Self-testing in Cryptography.

Tina Zhang

Based on joint work with Anand Natarajan and Tony Metger

Algorithms: try to prove that problems are <u>easy</u>.

Complexity: try to prove that problems are <u>hard</u>.

Crypto: try to prove that problems are <u>hard</u> based on assumptions.

<u>Complexity classes</u>: groups of problems with "similar" difficulty.

People often talk about:

P (time class)

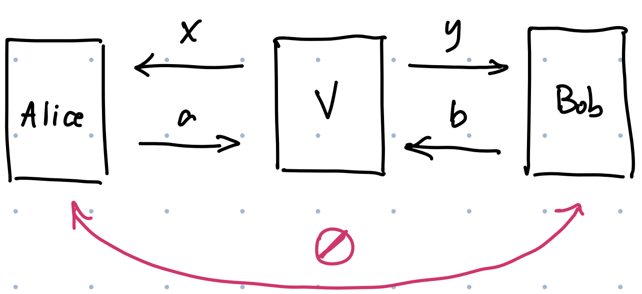
NP (verification class)

BPP (randomised time class)

In this talk we'll need:

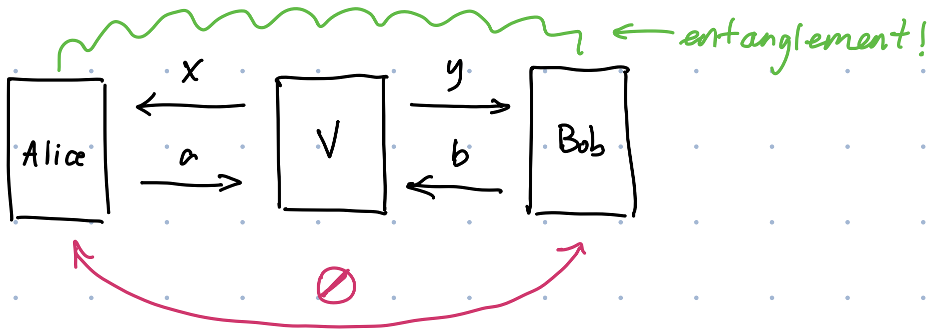
BQP (quantum time class)

MIP: (verification class)



(2 prover 1 round)

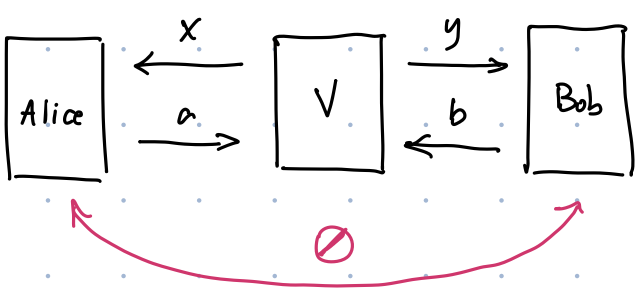
MIP* : (verification class)



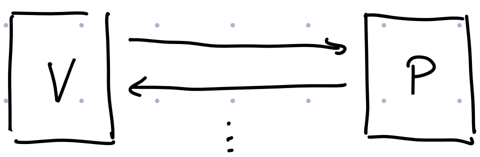
* MIP is a very foundational model in classical complexity (studying it led to PCP thm. and other things)

* However, no-communication may be somewhat difficult to enforce

In crypto, we prefer to consider a single prover who is cryptographically bounded (rather than 2 provers who can't communicate)



MIP



Crypto

* But there are lots of ideas complexity people have developed in MIP world which cryptographers might hope to apply in crypto world

Idea: "compilation"

↳ Use crypto to "simulate" the no-communication assumption

Cryptographic preliminary: HE
(homomorphic encryption)

Normal public-key encryption:

$\mathrm{Enc}(pk, m) \to c$

$\mathrm{Dec}(sk, c) \to m$ (suppressing randomness)

Homomorphic encryption adds one more algorithm:

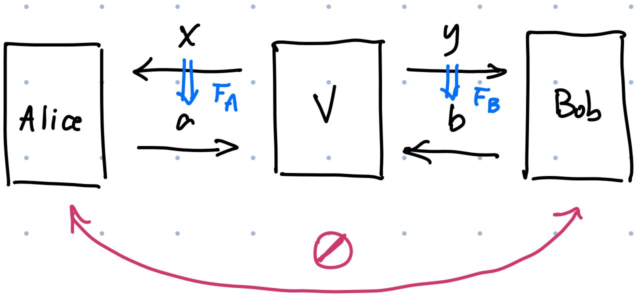
$\mathrm{Eval}(pk, c, F) \to \mathrm{Enc}_{pk}(F(m))$

↳ encrypts m

Does not violate encryption security because evaluator cannot decrypt.

Details are annoying, constructions are subtle and delicate, but primitive is intuitive and easy to work with.

