SECURE IDENTITY-BASED ENCRYPTION IN THE QUANTUM RANDOM ORACLE MODEL

Mark Zhandry – Stanford University

Random Oracle Model (ROM)

- Sometimes, we can’t prove a scheme secure in the standard model.
- Instead, model a hash function as a random oracle, and prove security in this model [BR 1993]

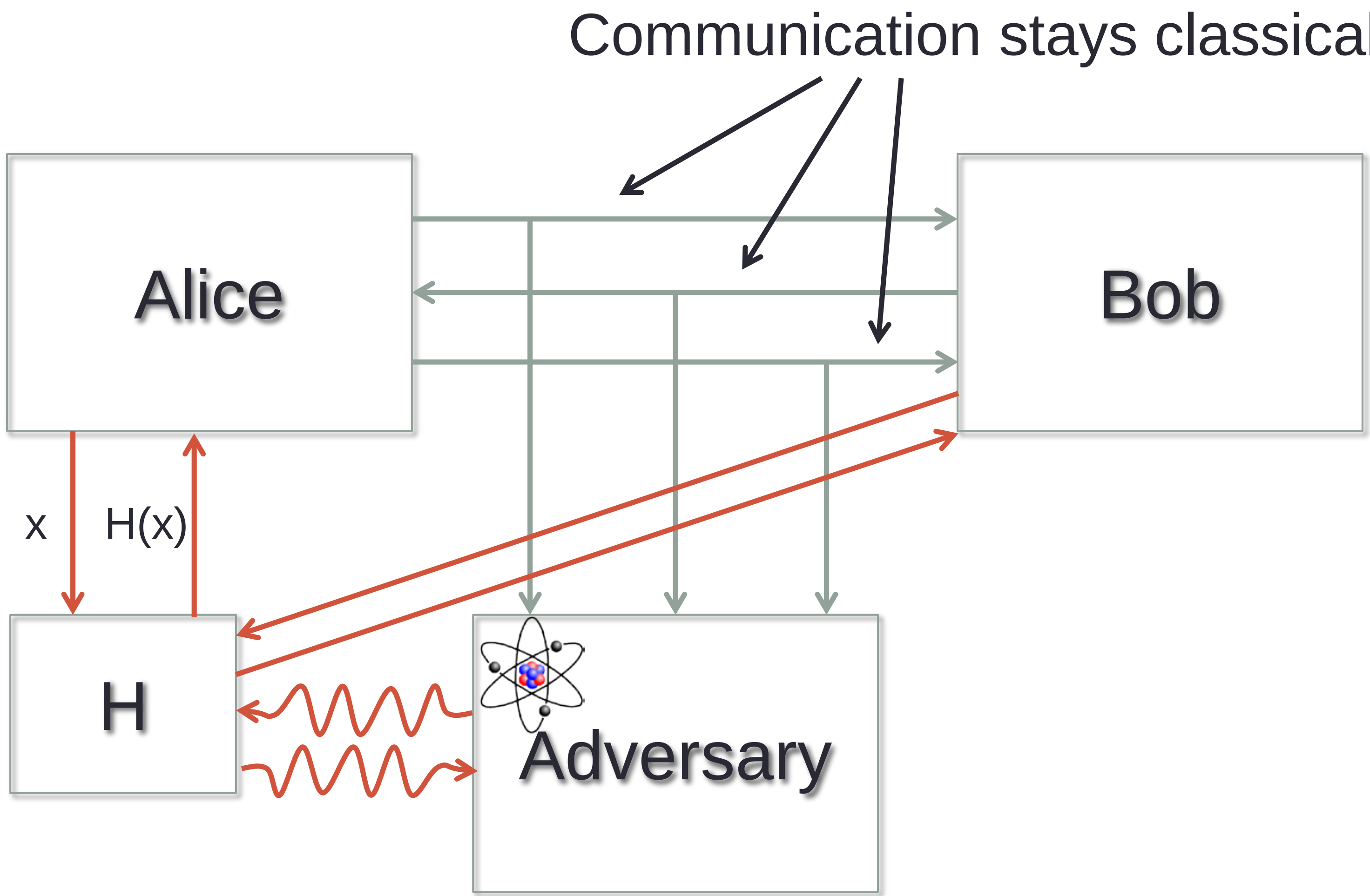
Why Use the Random Oracle Model?

- Most efficient schemes are often only proved secure in the random oracle model
- True even in post-quantum world
  - RO-based GPV signatures more efficient that non-RO CHKP and ABB signatures [GPV 2009, CHKP 2010, ABB 2010]
  - RO-based Hierarchical IBE more efficient than non-RO versions
- Unfortunately, these schemes are only proved secure in the classical ROM
  - Only consider classical queries to the random oracle

The Quantum Random Oracle Model

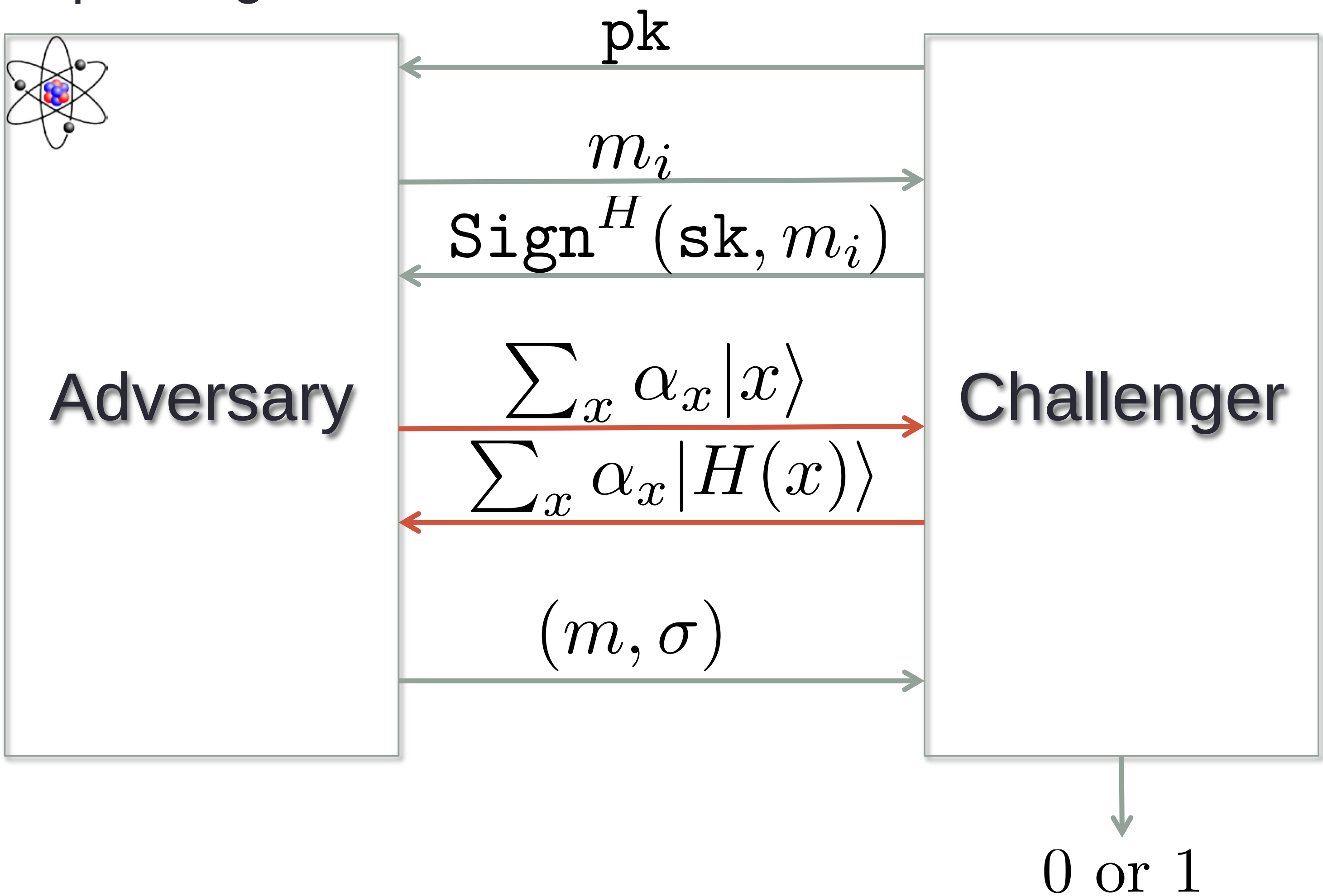
- Interaction with primitives is still classical
- Allow quantum queries to random oracle
  - When instantiated, random oracle replaced with hash function
  - Code for hash function is part of specification
  - Adversary can evaluate hash function on quantum superposition

