Always Have a Backup Plan: Fully Secure Synchronous MPC with Asynchronous Fallback

Erica Blum

U. Maryland

Chen-Da Liu-Zhang

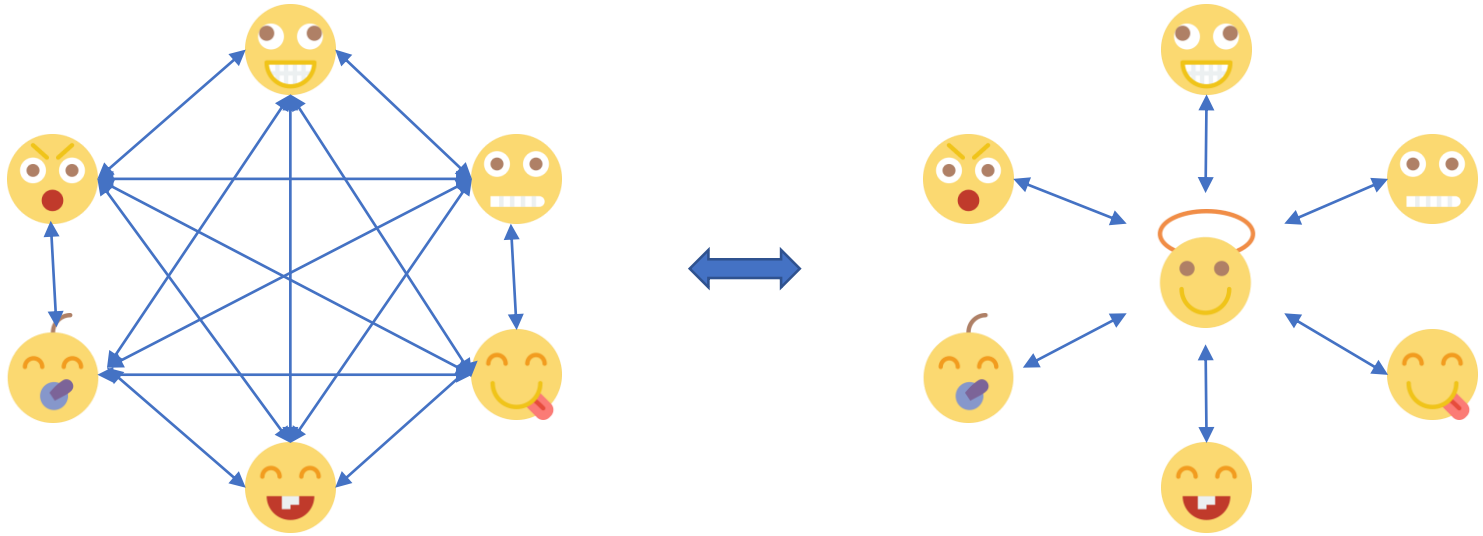
ETH Zurich

Julian Loss

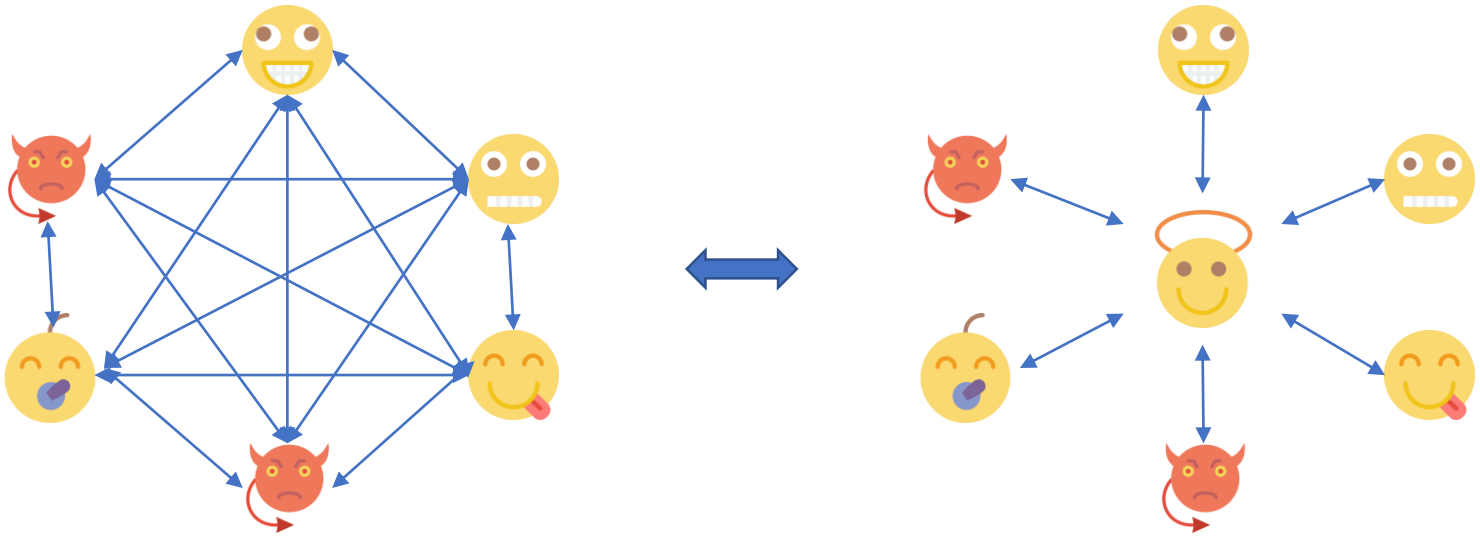
U. Maryland

CRYPTO 2020

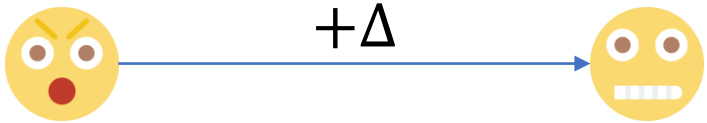
Multiparty Computation



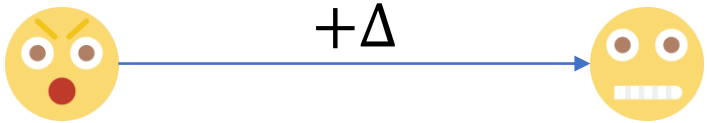
Multiparty Computation



Synchronous

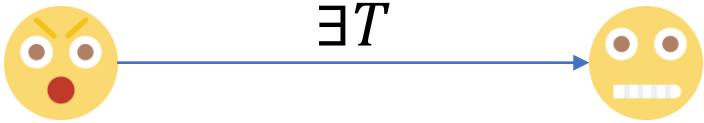


Synchronous



Asynchronous

-



Synchronous

$t < \frac{n}{2}$

Input completeness

$f(x_1, \ldots, x_n)$

Synchronous

$t < \frac{n}{2}$

Input completeness

$f(x_1, \dots, x_n)$

Asynchronous

$t < \frac{n}{3}$

Up to t inputs are ignored

Synchronous

$t < \frac{n}{2}$

Input completeness

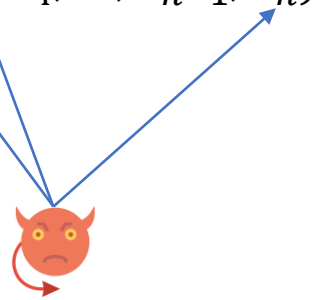
$f(x_1, \dots, x_n)$

Asynchronous

$t < \frac{n}{3}$

Up to t inputs are ignored

$f(x_1, x_2, x_3, x_4, \dots, x_{n-1}, x_n)$



Synchronous

$t < \frac{n}{2}$

Input completeness

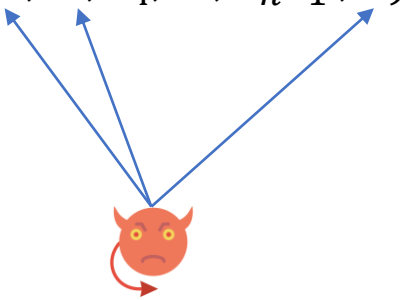
$f(x_1, \ldots, x_n)$

Asynchronous

$t < \frac{n}{3}$

Up to t inputs are ignored

$f(x_1, \perp, \perp, x_4, \ldots, x_{n-1}, \perp)$



Is there an MPC that achieves guarantees in both regimes?

Is there an MPC that achieves guarantees in both regimes?

Synchronous

$^{n}/_{3} \leq t_s < {}^{n}/_{2}$

Input completeness

Asynchronous

$t_a < {}^{n}/_{3}$

Up to L inputs are ignored

Is there an MPC that achieves guarantees in both regimes?

Synchronous

$n/3 \leq t_s < n/2$

Input completeness

Asynchronous

$t_a < n/3$

Up to L inputs are ignored

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

Related Work

[BKL19],[BKL20]: Trade-offs for Consensus primitives.

t_s-secure (resp. t_a) under a synchronous (resp. asynch.) network for any $t_a + 2t_s < n$

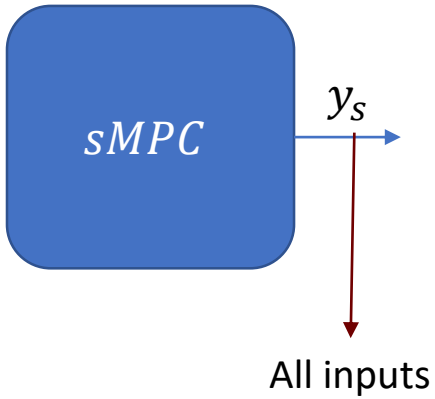
[HNP05]: Asynchronous MPC for $t < \frac{n}{3}$ with input completeness when the network is synchronous

