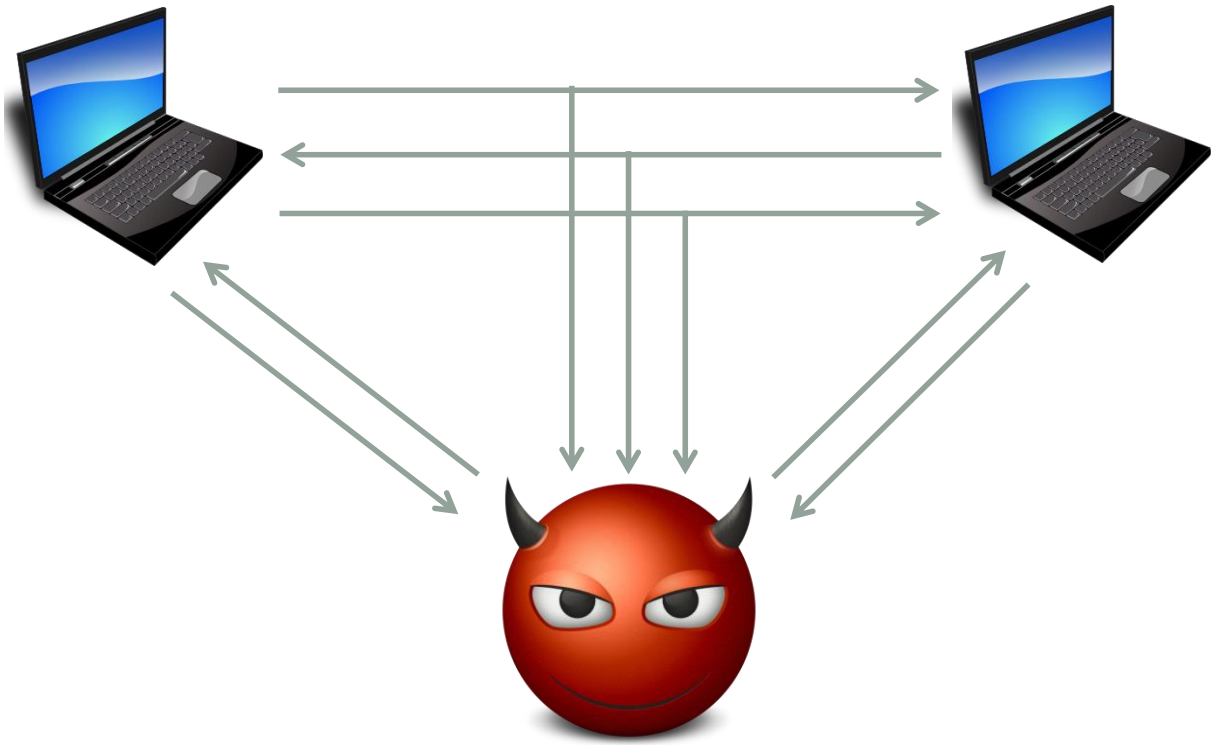
THE RANK METHOD AND APPLICATIONS TO POST-QUANTUM CRYPTOGRAPHY

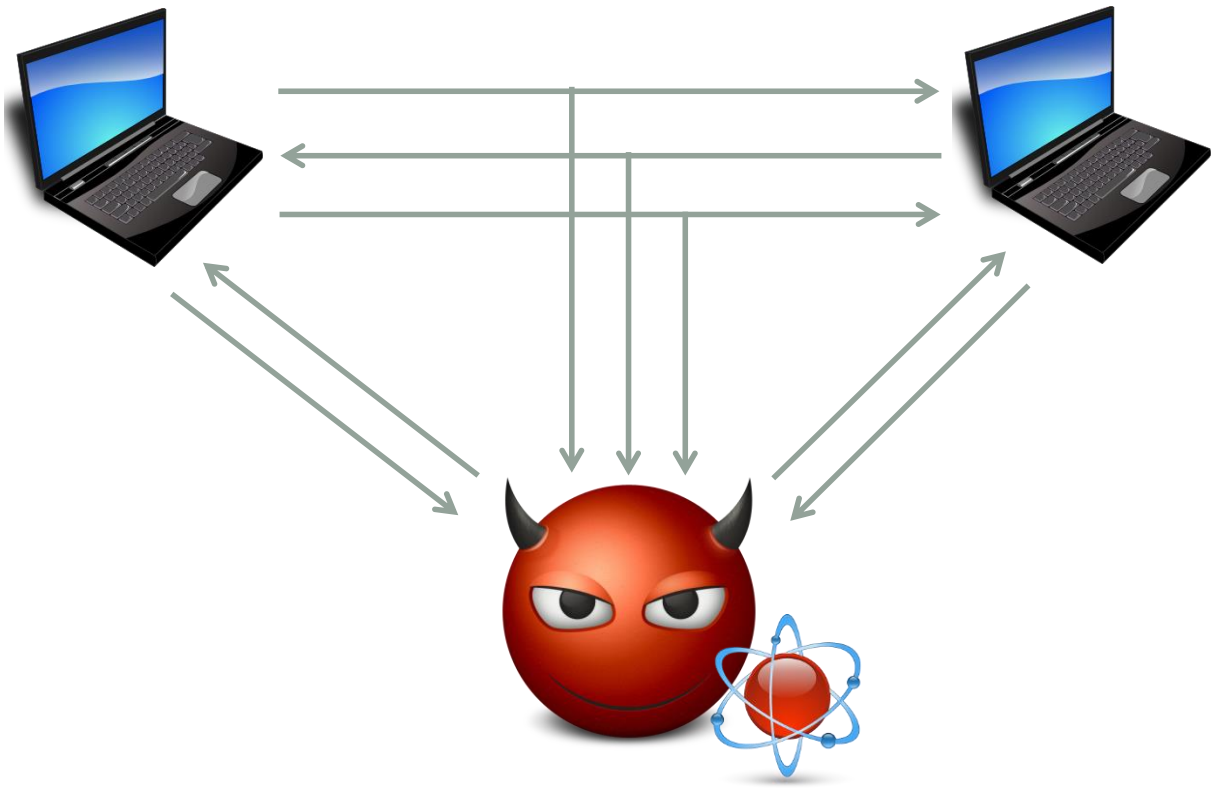
Mark Zhandry - Stanford University

Joint work with Dan Boneh

Classical Cryptography



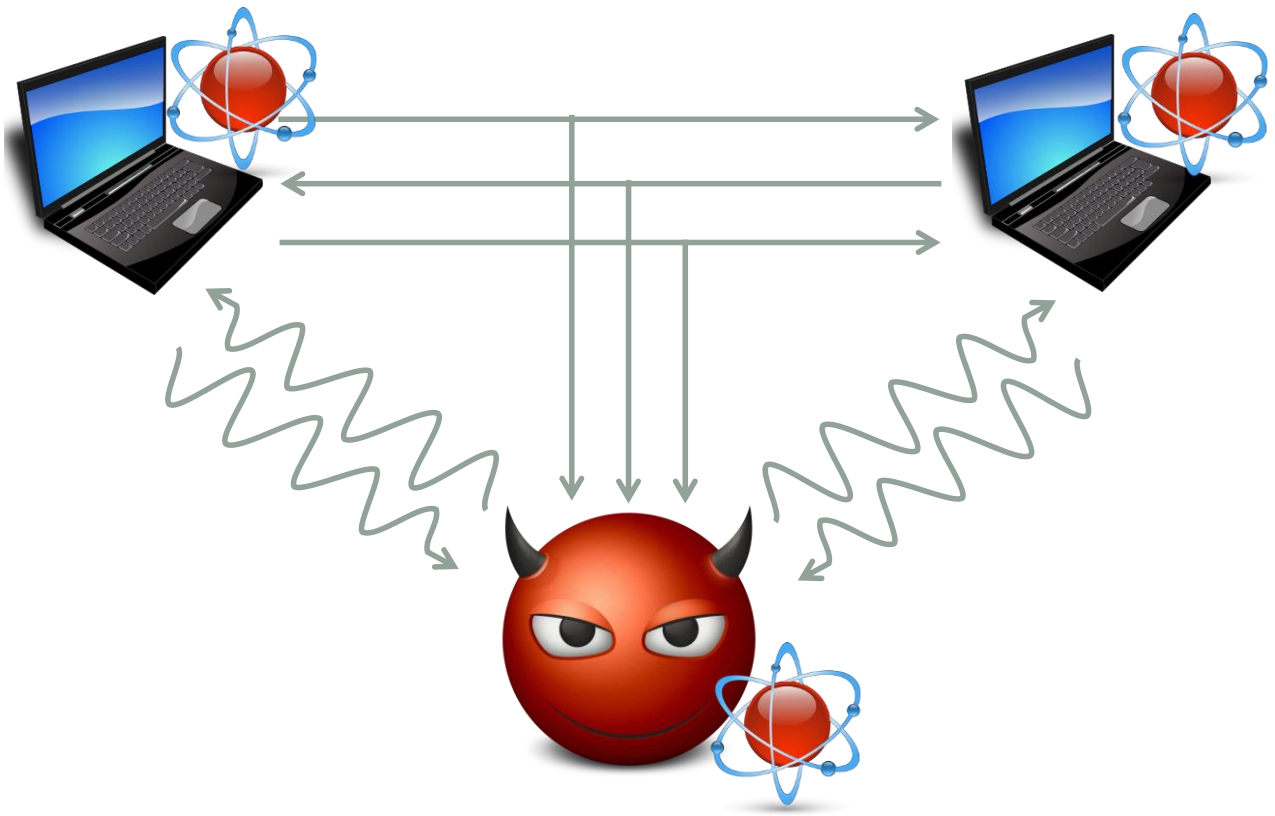
Post-Quantum Cryptography



All communication stays classical

Beyond Post-Quantum Cryptography

Eventually, all computers will be quantum



Adversary may use quantum channels

