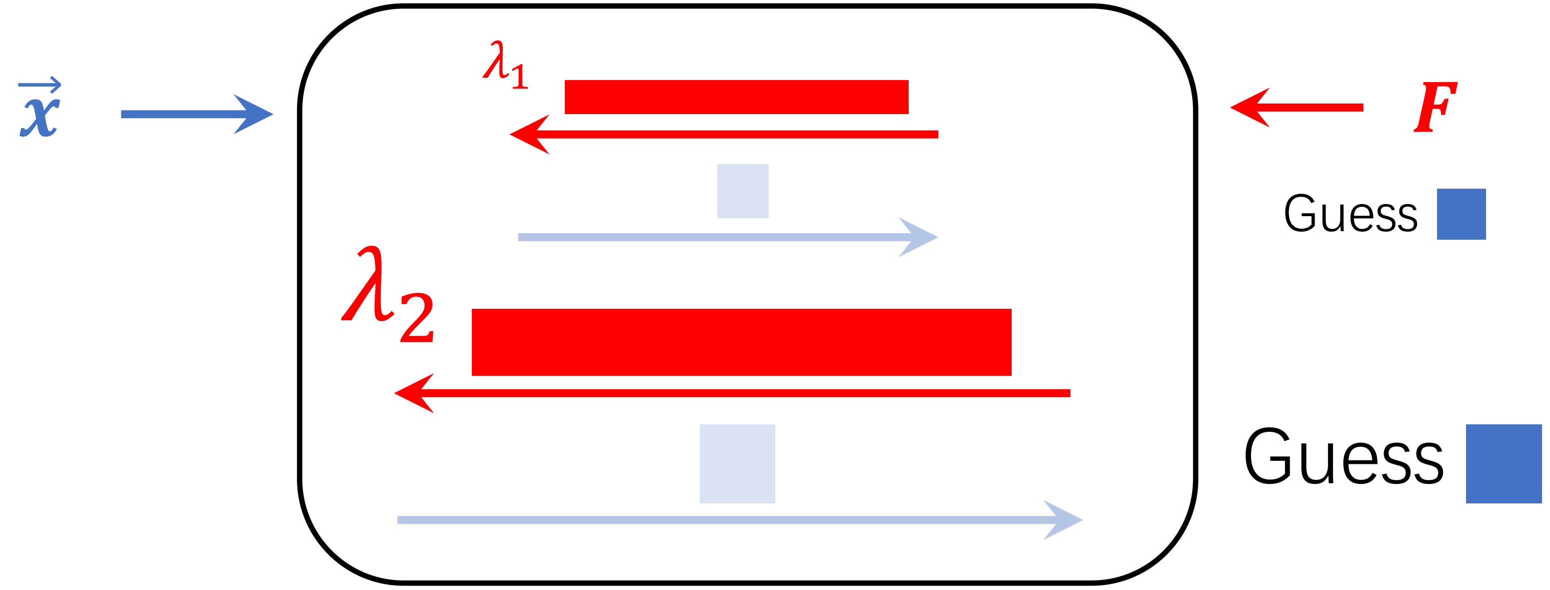
Leverage Function Hiding



- Guess ■ correctly with Pr. $2^{-\lambda_1}$ – $2^{-\lambda_2^c}$ (not too small)

Leverage Function Hiding

Sender **Receiver**

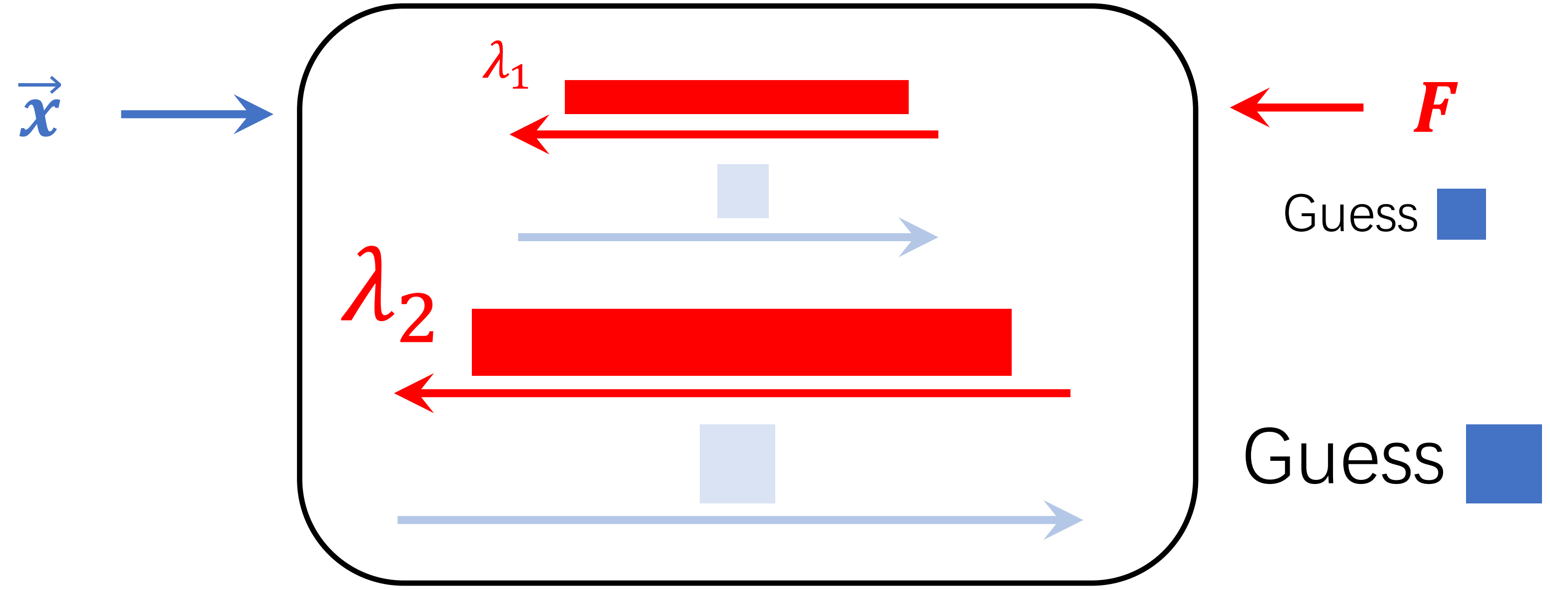


- **Guess ■ correctly with** $\mathbf{Pr.}\ 2^{-\lambda_1} - 2^{-\lambda_2^c}$ **(not too small)**

Uniform Random Guessing

Leverage Function Hiding

Sender **Receiver**



- Guess ■ correctly with Pr. $2^{-\lambda_1} - 2^{-\lambda_2^c}$ (not too small)

Uniform Random Guessing

Sub-exponential Function Hiding

Modified proof of Correlation Intractability

Modified proof of Correlation Intractability

- **Extend to $\boldsymbol{O(1)}$ rounds (or $\boldsymbol{O(\log\log\lambda)}$-rounds):**

Modified proof of Correlation Intractability

- **Extend to $\boldsymbol{O(1)}$ rounds (or $\boldsymbol{O(\log\log\lambda)}$-rounds):**

$$\lambda_1 < \lambda_2 < \lambda_3 \quad \cdots < \lambda_L$$

Modified proof of Correlation Intractability

- **Extend to $\boldsymbol{O}(\mathbf{1})$ rounds (or $\boldsymbol{O}(\mathbf{log\,log}\,\boldsymbol{\lambda})$-rounds):**

$$\lambda_1 < \lambda_2 < \lambda_3 \;\cdots < \lambda_L$$

From Guessing Correctness:

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x} : \vec{d} = \vec{u}] \geq 2^{-O(\lambda_1 + \lambda_2 \ldots + \lambda_L)}$$

(Not too small)

Modified proof of Correlation Intractability

- **Extend to $\boldsymbol{O}(\mathbf{1})$ rounds (or $\boldsymbol{O}(\mathbf{log\,log}\,\boldsymbol{\lambda})$-rounds):**

$$\lambda_1 < \lambda_2 < \lambda_3 \cdots < \lambda_L$$

Sparsity of $\vec{d}$:

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x}: \vec{d} = \vec{u}] \leq 2^{-\Omega(n)}$$

(Very small!)

From Guessing Correctness:

$$\Pr_{\vec{u} \leftarrow \{0,1\}^n}[\exists \vec{x}: \vec{d} = \vec{u}] \geq 2^{-O(\lambda_1 + \lambda_2 \ldots + \lambda_L)}$$

(Not too small)

Modified proof of Correlation Intractability

- **Extend to $\boldsymbol{O}(\mathbf{1})$ rounds (or $\boldsymbol{O}(\log\log\lambda)$-rounds):**

$$\lambda_1 < \lambda_2 < \lambda_3 \quad \cdots < \lambda_L$$

Sparsity of $\vec{d}$:

$$\Pr_{\vec{u}\leftarrow\{0,1\}^n}\left[\exists\vec{x}: \vec{d} = \vec{u}\right] \leq 2^{-\Omega(n)}$$

(Very small!)

From Guessing Correctness:

$$\Pr_{\vec{u}\leftarrow\{0,1\}^n}\left[\exists\vec{x}: \vec{d} = \vec{u}\right] \geq 2^{-O(\lambda_1+\lambda_2\ldots+\lambda_L)}$$

(Not too small)

If $n \gg \lambda$, Correlation Intractable!

Modified proof of Correlation Intractability

- **Extend to $\boldsymbol{O}(\mathbf{1})$ rounds (or $\boldsymbol{O}(\mathbf{log\,log}\,\boldsymbol{\lambda})$-rounds):**

$$\lambda_1 < \lambda_2 < \lambda_3 \quad \cdots < \lambda_L$$

Sparsity of $\vec{d}$:

$$\Pr_{\vec{\boldsymbol{u}} \leftarrow \{\mathbf{0},\mathbf{1}\}^{\boldsymbol{n}}}\left[\exists \vec{\boldsymbol{x}} \colon \vec{d} = \vec{\boldsymbol{u}}\right] \leq 2^{-\boldsymbol{\Omega}(\boldsymbol{n})}$$

(Very small!)

$\ll$

From Guessing Correctness:

$$\Pr_{\vec{\boldsymbol{u}} \leftarrow \{\mathbf{0},\mathbf{1}\}^{\boldsymbol{n}}}\left[\exists \vec{\boldsymbol{x}} \colon \vec{d} = \vec{\boldsymbol{u}}\right] \geq 2^{-\boldsymbol{O}(\boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2 \ldots + \boldsymbol{\lambda}_L)}$$

(Not too small)

If $n \gg \lambda$, Correlation Intractable!

