Software Side-Channel Analysis: Attack Synthesis

Lucas Bang

Dissertation Defense

Committee:
Tevfik Bultan (chair)
Ömer Eğecioğlu
Ben Hardekopf

Publications during PhD

- Aydin, **Bang**, Bultan. [CAV 2015]
  "Automata-Based Model Counting for String Constraints."
- **Bang**, Aydin, Bultan. [FSE 2015]
  "Automatically Computing Path Complexity of Programs."
- **Bang**, Aydin, Phan, Pasareanu, Bultan. [FSE 2016]
  "String Analysis for Side Channels with Segmented Oracles."
- Phan, **Bang**, Pasareanu, Malacaria, Bultan. [CSF 17]
  "Synthesis of Adaptive Side-Channel Attacks."
- **Bang**, Rosner, Bultan. [Euro S&P 2018]
  "Online Synthesis of Adaptive Side-Channel Attacks Based On Noisy Observations."
- Aydin, Eiers, **Bang**, Brennan, Gavrilov, Yu, Bultan. [FSE 2018 (accepted)]
  "Parameterized Model Counting for String and Numeric Constraints."

Submitted papers

- Tsiskaridze, **Bang**, McMahan, Bultan, Sherwood.
  "Information Leakage in Arbiter Protocols."
- Saha, Kadron, Eiers, **Bang**, Bultan.
  "Attack Synthesis for Strings via Incremental Model Counting and Meta-Heuristics."

Publications during PhD

- Aydin, **Bang**, Bultan. [CAV 2015]
  "Automata-Based Model Counting for String Constraints."
- **Bang**, Aydin, Bultan. [FSE 2015]
  "Automatically Computing Path Complexity of Programs."
- **Bang**, Aydin, Phan, Pasareanu, Bultan. [FSE 2016]
  "String Analysis for Side Channels with Segmented Oracles."
- Phan, **Bang**, Pasareanu, Malacaria, Bultan. [CSF 17]
  "Synthesis of Adaptive Side-Channel Attacks."
- **Bang**, Rosner, Bultan. [Euro S&P 2018]
  "Online Synthesis of Adaptive Side-Channel Attacks Based On Noisy Observations."
- Aydin, Eiers, **Bang**, Brennan, Gavrilov, Yu, Bultan. [FSE 2018 (accepted)]
  "Parameterized Model Counting for String and Numeric Constraints."

Submitted papers

- Tsiskaridze, **Bang**, McMahan, Bultan, Sherwood.
  "Information Leakage in Arbiter Protocols."
- Saha, Kadron, Eiers, **Bang**, Bultan.
  "Attack Synthesis for Strings via Incremental Model Counting and Meta-Heuristics."

Publications during PhD

- Aydin, **Bang**, Bultan. [CAV 2015]
  "Automata-Based Model Counting for String Constraints."
- **Bang**, Aydin, Bultan. [FSE 2015]
  "Automatically Computing Path Complexity of Programs."
- **Bang**, Aydin, Phan, Pasareanu, Bultan. [FSE 2016]
  "String Analysis for Side Channels with Segmented Oracles."
- Phan, **Bang**, Pasareanu, Malacaria, Bultan. [CSF 17]
  "Synthesis of Adaptive Side-Channel Attacks."
- **Bang**, Rosner, Bultan. [Euro S&P 2018]
  "Online Synthesis of Adaptive Side-Channel Attacks Based On Noisy Observations."
- Aydin, Eiers, **Bang**, Brennan, Gavrilov, Yu, Bultan. [FSE 2018 (accepted)]
  "Parameterized Model Counting for String and Numeric Constraints."

Submitted papers

- Tsiskaridze, **Bang**, McMahan, Bultan, Sherwood.
  "Information Leakage in Arbiter Protocols."
- Saha, Kadron, Eiers, **Bang**, Bultan.
  "Attack Synthesis for Strings via Incremental Model Counting and Meta-Heuristics."

Side-Channel Attacks

TIME

Monday, Aug. 13, 1990

And Bomb The Anchovies

By Paul Gray

Delivery people at various Domino's pizza outlets in and around Washington claim that they have learned to anticipate big news baking at the White House or the Pentagon by the upsurge in takeout orders. Phones usually start ringing some 72 hours before an official announcement. "We know," says one pizza runner. "Absolutely. Pentagon orders doubled up the night before the Panama attack; same thing happened before the Grenada invasion." Last Wednesday, he adds, "we got a lot of orders, starting around midnight. We figured something was up." This time the big news arrived quickly: Iraq's surprise invasion of Kuwait.

TIME

Monday, Aug. 13, 1990

And Bomb The Anchovies

By Paul Gray

Delivery people at various Domino's pizza outlets in and around Washington claim that they have learned to anticipate big news baking at the White House or the Pentagon by the upsurge in takeout orders. Phones usually start ringing some 72 hours before an official announcement. "We know," says one pizza runner. "Absolutely. Pentagon orders doubled up the night before the Panama attack; same thing happened before the Grenada invasion." Last Wednesday, he adds, "we got a lot of orders, starting around midnight. We figured something was up." This time the big news arrived quickly: Iraq's surprise invasion of Kuwait.

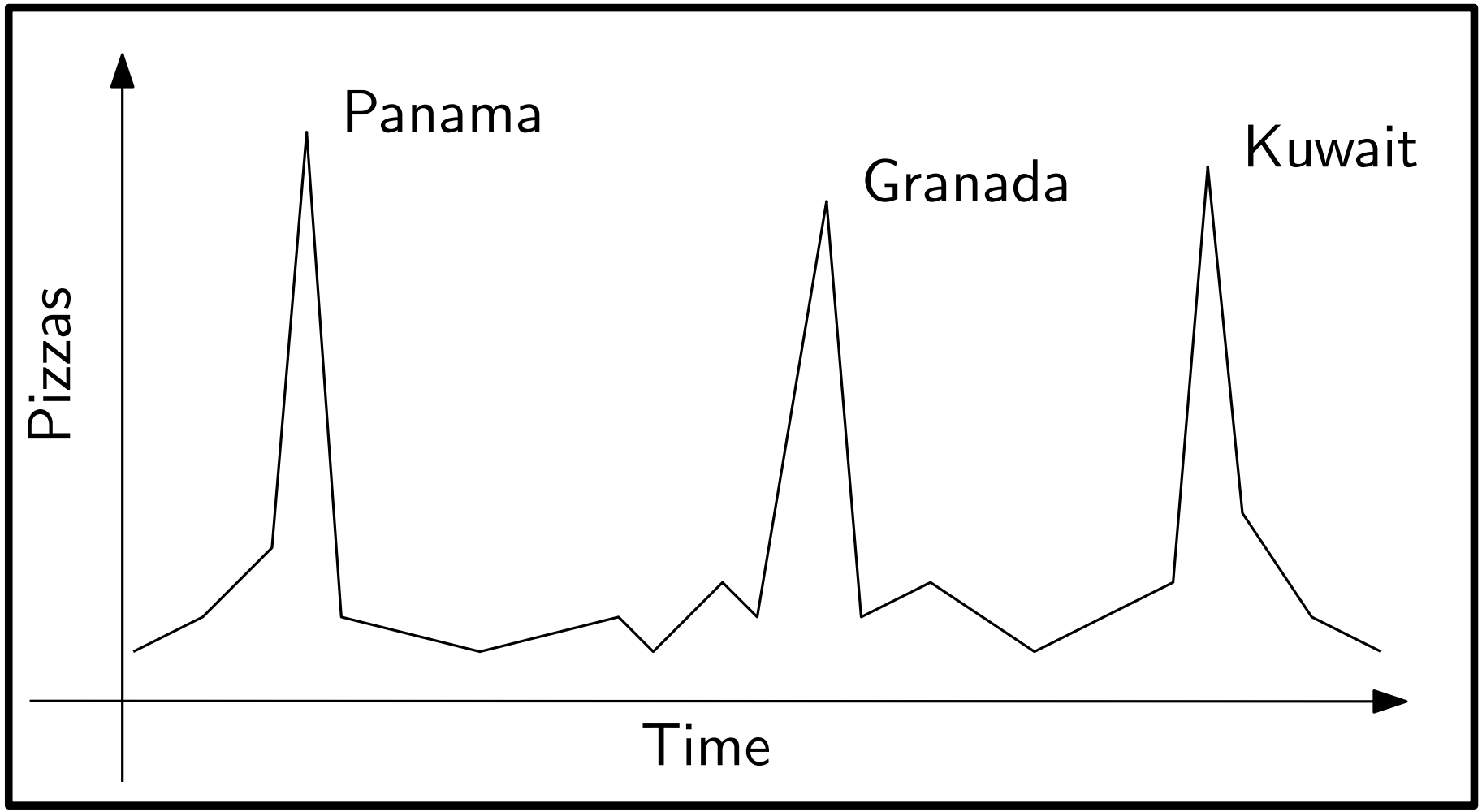
TIME

Monday, Aug. 13, 1990

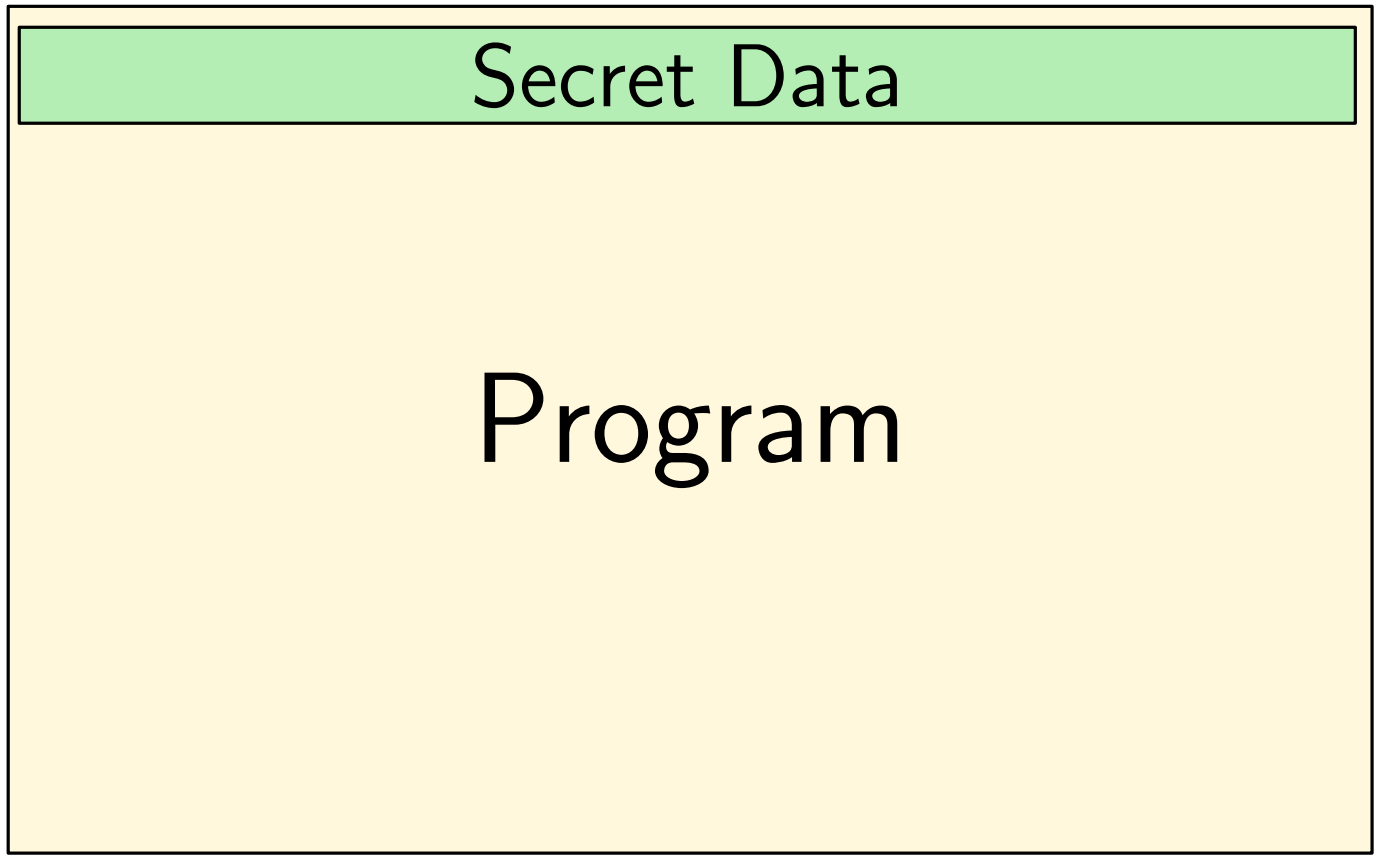
And Bomb The Anchovies

By Paul Gray

Delivery people at various Domino's pizza outlets in and around Washington claim that they have learned to anticipate big news baking at the White House or the Pentagon by the upsurge in takeout orders. Phones usually start ringing some 72 hours before an official announcement. "We know," says one pizza runner. "Absolutely. Pentagon orders doubled up the night before the Panama attack. same thing happened before the Grenada invasion." Last Wednesday, he adds, "we got a lot of orders, starting around midnight. We figured something was up." This time the big news arrived quickly: Iraq's surprise invasion of Kuwait.

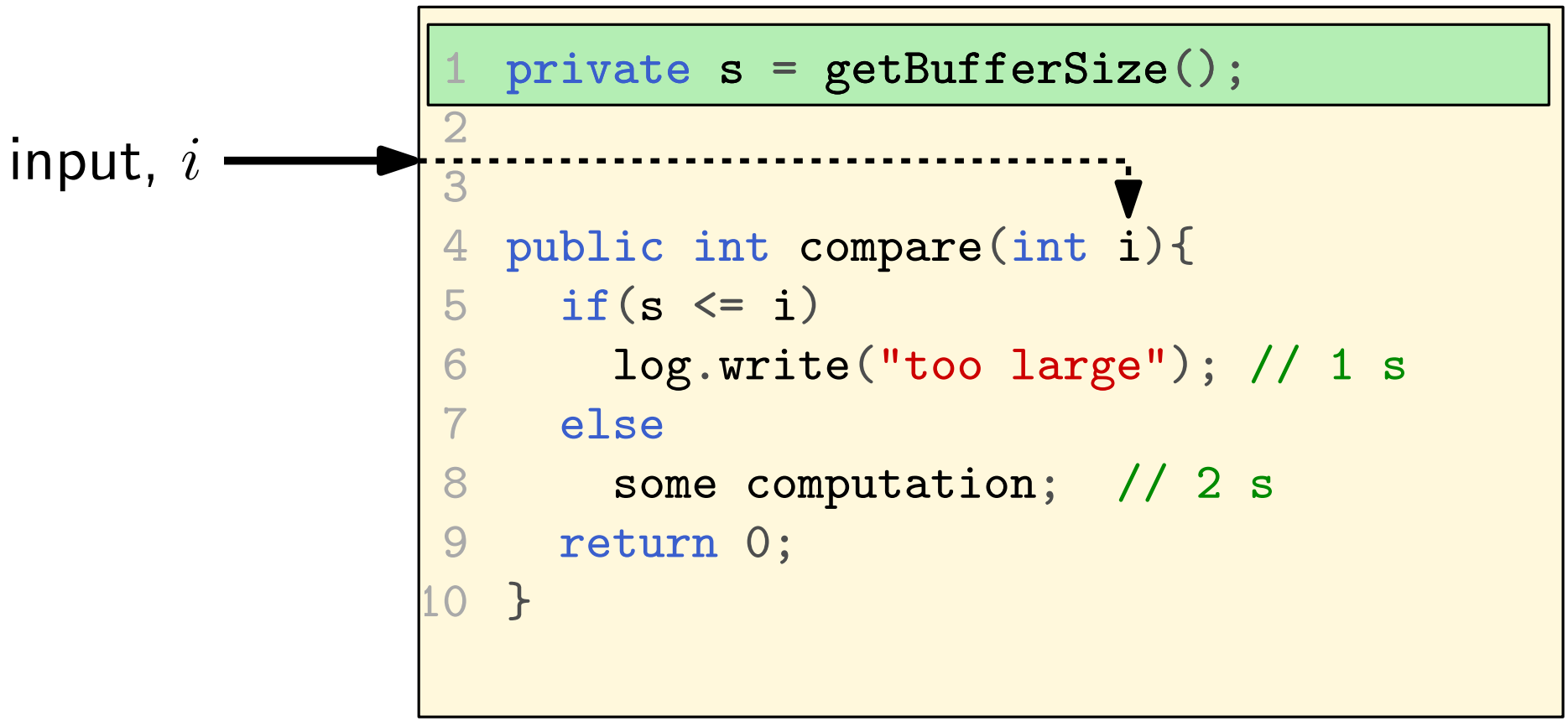


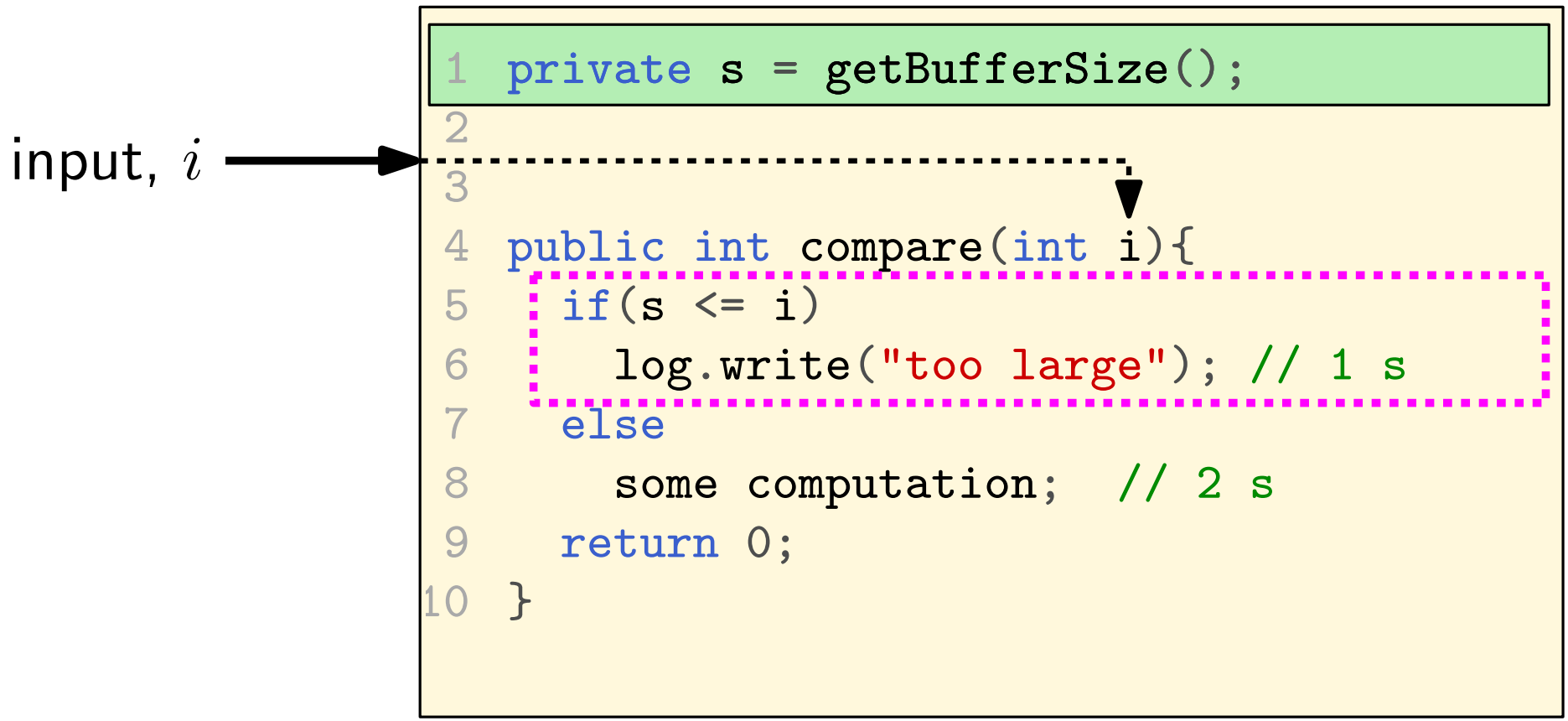
Side channel: learn secrets through indirect observation.


Secret Data
Program

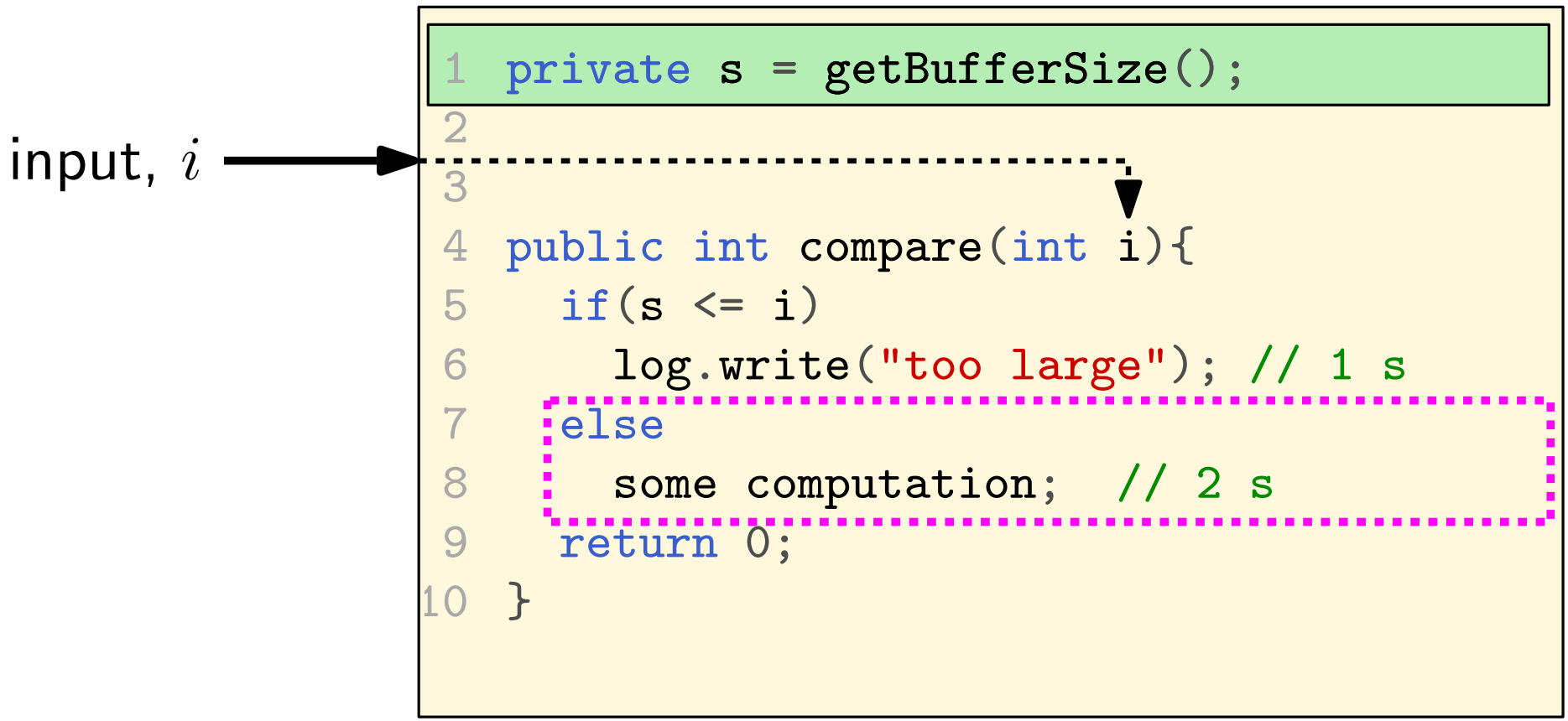
```
1  private s = getBufferSize();
```

Program


input, i
1 private s = getBufferSize();
2
3
4 public int compare(int i){
5 if(s <= i)
6 log.write("too large"); // 1 s
7 else
8 some computation; // 2 s
9 return 0;
10 }



$$s \leq i \Rightarrow o = 1$$



$$s \leq i \Rightarrow o = 1$$

$$s > i \Rightarrow o = 2$$

input, i

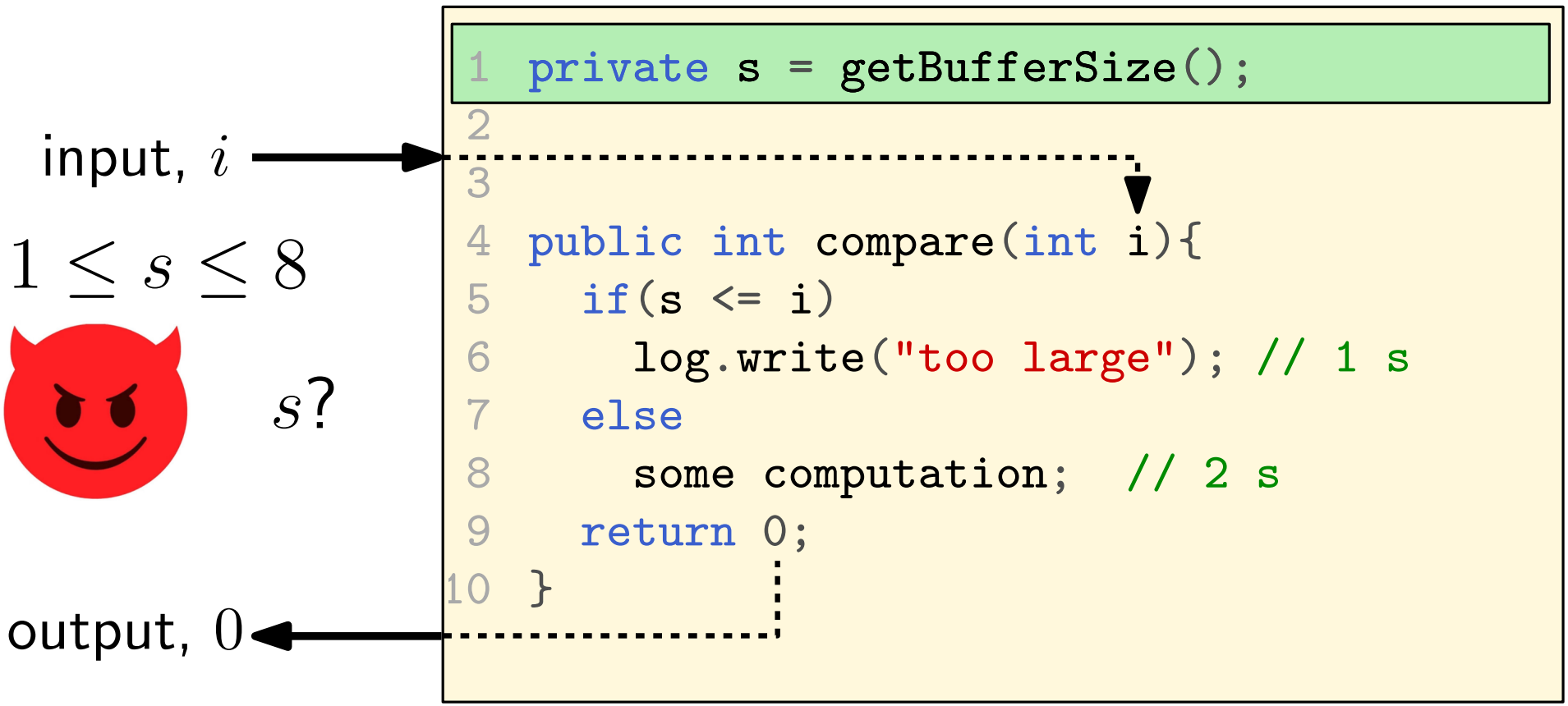
$1 \leq s \leq 8$

s?

```
1  private s = getBufferSize();
2
3
4  public int compare(int i){
5    if(s <= i)
6      log.write("too large"); // 1 s
7    else
8      some computation;  // 2 s
9    return 0;
10 }
```

