Semidefinite programming in two-party quantum cryptography

Part I : Basics of semidefinite programming

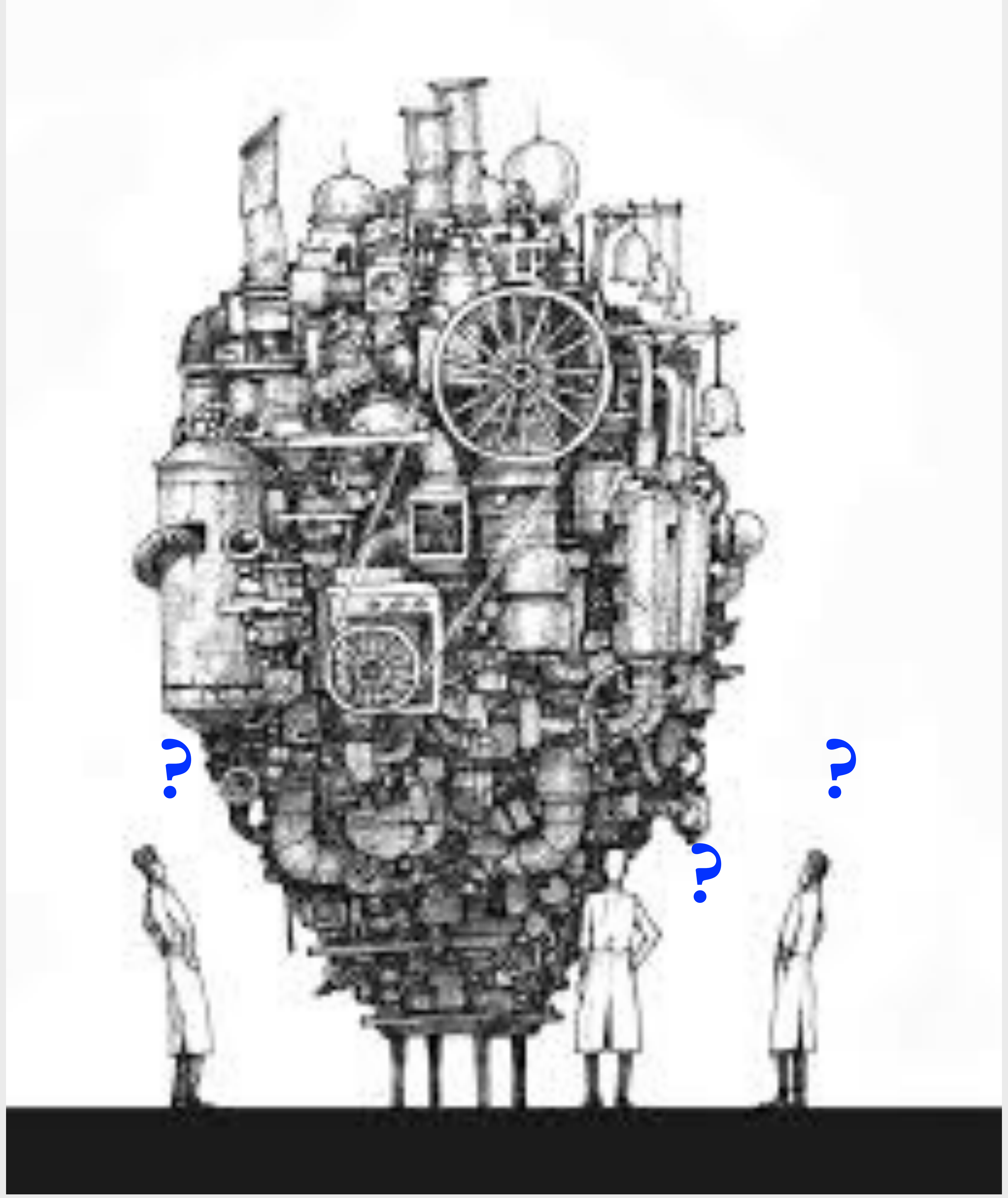
Presenter: Akshay Bansal (Slides courtesy: Jamie Sikora)

Should I pay attention?

Abstract

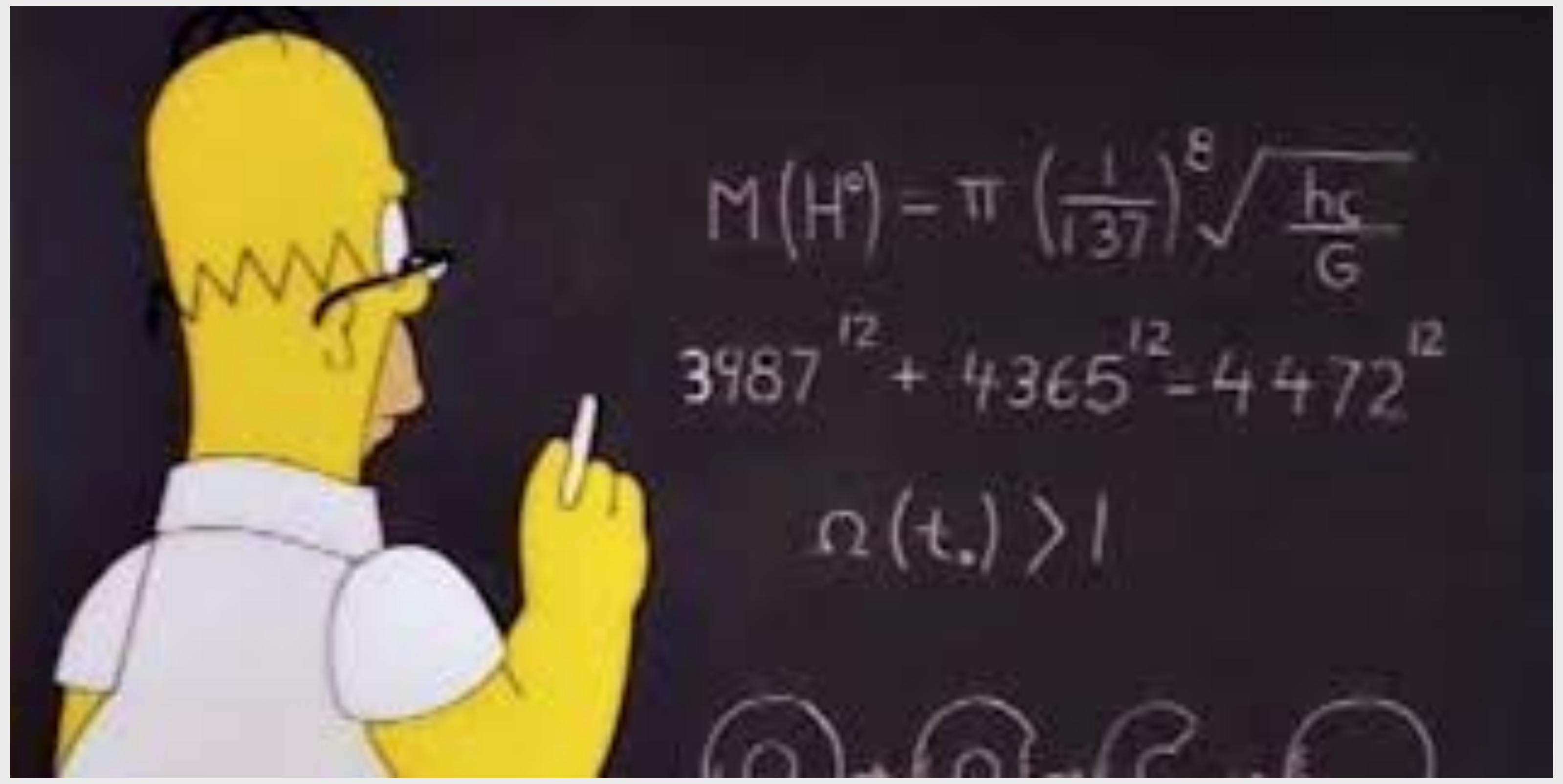
Quantum Mechanics

A bunch of positive semidefinite things that interact with other positive semidefinite things in some kind of linear way



Abstract

Semidefinite Programming (Optimization)

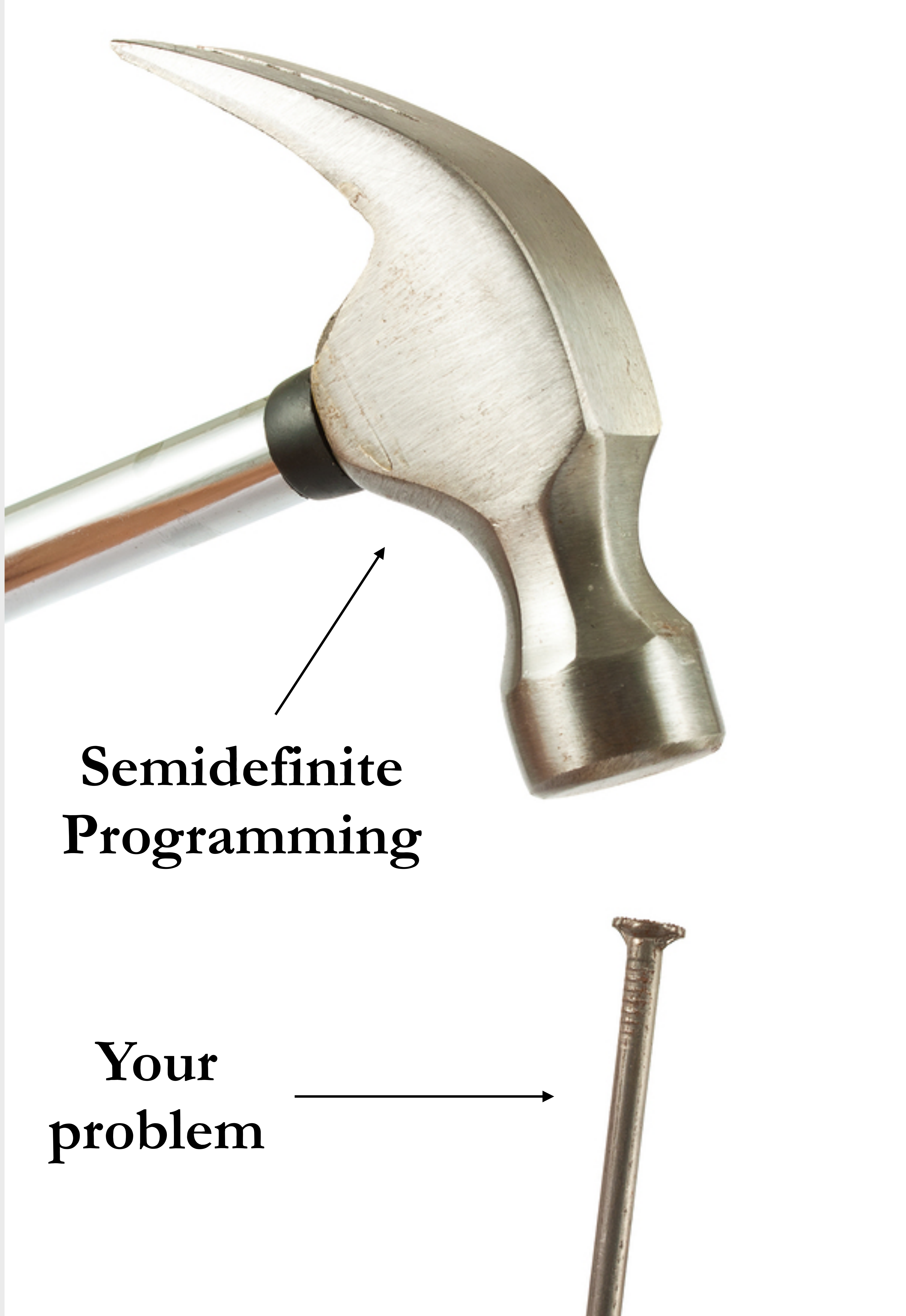


Optimizing linear functions of positive semidefinite things that satisfy some linear conditions

Where do semidefinite programs appear?