Interactive TDH for $\mathbf{TC^0}$

- Recap of Fiat-Shamir
- Main Challenges
- ITDH for $\mathbf{TC^0}$ → CIH for $\mathbf{TC^0}$
- **Construction of ITDH**

Background: Threshold Gates and $\mathbf{TC^0}$

Background: Threshold Gates and $\mathbf{TC^0}$

- **Threshold Gate** ($\vec{\boldsymbol{x}} \in \{0,1\}^{\boldsymbol{n}}$):

$$\mathrm{Th}^{t}\ (\vec{\boldsymbol{x}}) = \begin{cases} 1, & \mathrm{weight}(\vec{x}) \geq t \\ 0, & \mathrm{Otherwise} \end{cases}$$

Background: Threshold Gates and $\mathbf{TC^0}$

- **Threshold Gate** ($\vec{\boldsymbol{x}} \in \{0,1\}^{\boldsymbol{n}}$):

$$\mathrm{Th}^{t}\,(\vec{\boldsymbol{x}}) = \begin{cases} 1, & \mathrm{weight}(\vec{x}) \geq t \\ 0, & \mathrm{Otherwise} \end{cases}$$

- TC^0: constant depth circuits consists of **{NOT, Threshold}** gates

Background: Threshold Gates and $\mathbf{TC^0}$

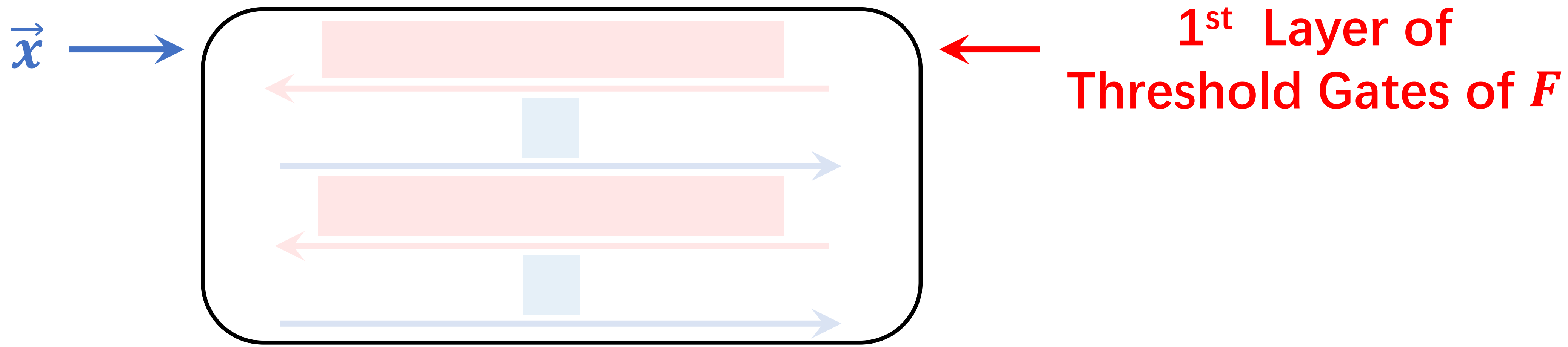
- **Threshold Gate** ($\vec{\boldsymbol{x}} \in \{0,1\}^{\boldsymbol{n}}$):

$$\mathrm{Th}^{t}\ (\vec{\boldsymbol{x}}) = \begin{cases} 1, & \mathrm{weight}(\vec{x}) \geq t \\ 0, & \mathrm{Otherwise} \end{cases}$$

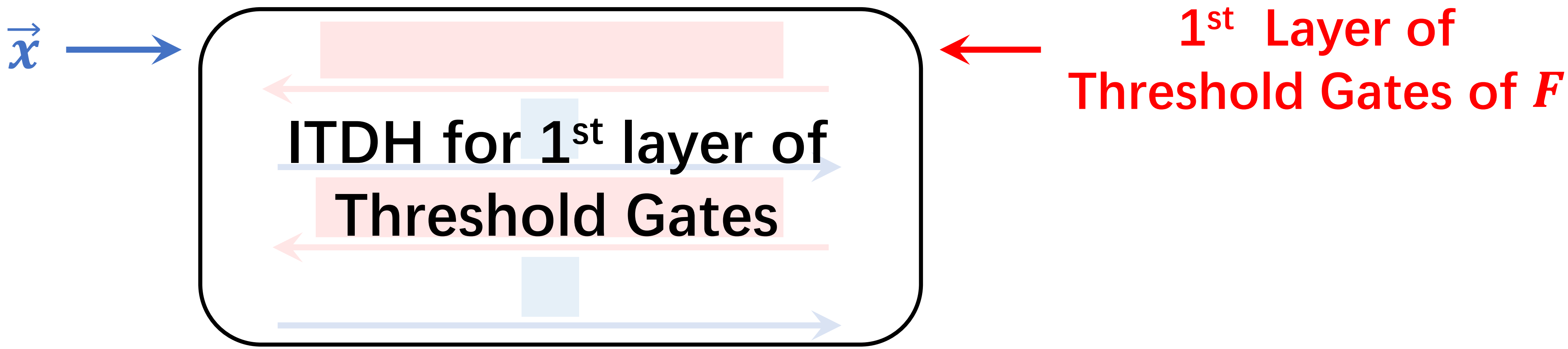
- TC^0: constant depth circuits consists of **{NOT, Threshold}** gates
- For simplicity, let's only consider the threshold gates.

ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation

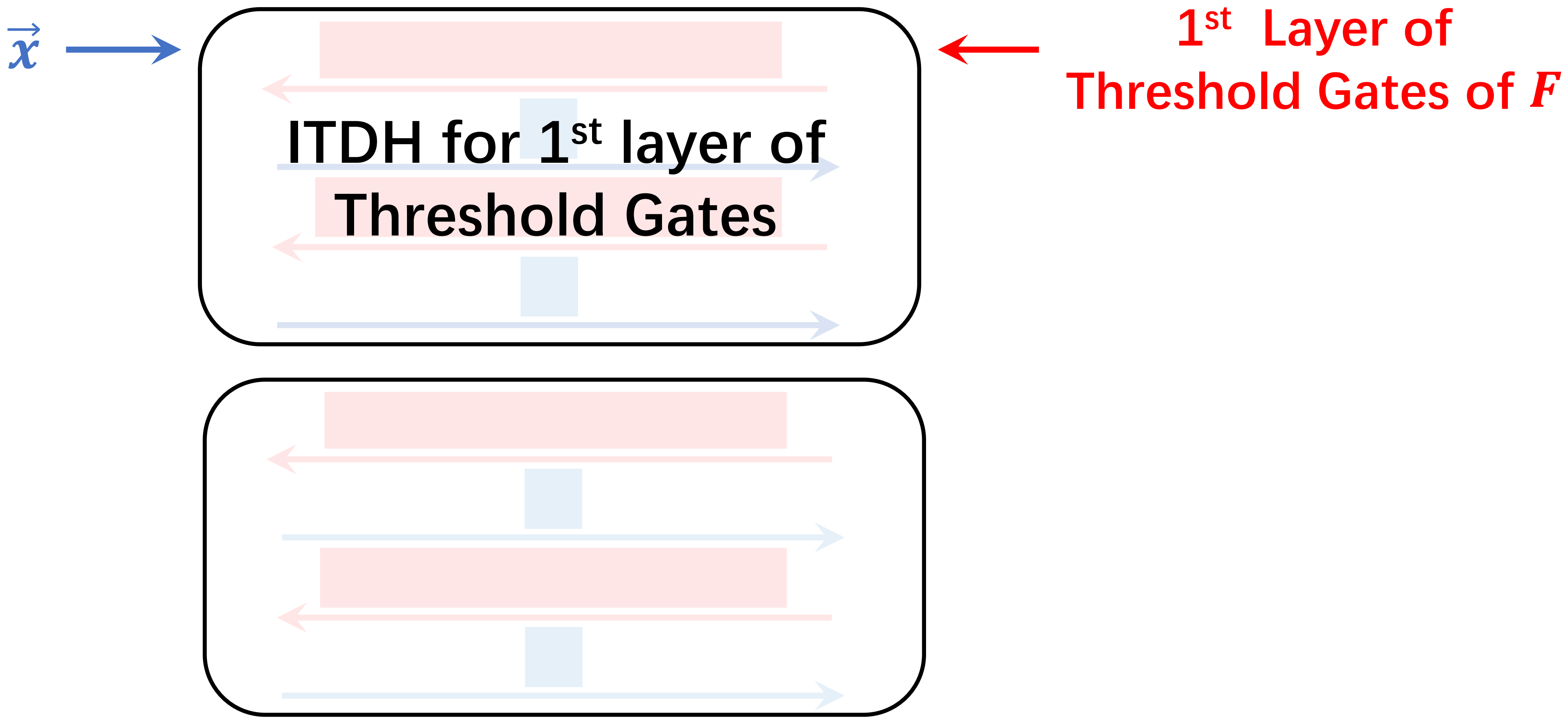
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



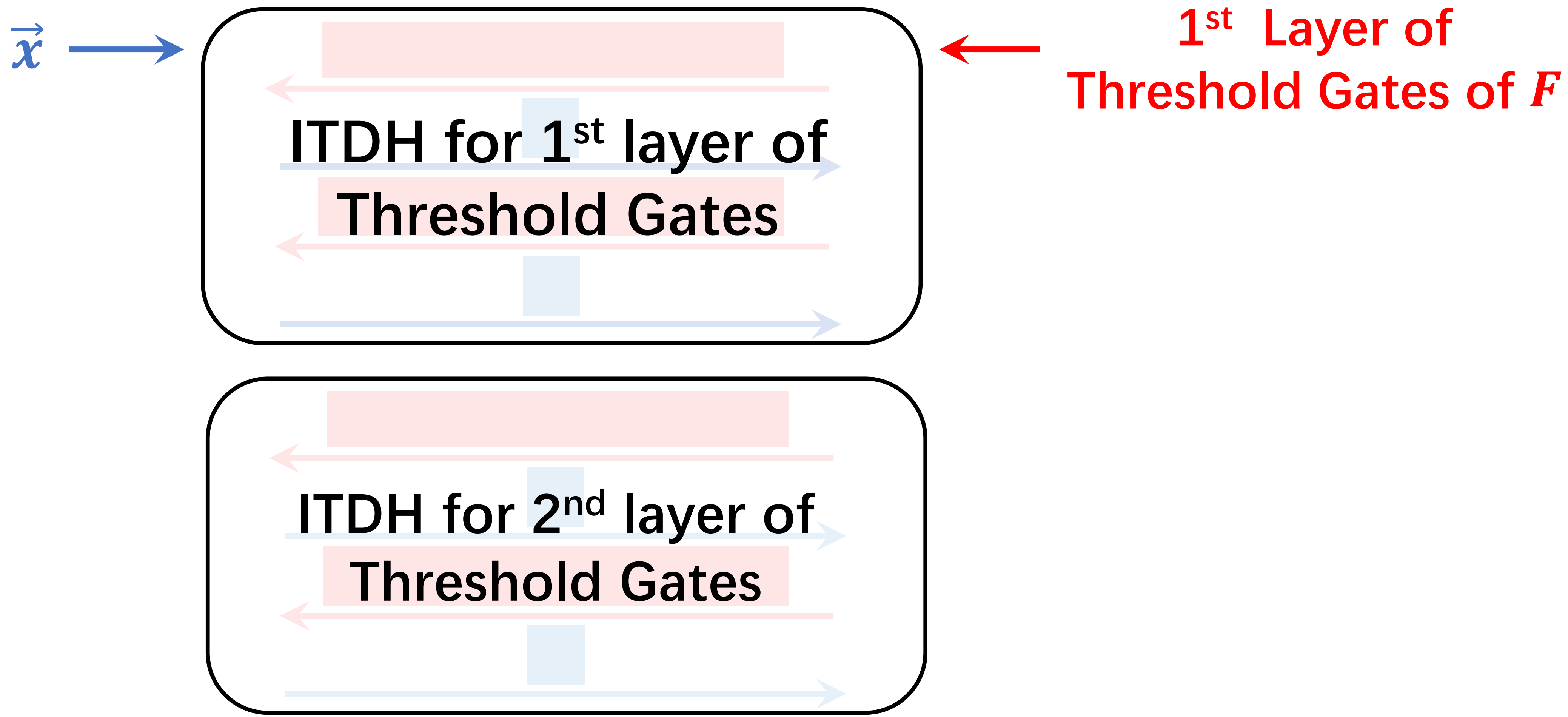
ITDH for $\mathbf{TC}^{\mathbf{0}}$: Layer-by-Layer Computation



