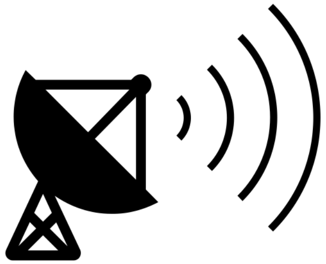
Tracing Quantum State Distinguishers via Backtracking

Mark Zhandry
NTT Research

Background

Traitor Tracing

[Chor-Fiat-Naor-Pinkas'94]



sk_1

sk_2

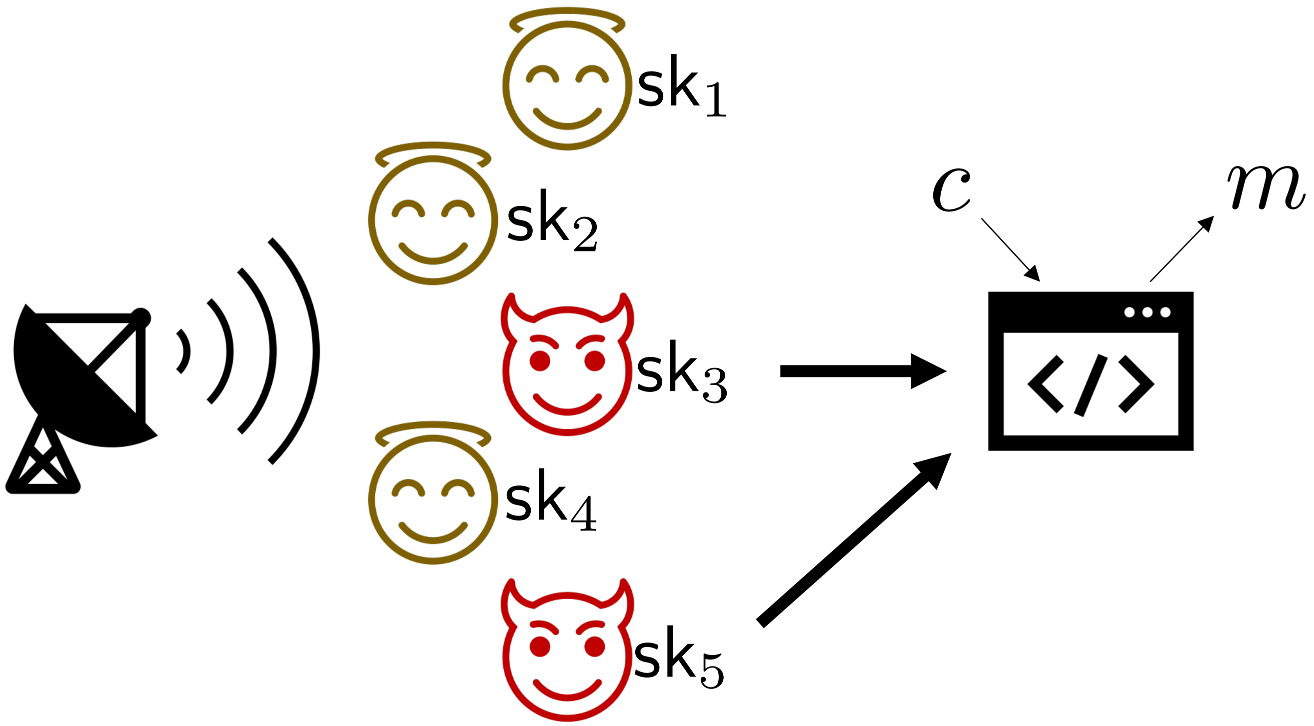
sk_3

sk_4

sk_5

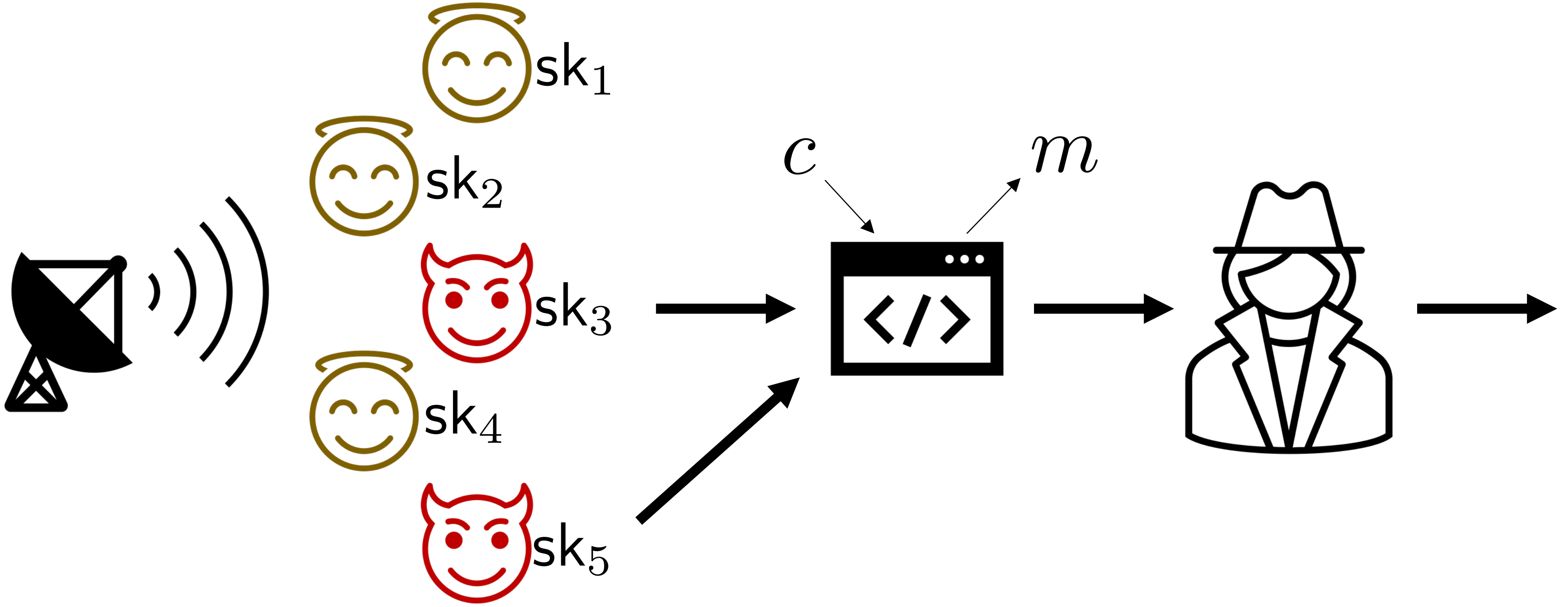
Traitor Tracing

[Chor-Fiat-Naor-Pinkas'94]



Traitor Tracing

[Chor-Fiat-Naor-Pinkas'94]



$S \subseteq \{3, 5\}$
$|S| \geq 1$

How Classical Traitor Tracing Works

$\mathcal{D} = \{D_q\}_{q \in [0,N]} =$ Family of ciphertext distributions

$D_N =$ Distribution of honest ciphertexts

$p_q =$ Success probability on D_q

How Classical Traitor Tracing Works

$\mathcal{D} = \{D_q\}_{q \in [0,N]} =$ Family of ciphertext distributions

$D_N =$ Distribution of honest ciphertexts

$p_q =$ Success probability on D_q

Guarantees:

- $p_N \gg 0$ by assumption that decoder works
- $p_0 \approx 0$
- $p_q \approx p_r$ if all users in $[r+1, q]$ honest

(The last two guarantees:) Enforced cryptographically

How Classical Traitor Tracing Works

$\mathcal{D} = \{D_q\}_{q \in [0,N]} =$ Family of ciphertext distributions

$D_N =$ Distribution of honest ciphertexts

$p_q =$ Success probability on D_q

Guarantees:

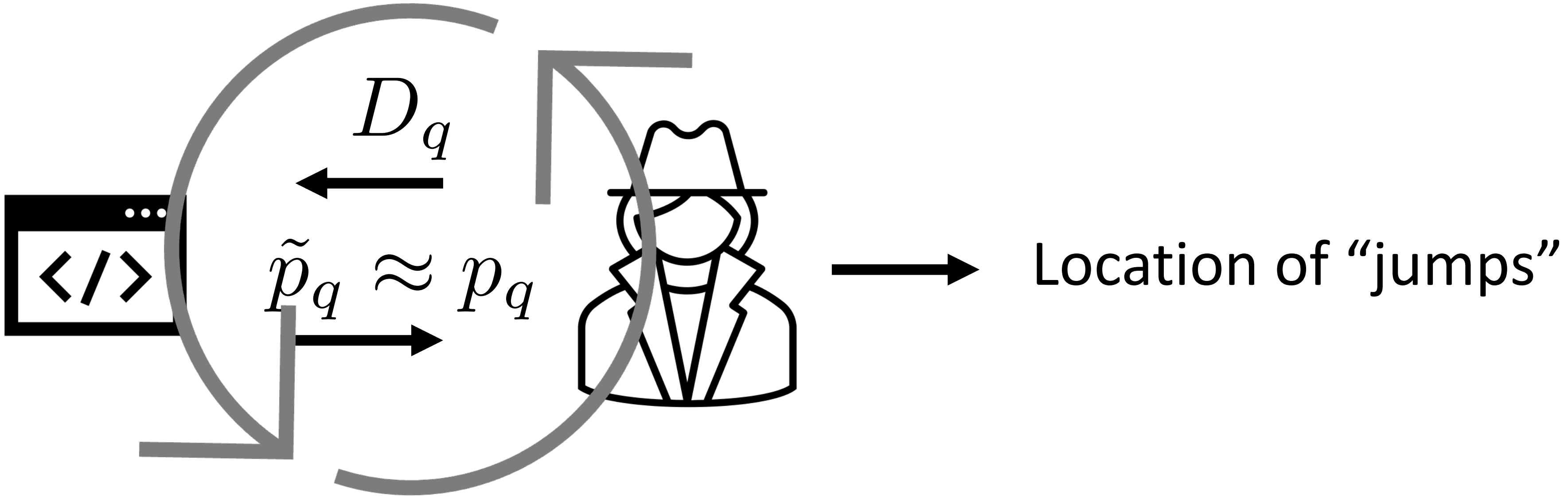
- $p_N \gg 0$ by assumption that decoder works
- $p_0 \approx 0$
- $p_q \approx p_r$ if all users in $[r+1, q]$ honest

} Enforced cryptographically (applies to the last two guarantees)

More general structures also used

How Classical Traitor Tracing Works

How Classical Traitor Tracing Works



$N =$ poly : Linear scan

$N =$ superpoly : Variant of binary search

[Boyle-Chung-Pass’14, Nishimaki-Wichs-Z’16]

Why super-poly domains?

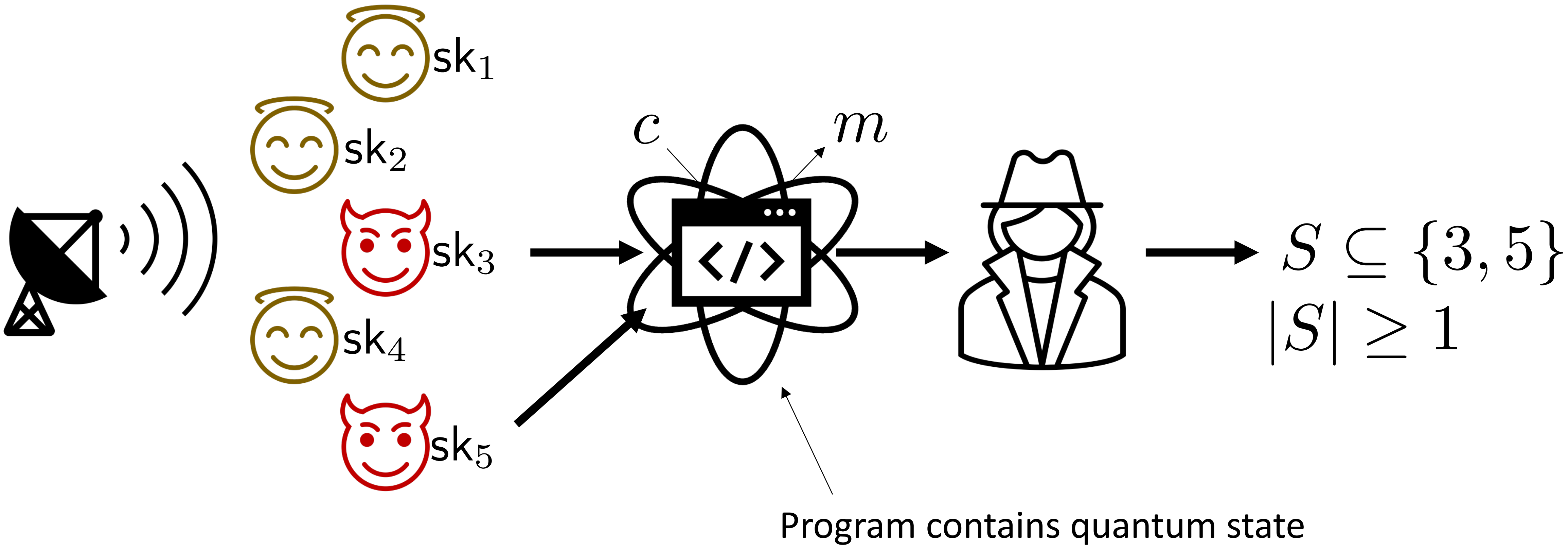
1) Can embed arbitrary info into key [Nishimaki-Wichs-Z'16]

