Two-Round Concurrent 2PC from Sub-Exponential LWE

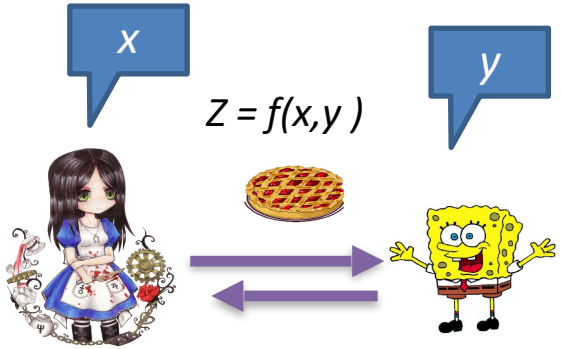
Behzad Abdolmaleki[1], Saikrishna Badrinarayanan[2], Rex Fernando[3], Giulio Malavolta[4,5], Ahmadreza Rahimi[5], and Amit Sahai[6]

1. University of Sheffield, UK
2. LinkedIn, USA
3. Carnegie Mellon University, USA
4. Bocconi University, Italy
5. Max Planck Institute for Security and Privacy, Germany
6. UCLA, USA

08.12.2023

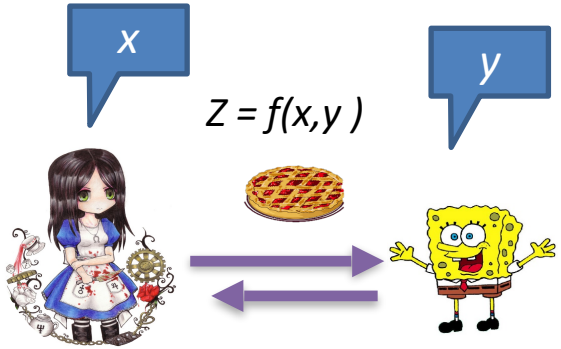
Problem Statement of Concurrent 2PC

-Def.:Two-party computation (2PC) protocols where both parties receive output z = f(x,y)



Problem Statement of Concurrent 2PC

-Def.:Two-party computation (2PC) protocols where both parties receive output z = f(x,y)



Security Goal:

- Adversary should learn nothing besides the output z.
- Formally: simulation-based security.

Concurrent Security:

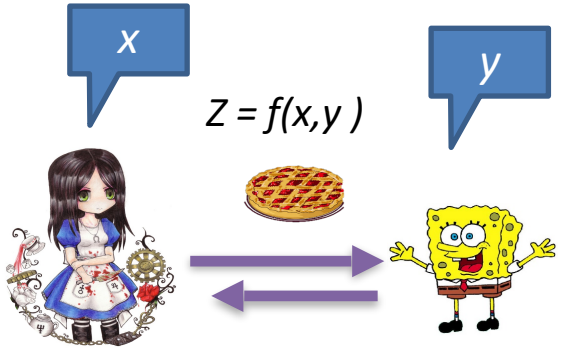
- Where adversary sees many instances of the protocol are executed in parallel.

Problem Statement of Concurrent 2PC

-Def.:Two-party computation (2PC) protocols where both parties receive output z = f(x,y)

Security Goal:

- Adversary should learn nothing besides the output z.
- Formally: simulation-based security.



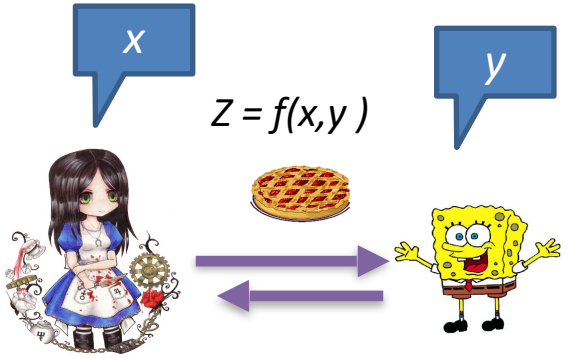
Concurrent Security:

- Where adversary sees many instances of the protocol are executed in parallel.

Can we achieve two-round concurrently secure two-party computation under simple, post-quantum assumptions, in the plain model?

The Concurrent Setting

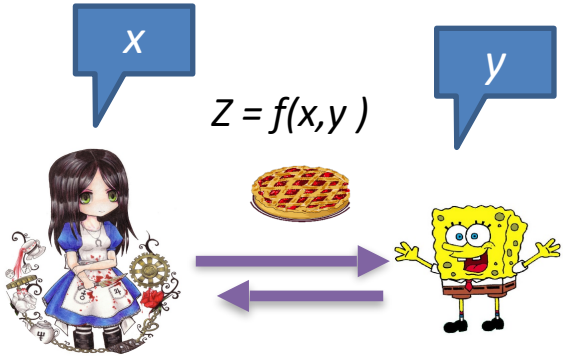
- A more realistic setting allows parties to participate concurrently in arbitrarily many instances.



Impossibility Result:

[BPS06]: Achieving the standard definition of concurrent security is impossible in any rounds in the plain model, without a trusted setup

The Concurrent Setting



- A more realistic setting allows parties to participate concurrently in arbitrarily many instances.

Impossibility Result:

[BPS06]: Achieving the standard definition of concurrent security is impossible in any rounds in the plain model, without a trusted setup

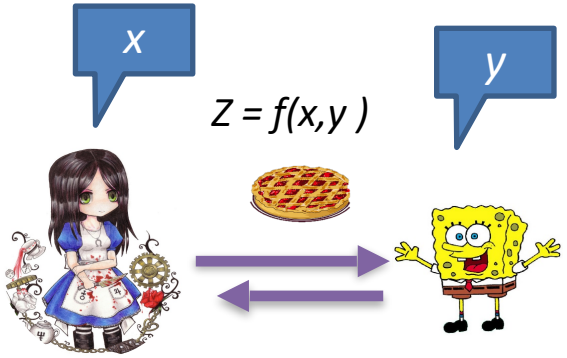
Overcome the above mentioned impossibility results:

- The bounded concurrent model [Pass04],
- In the multiple ideal-query model [GoyJai13],
- input-indistinguishable computation [MicPas06].

-And an standard relaxation of simulation security: the notion of super-polynomial simulation, or SPS [Pass03]. (which is widely used to circumvent many lower-bound results)

The Concurrent Setting

- A more realistic setting allows parties to participate concurrently in arbitrarily many instances.



Impossibility Result:

[BPS06]: Achieving the standard definition of concurrent security is impossible in any rounds in the plain model, without a trusted setup

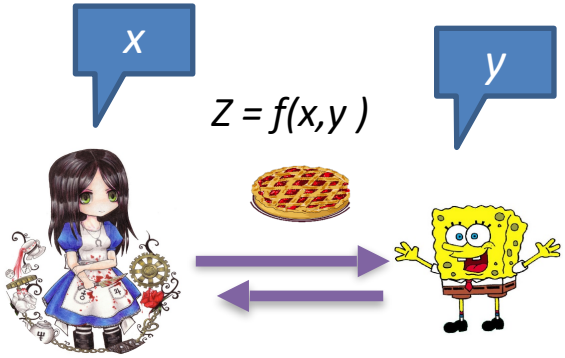
Overcome the above mentioned impossibility results:

- The bounded concurrent model [Pass04],
- In the multiple ideal-query model [GoyJai13],
- input-indistinguishable computation [MicPas06].