Feasibility

Feasibility

Core Primitive

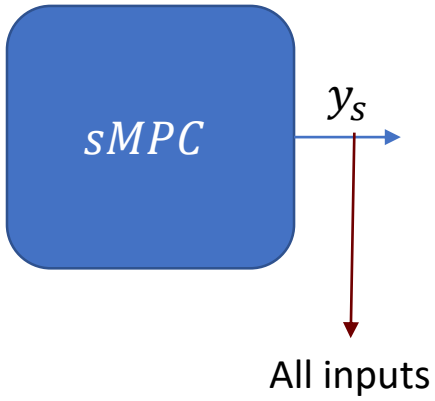
Synchronous



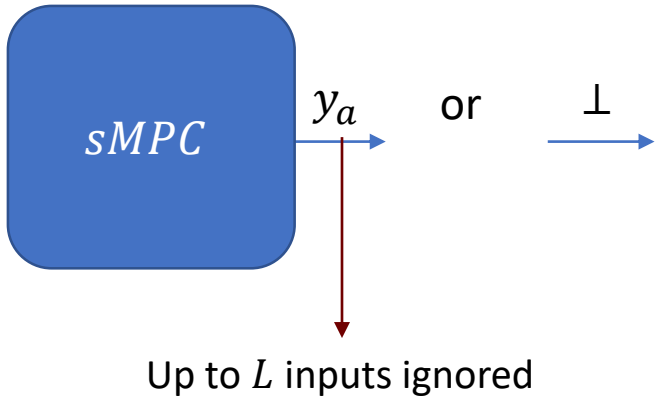
Feasibility

Core Primitive

Synchronous

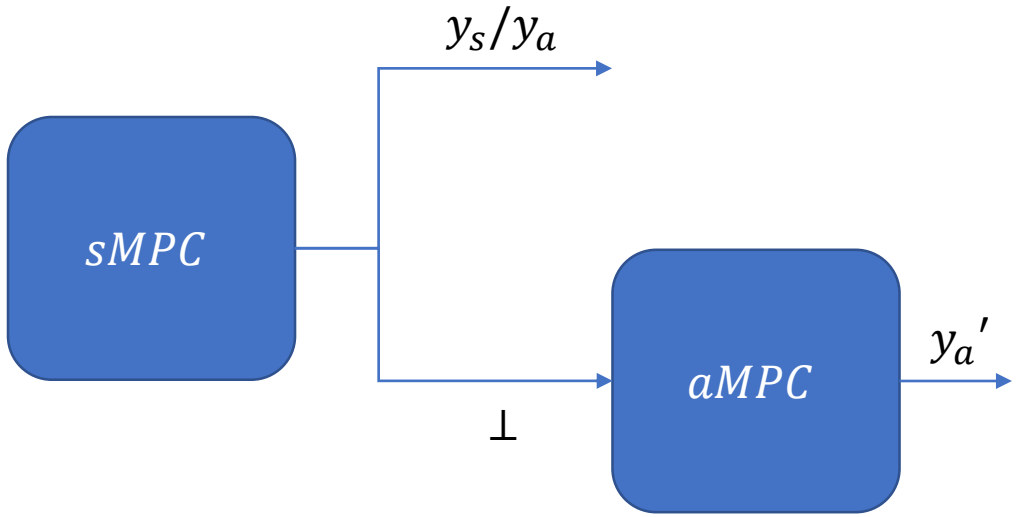


Asynchronous



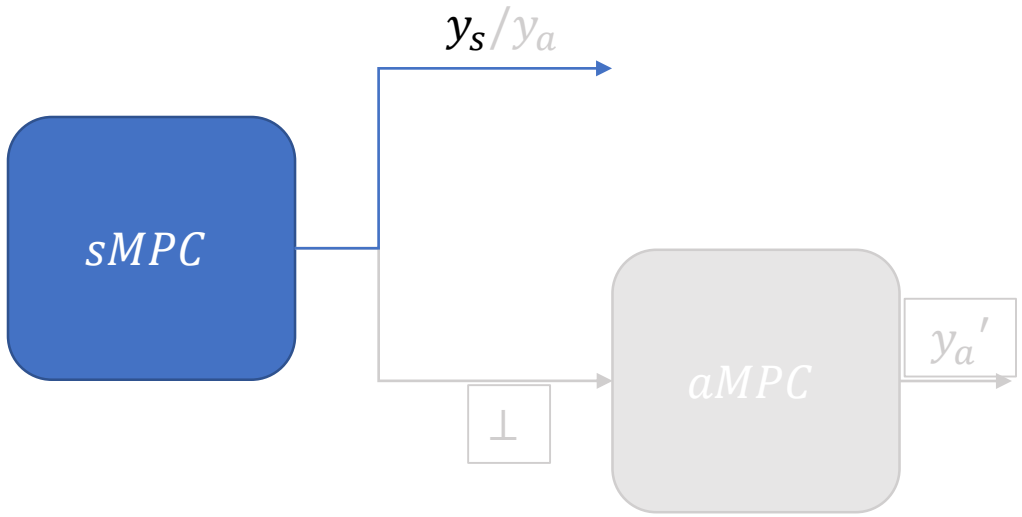
Feasibility

Compiler



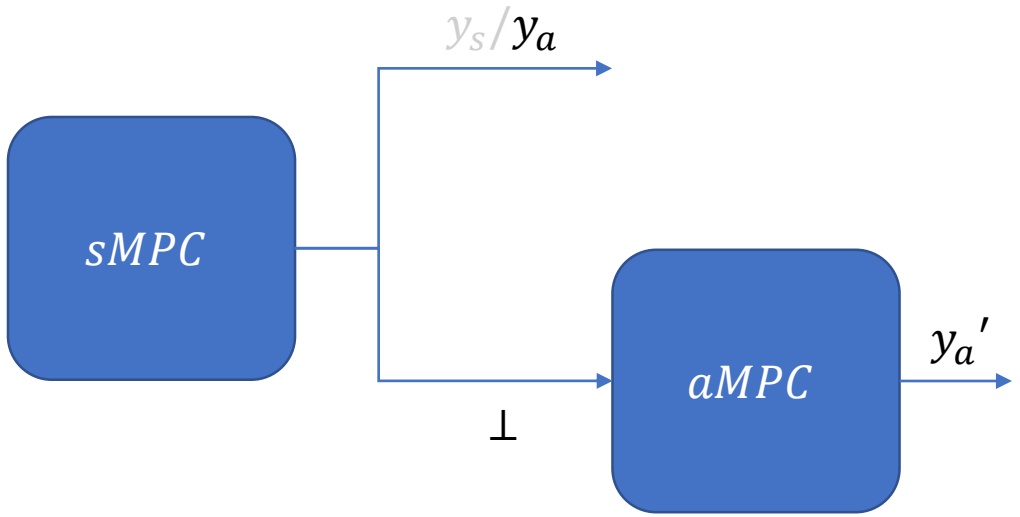
Feasibility

Compiler



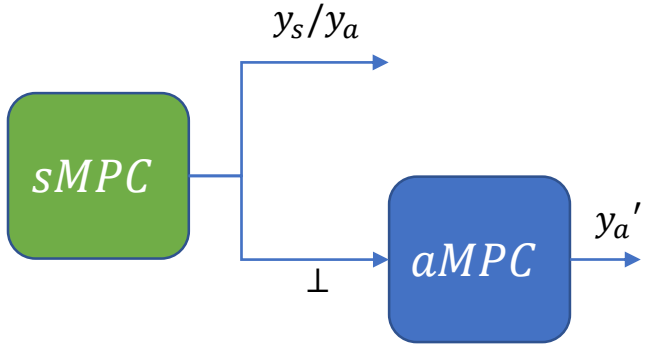
Feasibility

Compiler

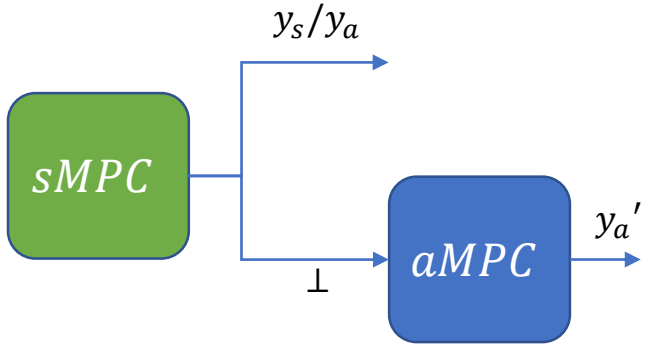


Feasibility

Threshold-Homomorphic Encryption



Feasibility

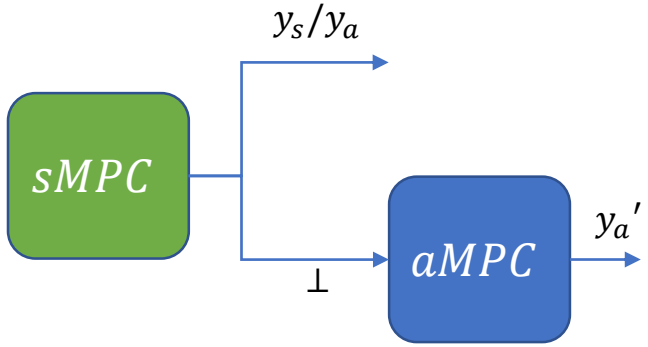


Threshold-Homomorphic Encryption

Public key ek
Secret key $dk = (dk_1, dk_2, \ldots, dk_i, \ldots dk_n)$

P_i

Feasibility



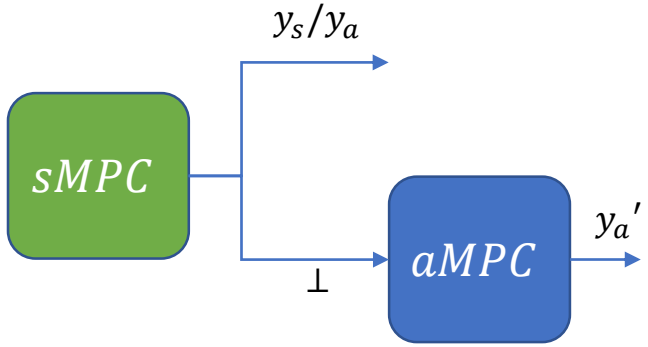
Threshold-Homomorphic Encryption

Public key ek