Quantum… Cryptography
Complexity Theory
Query Complexity
Information Theory
Entanglement Theory
Graph Theory

Linear Optics
Bell Non-locality
Causal Structures
and many more…


Semidefinite Programming
Your problem

What is a semidefinite program?

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints.

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$
\begin{aligned}
\alpha = \text{maximize: } & \langle A, X \rangle \\
\text{subject to: } & \Phi(X) = B \\
& X \in \text{Pos}(\mathcal{X})
\end{aligned}
$$

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\begin{aligned} \alpha = \text{maximize: } & \langle A, X\rangle \\ \text{subject to: } & \Phi(X) = B \\ & X \in \text{Pos}(\mathcal{X}) \end{aligned}$$

$\mathcal{X}, \mathcal{Y}$ are vector spaces

$A \in \text{Herm}(\mathcal{X})$

$B \in \text{Herm}(\mathcal{Y})$

Φ is linear and maps $\text{Herm}(\mathcal{X})$ to $\text{Herm}(\mathcal{Y})$

(A, B, Φ) is the **data**

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\begin{aligned} \alpha = \text{maximize: } & \langle A, X \rangle \\ \text{subject to: } & \Phi(X) = B \\ & X \in \text{Pos}(\mathcal{X}) \end{aligned}$$

$\mathcal{X}, \mathcal{Y}$ are vector spaces

$A \in \text{Herm}(\mathcal{X})$

$B \in \text{Herm}(\mathcal{Y})$

Φ is linear and maps $\text{Herm}(\mathcal{X})$ to $\text{Herm}(\mathcal{Y})$

(A, B, Φ) is the **data**

X is the **variable**

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\begin{aligned} \alpha = \text{maximize: } & \langle A, X \rangle \\ \text{subject to: } & \Phi(X) = B \\ & X \in \text{Pos}(\mathcal{X}) \end{aligned}$$

Objective function

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\alpha = \text{maximize: } \langle A, X \rangle$$
$$\text{subject to: } \Phi(X) = B$$
$$X \in \text{Pos}(\mathcal{X})$$

Constraints

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\begin{aligned} \alpha = \text{maximize: } & \langle A, X \rangle \\ \text{subject to: } & \Phi(X) = B \\ & X \in \text{Pos}(\mathcal{X}) \end{aligned}$$

Optimal objective function value (or, simply, the value)

This could be finite, $-\infty$, or $+\infty$

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$
\begin{aligned}
\alpha = \text{maximize: } & \langle A, X \rangle \\
\text{subject to: } & \Phi(X) = B \\
& X \in \text{Pos}(\mathcal{X})
\end{aligned}
$$

$\mathscr{A} = \{X \in \text{Pos}(\mathcal{X}) : \Phi(X) = B\}$ is called the **feasible region**.

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\begin{aligned} \alpha = \text{maximize: } & \langle A, X \rangle \\ \text{subject to: } & \Phi(X) = B \\ & X \in \mathrm{Pos}(\mathcal{X}) \end{aligned}$$

If $\mathscr{A} = \varnothing$, then the SDP is **infeasible**. Otherwise, the SDP is **feasible**.

$\mathscr{A} = \{X \in \mathrm{Pos}(\mathcal{X}) : \Phi(X) = B\}$ is called the **feasible region**.

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\begin{aligned} \alpha = \text{maximize: } & \langle A, X \rangle \\ \text{subject to: } & \Phi(X) = B \\ & X \in \text{Pos}(\mathscr{X}) \end{aligned}$$

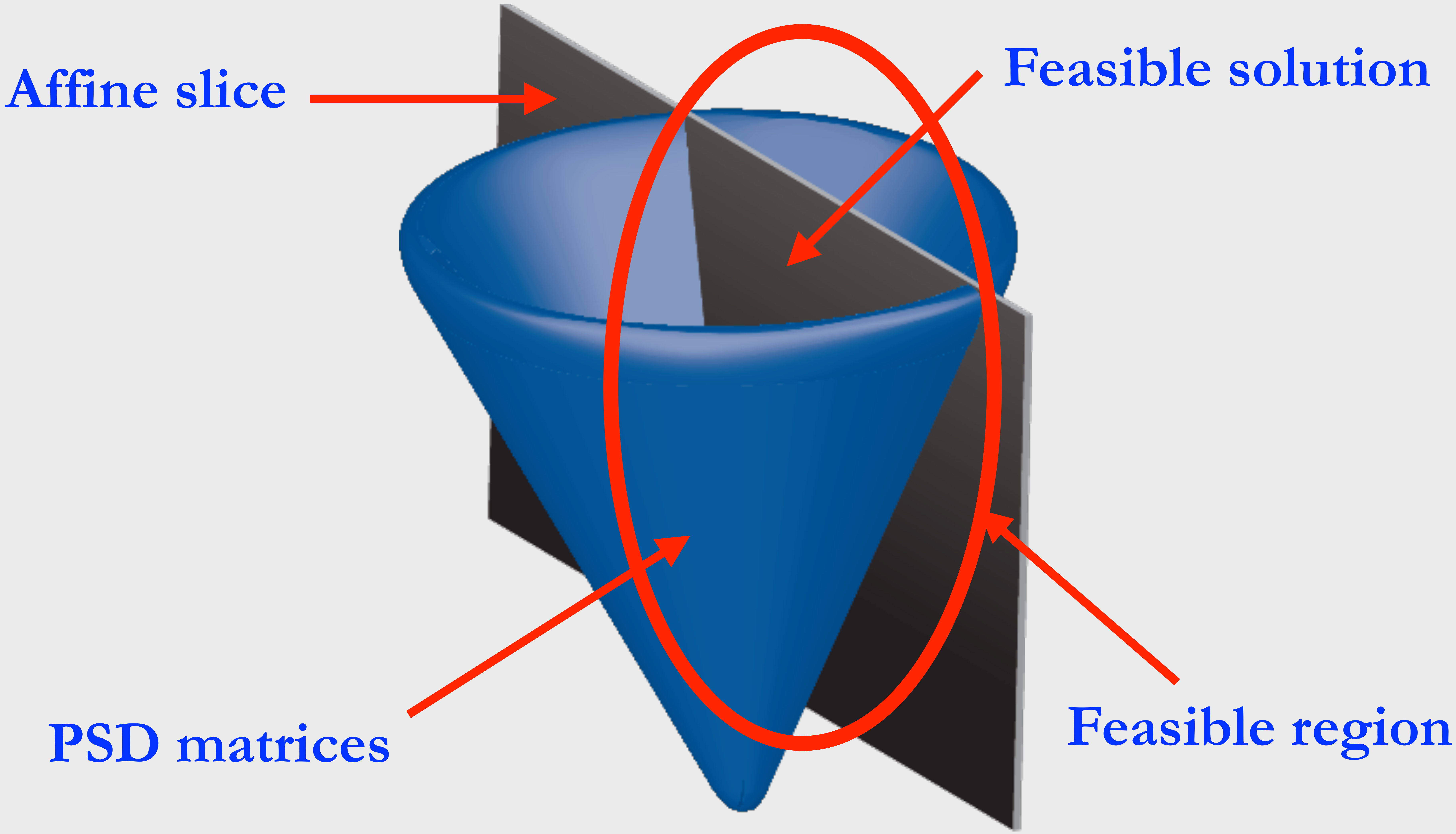
If $\mathscr{A} = \varnothing$, then the SDP is **infeasible**. Otherwise, the SDP is **feasible**.

$X \in \mathscr{A}$ is called **feasible**.

$X \in \mathscr{A} \cap \text{Pd}(\mathscr{X})$, it is called **strictly feasible**.

$\mathscr{A} = \{X \in \text{Pos}(\mathscr{X}) : \Phi(X) = B\}$ is called the **feasible region**.

Geometry



Credit: cvxr.com

SDPs

A semidefinite program (SDP) is an optimization problem of a linear function over a positive semidefinite variable subject to affine constraints. An SDP can be written in standard form as below:

$$\begin{aligned} \alpha = \text{maximize: } & \langle A, X \rangle \\ \text{subject to: } & \Phi(X) = B \\ & X \in \text{Pos}(\mathscr{X}) \end{aligned}$$

If $\mathscr{A} = \varnothing$, i.e., it is infeasible, then $\alpha = -\infty$.

If $\mathscr{A} \neq \varnothing$, i.e., it is feasible, then $\alpha > -\infty$.

If $\alpha = +\infty$ then it is said to be **unbounded**.

If $X \in \mathscr{A}$ satisfies $\langle A, X \rangle = \alpha$, then X is called an **optimal solution**.

(Note that even if α is finite, an optimal solution may not exist!)

Examples

$$\alpha = \text{maximize: } \mathrm{Tr}(X)$$
$$\text{subject to: } X = I_2$$
$$X \in \mathrm{Pos}(\mathbb{C}^2)$$

We have $\mathscr{A} = \{I_2\}$ (and thus feasible)

$\alpha = 2$

The optimal solution is $X = I_2$.

$$\alpha = \text{maximize: } \mathrm{Tr}(X)$$
$$\text{subject to: } X = -I_2$$
$$X \in \mathrm{Pos}(\mathbb{C}^2)$$

We have $\mathscr{A} = \varnothing$ (it is infeasible)

$\alpha = -\infty$

An optimal solution *does not exist*.

$$\alpha = \text{maximize: } \mathrm{Tr}(X)$$
$$\text{subject to: } X \geq I_2$$
$$X \in \mathrm{Pos}(\mathbb{C}^2)$$

We have $\mathscr{A} = \{X \in \mathrm{Pos}(\mathscr{X}) : X \geq I\}$

$\alpha = +\infty$ (the SDP is unbounded).

An optimal solution *does not exist*.

Nomenclature

$$\alpha = \text{minimize: } \langle A, X \rangle$$
$$\text{subject to: } \Phi(X) = B$$
$$X \in \text{Pos}(\mathscr{X})$$

We can minimize as well.

The SDP is unbounded if $\alpha = -\infty$ in this case.

Also, if the SDP is infeasible, then $\alpha = +\infty$.

All the definitions generalize as you'd expect them too.

Weird behaviour

$$\alpha = \text{minimize:} \quad s$$
$$\text{subject to:} \quad \begin{bmatrix} t & 1 \\ 1 & s \end{bmatrix} \in \text{Pos}(\mathbb{C}^2)$$

$(s, t) = (1,1)$ is feasible, thus $\alpha \leq 1$

The facts below imply that $s > 0$, thus $\alpha \geq 0$

$(s, t) = (\epsilon, 1/\epsilon)$, where $\epsilon > 0$, is feasible.

Since s can be made arbitrarily close to 0 we have $\alpha = 0$.

But there does not exist an optimal solution!

Fact: If $\begin{bmatrix} t & b \\ b^* & s \end{bmatrix} \in \text{Pos}(\mathbb{C}^2)$ and $s = 0$, then we must have $b = 0$ as well.

Fact: If $\begin{bmatrix} t & b \\ b^* & s \end{bmatrix} \in \text{Pos}(\mathbb{C}^2)$, then $s, t \geq 0$ and $st \geq |b|^2$

Fact: The converse of the above is true.

Quantum example

$$\begin{aligned} \alpha = \text{maximize: } & \langle H, X \rangle \\ \text{subject to: } & \text{Tr}(X) = 1 \\ & X \in \text{Pos}(\mathscr{X}) \end{aligned}$$

H is Hermitian.