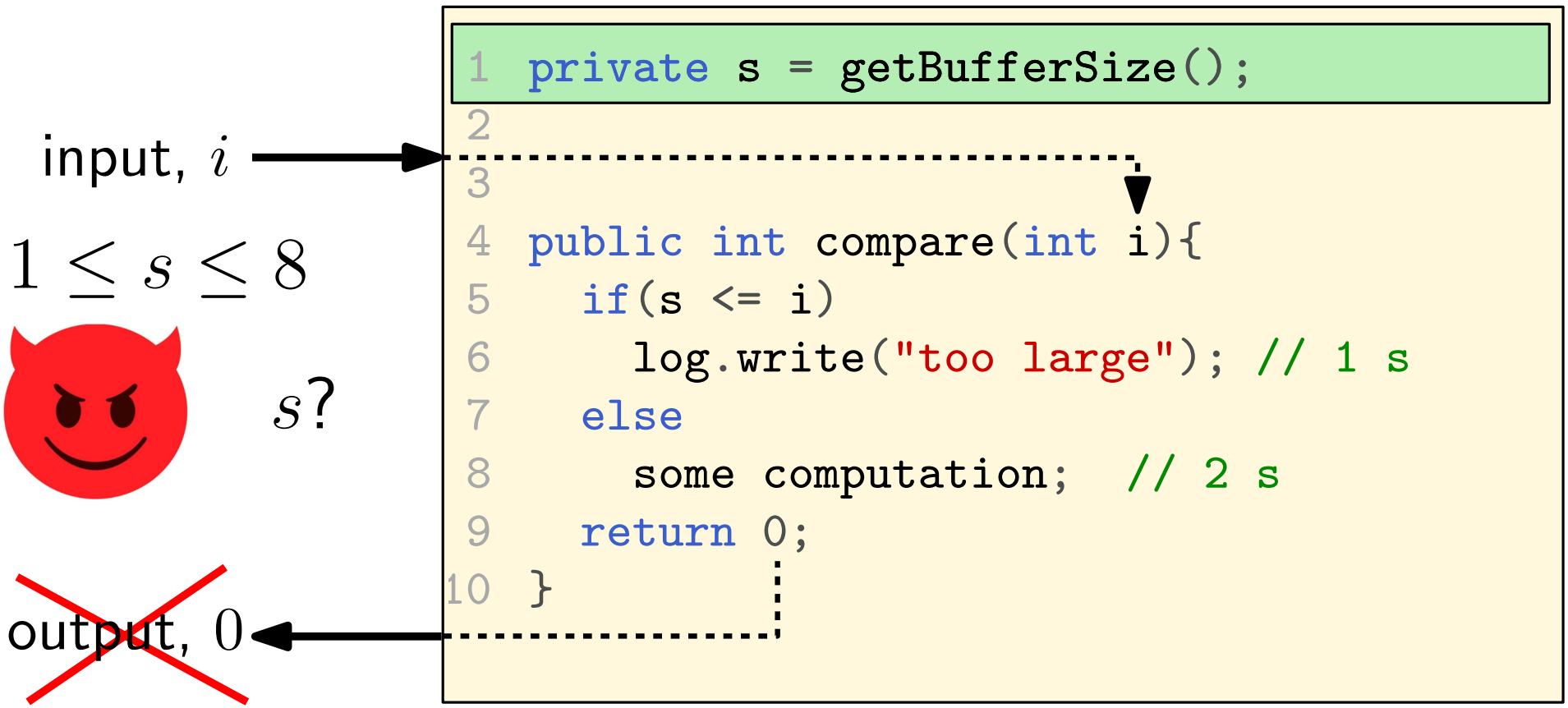
$$s \leq i \Rightarrow o = 1$$

$$s > i \Rightarrow o = 2$$



$$s \leq i \Rightarrow o = 1$$

$$s > i \Rightarrow o = 2$$



$$s \leq i \Rightarrow o = 1$$

$$s > i \Rightarrow o = 2$$

input, i

$1 \leq s \leq 8$

s?

```
1  private s = getBufferSize();
2
3
4  public int compare(int i){
5    if(s <= i)
6      log.write("too large"); // 1 s
7    else
8      some computation;  // 2 s
9    return 0;
10 }
```

$$s \leq i \Rightarrow o = 1$$

$$s > i \Rightarrow o = 2$$

input, i

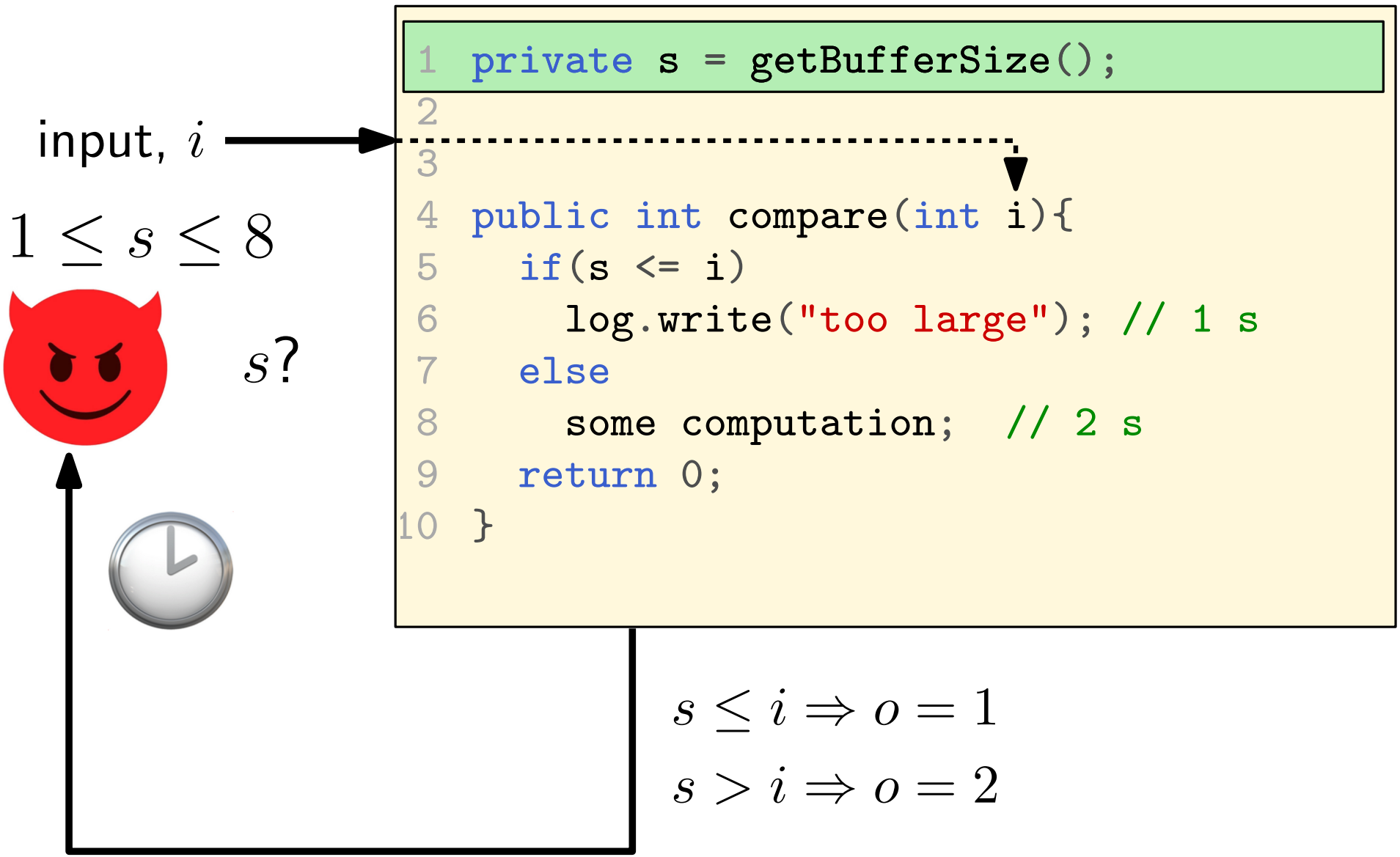
$1 \leq s \leq 8$

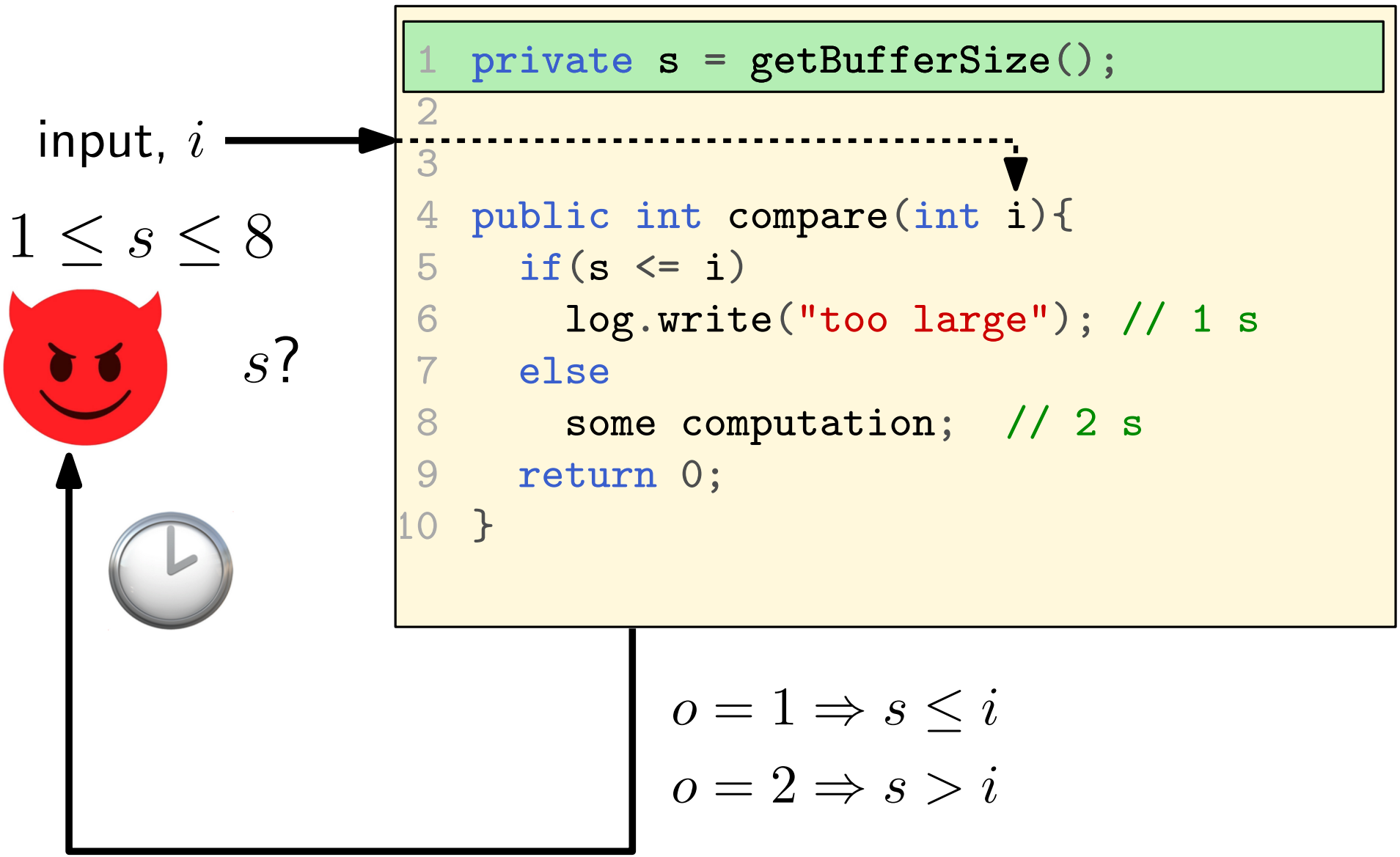
s?

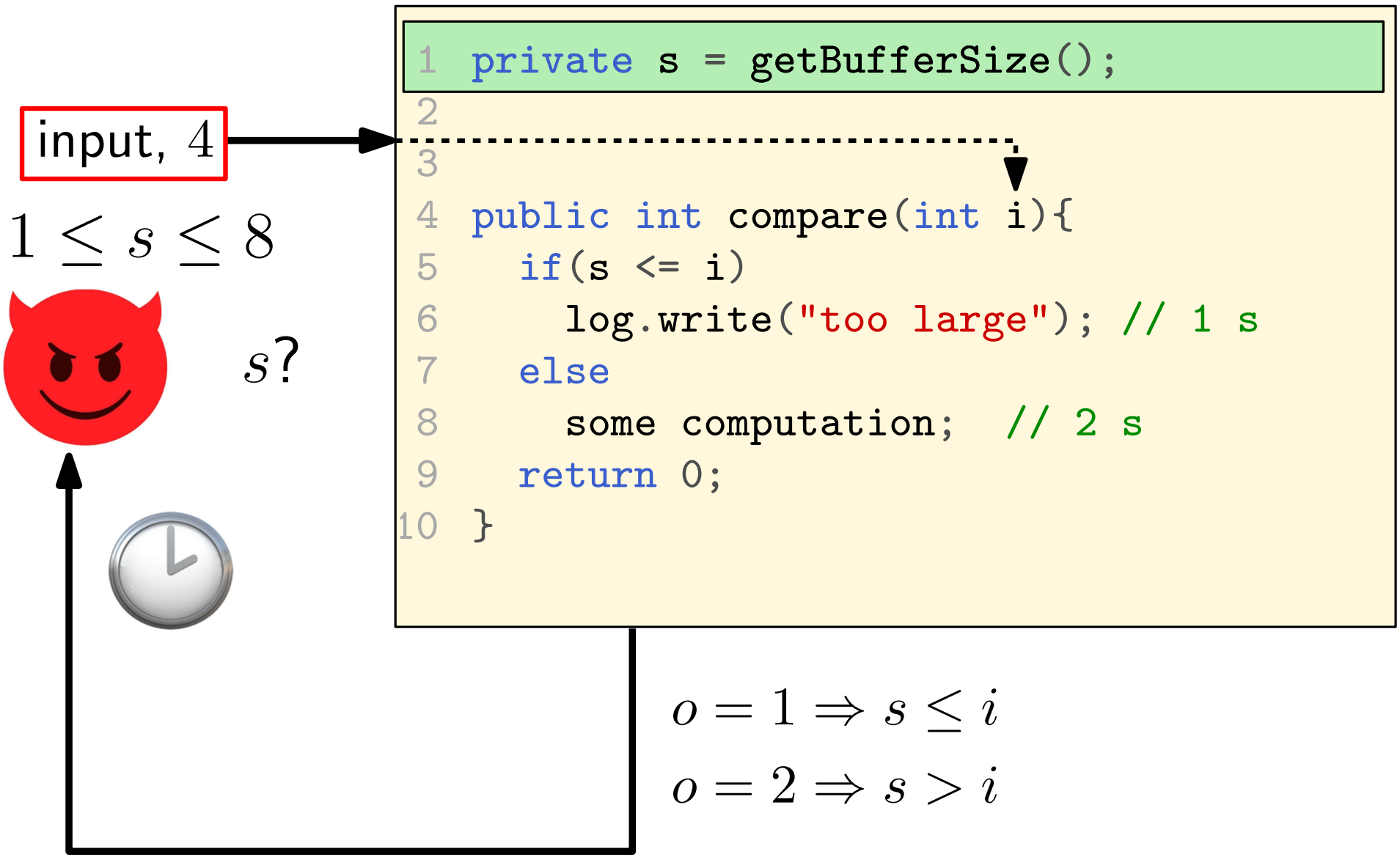
```
1  private s = getBufferSize();
2
3
4  public int compare(int i){
5    if(s <= i)
6      log.write("too large"); // 1 s
7    else
8      some computation;  // 2 s
9    return 0;
10 }
```

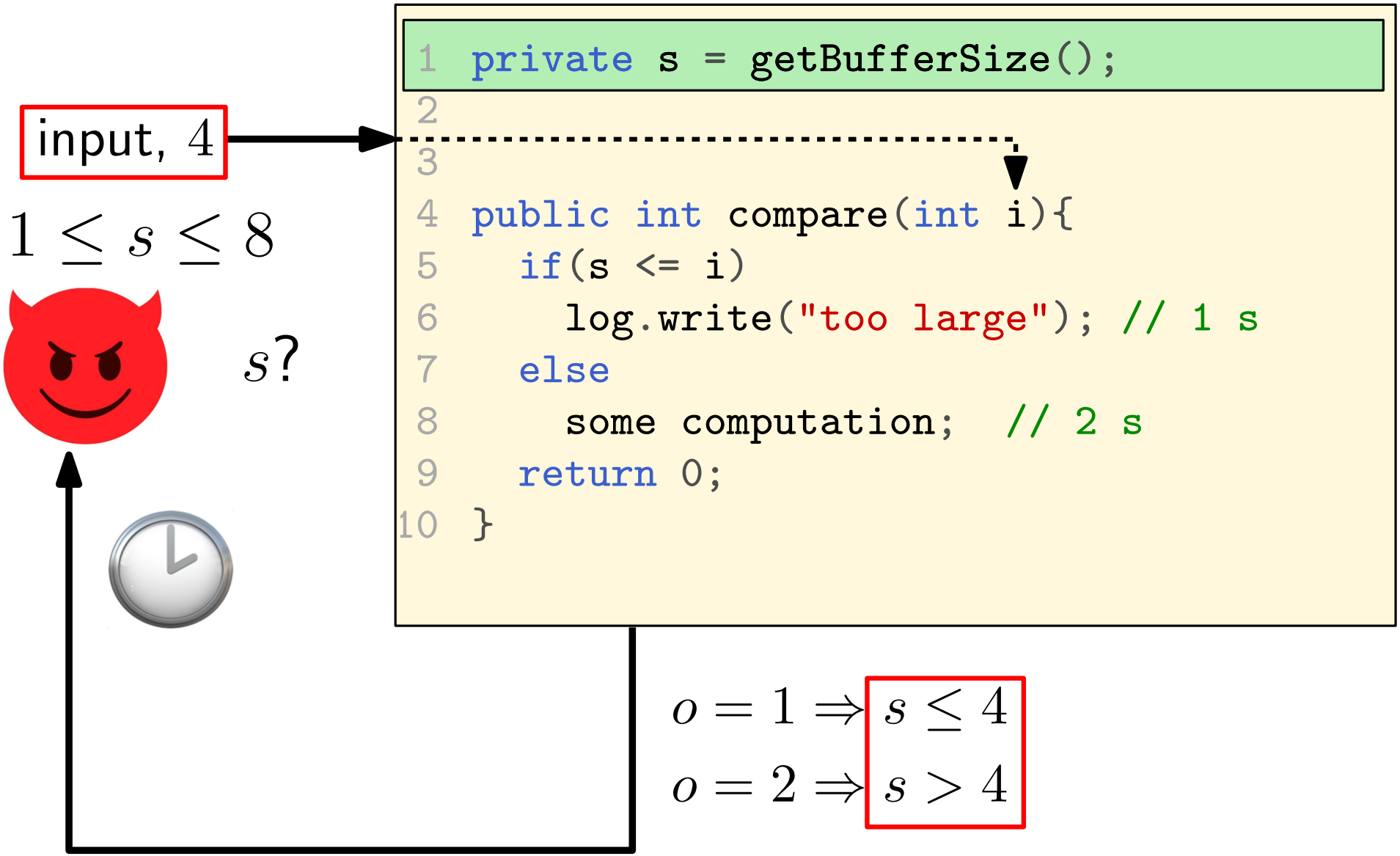
$$s \leq i \Rightarrow o = 1$$

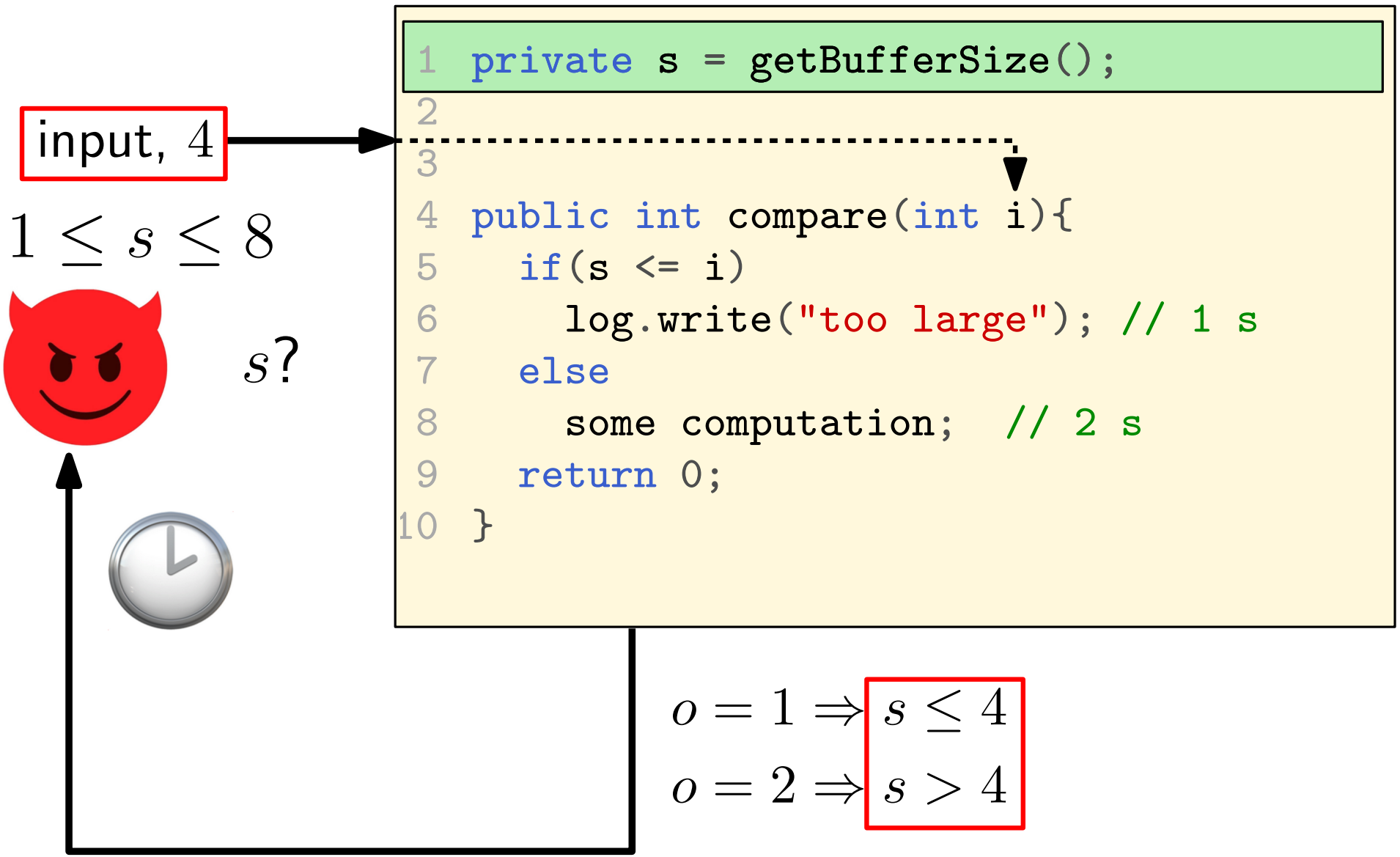
$$s > i \Rightarrow o = 2$$


input, i
$1 \leq s \leq 8$
s?
1 private s = getBufferSize();
2
3
4 public int compare(int i){
5 if(s <= i)
6 log.write("too large"); // 1 s
7 else
8 some computation; // 2 s
9 return 0;
10 }
$s \leq i \Rightarrow o = 1$
$s > i \Rightarrow o = 2$


input, i
$1 \leq s \leq 8$
s?
1 private s = getBufferSize();
2
3
4 public int compare(int i){
5 if(s <= i)
6 log.write("too large"); // 1 s
7 else
8 some computation; // 2 s
9 return 0;
10 }
$o = 1 \Rightarrow s \leq i$
$o = 2 \Rightarrow s > i$


input, 4
$1 \leq s \leq 8$
s?
1 private s = getBufferSize();
2
3
4 public int compare(int i){
5 if(s <= i)
6 log.write("too large"); // 1 s
7 else
8 some computation; // 2 s
9 return 0;
10 }
$o = 1 \Rightarrow s \leq i$
$o = 2 \Rightarrow s > i$


input, 4
$1 \leq s \leq 8$
s?
1 private s = getBufferSize();
2
3
4 public int compare(int i){
5 if(s <= i)
6 log.write("too large"); // 1 s
7 else
8 some computation; // 2 s
9 return 0;
10 }
$o = 1 \Rightarrow s \leq 4$
$o = 2 \Rightarrow s > 4$



Attacker can binary search on s using i and o.