2) Needed for other tracing structures (e.g. fingerprinting codes)

3) iO $\Longrightarrow$ diO for poly-many differing inputs [Boyle-Chung-Pass'14]
(algorithm inspiration for [Nishimaki-Wichs-Z'16])

Quantum Traitor Tracing

[Z'20]



Problem: quantum states disturbed by observations

⬇

p_q changes during tracing

Other issues as well: definitions + how to estimate p_q. Already handled by [Z'20]

How [Z'20] Works

$\mathcal{D} = \{D_q\}_{q \in [0, N]} =$ Family of ciphertext distributions

$D_N =$ Distribution of honest ciphertexts

How [Z'20] Works

$\mathcal{D} = \{D_q\}_{q \in [0,N]}$ = Family of ciphertext distributions

D_N = Distribution of honest ciphertexts

$q_0, q_1, q_2, \cdots$ = Tracer query sequence

$p_{q_0}, p_{q_1}, p_{q_2}, \cdots$ = Observed success probabilities

How [Z'20] Works

$\mathcal{D} = \{D_q\}_{q\in[0,N]}$ = Family of ciphertext distributions

D_N = Distribution of honest ciphertexts

$q_0, q_1, q_2, \cdots$ = Tracer query sequence

$p_{q_0}, p_{q_1}, p_{q_2}, \cdots$ = Observed success probabilities

Guarantees:

- $p_{q_0} \gg 0$ if $q_0 = N$ (no guarantees for p_N after first query)
- $p_0 \approx 0$ *always*
- $p_{q_i} \approx p_{q_{i-1}}$ if only honest users between q_i, q_{i-1}

Local consistency

[Z'20]:

- Local consistency good enough for linear scan / $N =$ poly
- Fails for binary search / $N =$ superpoly

Always valid outcome with just local consistency:

$$p_{q_0}, p_{q_1}, p_{q_2}, \cdots = \underbrace{1, 1, 1, 0, 0, 0, \cdots}_{\text{Only log bits of info}}$$

[Z'20]:

- Good enough for linear scan / $N = \text{poly}$
- Fails for binary search / $N = \text{superpoly}$

Always valid outcome with just local consistency:

$$p_{q_0}, p_{q_1}, p_{q_2}, \cdots = \underbrace{1, 1, 1, 0, 0, 0, \cdots}_{\text{Only log bits of info}}$$

[Kitagawa-Nishimaki'22]: global consistency, but only when no collusions

This Work

This Work

Guarantees:

- $p_{q_0} \gg 0$ if $q_0 = N$
- $p_0 \approx 0$ *always*
- $p_{q_i} \approx p_{q_{i-1}}$ if only honest users between q_i, q_{i-1}
- $p_{q_i} \approx p_{q_{i-2}}$ if $q_i = q_{i-2}$

NEW: single-step rewinding

Enforced using quantum state repair [Chiesa-Ma-Spooner-Z'21]

Note: No guarantees for $q_i = q_{i-k}, k \geq 3$

Case $k = 1$ Implied by local consistency

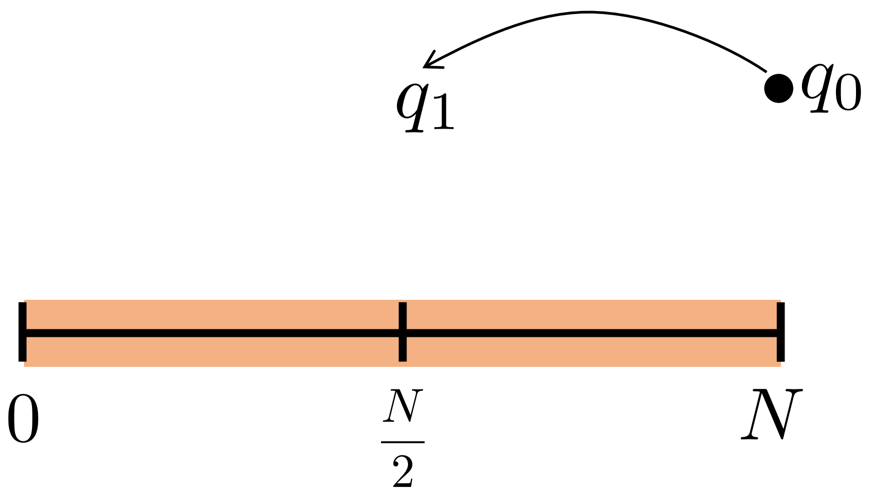
"Hesitant" Algorithms

Idea: always make sure one of last two queries has large p_{q_i}

$\Longrightarrow$ if ever get small p_{q_i}, immediately backtrack with $q_{i+1} = q_{i-1}$