Secret key $dk = (dk_1, dk_2, \ldots, dk_i, \ldots dk_n)$

P_i

$$[x_i] = Enc_{ek}(x_i)$$

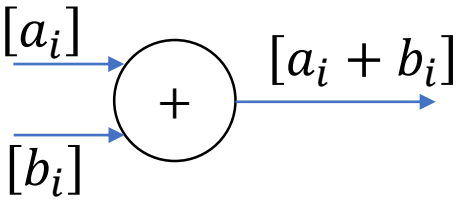
Feasibility



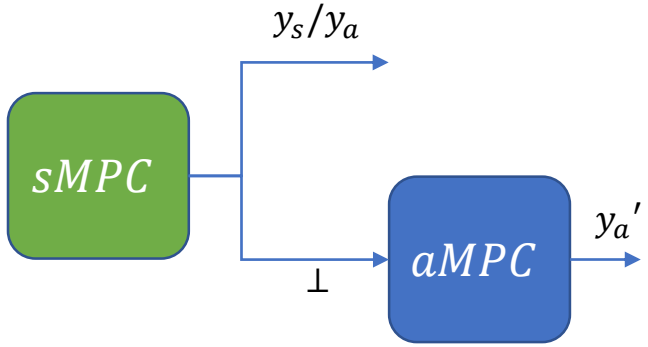
Threshold-Homomorphic Encryption

Public key ek
Secret key $dk = (dk_1, dk_2, \ldots, dk_i, \ldots dk_n)$

$$[x_i] = Enc_{ek}(x_i)$$

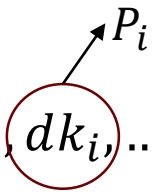


Feasibility

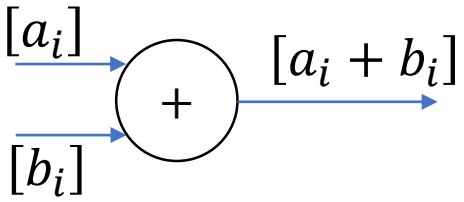


Threshold-Homomorphic Encryption

Public key ek
Secret key $dk = (dk_1, dk_2, \ldots, dk_i, \ldots dk_n)$

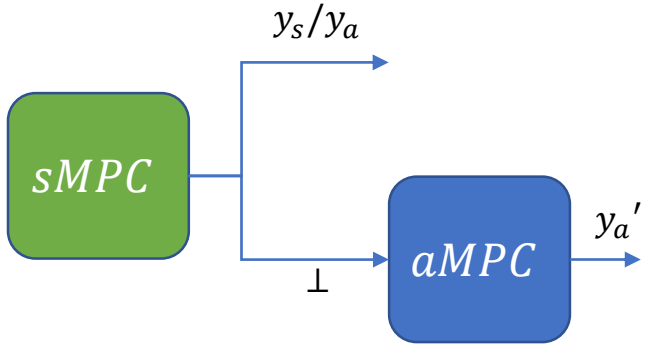


$$[x_i] = Enc_{ek}(x_i)$$



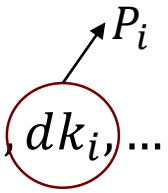
$$d_i = DecShare_{dk_i}([x_i])$$

Feasibility

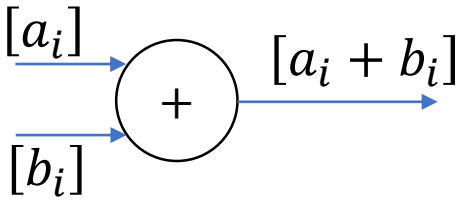


Threshold-Homomorphic Encryption

Public key ek
