Quantum Universal Composability

Michael Ben-Or and Dominic Mayers

Overview

- Why composability in quantum cryptography: few examples including Relativistic Bit Commitment.

- Real and Ideal Models for Universal Security Definition and Universal Composability (adapted to the quantum world).

- Universal Composability Theorem

- Connection with an achievable security definition for bit commitment (the one obtained with relativistic protocols and other protocols).

Some examples where composability would be useful

- One time pad on top of KD [Shannon] versus QKD [Mayers 96, and others].

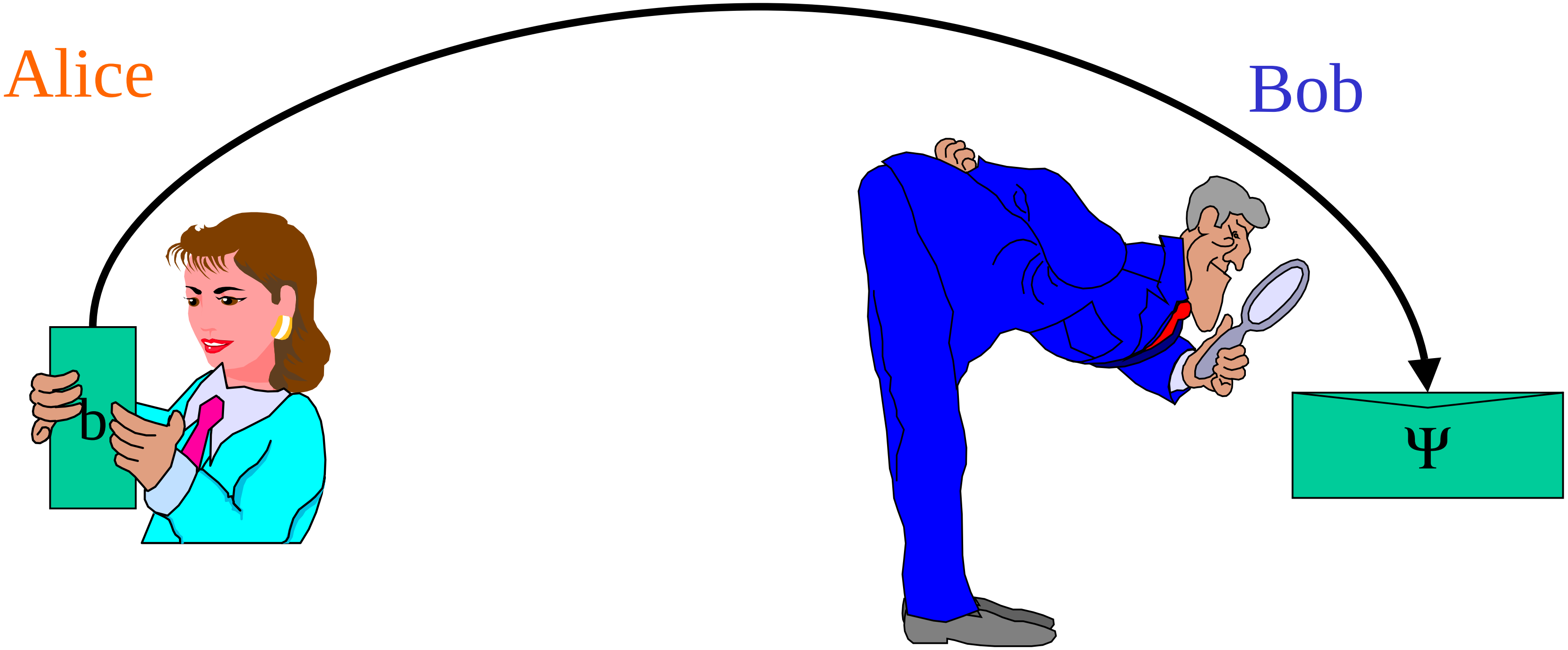
- Oblivious Transfer on top of (ideal) Bit Commitment [Yao 95] versus computationally secure quantum bit commitment [Dumais, Mayers, Salvail, 2000].

- Weak Bit Commitment on top of Coin Flipping [Ahoronov, Tashma, Vazirani, Yao, 1999] [Kent and Hardi, 1999] versus Ambainis protocol for coin flipping.

- Bit Commitment with equality on top of ordinary bit commitment [Rudich, Bennett] [Kent 1999] versus Temporary Relativistic Bit Commitment [Mayers 2002]. (Our running example)

Temporary Relativistic Bit Commitment

First we recall Bit Commitment



Concealing: The blob Ψ give no information about b to Bob.

Binding: Alice cannot change her mind.

Etc…. (Clarifying this etc. is part of the problem).

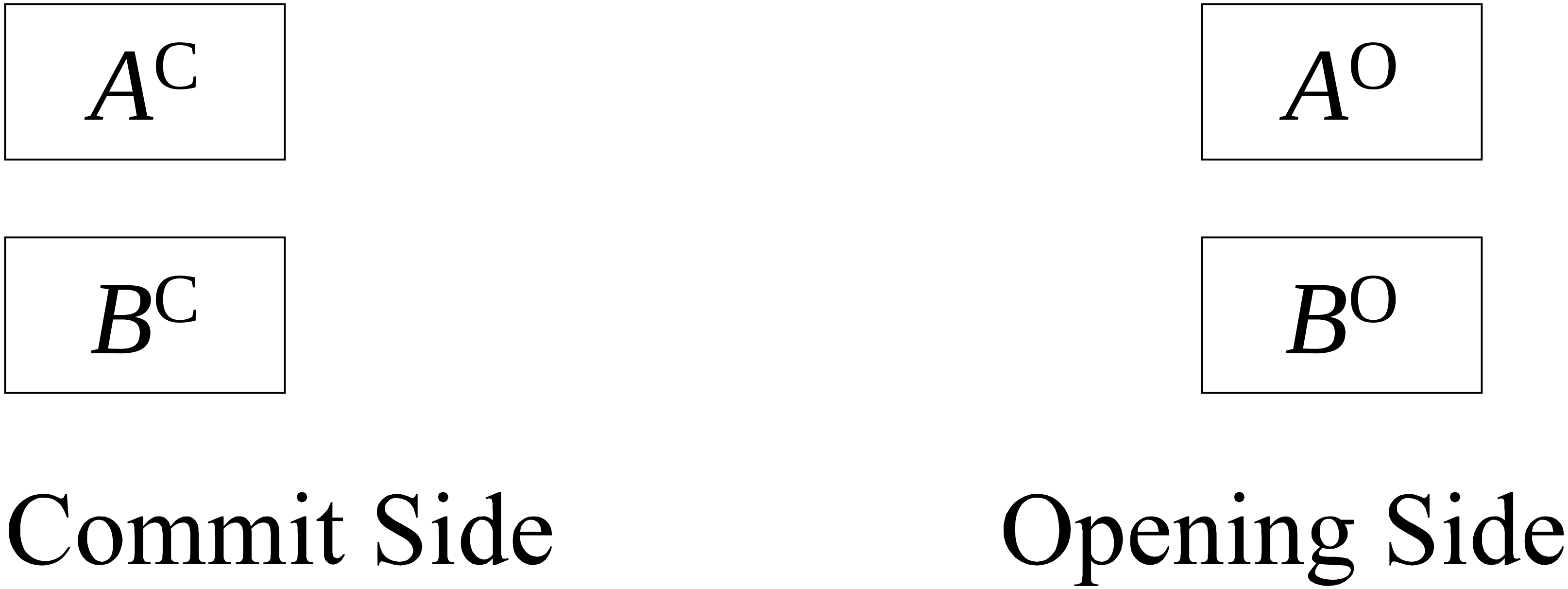
Historic on Relativistic Bit Commitment

Temporary Relativistic Bit Commitment is a relativistic variation on the two-prover protocol of Ben-Or, Goldwasser, Kilian and Wigderson (1988).

The relativistic variation was analysed by Brassard, Crépeau, Mayers and Salvail (1998) to show that it was not a permanent bit commitment.

Kent`s salient idea (1999) was to realise that it can be used as a building block for bit commitment with equality (Brassard, Crépeau) and (Bennett, Rudich). So, we can sustain the commitment by proving equality with a fresh one.

The Setting



A^{C}

A^{O}

B^{C}

B^{O}

Commit Side

Opening Side

A^{C} and A^{O} share a random string X.

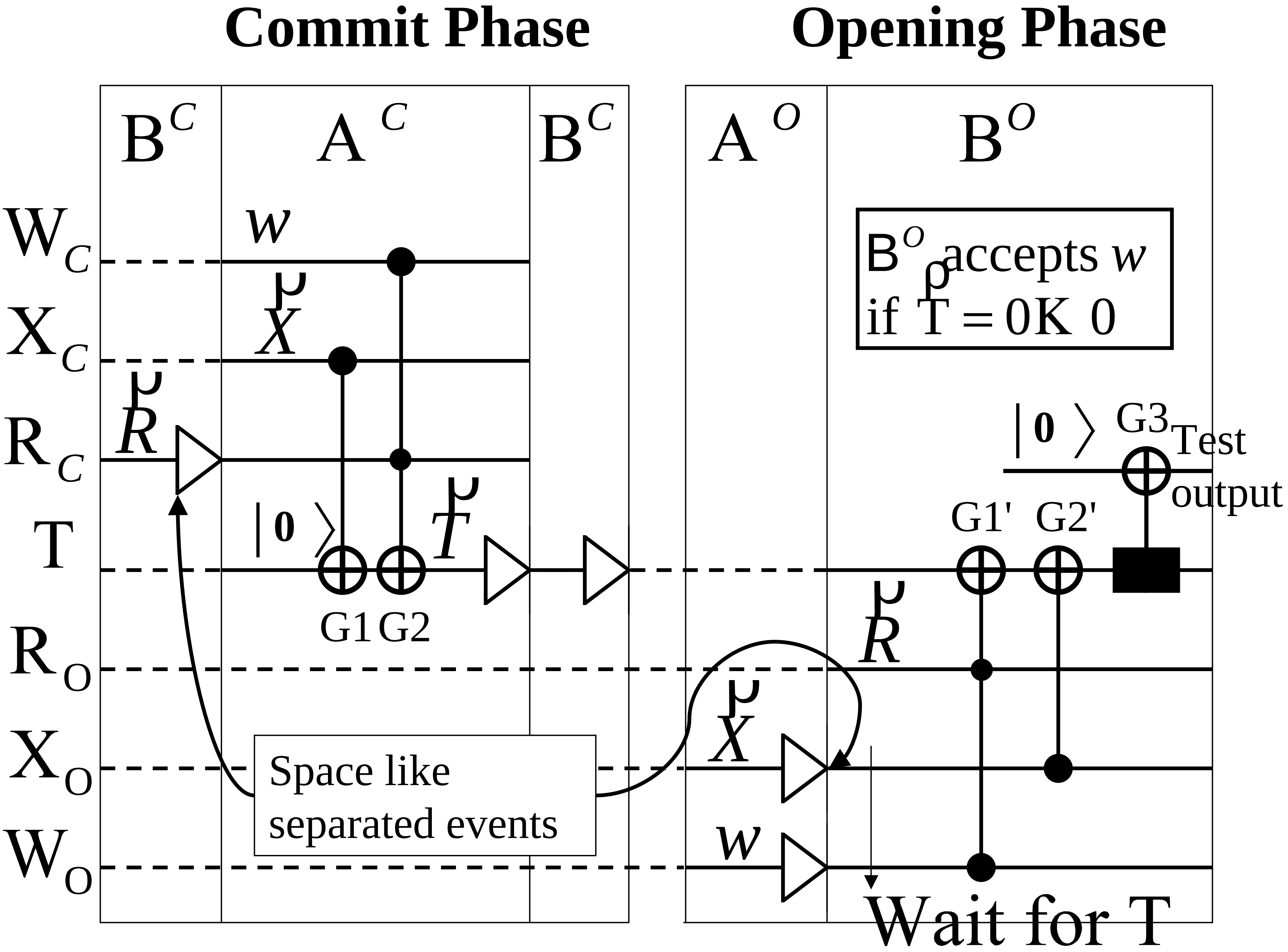
B^{C} and B^{O} share a random string R.