Otherwise, all future p_{q_i} may be small

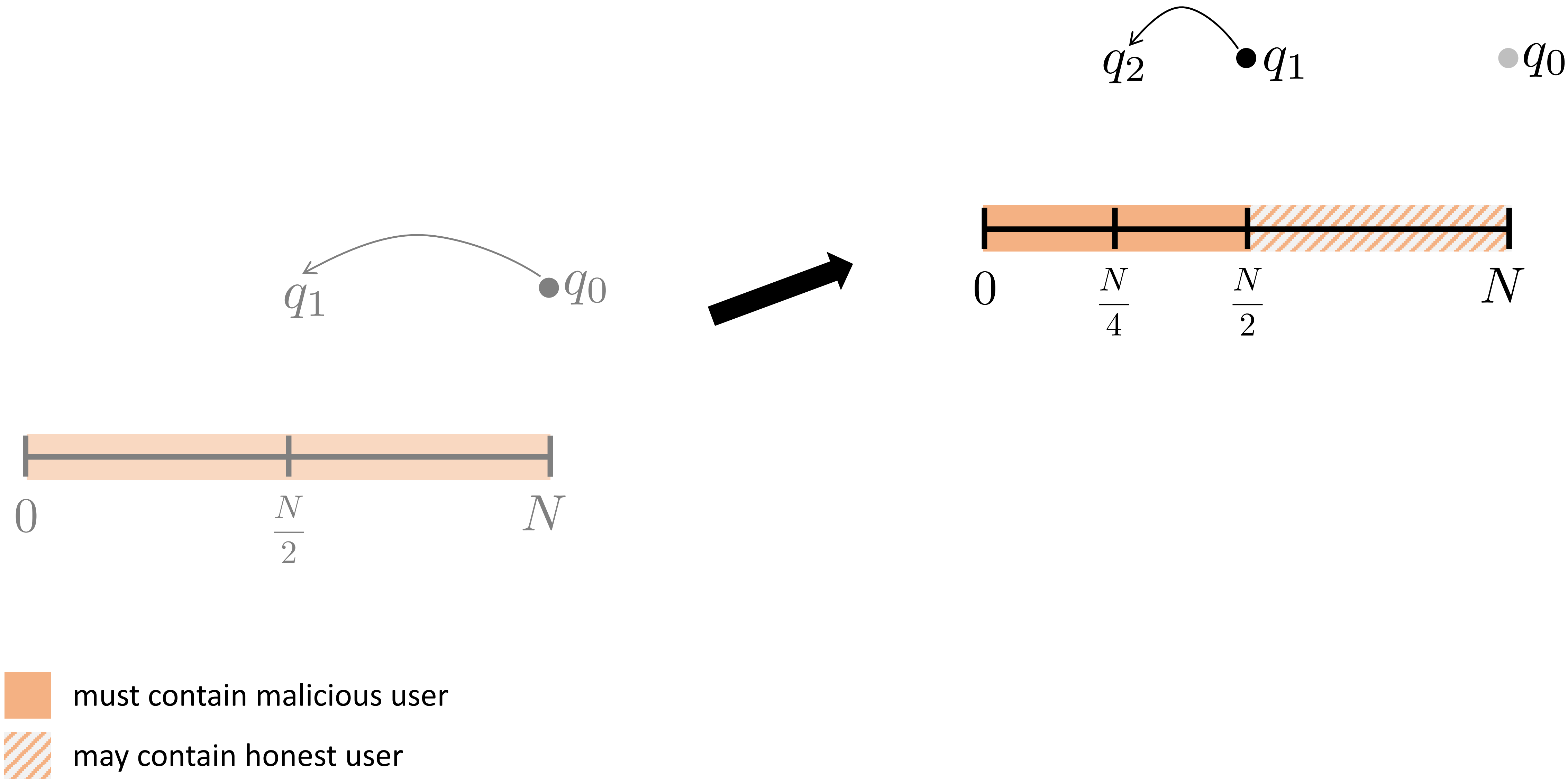
Hesitant Binary Search



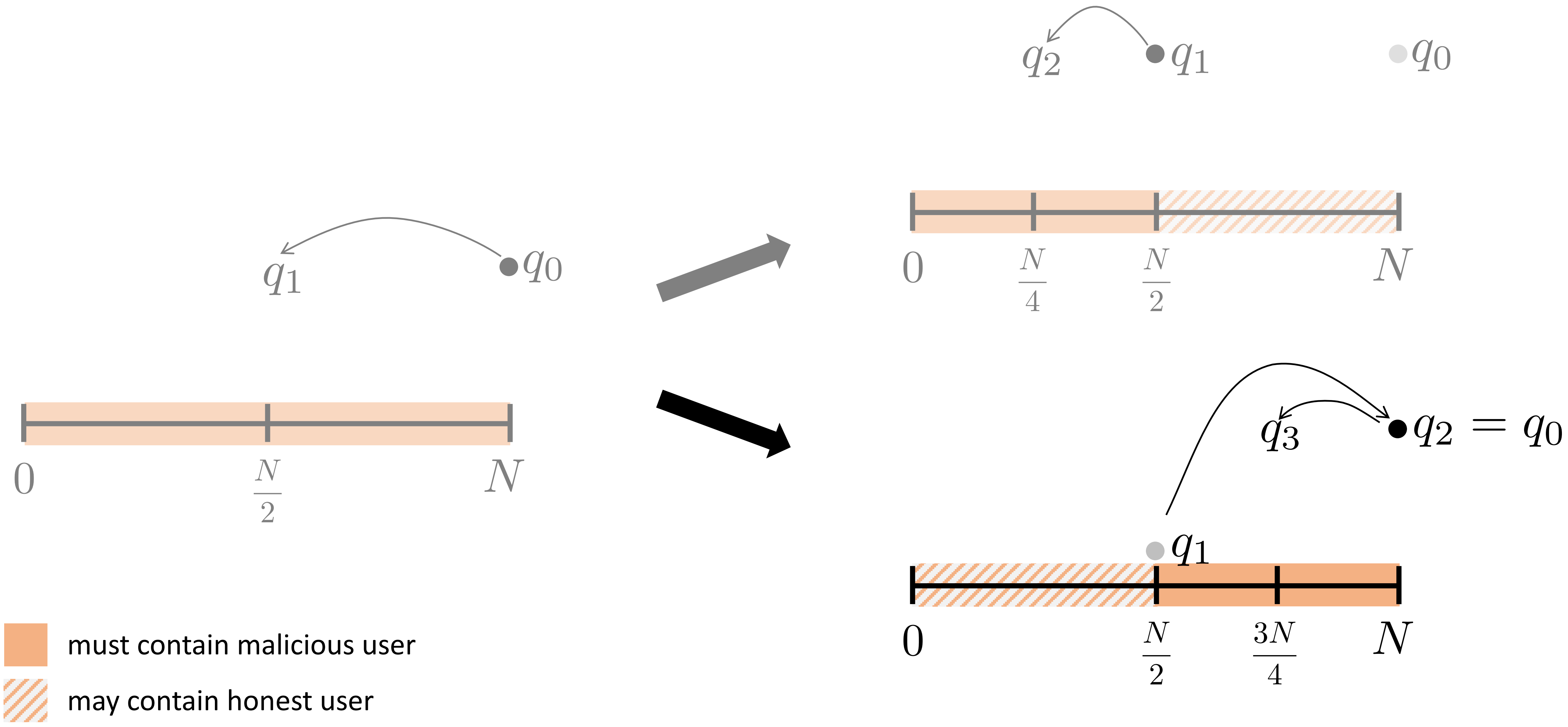
must contain malicious user

may contain honest user

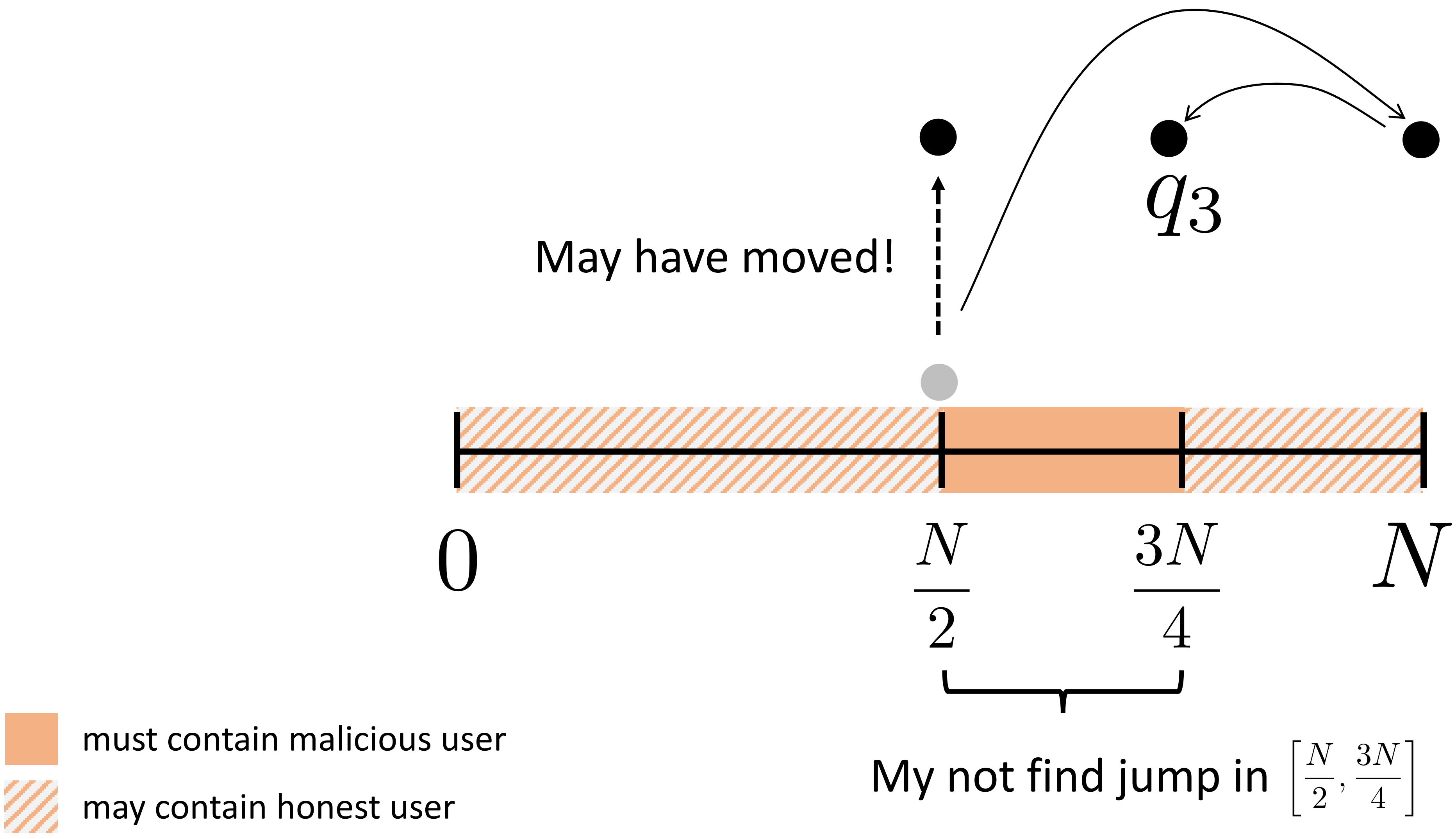
Hesitant Binary Search



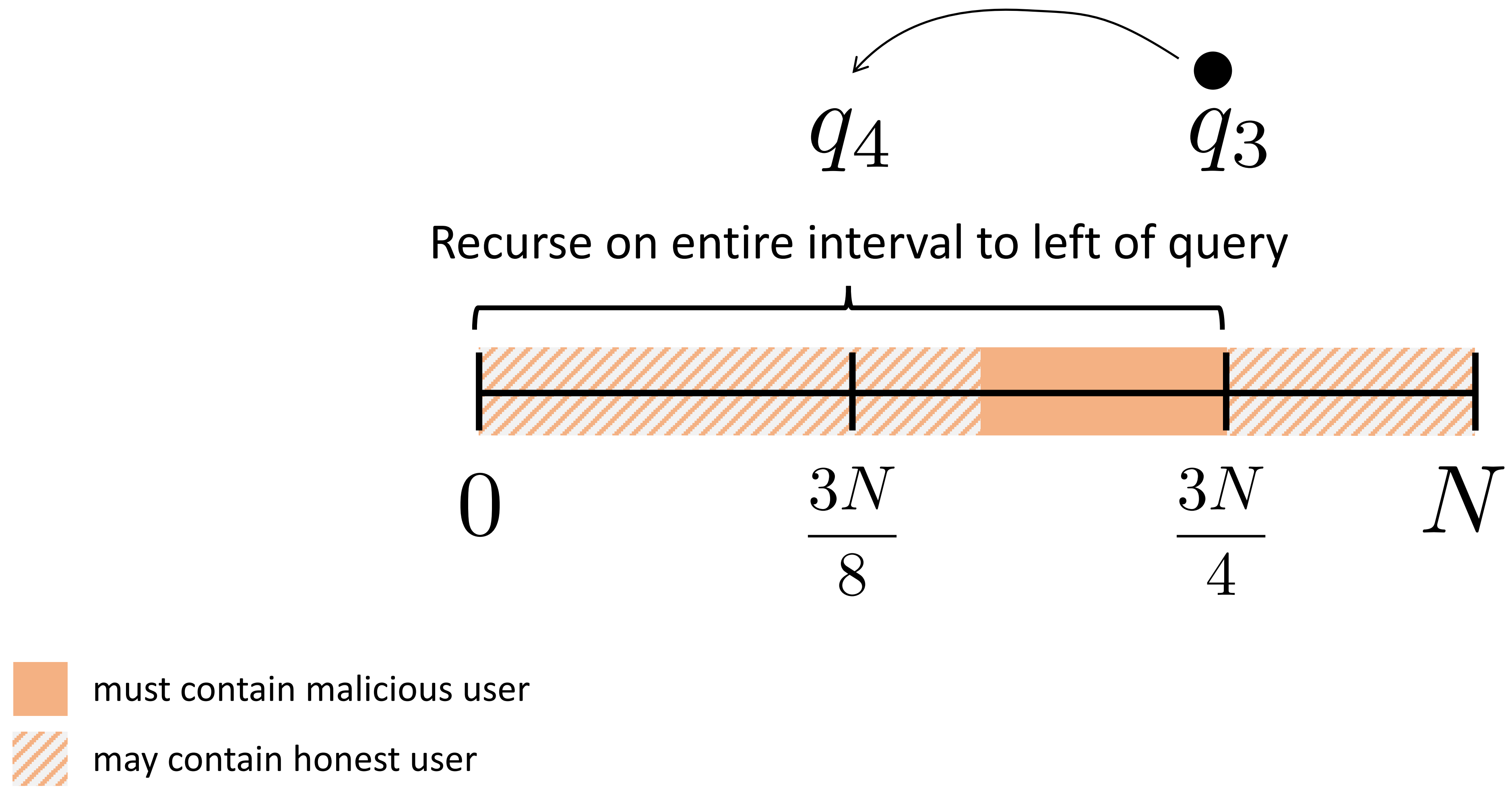
Hesitant Binary Search



Hesitant Binary Search



Hesitant Binary Search



Thm: Alg finds malicious user in $O(k \log^2 N)$ steps