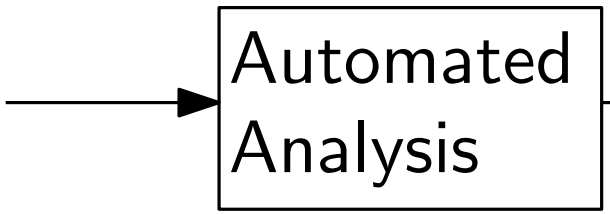

C
Java
python
OpenSSL
OAUTH
OpenID
Google
twitter
XBOX

Is my code vulnerable to side-channel attacks?

```
Boolean compare(pw, input){
  for(int i = 0; i < pw.length(), i++)
    if(pw[i] != input[i])
      return false;
  return true;
}
```

Is my code vulnerable to side-channel attacks?

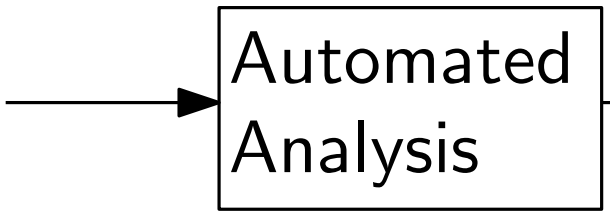
```
Boolean compare(pw, input){
  for(int i = 0; i < pw.length(), i++)
    if(pw[i] != input[i])
      return false;
  return true;
}
```



Synthesized Attack

Is my code vulnerable to side-channel attacks?

```
Boolean compare(pw, input){
  for(int i = 0; i < pw.length(), i++)
    if(pw[i] != input[i])
      return false;
  return true;
}
```

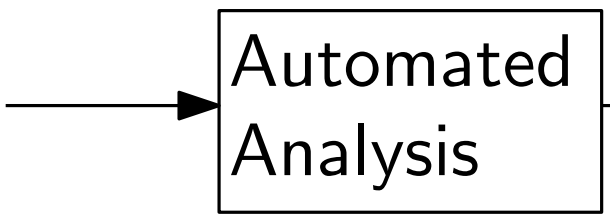


Synthesized Attack

1. Static Offline Analysis

Is my code vulnerable to side-channel attacks?

```
Boolean compare(pw, input){
  for(int i = 0; i < pw.length(), i++)
    if(pw[i] != input[i])
      return false;
  return true;
}
```

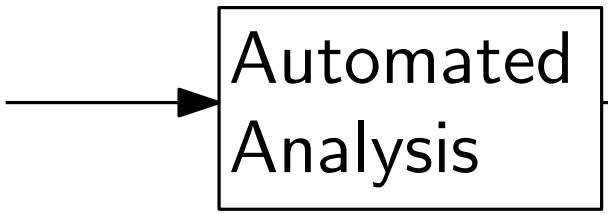


Synthesized Attack

1. Static Offline Analysis
2. Static Offline + Dynamic + Online Analysis

Is my code vulnerable to side-channel attacks?

```
Boolean compare(pw, input){
  for(int i = 0; i < pw.length(), i++)
    if(pw[i] != input[i])
      return false;
  return true;
}
```



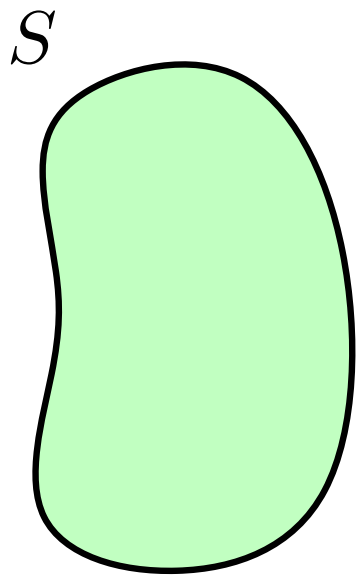
Synthesized Attack

1. Static Offline Analysis
2. Static Offline + Dynamic + Online Analysis

Side Channels and Searching: Entropy

Side Channels and Searching: Entropy

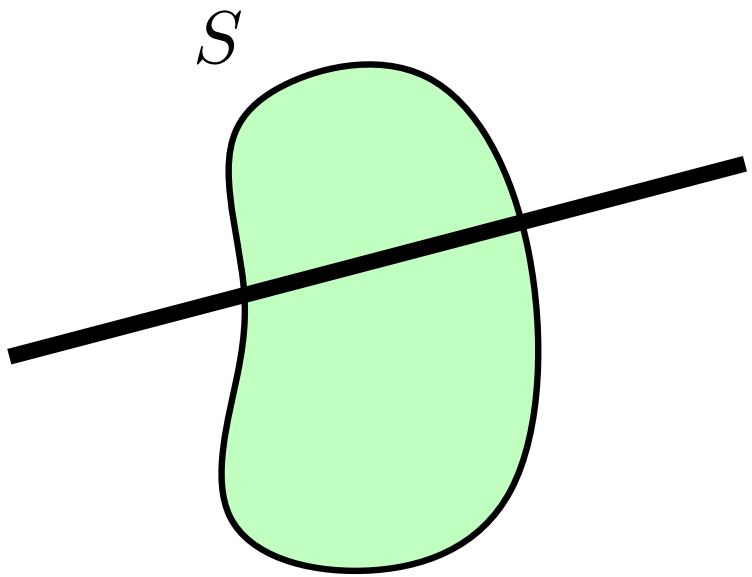
$i \in I$



secret $s \in S$

Side Channels and Searching: Entropy

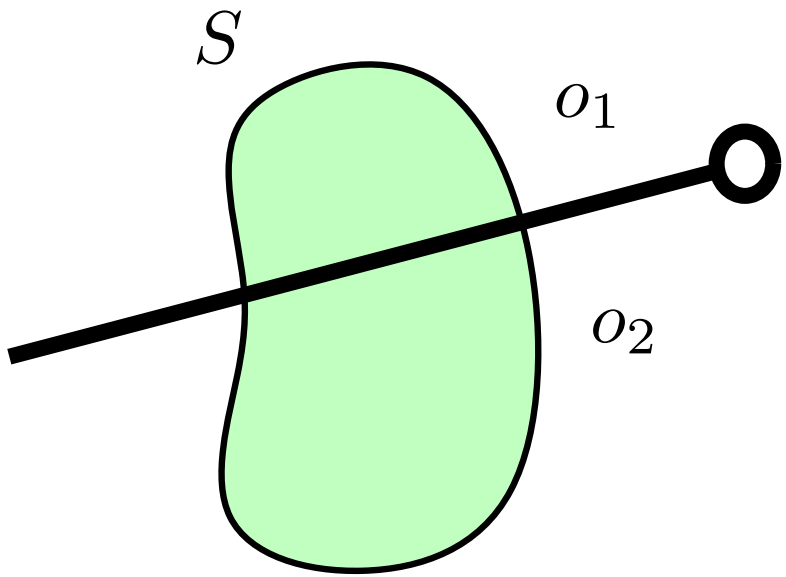
$i \in I$



secret $s \in S$

Side Channels and Searching: Entropy

$i \in I$

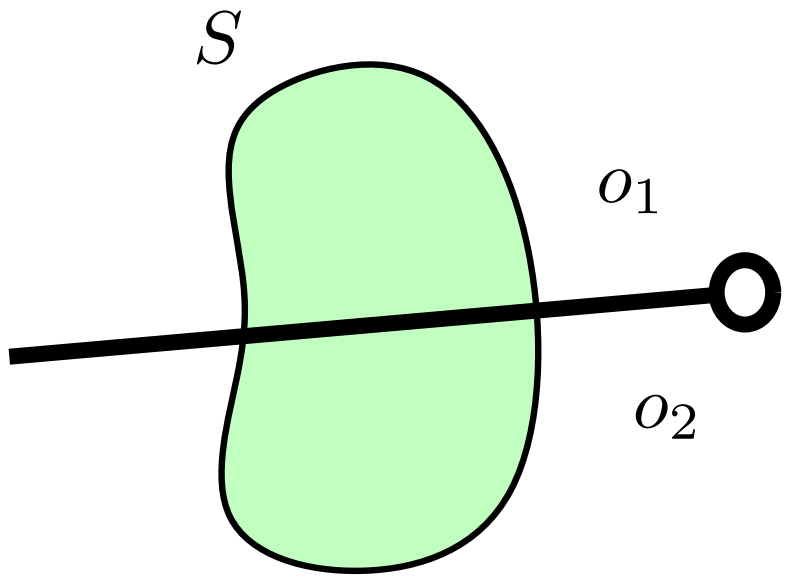


secret $s \in S$

Side Channels and Searching: Entropy

$i \in I$

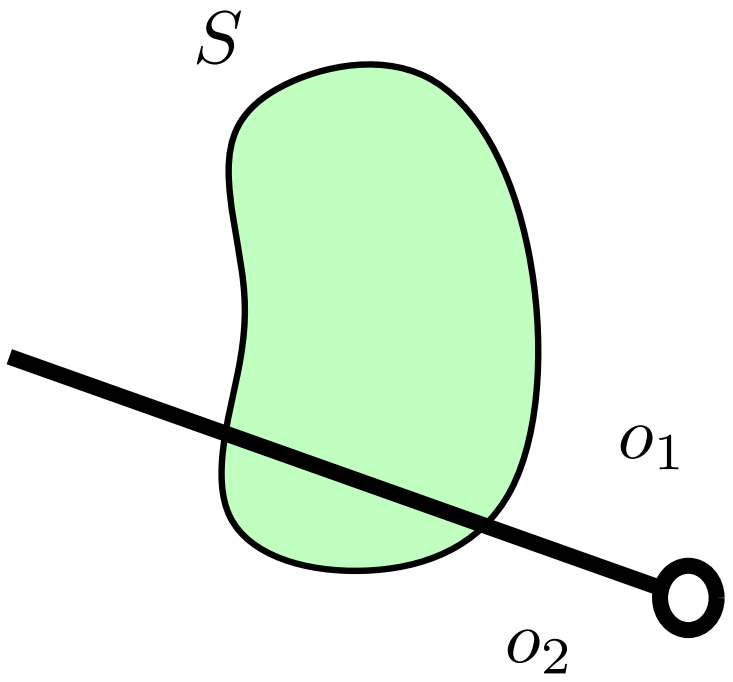
secret $s \in S$



Side Channels and Searching: Entropy

$i \in I$

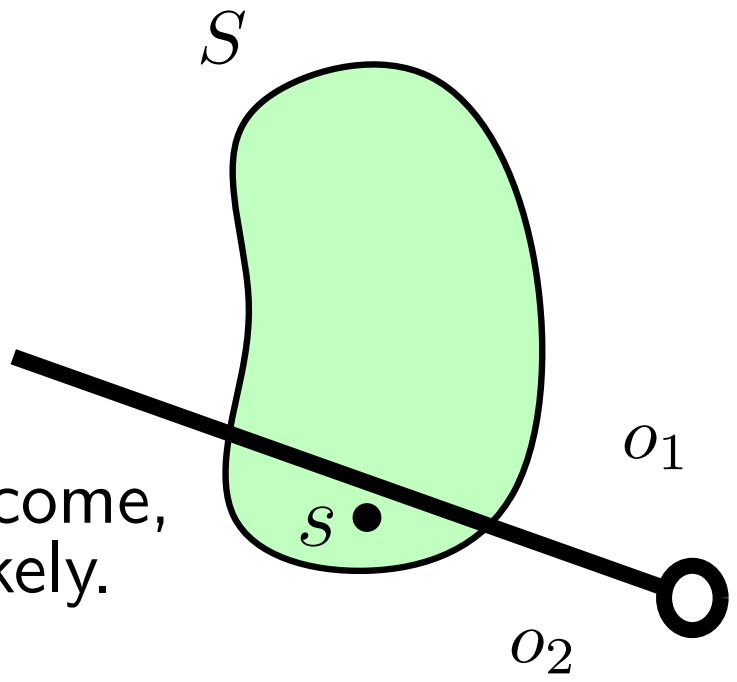
secret $s \in S$



Side Channels and Searching: Entropy

$i \in I$

secret $s \in S$

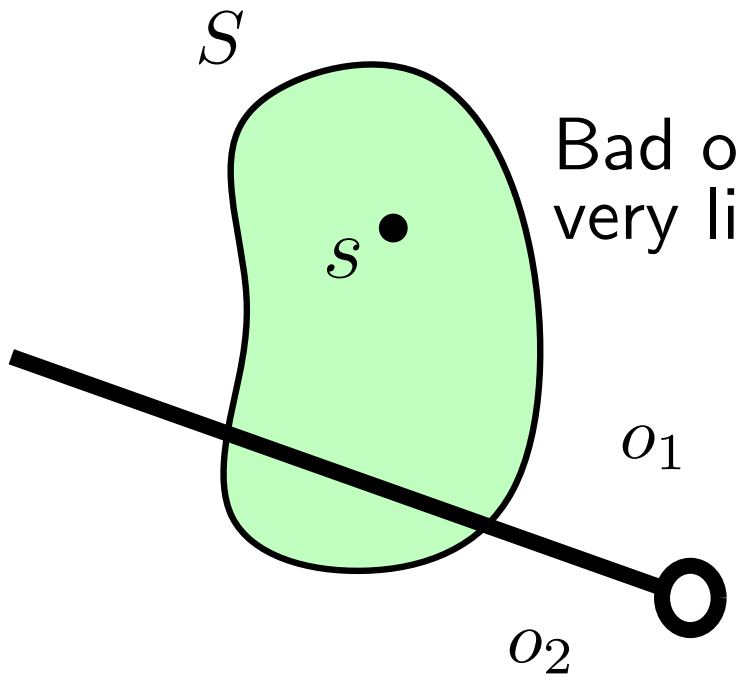


Good outcome,
very unlikely.

Side Channels and Searching: Entropy

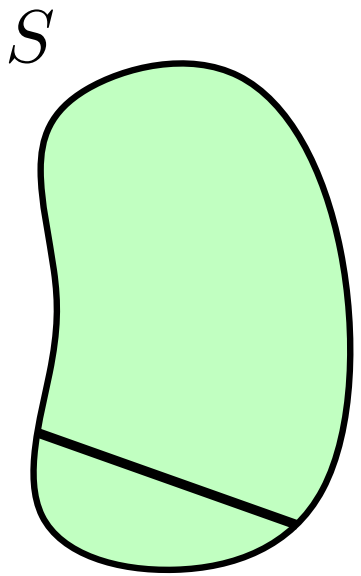
$i \in I$

secret $s \in S$



Side Channels and Searching: Entropy

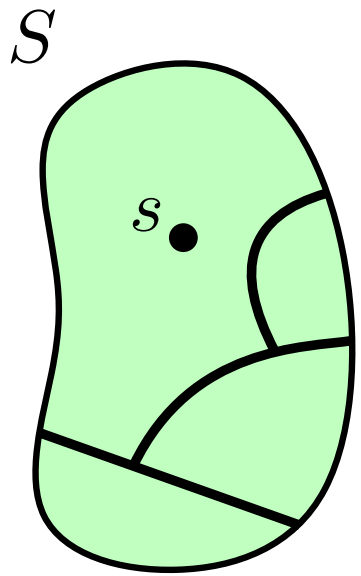
$i \in I$



secret $s \in S$

Side Channels and Searching: Entropy

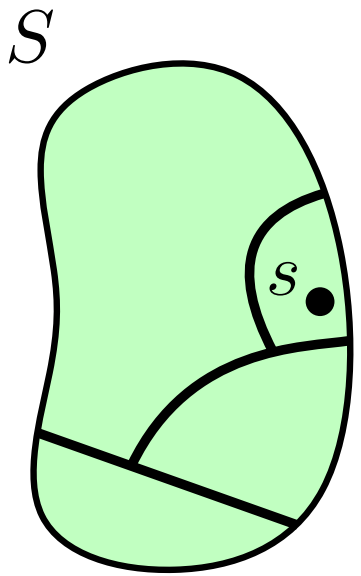
$i \in I$



secret $s \in S$

Side Channels and Searching: Entropy

$i \in I$

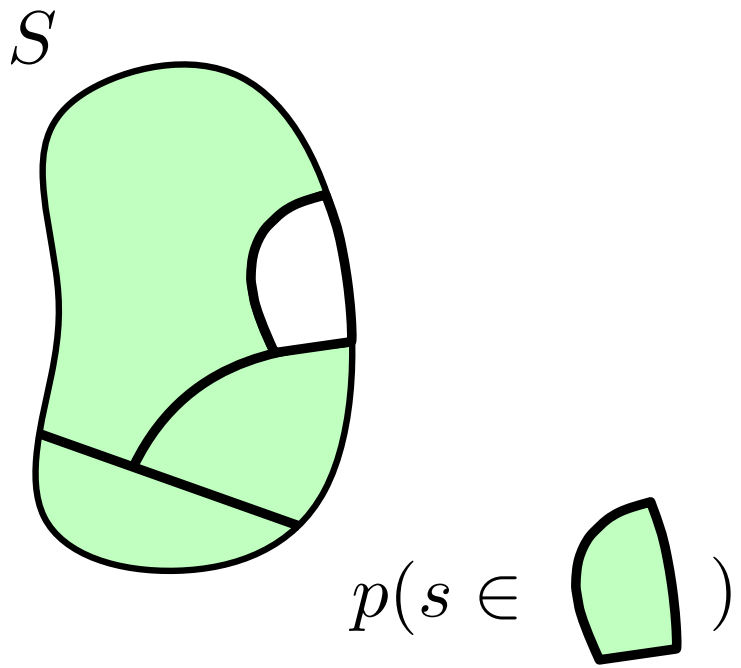


secret $s \in S$

Side Channels and Searching: Entropy

$i \in I$

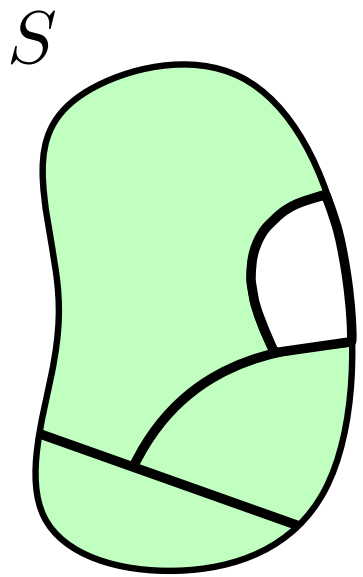
secret $s \in S$



Side Channels and Searching: Entropy

$i \in I$

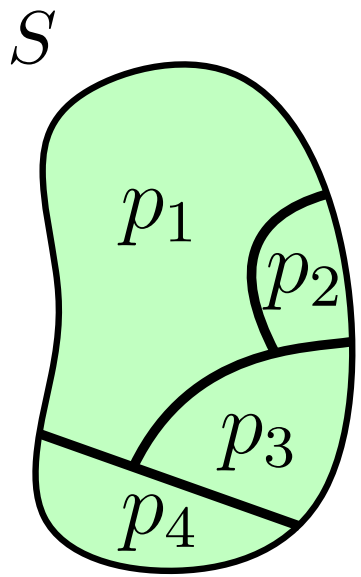
secret $s \in S$



$$p(s \in \quad) = \frac{|\quad|}{|\quad|}$$

Side Channels and Searching: Entropy

$i \in I$

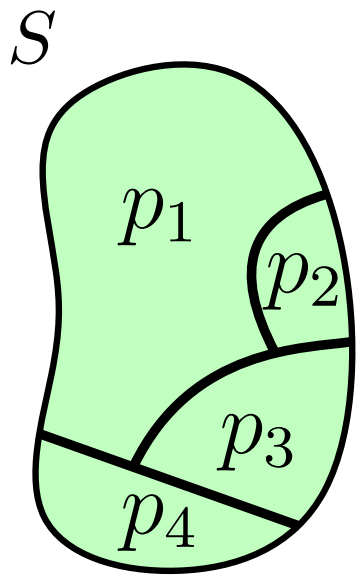


secret $s \in S$

Side Channels and Searching: Entropy

$i \in I$

secret $s \in S$

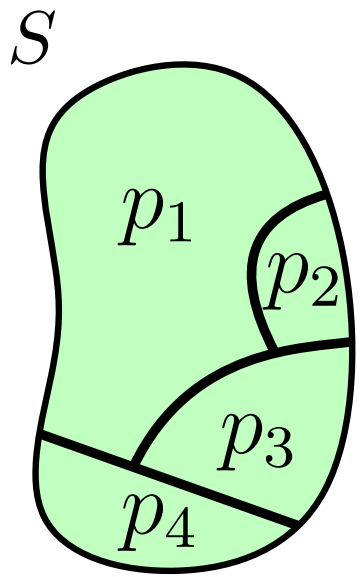


Quantify expected information gain measured in bits.

Side Channels and Searching: Entropy

$i \in I$

secret $s \in S$



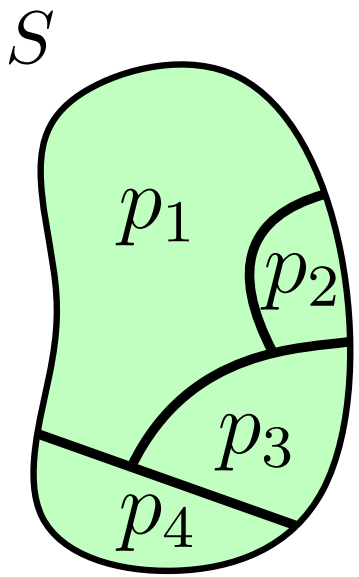
Quantify expected information gain measured in bits.

$$\frac{1}{p_j}$$

Side Channels and Searching: Entropy

$i \in I$

secret $s \in S$



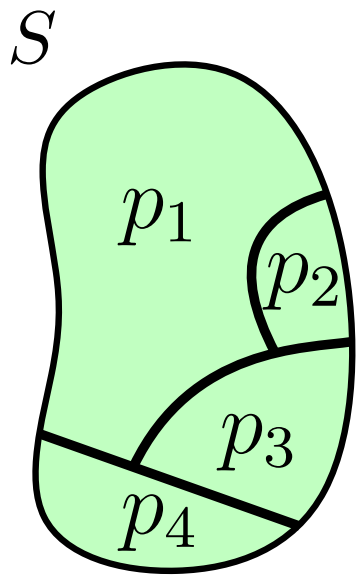
Quantify expected information gain measured in bits.

$$\log_2 \frac{1}{p_j}$$

Side Channels and Searching: Entropy

$i \in I$

secret $s \in S$



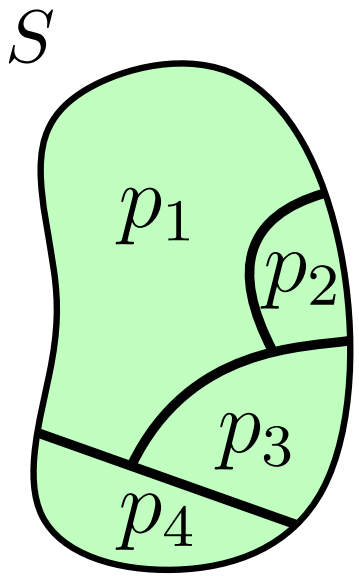
Quantify expected information gain measured in bits.

$$\sum_{j=1}^{n} p_j \log_2 \frac{1}{p_j}$$

Side Channels and Searching: Entropy

$i \in I$

secret $s \in S$



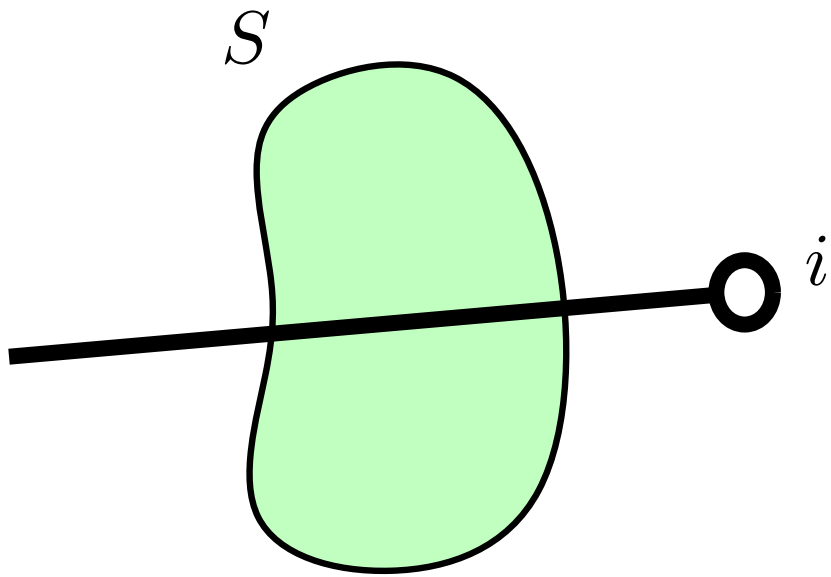
Quantify expected information gain measured in bits.

$$\mathcal{H} = \sum_{j=1}^{n} p_j \log_2 \frac{1}{p_j}$$

Side Channels and Searching: Entropy

$i \in I$

secret $s \in S$



Quantify expected information gain measured in bits.

$$\mathcal{H}(i) = \sum_{j=1}^{n} p_j \log_2 \frac{1}{p_j}$$

$\max \mathcal{H}(i) \Rightarrow$ Binary Search

$o = 1 \Rightarrow s \leq i$ $\qquad\qquad$ $o = 2 \Rightarrow s > i$

$) \wedge H_{1,0} = L_{1,0} \wedge H_{0,0} = L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 9$