The Quantum Random Oracle Model (QROM)



Security in the QROM

Example: Signatures



Security Proofs in the QROM

- Classical random oracle model security proofs do not carry over to the quantum setting
- Difficulties:
  - Simulating the random oracle
  - Peeking into the adversary
  - Programming the random oracle

Previous Results [BDFLSZ 2011]

- **Separation**: there exist schemes secure in the classical ROM against quantum adversaries, but that are insecure in the quantum ROM
- Some classical proofs can be adapted to the quantum setting:
  - Answer RO queries randomly, same across all queries
  - Use pseudorandom function to generate randomness
  - Examples: GPV Signatures [GPV 2008]

    Full Domain Hash with specific trapdoor permutations [Coron 2000]

    Katz-Wang Signatures [KW 2003]

    Hybrid encryption scheme

Our Results

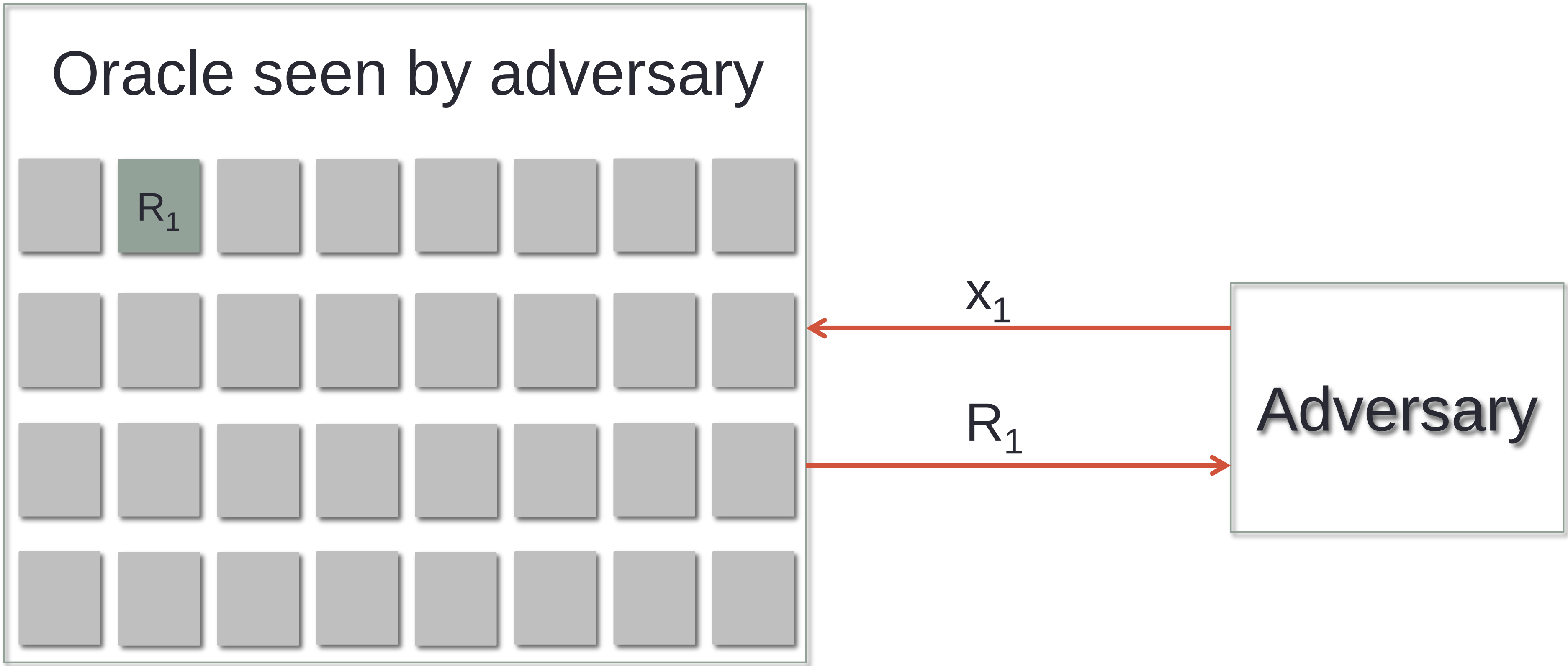
- Simulating the random oracle without additional assumptions
- New security proofs in the quantum random oracle model
  - Identity-Based Encryption
  - Hierarchical Identity-Based Encryption
  - Generic Full-Domain Hash
- New tools for arguing the indistinguishability of oracle distributions by quantum adversaries.

Common Proof Technique in Classical ROM

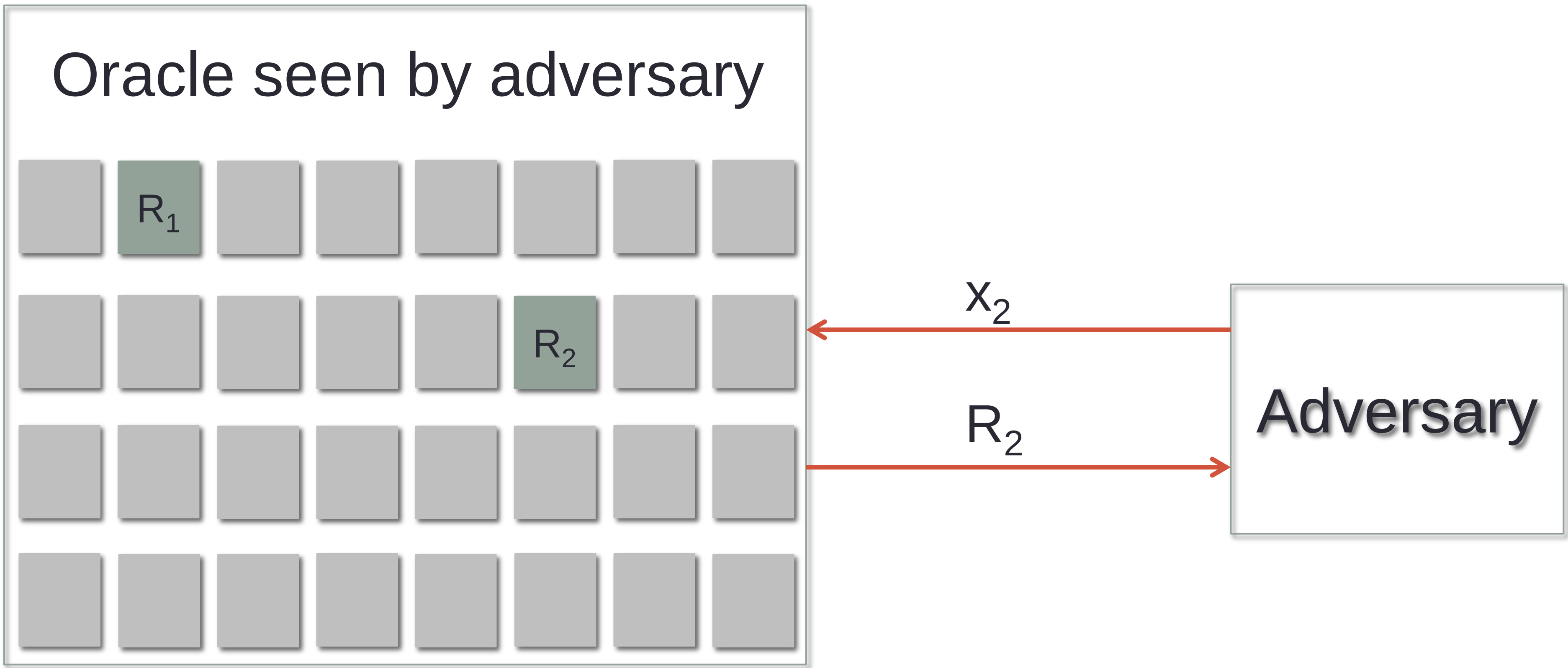
- Start with an adversary A that makes q queries to random oracle H
- Construct B that solves some problem:
  - Pick a random query i
  - For all other queries, answer in way that looks random
  - For query i, plug in some challenge c
  - If A happens to use query i, then we can solve our problem
  - A uses query i with probability 1/q, so happens with non-negligible probability