Temporary Relativistic Bit Commitment

Initial Setting: A^C and A^O share a random string X $\in_R$ { 0,1}^{m} and move at distant locations. B^C and B^O share a random string R $\in_R$ {0,1}^{m} and move close *to* A^C and A^O, respectively.

Commit(w):
B^C sends R to A^C
A^C sends T = [X, X $\oplus$ R]$^{(w)}$ back to B^C
B^C notes the time of receipt and forwards T to B^O

Opening:
A^O announces X and w to B^O
B^O notes the time of receipt and checks (1) [X, X $\oplus$ R]$^{(w)}$ = T and (2) the two receipts are space-like separated events.


Commit Phase
Opening Phase
B^C
A^C
B^C
A^O
B^O
W_C
X_C
R_C
T
R_O
X_O
W_O
w
μ X
μ R
|0⟩
G1
G2
μ T
B^O accepts w
ρ if T = 0K 0
|0⟩
G3
Test output
G1'
G2'
μ R
μ X
w
Space like separated events
Wait for T

What we can prove

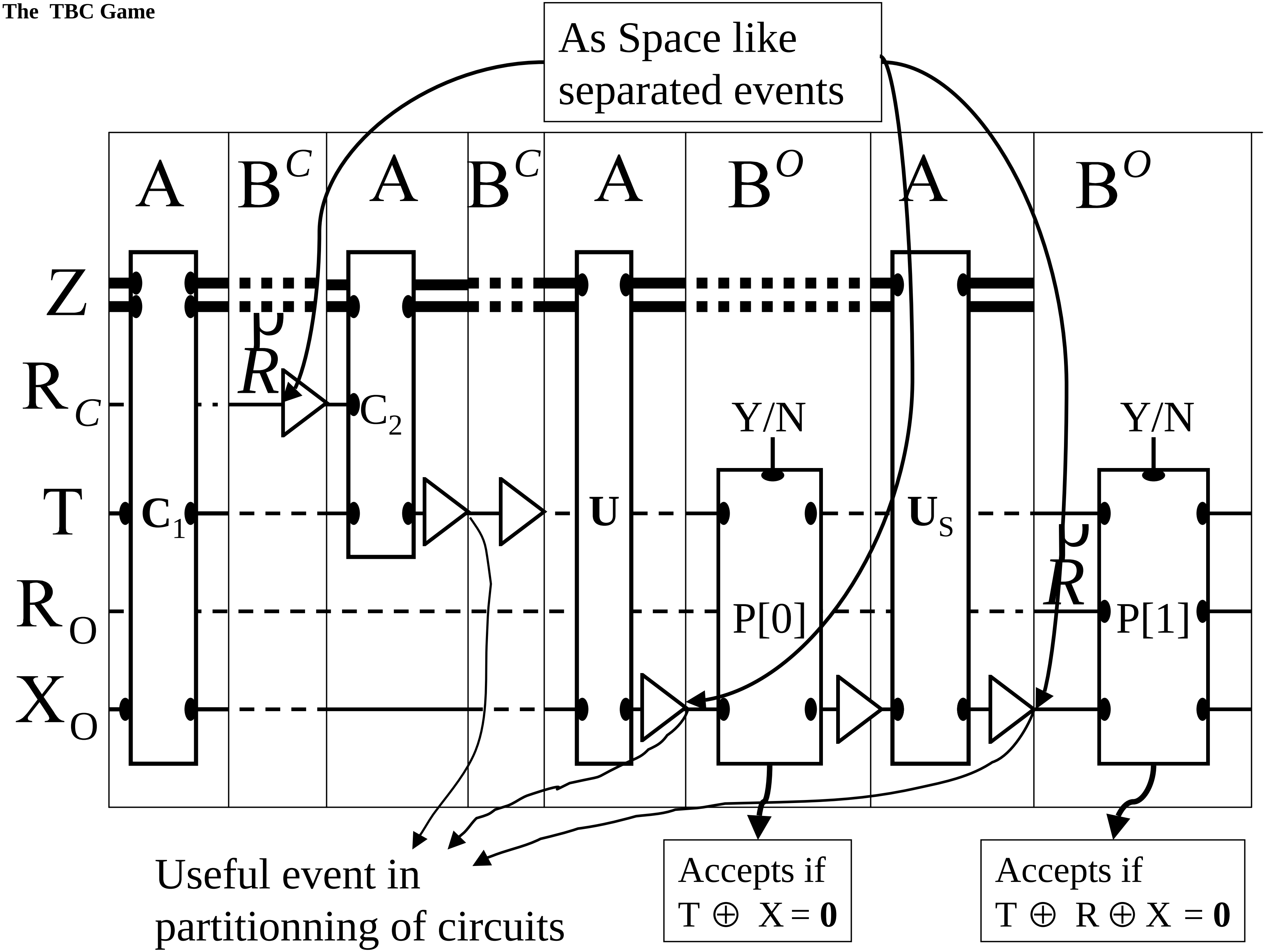
The concealing condition is easy! Let us consider the binding condition.

Let P_w, $w \in_R \{0,1,\perp\}$, be the measurement that corresponds to Bob`s opening. Let C be a commit circuit and U and U` be two opening circuits, and $U_S = U`U^\dagger$.

We can obtain the following binding condition: $(\forall C)(\forall U)(\forall U_S)$