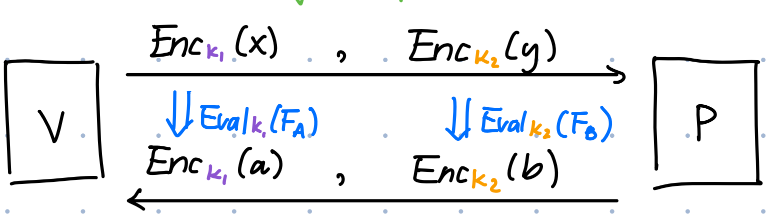
Compilation, attempt #1 → We want to preserve classical (for now) completeness and soundness



completeness: every nonlocal strategy has a corresponding compiled strategy with a value at least as high

soundness: any cheating compiled strategy can be mimicked in the nonlocal world

↓ compile

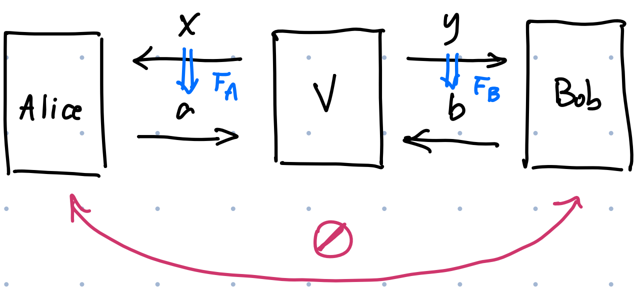


Encryption security "simulates" no-communication

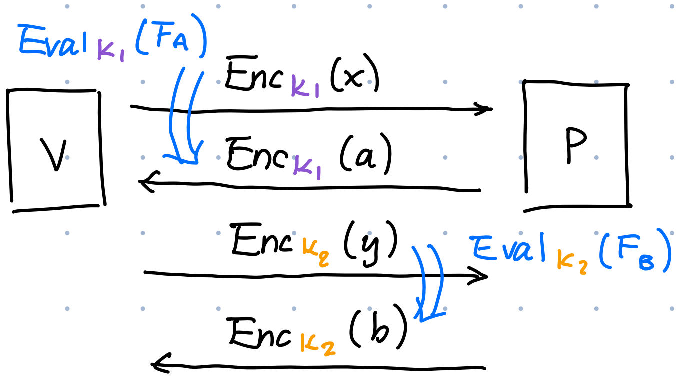
* This attempt fails in an interesting way:
P can simulate any non-signalling Alice/Bob strategy.
(non-signalling: even more general than quantum entanglement)

* Turns out this also preserves non-signalling soundness [KRR '14]

Compilation, attempt #2.



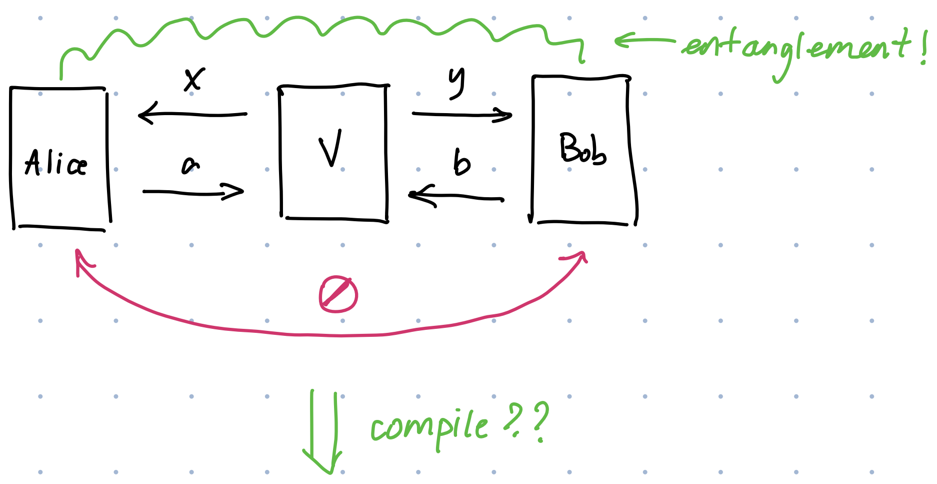
↓ compile



Encryption security "simulates" no-communication
^ and round structure

This works! [KLVY '22] : preserves classical completeness & soundness

What about quantum entangled completeness & soundness?



But why would you care?

Quantum verification.

* Setting: "quantum feudalism"
* Someone claims they solved a problem for you using their quantum computer. How do you know they solved it correctly?
* Some problems in BQP, like factoring, are in NP ⟹ answers are easy to verify

  Others are not, however (consider forrelation)
* Quantum verification: design a protocol by which they can prove (interactively) to you that the problem was solved correctly, where
  * they run in QPT,
  * you run in PPT.

Known results:

* In the 2-prover entangled model this is possible! [RUV '13]
* In the single prover model this is possible assuming quantum computers cannot solve LWE. [Mah '18]

  ↓ big result, uses carefully tailored crypto

* Hold on...