Secret key $dk = (dk_1, dk_2, \dots, dk_i, \dots dk_n)$



$$[x_i] = Enc_{ek}(x_i)$$



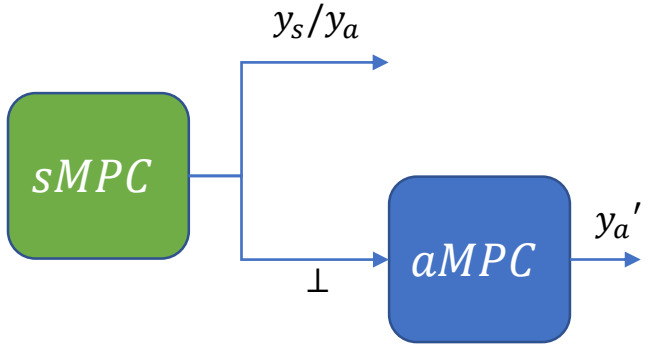
$$d_i = DecShare_{dk_i}([x_i])$$

$$x_i = Rec(\{d_j\})$$

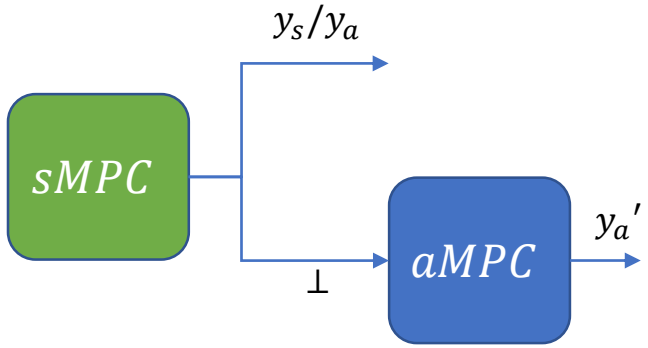
$t_s + 1$ shares

Feasibility

[CDN'01] approach



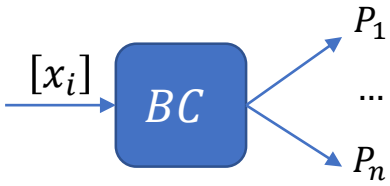
Feasibility



[CDN'01] approach

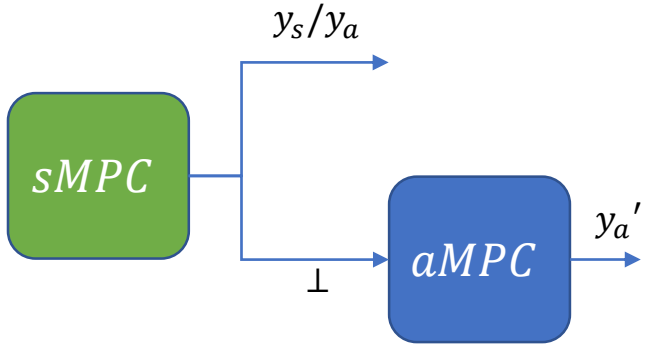
P_i:

$$[x_i] = Enc_{ek}(x_i)$$



Input Distribution

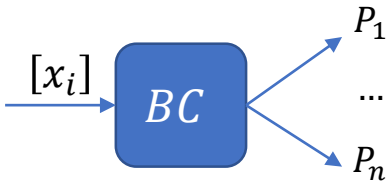
Feasibility



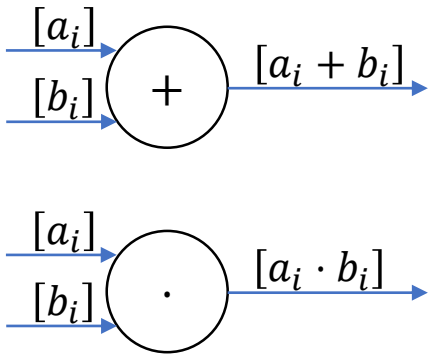
[CDN'01] approach

P_i:

$[x_i] = Enc_{ek}(x_i)$

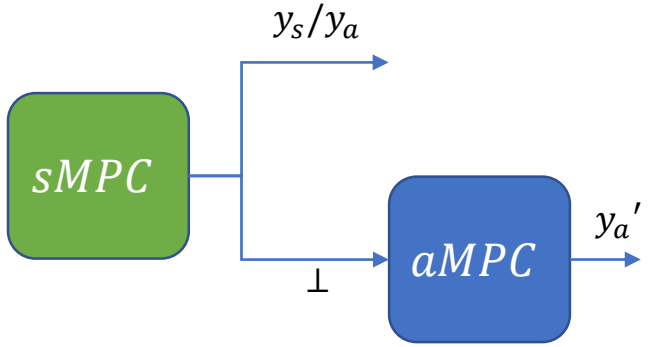


Input Distribution



Circuit Evaluation

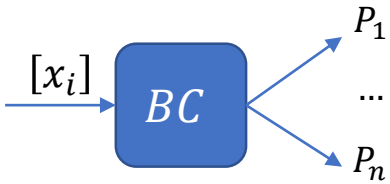
Feasibility



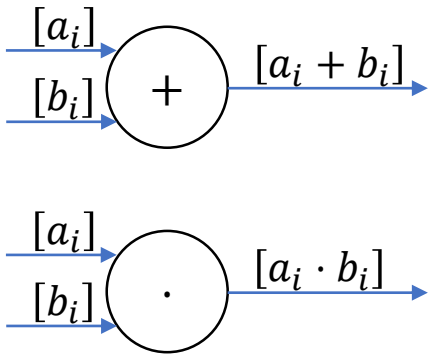
[CDN'01] approach

P_i:

$[x_i] = Enc_{ek}(x_i)$



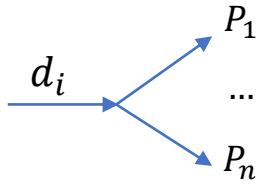
Input Distribution



Circuit Evaluation

c output ciphertext

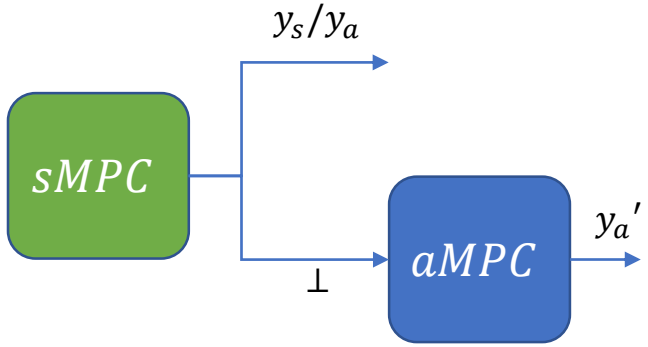
$d_i = DecShare_{dk_i}(c)$



$y_i = Rec(\{d_j\})$

Threshold Decryption

Feasibility

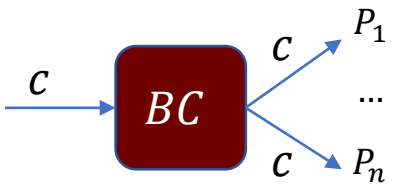


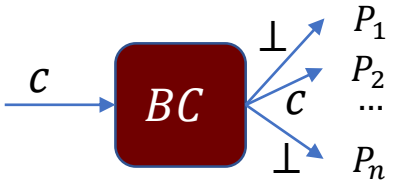
Synchronous

Asynchronous

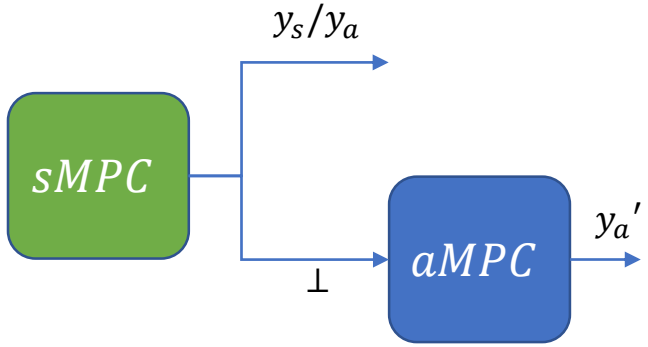
BC

[DS83]





Feasibility

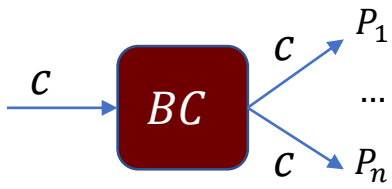


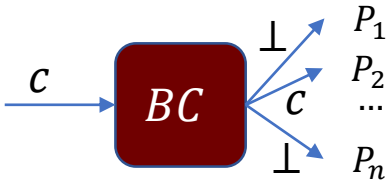
Synchronous

Asynchronous

BC

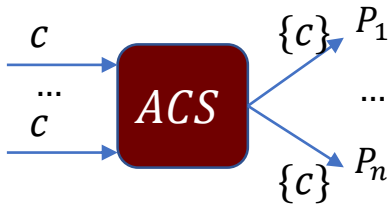
[DS83]

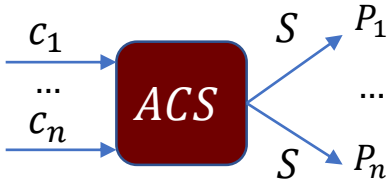




ACS

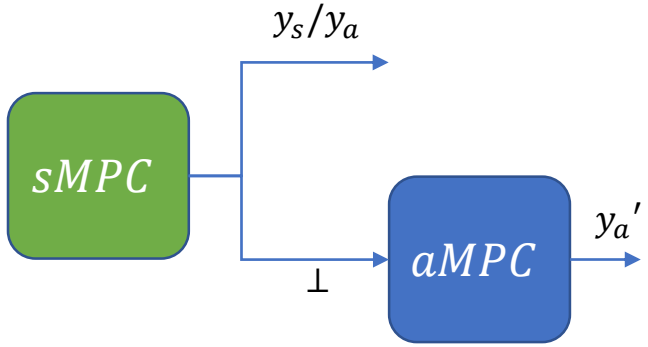
[BKR94,BKL20]





S includes an honest P_i's input

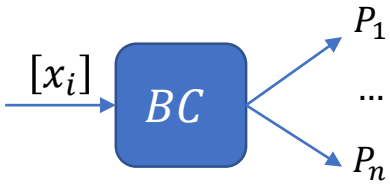
Feasibility



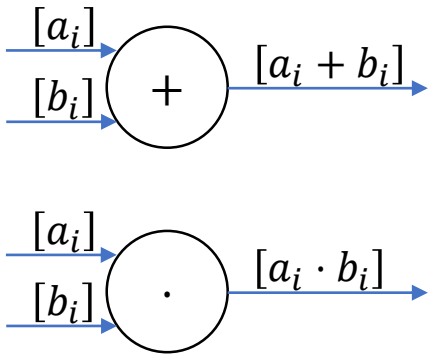
[CDN'01] approach

P_i:

$[x_i] = Enc_{ek}(x_i)$



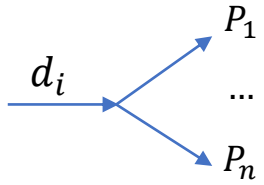
Input Distribution



Circuit Evaluation

c output ciphertext

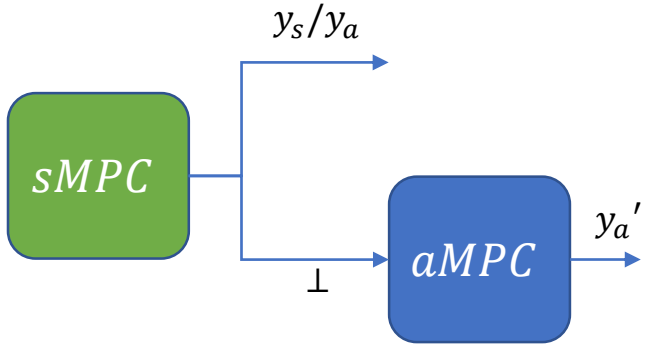
$d_i = DecShare_{dk_i}(c)$



$y_i = Rec(\{d_j\})$

Threshold Decryption

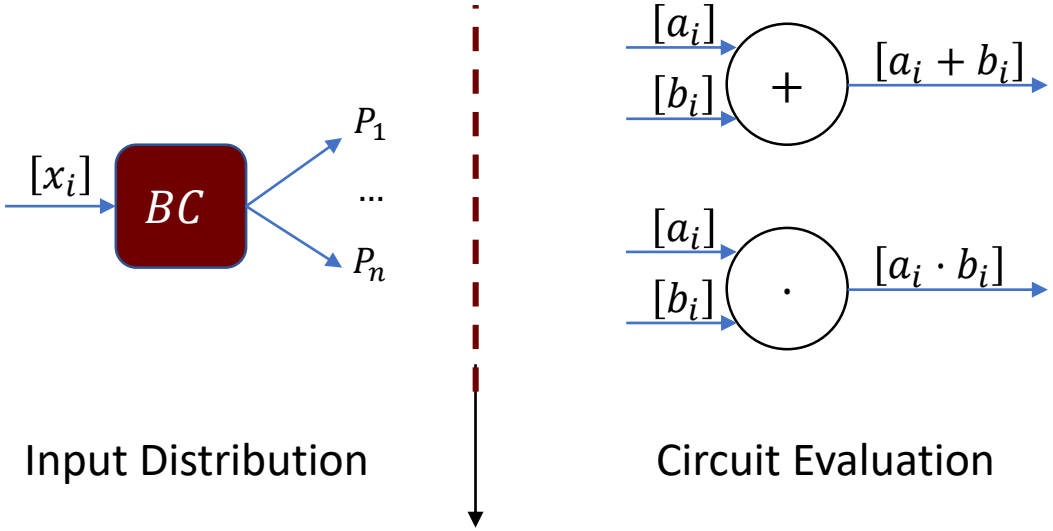
Feasibility



[CDN'01] approach

P_i:

$[x_i] = Enc_{ek}(x_i)$



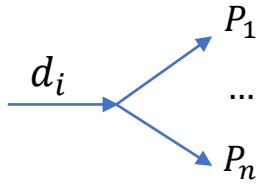
Input Distribution

Circuit Evaluation

Wait for T time
Check if there are
$n - t_s$ inputs

c output ciphertext

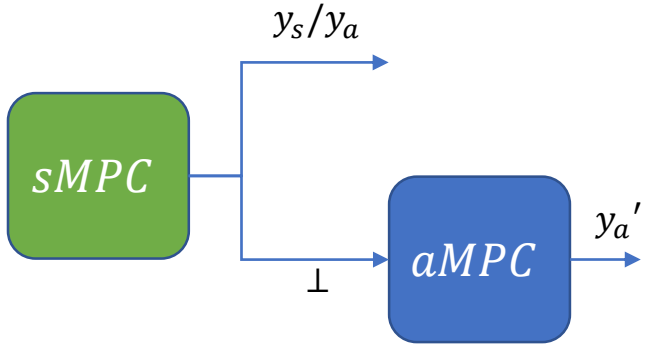
$d_i = DecShare_{dk_i}(c)$



$y_i = Rec(\{d_j\})$

Threshold Decryption

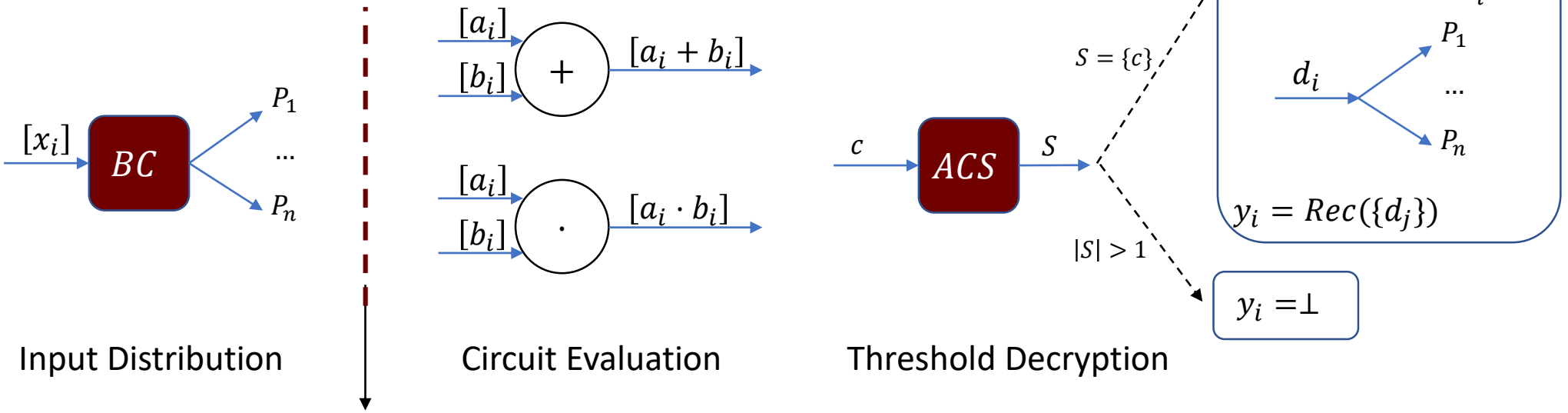
Feasibility



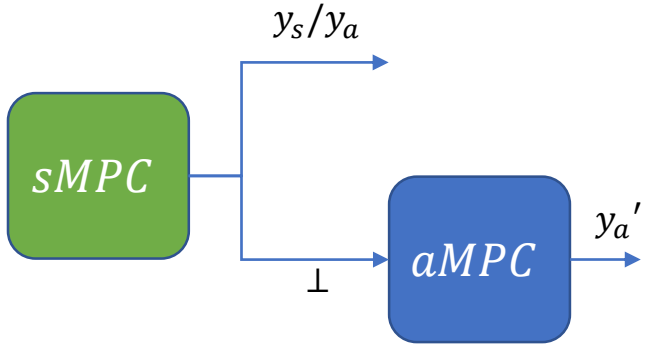
[CDN'01] approach

P_i:

$[x_i] = Enc_{ek}(x_i)$



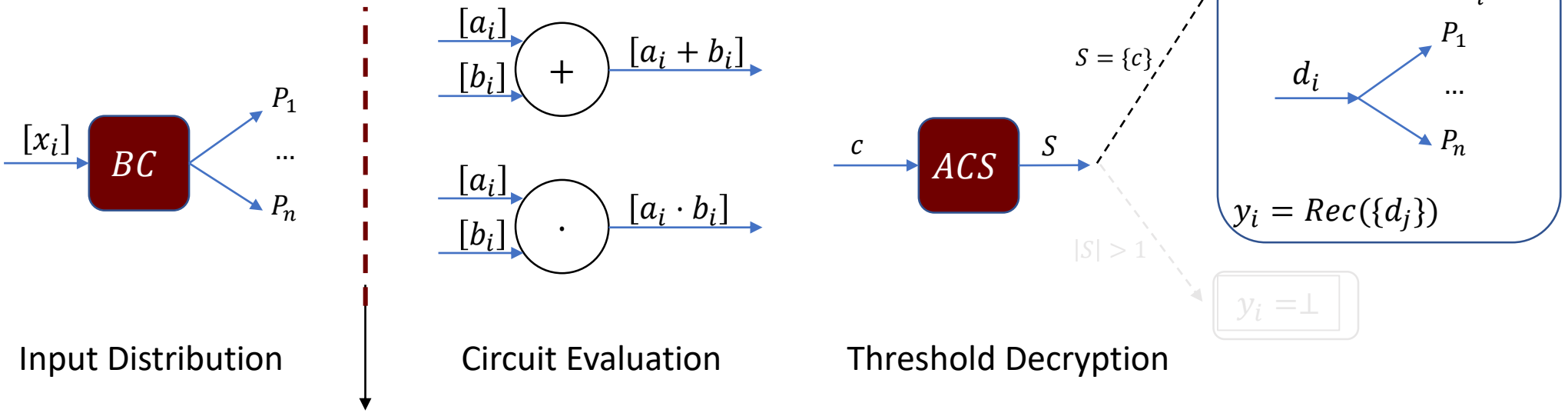
Feasibility



[CDN'01] approach

P_i:

$[x_i] = Enc_{ek}(x_i)$



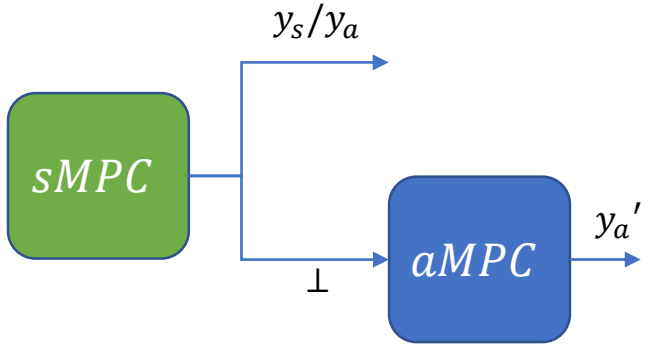
Input Distribution

Circuit Evaluation

Threshold Decryption

Wait for T time
Check if there are
$n - t_s$ inputs

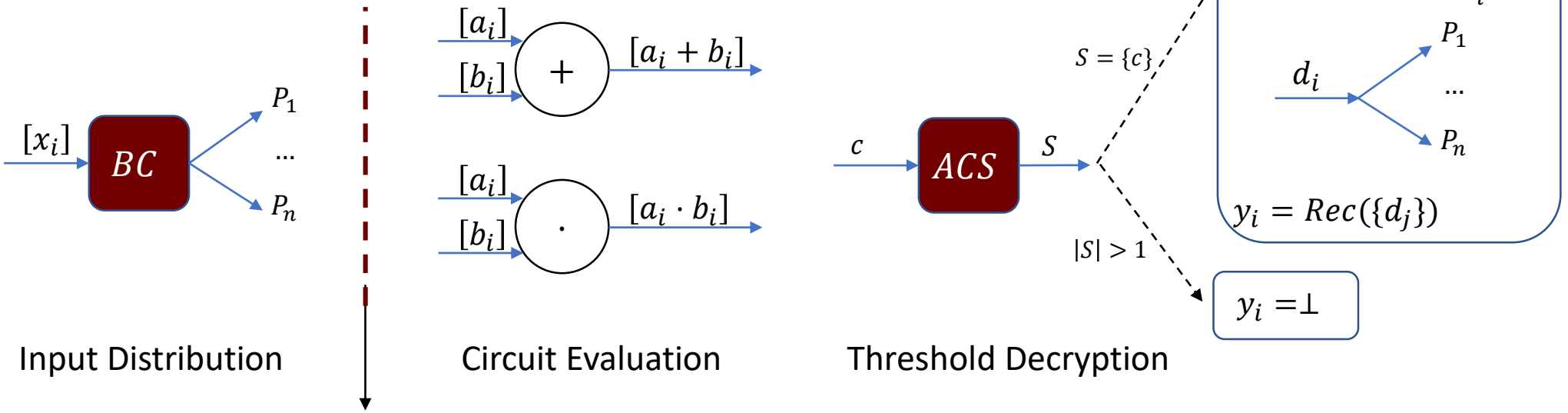
Feasibility



[CDN'01] approach

P_i:

$[x_i] = Enc_{ek}(x_i)$



Is there an MPC that achieves guarantees in both regimes?

Synchronous

$n/3 \leq t_s < n/2$

Input completeness

Asynchronous

$t_a < n/3$

Up to L inputs are ignored

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$L < t_s$ is impossible: t_s-security implies $L \geq t_s$

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$L < t_s$ is impossible: t_s-security implies $L \geq t_s$

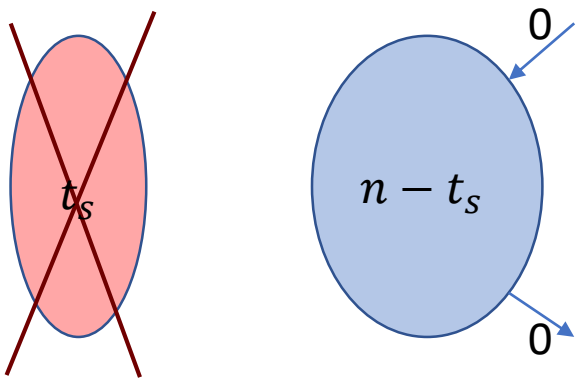
OR function

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$L < t_s$ is impossible: t_s-security implies $L \geq t_s$

OR function



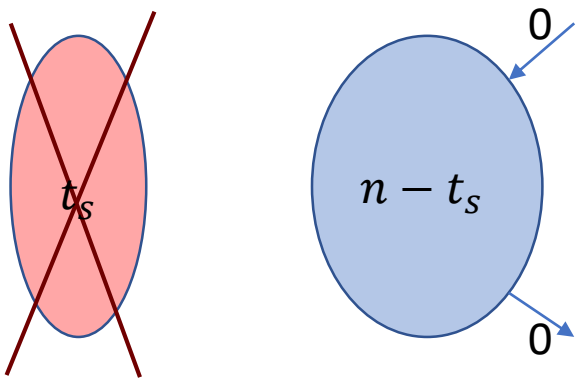
Synchronous

Impossibility

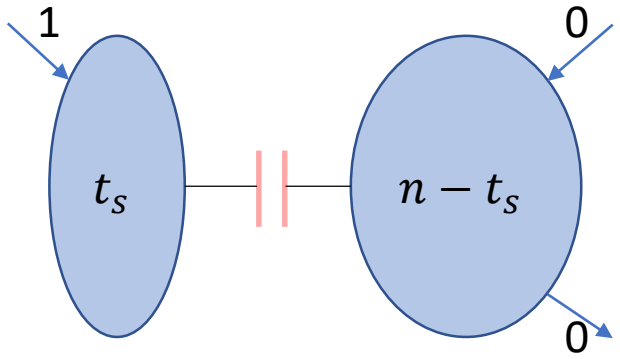
YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$L < t_s$ is impossible: t_s-security implies $L \geq t_s$