You can think of H as a Hamiltonian and α as its maximum energy (if you are familiar with such things).

We can also write this succinctly, below.

$$\begin{aligned} \alpha = \text{maximize: } & \langle H, X \rangle \\ \text{subject to: } & X \in D(\mathscr{X}) \end{aligned}$$

This is an optimization over quantum states!

Where $D(\mathscr{X}) := \{X \geq 0 : \text{Tr}(X) = 1\}$ are density operators

Quantum example

Given quantum states $\rho_1, \ldots, \rho_n \in D(\mathscr{X})$, consider the SDP:

$$\alpha = \text{maximize: } \frac{1}{n} \sum_{i=1}^{n} \langle \rho_i, M_i \rangle$$

$$\text{subject to: } \sum_{i=1}^{n} M_i = I$$

$$M_i \in \text{Pos}(\mathscr{X})$$

This is an optimization over POVMs.

Quantum example

Given a linear map $\Psi \in L(\mathscr{X}, \mathscr{Y})$ and its Choi representation $C \in L(\mathscr{Y} \otimes \mathscr{X})$, consider the SDP:

maximize: $\langle C, J \rangle$

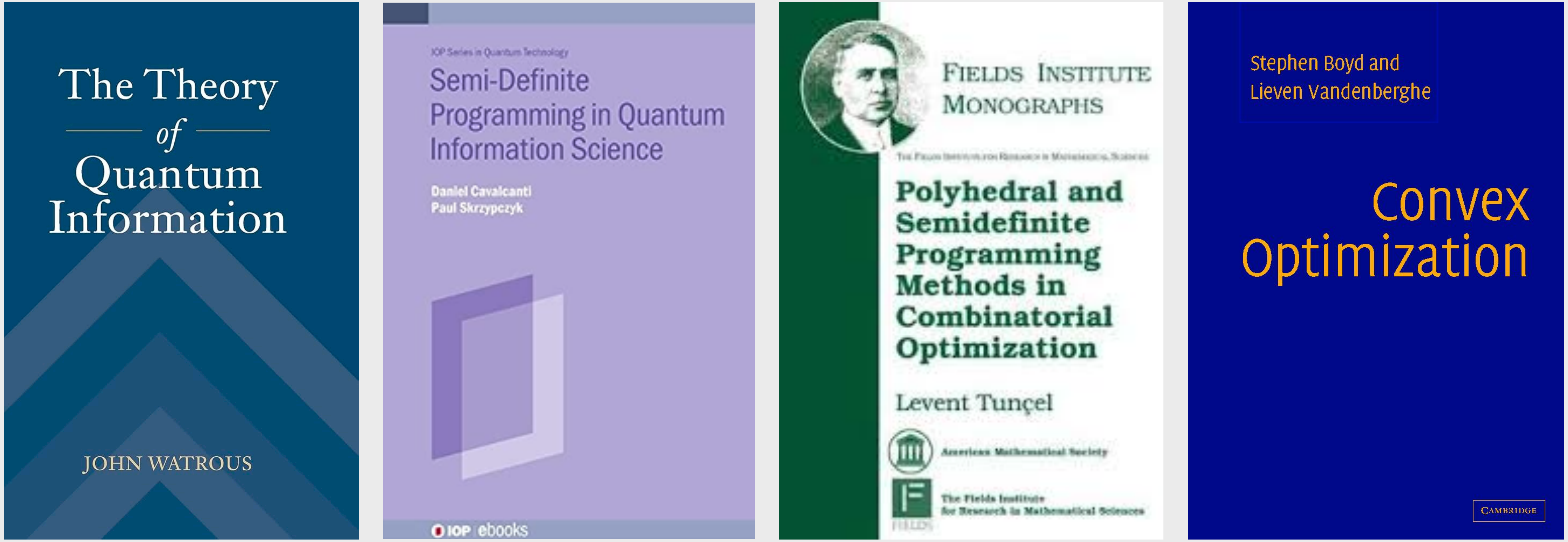
subject to: $\text{Tr}_{\mathscr{Y}}(J) = I_{\mathscr{X}}$

$J \in \text{Pos}(\mathscr{Y} \otimes \mathscr{X})$

This computes the maximum overlap a linear map has with a quantum channel.

References

- [Slides courtesy] Short course by Jamie Sikora at QIPSS School 2023
- Semidefinite programs in quantum information, 2011 (Ashwin Nayak)
- Advanced topics in quantum information theory (John Watrous)



Semidefinite programming in two-party quantum cryptography

Part II : Semidefinite programming for two-party cryptography

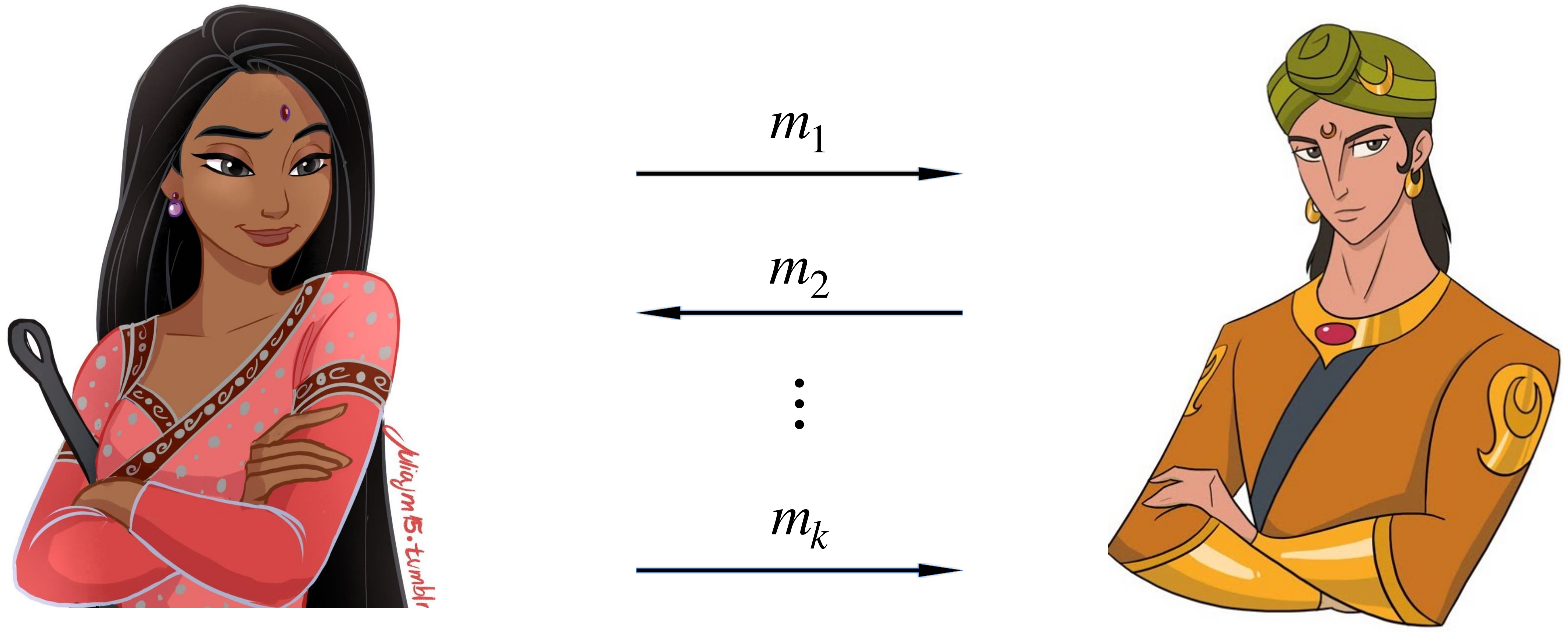
Presenter: Akshay Bansal

Outline of this talk

- Introduction to the two-party setup and security definitions
- Newer protocols for the two-party tasks
- Open questions

Introduction to two-party setup and security definitions

A general two-party cryptography setup



$$P_A = \max_S \Pr\,[\text{Alice successfully cheats}]$$

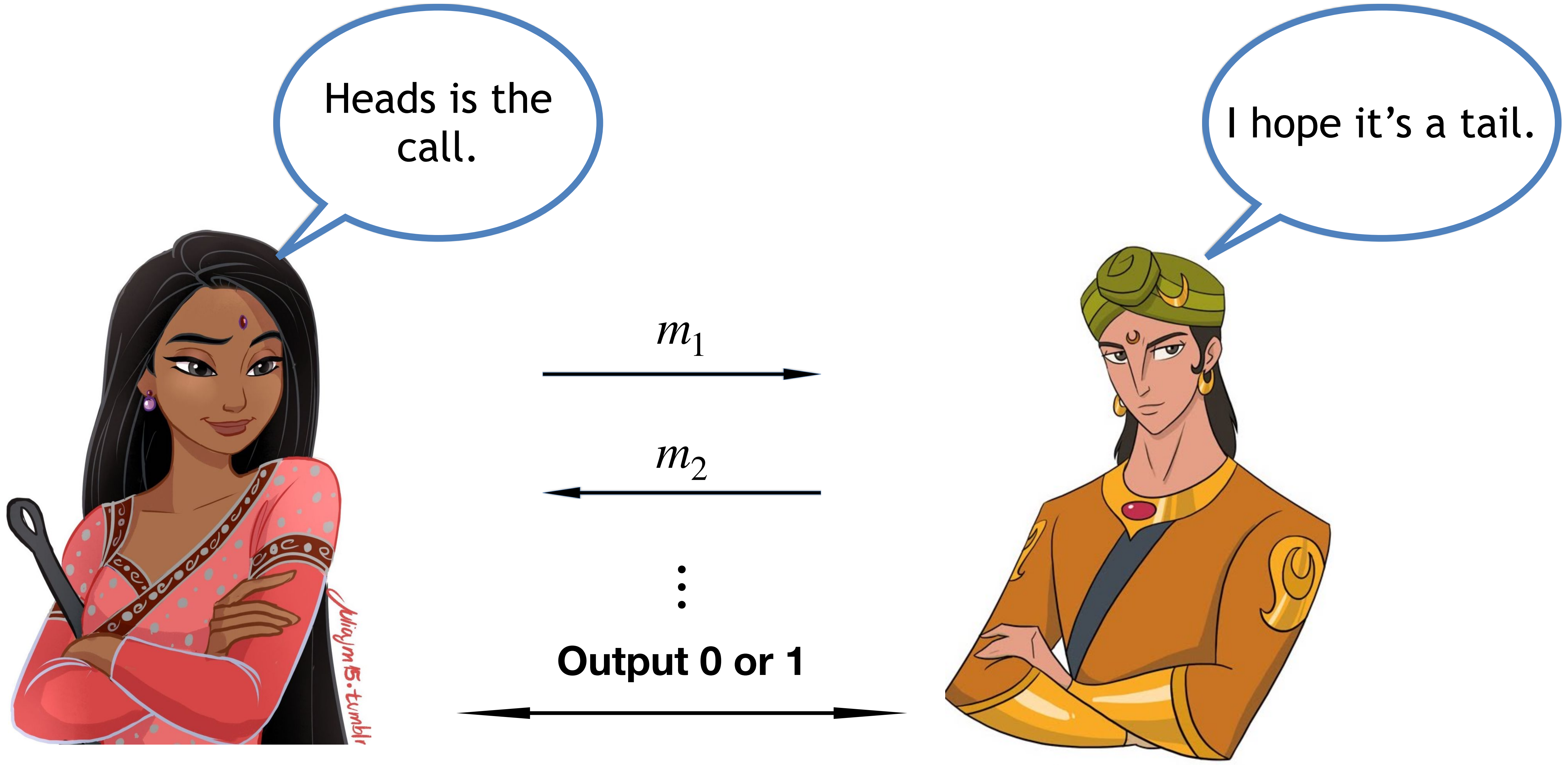
$$P_B = \max_S \Pr\,[\text{Bob successfully cheats}]$$

Security of the protocol $(\mathcal{S}) := \max\{P_A, P_B\}$

Some useful cryptographic primitives

- Coin flipping (weak and strong) - Commitment schemes, etc.
- Oblivious transfer (1-out-of-2, Rabin) - Secure MPC, PIR, secure auctions/voting, etc.
- Bit commitment - Secure coin flipping, ZKP, etc.