$) \wedge H_{1,0} \neq L_{1,0} \wedge H_{0,0} = L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 9$

$) \wedge H_{1,0} \neq L_{1,0} \wedge H_{0,0} = L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 9$

$) \wedge H_{1,0} = L_{1,0} \wedge H_{0,0} \neq L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 9$

$) \wedge H_{1,0} = L_{1,0} \wedge H_{0,0} \neq L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 9$

$) \wedge H_{1,0} \neq L_{1,0} \wedge H_{0,0} \neq L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 9$

$) \wedge H_{1,0} \neq L_{1,0} \wedge H_{0,0} \neq L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 9$

$\wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge 0 < H_{4,0} \wedge H_{4,0}$

$) \wedge H_{0,0} = L_{0,0} \wedge 2 \geq H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge$

$) \wedge H_{0,0} = L_{0,0} \wedge 2 \geq H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge$

$) \wedge H_{0,0} \neq L_{0,0} \wedge 2 \geq H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge$

$) \wedge H_{0,0} \neq L_{0,0} \wedge 2 \geq H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge$

$\wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge 0 < H_{4,0} \wedge H_{4,0} \geq 0 \wedge H_{4,0} \leq 3 \wedge L_{2,0} \leq 122 \wedge L_{2,0} \geq$

$) \wedge 1 \geq H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge 0 < H_{4,0} \wedge H_{4,0} \geq 0 \wedge H_{4,0} \leq 3 \wedge L_{2,0} \leq 122 \wedge L_{2,0}$

$) \wedge 1 \geq H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge 0 < H_{4,0} \wedge H_{4,0} \geq 0 \wedge H_{4,0} \leq 3 \wedge L_{2,0} \leq 122 \wedge L_{2,0}$

$\wedge 0 < H_{4,0} \wedge H_{4,0} \geq 0 \wedge H_{4,0} \leq 3 \wedge L_{2,0} \leq 122 \wedge L_{2,0} \geq 97 \wedge 2 < L_{4,0} \wedge L_{1,0} \leq 122 \wedge L_{1,0} \geq$

$H_{4,0} \geq 0 \wedge H_{4,0} \leq 3 \wedge L_{2,0} \leq 122 \wedge L_{2,0} \geq 97 \wedge 2 < L_{4,0} \wedge L_{1,0} \leq 122 \wedge L_{1,0} \geq 97 \wedge 1 < L_4$

$L_{2,0} \leq 122 \wedge L_{2,0} \geq 97 \wedge 2 < L_{4,0} \wedge L_{1,0} \leq 122 \wedge L_{1,0} \geq 97 \wedge 1 < L_{4,0} \wedge L_{0,0} \leq 122 \wedge L_{0,0} \geq$

$2 < L_{4,0} \wedge L_{1,0} \leq 122 \wedge L_{1,0} \geq 97 \wedge 1 < L_{4,0} \wedge L_{0,0} \leq 122 \wedge L_{0,0} \geq 97 \wedge 0 < L_{4,0} \wedge L_{4,0} \geq$

$H_{4,0} \leq 3 \wedge 2 \geq L_{4,0} \wedge L_{1,0} \leq 122 \wedge L_{1,0} \geq 97 \wedge 1 < L_{4,0} \wedge L_{0,0} \leq 122 \wedge L_{0,0} \geq 97 \wedge 0 < L_{4,}$

$\wedge H_{0,0} \geq 97 \wedge 0 < H_{4,0} \wedge H_{4,0} \geq 0 \wedge H_{4,0} \leq 3 \wedge 2 \geq L_{4,0} \wedge L_{1,0} \leq 122 \wedge L_{1,0} \geq 97 \wedge 1 < L_{4}$

$\wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge 0 < H_{4,0} \wedge H_{4,0} \geq 0 \wedge H_{4,0} \leq 3 \wedge 2 \geq L$

$\wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge H_{0,0} \leq 122 \wedge H_{0,0} \geq 97 \wedge 0$

$) \wedge H_{0,0} = L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge$

$) \wedge H_{0,0} = L_{0,0} \wedge H_{2,0} \leq 122 \wedge H_{2,0} \geq 97 \wedge 2 < H_{4,0} \wedge H_{1,0} \leq 122 \wedge H_{1,0} \geq 97 \wedge 1 < H_{4,0} \wedge$

Password Checker Constraints

$\max \mathcal{H}(i) \Rightarrow$ Binary Search

$o = 1 \Rightarrow s \leq i$ $\qquad\qquad$ $o = 2 \Rightarrow s > i$

$\max \mathcal{H}(i) \Rightarrow$ Binary Search

$o = 1 \Rightarrow s \leq i$ $\qquad\qquad$ $o = 2 \Rightarrow s > i$

$\max \mathcal{H}(i) \Rightarrow$ Optimal Search

any program constraints

$\max \mathcal{H}(i) \Rightarrow$ Binary Search

$o = 1 \Rightarrow s \leq i$ $\qquad$ $o = 2 \Rightarrow s > i$

$\max \mathcal{H}(i) \Rightarrow$ Optimal Search

any program constraints

$\longrightarrow$??? $\longrightarrow \mathcal{H}(i)$

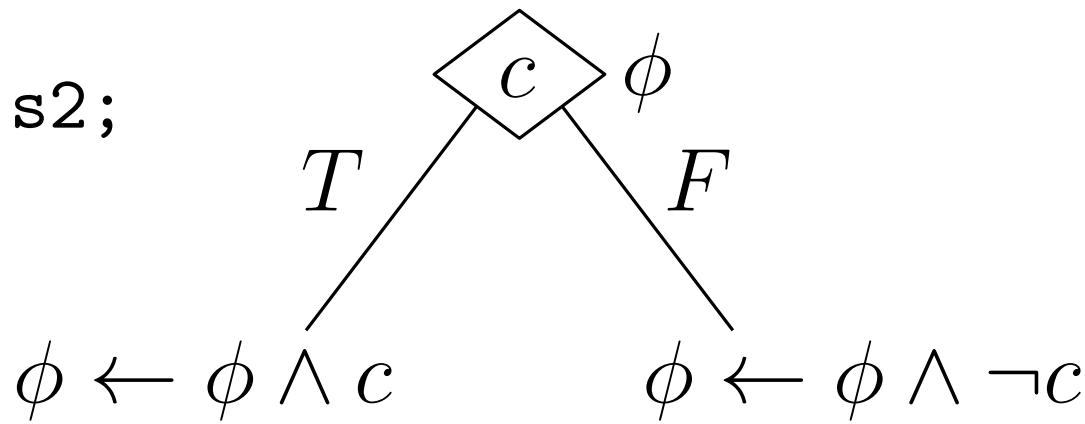
Symbolic Execution

Execute program on **symbolic** rather than concrete inputs.

Maintain **path constraints**, PCs, ϕ_j over symbolic inputs.

For branch instructions:

```
if(c) then s1; else s2;
```



$\phi_j(s, i)$ characterizes the relation between s, i, and o_j

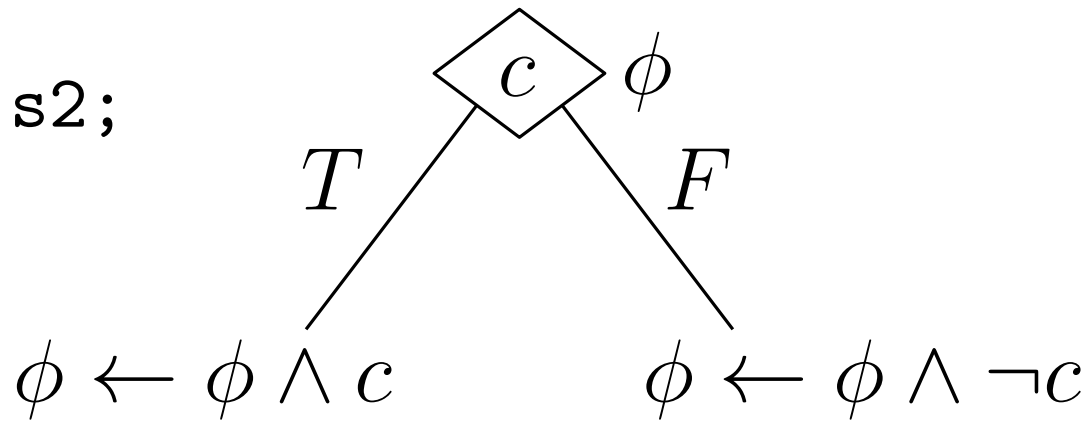
Symbolic Execution

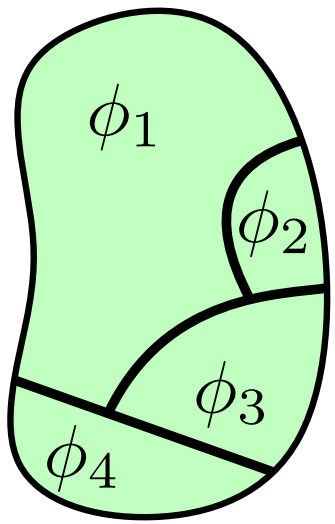
Execute program on **symbolic** rather than concrete inputs.

Maintain **path constraints**, PCs, ϕ_j over symbolic inputs.

For branch instructions:

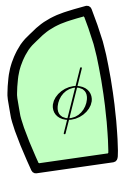
```
if(c) then s1; else s2;
```





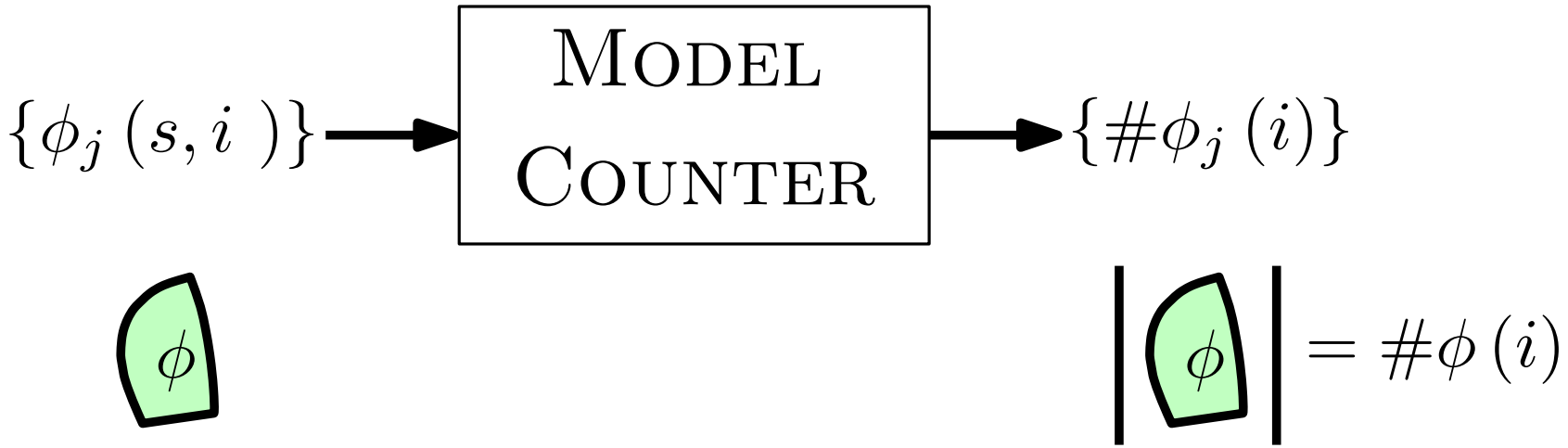
$\phi_j(s, i)$ characterizes the relation between s, i, and o_j

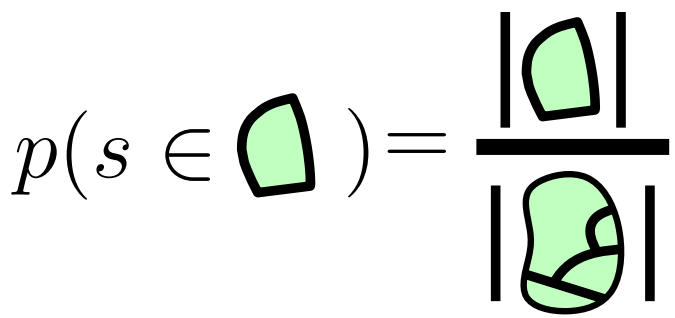
$$p(s \in \text{  }) = \frac{|\text{  }|}{|\text{  }|}$$

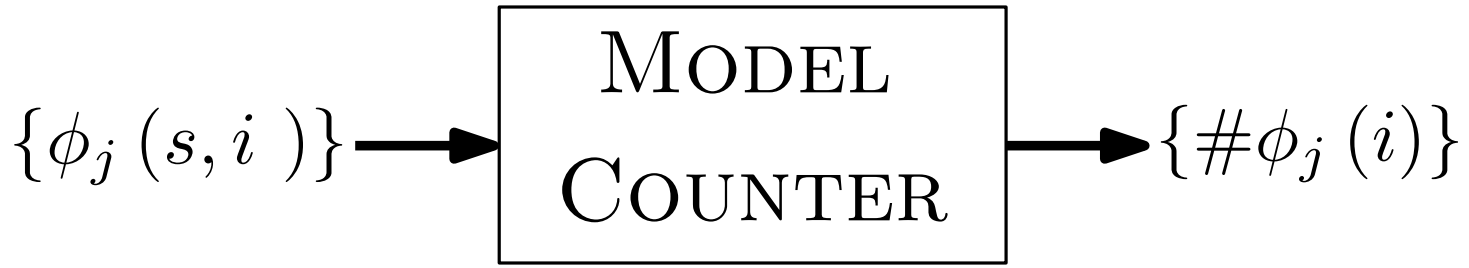


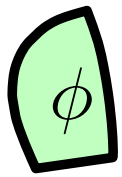
$$\left| \phi \right| = \#\phi\,(i)$$

$$p(s \in \;) = \frac{|\;|}{|\;|}$$


$\{\phi_j\,(s,i\,)\}$
Model Counter
$\{\#\phi_j\,(i)\}$
ϕ
$\left|\phi\right| = \#\phi\,(i)$


$p(s \in$)$=$

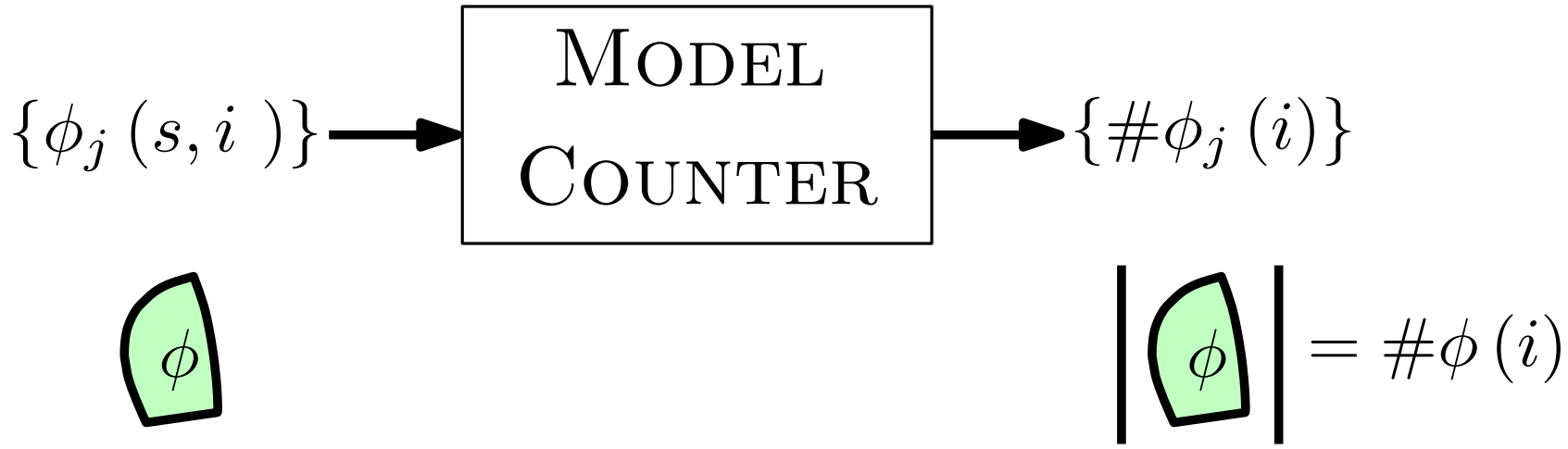




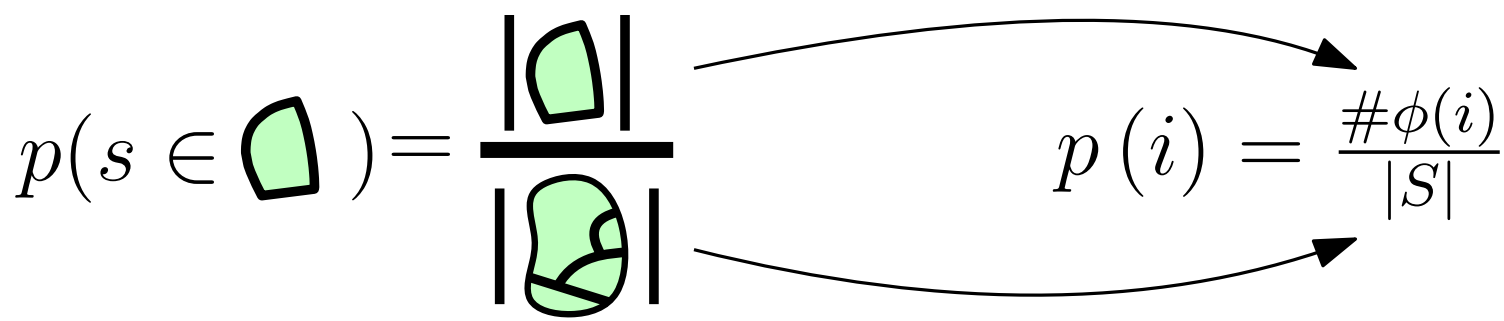
$$\left|\;\right| = \#\phi\,(i)$$

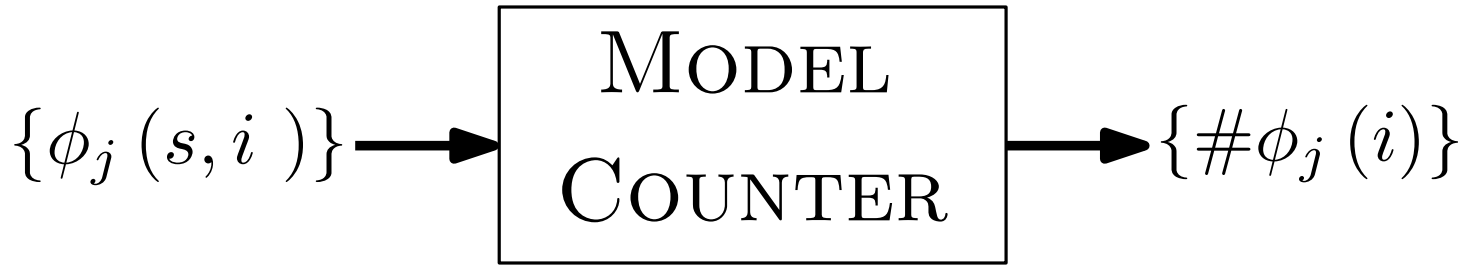
$\#\phi\,(i)$ is the number of satisfying solutions (models) for $\phi\,(s,i)$ for a given i.

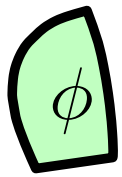
$$p(s \in \;) = \frac{\left|\;\right|}{\left|\;\right|}$$



$\#\phi\,(i)$ is the number of satisfying solutions (models) for $\phi\,(s,i)$ for a given i.





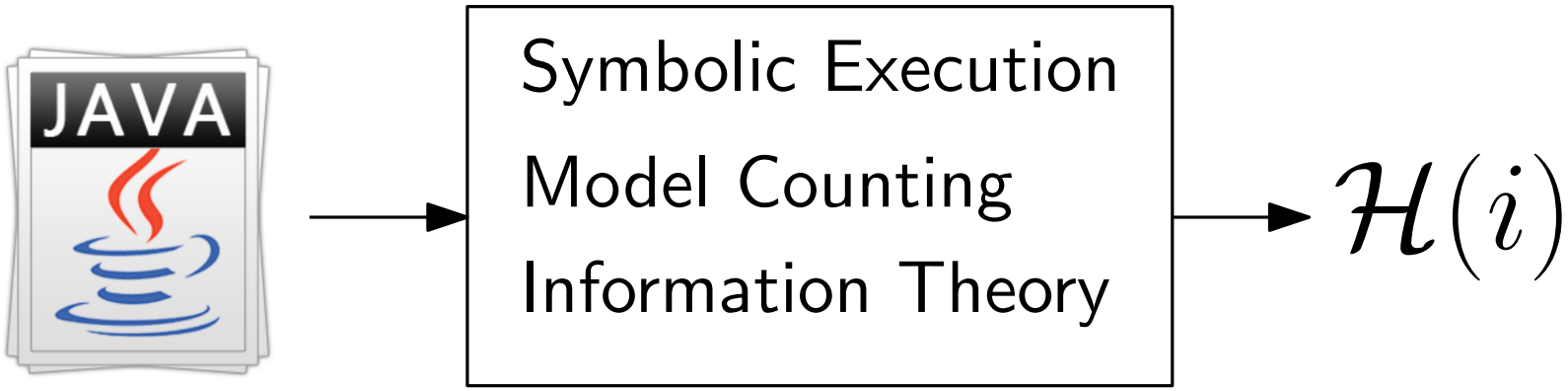


$$\left|\phi\right| = \#\phi\,(i)$$

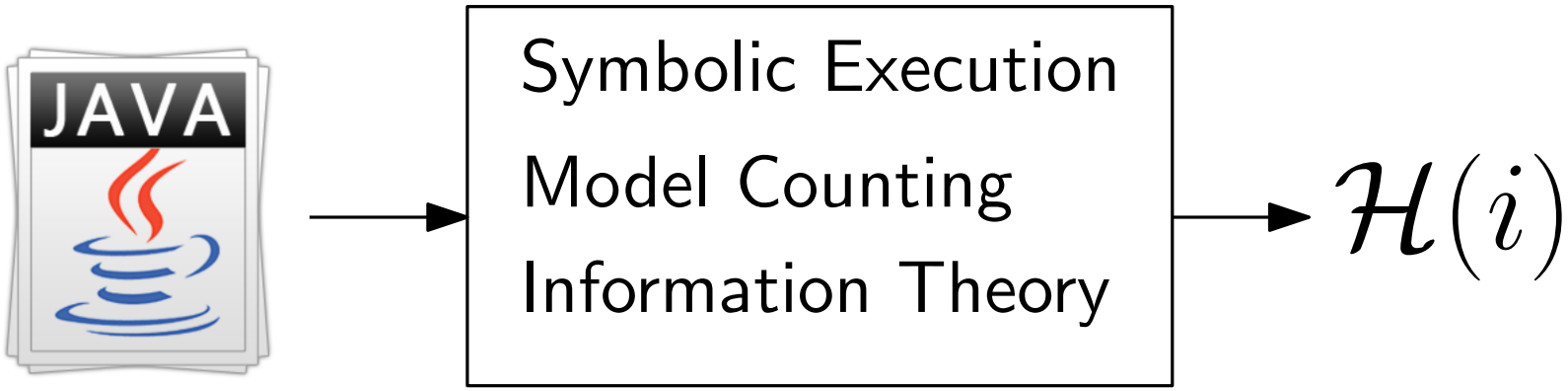
$\#\phi\,(i)$ is the number of satisfying solutions (models) for $\phi\,(s,i)$ for a given i.

$$p(s \in \phi\,) = \frac{|\phi|}{|\Omega|} \qquad p\,(i) = \frac{\#\phi(i)}{|S|}$$

$$\mathcal{H}(i) = \sum_{j=1}^{n} p_j(i) \log_2 \frac{1}{p_j(i)}$$



$\mathcal{H}(i)$ is a symbolic expression that measures the expected information an attacker gains when making input i.



$\mathcal{H}(i)$ is a symbolic expression that measures the expected information an attacker gains when making input i.

Maximizing $\mathcal{H}(i)$ gives an optimal side-channel attack. [IEEE Computer Security Foundations 2017]

1. Fully Static Offline Approach

Assumes an ideal observation model (i.e. instruction counts).

Does not account for actual runtime behavior.

1. Fully Static Offline Approach

Assumes an ideal observation model (i.e. instruction counts).

Does not account for actual runtime behavior.

2. Static / Dynamic + Offline / Online Approach

Automatically, dynamically estimates runtime observations.

Uses Bayesian inference and weighted model counting to account for noise.