Quantum Interactions

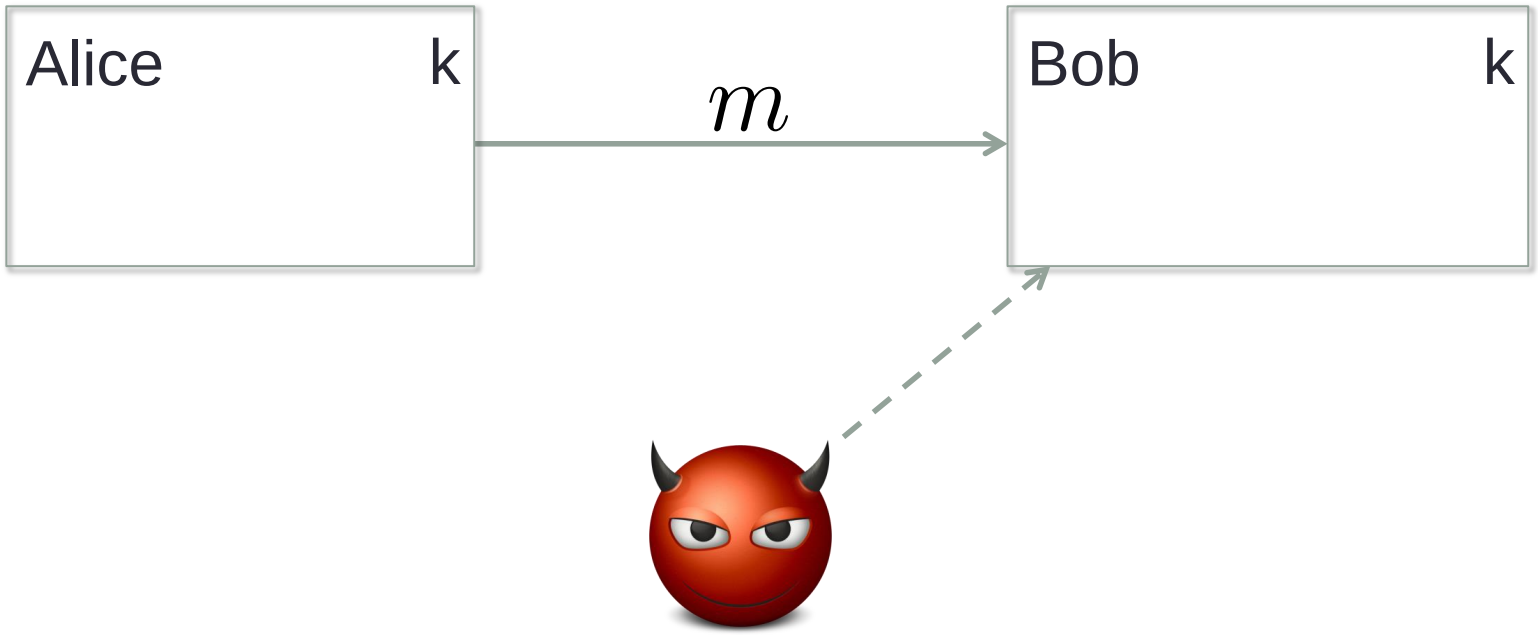
Allowing quantum interactions makes proving statements (i.e. lower bounds) much harder

Examples:

- Parity [BHCMdW'1998]
- Unstructured search [BBBV'97]
- Collision finding [Aar'01]

Same is true for cryptography

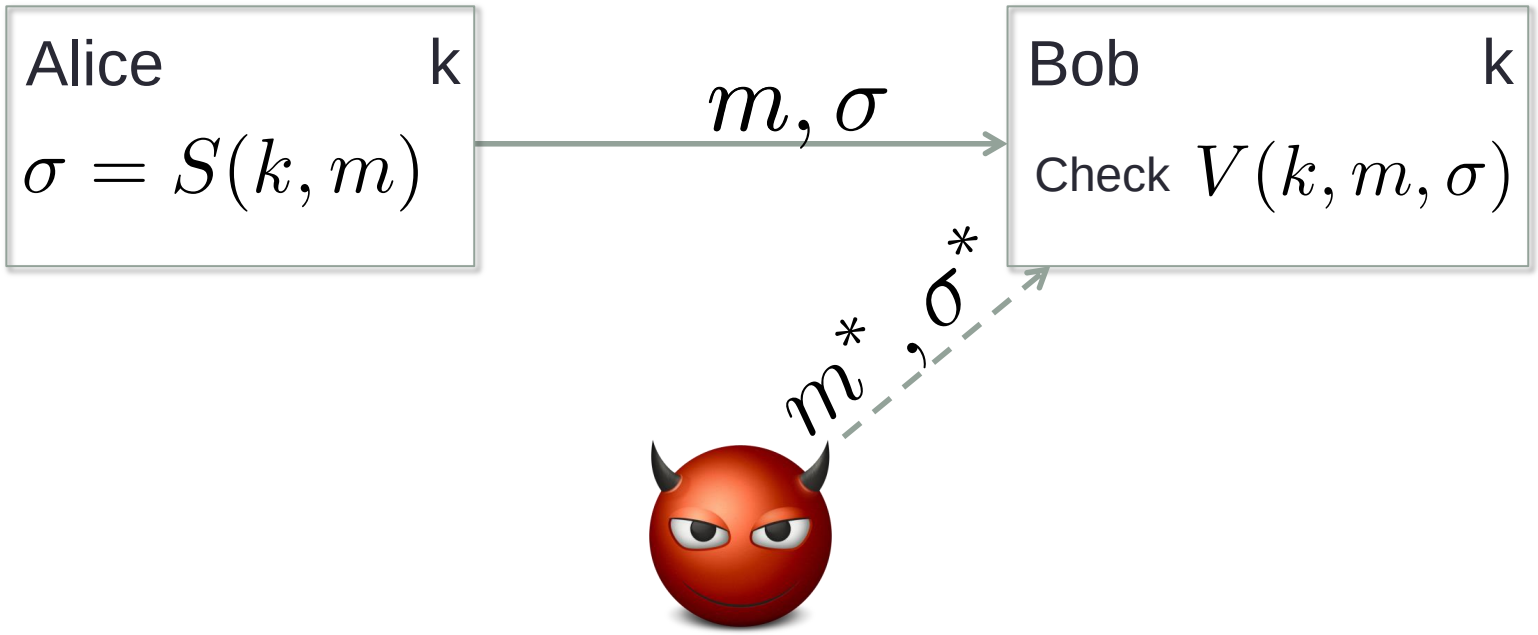
Example: Message Authentication Codes



How does Bob verify that m was sent by Alice and wasn't modified in transit?