The task of coin flipping

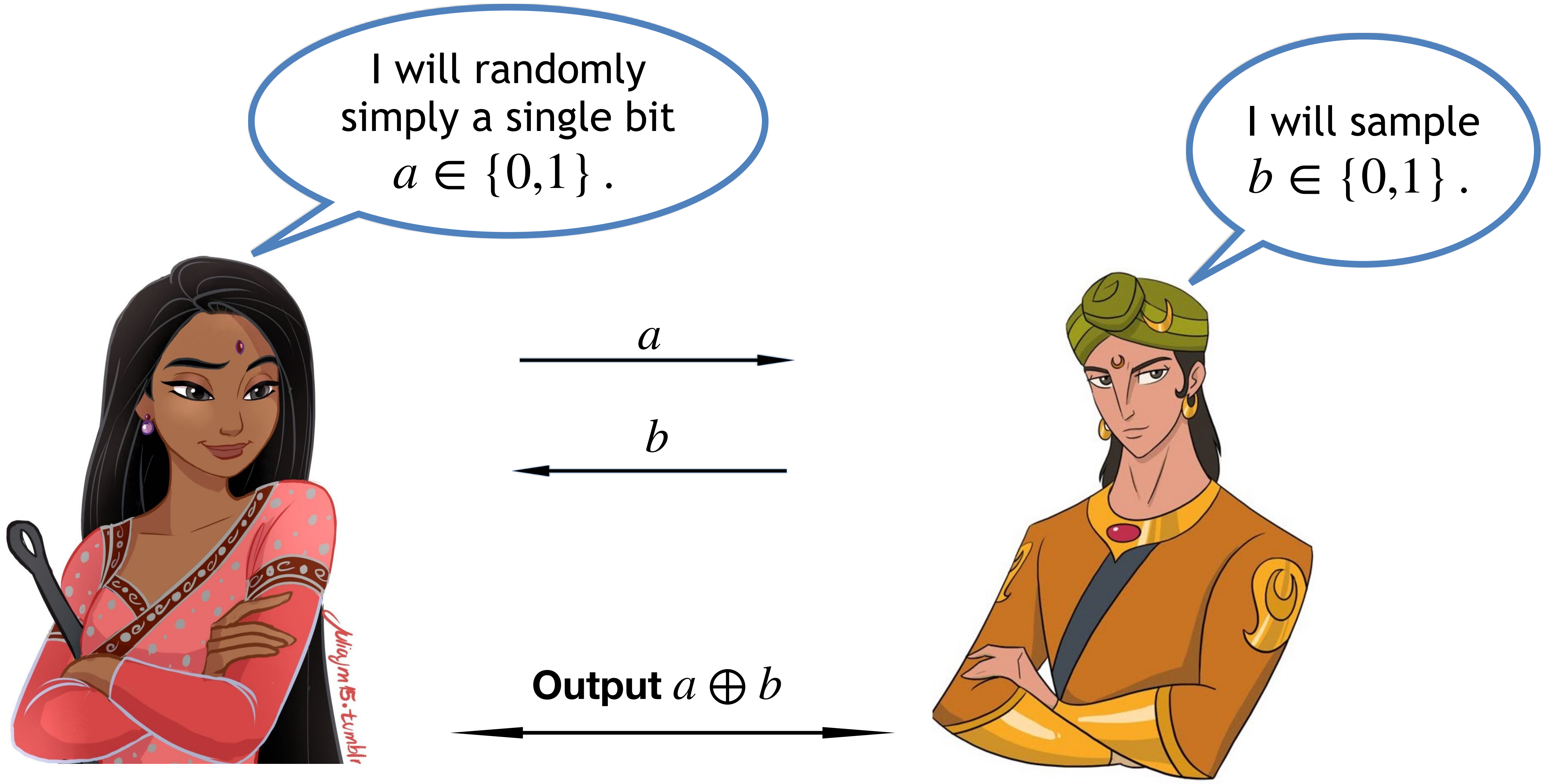


$$P_A = \max_S \Pr \text{[Dishonest Alice successfully forces outcome heads]}$$

$$P_B = \max_S \Pr \text{[Dishonest Bob successfully forces outcome tails]}$$

$$\text{Security of the protocol } (\mathcal{S}) := \max\{P_A, P_B\}$$

A bad coin flipping protocol

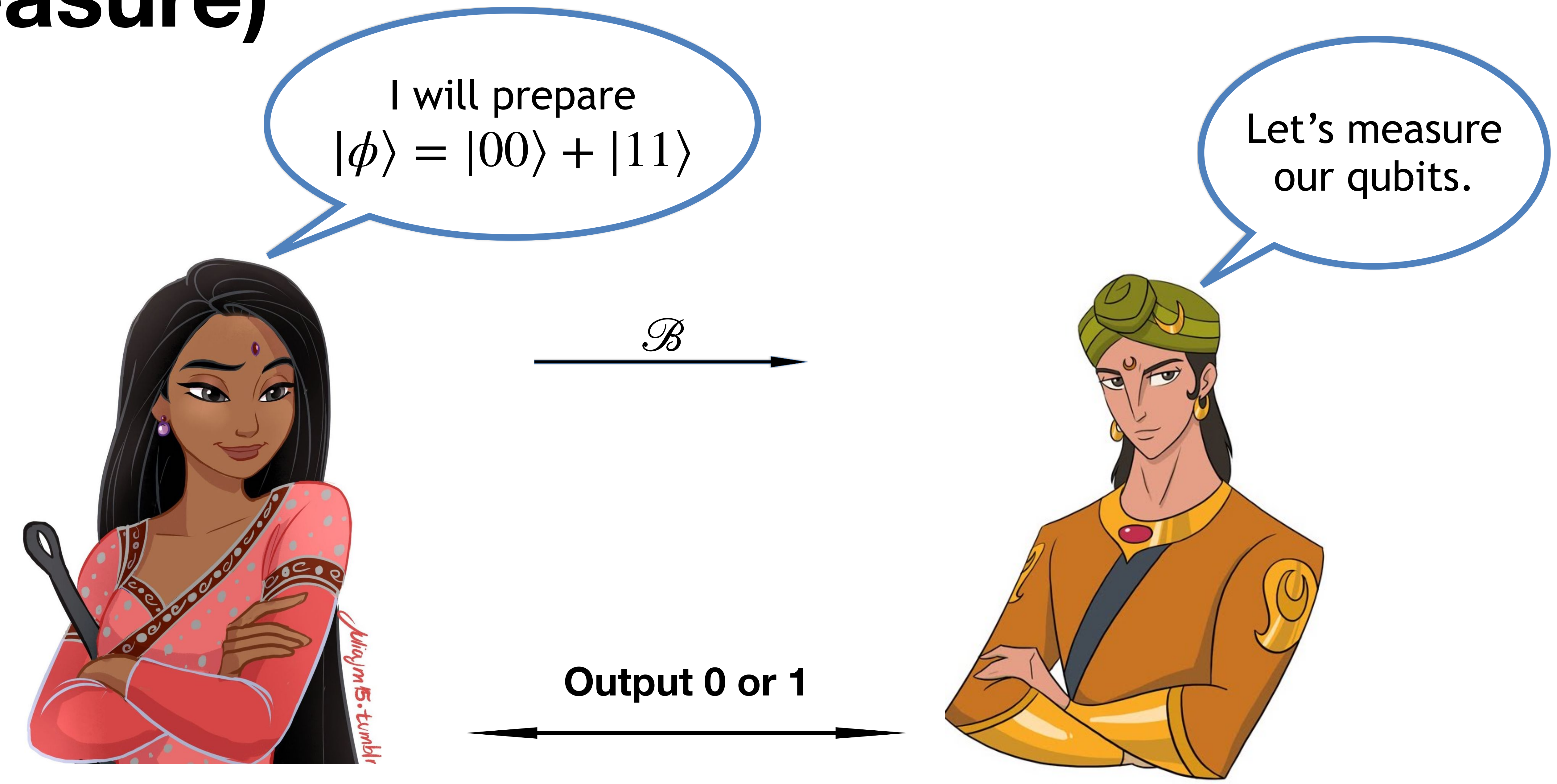


Strategy: Dishonest Bob can simply send $a \oplus 1$

Security of the protocol $(\mathcal{S}) := \max\{P_A, P_B\} = 1$

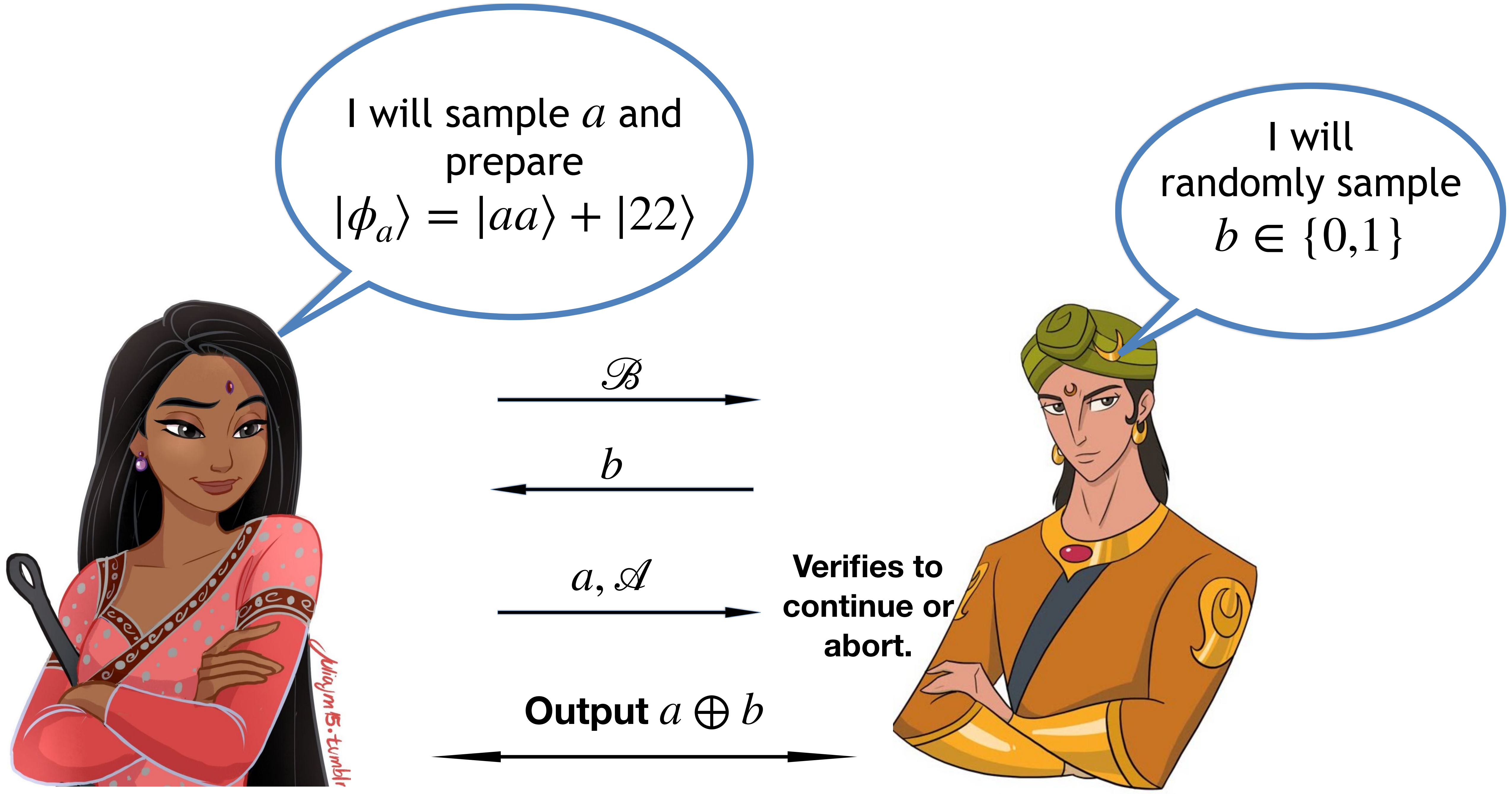
Another bad coin flipping protocol (quantum)

(Prepare-and-measure)



Strategy: Dishonest Alice can simply prepare $|00\rangle$.