$$\| P_1 U_S P_0 UC \, |init\rangle \, \|^2 \leq 2^{-m}$$

The TBC Game



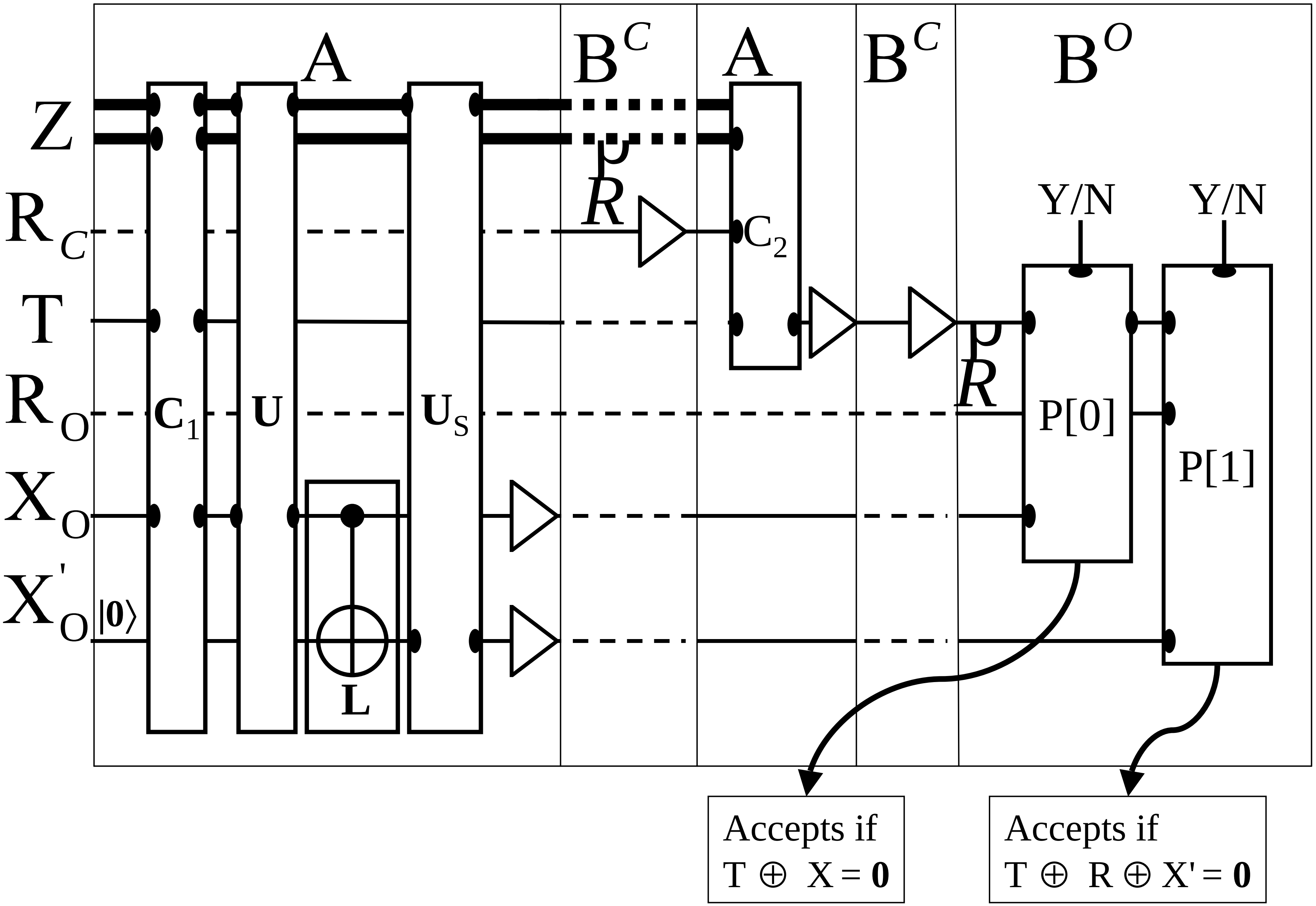
As Space like separated events

Useful event in partitionning of circuits

Accepts if T ⊕ X = **0**

Accepts if T ⊕ R ⊕ X = **0**

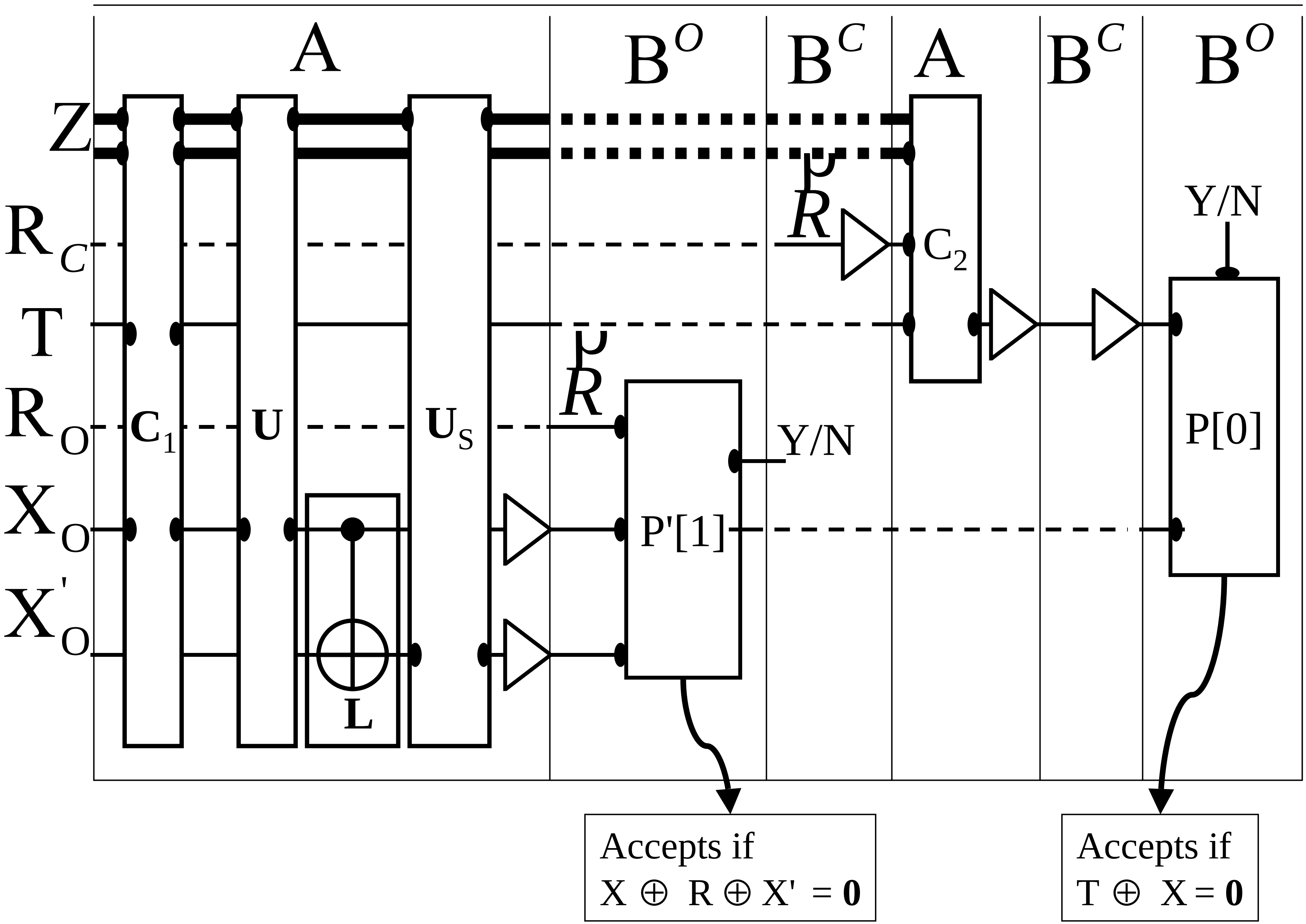
The First Modified TBC Game



Accepts if
T ⊕ X = **0**

Accepts if
T ⊕ R ⊕ X' = **0**

The Second Modified TBC Game



Accepts if
X ⊕ R ⊕ X' = **0**

Accepts if
T ⊕ X = **0**

So, we have a natural and achievable binding condition:

$$\| P_1 U_S P_0 UC |\mathrm{init}\rangle \|^2 \leq \alpha$$

We can obtain the same binding condition with a completely different bit commitment protocol based on any quantum one-way permutation (Dumais, Mayers, Salvail 2000).

Is this binding condition composable?

Borrowing from Classical Composability.

The issue of composability is important in standard cryptography and was progressively addressed in the last 10 years!

Canetti`s work sumarize these 10 years.

The techniques currently used for classical composability can be useful to built a theory of quantum composability.

Universal Security Definition
and
Universal Composability
in the
Quantum World

Universal Security Definition

A *universal security definition* is a relation of the form ¨Π securely realises F¨ where Π is any real protocol and F is any ideal functionality. The *ideal functionality* F is part of an *ideal protocol* also denoted F. We will be more precise later.

Canetti`s Security Definition

For every real adversary A against the real protocol Π, there must exist an ideal adversary S (also called a simulator) against the ideal protocol F, such that no environment Z can distinguish between Π^A (the real protocol Π corrupted by the real adversary A) and F^S (the ideal protocol F corrupted by the ideal adversary S).

Universal Composability: Basic Idea

Notation. We denote Π^F a protocol Π that calls an ideal protocol F. If ρ provides the same I/O interface as F, Π^ρ is the same protocol but calls ρ instead of F. We denote $F^{(m)}$ the ideal functionality that can run up to m invocations of the ideal functionality F. We denote $\rho^{(m)}$ the protocol that can run up to m invocations of the protocol ρ. No global synchronisation, except in between partners in the relativistic scenario.

A universal security definition is composable if:

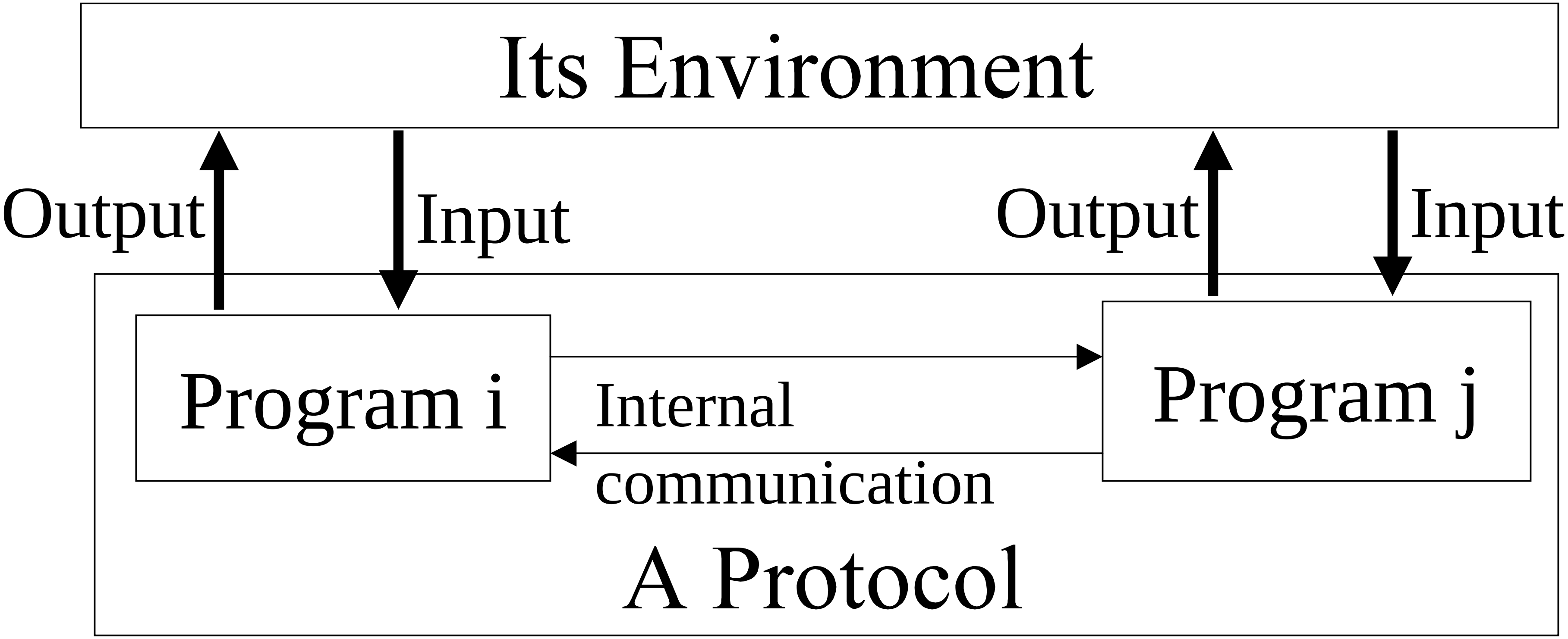
(1) If Π^F securely realises G and ρ securely realises F, then Π^ρ securely realises G.

(2) If ρ securely realises F, then $\rho^{(m)}$ securely realises $F^{(m)}$.

The Models

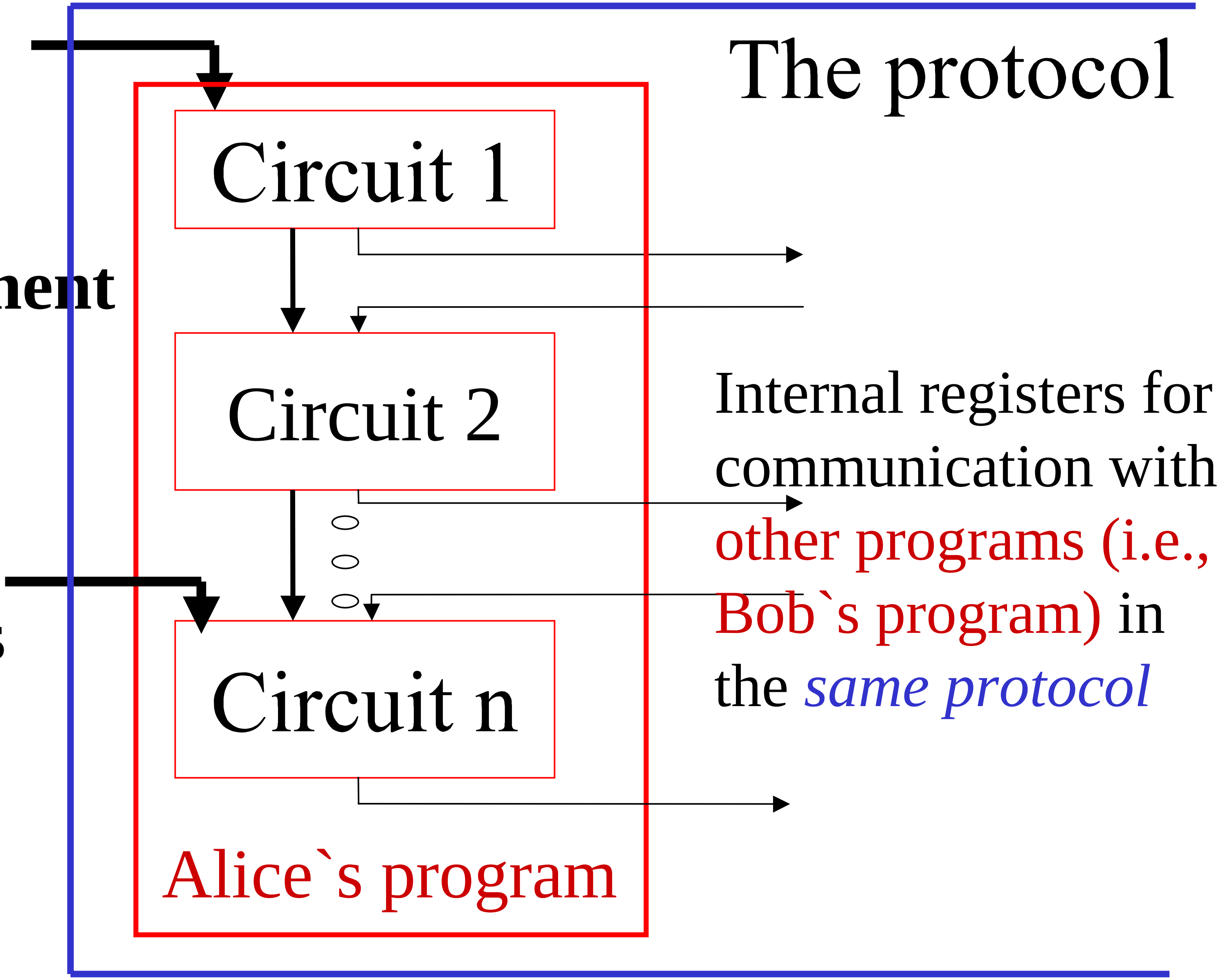
We need a model to represent what is the real situation and a model for what we would like to have. Actually, the two models can be unified into a single model which will simplify the analysis.