Common Proof Technique in Classical ROM

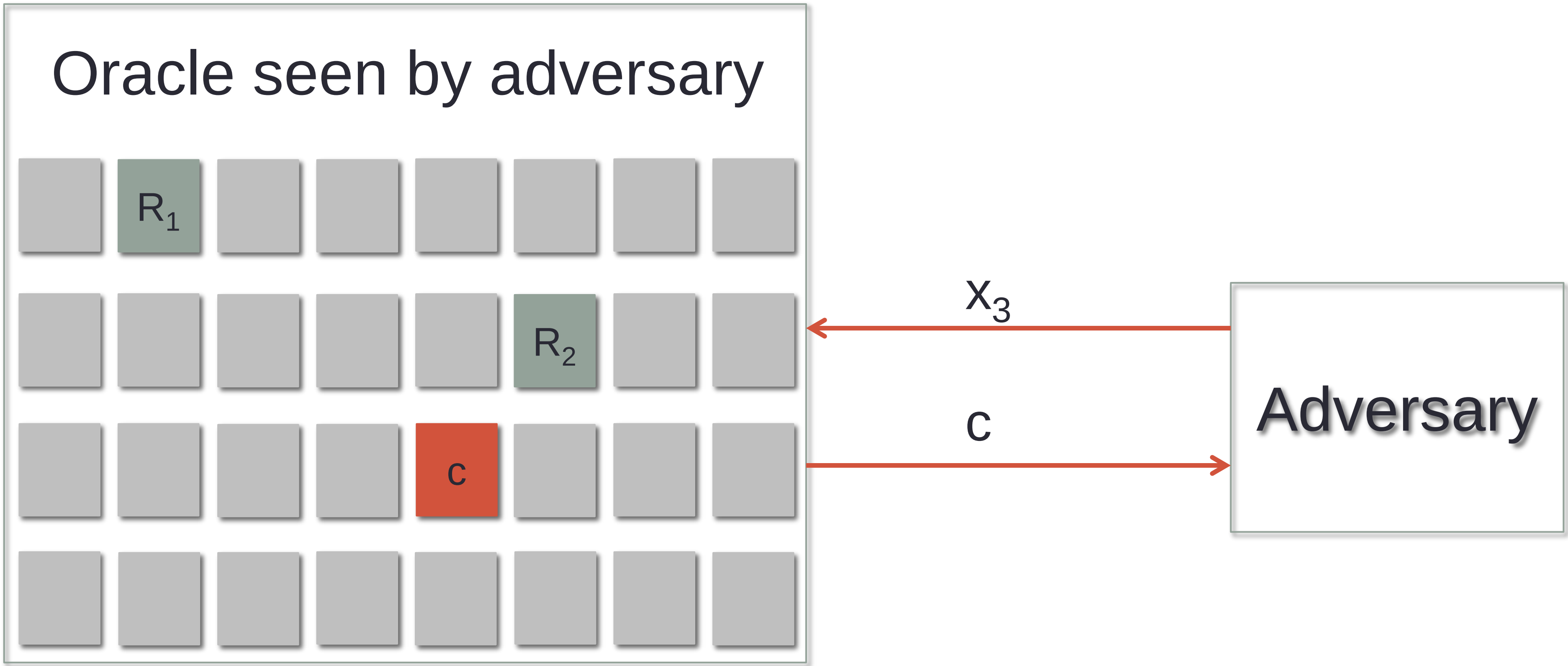
Common Proof Technique in Classical ROM



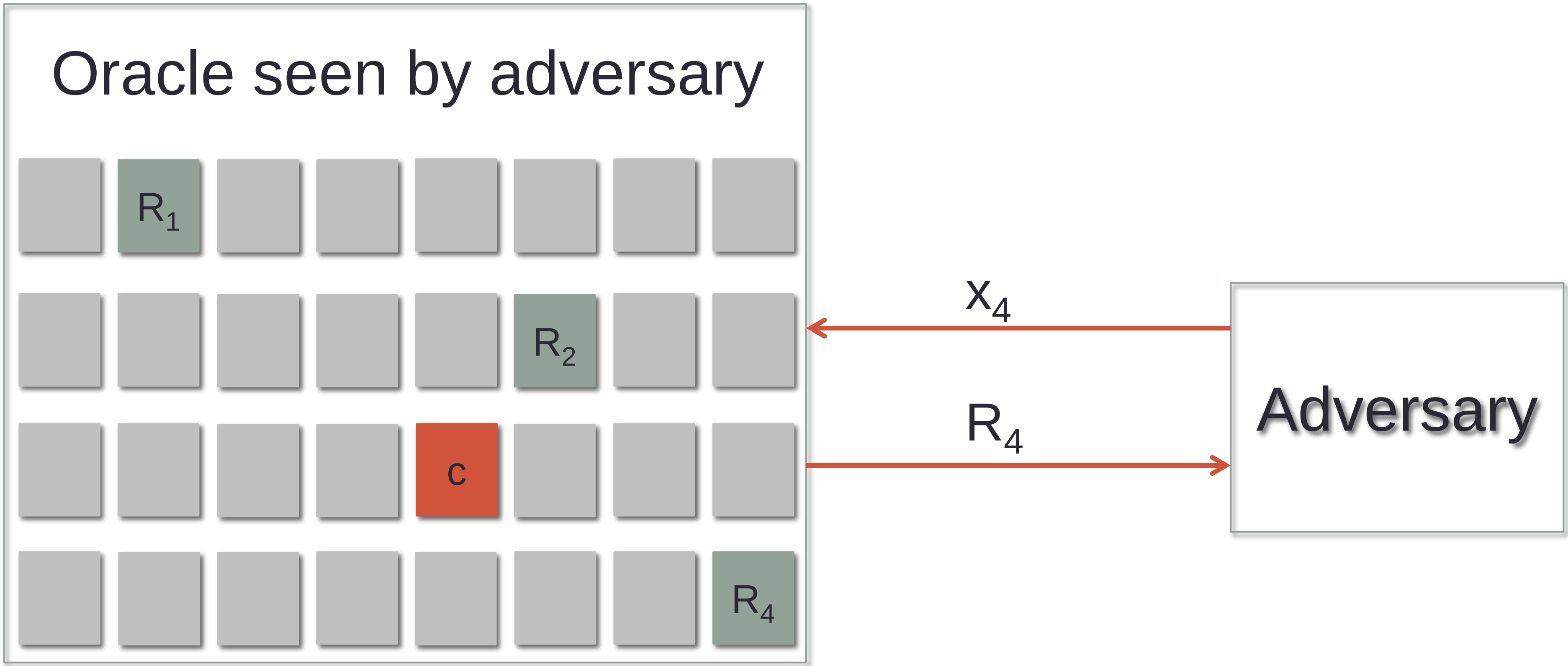
Common Proof Technique in Classical ROM



Common Proof Technique in Classical ROM

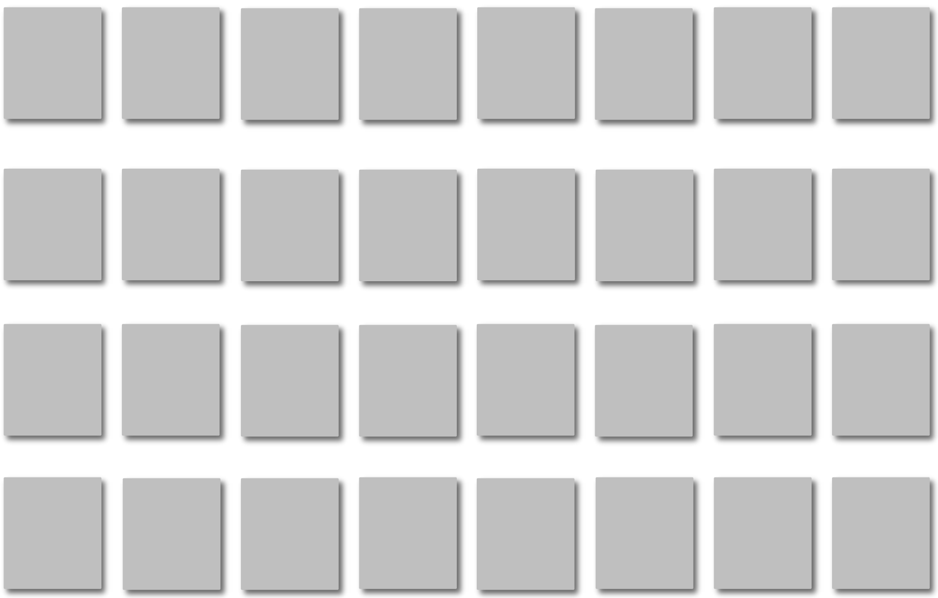


Common Proof Technique in Classical ROM

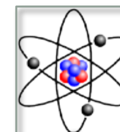


Quantum Attempt 1

Oracle seen by adversary