- Classical solution: MAC

Example: Message Authentication Codes



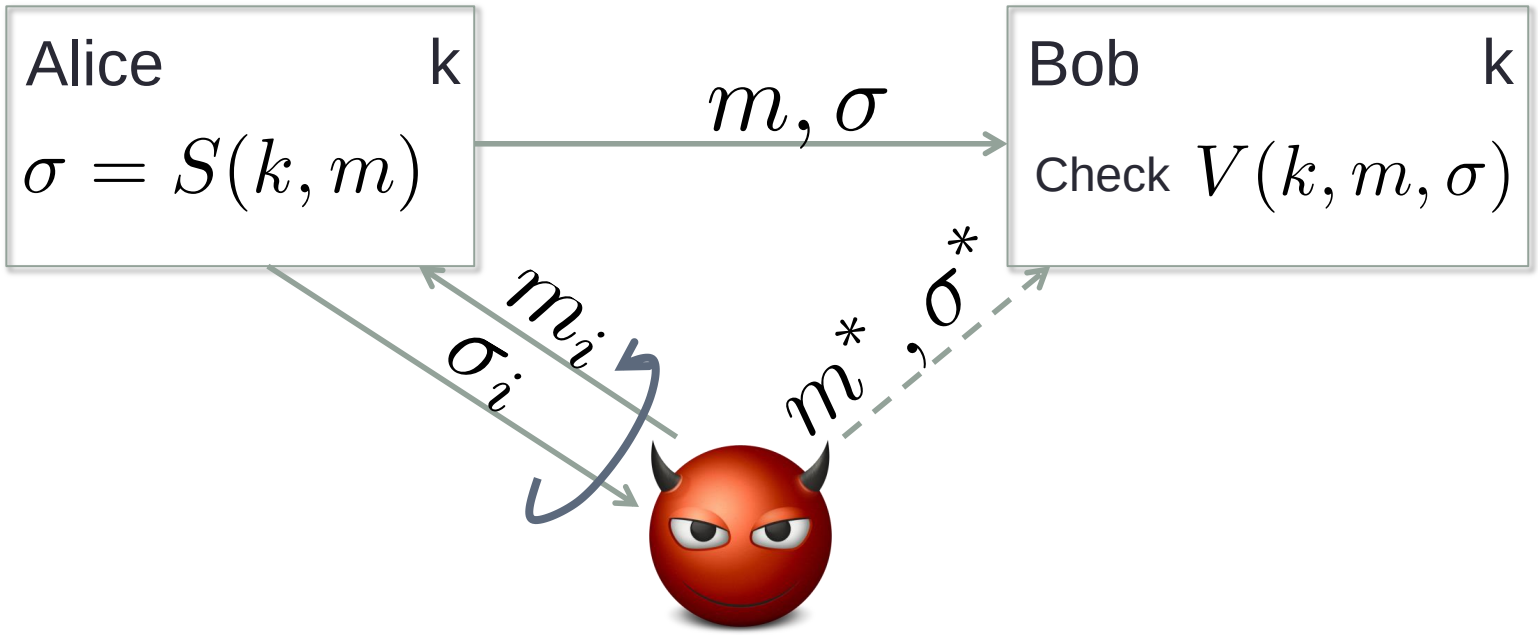
Correctness: All message/tags sent by Alice verify:

$$V(k, m, S(k, m)) = \texttt{True}$$

Security: Only messages sent by Alice verify:

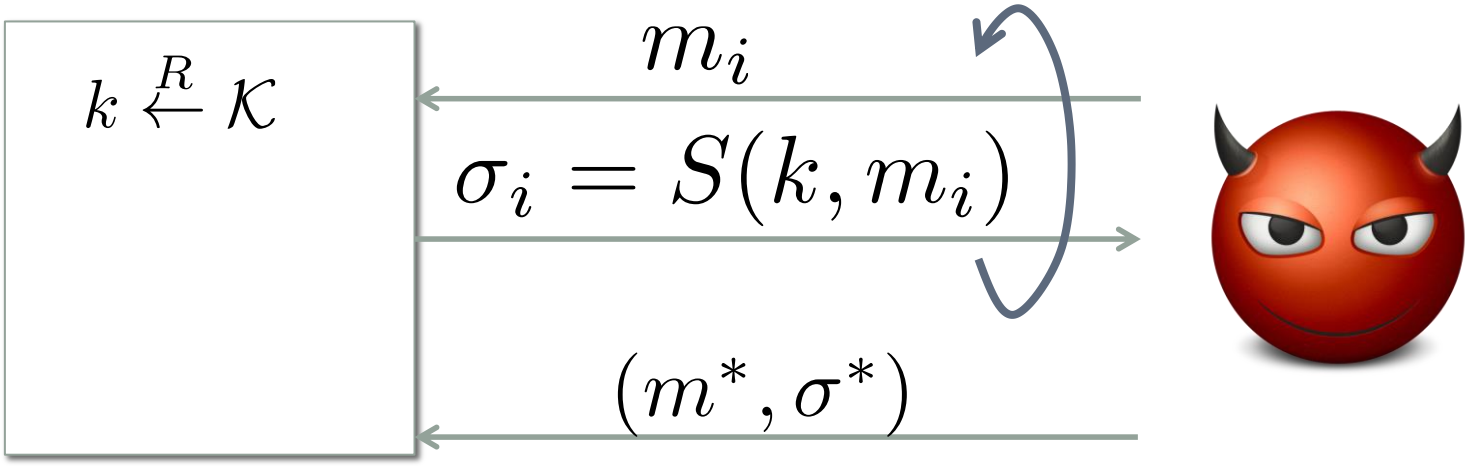
$$V(k, m^*, \sigma^*) = \texttt{False}$$

Example: Message Authentication Codes



Before forgery attempt, adversary sees valid tags on many messages, possibly of his choice

(Classical) Security For MACs

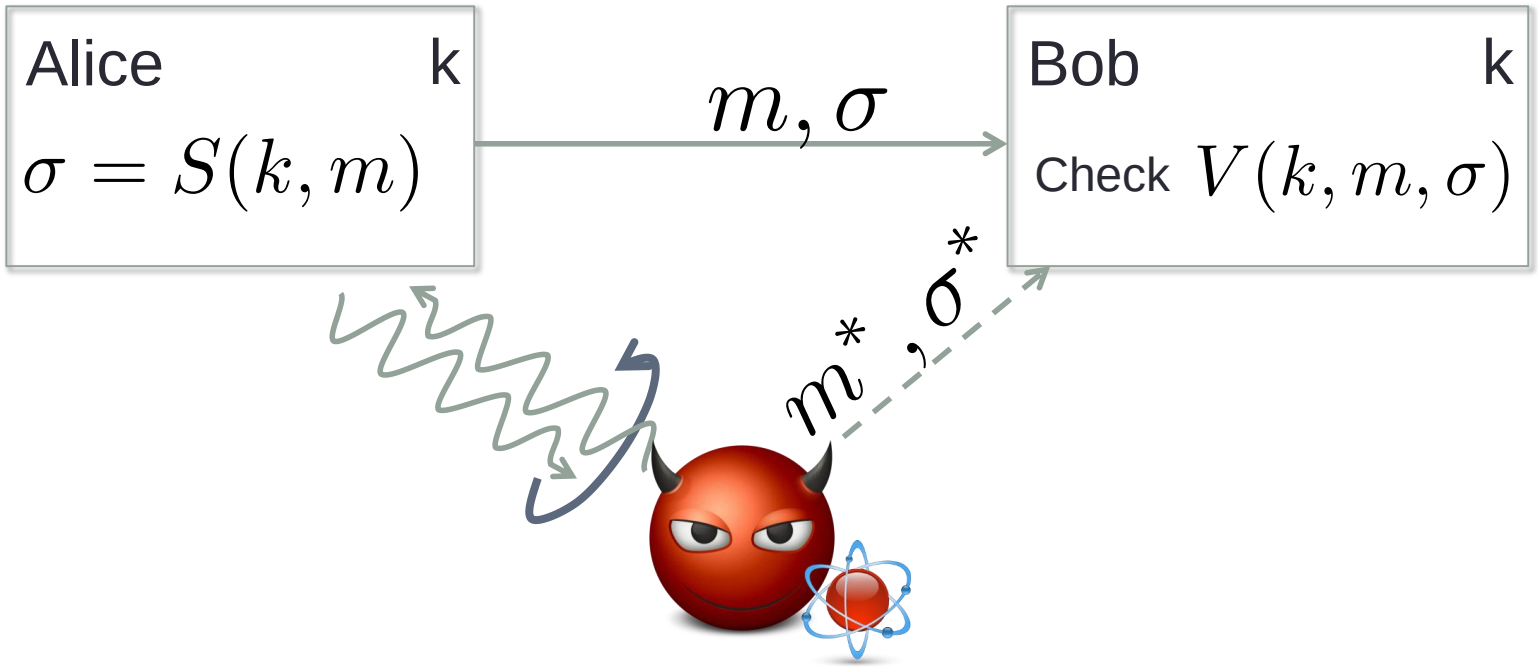


Adversary wins if, after many MAC queries, it can produce a valid message/tag pair that was not seen before:

$$(m^*, \sigma^*) \neq (m_i, \sigma_i) \forall i$$
$$V(k, m^*, \sigma^*) = \texttt{True}$$