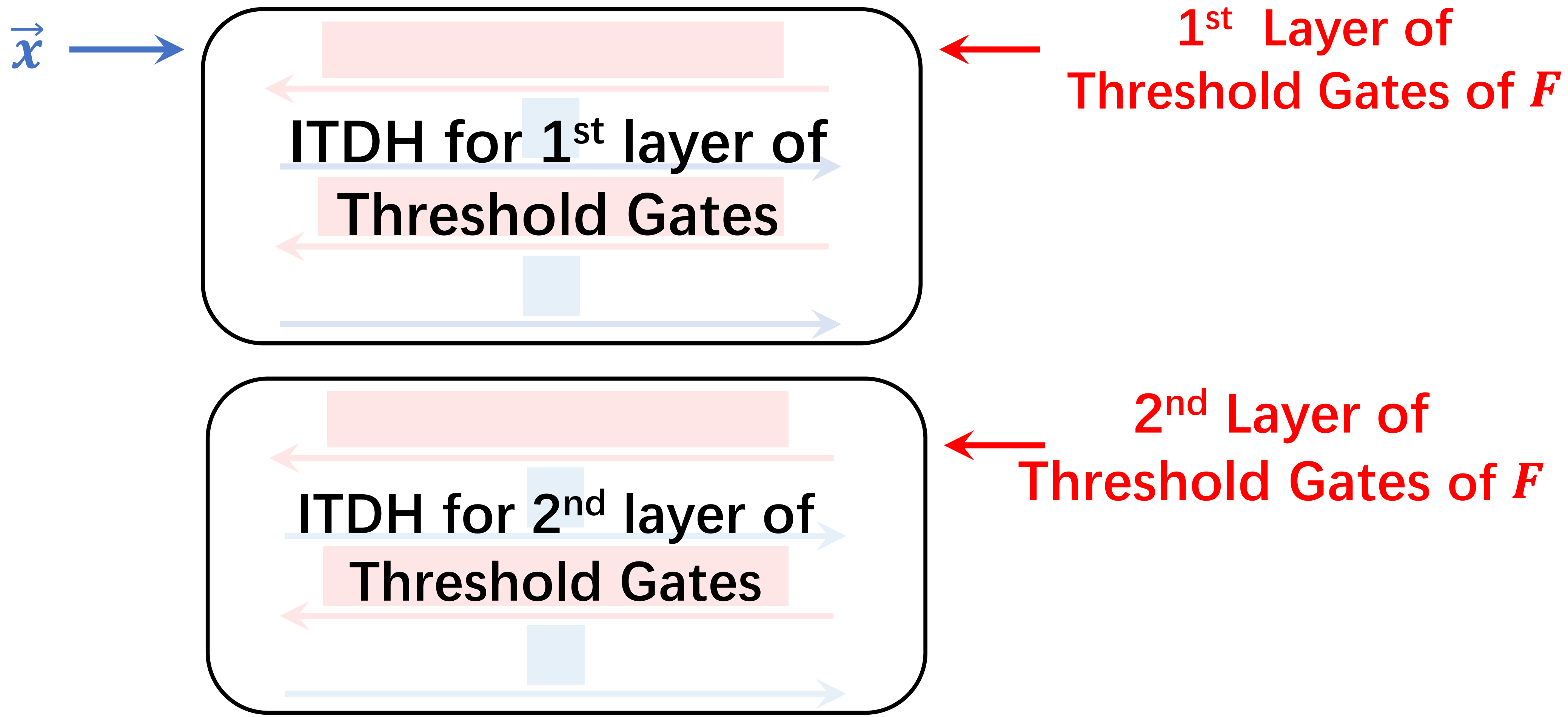
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



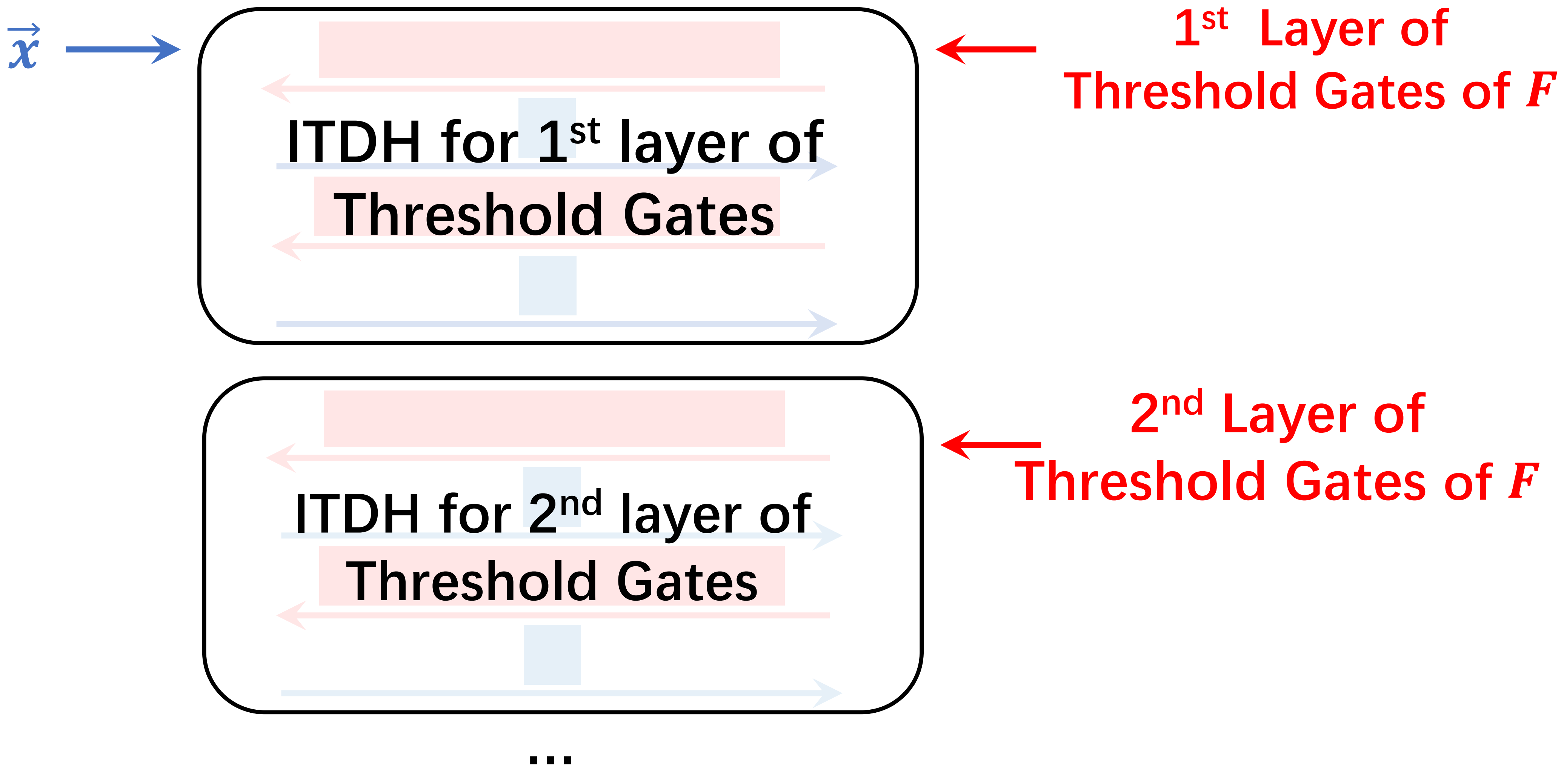
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



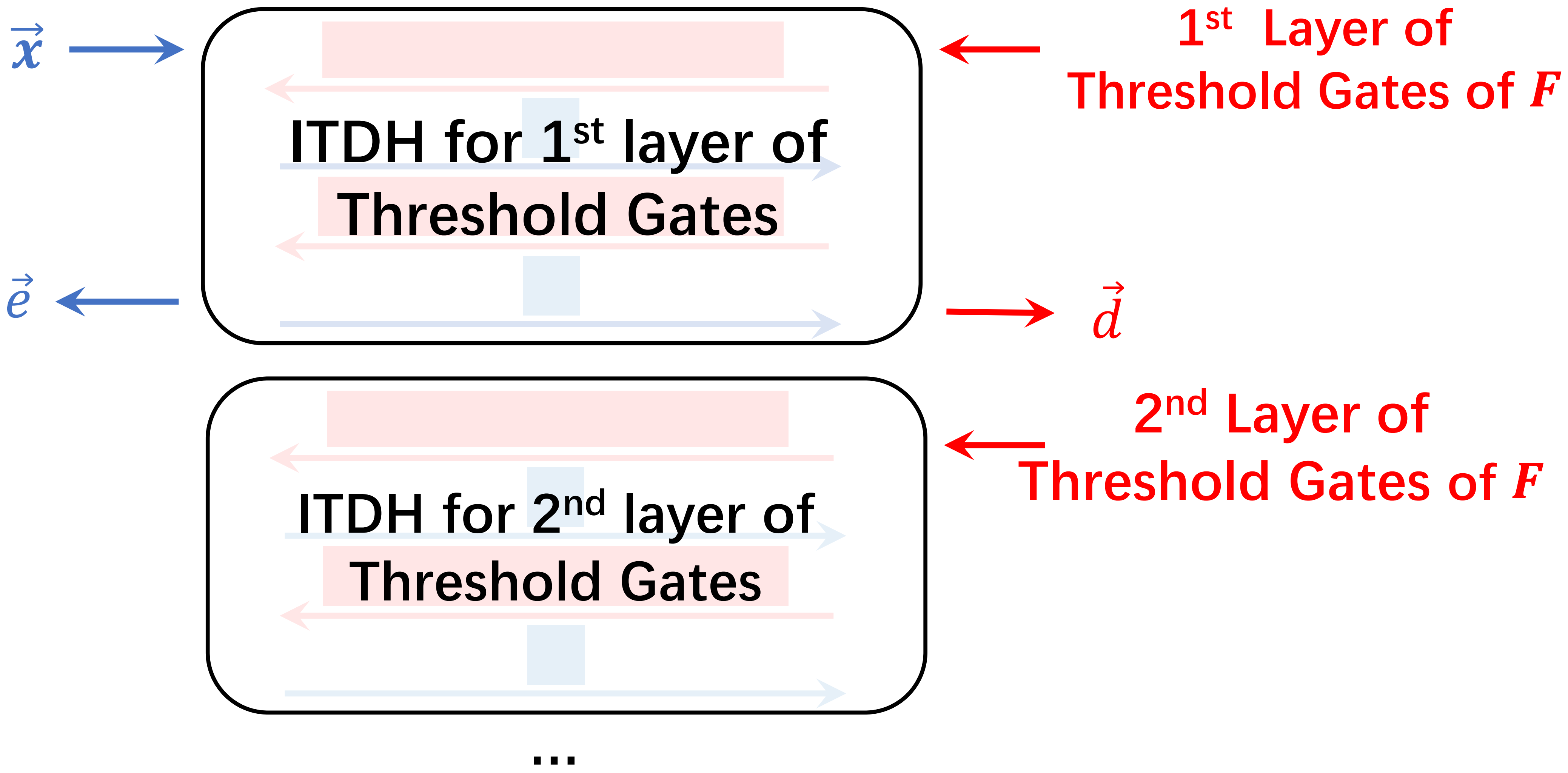
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