-And an standard relaxation of simulation security: the notion of super-polynomial simulation, or SPS [Pass03]. (which is widely used to circumvent many lower-bound results)

The Round Complexity

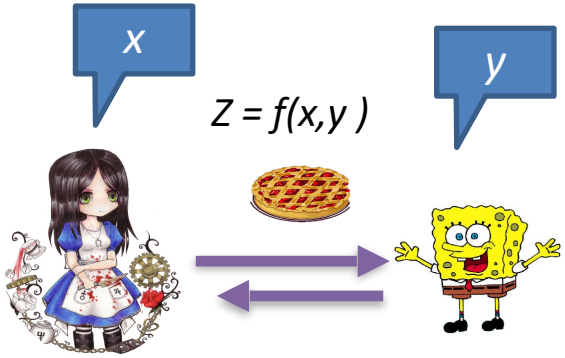
- In 2PC (with two-side output) setting, a round is defined in the simultaneous exchange message model



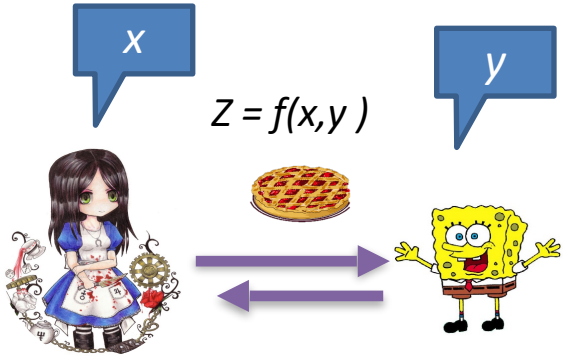
The Round Complexity

- In 2PC (with two-side output) setting, a round is defined in the simultaneous exchange message model

- In every round, two parties can simultaneously send the next round message to each other parties



The Round Complexity



- In 2PC (with two-side output) setting, a round is defined in the simultaneous exchange message model

- In every round, two parties can simultaneously send the next round message to each other parties

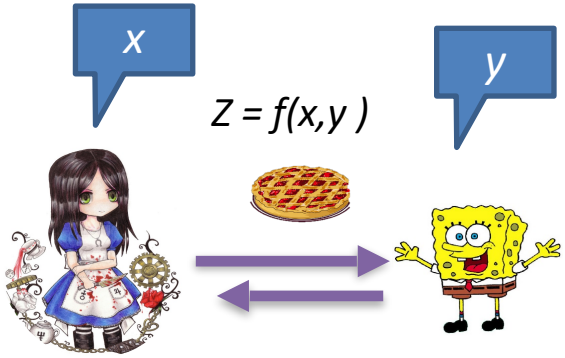
Time line and Results:

[GGJS12, KMO14]: Constant-round protocols (approximately 20 rounds).

[GKP17] : 5 rounds with SPS security from standard sub-exponential assumptions.

[BGJKS17]: Concurrent MPC in four-round with SPS security.

The Round Complexity



- In 2PC (with two-side output) setting, a round is defined in the simultaneous exchange message model

- In every round, two parties can simultaneously send the next round message to each other parties

Time line and Results:

[GGJS12, KMO14]: Constant-round protocols (approximately 20 rounds).

[GKP17] : 5 rounds with SPS security from standard sub-exponential assumptions.

[BGJKS17]: Concurrent MPC in four-round with SPS security.

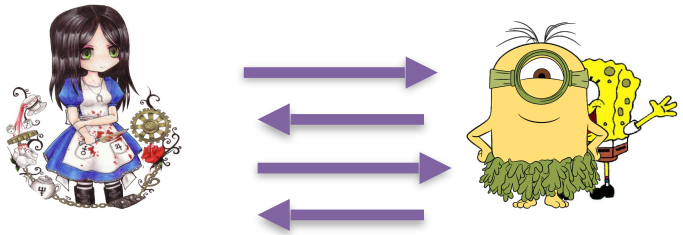
[ABGKM21] : Two-round MPC with standalone security in the plain model assuming subexponential NIWI arguments, the subexponential SXDH assumption, and the existence of non-interactive NMC

[FJK22]: Concurrent two-round MPC protocol, assuming subexponential quantum hardness of LWE, subexponential classical hardness of SXDH, the existence of a subexponentially-secure (classically-hard) iO, and time-lock puzzles