A decent coin flipping protocol [Nayak & Shor, 2003]



Strategy: ?

A security analysis using SDPs

Cheating Bob

$$\text{max.} \quad \frac{1}{2}\langle M_0, \mathrm{Tr}_A(|\phi_0\rangle\langle\phi_0|)\rangle + \frac{1}{2}\langle M_1, \mathrm{Tr}_A(|\phi_0\rangle\langle\phi_0|)\rangle$$

subject to:

$$M_0 + M_1 = \mathbb{1},$$

$$M_0, M_1 \succeq 0.$$

Cheating Alice

$$\text{max.} \quad \frac{1}{2}\langle \sigma_0, |\phi_0\rangle\langle\phi_0|\rangle + \frac{1}{2}\langle \sigma_1, |\phi_1\rangle\langle\phi_1|\rangle$$

subject to:

$$\mathrm{Tr}_A(\sigma_0) = \mathrm{Tr}_A(\sigma_1) = \sigma,$$

$$\mathrm{Tr}(\sigma_0) = \mathrm{Tr}(\sigma_1) = 1,$$

$$\sigma_0, \sigma_1, \sigma \succeq 0.$$

Some results on weak coin flipping

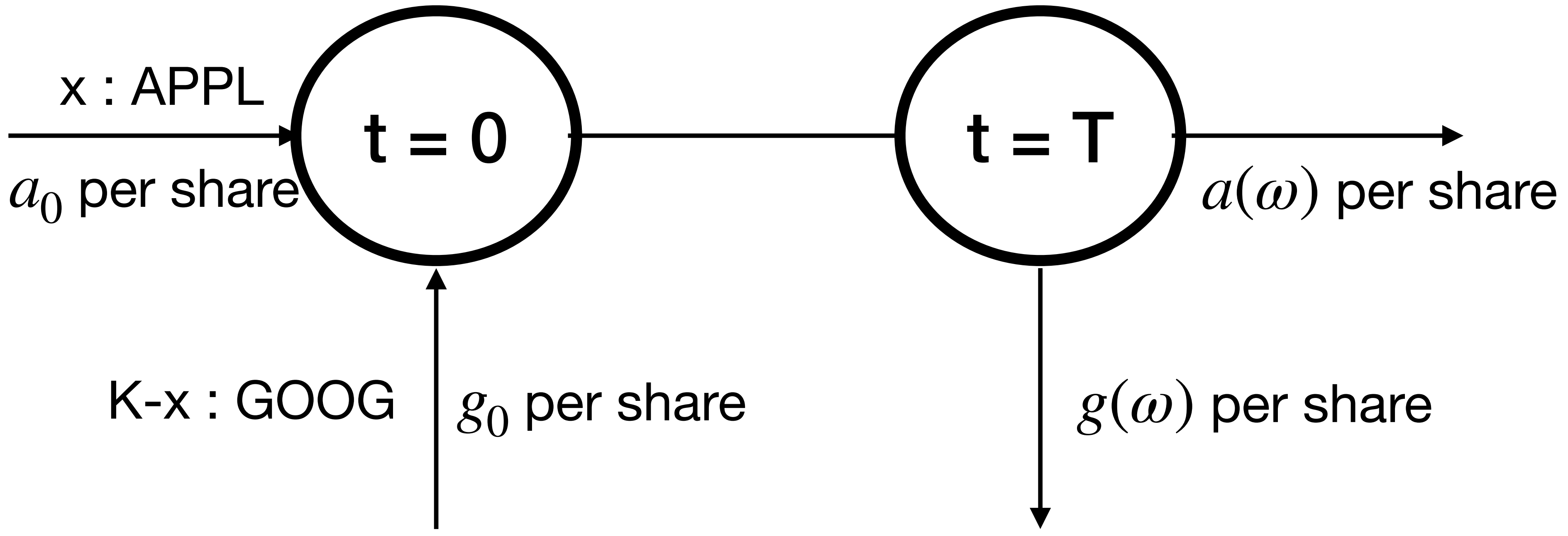
- [Moc07] Given $\epsilon > 0$, there exists quantum protocol with $\max\{P_A^{WCF}, P_B^{WCF}\} < 1/2 + \epsilon$.
- [ARV21] Explicit construction of protocols with arbitrarily small bias.
- [Mil20] Impossibility of efficient weak coin flipping.
- [WH**B**T24] (In)composable security of weak coin flipping.

Newer protocols for the two-party tasks

Stochastic programming

(An classical example from stock investment)

Given a total K number of shares to be invested between two different stocks (under certain constraints), propose a useful investment strategy.



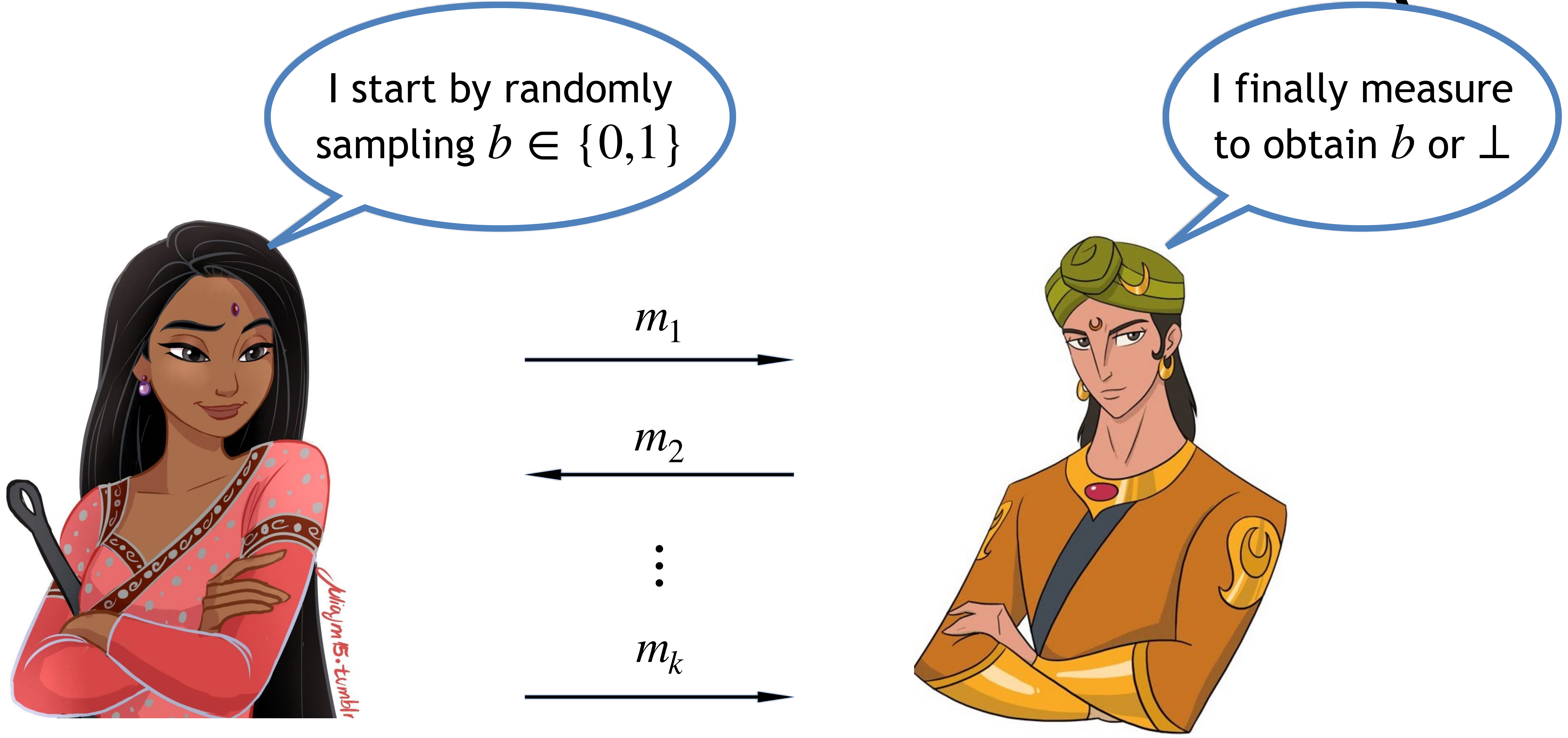
$$s(x) = \begin{bmatrix} x \\ K - x \end{bmatrix} \qquad c_0 = \begin{bmatrix} a_0 \\ g_0 \end{bmatrix} \qquad c(\omega) = \begin{bmatrix} a(\omega) \\ g(\omega) \end{bmatrix}$$