$k =$ upper bound on #(malicious users)

Compare to classical binary search: $O(k \log N)$

[Boyle-Chung-Pass'14, Nishimaki-Wichs-Z'16]

Results

Embedded identity collusion-resistant traitor tracing against quantum decoders

- iO $\Longrightarrow$ optimal params
- PKE $\Longrightarrow |\text{params}| = \text{poly}(\#(\text{users}))$

Results

Embedded identity collusion-resistant traitor tracing against quantum decoders

- iO $\Longrightarrow$ optimal params
- PKE $\Longrightarrow$ $|\text{params}| = \text{poly}(\#(\text{users}))$

iO $\Longrightarrow$ diO w/ quantum auxiliary input for poly-many differing inputs

Results

Embedded identity collusion-resistant traitor tracing against quantum decoders

- iO $\Longrightarrow$ optimal params
- PKE $\Longrightarrow |\text{params}| = \text{poly}(\#(\text{users}))$

iO $\Longrightarrow$ diO w/ quantum auxiliary input for poly-many differing inputs

PKE $\Longrightarrow$ col-res. TT against quantum decoders, $|\text{ctxt}| = O(1), |\text{pk}| = |\text{sk}| = \text{poly}(\#(\text{users}))$

PKE $\Longrightarrow$ bounded collusion TT against quantum decoders, $|\text{params}| = \text{poly}(\text{collusion bound})$

Develop hesitant algorithms for fingerprinting code-based traitor tracing [Chor-Fiat-Naor-Pinkas'94,Boneh-Naor'08,Sirvent'08, Billet-Phan'08]

Thanks!