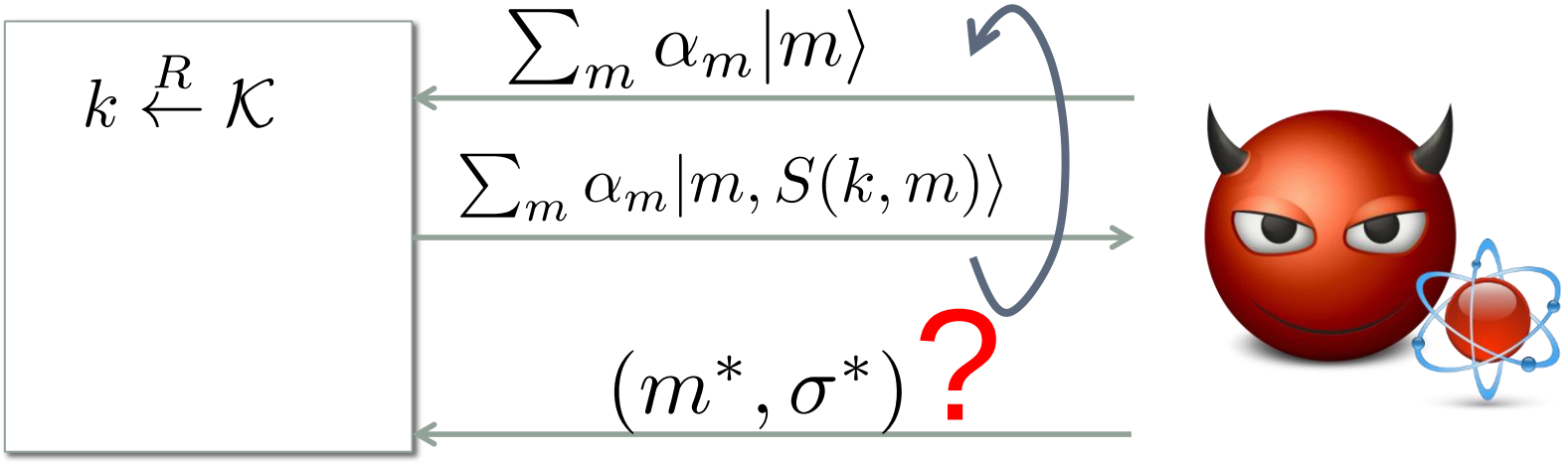
Security: No efficient adversary can win, except with negligible probability

Full Quantum Security for MACs



Now adversary has a quantum channel

Full Quantum Security for MACs?



What does a forgery mean here?

- Cannot record query to check against forgery
- Adversary can “see” tags for all messages

Ideas

1. Wins only if weight of forgery in queries is 0
- Impossible to check
- Uniform superposition → all weights non-zero

X

2. Wins only if weight is low
- Still impossible to check
- Uniform superposition → all weights low

X

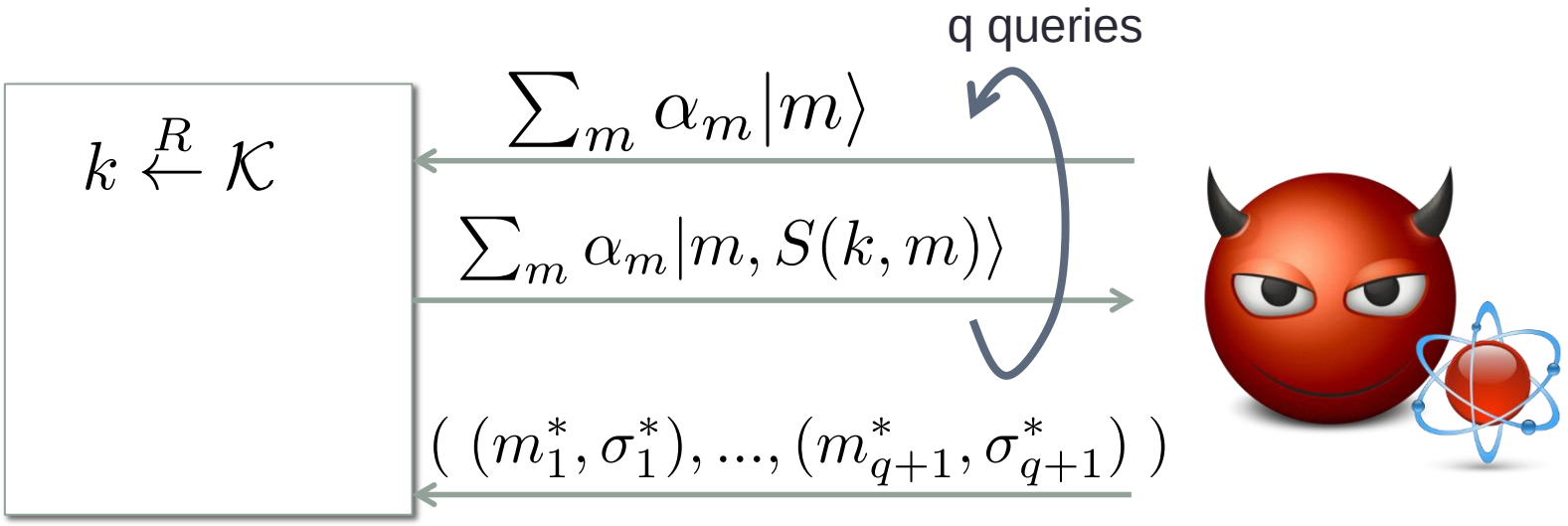
3. Ask adversary to commit to message ahead of time
- Called selective security → too weak

X

4. After q queries, adversary can easily produce q tags. Instead ask him to produce q+1.

✓

Full Quantum Security for MACs



Adversary wins if, after making q quantum MAC queries, it produces q+1 distinct valid message/tag pairs:

$$(m_i^*, \sigma_i^*) \neq (m_j^*, \sigma_j^*) \forall i \neq j$$
$$V(k, m_i^*, \sigma_i^*) = \texttt{True} \forall i$$

Security: No efficient adversary can win, except with negligible probability

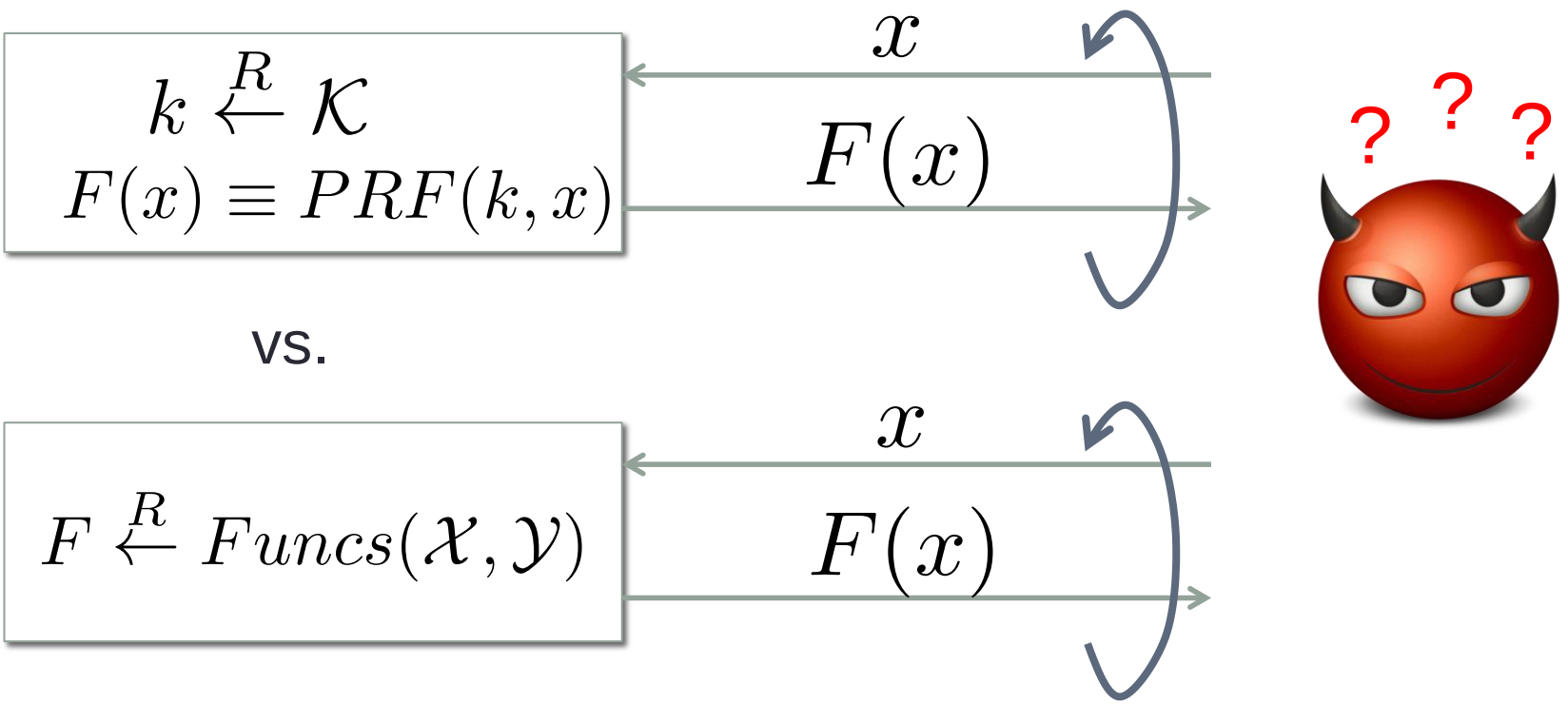
How to Build Quantum-Secure MACs?