A Protocol and its Environment



A quantum protocol is a collection of circuits regrouped in disjoint sets called programs together with channels for internal communication and for communication with the environment.

A Program in its context



The protocol

Its Environment

Inputs and outputs registers

Circuit 1

Circuit 2

Circuit n

Alice`s program

Internal registers for communication with other programs (i.e., Bob`s program) in the *same protocol*

Some Remarks

- Registers are sent trough communication channels that respect assumptions (e.g., private or not, authenticated or not, etc.) We consider that these channels and assumptions are part of the definition of the protocol.

- Circuits are automatically activated when all the required registers are received (but transmission can be delayed in a relativistic scenario).

- All internal channels pass trough the adversary

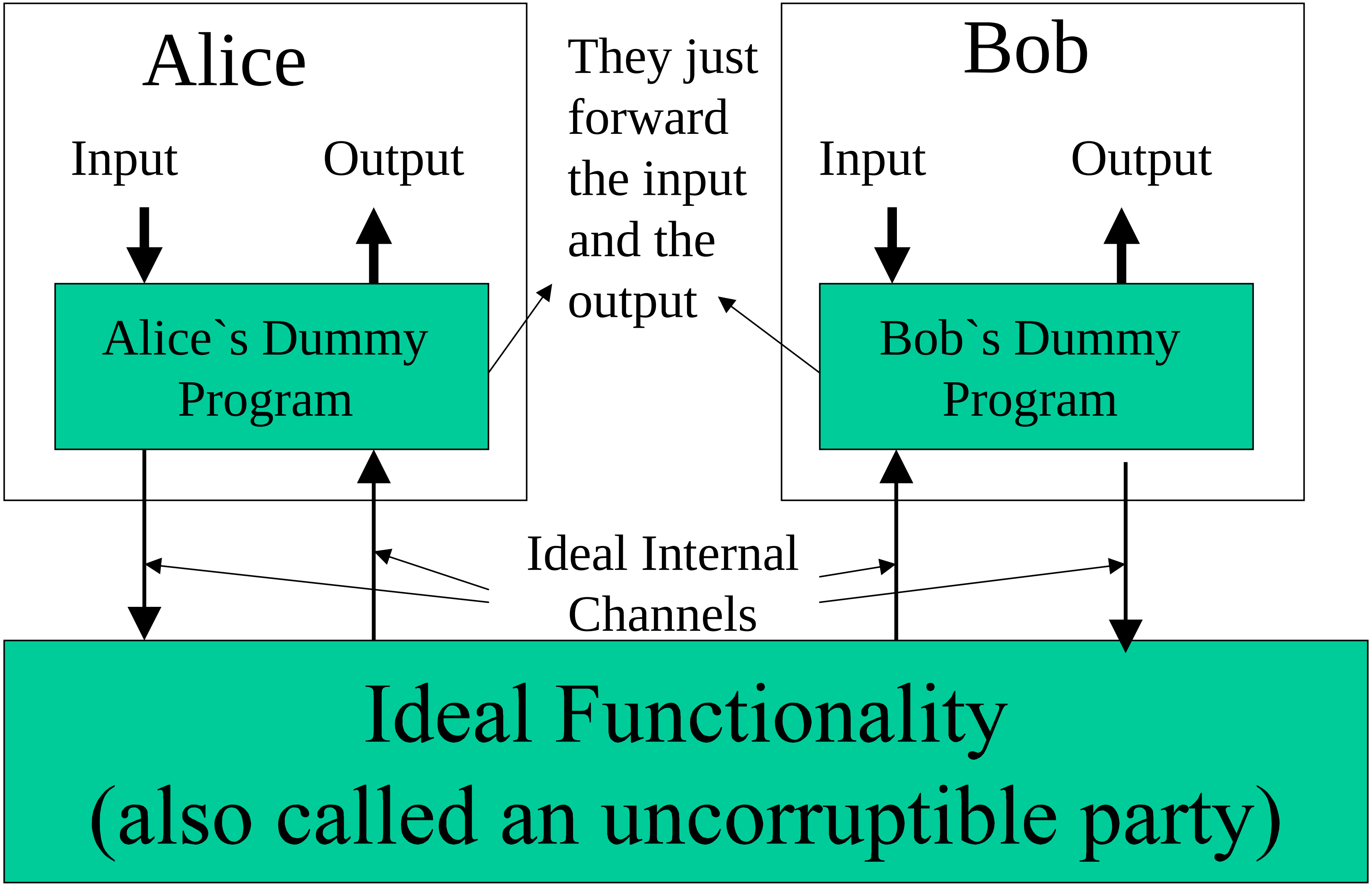
- Every program runs at a different location which is important in a relativistic scenario.

Functional and Internal Layers

The communication structure of a protocol determines two layers: the functional and the internal layers. The *functional layer* is defined by the relationship between the input and output registers of the protocol. It`s the protocol as seen by the environment.

The *internal layer* is defined by the circuits and the incoming and outgoing communication channels of the protocol. It`s the mean by which the functionality layer is realised.

Format of an ideal protocol



Alice

Input

Output

Alice`s Dummy Program

They just forward the input and the output

Bob

Input

Output

Bob`s Dummy Program

Ideal Internal Channels

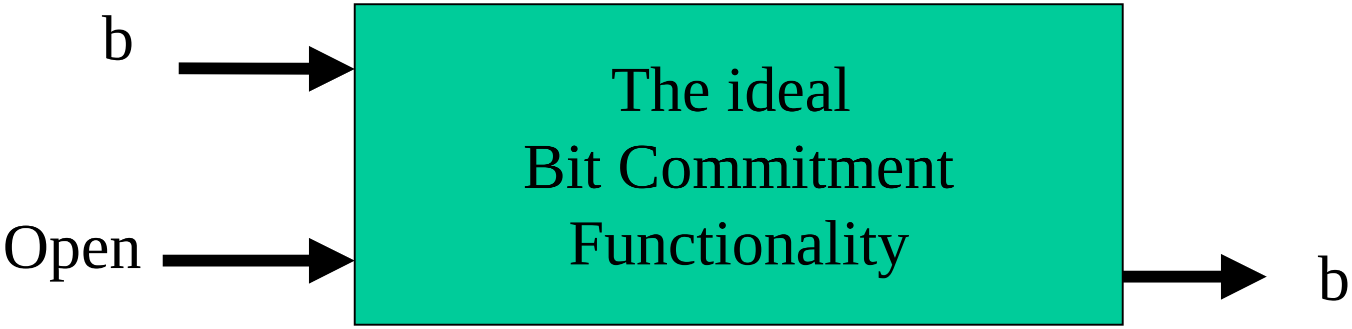
Ideal Functionality (also called an uncorruptible party)

Ideal Protocol: Internal and functionality layers.

An ideal protocol is simply a special case of a real protocol.

The functionality layer is defined by the relationship between the inputs and the outputs of the dummy parties. This relationship is in turn determined by the internal layer which is the ideal functionality F and the communication between the dummy parties and F.

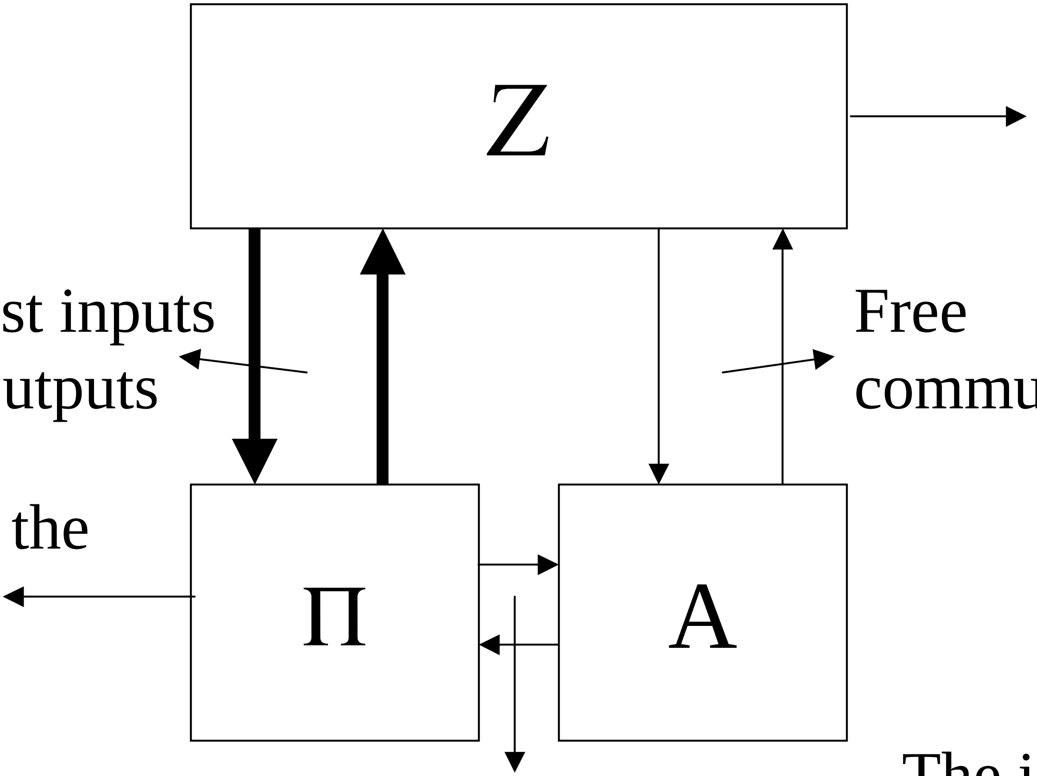
An example: The BC ideal functionality



1- Upon receiving (Commit, sid, Alice, Bob, b) from Alice, send (Receipt, sid, Alice, Bob) to Bob. (Ignore any subsequent Commit messages.)

2- Upon receiving a value (Open, sid, Alice, Bob) proceed as follows: if a previous (Commit, sid, Alice, Bob, b) was received from Alice, send (Open, sid, Alice, Bob, b) to Bob. Otherwise, do nothing.

The overall Model



Bit $Z(\Pi^A)$

Honest inputs and outputs

Free communication

Contains only the non corrupted programs.

The adversary A replaces the corrupted programs. Moreover, all internal communications in Π pass trough the adversary A.