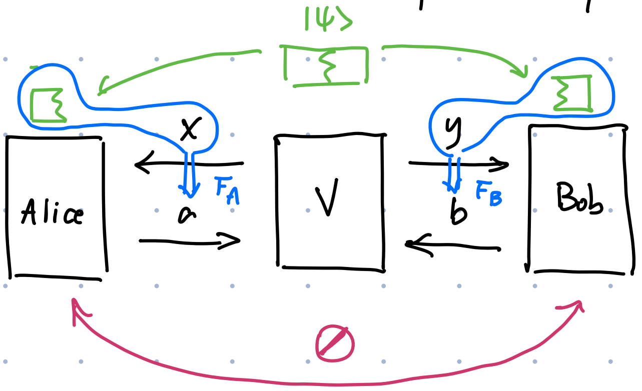
If we have compilation for MIP* and not just MIP protocols, why don't we just compile the 2-prover entangled verification protocol?

More modular + other advantages (may discuss later)

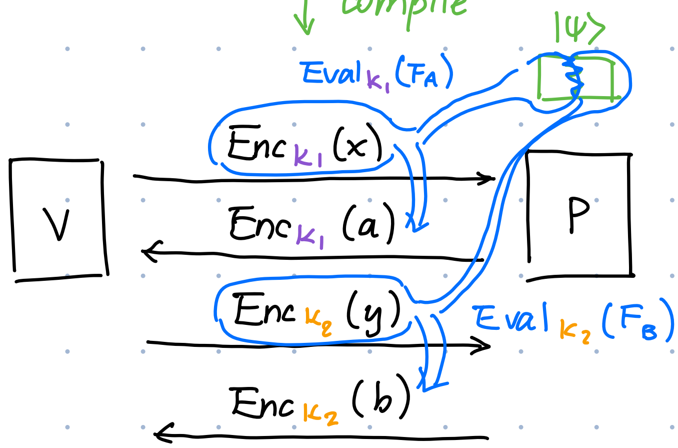
Turns out KLVY works for MIP* protocols too!

"

preserves quantum completeness.



↓ compile



With good enough QHE this will work.

Quantum soundness?

Not known in general. → Recent result shows KLVY preserves quantum soundness in the limit as security parameter goes to ∞ [KMP'24]

Unfortunately this does not give you explicit cryptographic security

So let's take a step back: what exactly do we need to make verification work?

Intuition: as a totally classical verifier, want to somehow force the quantum prover to do the quantum computation honestly.

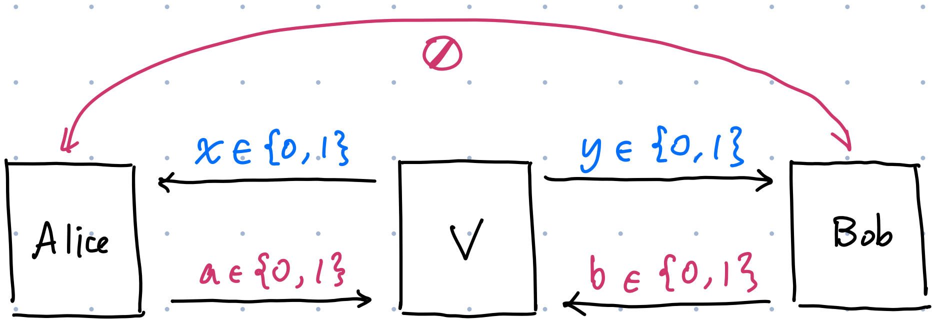
(You know the circuit you want it to run, e.g. the circuit for Shor's alg.; you just don't have the power to run or simulate this circuit yourself.)

Let's start with a very simple baby case:

we'll try to make the prover measure in the X and Z bases honestly.

→ or even just any anticommuting bases

The CHSH game (a particular MIP* protocol)



Win condition: $x \cdot y = a \oplus b$.

If $x = y = 1$, Alice and Bob should disagree

In all other cases they should agree

1. Classical (max) winning probability: $\frac{3}{4}$
2. Quantum winning probability: $\cos^2(\frac{\pi}{8})$ ($\approx 0.85 > 0.75$) ← certification of quantumness
3. There is a unique quantum winning strategy

(characterised by the algebraic relations between the measurement operators Alice uses and the measurement operators Bob uses, as well as their shared entangled state: a single EPR pair)

4. This unique strategy involves Bob measuring 2 anticommuting operators!

We [NZ'23] were able to show properties 2–4 hold for KLVY-compiled CHSH as well (making the correct definitions of Alice and Bob &c.)

Tldr: we can, by playing compiled CHSH with our single prover and checking that it wins w.p. $\cos^2(\frac{\pi}{8})$, force it to measure 2 anticommuting operators.

And actually... it turns out that this "baby case" is pretty much the general case.

(Kitaev circuit-to-Hamiltonian reduction + XZ gadgets)

Summary & discussion.

* [NZ'23]: recovers seminal result of [Mah'18] with a different, more modular approach.
  Also uses weaker assumptions!
* [MNZ'24]: combines advantages of self-testing techniques and crypto techniques to get succinct arguments for QMA from standard assumptions
* Open, approachable problem: linear-time verification.