To build quantum-secure MACs, look to classical constructions.

Example: PRFs

Pseudorandom Functions (PRFs)

Efficient keyed functions that look like random functions to efficient adversaries:



Quantum Pseudorandom Functions (QPRFs)

Efficient keyed functions that look like random functions to efficient quantum adversaries:

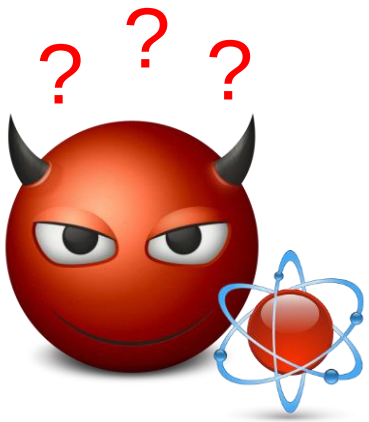
$$k \xleftarrow{R} \mathcal{K}$$
$$F(x) \equiv PRF(k, x)$$

$$\sum_x \alpha_x |x\rangle$$
$$\sum_x \alpha_x |x, F(x)\rangle$$

vs.

$$F \xleftarrow{R} Funcs(\mathcal{X}, \mathcal{Y})$$

$$\sum_x \alpha_x |x\rangle$$
$$\sum_x \alpha_x |x, F(x)\rangle$$



PRFs, QPRFs, and MACs

PRFs are fundamental building block in classical crypto

→ QPRFs are fundamental to post-quantum crypto

We know how to build PRFs [GGM'84] and QPRFs [Zha'12b]

Given PRF, can define a new MAC:

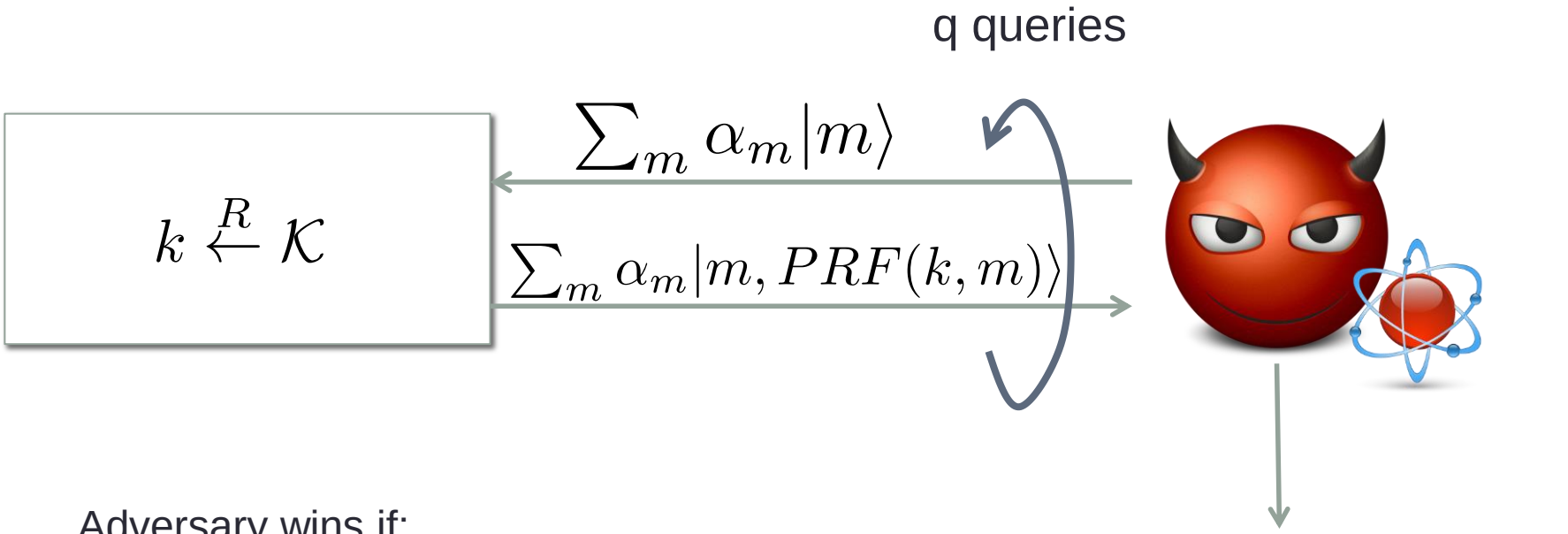
S(k,m) = PRF(k,m)

V(k,m,σ) = (PRF(k,m) == σ)

Classically, this gives a secure MAC

→ Same true in quantum world?

Security of PRF as a MAC



Adversary wins if:

$$(m_i^*, \sigma_i^*) \neq (m_j^*, \sigma_j^*) \forall i \neq j$$

$$\sigma_i^* = PRF(k, m_i^*) \forall i$$

Security of PRF as a MAC

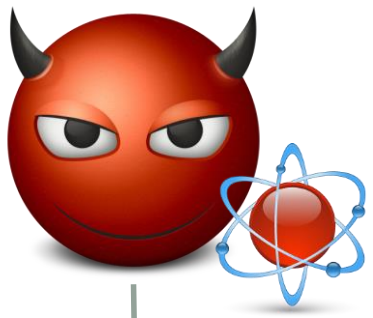
PRF looks random:

q queries

$F \xleftarrow{R} Funcs(\mathcal{X}, \mathcal{Y})$

$\sum_m \alpha_m |m\rangle$

$\sum_m \alpha_m |m, F(m)\rangle$



$\{(m_1^*, \sigma_1^*), ..., (m_{q+1}^*, \sigma_{q+1}^*)\}$

Adversary wins if:

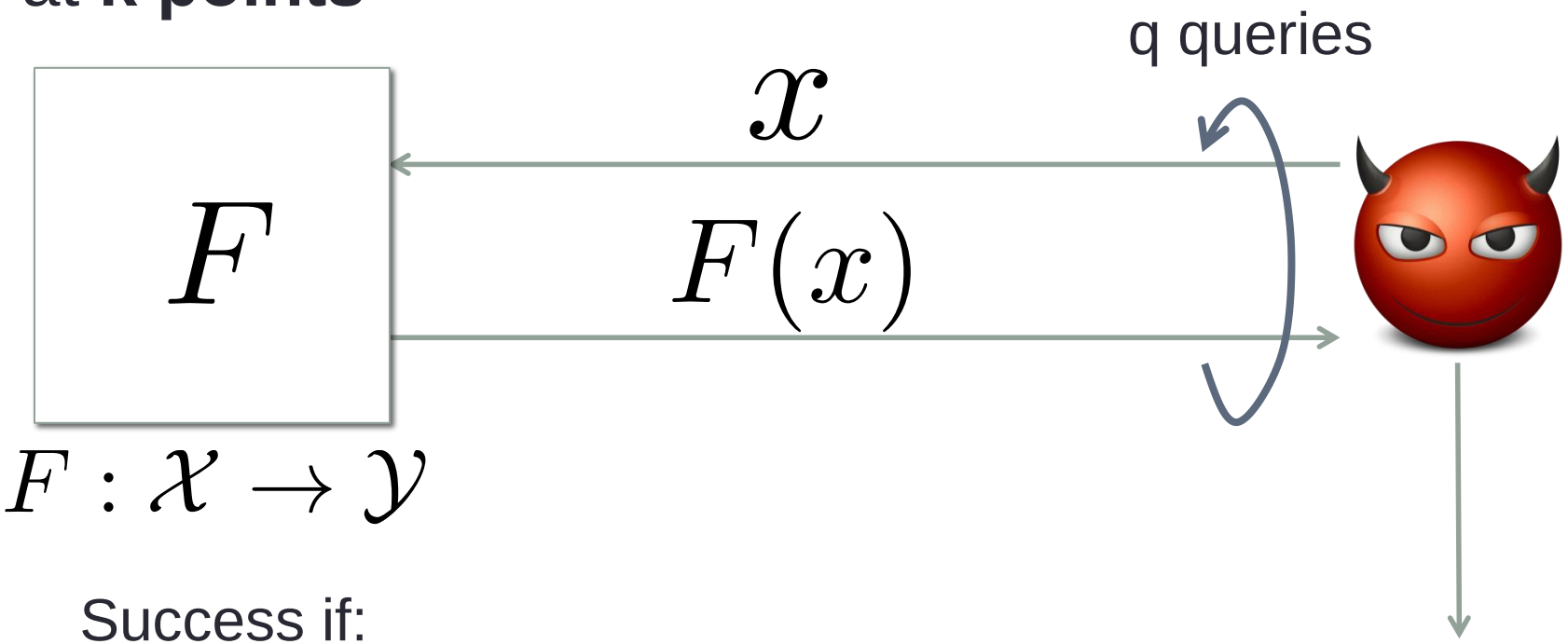
$$(m_i^*, \sigma_i^*) \neq (m_j^*, \sigma_j^*) \forall i \neq j$$

$$\sigma_i^* = F(m_i^*) \forall i$$

After q quantum queries to F, adversary gives q+1 input/output pairs of F

(Classical) Oracle Interrogation

After **q queries** to a random function F, predict F at **k points**



Success if:

- $z_i = F(x_i) \forall i$
- $x_i \neq x_j \forall i \neq j$

(Classical) Oracle Interrogation

Choose a random function F from set X to set Y

Allow algorithm A to make q oracle queries to F

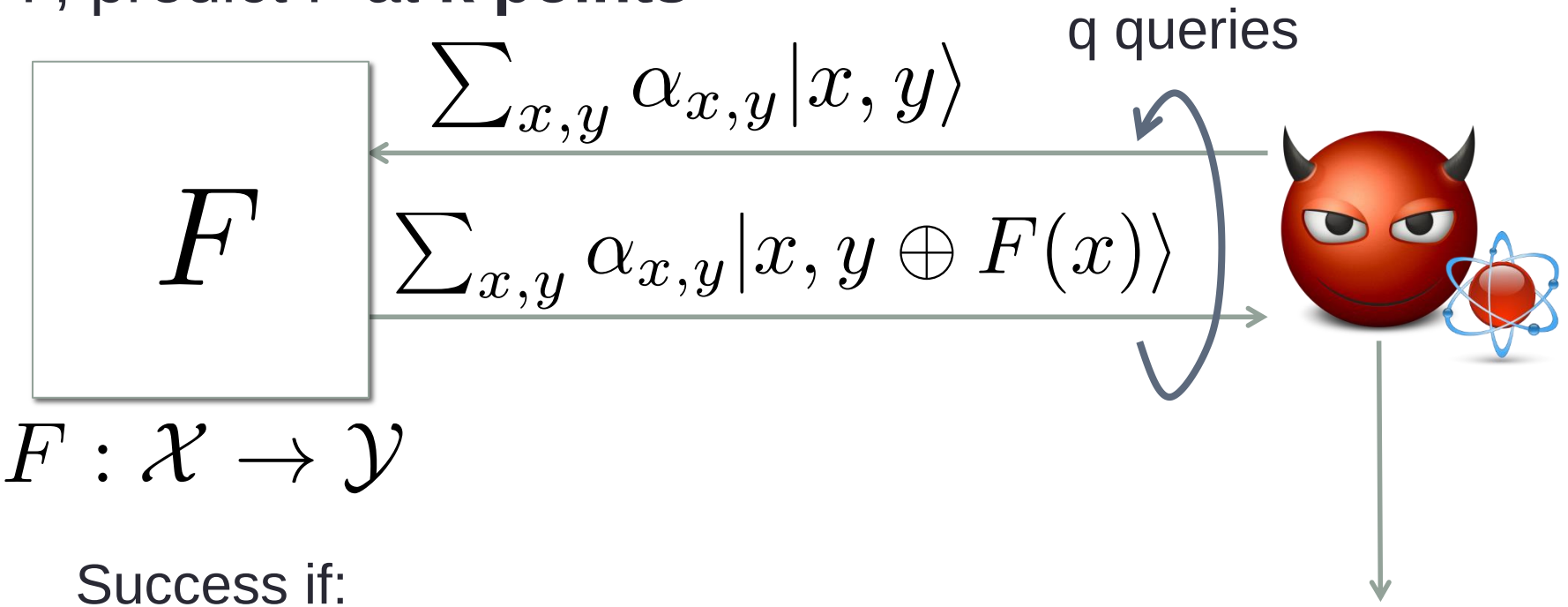
A's goal: evaluate F on any k distinct points

Let $P_C(\mathcal{X}, \mathcal{Y}, q, k)$ denote the optimal success probability

$$P_C(\mathcal{X}, \mathcal{Y}, q, k) = \begin{cases} 1 & \text{if } q \geq k \\ \left(\frac{1}{|\mathcal{Y}|}\right)^{(k-q)} & \text{if } q < k \end{cases}$$

Quantum Oracle Interrogation

After **q quantum queries** to a random function F, predict F at **k points**



Success if:

- $z_i = F(x_i) \forall i$
- $x_i \neq x_j \forall i \neq j$

Quantum Oracle Interrogation

Choose a random function F from set X to set Y

Allow algorithm A to make q **quantum** oracle queries to F

A's goal: evaluate F (classically) on any k distinct points

Let $P_Q(\mathcal{X}, \mathcal{Y}, q, k)$ denote the optimal success probability

$$P_Q(\mathcal{X}, \mathcal{Y}, q, k) = \begin{cases} 1 & \text{if } q \geq k \\ ??? & \text{if } q < k \end{cases}$$

Problem: A gets to "see" F on all inputs!

Prior Work

[van Dam'98]: Quantum algorithm for binary outputs

Theorem([vD'98]): For any constant $\epsilon > 0$, there exists a quantum algorithm that makes q quantum queries to an oracle F:X→{0,1} and evaluates F at $k = (2 - \epsilon)q$ points with overwhelming probability.

In other words, for binary outputs, oracle interrogation is easy

Larger outputs?

- Might expect that we can apply [vD'98] in parallel on each bit of output → probability extends to arbitrary output size?
- Short answer: no

Extending to Arbitrary Range Sizes

[vD'98]: Given q queries to a binary oracle F, can evaluate F at $k = (2 - \epsilon)q$ points.

Our result: Let F be an oracle with range Y of size N. There exists a constant c_N such that we can evaluate F at $k = (c_N - \epsilon)q$ points.

- Example: N = 4 → $c_4 = 4/3$

→ Can make 1000 queries to 2-bit oracle, evaluate F at 1300 points with probability ~95%

(one bit oracle: 1930 points with prob ~95%)

Can We Do Better?

Not much known

- If $k>2q$, success probability in oracle interrogation problem must be $\leq \frac{1}{2}$, else we can solve parity

Many questions remain:

- What about $k<2q$?
- Tighter bounds on success probability?

Can we use existing lower bound techniques to show optimality?

Existing Quantum Impossibility Techniques

Need: upper bound **average case** success probability when given **exact** number of queries

Existing techniques: asymptotic lower bound on query number needed for high success probability in **worst case**

Can fix some of these issues, but problems remain:

Polynomial Method:
- Lose factor of 2 in query number
  → only useful when $q \leq k/2$

Adversary Method:
- Weight of each input can be high
  → only useful when $q << k$

Need new lower bound technique

The Rank Method

A new way to prove quantum impossibility results.

Fix a quantum oracle algorithm A that makes q quantum queries to an oracle F, and tries to learn a function F at k points.

Let $|\psi_F\rangle$ be the final state of A after q queries to F:

$$|\psi_F\rangle = \mathbf{U}_q\mathbf{F}\mathbf{U}_{q-1}\mathbf{F}\cdots\mathbf{F}\mathbf{U}_0|0\rangle$$

Define $\mathrm{Rank}(A) = \mathrm{Dim\ Span}\{|\psi_F\rangle\}$

Suppose we have a bound on Rank(A). What does this buy us?