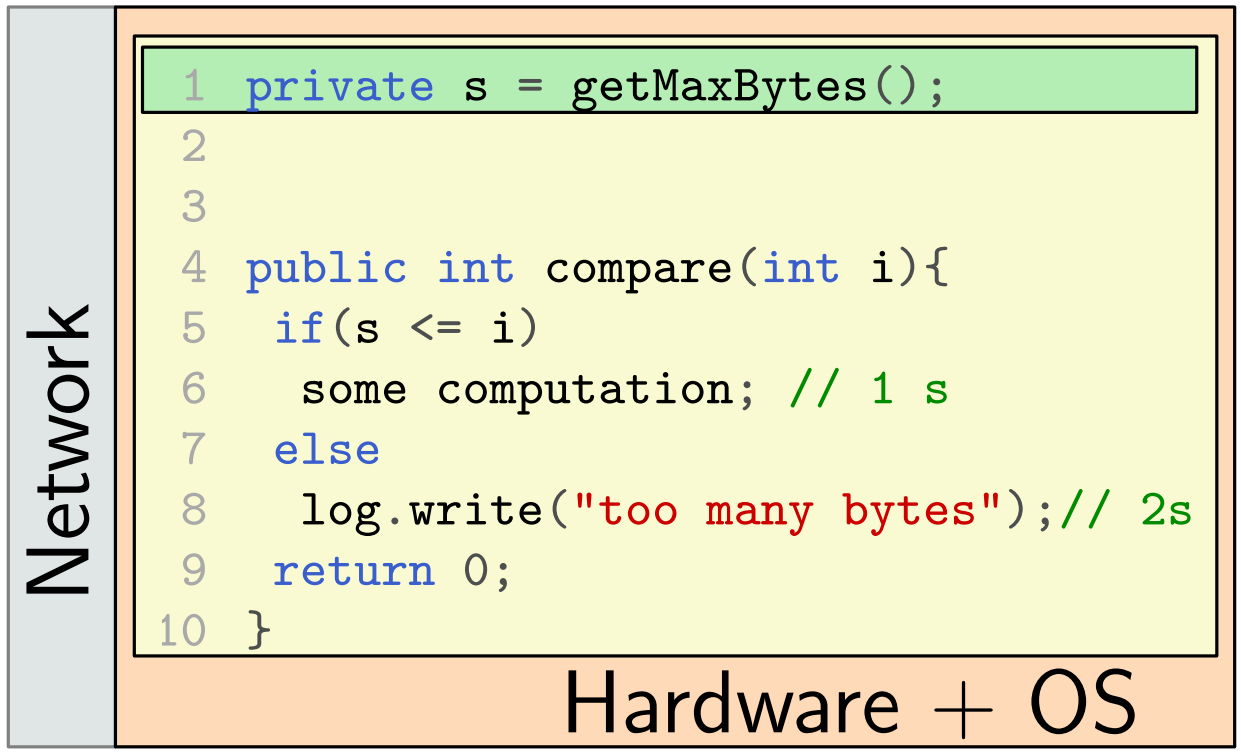
Side-Channel Attack Synthesis Under Noisy Conditions

[IEEE European Security & Privacy 2018]

```
1  private s = getMaxBytes();
2
3
4  public int compare(int i){
5   if(s <= i)
6    some computation; // 1 s
7   else
8    log.write("too many bytes");// 2s
9   return 0;
10 }
```

```
1  private s = getMaxBytes();
2
3
4  public int compare(int i){
5   if(s <= i)
6    some computation; // 1 s
7   else
8    log.write("too many bytes");// 2s
9   return 0;
10 }
```

Hardware + OS


Network
1 private s = getMaxBytes();
2
3
4 public int compare(int i){
5 if(s <= i)
6 some computation; // 1 s
7 else
8 log.write("too many bytes");// 2s
9 return 0;
10 }
Hardware + OS

s?

Network

```
1  private s = getMaxBytes();
2
3
4  public int compare(int i){
5   if(s <= i)
6    some computation; // 1 s
7   else
8    log.write("too many bytes");// 2s
9   return 0;
10 }
```

Hardware + OS

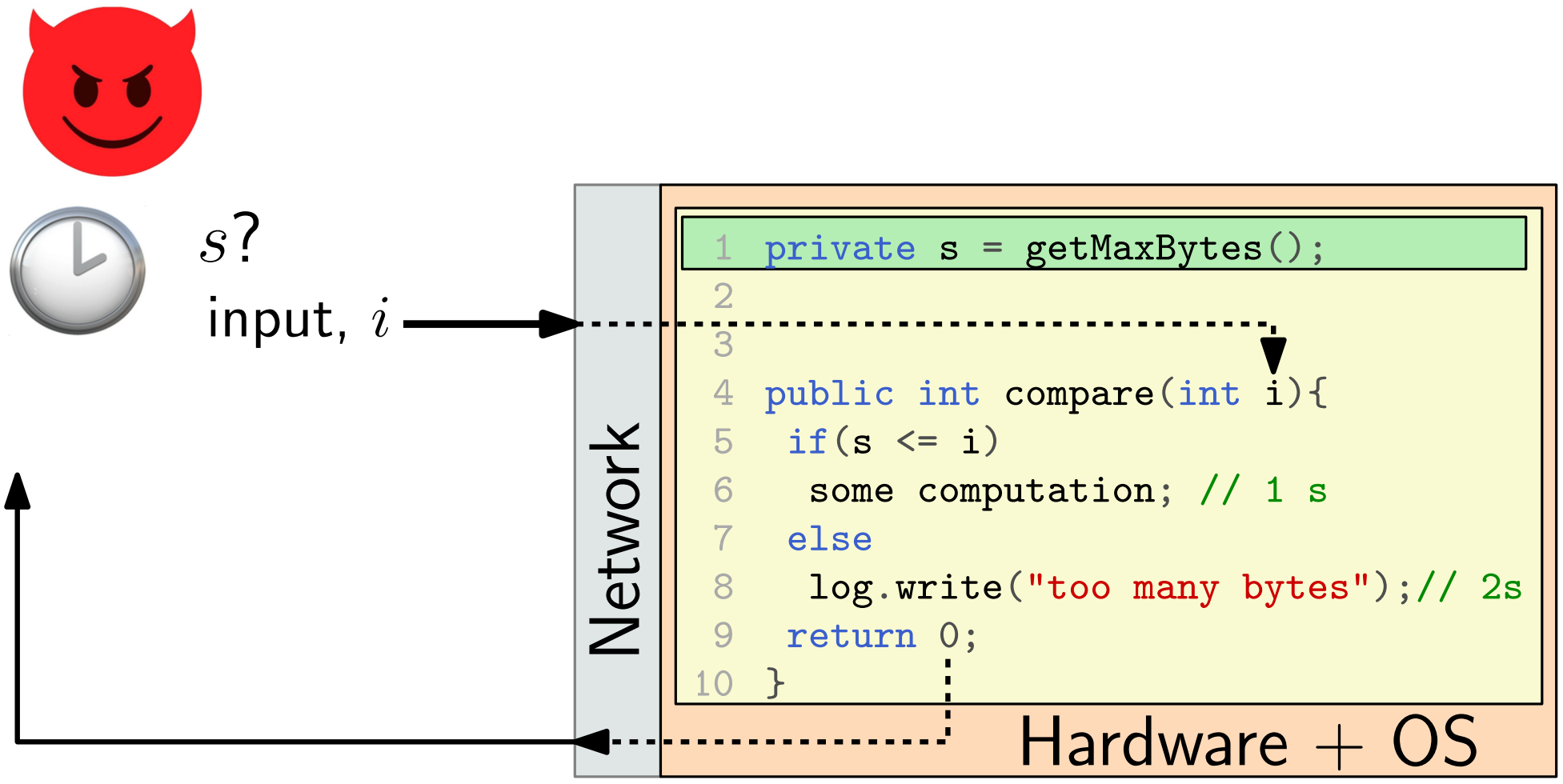
s?

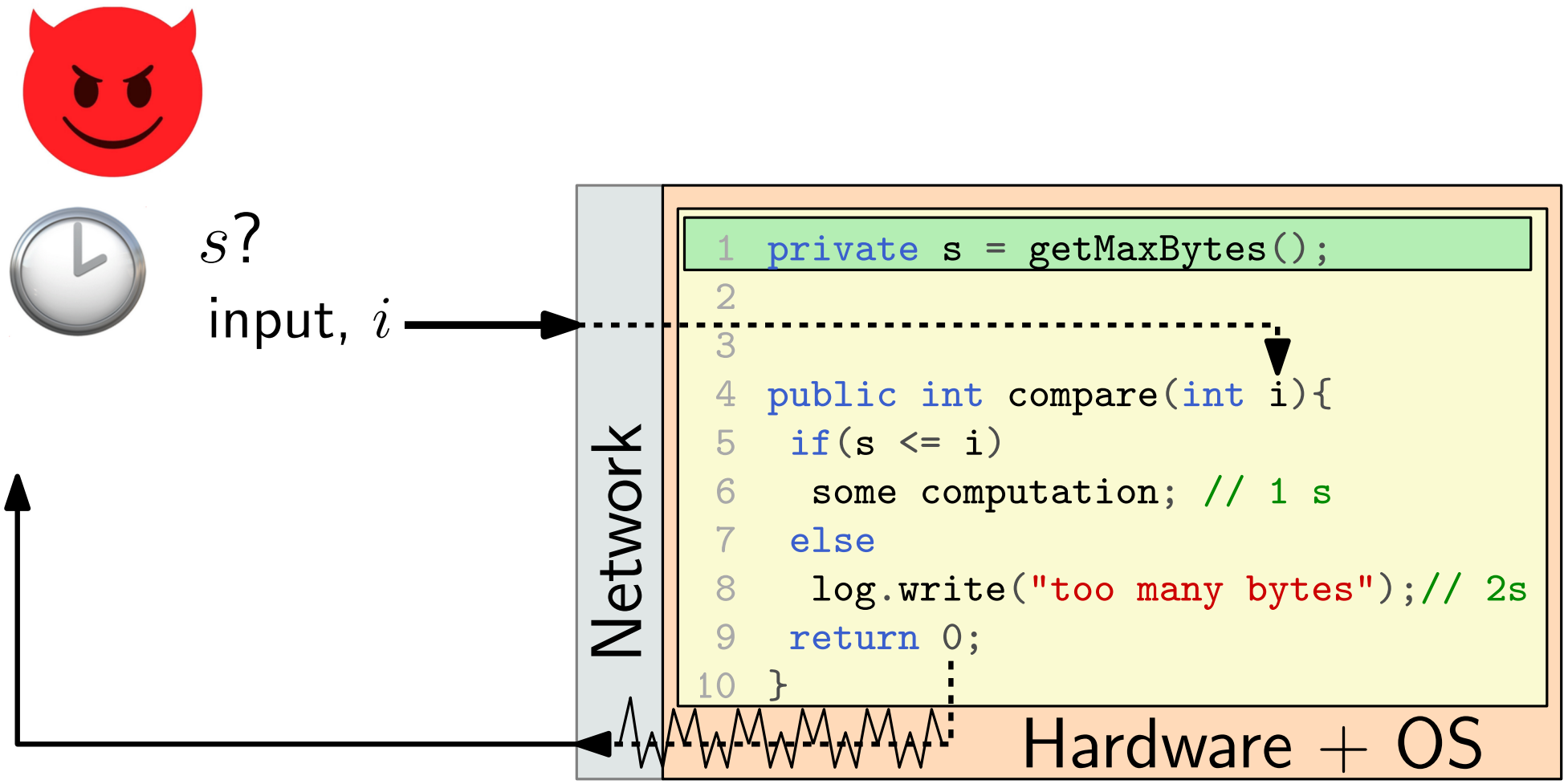
input, i

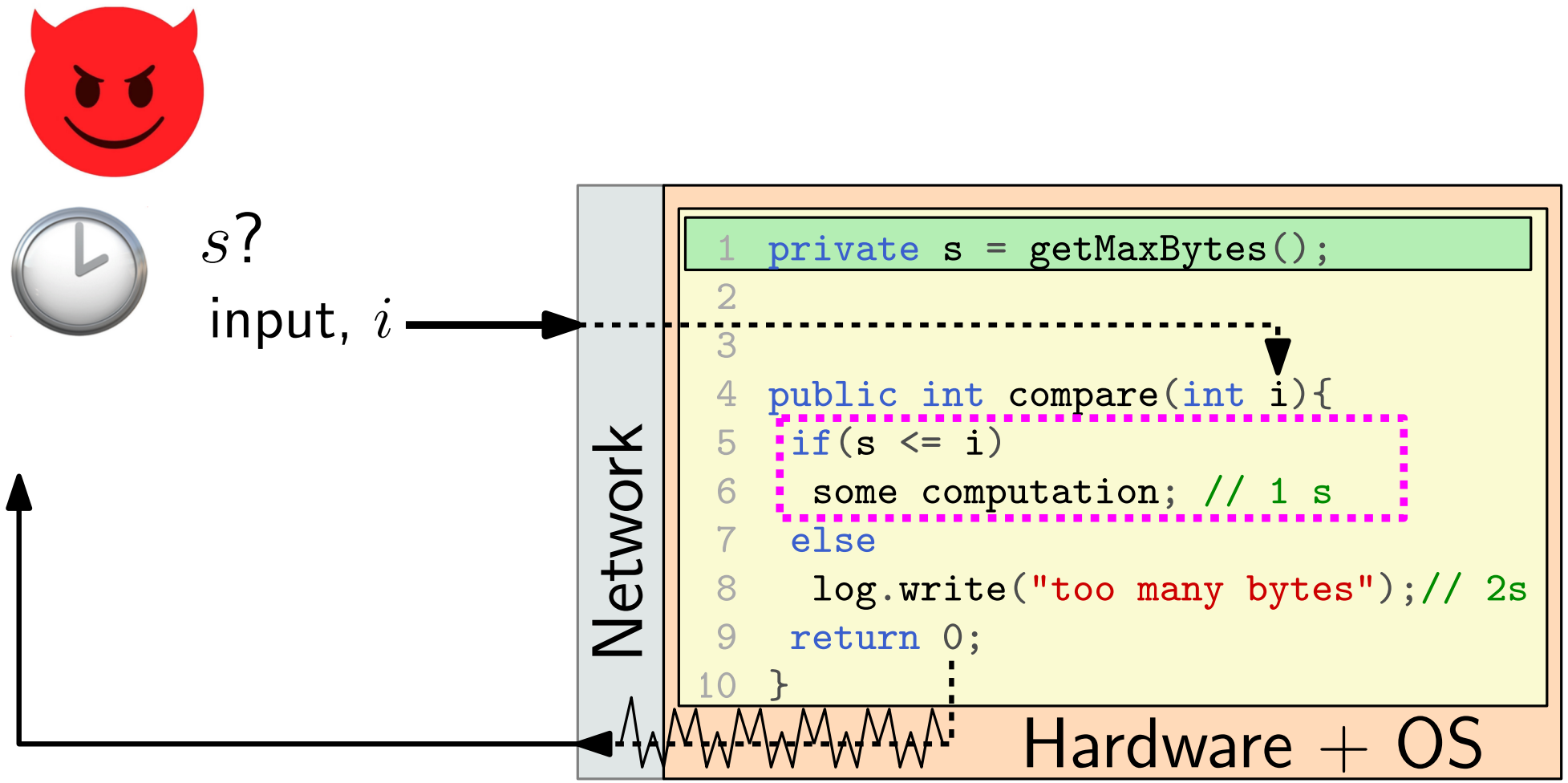
Network

```
1  private s = getMaxBytes();
2
3
4  public int compare(int i){
5   if(s <= i)
6    some computation; // 1 s
7   else
8    log.write("too many bytes");// 2s
9   return 0;
10 }
```

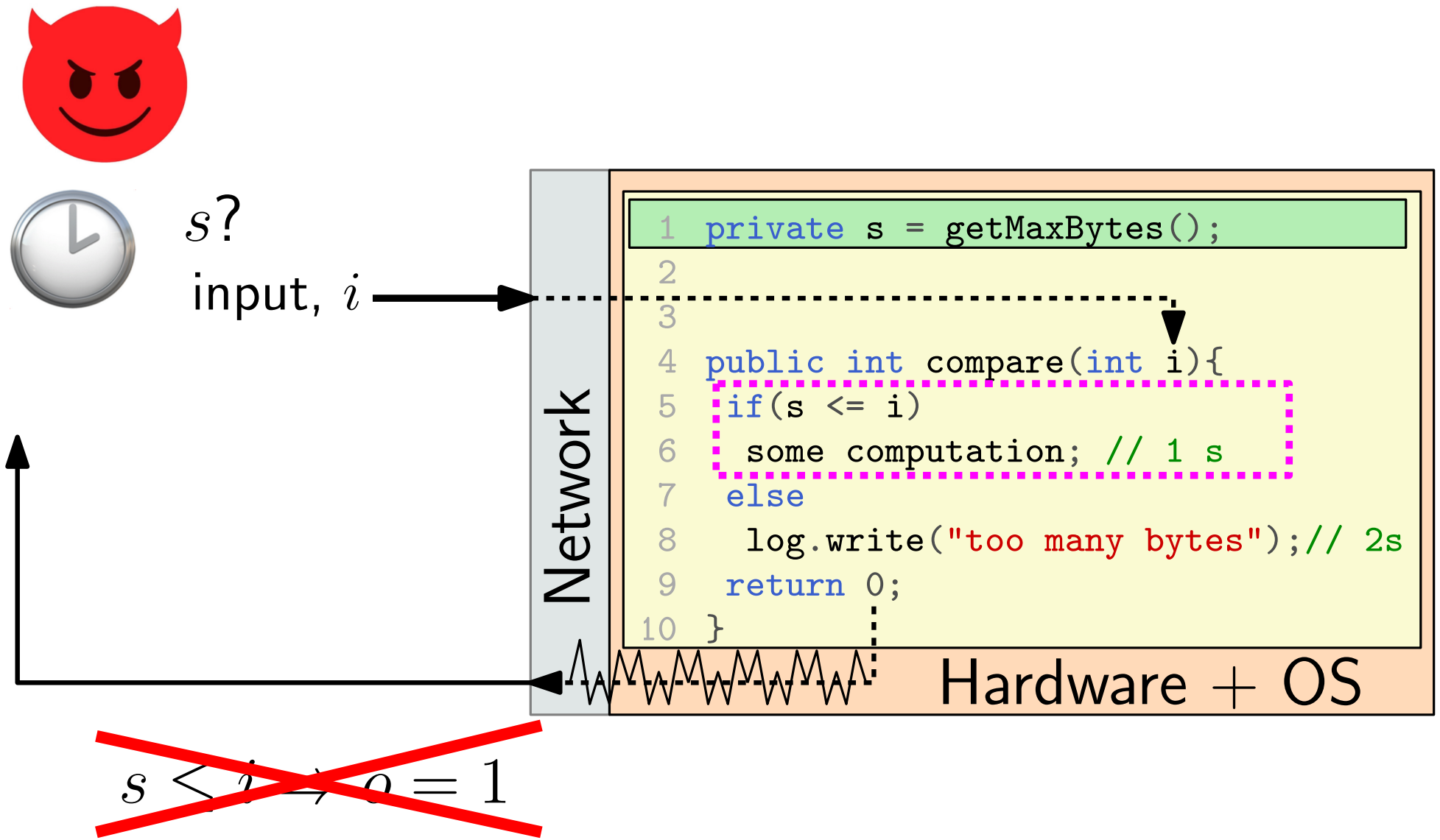
Hardware + OS

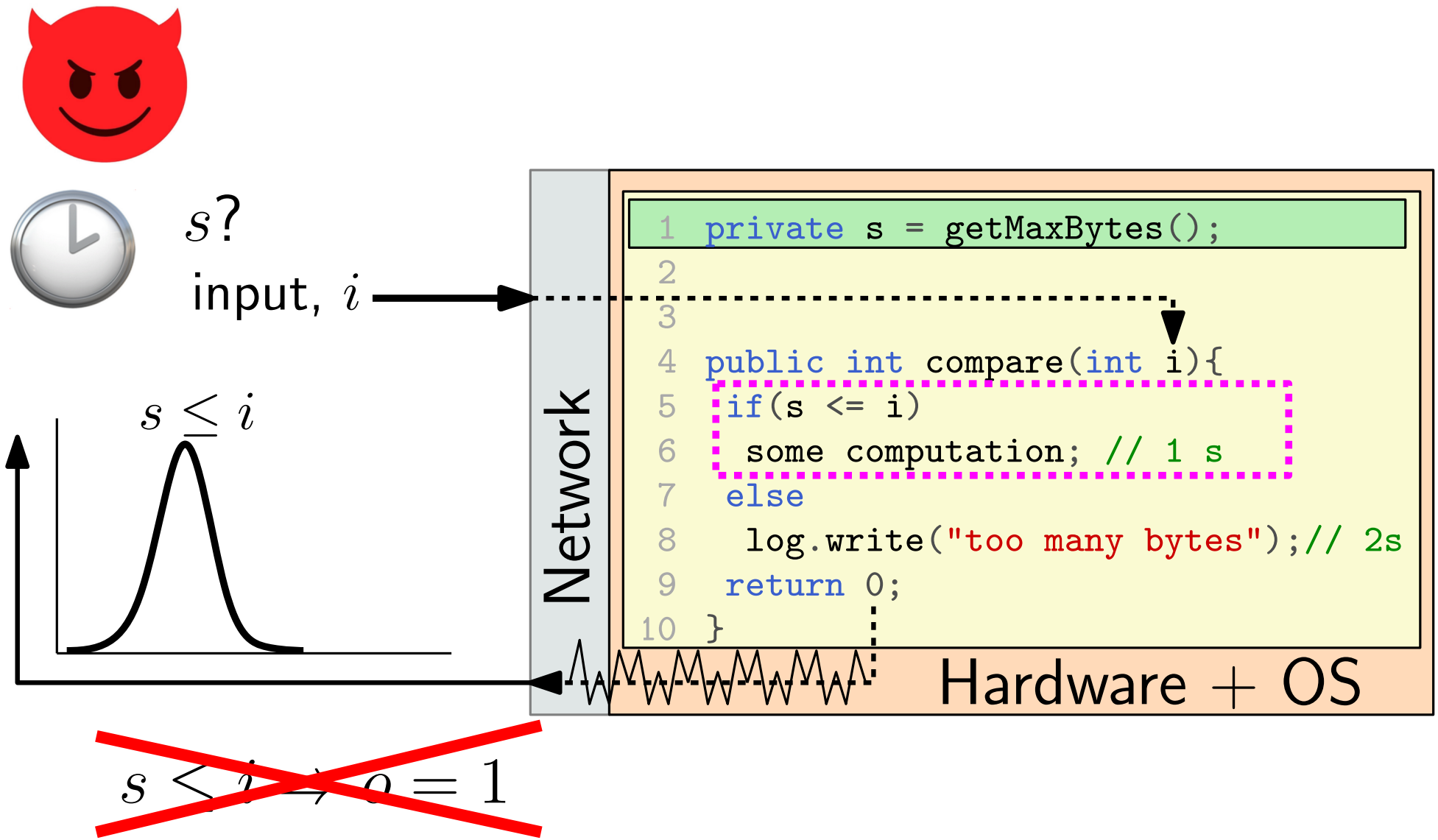

s?
input, i
Network
1 private s = getMaxBytes();
2
3
4 public int compare(int i){
5 if(s <= i)
6 some computation; // 1 s
7 else
8 log.write("too many bytes");// 2s
9 return 0;
10 }
Hardware + OS

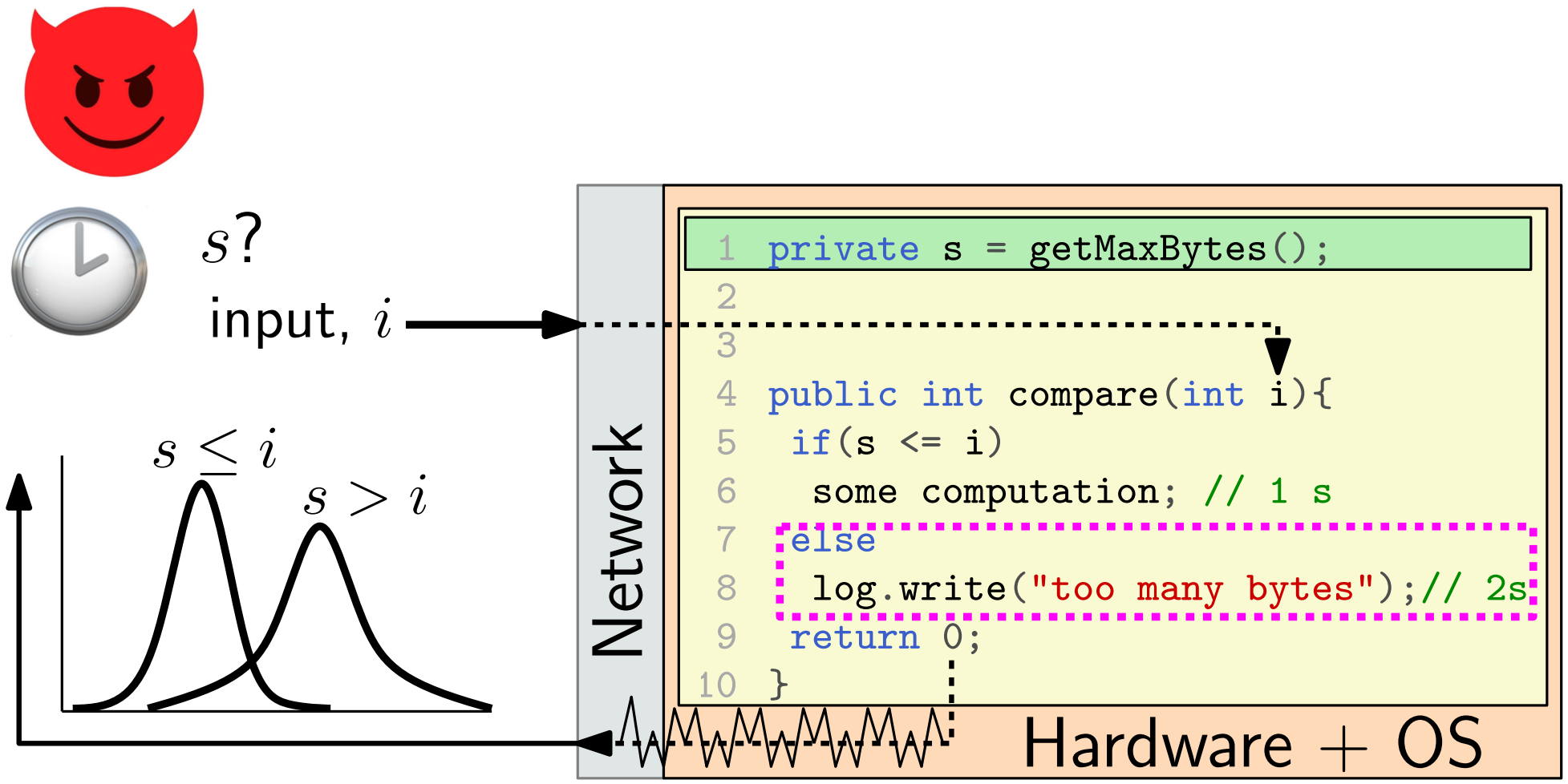

s?
input, i
Network
1 private s = getMaxBytes();
2
3
4 public int compare(int i){
5 if(s <= i)
6 some computation; // 1 s
7 else
8 log.write("too many bytes");// 2s
9 return 0;
10 }
Hardware + OS



$$s \leq i \Rightarrow o = 1$$


s?
input, i
Network
1 private s = getMaxBytes();
2
3
4 public int compare(int i){
5 if(s <= i)
6 some computation; // 1 s
7 else
8 log.write("too many bytes");// 2s
9 return 0;
10 }
Hardware + OS
$s \leq i \rightarrow o = 1$

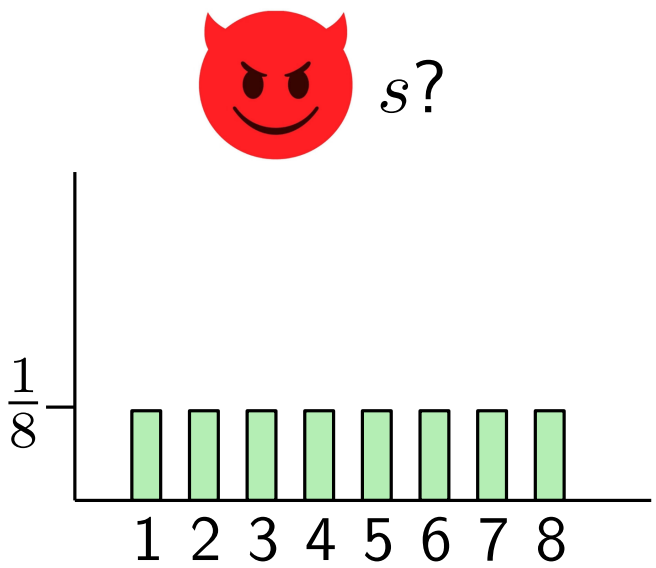

s?
input, i
$s \leq i$
Network
1 private s = getMaxBytes();
2
3
4 public int compare(int i){
5 if(s <= i)
6 some computation; // 1 s
7 else
8 log.write("too many bytes");// 2s
9 return 0;
10 }
Hardware + OS
$s \leq i \rightarrow o = 1$


s?
input, i
$s \leq i$
$s > i$
Network
1 private s = getMaxBytes();
2
3
4 public int compare(int i){
5 if(s <= i)
6 some computation; // 1 s
7 else
8 log.write("too many bytes");// 2s
9 return 0;
10 }
Hardware + OS

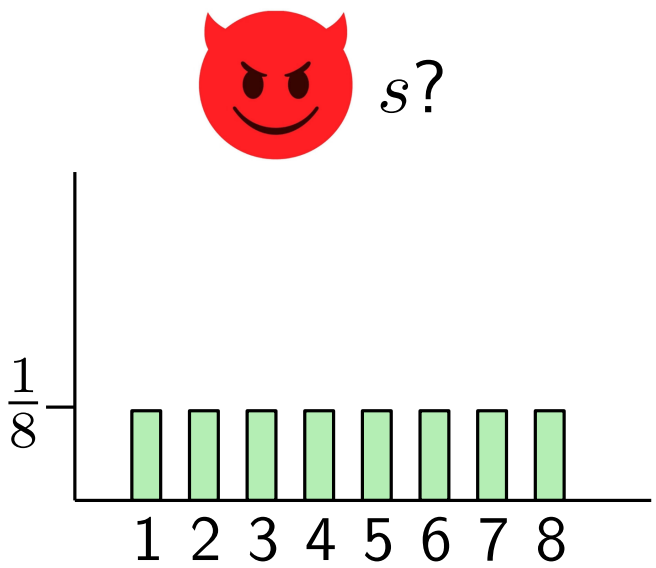
Attacker Belief

 s?