Standard Simulation-Real Ideal Paradigm

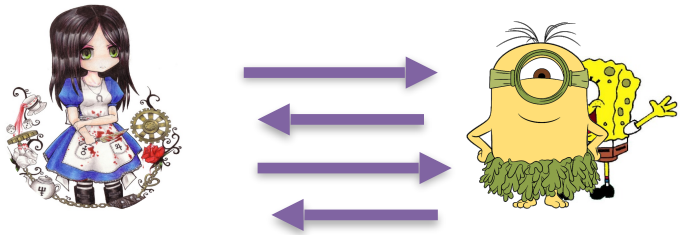
Real

Ideal

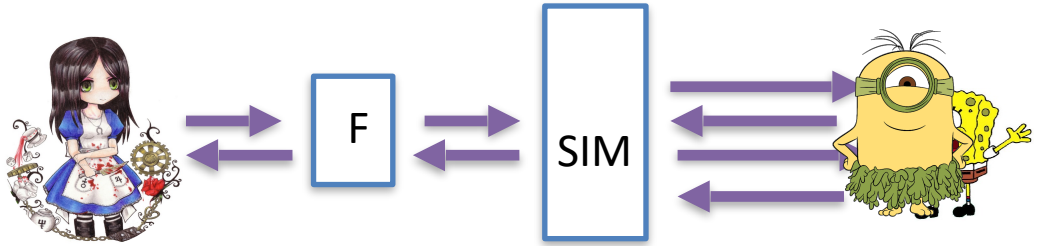


Standard Simulation-Real Ideal Paradigm

Real



Ideal

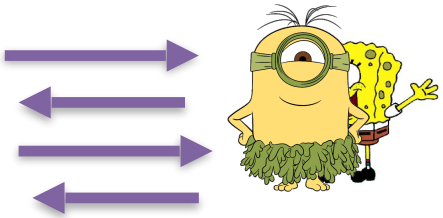


Standard Simulation-Real Ideal Paradigm

Real

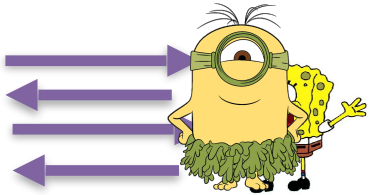
Ideal

SIM runs in Poly time



F

SIM

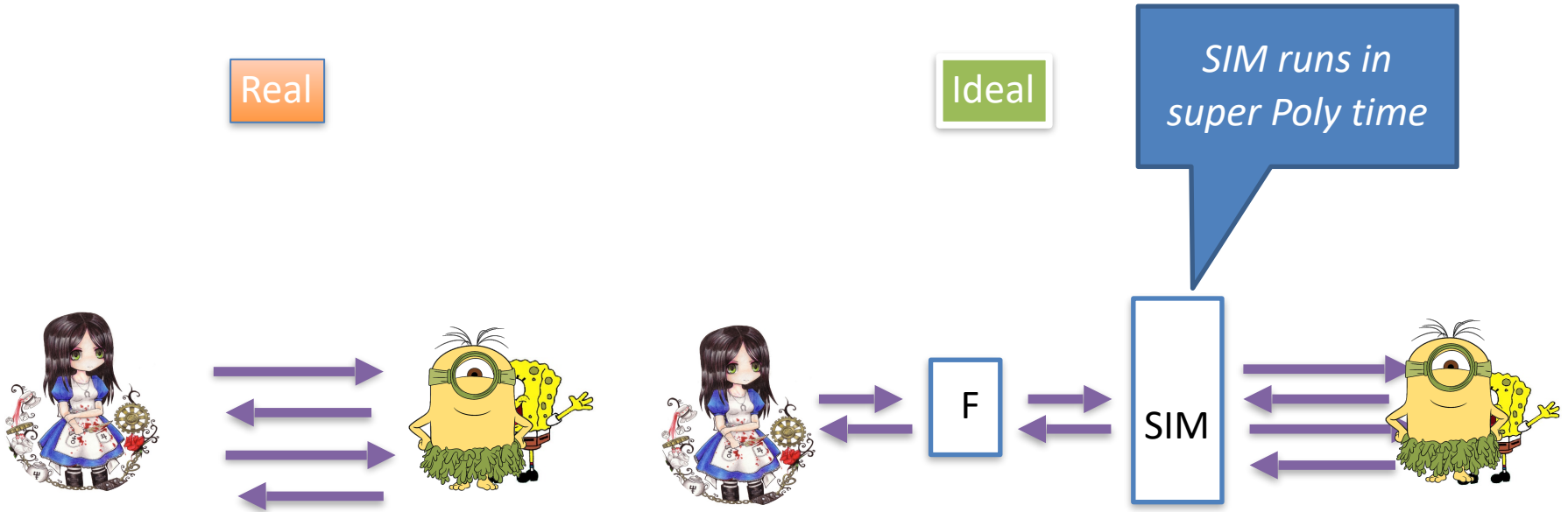


Real

Computationally Indistinguishable

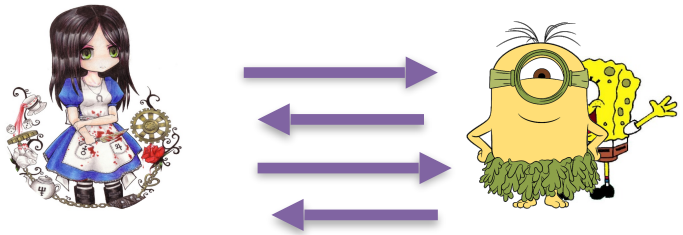
Ideal

SPS Simulation Paradigm [Pass03, PS04, BS05, BGJKS17]

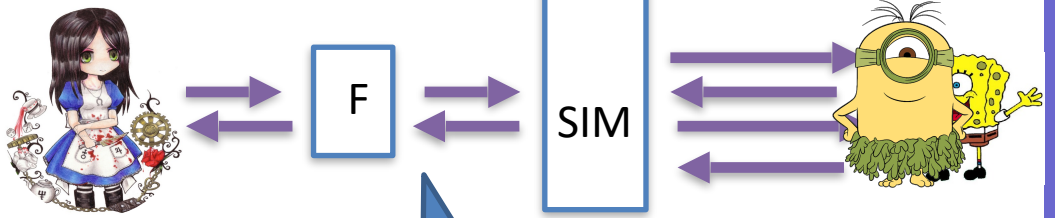


SPS Simulation Paradigm [Pass03, PS04, BS05, BGJKS17]

Real



Ideal



SIM runs in super Poly time

But F is secure even against super poly adversary

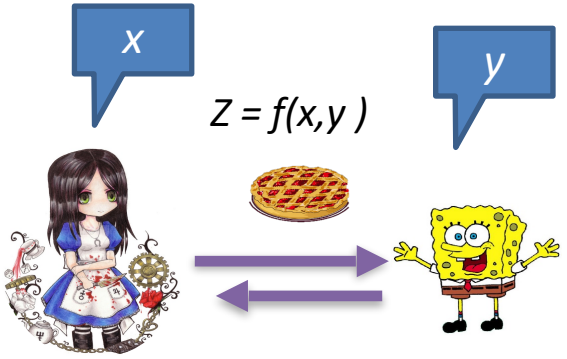
Real

Computationally Indistinguishable

Ideal

Our Results

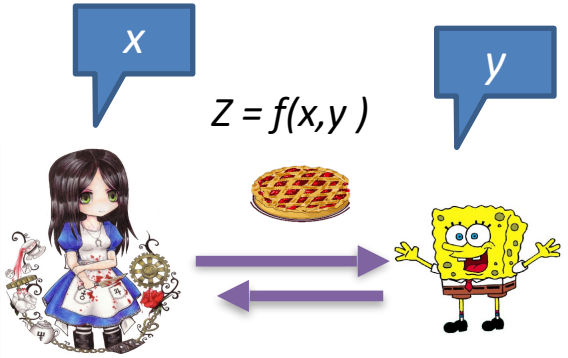
Main Result:



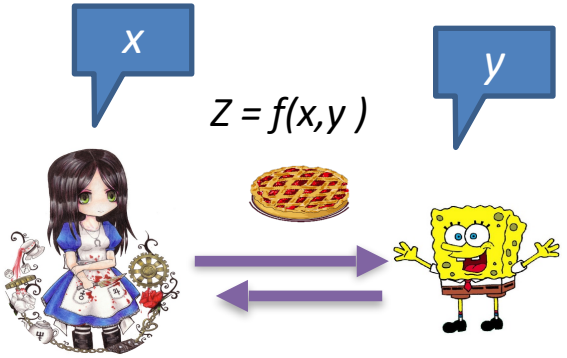
Our Results

Main Result:

A two-round, concurrent-secure, two-party secure computation based on a single, standard, post-quantum assumption, namely sub-exponential the hardness of LWE problem.



Our Results



Main Result:

A two-round, concurrent-secure, two-party secure computation based on a single, standard, post-quantum assumption, namely sub-exponential the hardness of LWE problem.

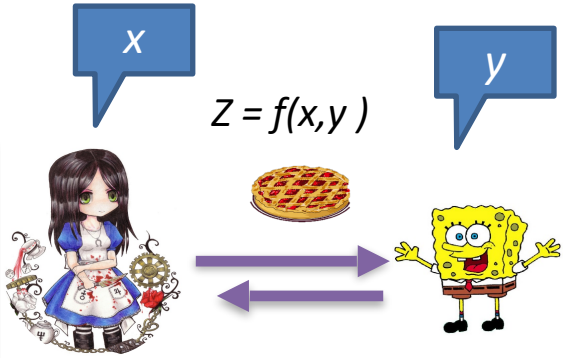
First two-round concurrent-secure 2PC thaty does not require:

- The existence of a one-round NMC. Instead, we are able to use the two-round NMCs of [KhuSah17], which is instantiable from sub-exponential LWE.
- The existence of non-interactive witness indistinguishable arguments or time-lock puzzles.

Our Results

Main Result:

A two-round, concurrent-secure, two-party secure computation based on a single, standard, post-quantum assumption, namely sub-exponential the hardness of LWE problem.



First two-round concurrent-secure 2PC thaty does not require:

- The existence of a one-round NMC. Instead, we are able to use the two-round NMCs of [KhuSah17], which is instantiable from sub-exponential LWE.
- The existence of non-interactive witness indistinguishable arguments or time-lock puzzles.