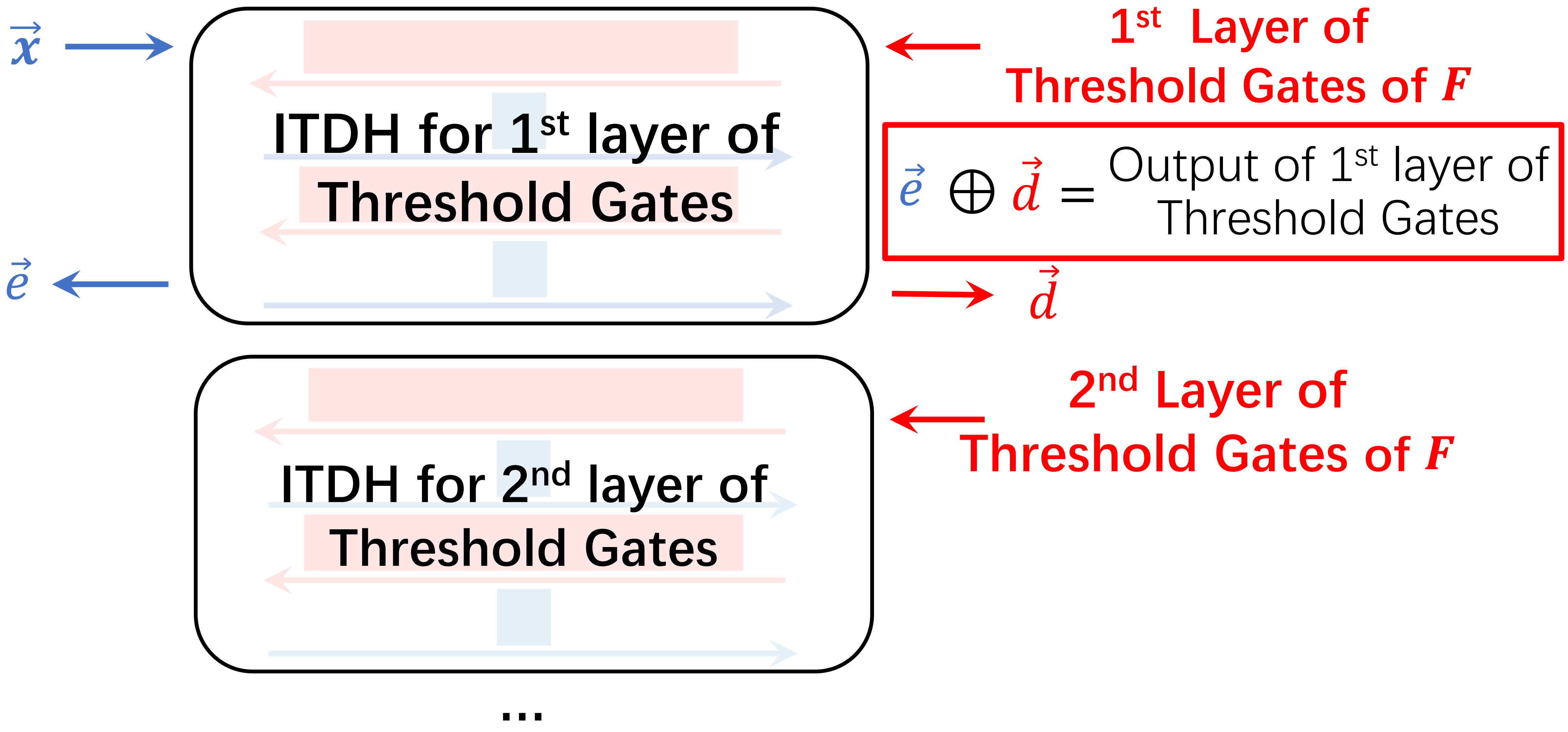
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



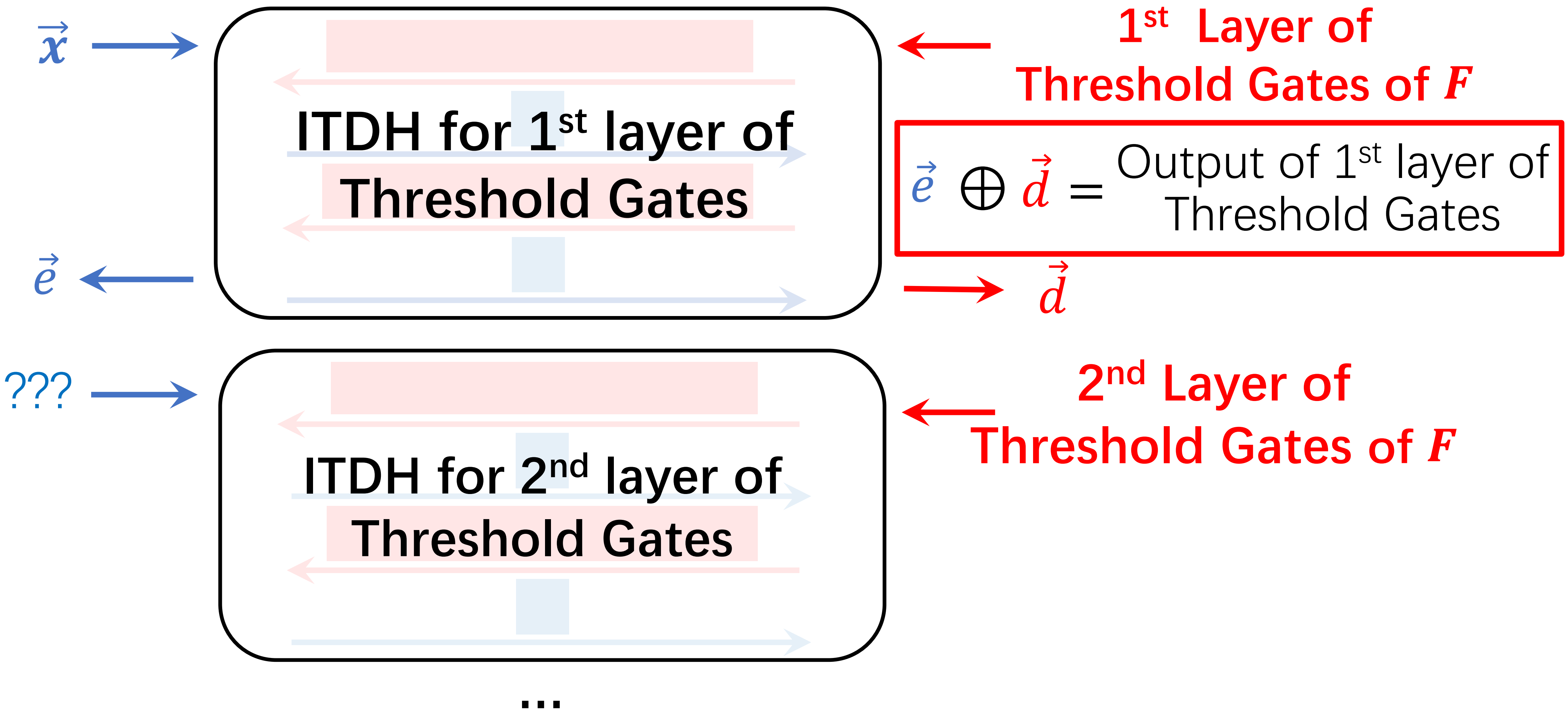
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