Attacker Belief



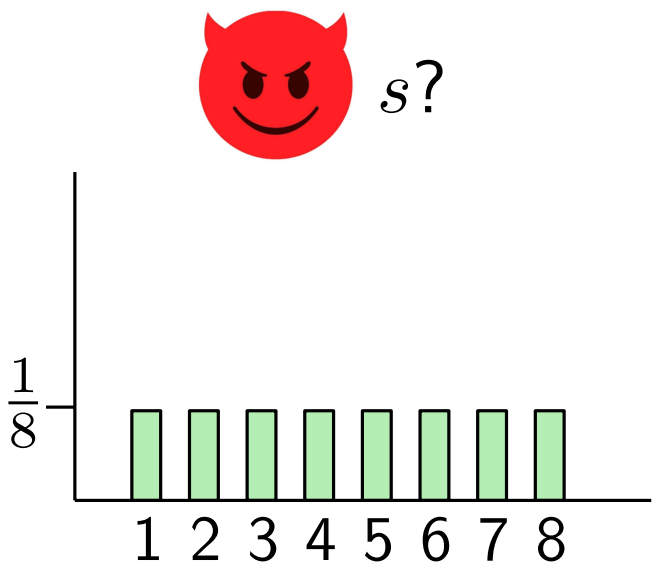
Attacker Belief



Input Choice

i^*

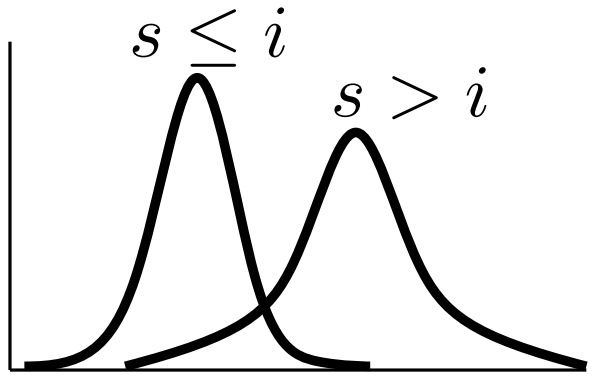
Attacker Belief



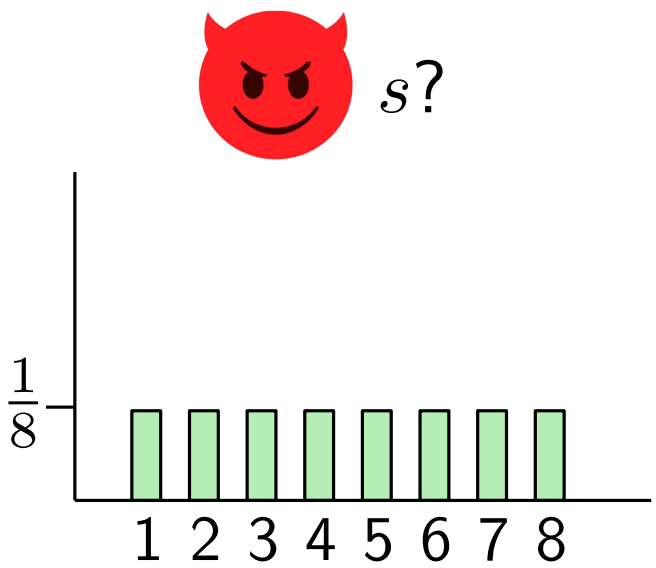
Input Choice

i^*

Observation Noise



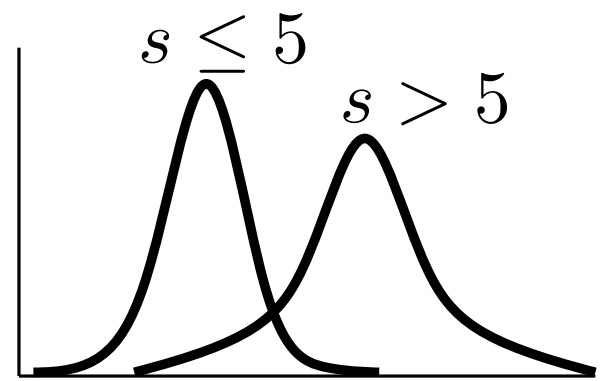
Attacker Belief



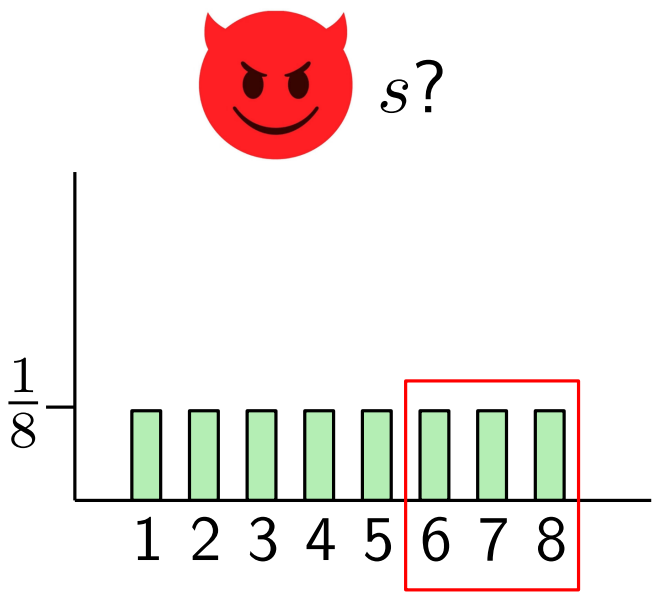
Input Choice

$$i^* = 5$$

Observation Noise



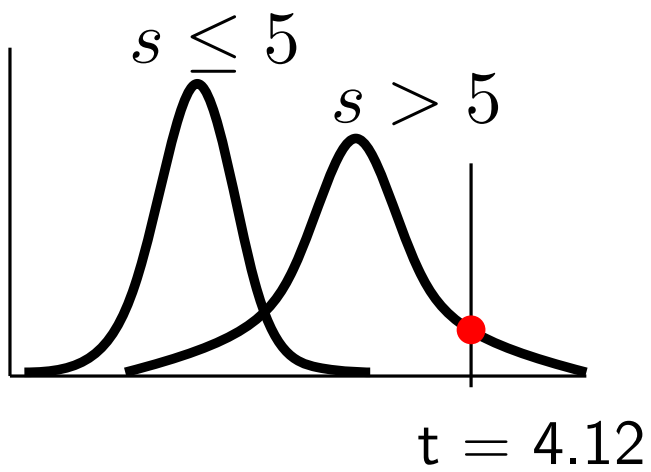
Attacker Belief



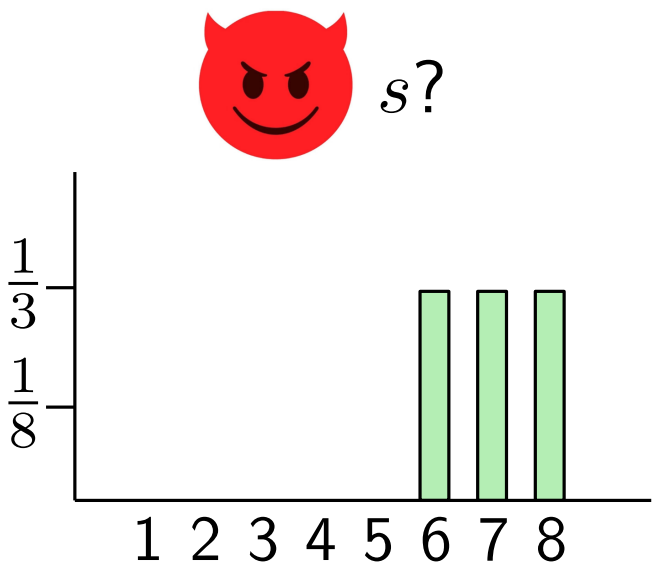
Input Choice

$i^* = 5$

Observation Noise



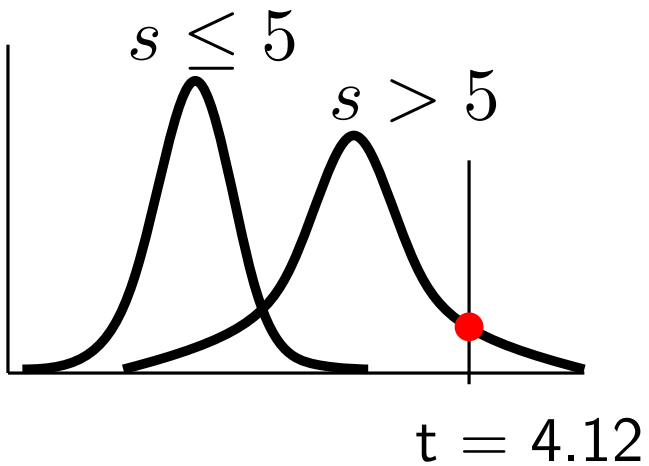
Attacker Belief



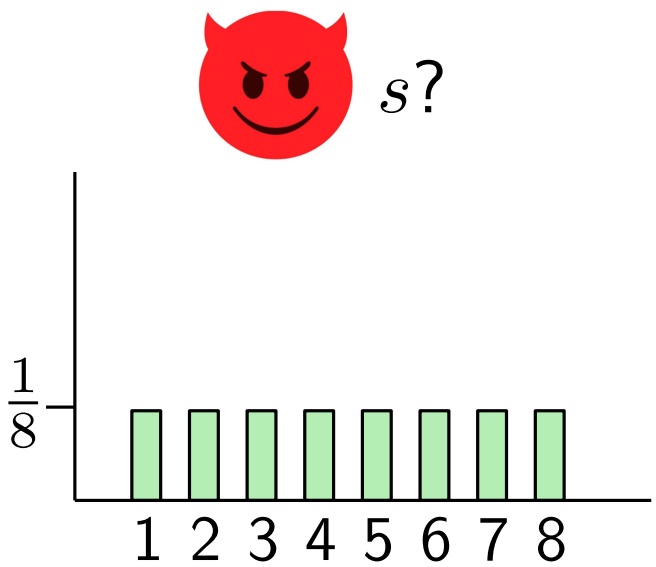
Input Choice

$$i^* = 5$$

Observation Noise



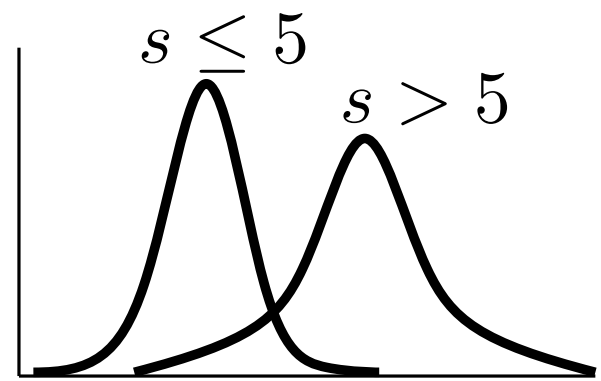
Attacker Belief



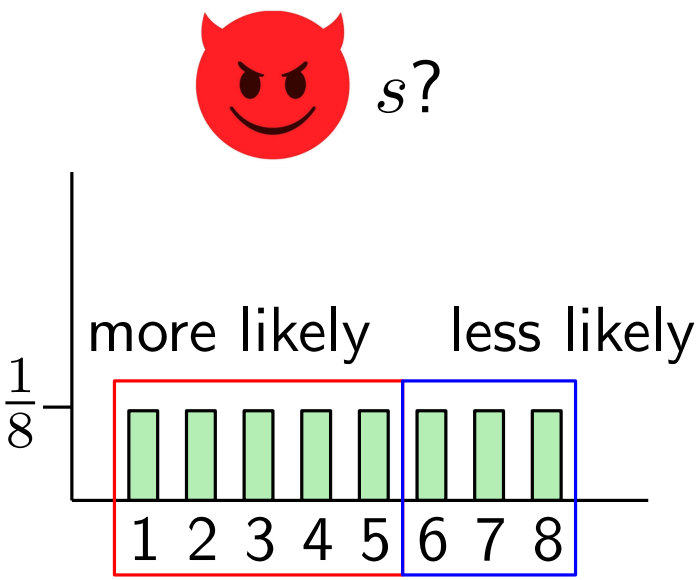
Input Choice

$$i^* = 5$$

Observation Noise



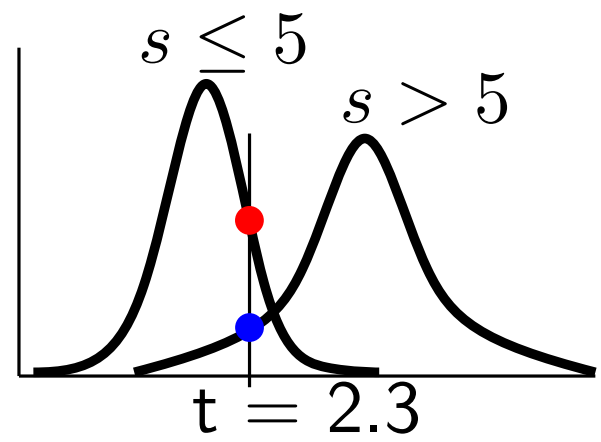
Attacker Belief



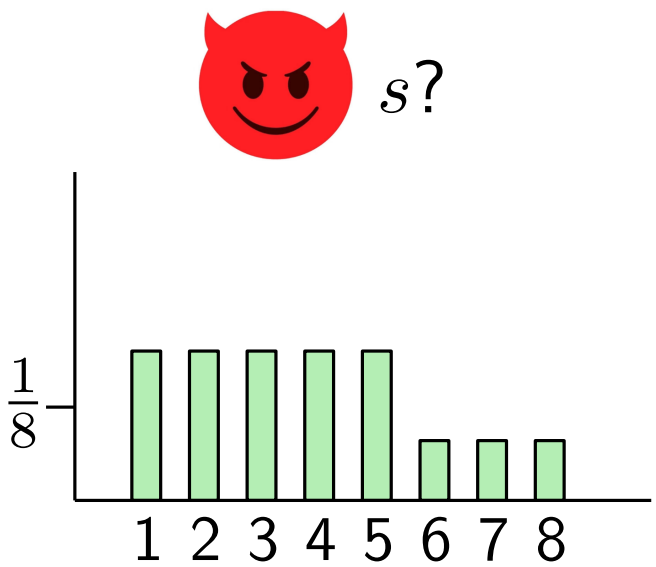
Input Choice

$i^* = 5$

Observation Noise



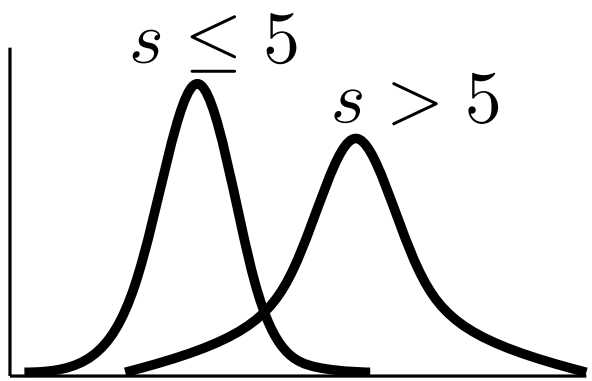
Attacker Belief



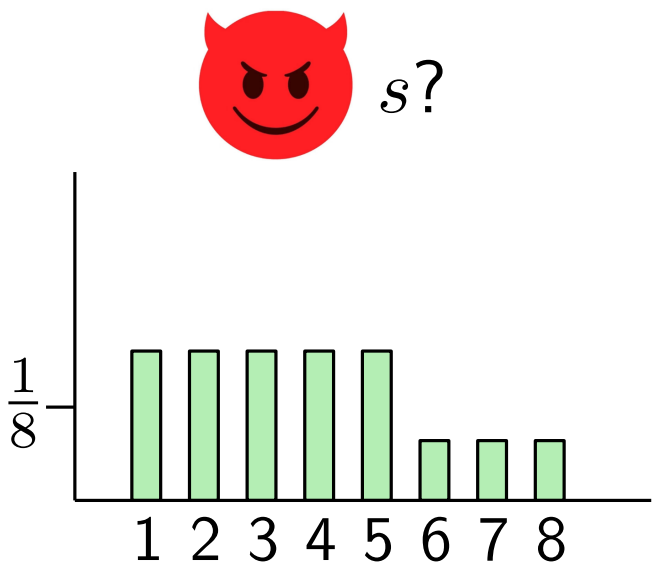
Input Choice

$$i^* = 5$$

Observation Noise



Attacker Belief

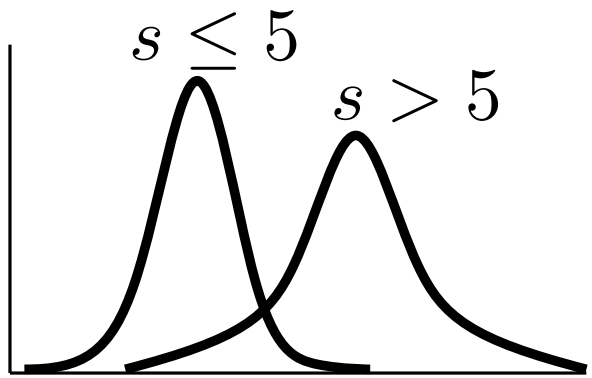


$p(s|o, i^*)$

Input Choice

$i^* = 5$

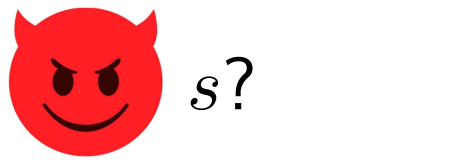
Observation Noise



Attacker Belief

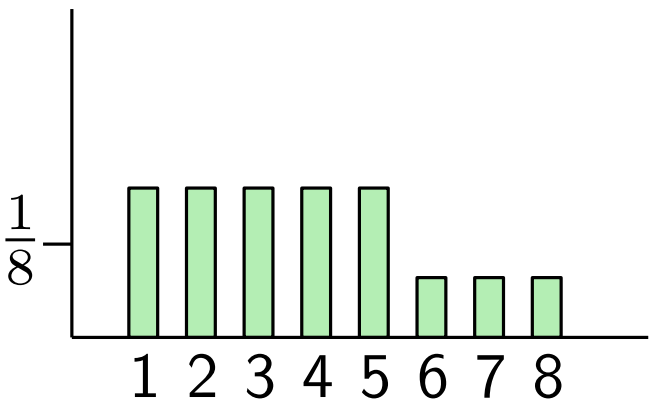
Input Choice

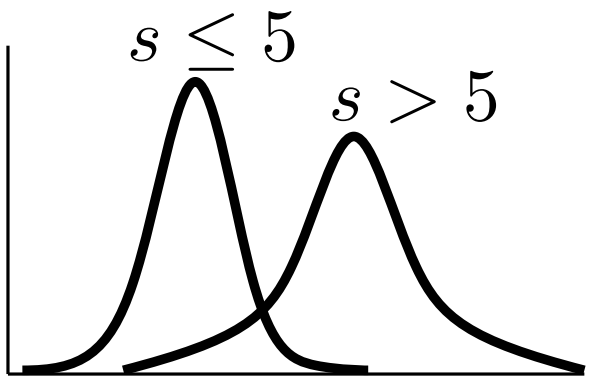
Observation Noise



$s?$

$i^* = 5$



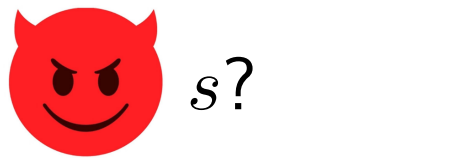


$p(s|o, i^*)$ ← $p(o|s, i)$

Attacker Belief

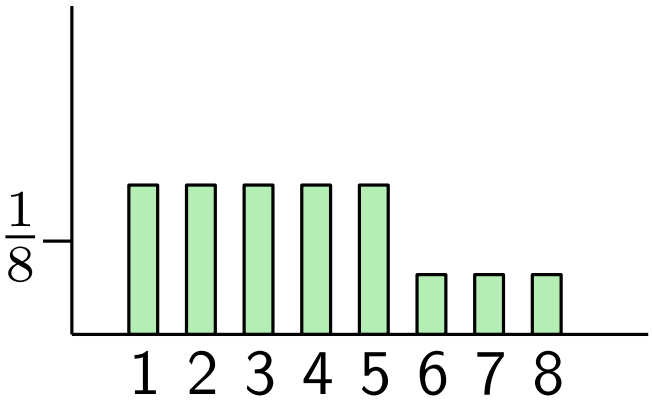
Input Choice

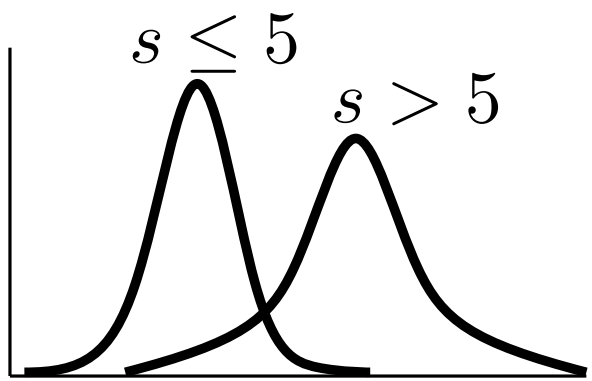
Observation Noise



$s?$

$i^* = 5$





$p(o|s,i)$

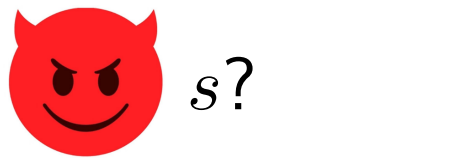
$p(s|o,i^*)$

$p(o|s,i)$

Attacker Belief

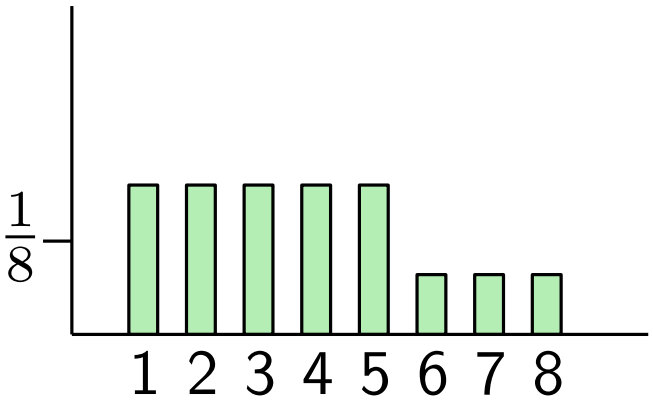
Input Choice

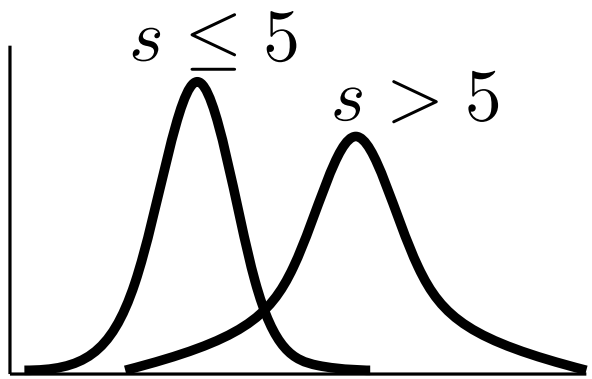
Observation Noise



$s?$

$i^* = 5$





$p(o|s, i^*)$

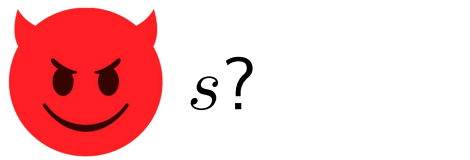
$p(s|o, i^*)$

$p(o|s, i)$

Attacker Belief

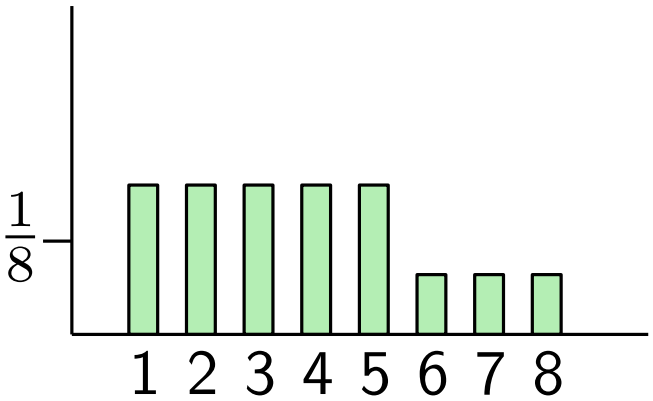