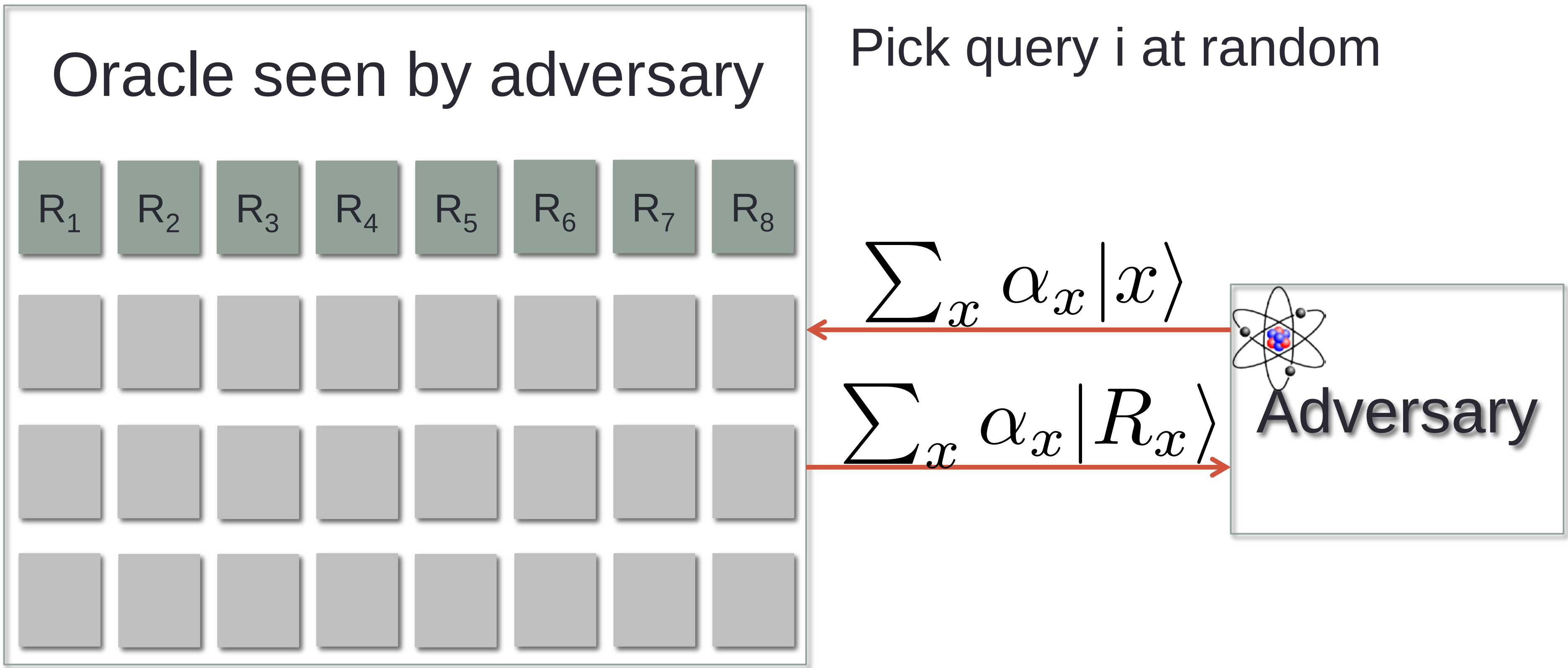
Pick query i at random



Adversary

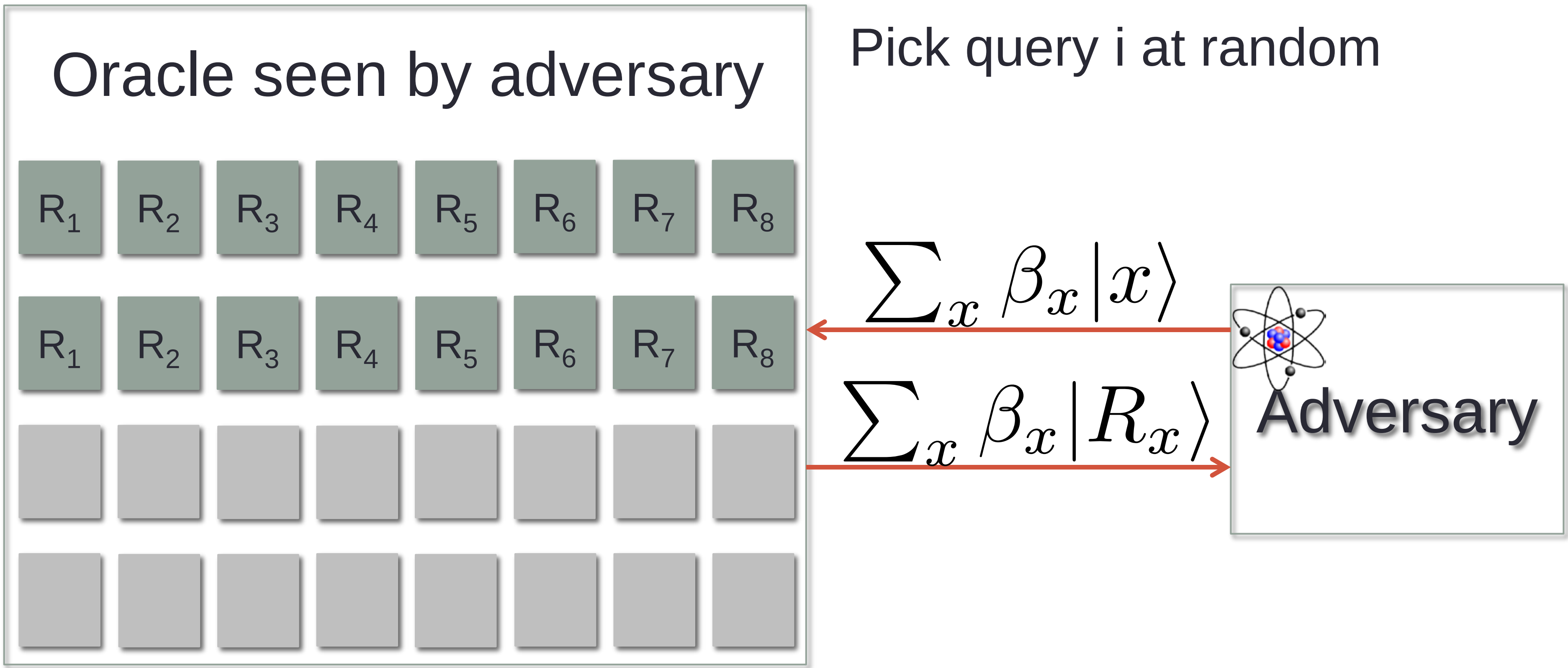
Quantum Attempt 1

Pick query i at random



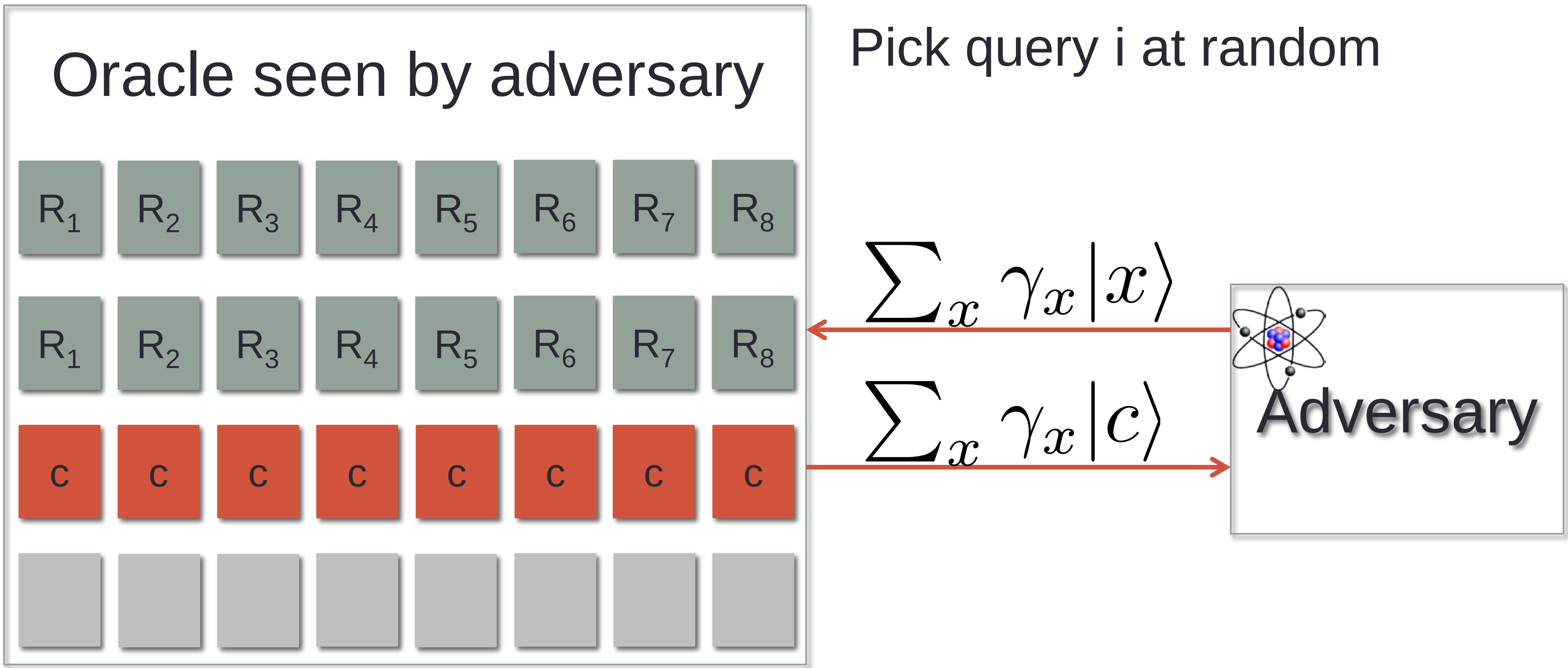
Quantum Attempt 1

Pick query i at random

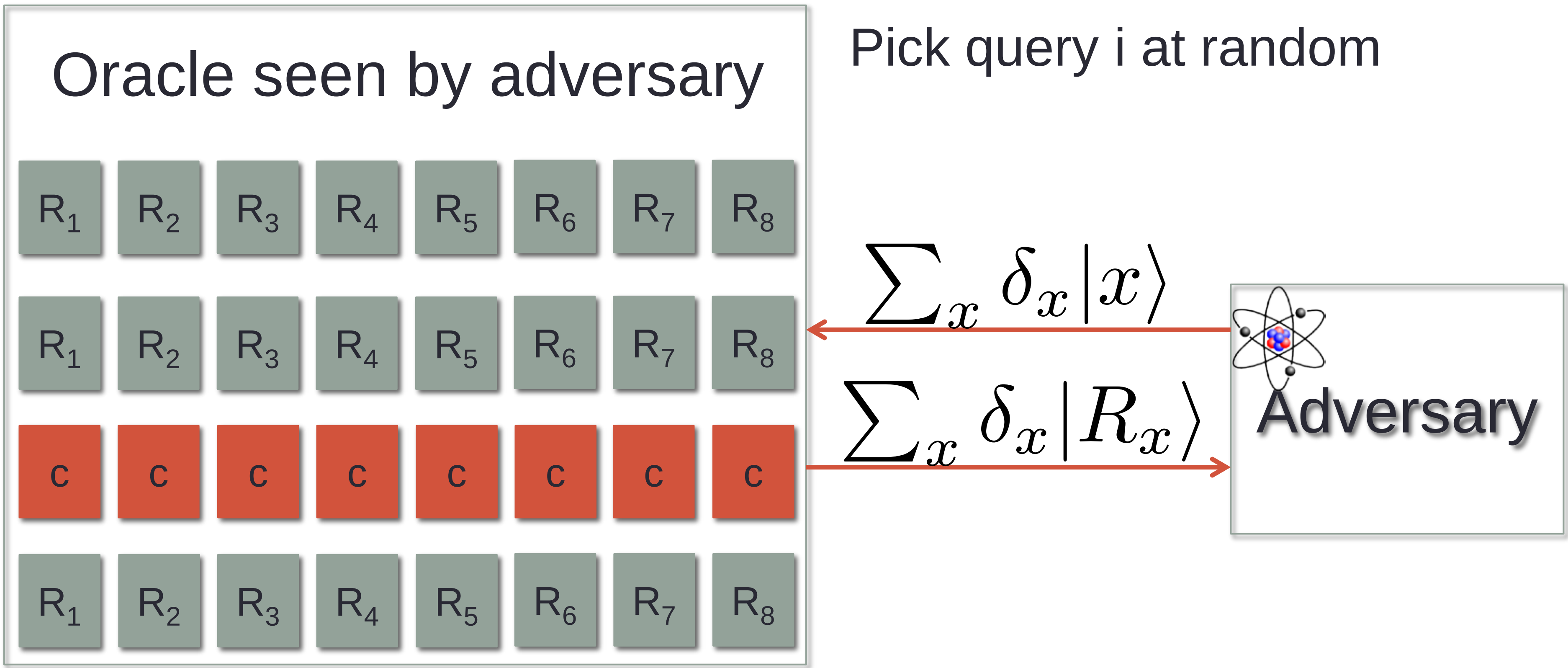


Quantum Attempt 1

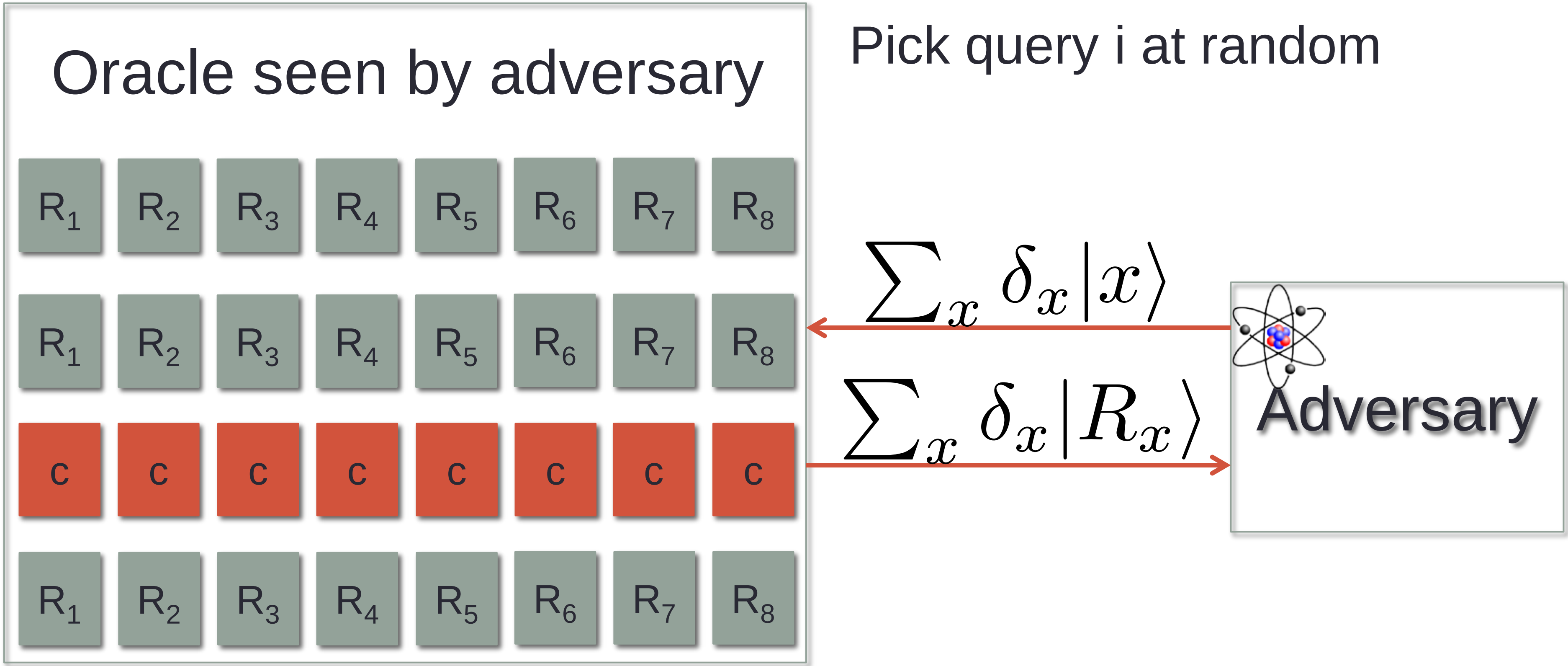
Pick query i at random



Quantum Attempt 1