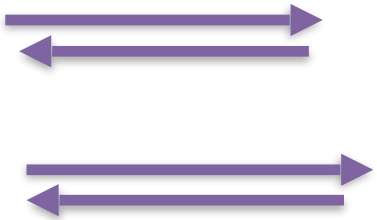
The Applications:

1) The first two-round PAKE scheme in the plain model, resolving a long-standing open problem in the area

2) The first concurrent 2PC for quantum functionalities (in the plain model) with classical inputs and outputs

Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability

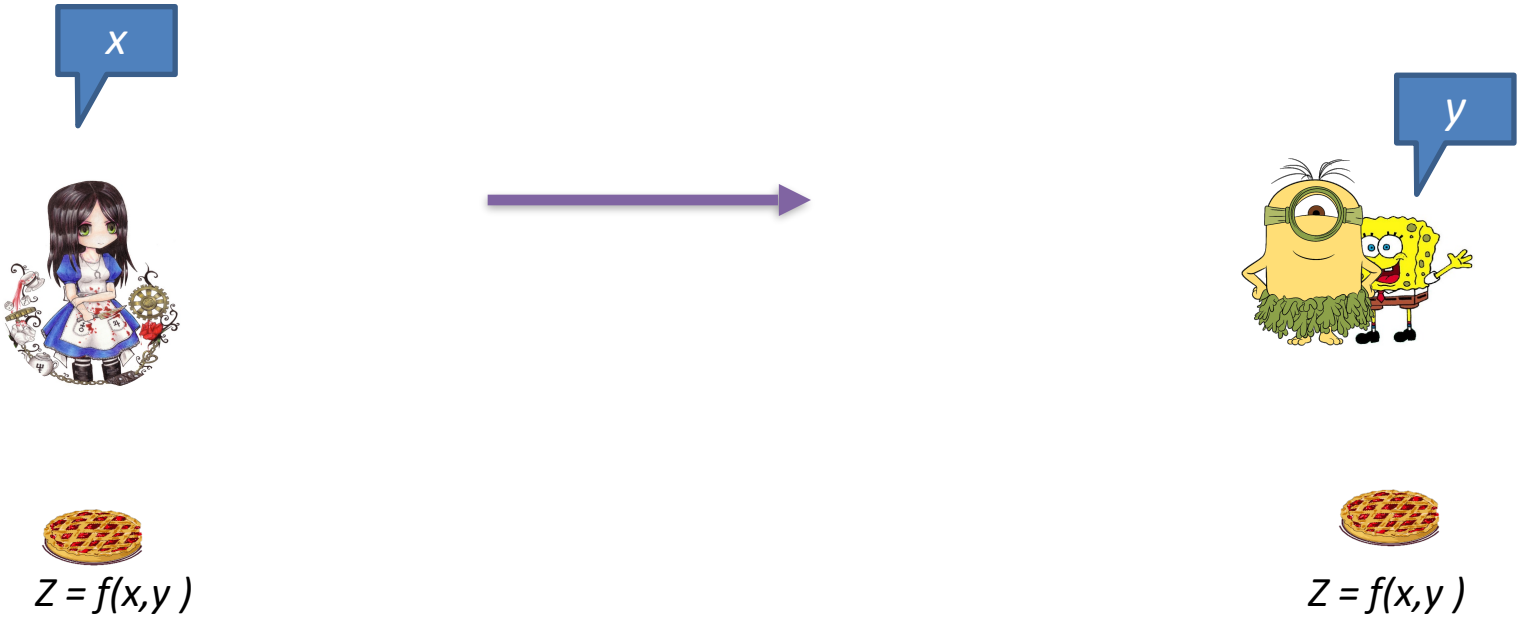


$Z = f(x,y)$

$Z = f(x,y)$

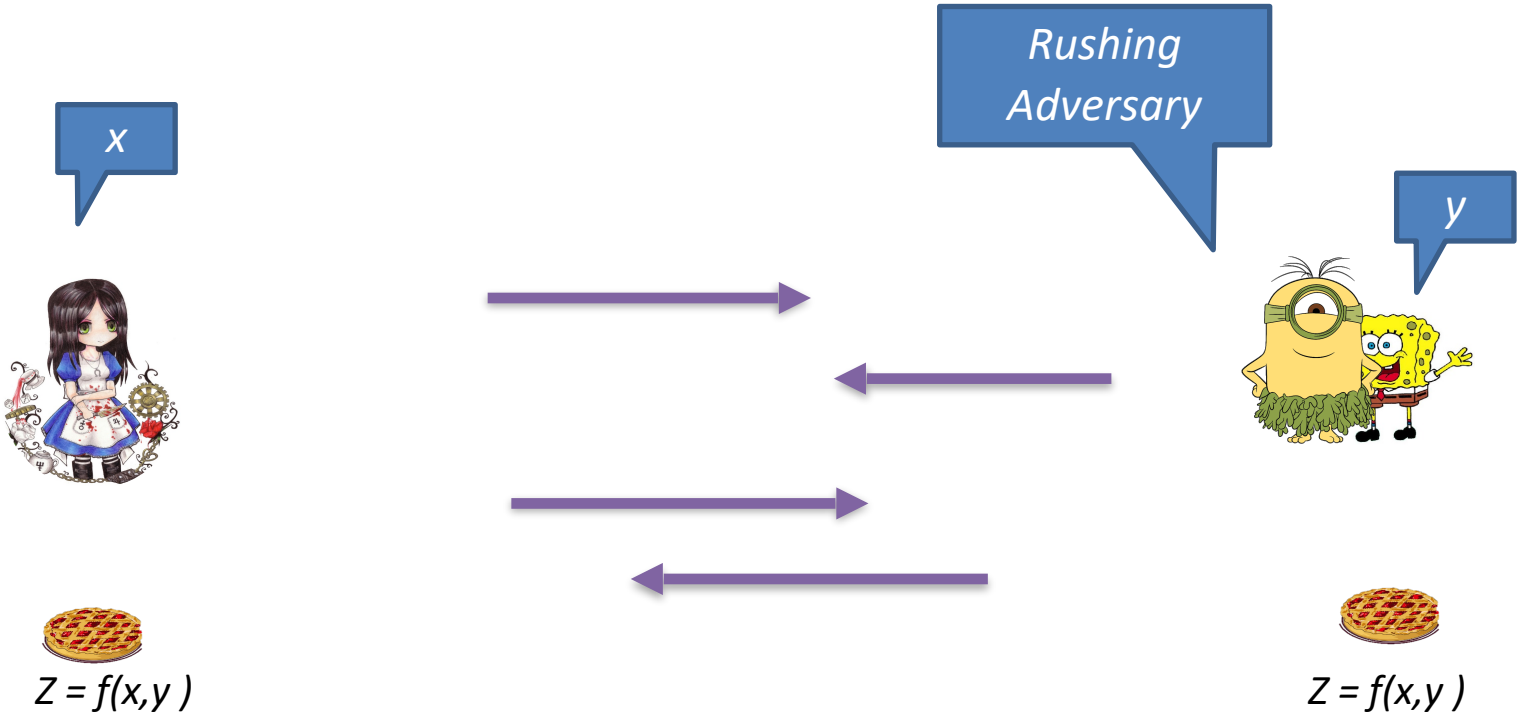
Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability



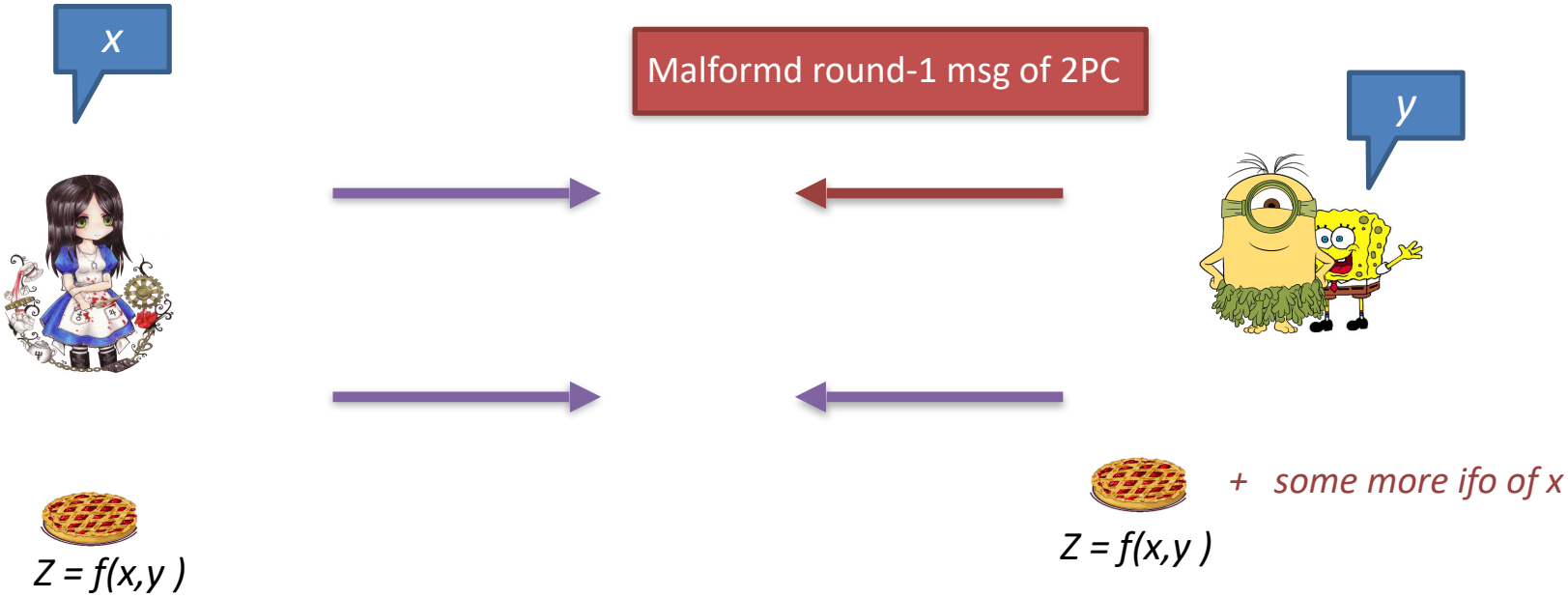
Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability



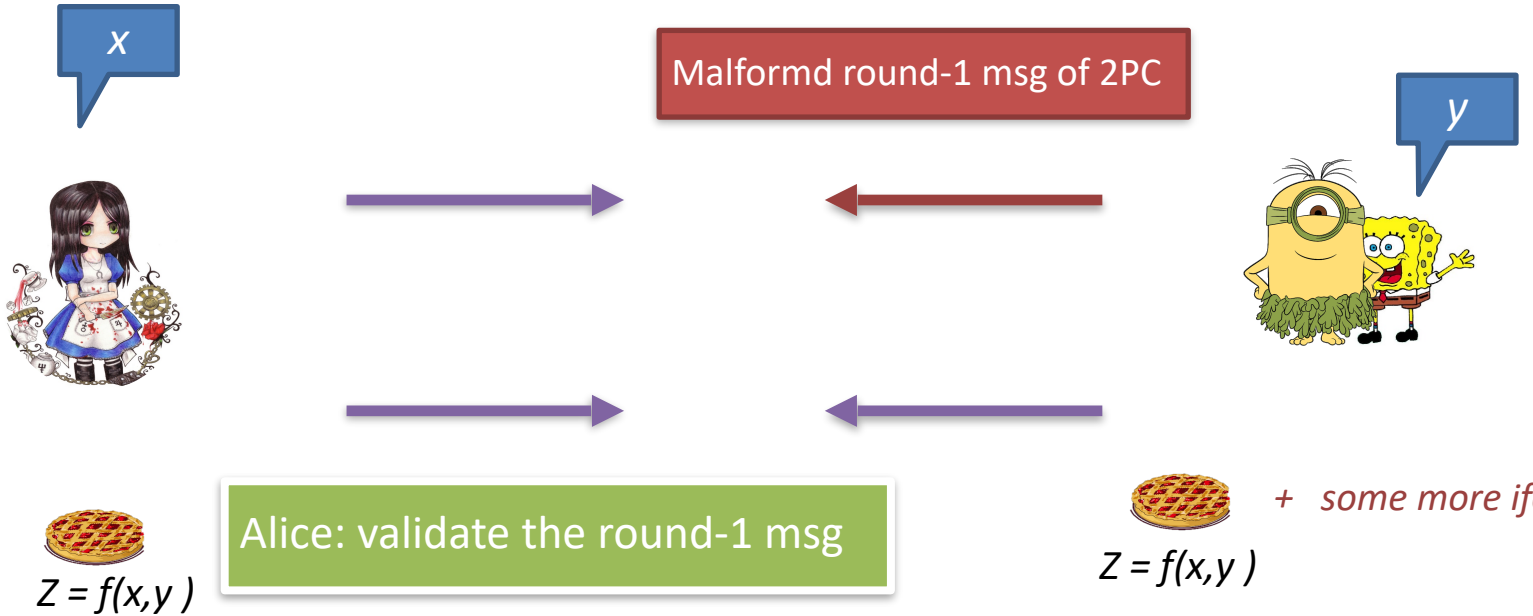
Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability



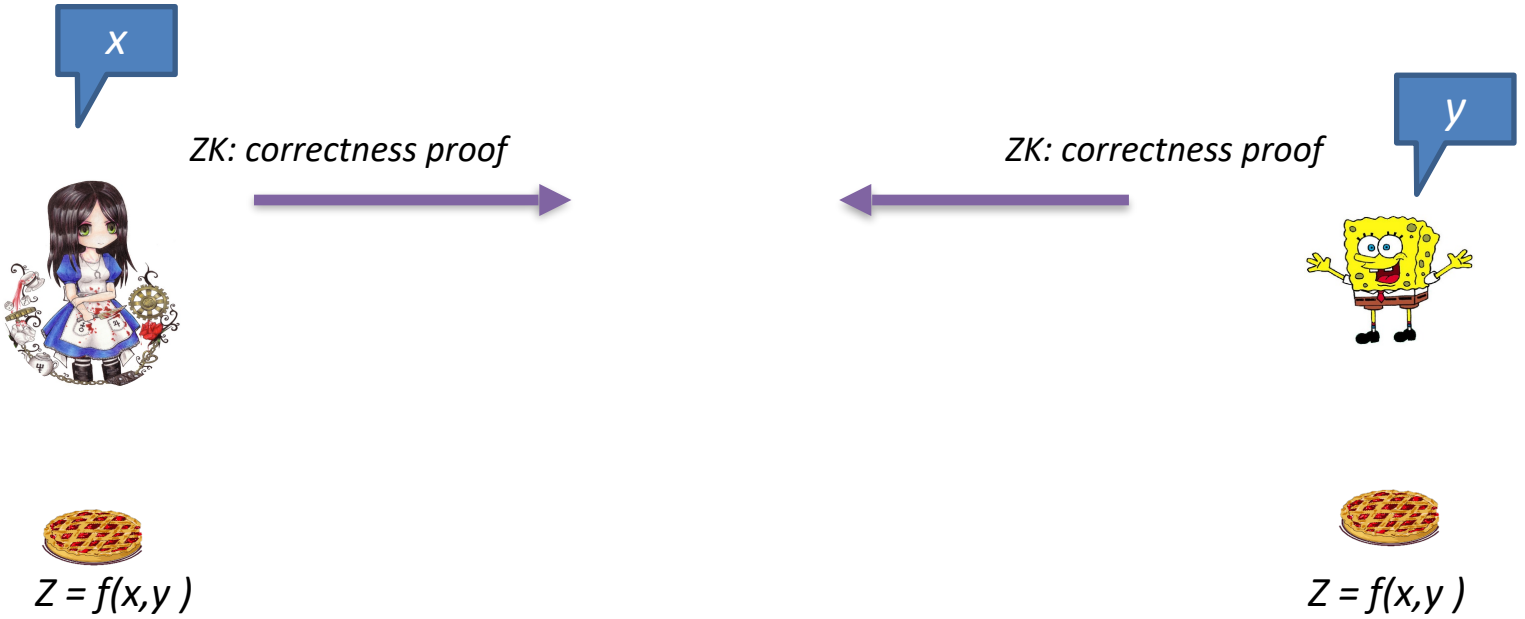
Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability



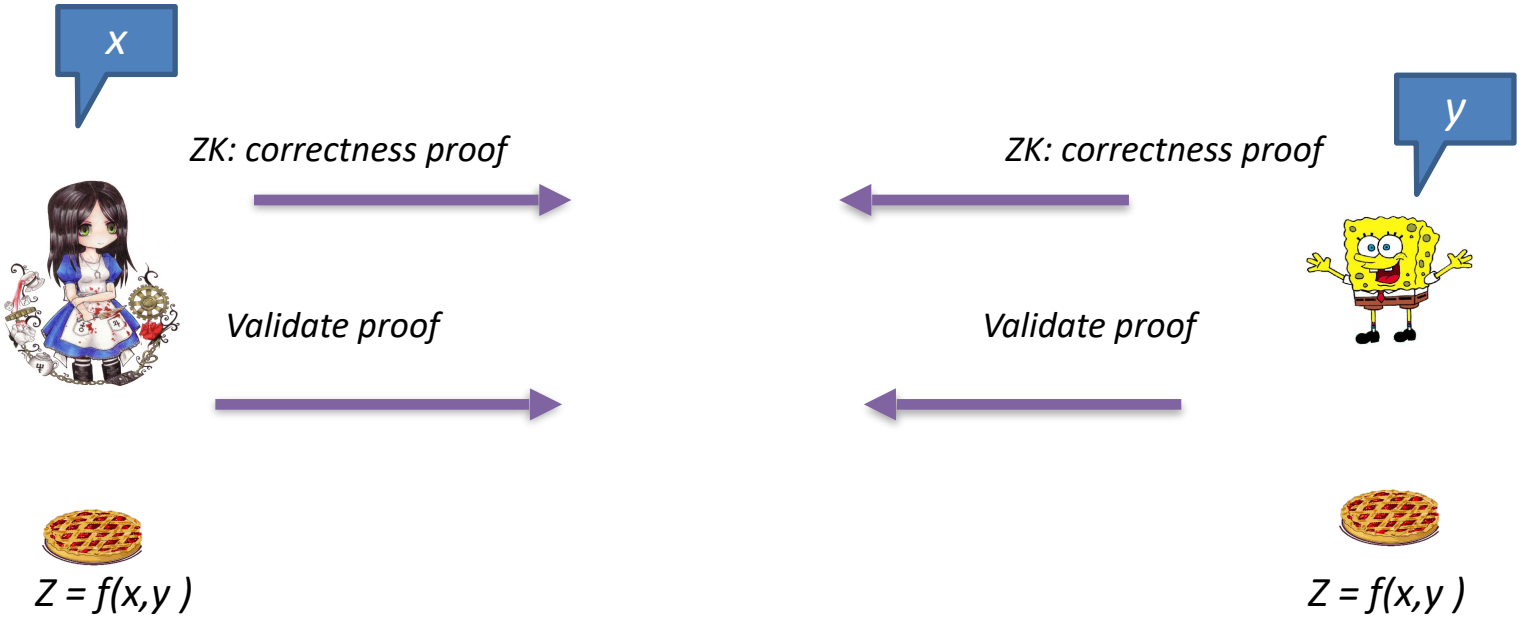
Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability



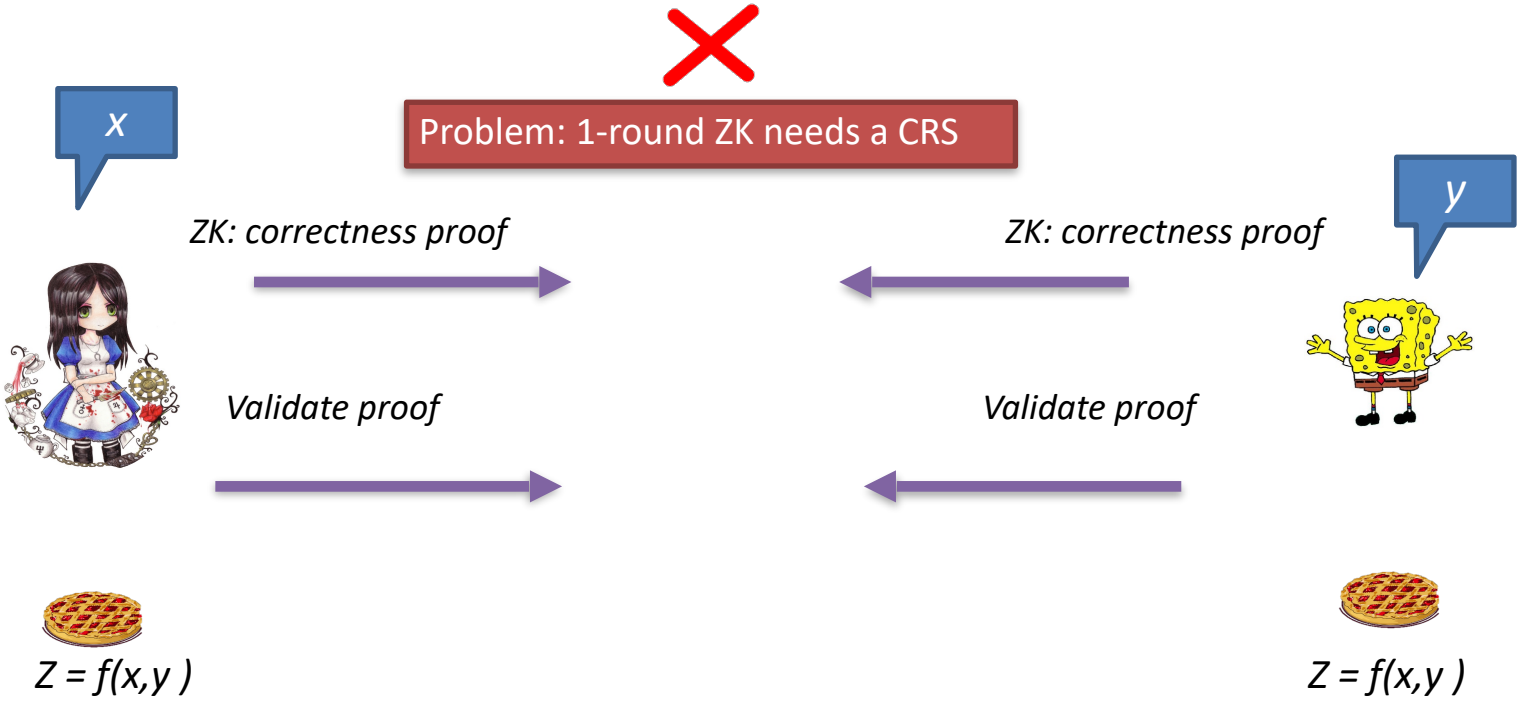
Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability



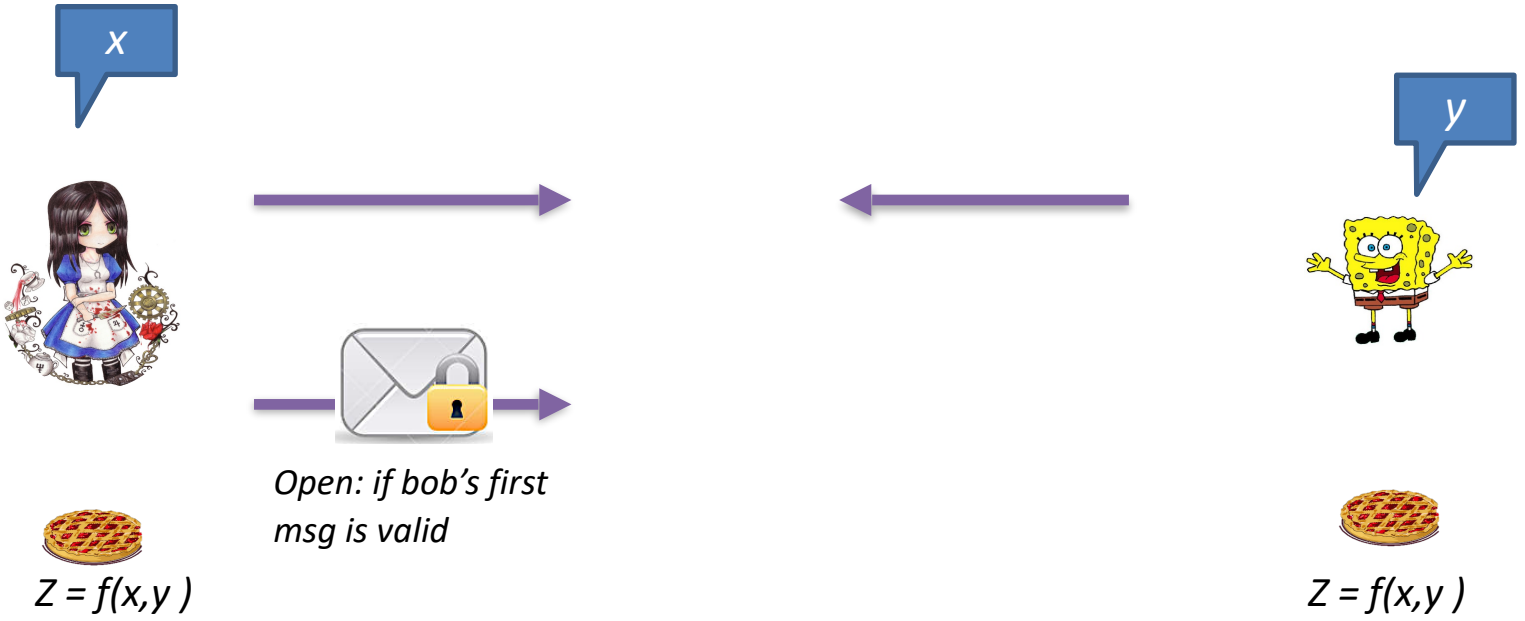
Observarion: The 2PC Construction

Observarion from [ABGKM21] -The need for Verifiability



Intuition: Alternative Approach

Observarion from [ABGKM21] -The need for Verifiability



Intuition: Alternative Approach

Observarion from [ABGKM21] -The need for Verifiability

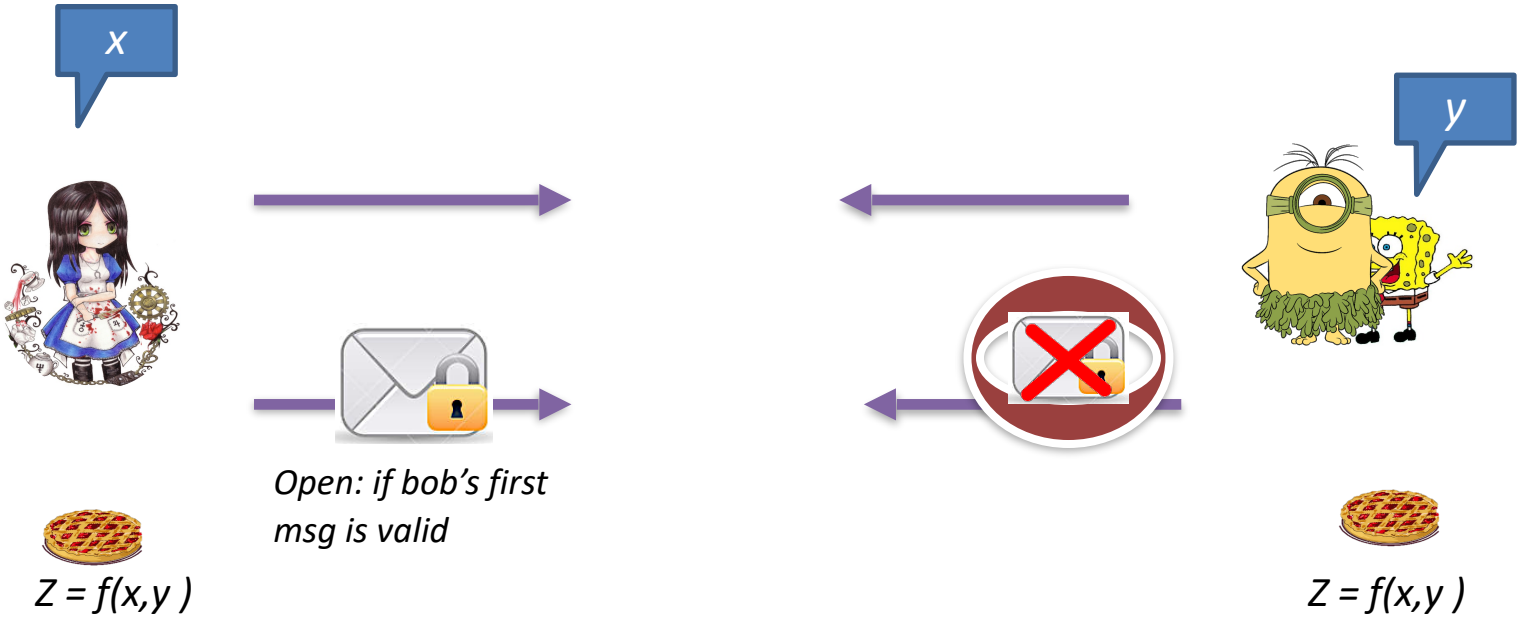
Open: if bob's first msg is valid

Z = f(x,y)

Z = f(x,y)

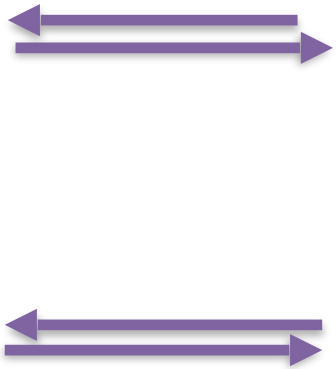
Intuition: Alternative Approach

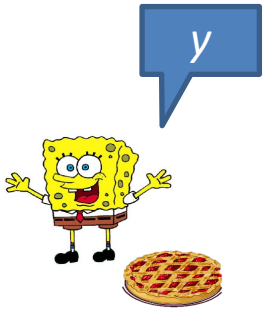
Observarion from [ABGKM21] -The need for Verifiability



Our Technique/Construction

Four main tools in our construction:





$Z = f(x,y)$

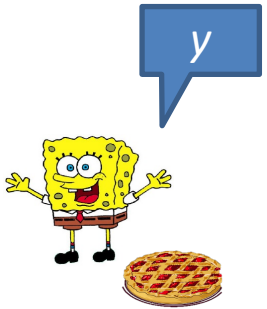
$Z = f(x,y)$

Our Technique/Construction

Four main tools in our construction:

a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

P_i: Two different types of commitments to its input, (NMC1 and OT1 message).