$$\max_{x} \mathbb{E}[c(\omega)^T s(x)] - c_0^T s(x)$$

subject to: $\quad s(x) \in \mathcal{S}(\omega)$

Rabin oblivious transfer

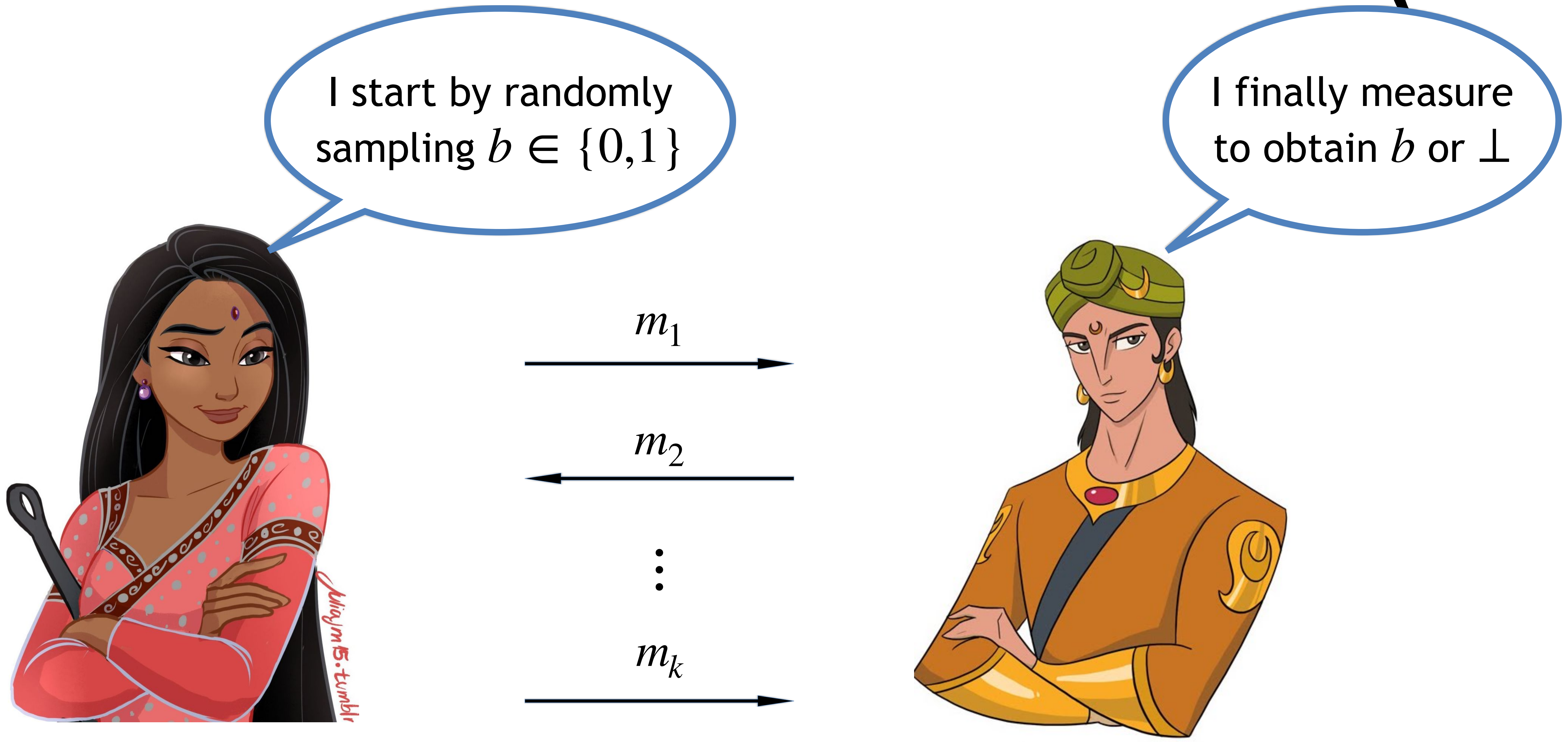
How to exchange secrets with oblivious transfer? (Rabin, 1981)



Rabin oblivious transfer is the cryptographic task where Alice sends a bit $b \in \{0,1\}$ to Bob which he receives with probability $1/2$ and with the probability $1/2$ he receives $\perp$ indicating that the bit was lost.

Rabin oblivious transfer

How to exchange secrets with oblivious transfer? (Rabin, 1981)

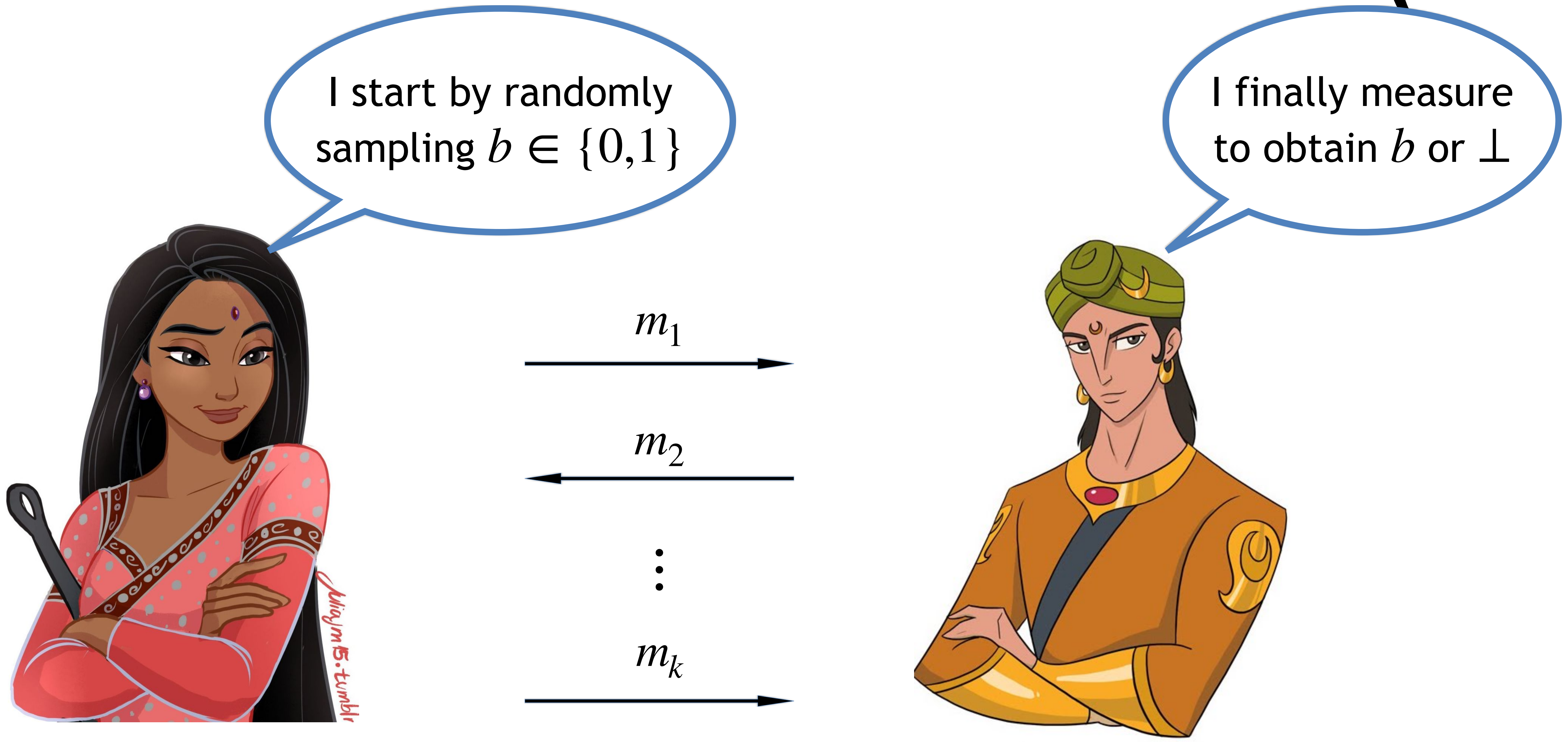


$$P_A^{ROT}(\mathscr{P}) = \max_S \Pr\,[\text{Alice correctly guesses whether Bob } \textit{asserts } b \text{ or } \perp]$$

$$P_B^{ROT}(\mathscr{P}) = \max_S \Pr\,[\text{Bob correctly guesses } b]$$

Rabin oblivious transfer

How to exchange secrets with oblivious transfer? (Rabin, 1981)



$$\mathcal{S}(\mathcal{P}) := \max\{P_A^{ROT}(\mathcal{P}), P_B^{ROT}(\mathcal{P})\}$$

Motivation: Almost nothing is known about the security of Rabin oblivious transfer task under the regime of unconditional security.

A bad Rabin-OT protocol

(Prepare-and-measure)

I will sample $b \in \{0,1\}$ and create $|\phi\rangle = |b\rangle + |\perp\rangle$

$\mathscr{B}$ →

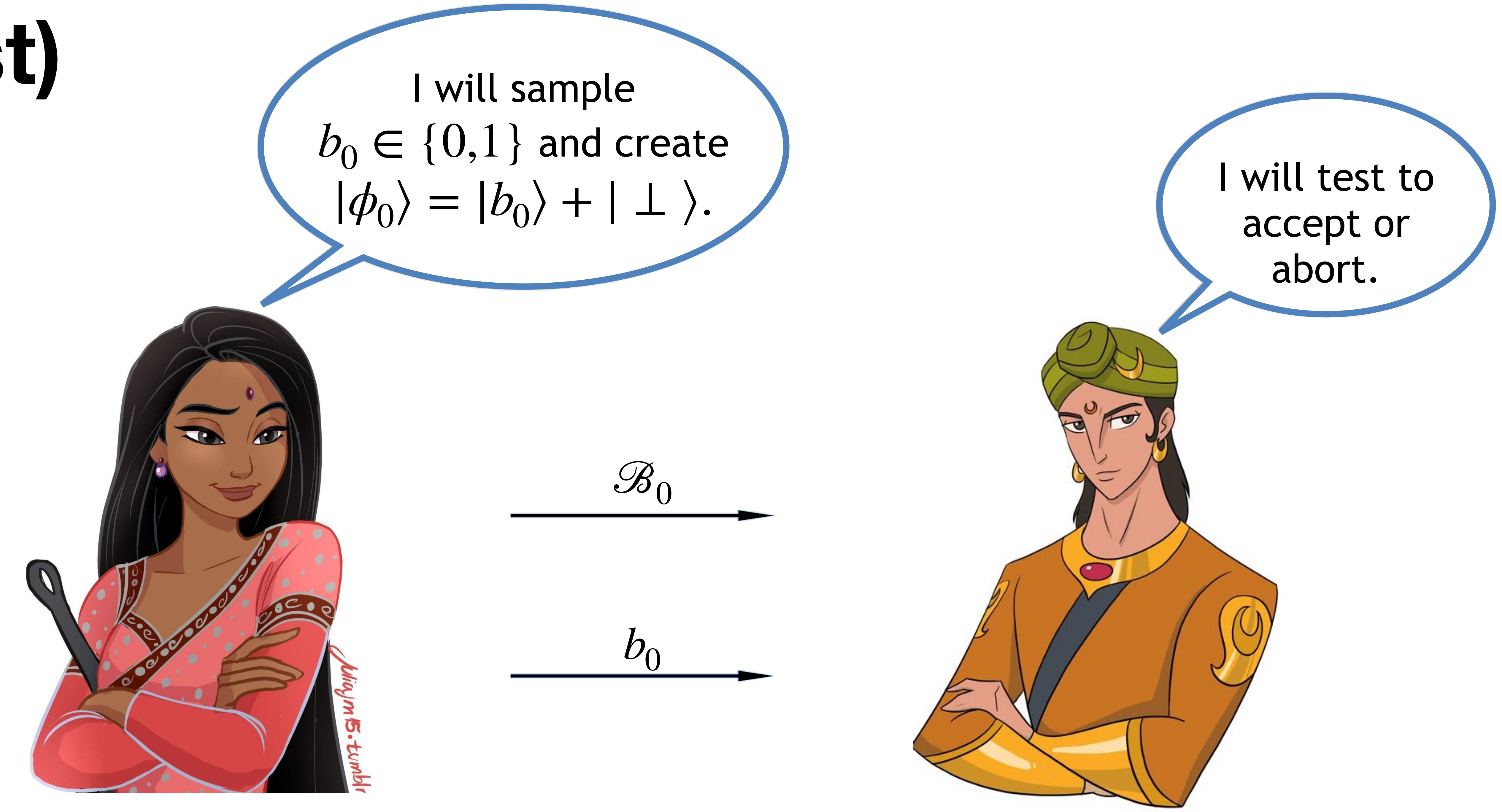
Let me measure the qutrit.

Output b **or** $\perp$

Strategy: Dishonest Alice can simply send $|\perp\rangle$.

Another bad Rabin-OT protocol

(Prepare-and-test)



Another bad Rabin-OT protocol

(Prepare-and-test)

If Bob accepts, I will sample $b_1 \in \{0,1\}$ and create $|\phi_1\rangle = |b_1\rangle + |\perp\rangle$.