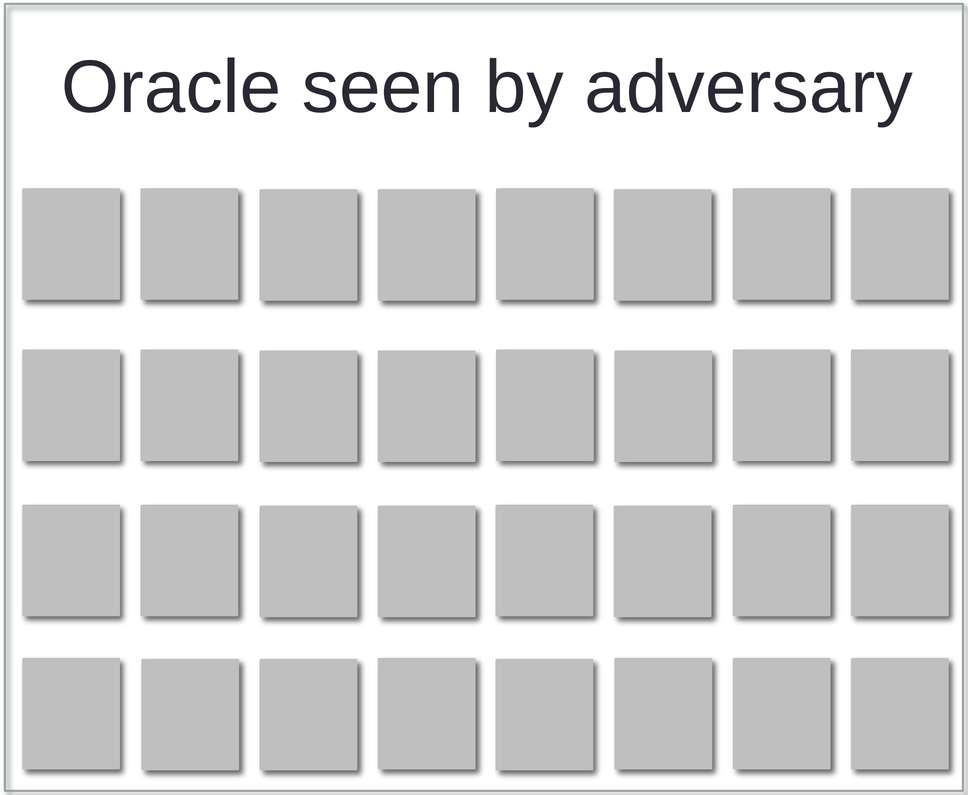


Quantum Attempt 1



Query i is inconsistent and does not look random

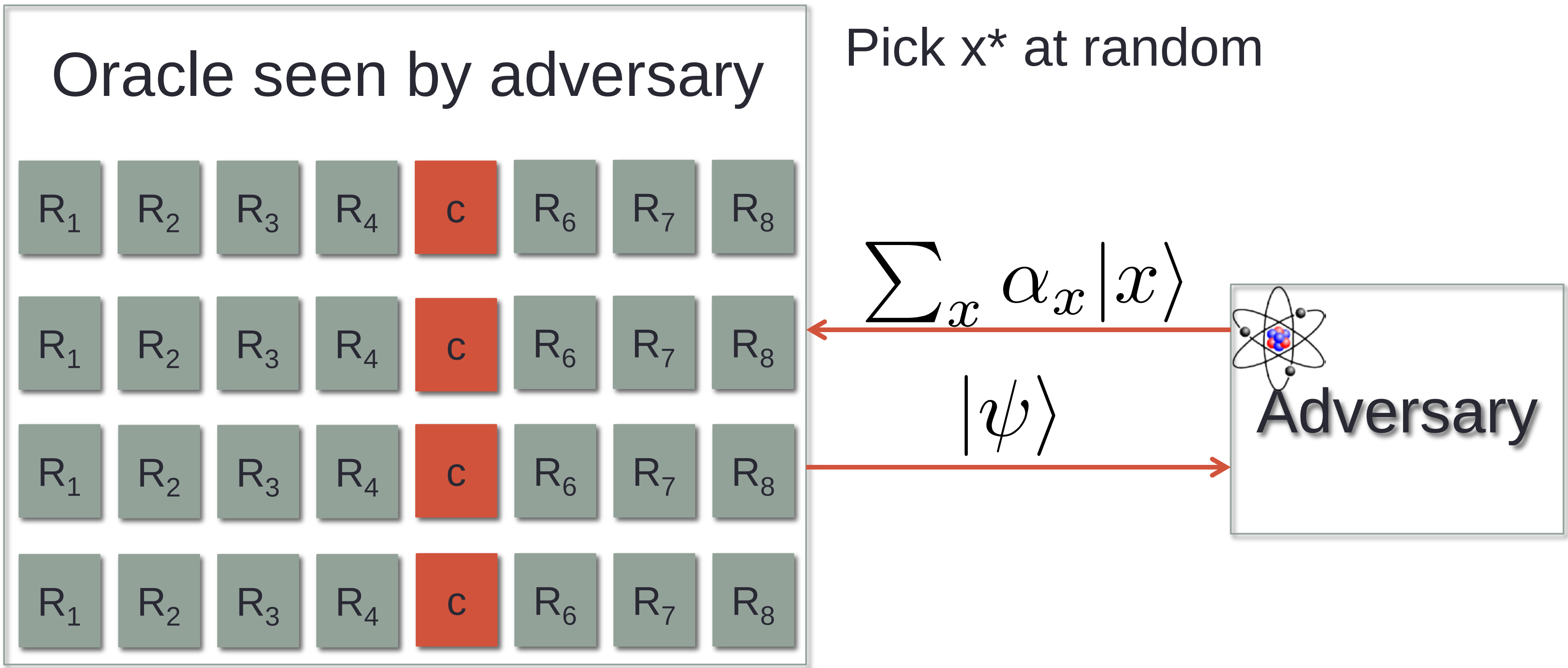
Quantum Attempt 2



Pick x* at random

Quantum Attempt 2

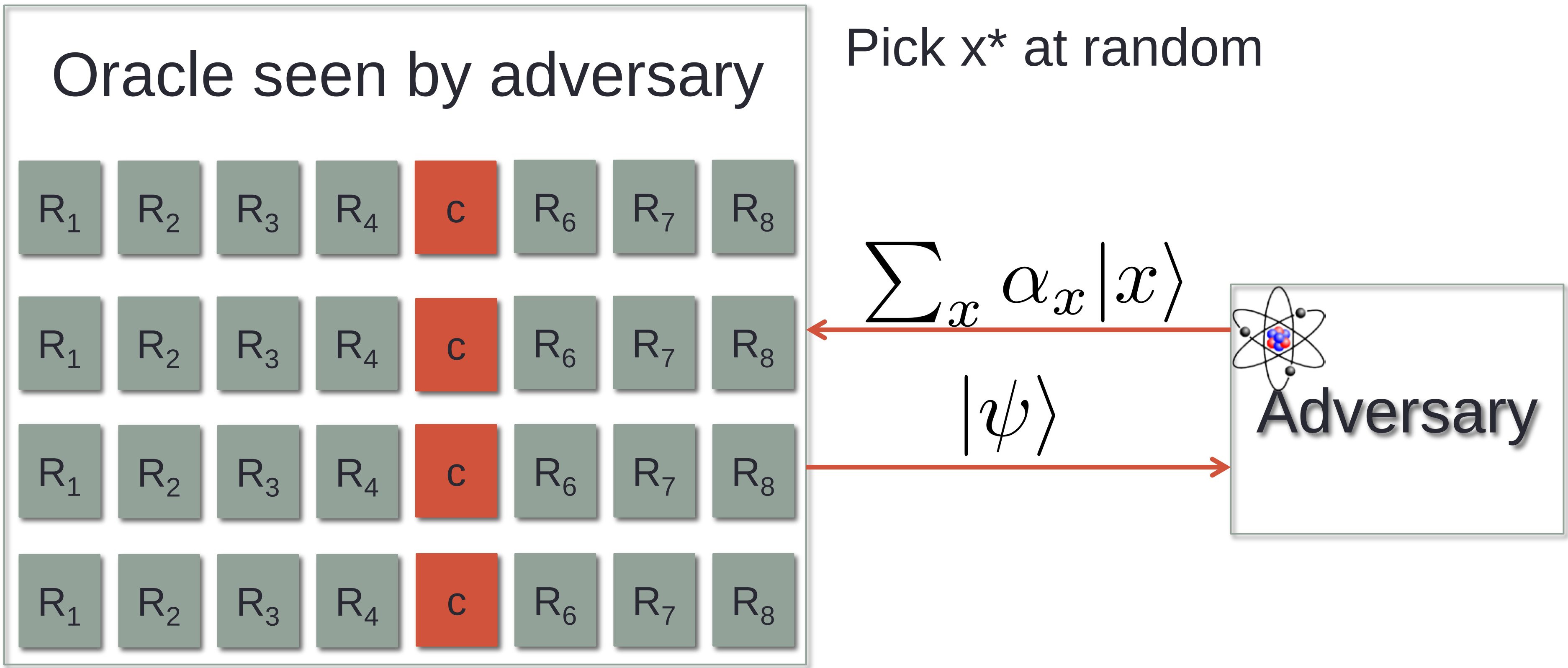
Pick x* at random



$$|\psi\rangle = \sum_{x \neq x*} \alpha_x |R_x\rangle + \alpha_{x*} |c\rangle$$

Quantum Attempt 2

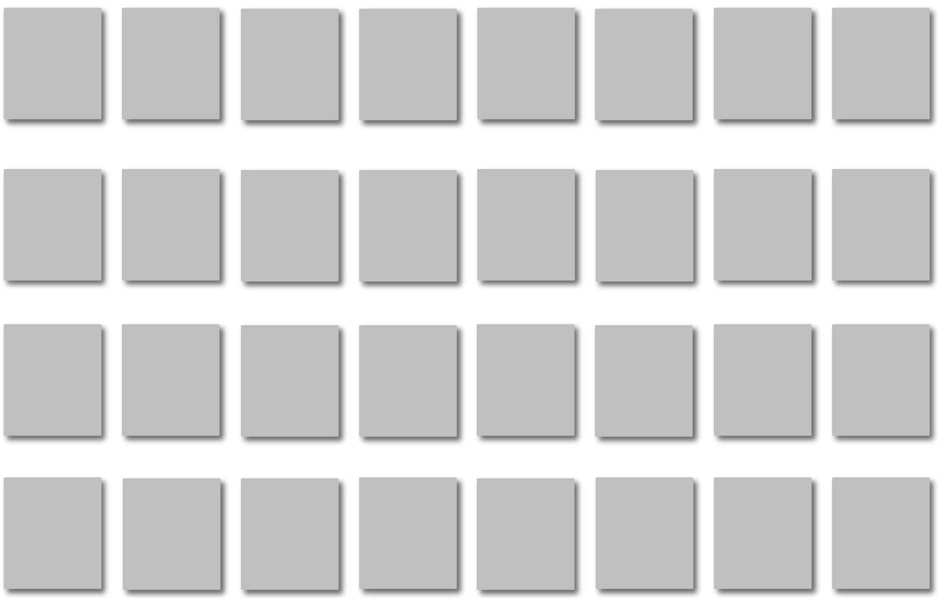
Pick x* at random