The Rank Method

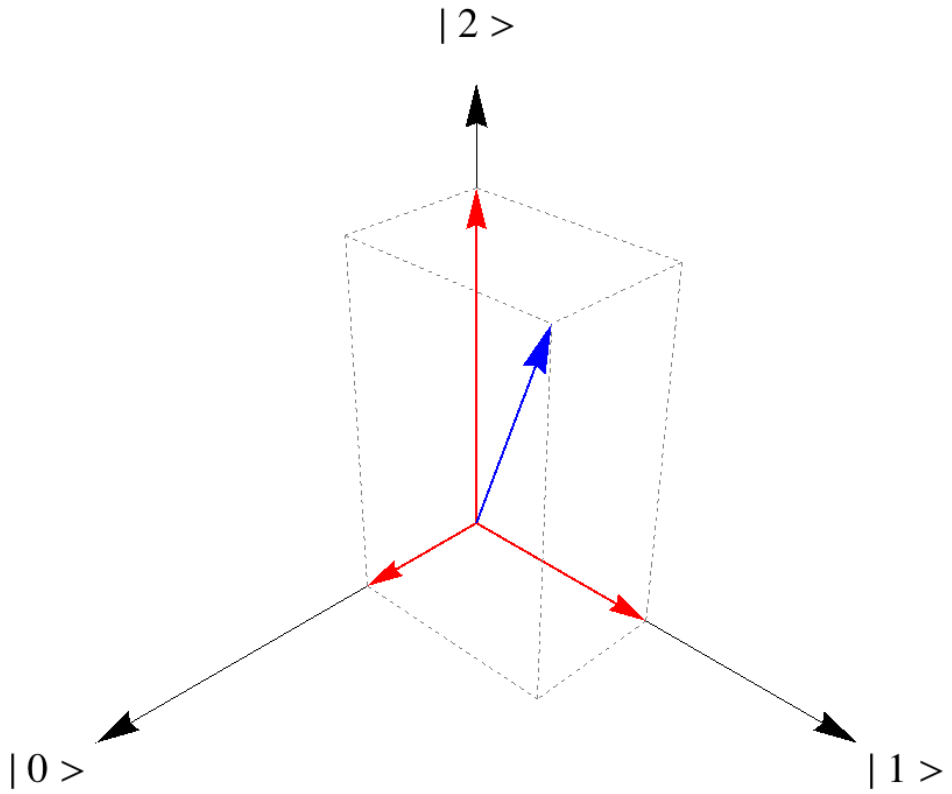
Knowing nothing but the rank of A, get good bounds on A's success probability

Toy example:

- Suppose just 3 functions: F_1, F_2, F_3 (say from {0,1}→{0,1})
- We are given F_i for a randomly chosen $i \in \{1,2,3\}$
- Goal: given oracle for F_i, determine i
- Rank = 1, 2, 3

Rank = 1

$|\psi_F\rangle$ independent* of z

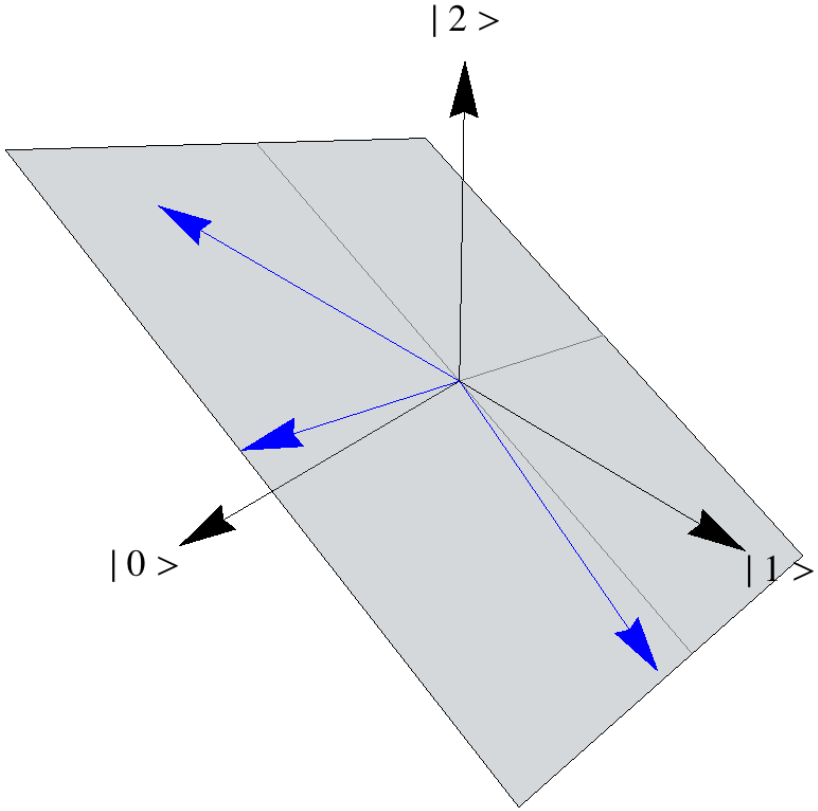


No matter what, win with probability 1/3

* Up to phase factors, which don't affect output distribution

Rank = 2

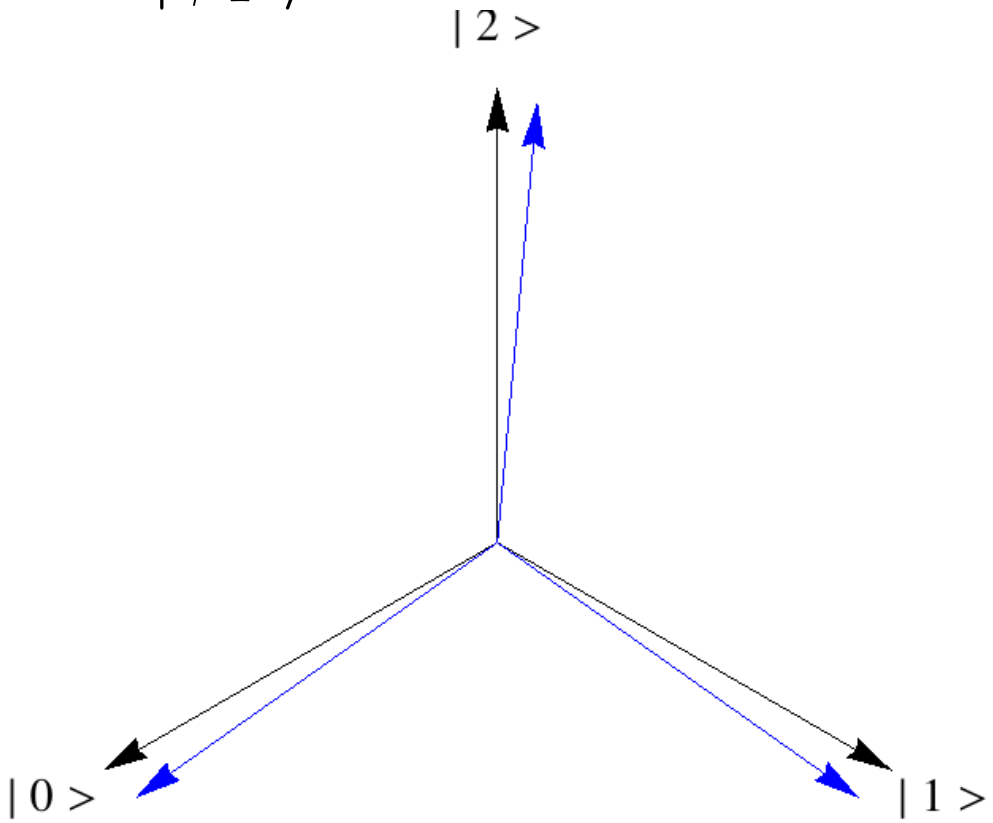
$|\psi_F\rangle$ depends on F, but still far from measurement basis



Can show best case probability is 2/3

Rank = 3

No constraints on $|\psi_F\rangle$



If $|\psi_{F_i}\rangle = |i\rangle$, win with probability 1=3/3

Toy Example

For this example, success prob is linear in rank.

Rank	Success Probability
1	1/3
2	2/3
3	3/3

Coincidence, or general phenomenon?

The Rank Method

Theorem: For any distribution D on F, the probability that a quantum algorithm with rank r learns F at k points is at most r times the probability a rank 1 algorithm learns F at k points.

To bound success prob, we need to:

- Bound success prob of rank 1 (0-query) algorithm - hopefully easy
- Bound rank of q-query algorithm - ???