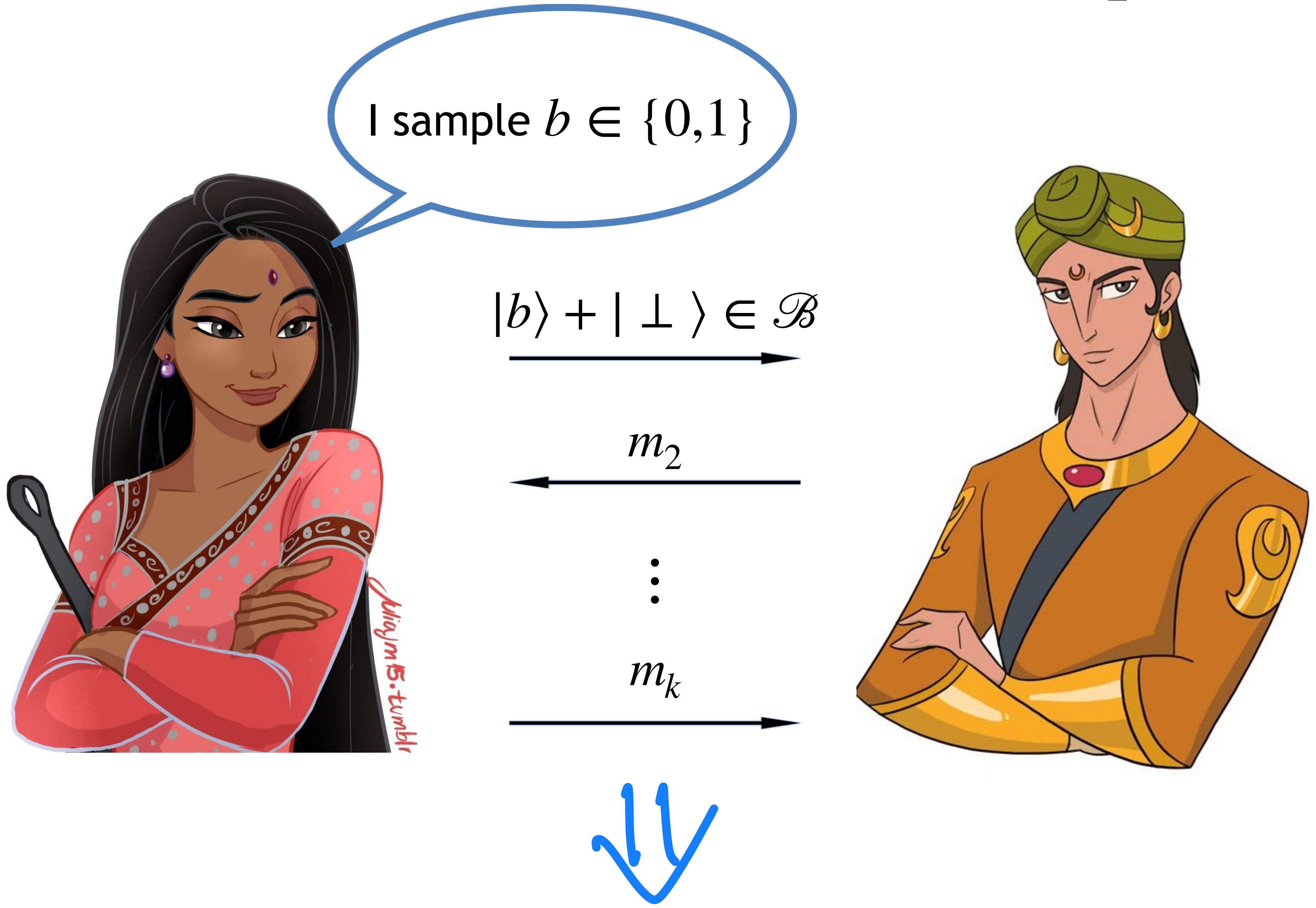
$\xrightarrow{\mathscr{B}_1}$

I finally measure to obtain b_1 or $\perp$.

Strategy: Dishonest Alice can prepare $|\phi_0\rangle$ to always accept and send $|\perp\rangle$ next.

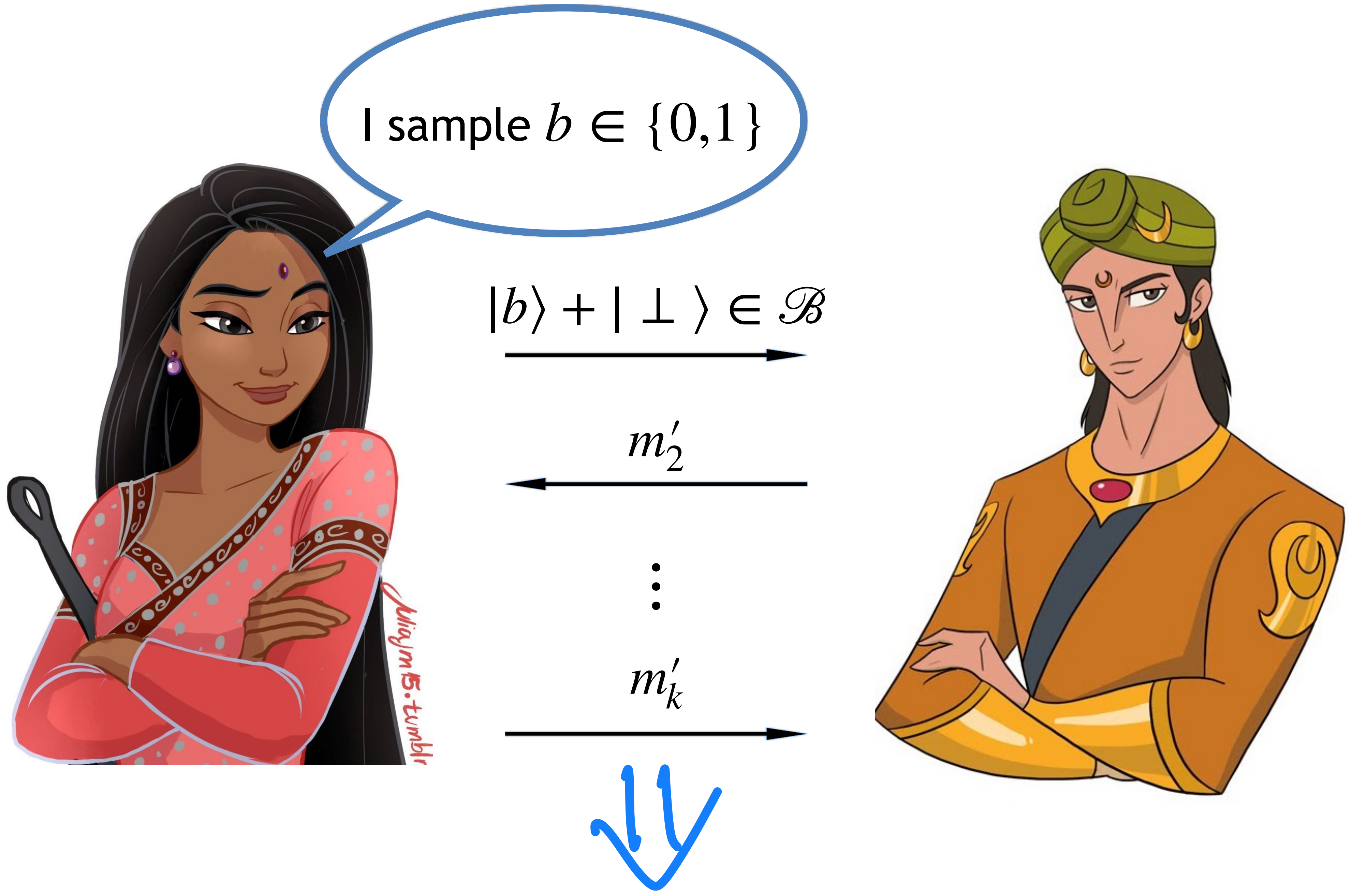
Some bad Rabin OT protocols

Prepare-and-measure ($\mathscr{P}_1$)



$$P_A(\mathscr{P}_1) = 1$$

Prepare-and-test ($\mathscr{P}_2$)

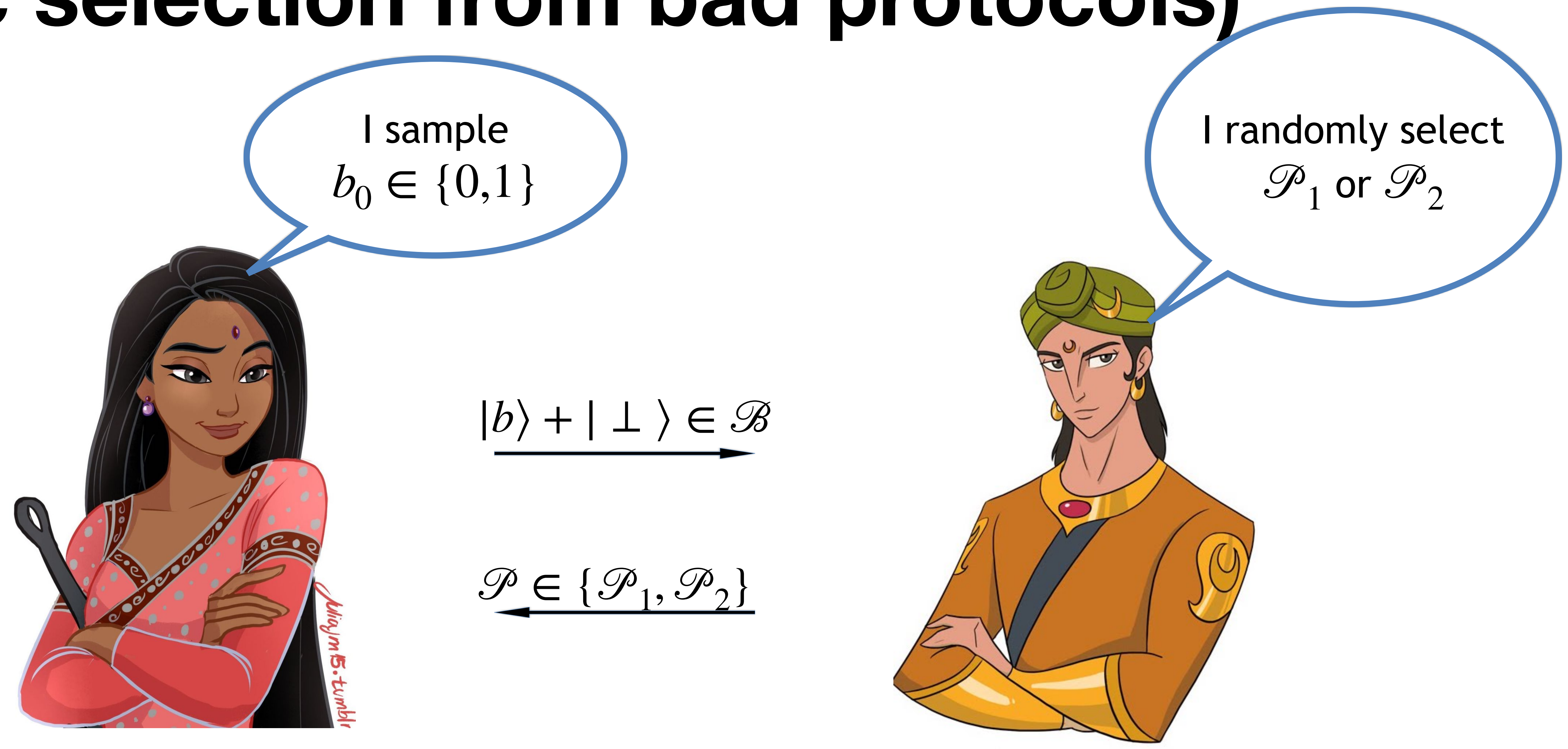


$$P_A(\mathscr{P}_2) = 1$$

Alice can cheat perfectly in both $\mathscr{P}_1$ and $\mathscr{P}_2$.