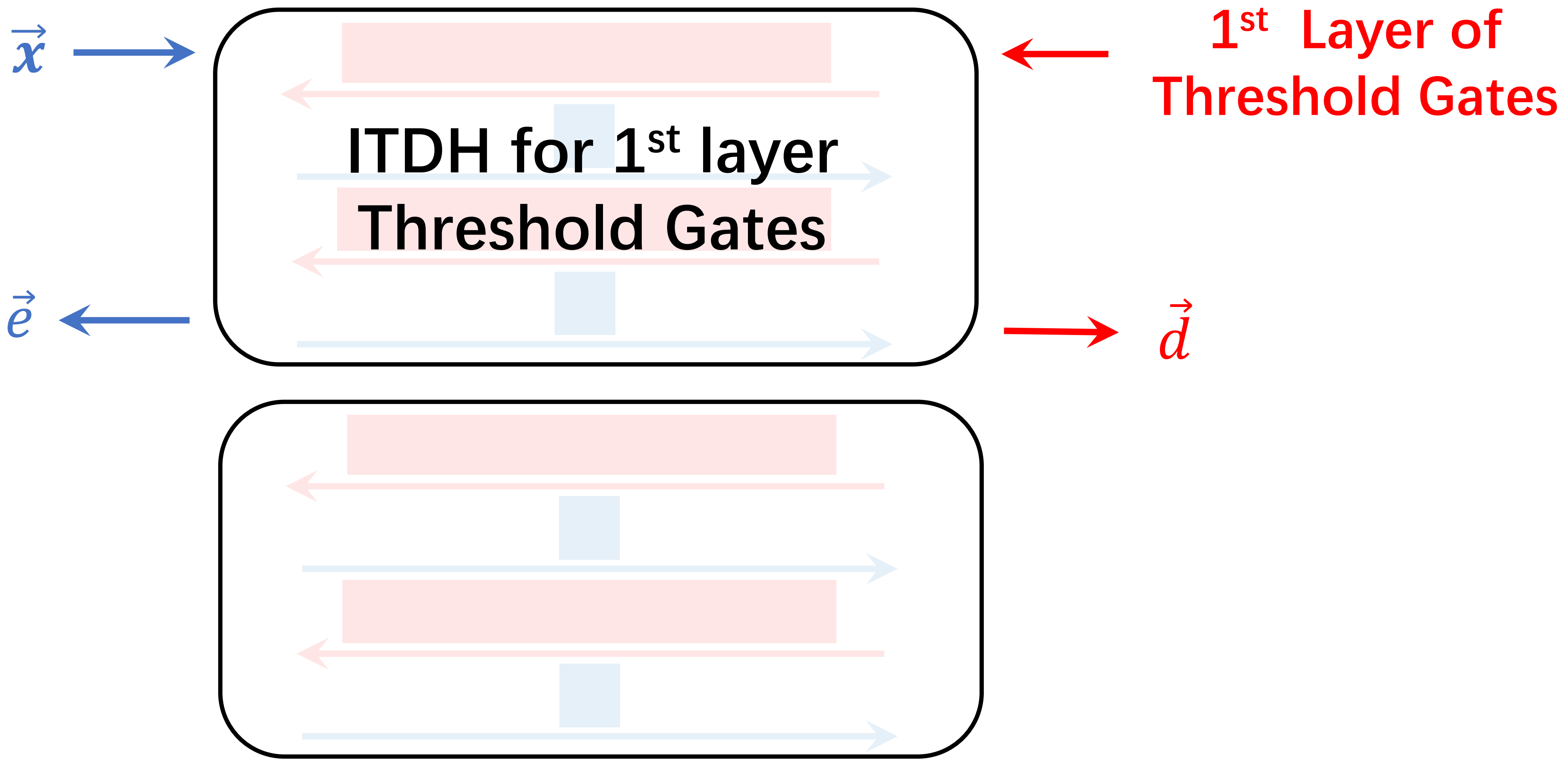
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



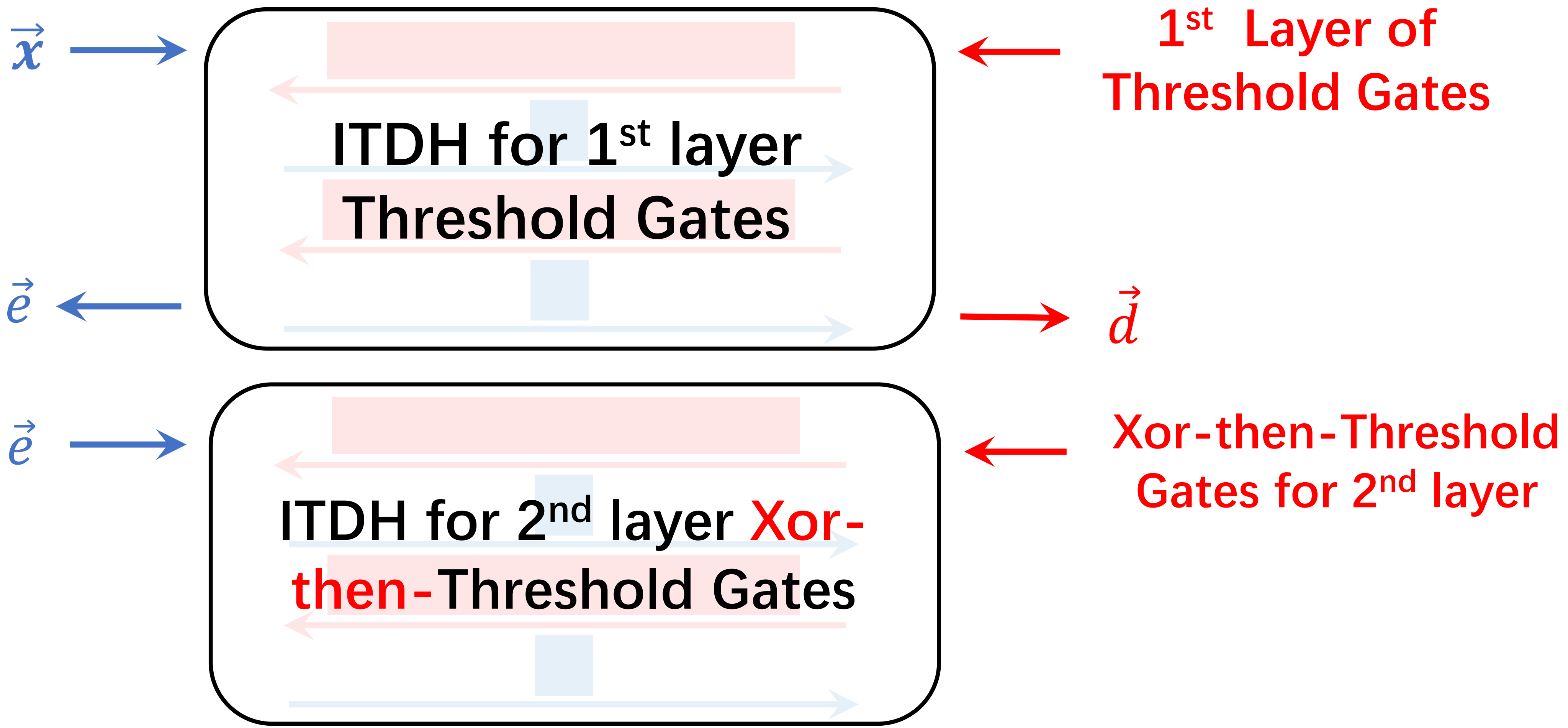
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



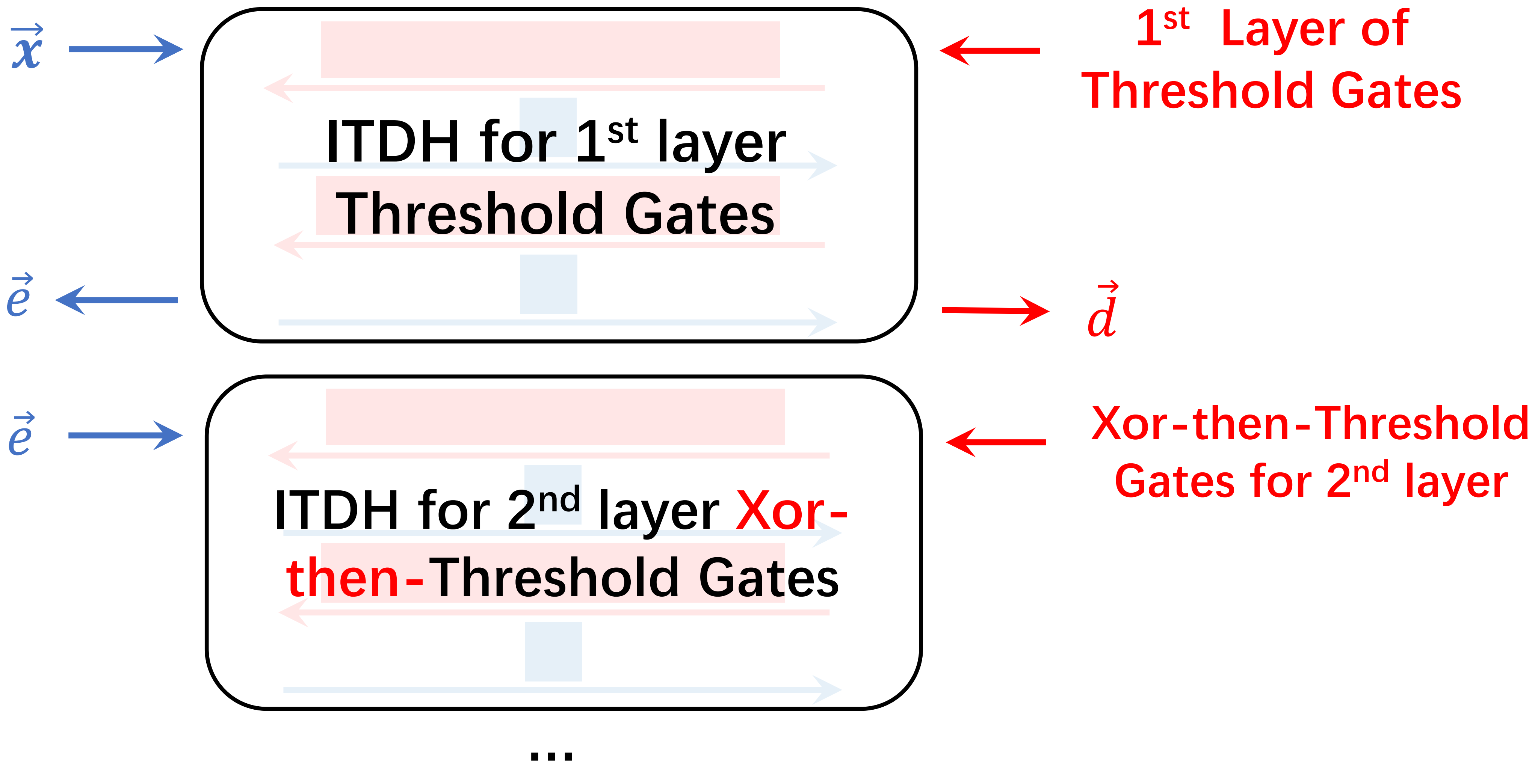
ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



ITDH for $\mathbf{TC^0}$: Layer-by-Layer Computation



Xor-then-Threshold Gate

Xor-then-Threshold = Threshold Gate ∘ XOR

Xor-then-Threshold Gate

Xor-then-Threshold = Threshold Gate ∘ XOR

$$\mathrm{Th}_{\vec{y}}^{t}(\vec{x}) = \begin{cases} 1, & \mathrm{weight}(\vec{x} \oplus \vec{y}) \geq t \\ 0, & \text{Otherwise} \end{cases}$$