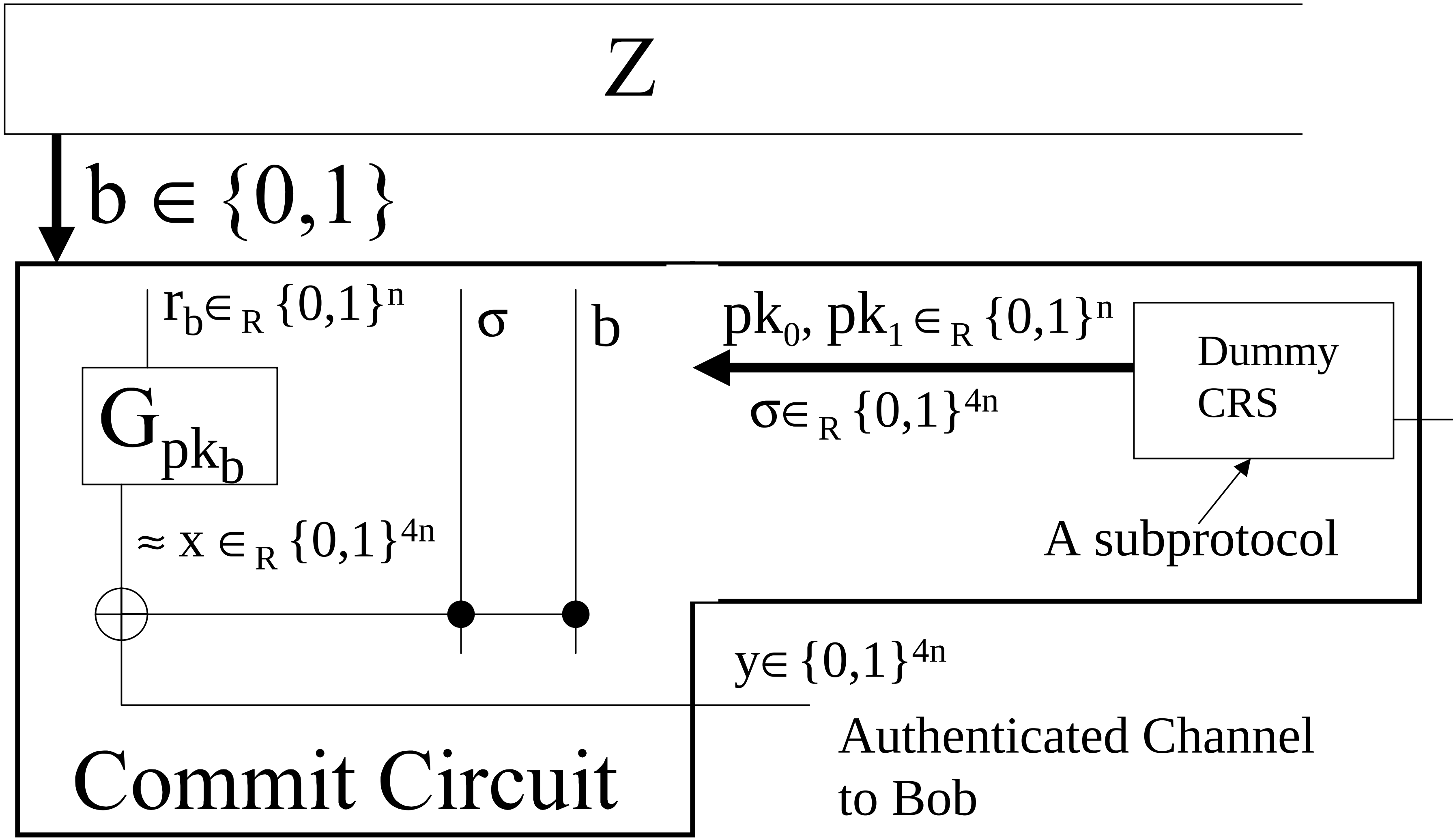
The ideal model has the same structure, but Π is replaced by F and A by S.

A Trade off

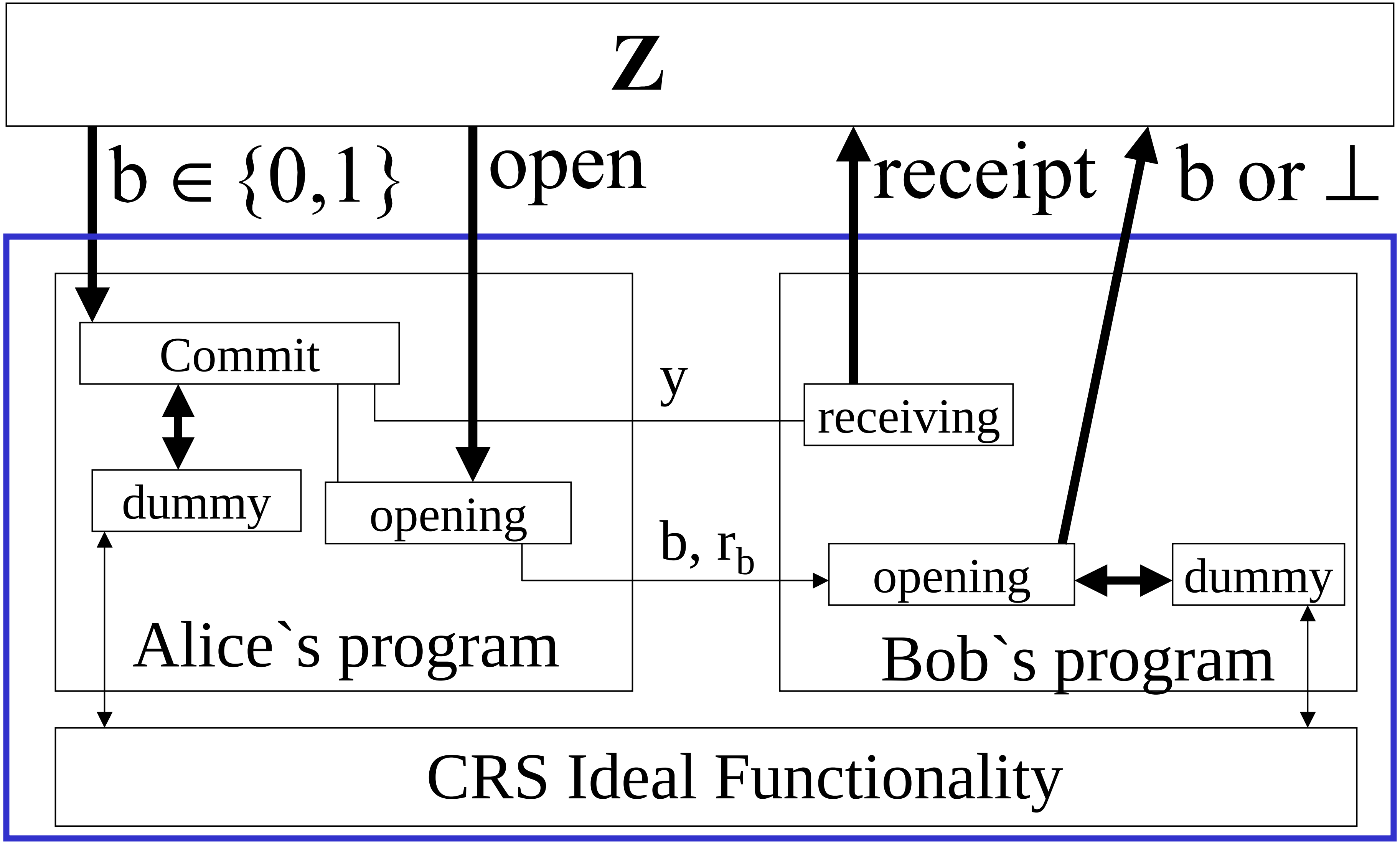
A more secure ideal protocol provides a stronger (more secure) definition of security. The strongest definition will state that no party can be corrupted in the ideal protocol and it will use ideal channels with guarantee of delivery, etc. The problem, of course, is to find protocols that achieve this level of security and I guess that it is impossible.   So, a trade off is neccessary.

Example: the ideal internal channels offer no guarantee of delivery because the real channels can be jammed.

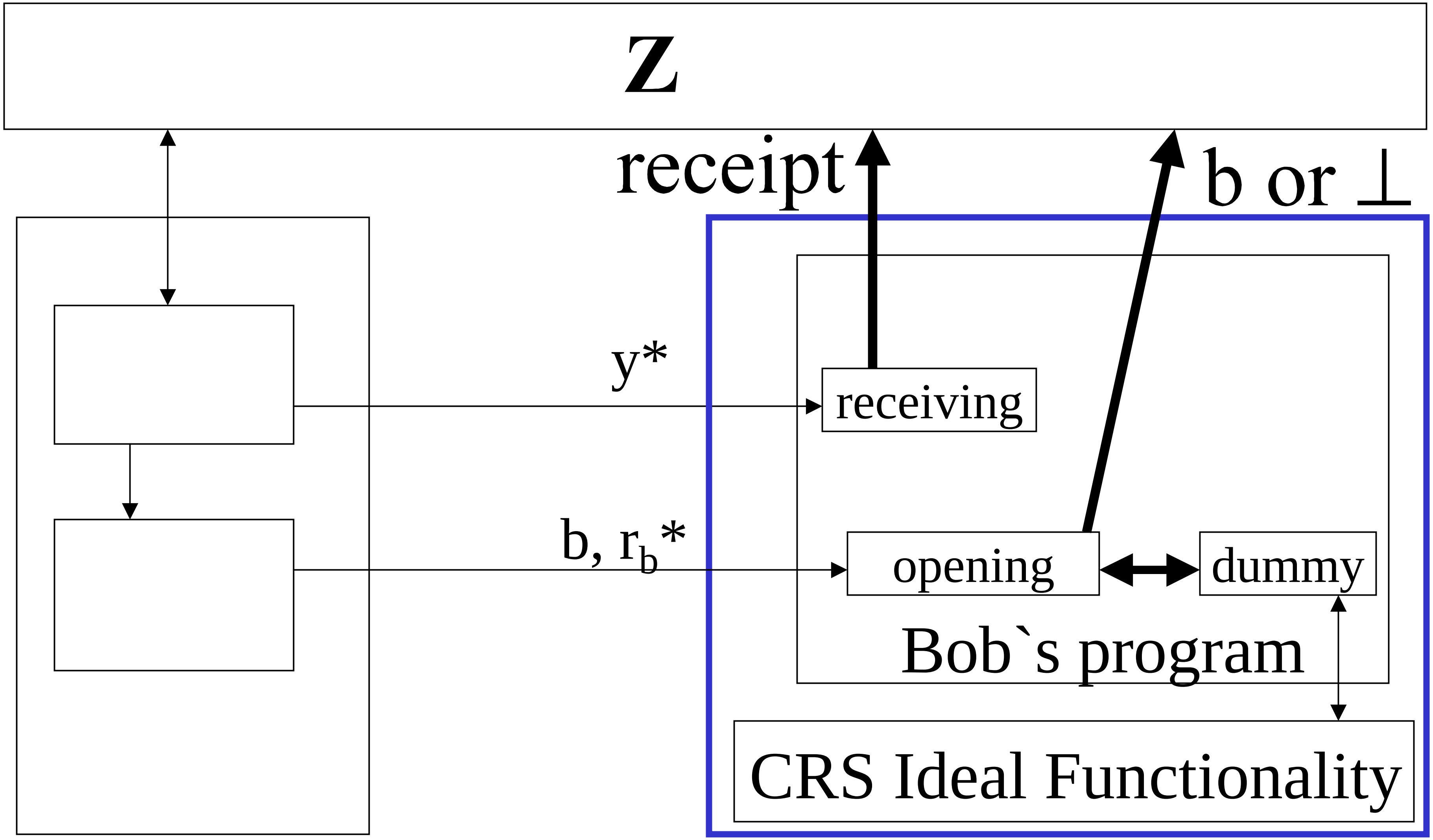
The Real NAOR-DIO Bit Commitment Scheme


Z
$b \in \{0,1\}$
$r_b \in_R \{0,1\}^n$
σ
b
$pk_0, pk_1 \in_R \{0,1\}^n$
G_{pk_b}
$\sigma \in_R \{0,1\}^{4n}$
Dummy CRS
$\approx x \in_R \{0,1\}^{4n}$
A subprotocol
$y \in \{0,1\}^{4n}$
Authenticated Channel to Bob
Commit Circuit