A useful Rabin-OT protocol

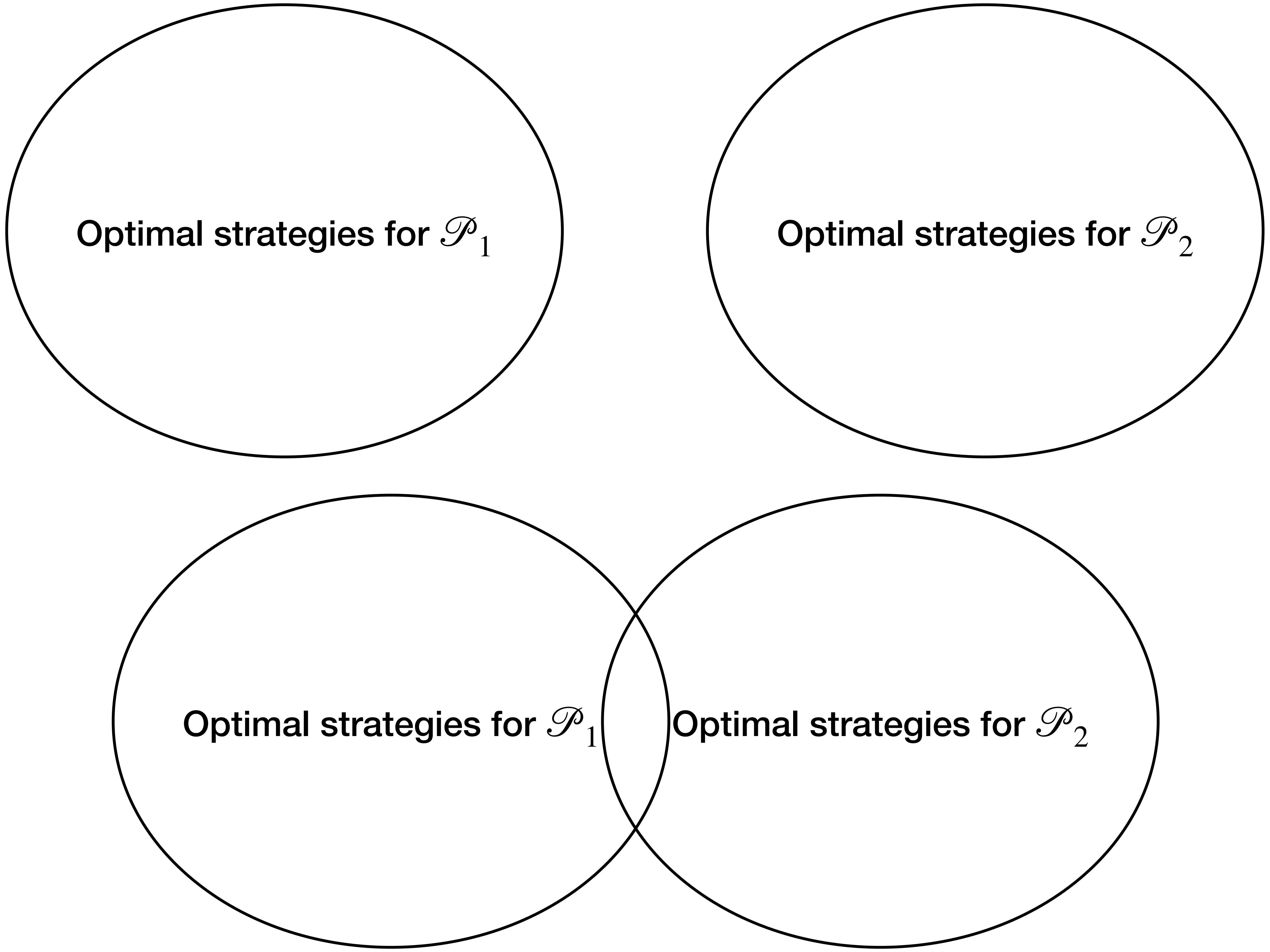
(Using stochastic selection from bad protocols)



Theorem [**B**S25]: There exists a quantum protocol for Rabin OT where Alice can correctly guess whether Bob received the message or $\perp$ with probability at most 0.9330 and Bob can learn Alice's bit with probability at most 0.9691 implying

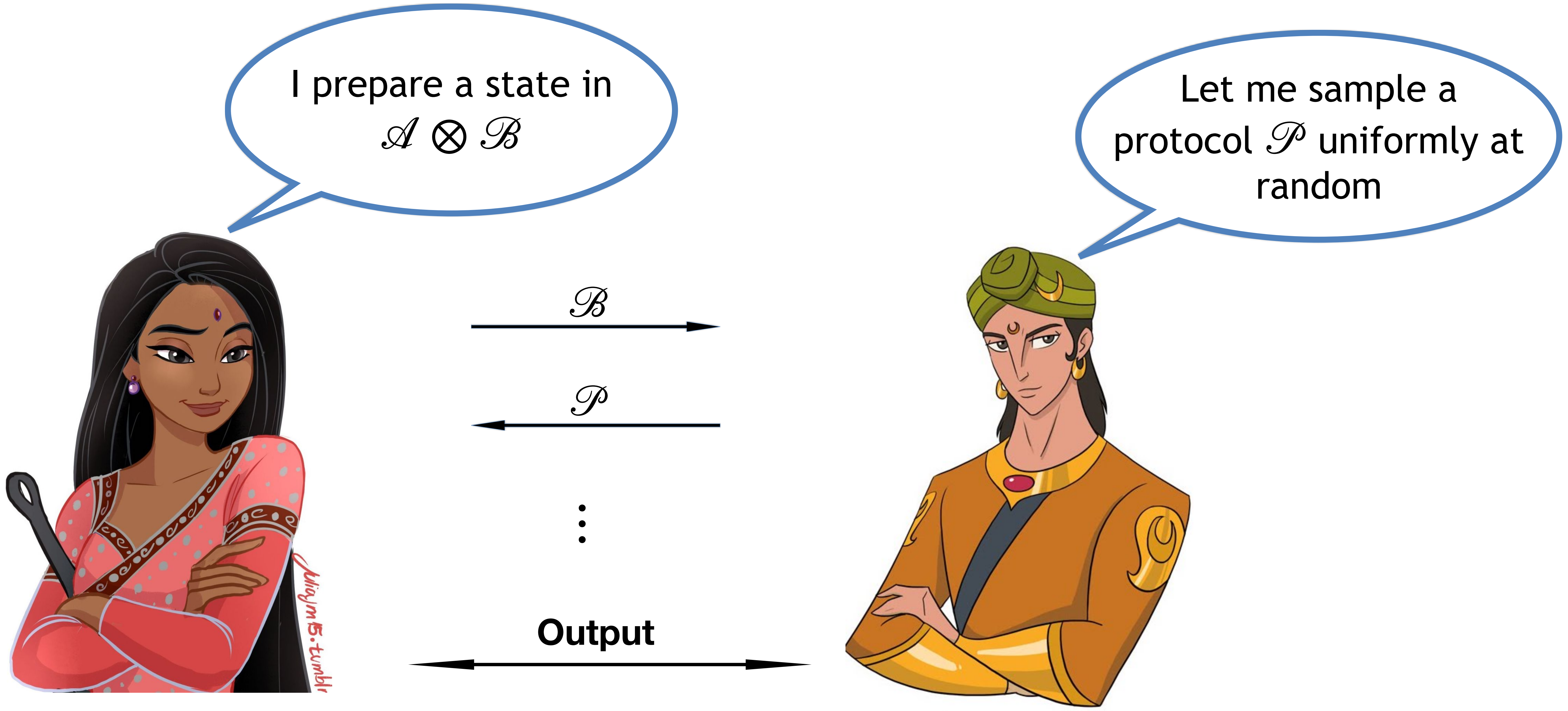
$$\max\{P_A^{ROT}, P_B^{ROT}\} = 0.9691 < 1$$

An optimization viewpoint



Fact: The security of the protocol with stochastic selection is strictly better than the constituent protocols iff the optimal strategies do not overlap.

General stochastic selection setup



$$P_A = \max_S \Pr\,[\text{Alice cheats successfully}] = \max \sum p_j \, P_A^{(j)}$$

$$P_B = \max_S \Pr\,[\text{Bob cheats successfully}] = \max \sum \Pr[j] \, P_B^{(j)}$$

Cheating Alice in stochastic selection (2/3)

Protocol 1

$$\max \langle C_1, Y_1 \rangle$$
$$\Phi(Y_1) = B_1$$
$$\Xi(Y_1) = X_1$$
$$X_1, Y_1 \succcurlyeq 0$$

Protocol 2

$$\max \langle C_2, Y_2 \rangle$$
$$\Phi(Y_2) = B_2$$
$$\Xi(Y_2) = X_2$$
$$X_2, Y_2 \succcurlyeq 0$$

$$\boxed{X_1 = X_2}$$

Cheating Alice in stochastic selection (3/3)

$$\begin{aligned} \max \ & \mathbb{E}_\omega[\langle C_\omega, Y_\omega \rangle] \\ & \Phi(Y_\omega) = B_\omega, \ \forall \omega \\ & \Xi(Y_\omega) = X, \ \forall \omega \\ & Y_\omega \succcurlyeq 0, \ \forall \omega \\ & X \succcurlyeq 0. \end{aligned}$$

Note: For large $|\Omega|$, use techniques based on Benders decomposition.

Some open questions

- Protocols with optimal communication complexity for WCF.
- Optimality of [CK09] and bounds on communication complexity for SCF.
- Secure device independent weak coin flipping protocol [**B**AHS24]
- Optimal protocols and lower bounds for 1-out-of-2-OT and Rabin OT [A**B**SW25].
- Composability of oblivious transfer (Ongoing work with Wu)

References

[ARV21] Arora, Atul Singh, Jérémie Roland, and Chrysoula Vlachou. "Analytic quantum weak coin flipping protocols with arbitrarily small bias." *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, 2021.

[CK09] Chailloux, André, and Iordanis Kerenidis. "Optimal quantum strong coin flipping." *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2009.

[CGS13] Chailloux, André, Gus Gutoski, and Jamie Sikora. "Optimal bounds for semi-honest quantum oblivious transfer." *Chicago Journal of Theoretical Computer Science* 13 (2016): 1-17.

[CKS10] Chailloux, André, Iordanis Kerenidis, and Jamie Sikora. "Lower bounds for quantum oblivious transfer." *Quantum Information & Computation* 13.1-2 (2013): 158-177.

[Kit02] Alexei Kitaev. Quantum coin flipping. Unpublished result. Talk at the 6th Annual workshop on Quantum Information Processing (QIP 2003), 2002.

[KN03] Nayak, Ashwin, and Peter Shor. "Bit-commitment-based quantum coin flipping." *Physical Review A* 67.1 (2003): 012304.

[Mil20] Miller, Carl A. "The impossibility of efficient quantum weak coin flipping." *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020.

[Moc07] Mochon, Carlos. "Quantum weak coin flipping with arbitrarily small bias." *arXiv preprint arXiv:0711.4114* (2007).

[Lo98] Lo, Hoi-Kwong. "Insecurity of quantum secure computations." *Physical Review A* 56.2 (1997): 1154.

[WHBT25] Wu, Jiawei, Yanglin Hu, Akshay Bansal, and Marco Tomamichel. "On the composable security of weak coin flipping." *arXiv preprint arXiv:2402.15233* (2024).