$Z = f(x,y)$

$Z = f(x,y)$

Our Technique/Construction

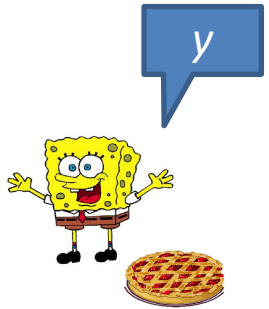
Four main tools in our construction:

a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

the OT1 message will be used by party P_i in reconstructing its own output

NMC1 will be used to help P1–i to reconstruct its output

P_i: Two different types of commitments to its input, (NMC1 and OT1 message).



$Z = f(x,y)$

$Z = f(x,y)$

Our Technique/Construction

Four main tools in our construction:

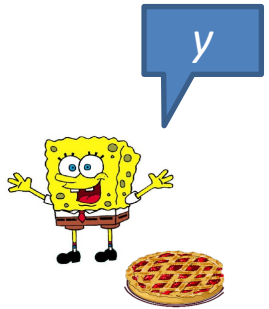
a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

the OT1 message will be used by party P_i in reconstructing its own output

NMC :To prevent the honest party from learning “mauled” outputs

NMC will be used to help P1–i to reconstruct its output

P_i: Two different types of commitments to its input, (NMC and OT1 message).



$Z = f(x,y)$

$Z = f(x,y)$

Our Technique/Construction

Four main tools in our construction:

a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

the OT1 message will be used by party P_i in reconstructing its own output

NMC :To prevent the honest party from learning "mauled" outputs

NMC1 will be used to help P1–i to reconstruct its output

P_i: Two different types of commitments to its input, (NMC1 and OT1 message).

x

Z = f(x,y)

How do we prevent the adversary from learning f(x, y), then?

y

Z = f(x,y)

Our Technique/Construction

Four main tools in our construction:

a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

the OT1 message will be used by party P_i in reconstructing its own output

NMC :To prevent the honest party from learning "mauled" outputs

NMC1 will be used to help P1–i to reconstruct its output

P_i: Two different types of commitments to its input, (NMC1 and OT1 message).

x

$Z = f(x,y)$

How do we prevent the adversary from learning f(x, y), then?

The SSP OT: an adversary can only unlock the protocol output if it knows the input of its OT1 message.

y

$Z = f(x,y)$

Our Technique/Construction

Four main tools in our construction:

a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

the OT1 message will be used by party P_i in reconstructing its own output

NMC :To prevent the honest party from learning "mauled" outputs

NMC1 will be used to help P1–i to reconstruct its output

P_i: Two different types of commitments to its input, (NMC1 and OT1 message).

x

$Z = f(x,y)$

How do we prevent the adversary from learning f(x, y), then?

The SSP OT: an adversary can only unlock the protocol output if it knows the input of its OT1 message.

Are we done?

y

$Z = f(x,y)$

Our Technique/Construction

Four main tools in our construction:

a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

Z = f(x,y)

x

P_i: Two different types of commitments to its input, (NMC1 and OT1 message).

How do we prevent the adversary from learning f(x, y), then?

The SSP OT: an adversary can only unlock the protocol output if it knows the input of its OT1 message.

Z = f(x',y)

y

Our Technique/Construction

Four main tools in our construction:

a two-round NMC | a two-round SSP OT, | a two-round strong SPS zero-knowledge | garbled circuits

we must somehow connect the NMC1 with the OT1

P_i: Two different types of commitments to its input, (NMC1 and OT1 message).

x

$Z = f(x,y)$

How do we prevent the adversary from learning f(x, y), then?

The SSP OT: an adversary can only unlock the protocol output if it knows the input of its OT1 message.

y

$Z = f(x,y)$

Our Technique/Construction

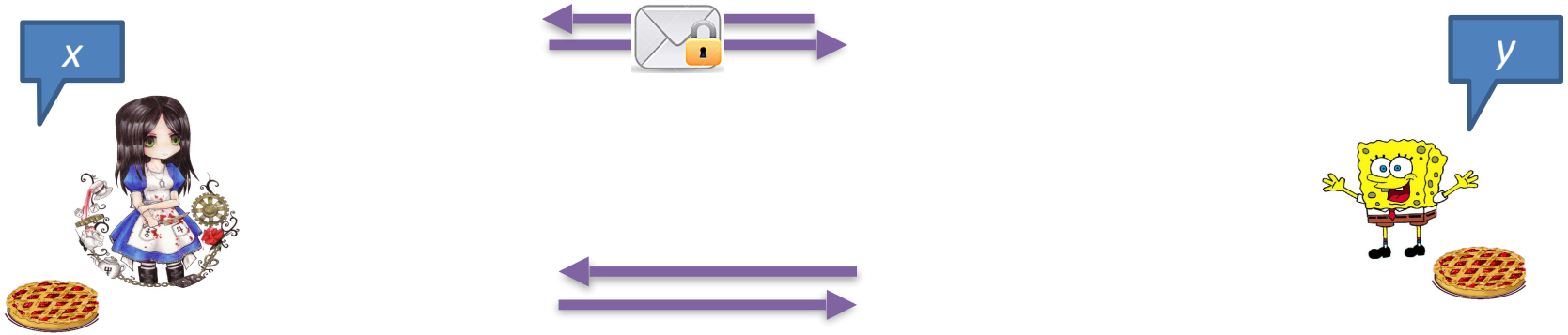
Four main tools in our construction:

A two-round NMC | Atwo-round SSP OT, | A two-round strong SPS zero-knowledge | Garbled Circuits

P_i: Two different types of commitments to its input: NMC1 and OT1 message).

+

P_i: Commit to the randomness used for its NMC1



Our Technique/Construction

Four main tools in our construction:

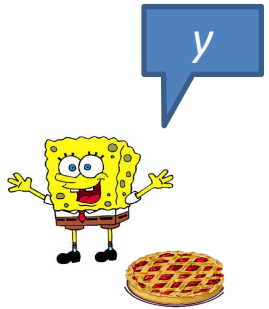
A two-round NMC | Atwo-round SSP OT, | A two-round strong SPS zero-knowledge | Garbled Circuits

P_i: Two different types of commitments to its input: NMC1 and OT1 message).

+

P_i: Commit to the randomness used for its NMC1

P_i: NMC2 , OT2



$Z = f(x,y)$

$Z = f(x,y)$

Our Technique/Construction

Four main tools in our construction:

A two-round NMC | Atwo-round SSP OT, | A two-round strong SPS zero-knowledge | Garbled Circuits

P1−i: can construct its garbled circuit to only reveal the output if this randomness is correct.

P_i: Two different types of commitments to its input: NMC1 and OT1 message). + **P_i: Commit to the randomness used for its NMC1**

x

P_i: NMC2 , OT2, Garbled Circuit C, and SPS.ZK.

y

$Z = f(x,y)$

$Z = f(x,y)$

Our Technique/Construction

Four main tools in our construction:

A two-round NMC | Atwo-round SSP OT, | A two-round strong SPS zero-knowledge | Garbled Circuits

P1–i: can construct its garbled circuit to only reveal the output if this randomness is correct.

P_i: Two different types of commitments to its input: NMC1 and OT1 message). + **P_i: Commit to the randomness used for its NMC1**

x

P_i: NMC2 , OT2, Garbled Circuit C, and SPS.ZK.

Done!

y

$Z = f(x,y)$

$Z = f(x,y)$

Thanks