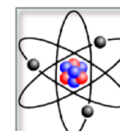
Adversary uses c with exponentially small probability

Our Solution

Oracle seen by adversary



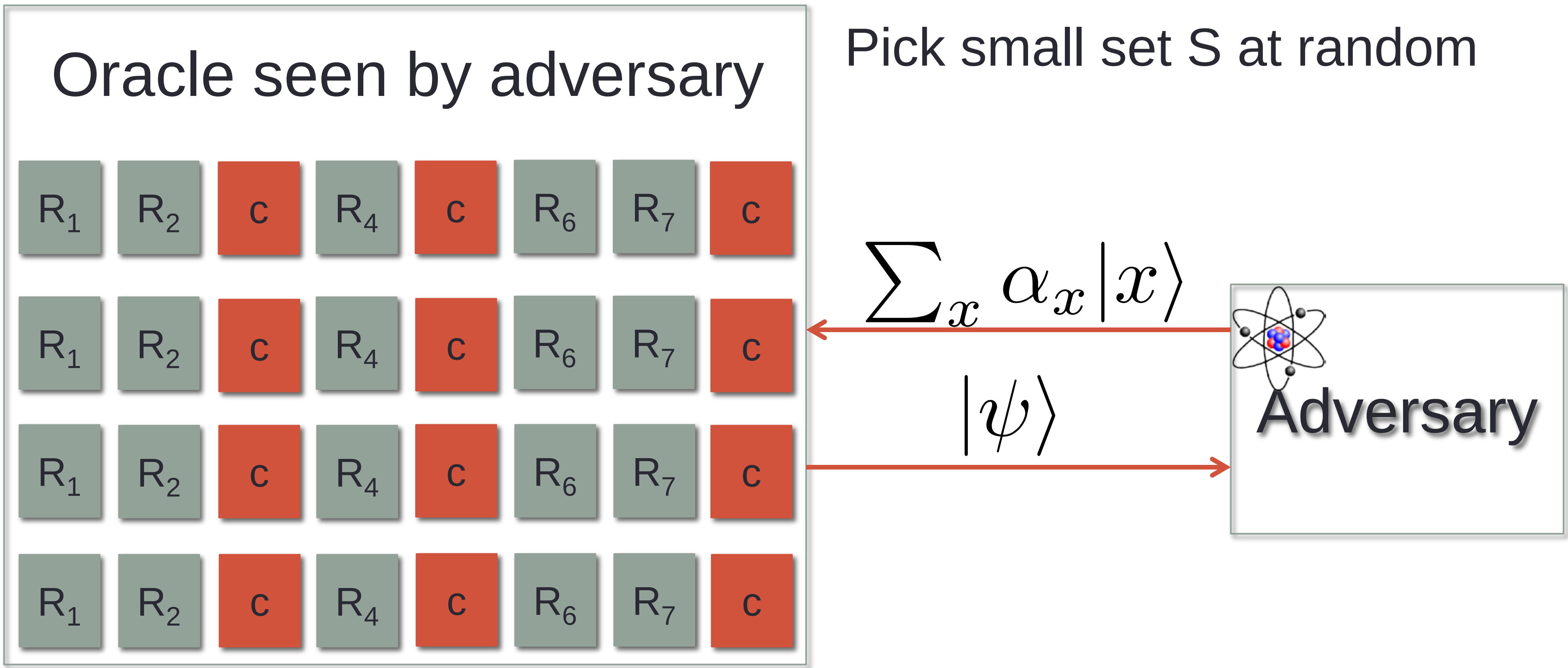
Pick small set S at random



Adversary

Our Solution

Pick small set S at random



$$|\psi\rangle = \sum_{x \notin S} \alpha_x |R_x\rangle + \sum_{x \in S} \alpha_x |c\rangle$$

Semi-Constant Distributions

- Parameterized by λ
- Pick a set S as follows: each x in the domain is in S with probability λ
- Pick a random c
- For all x in S, set $H(x) = c$
- For all other x, chose $H(x)$ randomly and independently

Semi-Constant Distributions

- Parameterized by λ
- Pick a set S as follows: each x in the domain is in S with probability λ
- Pick a random c
- For all x in S, set H(x) = c
- For all other x, chose H(x) randomly and independently