ITDH for An Xor-then-Threshold Gate From TDH for Linear functions

ITDH for An Xor-then-Threshold Gate From TDH for Linear functions

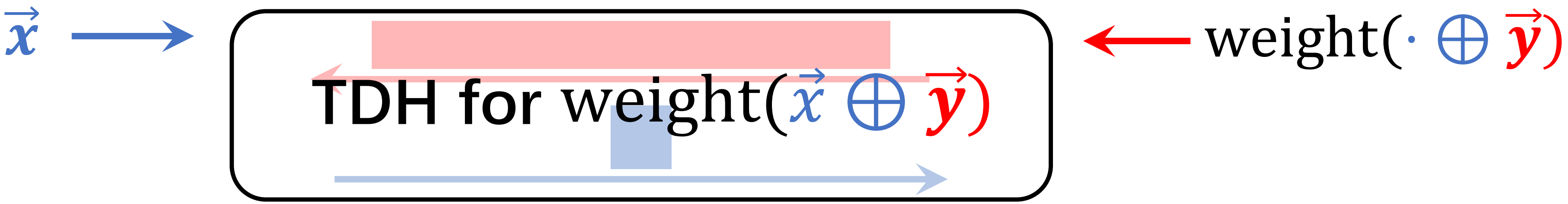
- An overview

ITDH for An Xor-then-Threshold Gate From TDH for Linear functions

- An overview

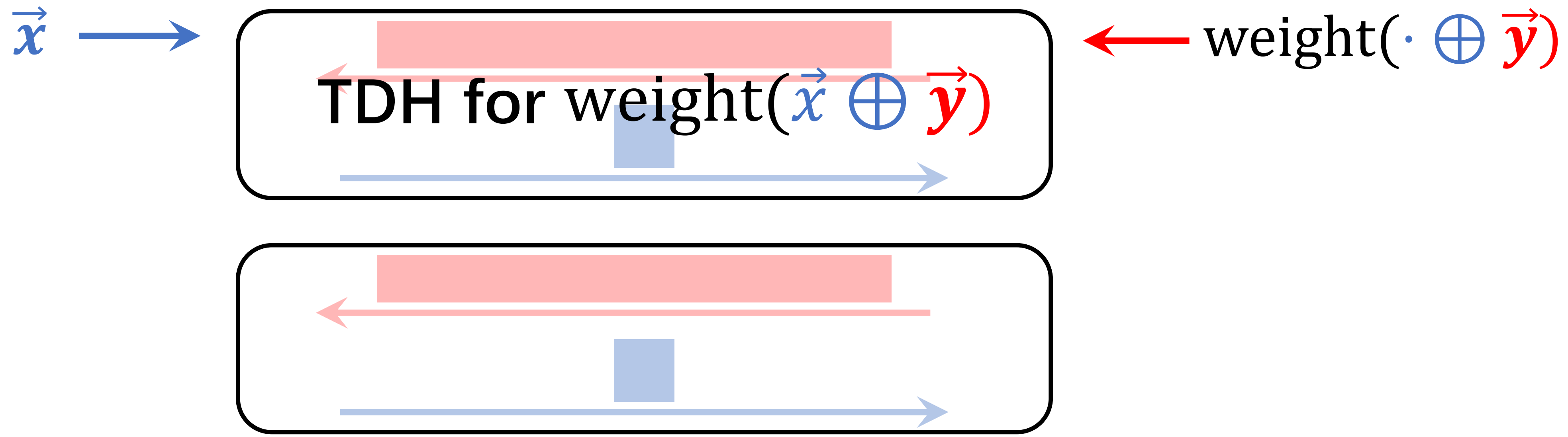
ITDH for An Xor-then-Threshold Gate From TDH for Linear functions

- An overview



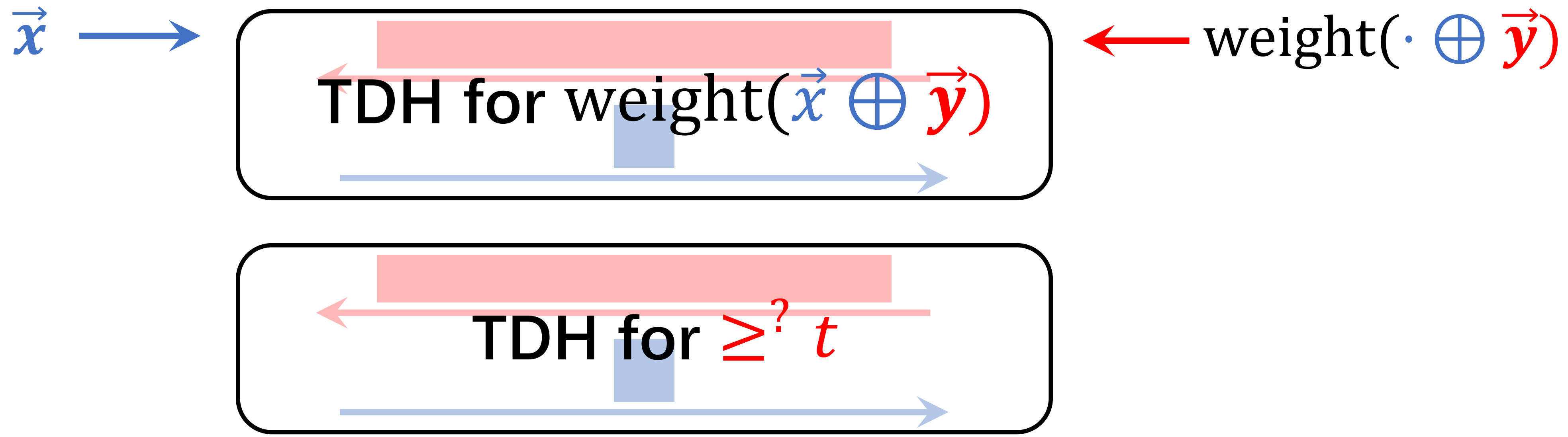
ITDH for An Xor-then-Threshold Gate From TDH for Linear functions

- An overview



ITDH for An Xor-then-Threshold Gate From TDH for Linear functions

- An overview



weight$(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$\mathrm{weight}(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$$\mathrm{weight}(\vec{x} \oplus \vec{y}) = \sum_{i} x_i \oplus y_i$$

$\text{weight}(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$$\text{weight}(\vec{x} \oplus \vec{y}) = \sum_i x_i \oplus y_i$$

$$= \sum_i (1 - x_i) \cdot y_i + x_i \cdot (1 - y_i)$$

weight($\vec{x} \oplus \vec{y}$) as a Linear Function of $\vec{x}$

$$\text{weight}(\vec{x} \oplus \vec{y}) = \sum_{i} x_i \oplus y_i$$

$$= \sum_{i} (1 - x_i) \cdot y_i + x_i \cdot (1 - y_i) \mod (n+1)$$

$\text{weight}(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$$\text{weight}(\vec{x} \oplus \vec{y}) = \sum_i x_i \oplus y_i$$

We extend TDH (from DDH) to linear functions over Z_{n+1}

$$= \sum_i (1 - x_i) \cdot y_i + x_i \cdot (1 - y_i) \mod (n+1)$$

$\text{weight}(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$$\text{weight}(\vec{x} \oplus \vec{y}) = \sum_i x_i \oplus y_i$$

We extend TDH (from DDH) to linear functions over Z_{n+1}

$$= \sum_i (1 - x_i) \cdot y_i + x_i \cdot (1 - y_i) \mod (n+1)$$

- Use TDH for Linear Functions over Z_{n+1}

$\text{weight}(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$$\text{weight}(\vec{x} \oplus \vec{y}) = \sum_i x_i \oplus y_i$$

$$= \sum_i (1 - x_i) \cdot y_i + x_i \cdot (1 - y_i) \mod (n+1)$$

We extend TDH (from DDH) to linear functions over Z_{n+1}

- Use TDH for Linear Functions over Z_{n+1}

$\mathrm{weight}(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$$\mathrm{weight}(\vec{x} \oplus \vec{y}) = \sum_i x_i \oplus y_i$$

$$= \sum_i (1 - x_i) \cdot y_i + x_i \cdot (1 - y_i) \mod (n+1)$$

We extend TDH (from DDH) to linear functions over Z_{n+1}

- Use TDH for Linear Functions over Z_{n+1}

$\vec{x} \longrightarrow$ **TDH for linear functions over Z_{n+1}** $\longleftarrow \mathrm{weight}(\cdot \oplus \vec{y})$

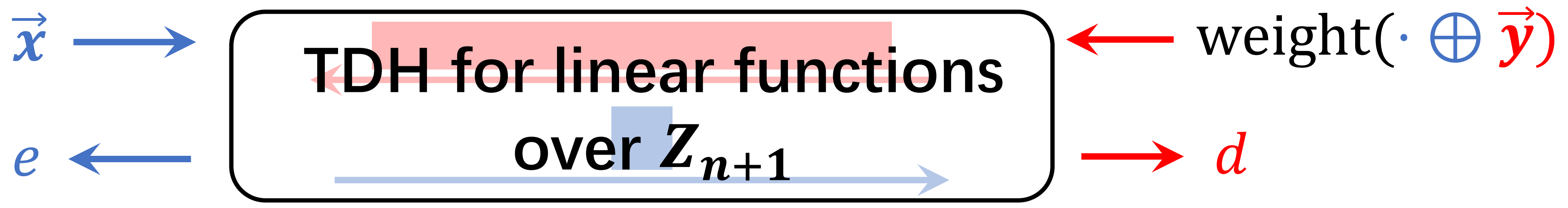
weight($\vec{x} \oplus \vec{y}$) as a Linear Function of $\vec{x}$

$$\text{weight}(\vec{x} \oplus \vec{y}) = \sum_i x_i \oplus y_i$$

$$= \sum_i (\mathbf{1} - x_i) \cdot y_i + x_i \cdot (\mathbf{1} - y_i) \mod (n+1)$$

We extend TDH (from DDH) to linear functions over Z_{n+1}

- Use TDH for Linear Functions over Z_{n+1}



$\text{weight}(\vec{x} \oplus \vec{y})$ as a Linear Function of $\vec{x}$

$$\text{weight}(\vec{x} \oplus \vec{y}) = \sum_i x_i \oplus y_i$$

We extend TDH (from DDH) to linear functions over Z_{n+1}

$$= \sum_i (1 - x_i) \cdot y_i + x_i \cdot (1 - y_i) \mod (n+1)$$

- Use TDH for Linear Functions over Z_{n+1}

$\vec{x} \longrightarrow$ **TDH for linear functions over Z_{n+1}** $\longleftarrow \text{weight}(\cdot \oplus \vec{y})$

$e \longleftarrow$ $\longrightarrow d$

How do we use TDH to compute $(e + d) \mod (n+1) \overset{?}{\geq} t$?

Comparison as a Linear Function

Comparison as a Linear Function

- A simpler case: equality check $e \stackrel{?}{=} d$

Comparison as a Linear Function

- A simpler case: equality check $e \stackrel{?}{=} d$

  $e, d \in [0, 1, \ldots, n]$: a poly range!

Comparison as a Linear Function

- A simpler case: equality check $e \stackrel{?}{=} d$

  $e, d \in [0, 1, \ldots, n]$: a poly range!

	0	1	...			n
$e \to 1_e =$	0	0	1	0	0	0
$d \to 1_d =$	0	0	0	1	0	0

Comparison as a Linear Function

- A simpler case: equality check $e =^? d$

  $e, d \in [0, 1, \ldots, n]$: a poly range!