Application to Oracle Interrogation

Recall:

- Given q queries to random oracle $F:X \rightarrow Y$
- Goal: Learn F at k points

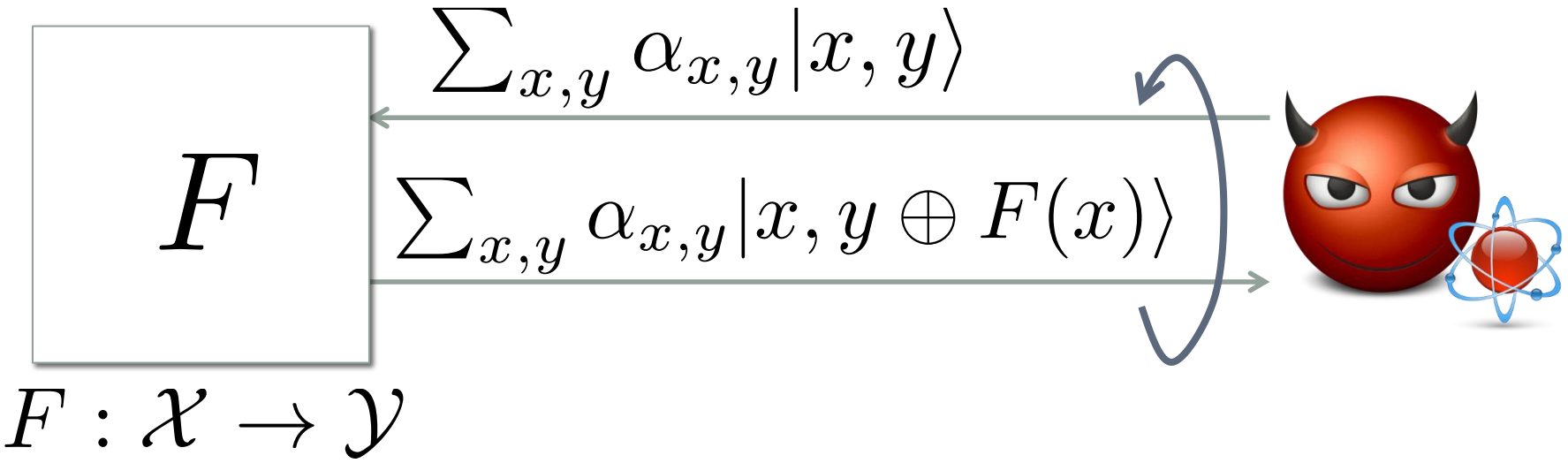
Best rank 1 (0-query) algorithm?

- Pick k distinct inputs arbitrarily, guess outputs arbitrarily
- Success prob: $1/|Y|^k$

Best q-query algorithm?

- $\text{Rank}/|Y|^k$
- Need to bound Rank of q query algorithms

Rank of Query Algorithms



Theorem: The rank of any algorithm making q queries to F: X→Y is at most

$$C_{|\mathcal{X}|,q,|\mathcal{Y}|} \equiv \sum_{r=0}^{q} \binom{|\mathcal{X}|}{r} (|\mathcal{Y}|-1)^r \leq \binom{|\mathcal{X}|}{q} |\mathcal{Y}|^q$$

Proof Idea

To bound rank of q-query algorithm, we give a spanning set for the possible final states.

Pick function F_0 (say, the all-zeros function)

Claim: $\{|\psi_F\rangle$: F differs from F_0 on at most q points$\}$ is a spanning set

Size of set: $$\sum_{r=0}^{q} \underbrace{\binom{|\mathcal{X}|}{r}(|\mathcal{Y}|-1)^r}_{\text{\# functions differing from } F_0 \text{ on exactly r points}}$$

Summary So Far

Theorem: For any distribution D on F, the probability that a quantum algorithm with rank r learns F at k points is at most r times the probability a rank 1 algorithm learns F at k points.

Theorem: The rank of any algorithm making q queries to H: X→Y is at most

$$C_{|\mathcal{X}|,q,|\mathcal{Y}|} \equiv \sum_{r=0}^{q} \binom{|\mathcal{X}|}{r} (|\mathcal{Y}|-1)^r \leq \binom{|\mathcal{X}|}{q} |\mathcal{Y}|^q$$

Application to Oracle Interrogation

Best success probability of q query algorithm

≤ best success probability of 0 query algorithm

times the largest rank of any q query algorithm

$$= \frac{1}{|\mathcal{Y}|^k} \times \sum_{r=0}^{q} \binom{|\mathcal{X}|}{r} (|\mathcal{Y}| - 1)^r \leq \frac{\binom{|\mathcal{X}|}{q}}{|\mathcal{Y}|^{k-q}}$$

Very large, trivial for many settings of parameters

Observation

If we want F(x) at k points, knowing F(x) at other points will not help us

→ might as well only query on superpositions of k points

$$\frac{1}{|\mathcal{Y}|^k} \sum_{r=0}^{q} \binom{|\mathcal{X}|}{r} (|\mathcal{Y}| - 1)^r$$

Exactly matches our algorithm!

$$P_Q(\mathcal{X}, \mathcal{Y}, q, k) \leq \frac{1}{|\mathcal{Y}|^k} \sum_{r=0}^{q} \binom{k}{r} (|\mathcal{Y}| - 1)^r \leq \frac{\binom{k}{q}}{|\mathcal{Y}|^{k-q}}$$

Putting it all Together

Theorem: No quantum algorithm, making q quantum queries to a random oracle F: X → Y, can learn F at k points, except with probability

$$\frac{1}{|\mathcal{Y}|^k} \sum_{r=0}^{q} \binom{k}{r} (|\mathcal{Y}| - 1)^r$$

Moreover, there is an algorithm that exactly achieves this bound

How do we analyze this probability?

Analyzing the Success Probability

Case 1: Fixed-size range (|Y| is fixed)

→ can let $k = cq$ for some constant c (depends on |Y|), and success with overwhelming probability

→ Example: Can make q queries to 3-bit oracle, evaluate at 1.14q points

Interrogation is **Easy**

Case 2: Exponential-size Range ($|Y| > 2^q$)

→ Even for k=q+1, success probability is exponentially small (in q)

Interrogation is **Hard**

Takeaways

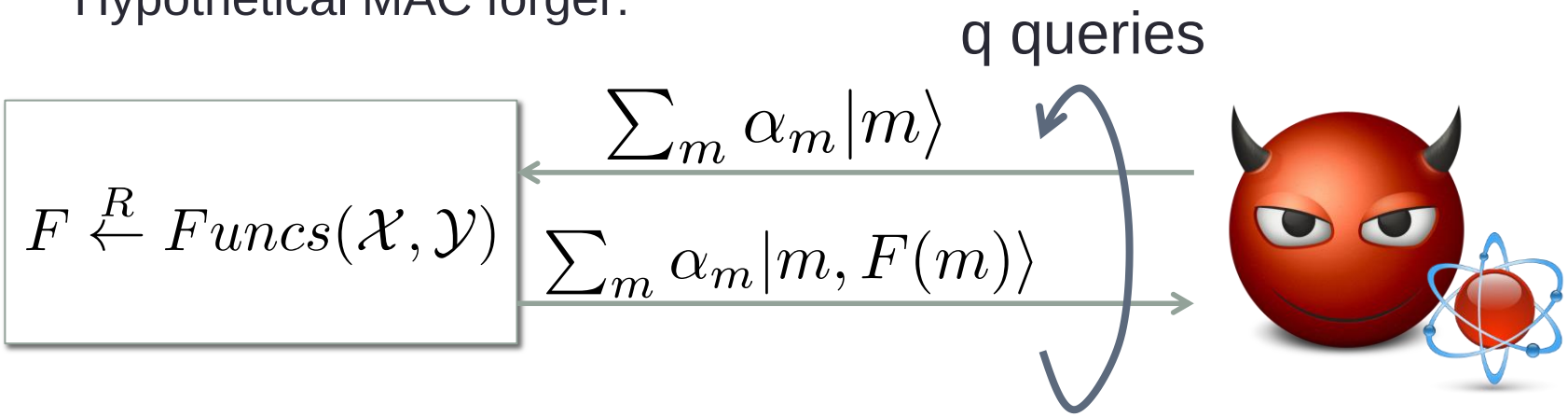
Cannot apply [vD'98] in parallel to each bit of the output

Quantum oracle interrogation is easier than classical oracle interrogation when the range is small

When range is large, quantum queries don't help

Back to MAC Security

Hypothetical MAC forger:



After q quantum queries to F, adversary gives q+1 input/output pairs of F

→ Solves oracle interrogation for case k=q+1

→ We just showed this is hard! (assuming large range)

→ No such forger exists ✓

A Generalization or Oracle Interrogation

What if the oracle F is not uniform?

- Many lattice-based schemes have non-uniform outputs
- Might have auxiliary information about F

Theorem: As long as each output is drawn independently (not necessarily identically) from a distribution that is unpredictable, then quantum oracle interrogation is still hard, even when k=q+1

Beyond MACs

There are many types of crypto we might want in the quantum setting:

- PRFs
- MACs
- Signatures
- Encryption

For each of these, we show that small modifications to existing schemes give schemes secure against quantum interrogations [Zha'12b, BZ'13a, BZ'13b]

Conclusion

Develop the Rank Method, a new quantum impossibility technique

Use Rank Method to exactly characterize success probability for quantum oracle interrogation of random oracles

- Also give algorithm that exactly achieves this probability

Use results to build crypto secure against quantum queries

→ No need for physical protection against quantum interrogation