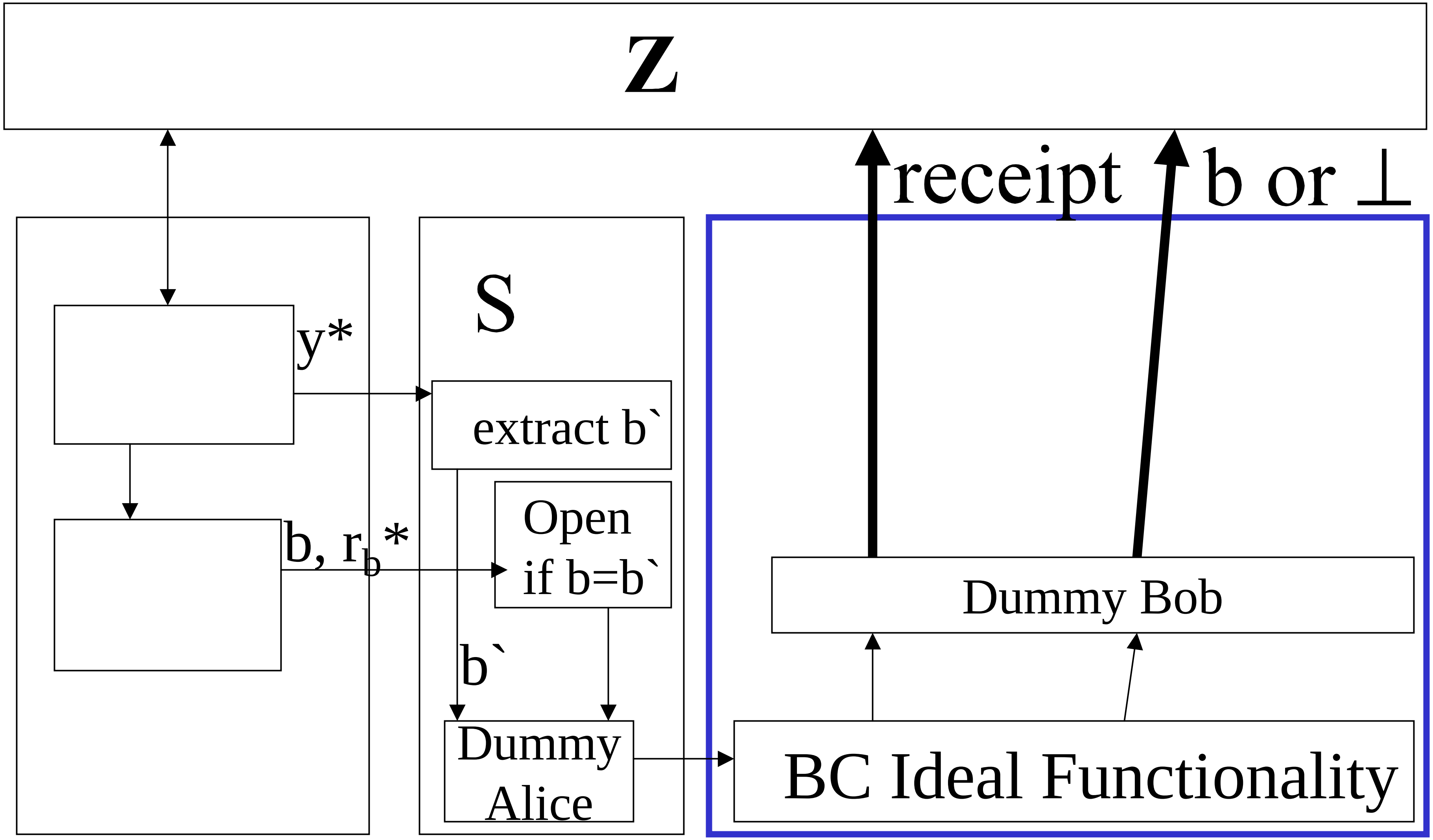
The Real NAOR-DIO Bit Commitment Scheme



The Real NAOR-DIO Scheme where Alice is corrupted



The Ideal NAOR-DIO Scheme where Alice is corrupted



The dummy and bad (real) adversary

Recall that the environment Z acts as a distinguisher. It must challenge the simulator S. To do so, it must obtain as much as possible about the internal layer from the real adversary A and give as little as possible to the simulator S.

The worst case real adversary (to challenge the simulator S) is the dummy adversary Ã which simply accepts to follow any of the following requests from the environment:

1- forward any new outgoing message in the protocol to the environment
2- corrupt a new party (if allowed by the access rule) and pass the information to the environment
3- deliver a message chosen by the environment to a party also chosen by the environment.

Quantum Universal Security Definition

For any two random binary variables Y, Y` let us write

$$Y \approx_e Y` \quad \text{if} \quad |\Pr(Y = 0) - \Pr(Y` = 0)| \leq e.$$

Let P be the set of all polynomial functions.

Definition. A protocol Π for an ideal functionality F is secure, if for any environment Z there exists a simulator S such that $(\forall d \in P)\ (\exists k_0 \in \aleph)(\forall k > k_0)$

$$Z(\Pi) \approx_e Z(F^S)$$

where e = 1/d(k). Moreover, the simulator S must have a polynomial complexity c ∈ P that depends only on Π, and k_0 can only depend on d and on the polynomial complexity c, c` ∈ P of S and Z, not on the actual circuits.

Universal Quantum Composability Theorem

Two Lemma

Let Π be an environment that calls an I/O interface shared by an ideal protocol F and a real protocol ρ.

Lemma 1. If ρ securely realizes F and Π^F securely realises G, then Π^ρ securely realises G.

Lemma 2. If ρ securely realizes F, then, $(\forall m \in P)$, $\rho^{(m)}$ securely realizes $F^{(m)}$.

Note: Lemma 1 and 2 can be combined to obtain: $(\forall m \in P)$, if ρ securely realizes F and $\Pi^{F(m)}$ securely realises G, then $\Pi^{\rho(m)}$ securely realises G.

Proof of lemma 1

There are three steps in the proof.

(1) Essentially, we must construct a simulator $S(\Pi^{\rho})$ for Π^{ρ} given the simulators $S(\Pi^{F})$ for Π^{F} and $S(\rho)$ for ρ.

(2) We must show that the size of the simulator $S(\Pi^{\rho})$ is a polynome that depends only on the protocol Π^{ρ}.

(3) We must show that the lower bound k_0 for k depends only on the polynome d (for the indistinguishability) and on the complexity of the circuits Z and S, not on the actual circuits.

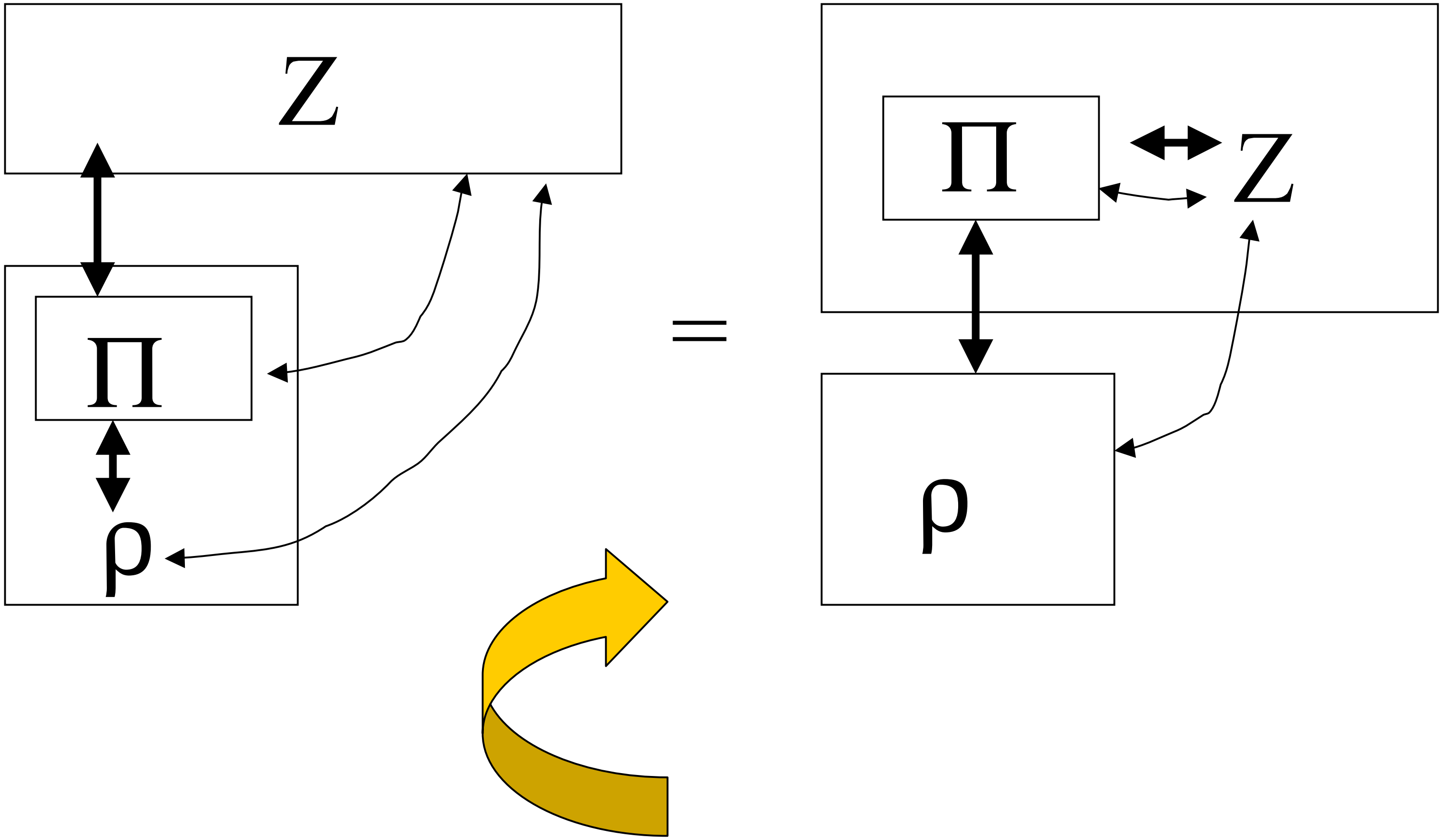
Proof of lemma 1 (Cont)

The line of argument is the following.