	0	1	…			n
$e \to 1_e =$	0	0	1	0	0	0
$d \to 1_d =$	0	0	0	1	0	0

$$(e =^? d) = < 1_e, 1_d >$$

Comparison as a Linear Function

Comparison as a Linear Function

- Comparison: $(e + d) \bmod (n+1) \geq^? t$

Comparison as a Linear Function

- Comparison: $(e + d) \bmod (n+1) \overset{?}{\geq} t$

$$\Leftrightarrow \exists^{?}\, j \geq t : (e + d) \bmod (n+1) = j$$

Comparison as a Linear Function

- Comparison: $(e + d) \bmod (n+1) \geq^{?} t$

$$\Leftrightarrow \exists^{?}\, j \geq t : \boxed{(e + d) \bmod (n+1) = j}$$

Comparison as a Linear Function

- Comparison: $(e+d) \bmod (n+1) \overset{?}{\geq} t$

$$\Leftrightarrow \exists^{?}\, j \geq t : \boxed{(e+d) \bmod (n+1) = j}$$

Equality Check! $\quad e = (j-d) \bmod (n+1)$

Comparison as a Linear Function

- Comparison: $(e + d) \bmod (n+1) \geq^? t$

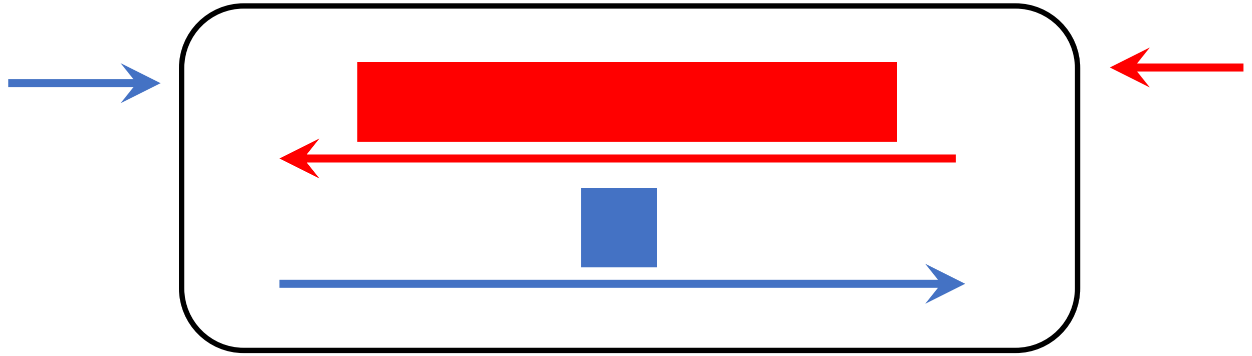
$$\Leftrightarrow \exists^? j \geq t : \boxed{(e + d) \bmod (n+1) = j}$$

Equality Check! $e = (j - d) \bmod (n+1)$

$$\Leftrightarrow < 1_e, \sum_{j \geq t} 1_{(j-d) \bmod (n+1)} >= 1$$

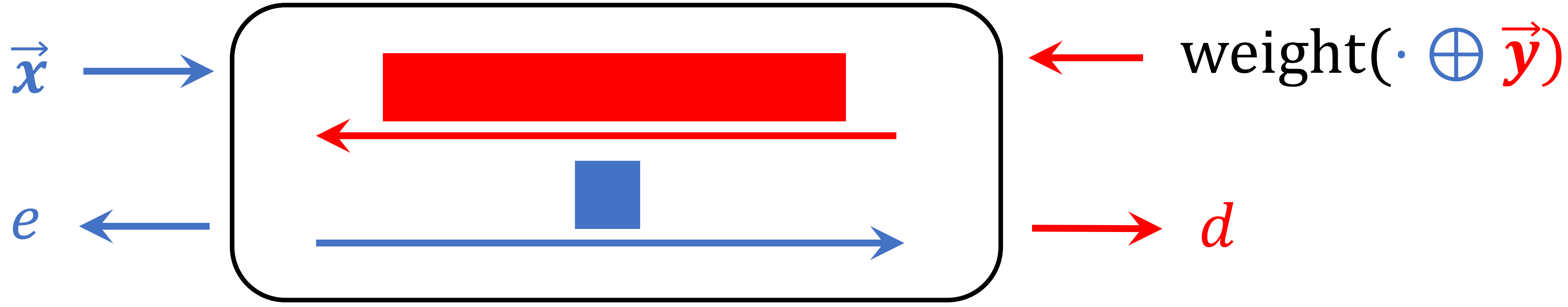
ITDH for An Xor-then-Threshold Gate: Putting Things Together

$\vec{x}$

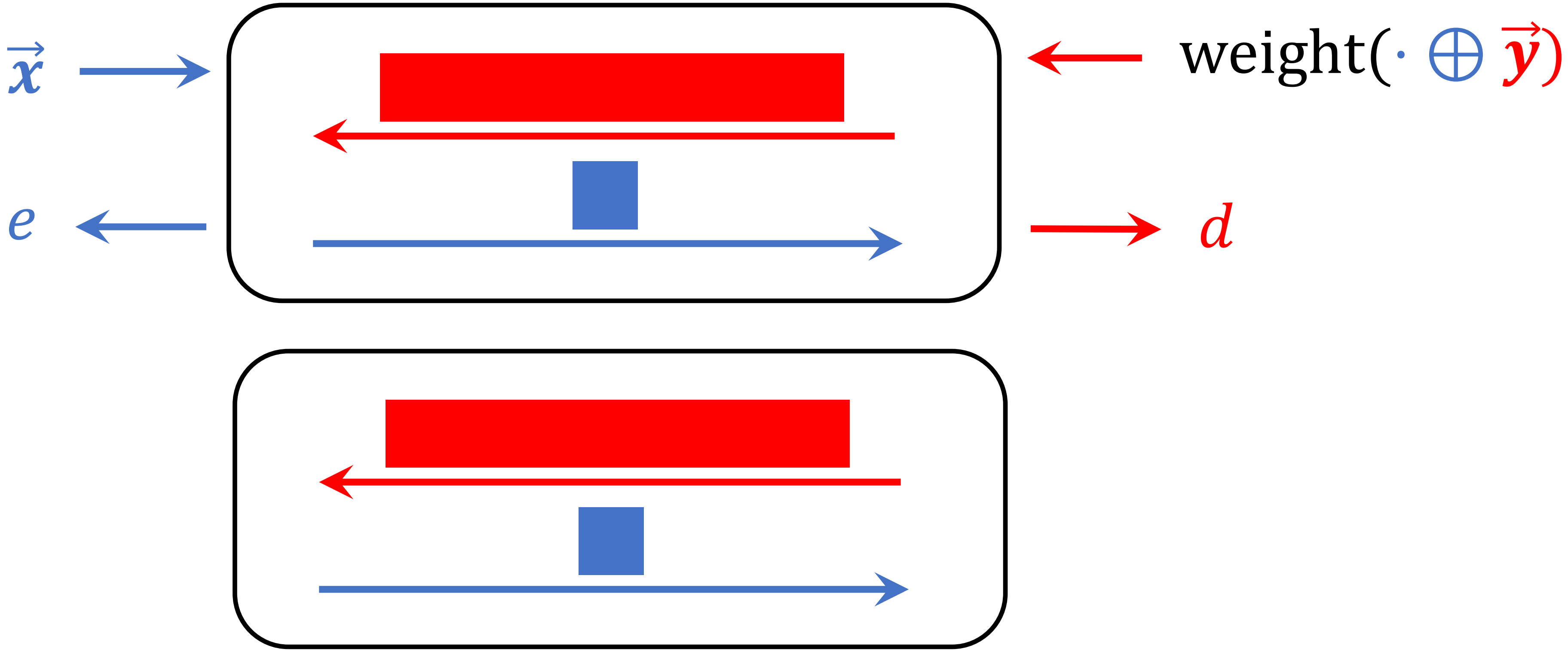


weight$(\cdot \oplus \vec{y})$

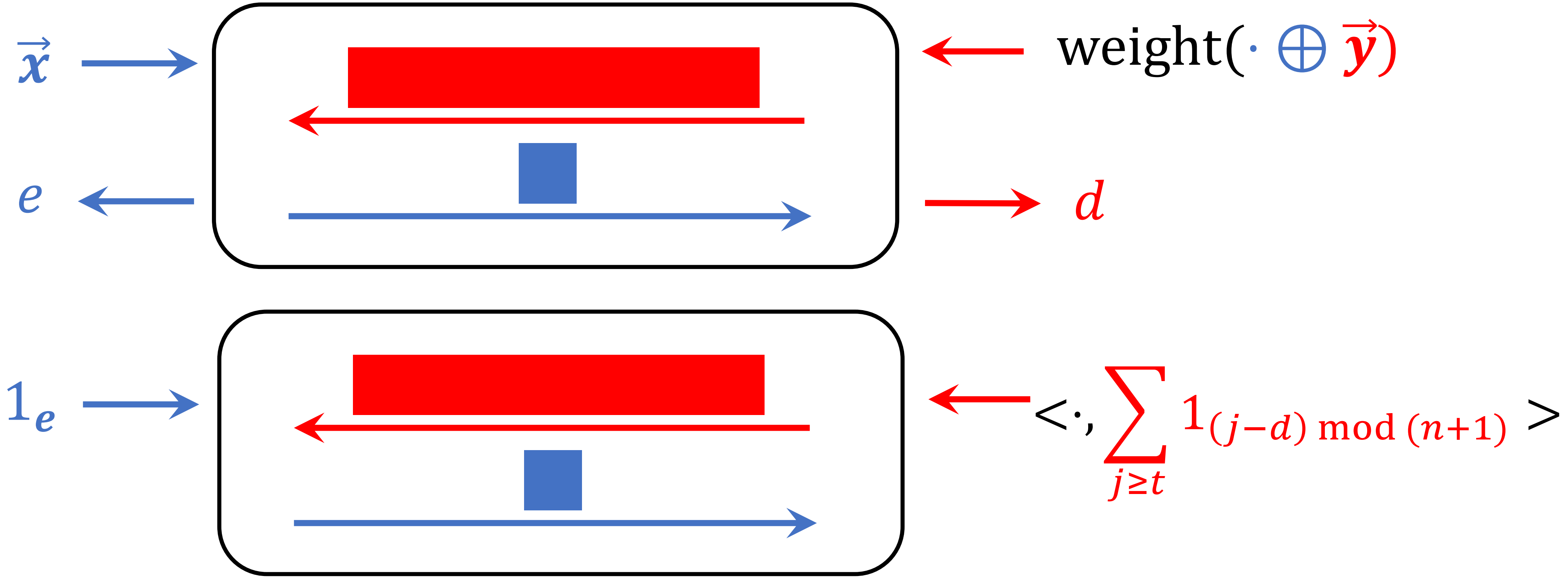
ITDH for An Xor-then-Threshold Gate: Putting Things Together