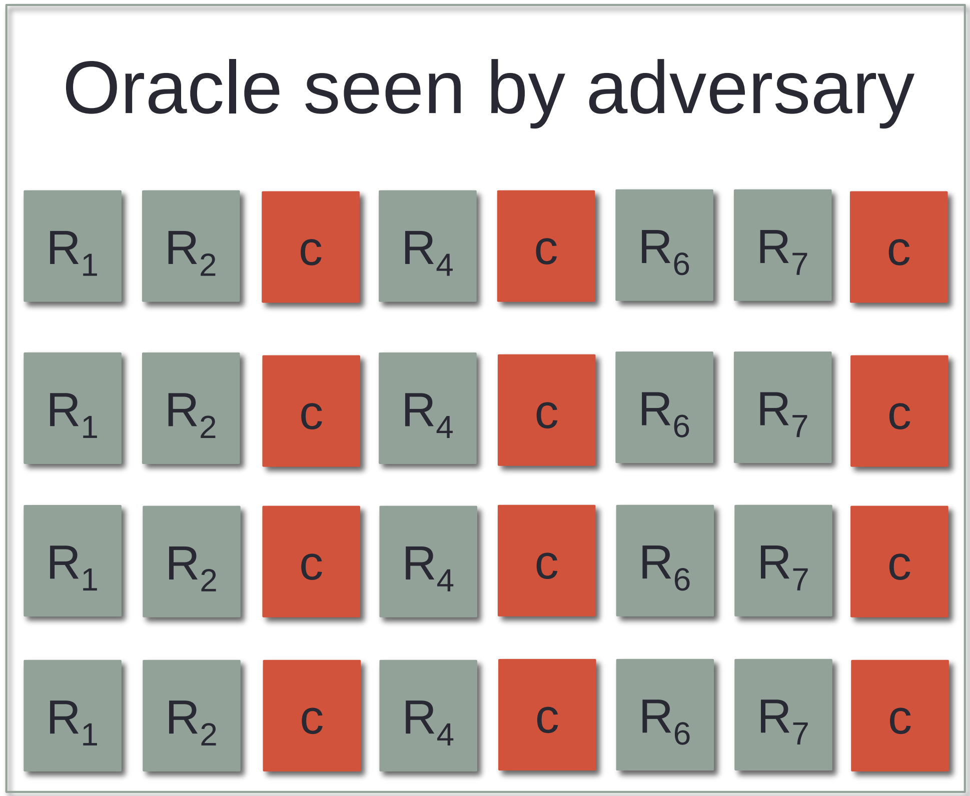
Theorem: Any quantum adversary making q queries to a semi-constant function can only tell it's not random with probability $O(q^4\lambda^2)$

Quantum Security Proof

- Suppose adversary wins with probability ε
- Pick the set S, still let oracle be random
- Probability adversary uses one of the points in S: λ
- Probability wins and uses a point in S: $\lambda\varepsilon$
- Set $H(x) = c$ for all x in S
- Probability we succeed: $\lambda\varepsilon - O(q^4\lambda^2)$
- Choose λ to maximize
- Succeed with probability $O(\varepsilon^2/q^4)$

Generating the Random Values

Need to generate random values
for exponentially many positions



Generating the Random Values

- BDFLSZ 2011:
  - Assume existence of quantum-secure PRF
  - Pick a random key k before any queries
  - Let R_x = PRF(k,x)
- Our solution:
  - Adversary makes some polynomial q of queries
  - Pick a random 2q-wise independent function f
  - Let R_x = f(x)
  - We show 2q-wise independence suffices using a standard technique called the polynomial method

Generating the Random Values

- BDFLSZ 2011:
  - Assume existence of quantum-secure PRF
  - Pick a random key k before any queries
  - Let R_x = PRF(k,x)
- Our solution:
  - Adversary makes some polynomial q of queries
  - Pick a random 2q-wise independent function f
  - Let $R_x = f(x)$
  - We show 2q-wise independence suffices using a standard technique called the polynomial method

We can remove the quantum-secure PRF assumption from prior results as well

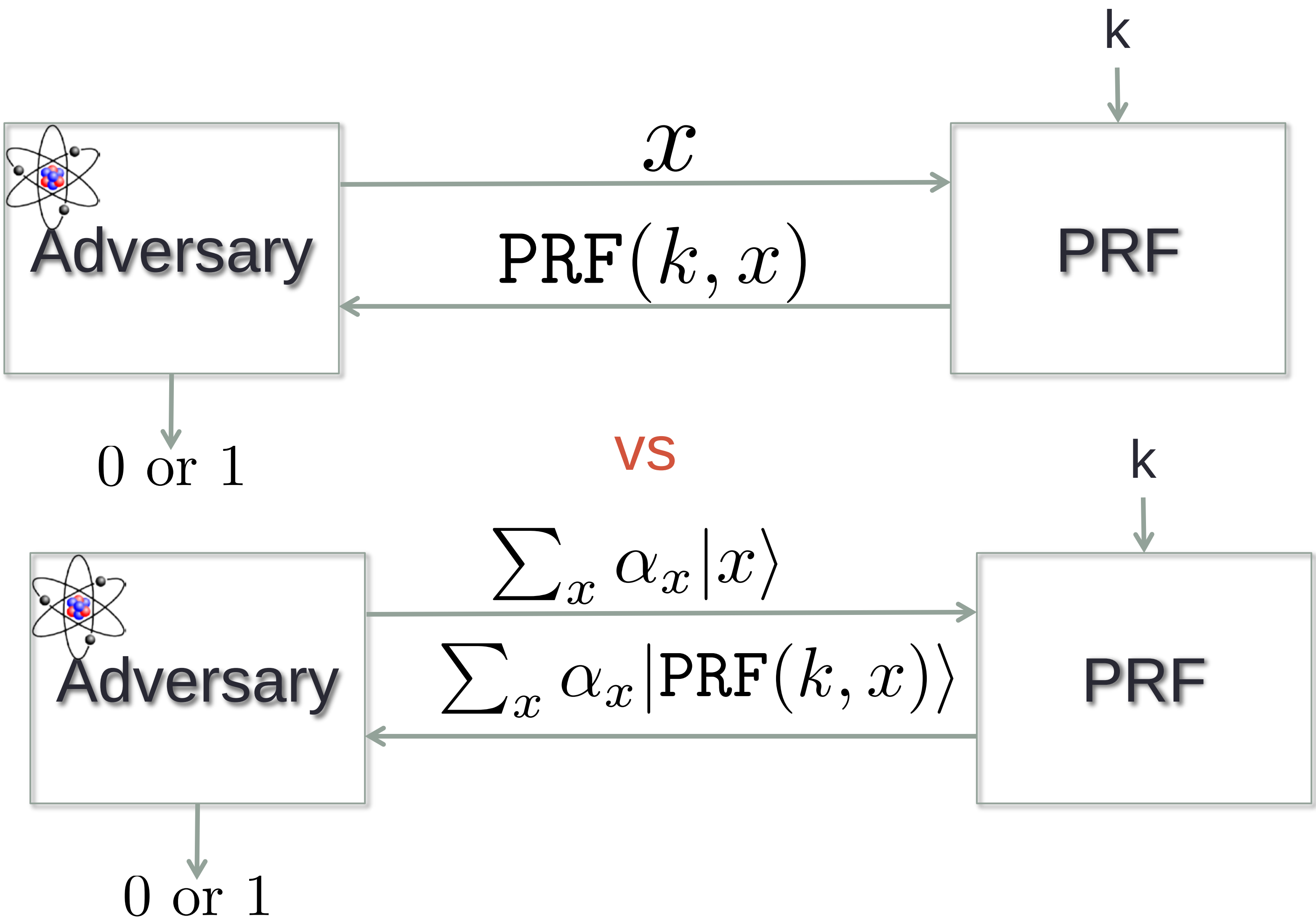
Applications of this method

- IBE scheme [GPV 2008]
- Generic Full Domain Hash
  - Previous results only showed for specific trapdoor permutations
- Apply iteratively for Hierarchical IBE [CHPK 2010, ABB 2010]
  - Security degrades doubly exponentially in depth of identity tree
  - Classically, only singly exponential

Quantum-Secure PRFs [Zhandry, FOCS 2012]

- So far, only considered case where interaction with primitive remains classical
- What if we allow quantum queries to primitive?
  - Example: pseudorandom functions

Standard Security vs Quantum Security



Quantum-Secure PRFs

- Results [Zhandry, FOCS 2012]
  - In general, PRF secure against classical queries not secure against quantum queries
  - However, several classical constructions remain secure, even against quantum queries
    - From pseudorandom generators [GGM 1984]
    - From pseudorandom synthesizers [NR 1995]
    - Direct constructions based on lattices [BPR 2011]
- Also have MACs secure when adversary can get tags on a superposition

Open Questions

- Proving the quantum security of constructions based on Fiat-Shamir [FS 1987]
  - Signatures
  - Group Signatures
  - CS Proofs
- Other constructions
  - CCA security from weaker notions [FO 1999]

Open Questions

- Proving the quantum security of constructions based on Fiat-Shamir [FS 1987]
  - Signatures
  - Group Signatures
  - CS Proofs
- Other constructions
  - CCA security from weaker notions [FO 1999]

Thank You!