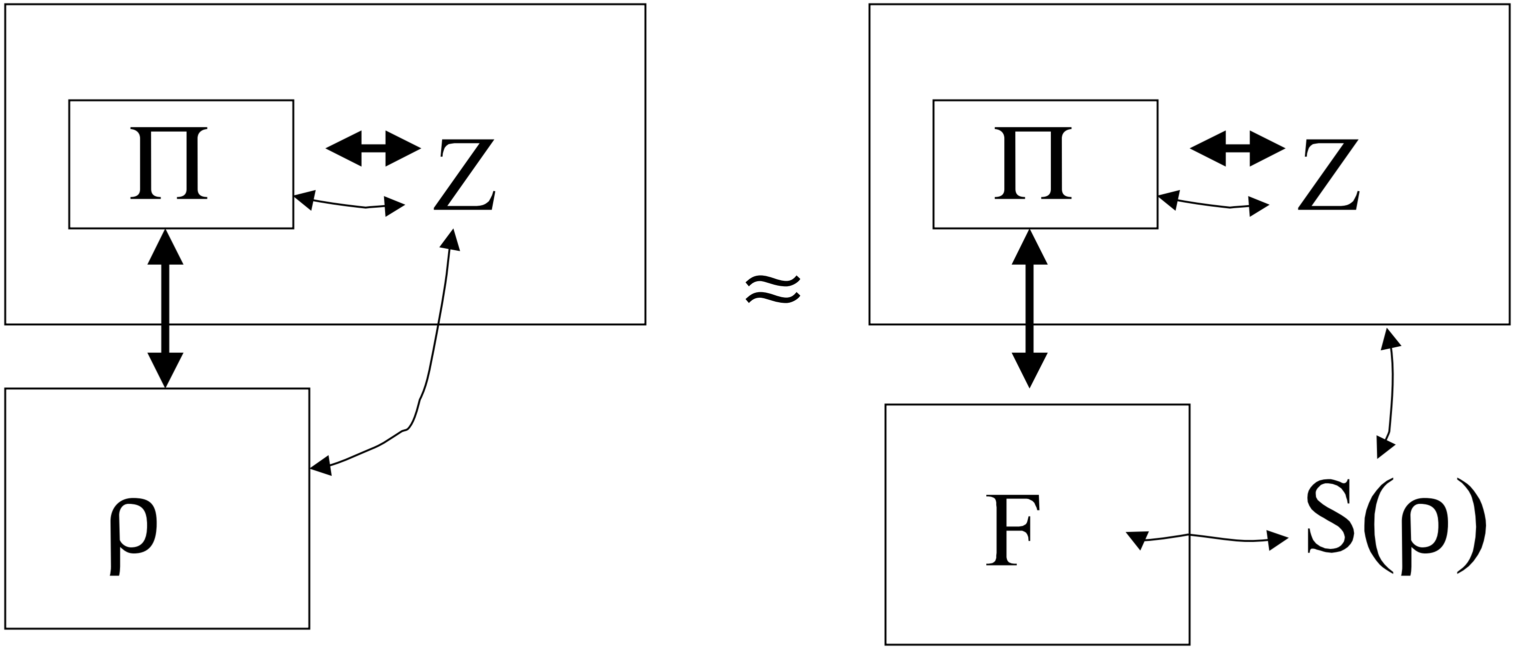
	Statements	Justifications
A	$Z(\Pi^{\rho}) = [Z \cup \Pi](\rho)$	Different views on the same process
B	$[Z \cup \Pi\ ](\rho) \approx_{1/(2d(k))} [Z \cup \Pi](F^{S(\rho)})$	This is (1)
C	$[Z \cup \Pi](F^{S(\rho)}) = [Z \cup S(\rho)](\Pi^{F})$	Different views on the same process
D	$[Z \cup S(\rho)](\ \Pi^{F}) \approx_{1/(2d(k))} [Z \cup S(\rho)](G^{S(\Pi^{F})})$	This is (2)
E	$[Z \cup S(\rho)](G^{S(\Pi^{F})}) = Z(G^{S(\Pi^{\rho})})$.	Different views on the same process

This concludes the proof! (See next slides for details)

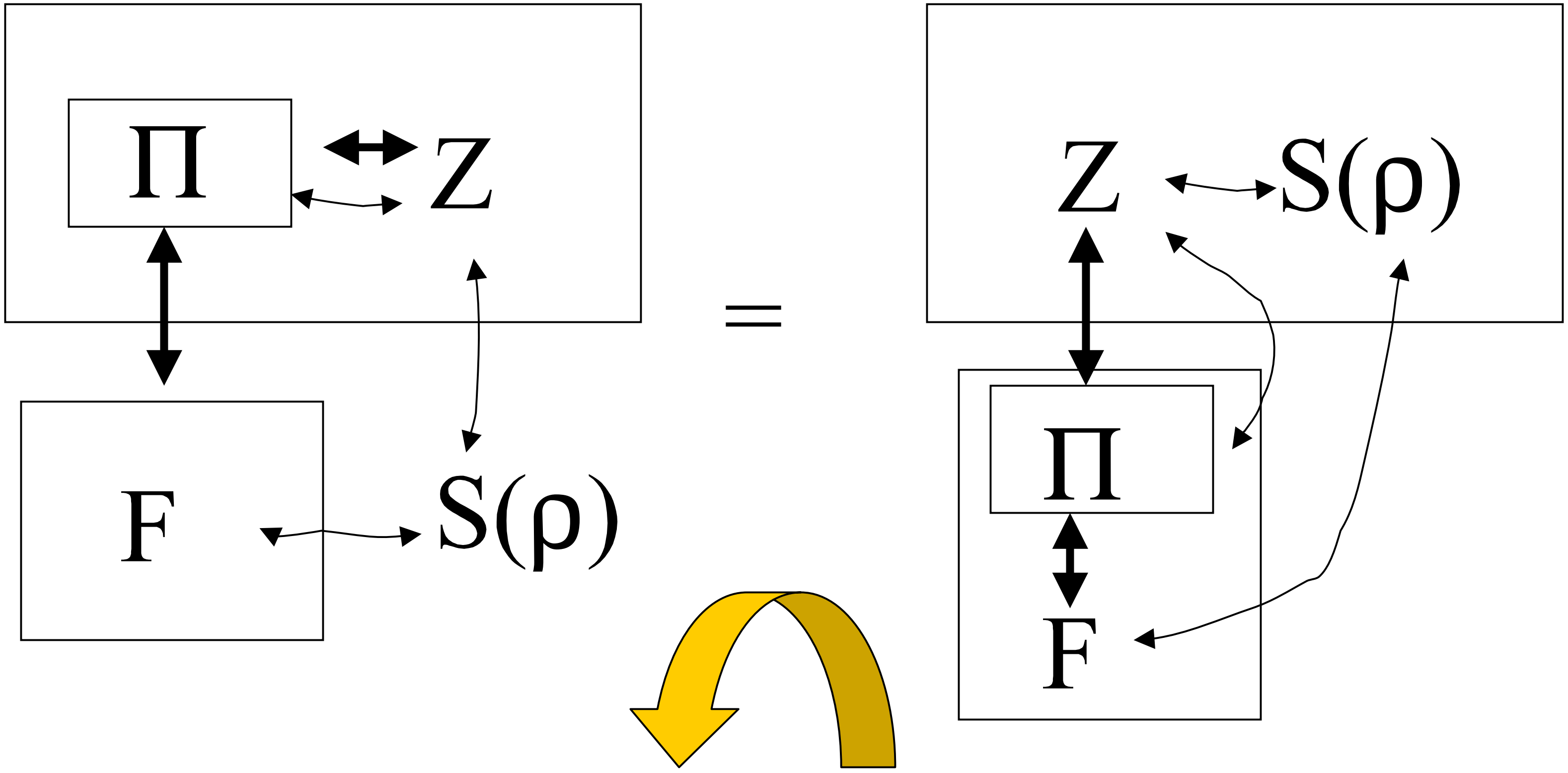
A: $Z(\Pi^{\rho}) =[Z \cup \Pi](\rho)$



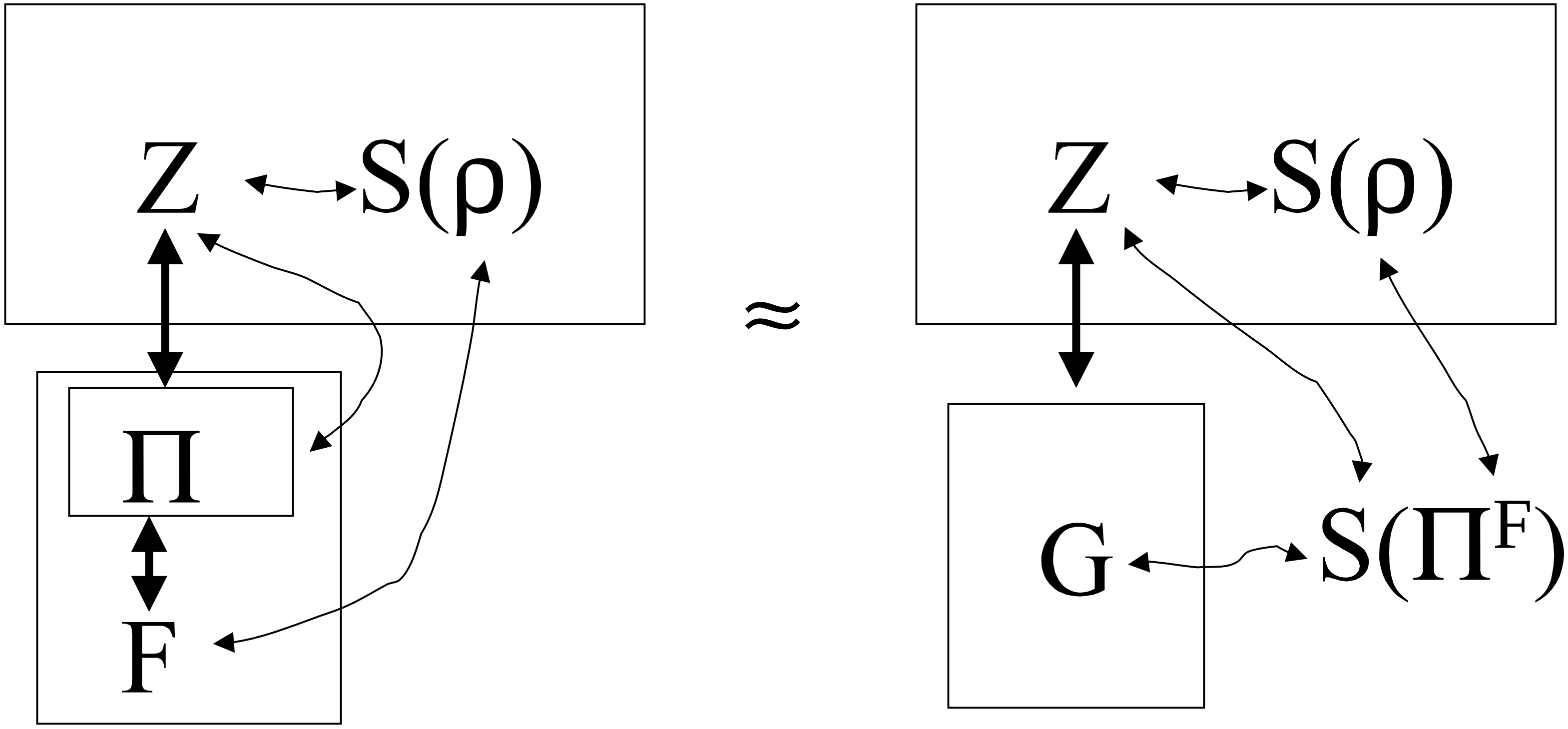
B: $[Z \cup \Pi\,](\rho) \approx_{1/(2d(k))} [Z \cup \Pi](F^{S(\rho)})$



$$C: \; [Z \cup \Pi](F^{S(\rho)}) = [Z \cup S(\rho)](\Pi^{F})$$



$$D: [Z \cup S(\rho)](\Pi^F) \approx_{1/(2d(k))} [Z \cup S(\rho)](G^{S(\Pi^F)})$$