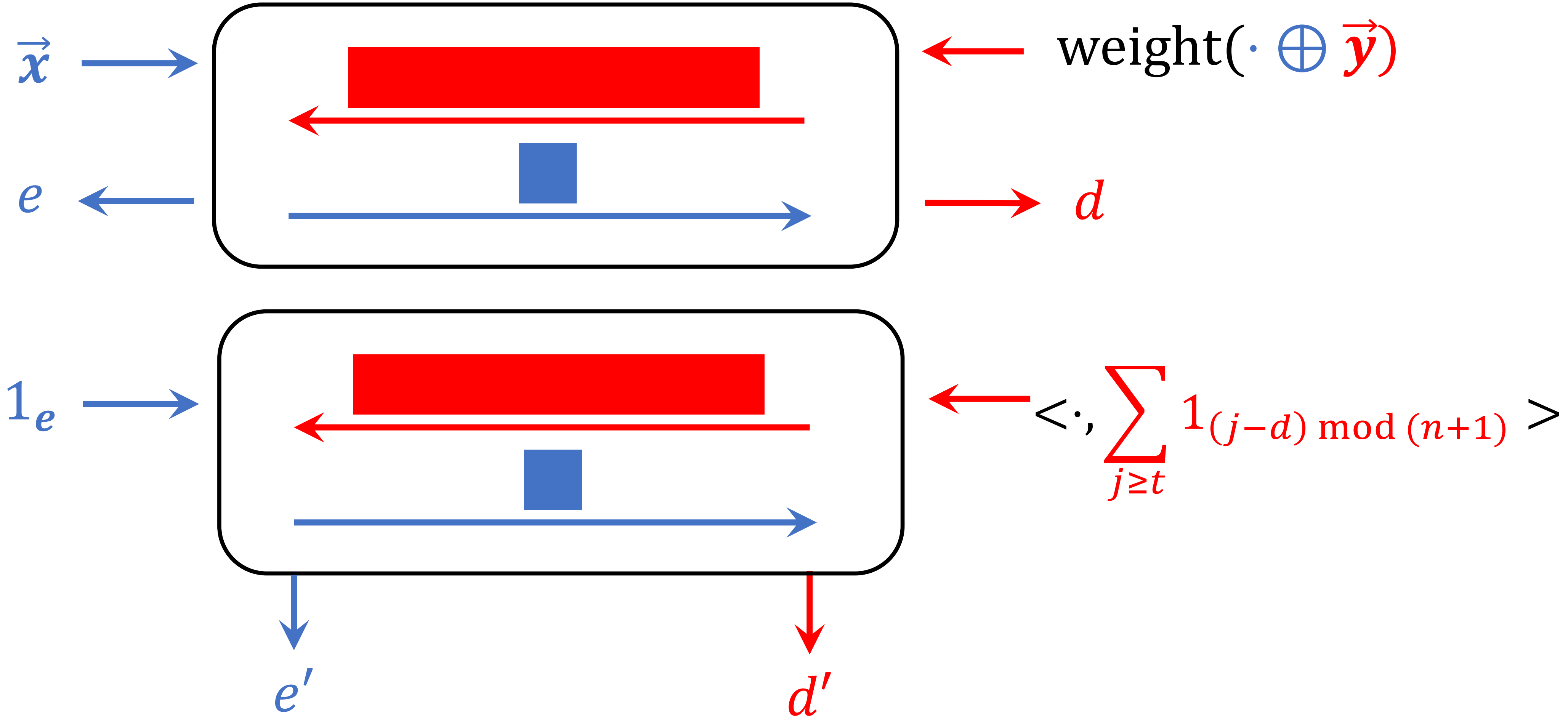
ITDH for An Xor-then-Threshold Gate: Putting Things Together



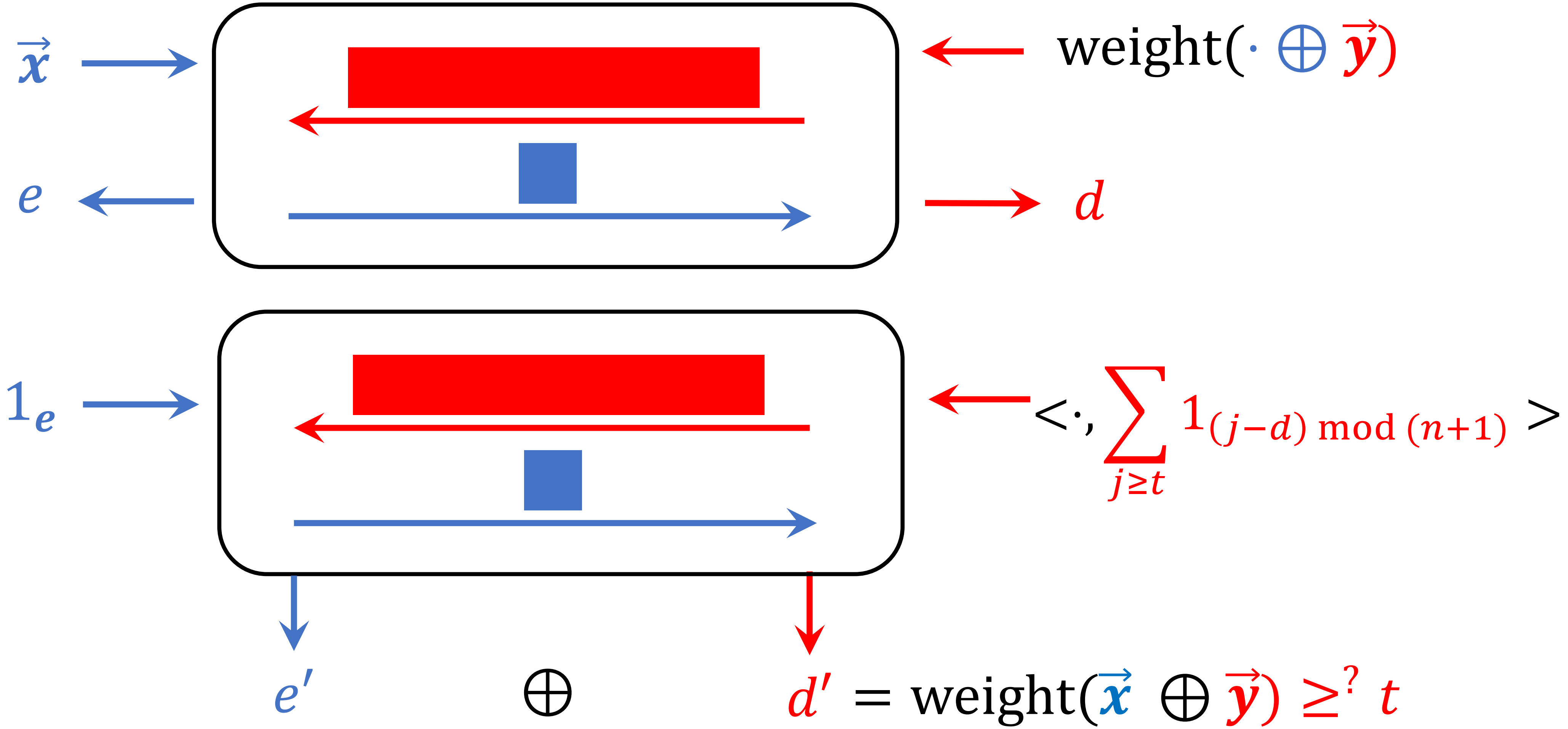
ITDH for An Xor-then-Threshold Gate: Putting Things Together



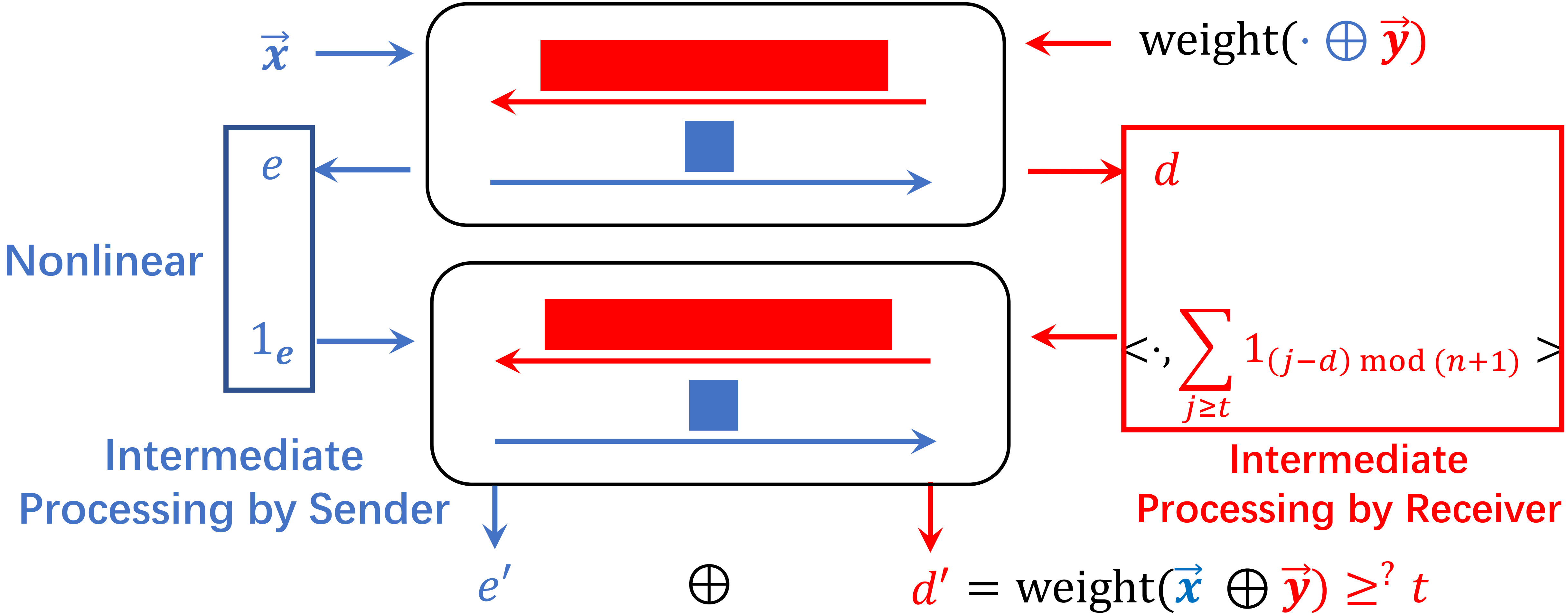
ITDH for An Xor-then-Threshold Gate: Putting Things Together



ITDH for An Xor-then-Threshold Gate: Putting Things Together



ITDH for An Xor-then-Threshold Gate: Putting Things Together



Summary of Results

- NIZKs from sub-exponential DDH:

	Zero-Knowledge	Soundness	CRS
I	Statistical	Non-adaptive	Random
II	Computational	Adaptive	Random

  - $O(1)$-round Interactive Trapdoor Hashing Protocol for $\mathbf{TC}^0$
  - Correlation Intractable Hash for $\mathbf{TC}^0$.

- Statistical Zap arguments from sub-exponential DDH.

Open Questions

- NIZKs from polynomial-hard DDH?
- NIZKs from public key encryption?
- Correlation intractable hash for $\mathbf{P/poly}$ from DDH?

Thank you!

Q & A