Input Choice

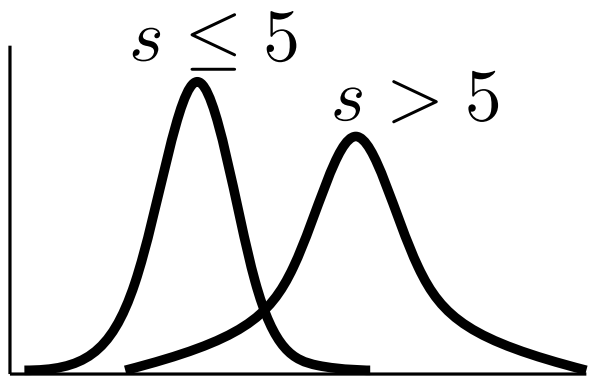
Observation Noise



$s?$

$i^* = 5$





$p(o|s, i^*)$

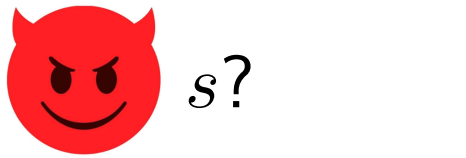
$p(s|o, i^*)$

$p(o|s, i)$

Attacker Belief

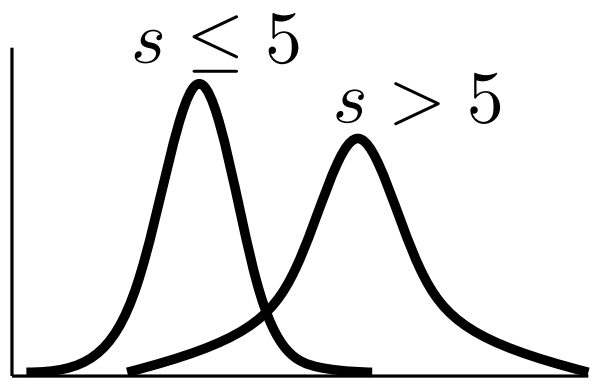
Input Choice

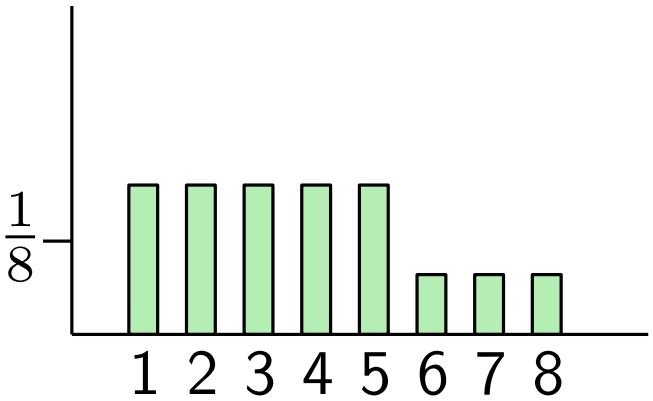
Observation Noise



$s?$

$i^* = 5$





$p(s|o, i^*)$

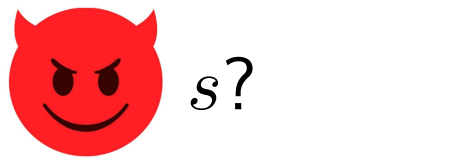
$p(s|o, i^*)$

$p(o|s, i)$

Attacker Belief

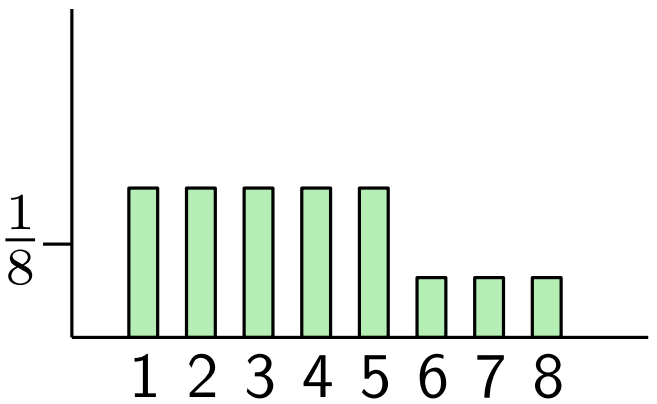
Input Choice

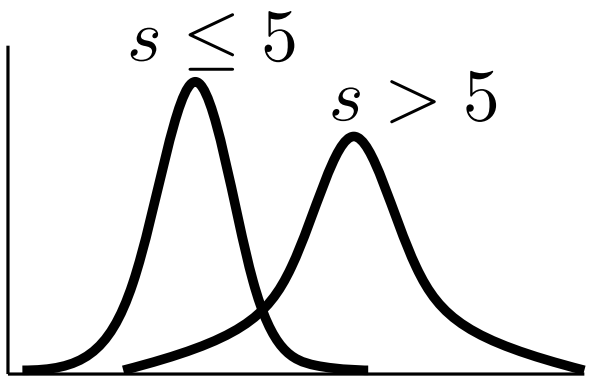
Observation Noise



$s?$

$i^* = 5$





$p(s|o, i^*)$

$p(s|o, i^*)$

Bayes' Rule

$p(o|s, i)$

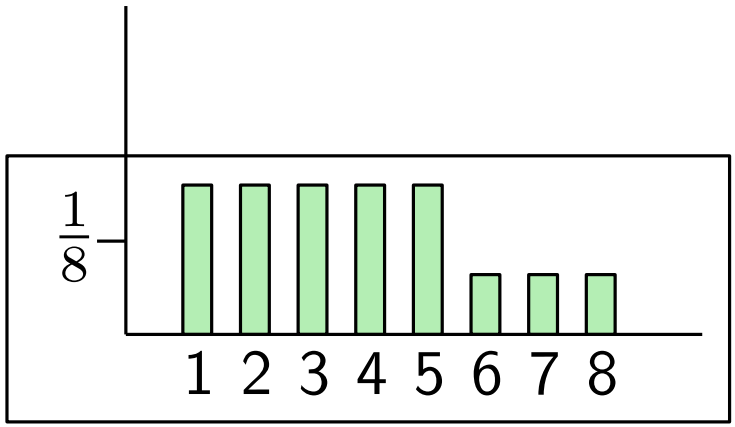
Attacker Belief

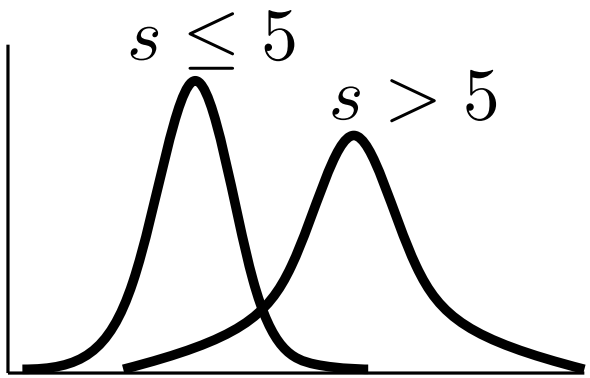
Input Choice

Observation Noise

$s?$

$i^* = 5$



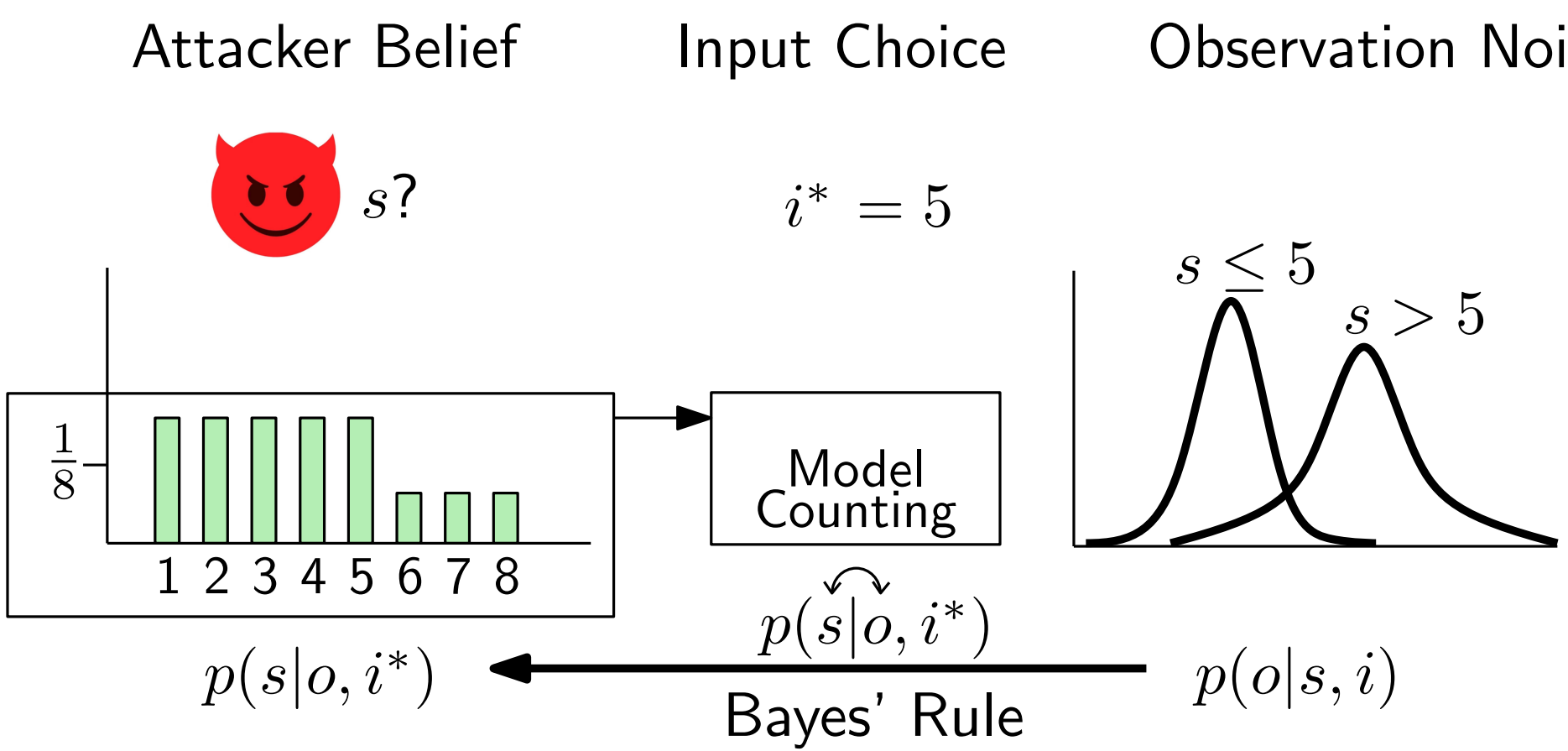


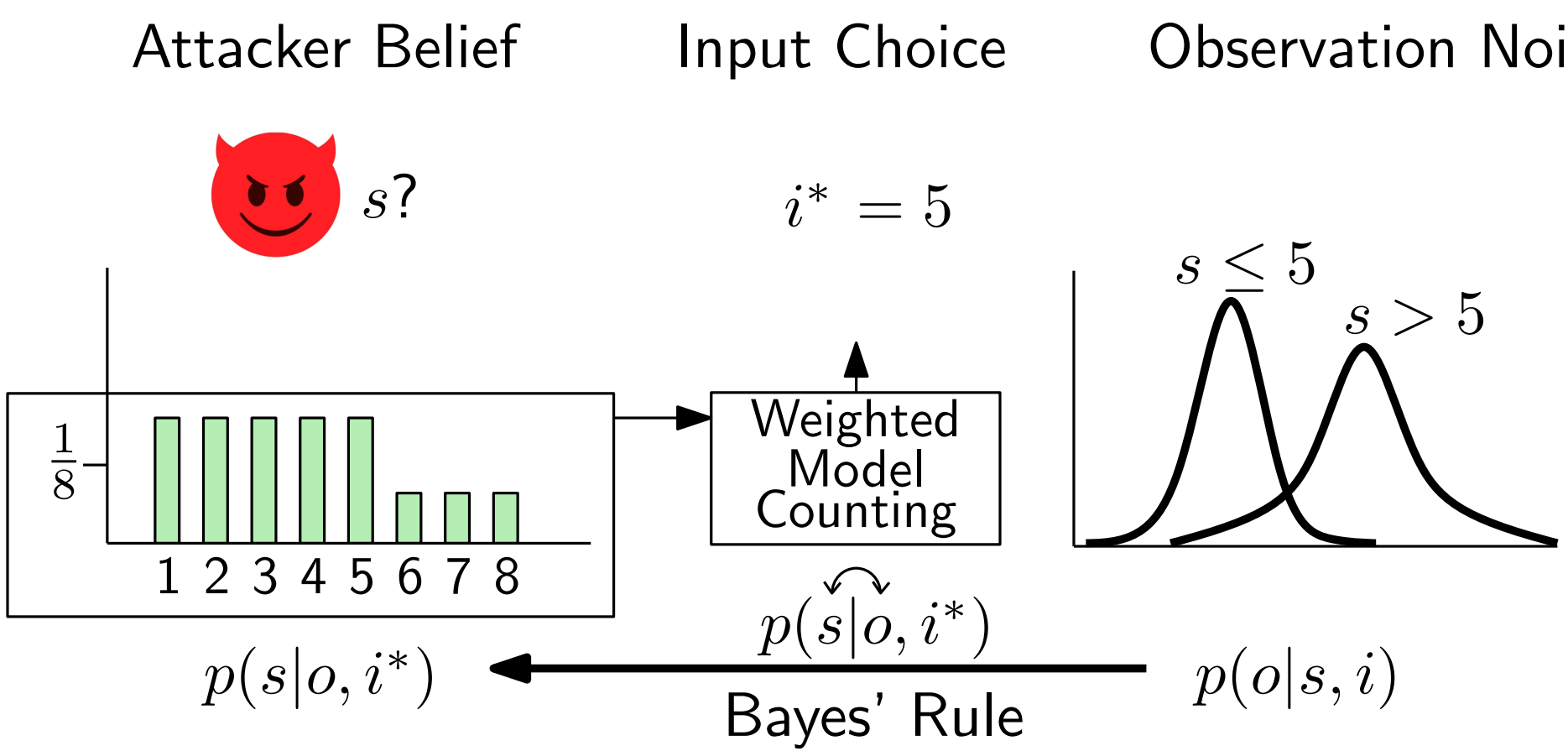
$p(s|o, i^*)$

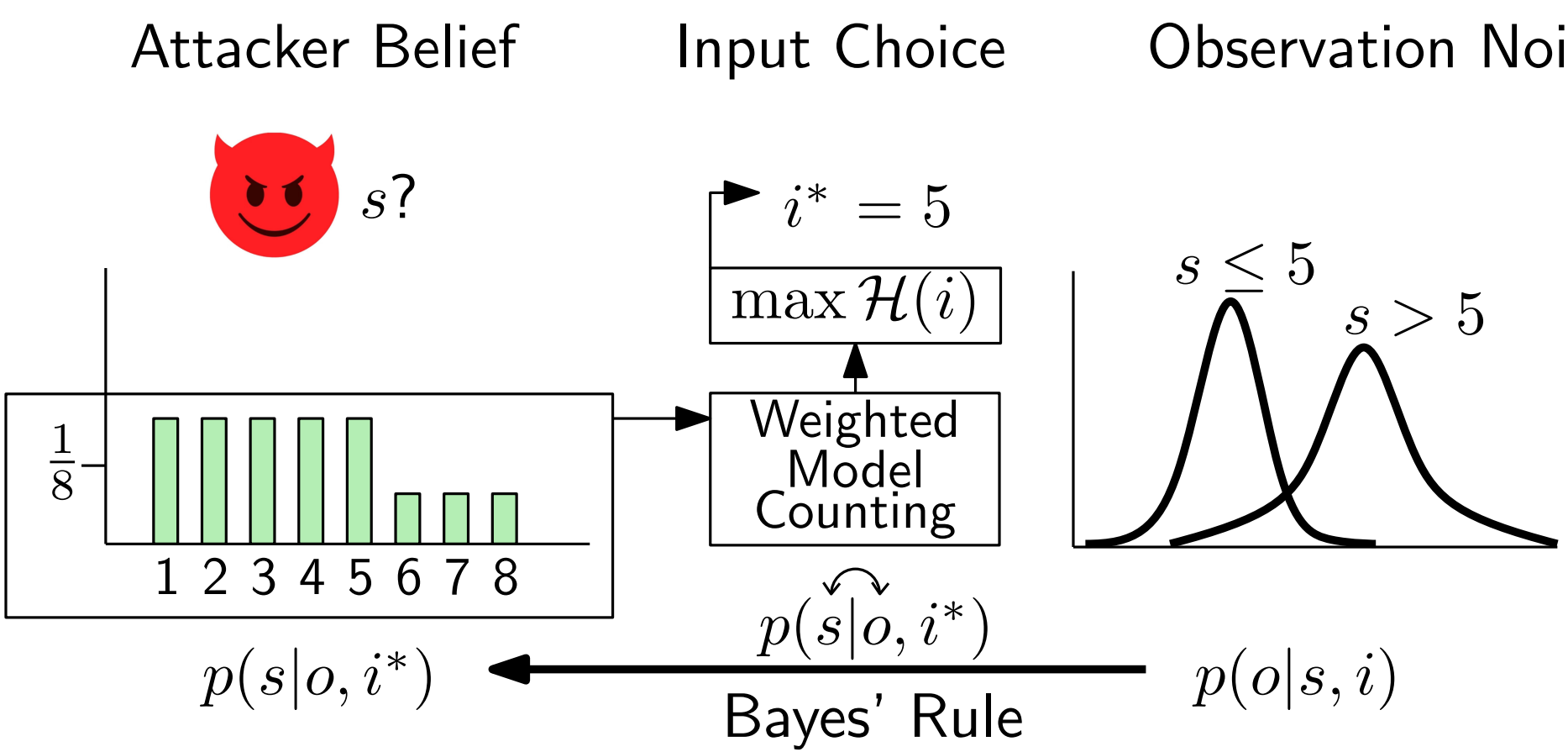
$p(s|o, i^*)$

Bayes' Rule

$p(o|s, i)$


Attacker Belief
Input Choice
Observation Noise
s?
$i^* = 5$
$s \leq 5$
$s > 5$
$\frac{1}{8}$
1 2 3 4 5 6 7 8
Model Counting
$p(s|o, i^*)$
$p(s|o, i^*)$
$p(o|s, i)$
Bayes' Rule


Attacker Belief
Input Choice
Observation Noise
s?
$i^* = 5$
$s \leq 5$
$s > 5$
$\frac{1}{8}$
1 2 3 4 5 6 7 8
Weighted Model Counting
$p(s|o, i^*)$
$p(s|o, i^*)$
$p(o|s, i)$
Bayes' Rule


Attacker Belief
Input Choice
Observation Noise
s?
$i^* = 5$
$\max \mathcal{H}(i)$
Weighted Model Counting
$s \leq 5$
$s > 5$
$\frac{1}{8}$
1 2 3 4 5 6 7 8
$p(s|o, i^*)$
$p(s|o, i^*)$
Bayes' Rule
$p(o|s, i)$

1. Offline Static Analysis

1. Offline Static Analysis

2. Offline Dynamic Analysis

1. Offline Static Analysis

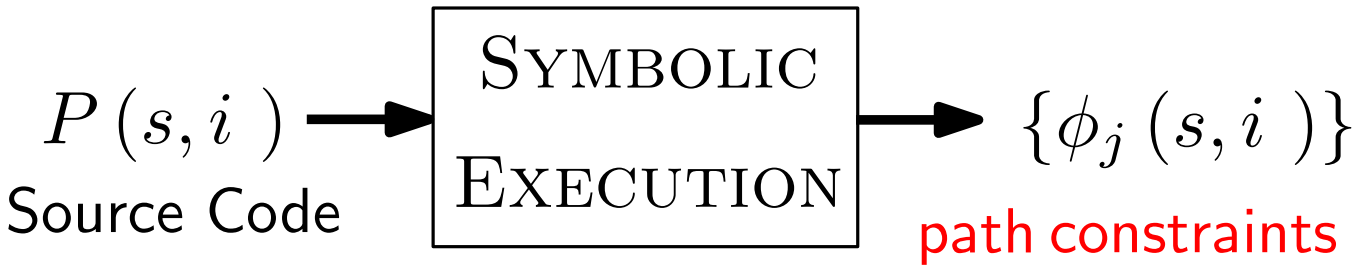
2. Offline Dynamic Analysis

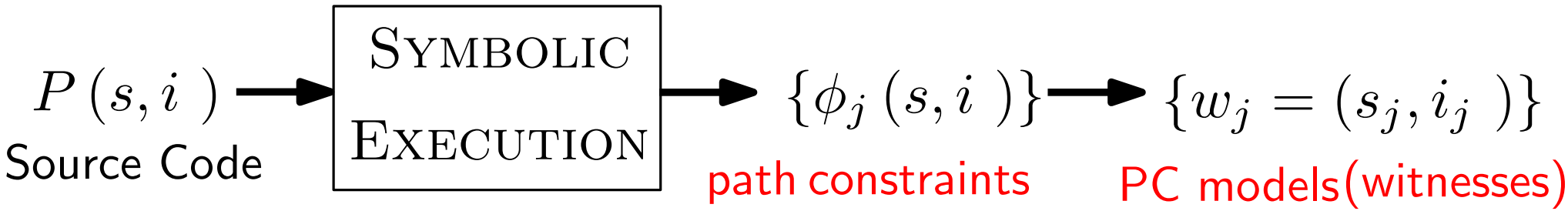
3. Online Attack Synthesis

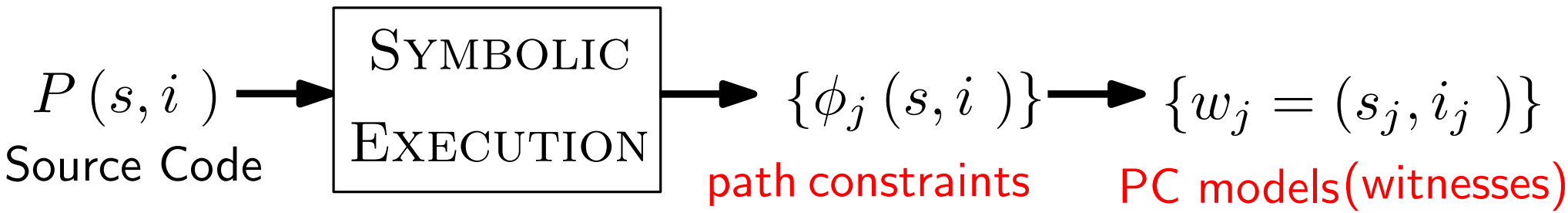
1. Offline Static Analysis

2. Offline Dynamic Analysis

3. Online Attack Synthesis

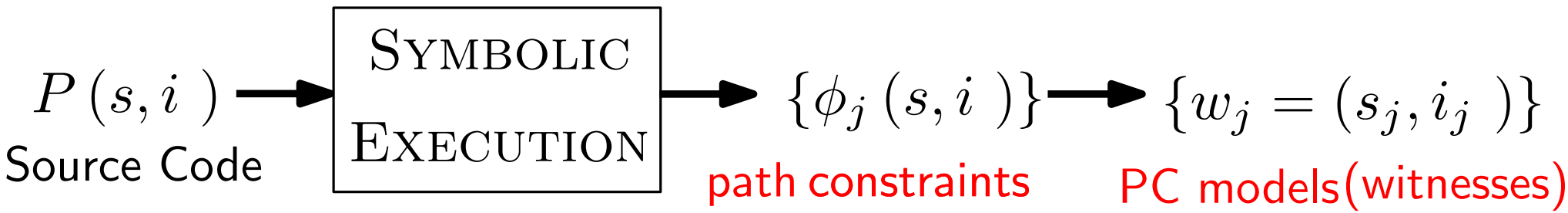

$P\,(s, i\,)$
Source Code
Symbolic
Execution
$\{\phi_j\,(s, i\,)\}$
path constraints