Z

$S(\rho)$

Π

F

$\approx$

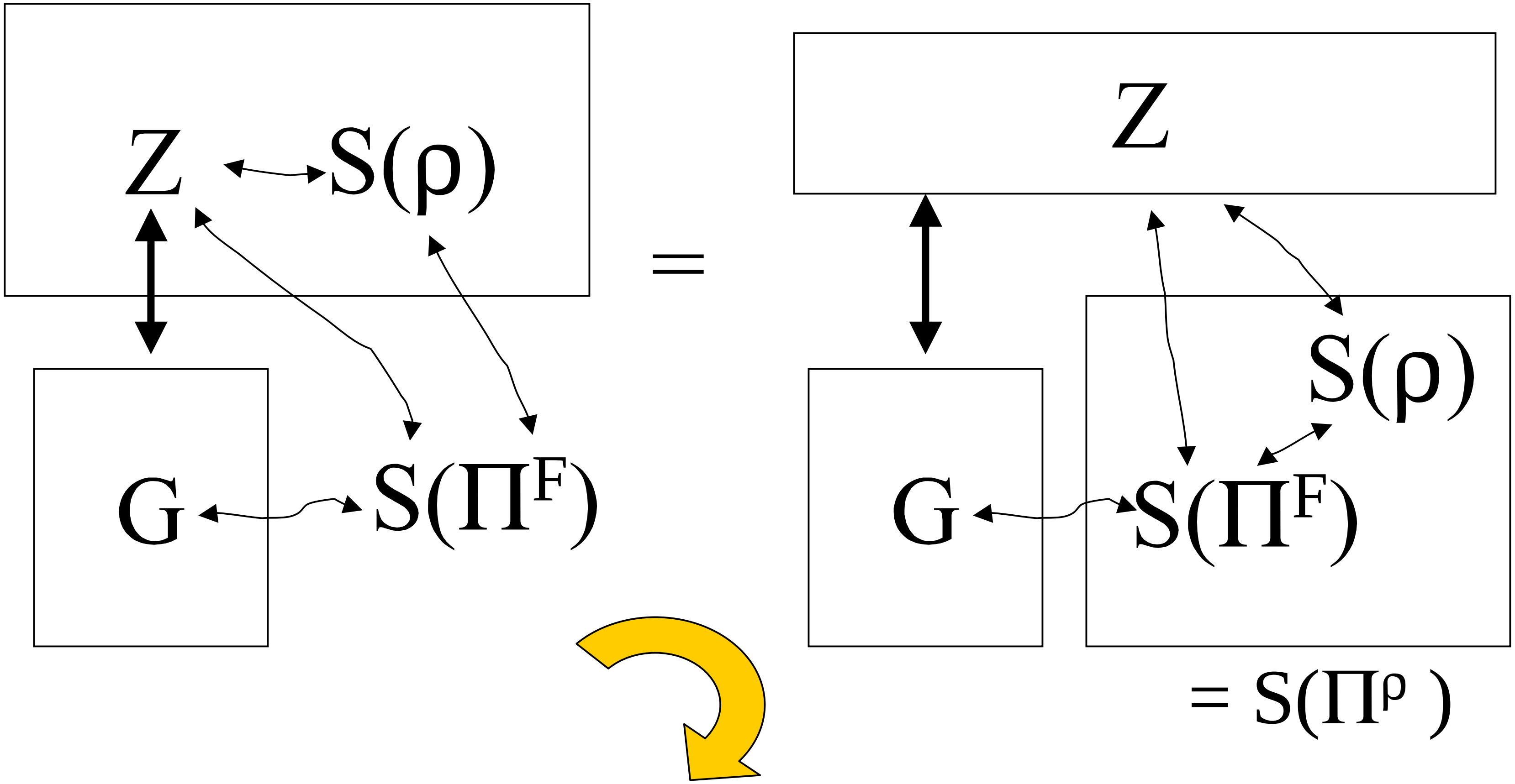
Z

$S(\rho)$

G

$S(\Pi^F)$

$$E\colon [Z \cup S(\rho)](G^{S(\Pi^{F})}) = Z(G^{S(\Pi^{\rho})})$$



$= S(\Pi^{\rho})$

The main question (work in progress)

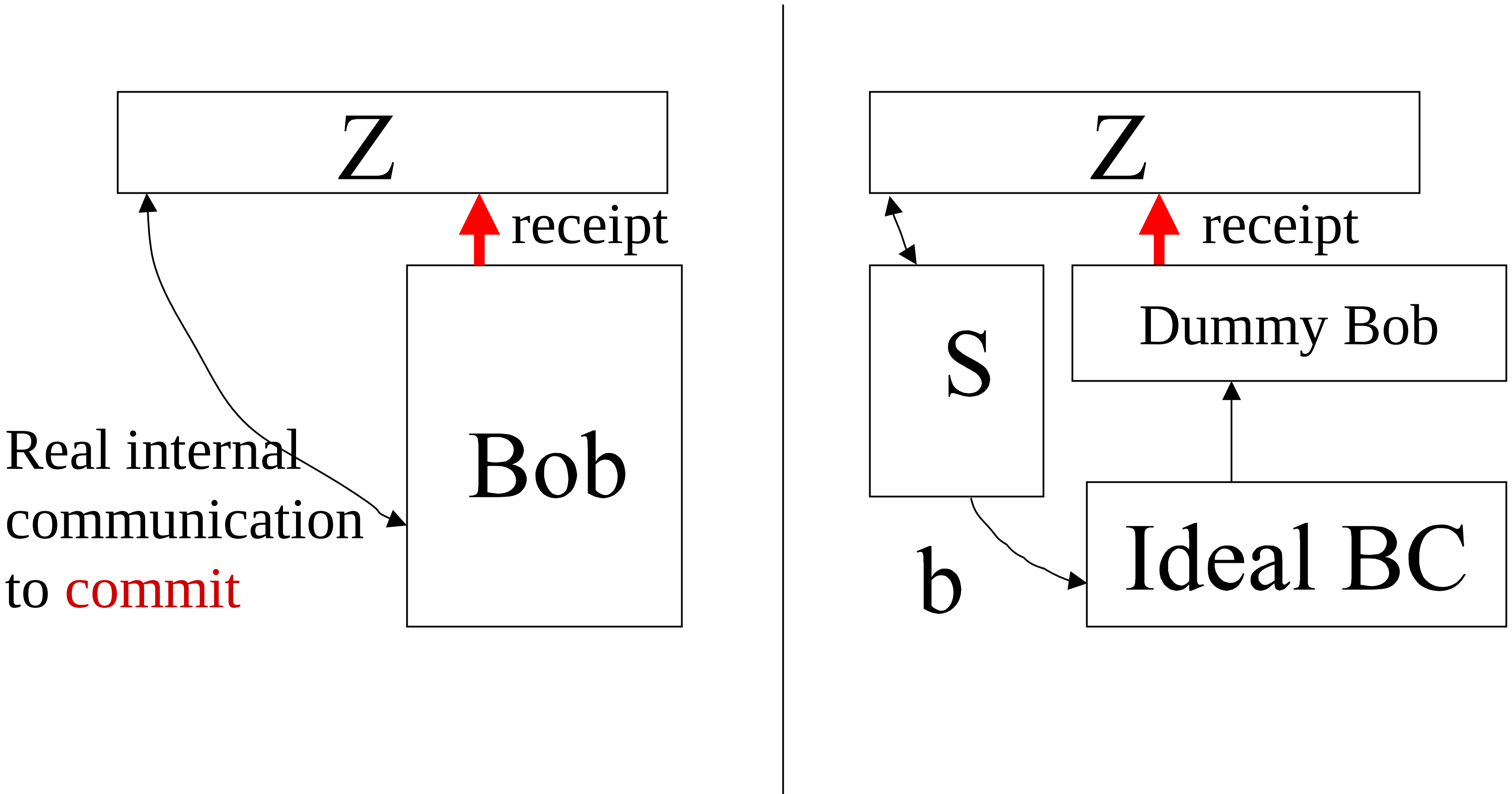
How can we use the binding condition

$$\| P_1 U_S P_0 U C \, |\mathrm{init}\rangle \; \|^2 \leq \alpha$$

to obtain a universally composable binding condition?

We must extract the bit!

We recall that when Alice is corrupted the simulator must extract the bit in the commit phase:



The basic idea to extract the bit!

$$\| P_1 U_S P_0 UC |\mathrm{init}\rangle \|^2 \leq \alpha$$

$$\Rightarrow$$

$$\| \langle\mathrm{init}| C^\dagger U`^\dagger P_1 U` U^\dagger P_0 UC |\mathrm{init}\rangle \|^2 \leq \alpha$$

A Variation on the Ideal Adversary (Work in Progress)

The Extra Circuit

The alternative ideal adversary is the same except that it contains an extra circuit with the following properties:

1- on request, it receives all registers, except the ``input´´ registers of Z,   execute a measurement and sends these registers back to the programs and the environment, and

2- for any fixed value of the honest input or output in the protocol analysed (e.g. the bit that is open by Bob when Alice is corrupted), this measurement does not disturb the state of the entire application protocol.

Any register in the environment which is only used as a source qbit in CNOT gates (no rotation and never the target of a CNOT) is an ``input´´ register.

Interesting Open Question:
A theory of Cheat Sensitive Security?

Recall the protocol for Coin Tossing on Top of Bit Commitment

CT

Alice commits a bit $A \in_R \{0, 1\}$
Bob announces $B \in_R \{0, 1\}$
Alice opens A
Alice and Bob compute $A \oplus B$

Alice cannot create a bias on $A \oplus B$ because she does not know B when she picks A. Similarly, Bob cannot create a bias because he does not know A when he picks B.

Cheat Sensitive Security.

Aharonov, Ta-Shma, Vazirani and Yao (1999) and independently Kent and Hardi (1999) proposed Weak Bit Commitment. Intuitively, in a weak bit commitment no participant can cheat without running a chance to be detected.

Spekkens (2002) proposed Cheat Sensitive Coin Tossing. If the cheater creates a bias above some threshold $\varepsilon \geq 0$, he runs a chance to be detected. It is likely that the optimal threshold is $\varepsilon = 0$, i.e. no bias, on both sides but this is not proven.

Natural Question: Can we built a (better) cheat sensitive coin tossing on top of a cheat sensitive bit commitment? The answer is no, if we use weak bit commitment as a cheat sensitive bit commitment. However, it is possible if we slightly modify the definition.

A theory of Cheat Sensitivity?

Cheat sensitivity is interesting because we can hope to obtain it with unconditional security for most cryptographic tasks without the help of trusted parties!

It would be much more interesting if cheat sensitivity was composable as we hope it is the case in the particular case of cheat sentitive bit commitment (given an adequate definition). At this time, we have no theory of cheat sensitivity.

Perhaps, a general composability lemma is possible with cheat sensitive security. Such a lemmma would provide cheat sensitive coin flipping, cheat sensitive oblivious transfer, etc. on top of cheat sensitive bit commitment. The other standard reductions would hold as well.

Summary of results and Conclusions

1- Here, we have verified that the universal classical composability theorem is valid in the quantum world, even for tasks with quantum inputs! It should applied to quantum multiparty computation (but not yet checked).

2- Here, we have obtained a natural binding condition for a relativistic bit commitment protocol. The same condition applies to other kind of bit commitment protocols (DMS 2000). We believe that this definition is composable in some way.

3- Point 2 is our main motivation to look for a variation on our universal definition that is easier to achieve and yet composable. (Work in progress)

4- Universal Composability might also be interesting for Cheat Sensitive Security (Work in progress).