OR function



Synchronous



Asynchronous

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$t_a + 2t_s \geq n$ is impossible (even when all inputs are ignored) [BKL19]

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$t_a + 2t_s \geq n$ is impossible (even when all inputs are ignored) [BKL19]

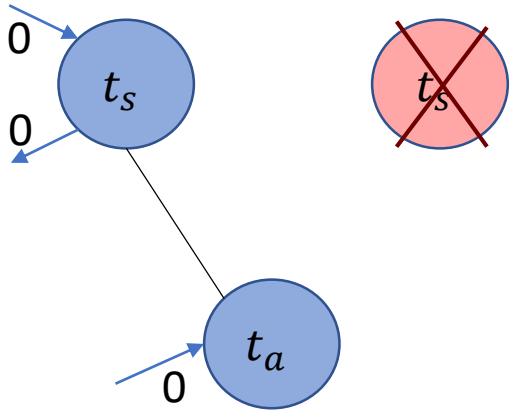
Majority function

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$t_a + 2t_s \geq n$ is impossible (even when all inputs are ignored) [BKL19]

Majority function

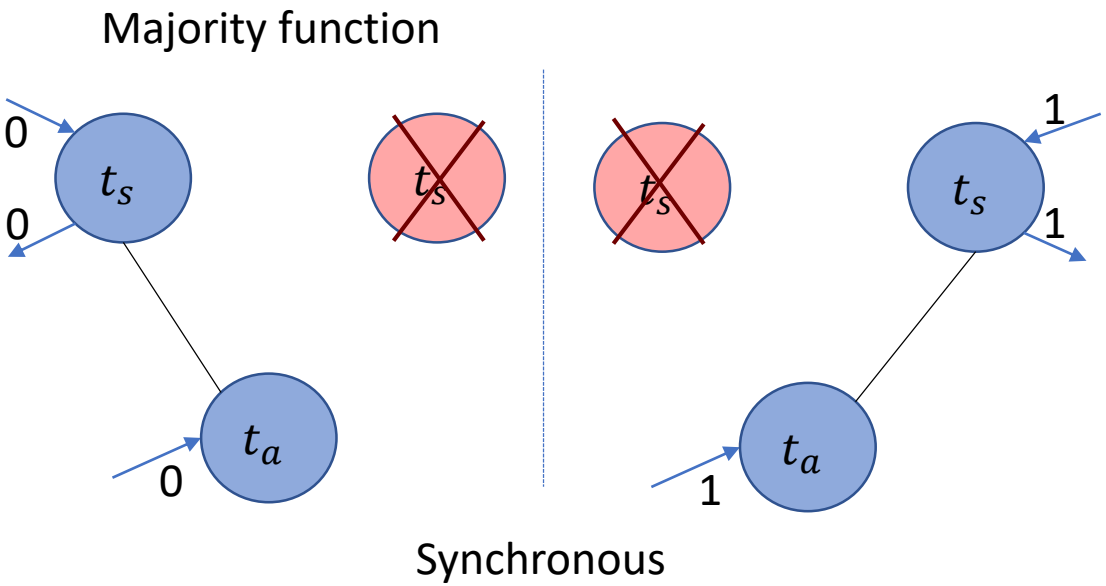


Synchronous

Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

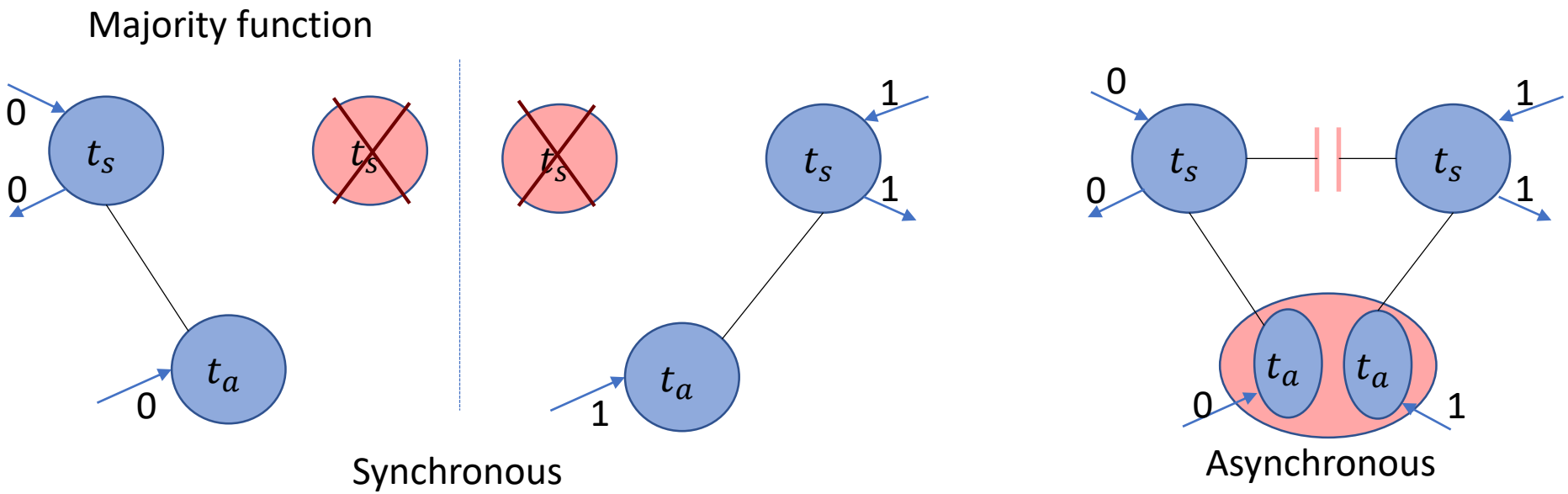
$t_a + 2t_s \geq n$ is impossible (even when all inputs are ignored) [BKL19]



Impossibility

YES! If and only if $t_a + 2t_s < n$ and $L \geq t_s$

$t_a + 2t_s \geq n$ is impossible (even when all inputs are ignored) [BKL19]



References and Credits

ETH Zurich, Department of Computer Science
lichen@inf.ethz.ch

References:

[BLL20]: Erica Blum, C. Liu-Zhang, J. Loss. Always Have a Backup Plan: Fully Secure Synchronous MPC with Asynchronous Fallback. **Full version:** https://eprint.iacr.org/2020/740

[DS83]: Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement, Journal on Computing 1983.

[BKR94]: Michael Ben-Or, Boaz Kelmer, and Tal Rabin. Asynchronous secure computations with optimal resilience, PODC'94.

[HNP05]: Martin Hirt, Jesper Buus Nielsen, Bartosz Przydatek. Cryptographic Asynchronous Multi-Party Computation with Optimal Resilience, EUROCRYPT'05.

[BKL19]: Erica Blum, Jonathan Katz, Julian Loss. Synchronous Consensus with Optimal Asynchronous Fallback Guarantees, TCC'19.

[BKL20]: Erica Blum, Jonathan Katz, Julian Loss. Network-Agnostic State-Machine Replication. https://eprint.iacr.org/2020/142.

Credits:

Icons: https://www.flaticon.com/