$P\,(s, i\,)$
Source Code
SYMBOLIC EXECUTION
$\{\phi_j\,(s, i\,)\}$
path constraints
$\{w_j = (s_j, i_j\,)\}$
PC models(witnesses)



Each PC characterizes an observable program behavior

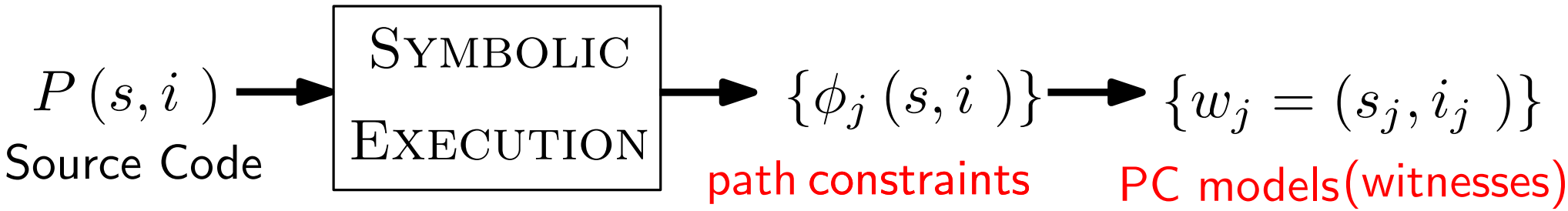
$$(s, i\,) \models \phi_j \qquad\qquad (s', i'\,) \models \phi_j$$



Each PC characterizes an observable program behavior

$$(s,i\,) \models \phi_j \qquad\qquad (s',i'\,) \models \phi_j$$

$$P\,(s,i\,) \quad ? \quad ? \quad P\,(s',i'\,)$$



Each PC characterizes an observable program behavior

$$(s,i\,) \models \phi_j \qquad\qquad (s',i'\,) \models \phi_j$$

$$P\,(s,i\,) \quad ? \quad \text{  } \quad ? \quad P\,(s',i'\,)$$

$\phi_j\,(s,i\,)$ characterizes observationally indistinguishable behaviors

$P\,(s,i\,)$ is a representative of all behaviors in that class

1. Offline Static Analysis

2. Offline Dynamic Analysis

3. Online Attack Synthesis

1. Offline Static Analysis

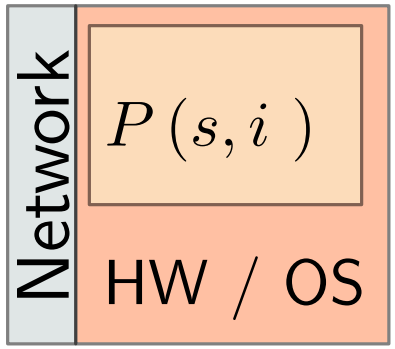
2. Offline Dynamic Analysis

3. Online Attack Synthesis

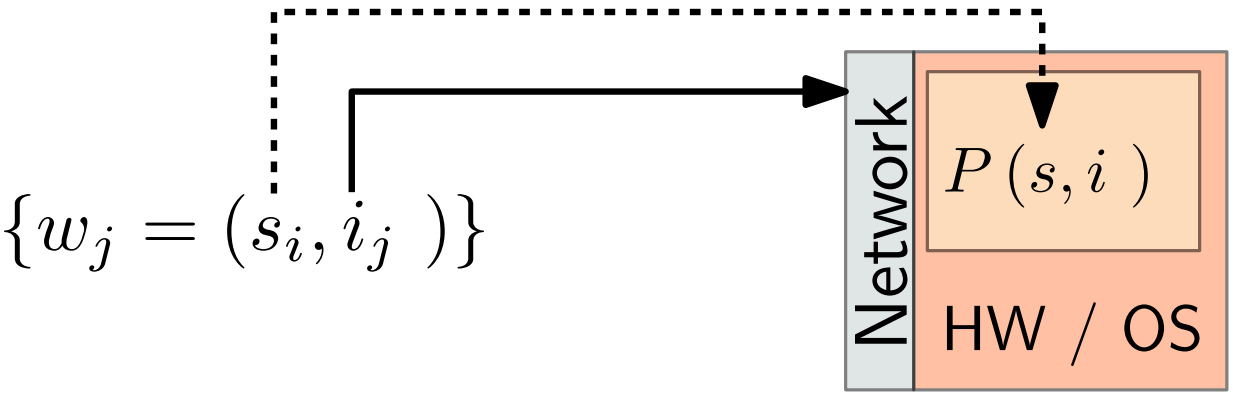
Characterize effect of noise on each class of program behaviors using the witness for that behavior.

Characterize effect of noise on each class of program behaviors using the witness for that behavior.

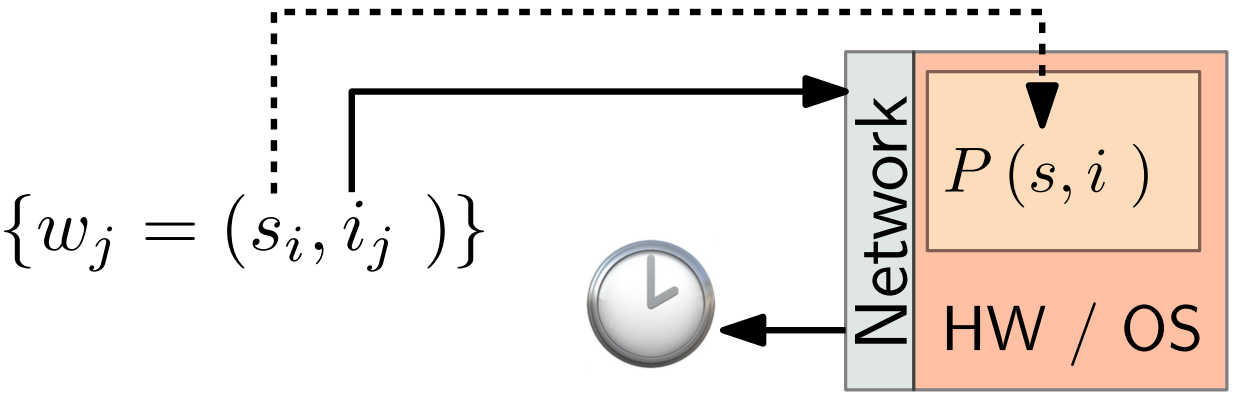
$\{w_j = (s_i, i_j\ )\}$



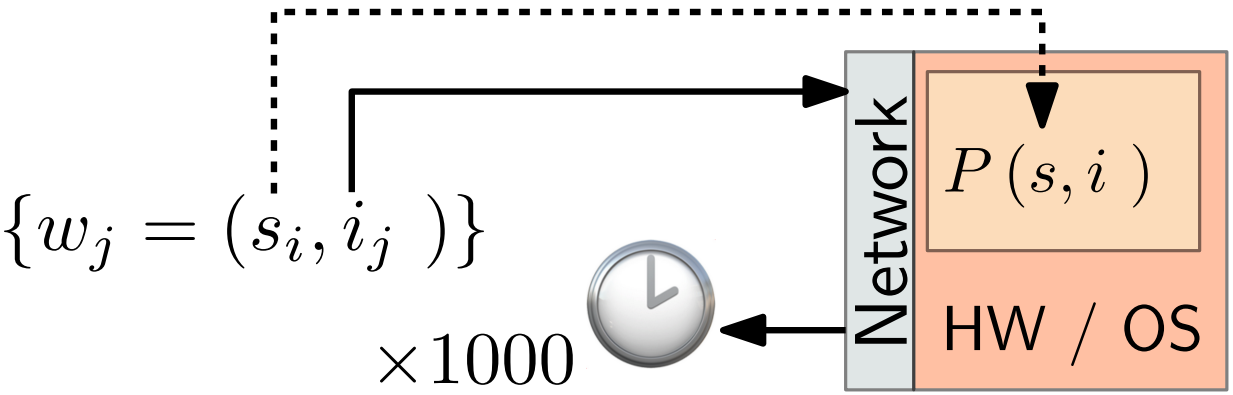
Characterize effect of noise on each class of program behaviors using the witness for that behavior.



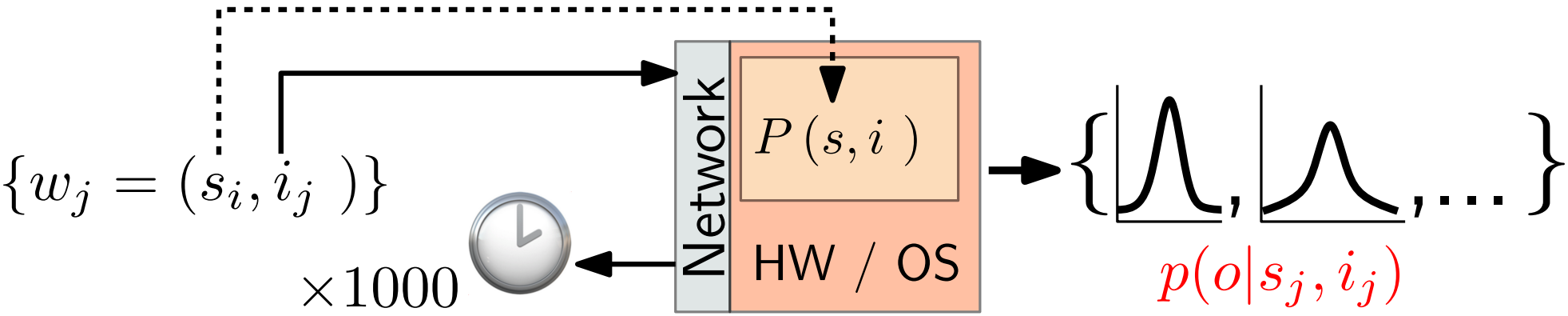
Characterize effect of noise on each class of program behaviors using the witness for that behavior.



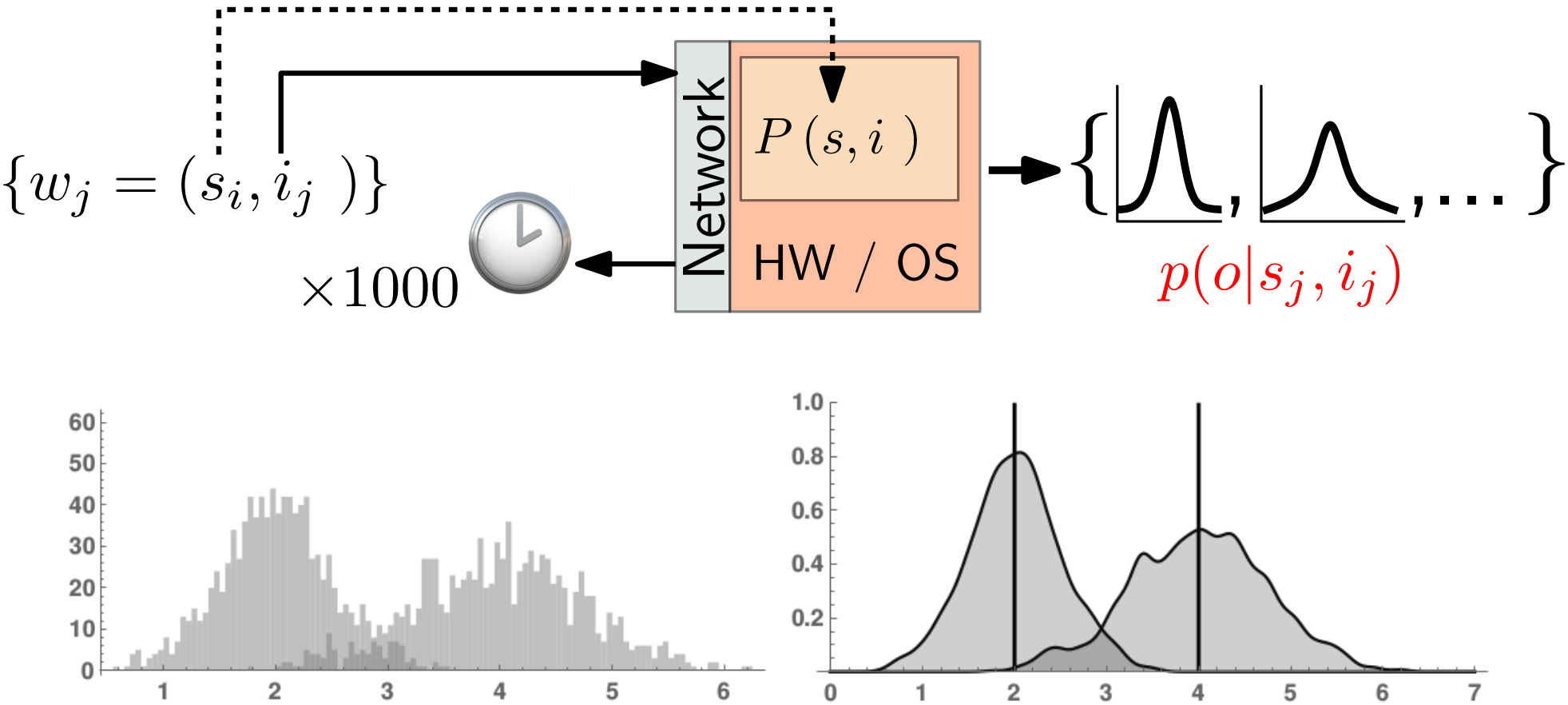
Characterize effect of noise on each class of program behaviors using the witness for that behavior.



Characterize effect of noise on each class of program behaviors using the witness for that behavior.

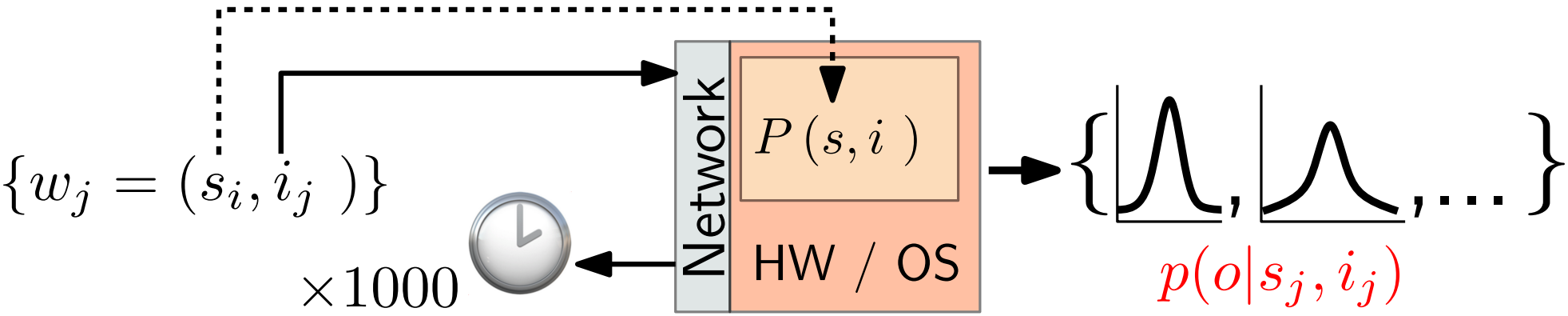


Characterize effect of noise on each class of program behaviors using the witness for that behavior.



Smooth Kernel Density Estimation

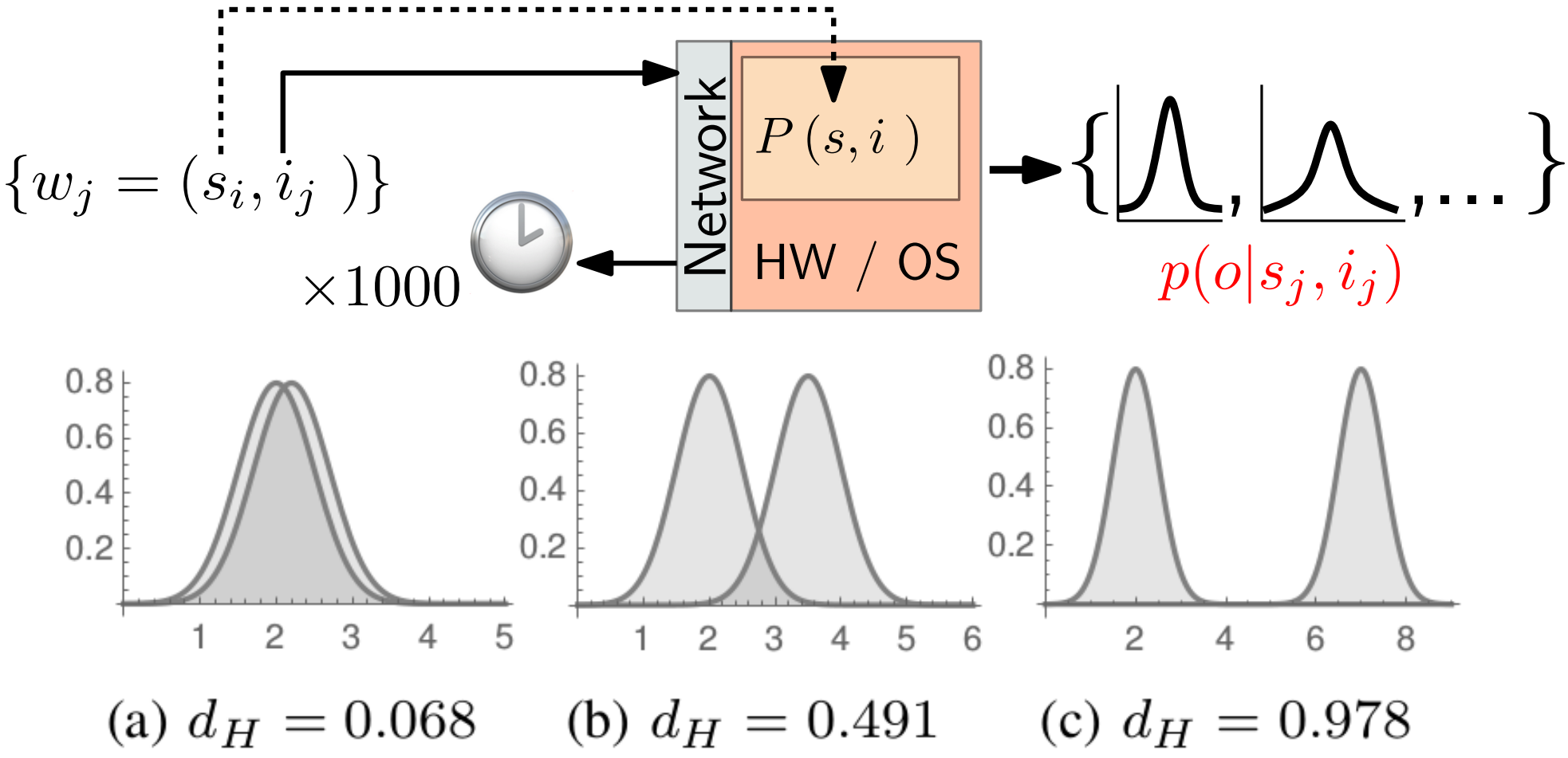
Characterize effect of noise on each class of program behaviors using the witness for that behavior.



$$\hat{p}(o|\phi) = \hat{p}(o|w) = \frac{1}{nh}\sum_{r=1}^{n} K\left(\frac{o - o_r}{h}\right)$$

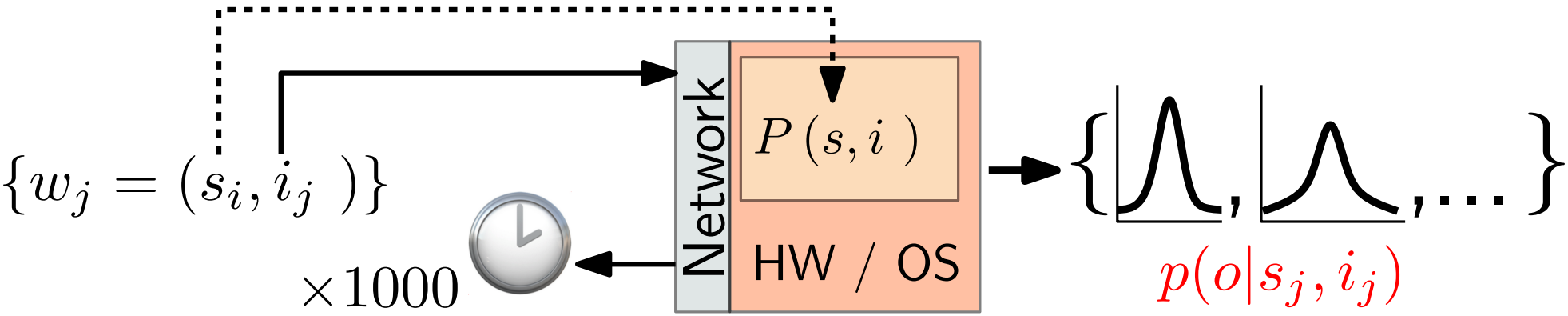
Smooth Kernel Density Estimation

Characterize effect of noise on each class of program behaviors using the witness for that behavior.



Merging via Hellinger Distance

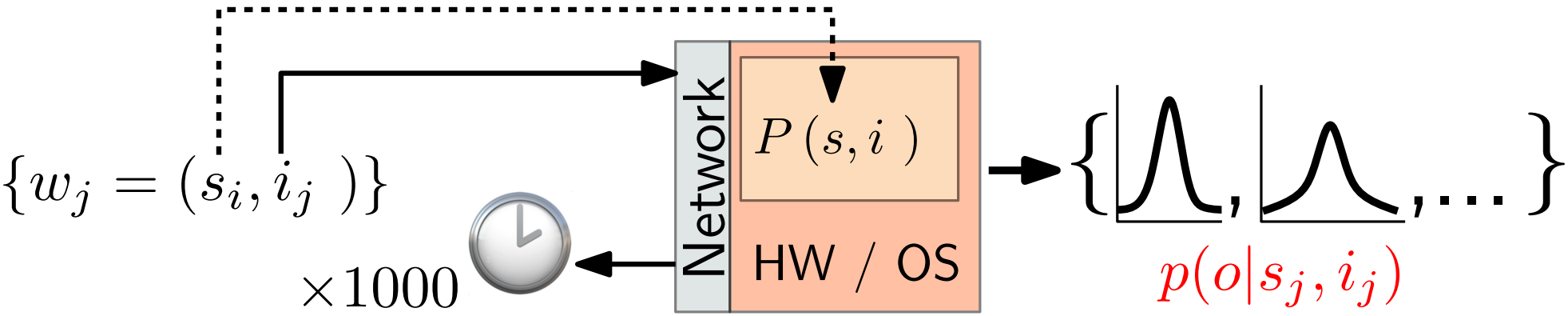
Characterize effect of noise on each class of program behaviors using the witness for that behavior.



$$d_H(p,q) = \sqrt{\frac{1}{2}\int_{-\infty}^{\infty}\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx}$$

Merging via Hellinger Distance

Characterize effect of noise on each class of program behaviors using the witness for that behavior.



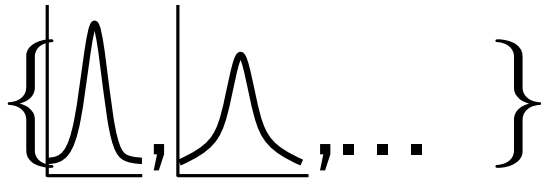
$$d_H(\hat{p}(o|\phi_1), \hat{p}(o|\phi_2)) < \tau \quad \Rightarrow \quad \text{let} \;\; \phi' = \phi_1 \vee \phi_2$$

Observation constraint ϕ':
Disjunction over path constraints which characterizes inputs that are observationally indistinguishable via side-channel observation.

1. Offline Static Analysis

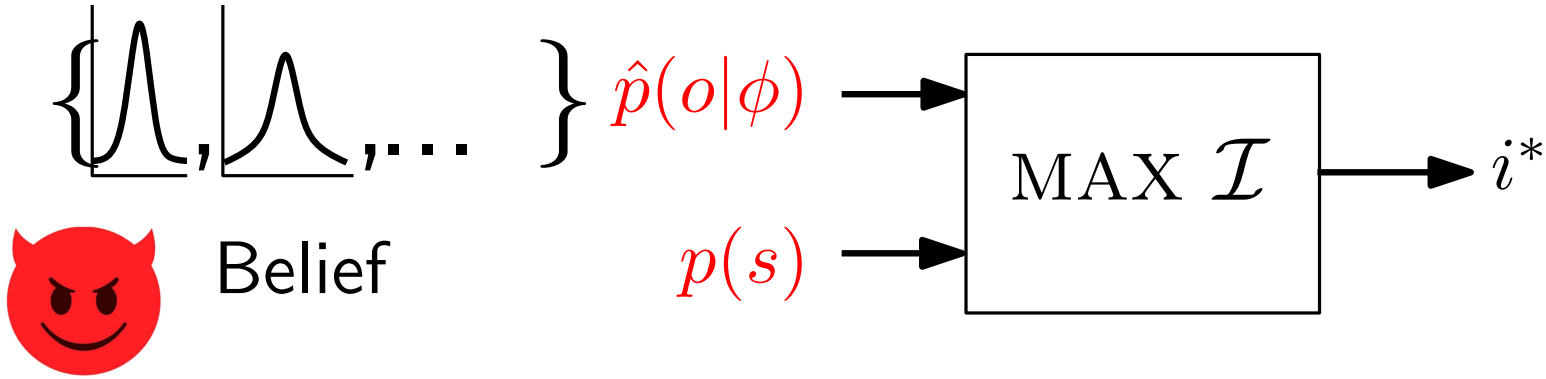
2. Offline Dynamic Analysis

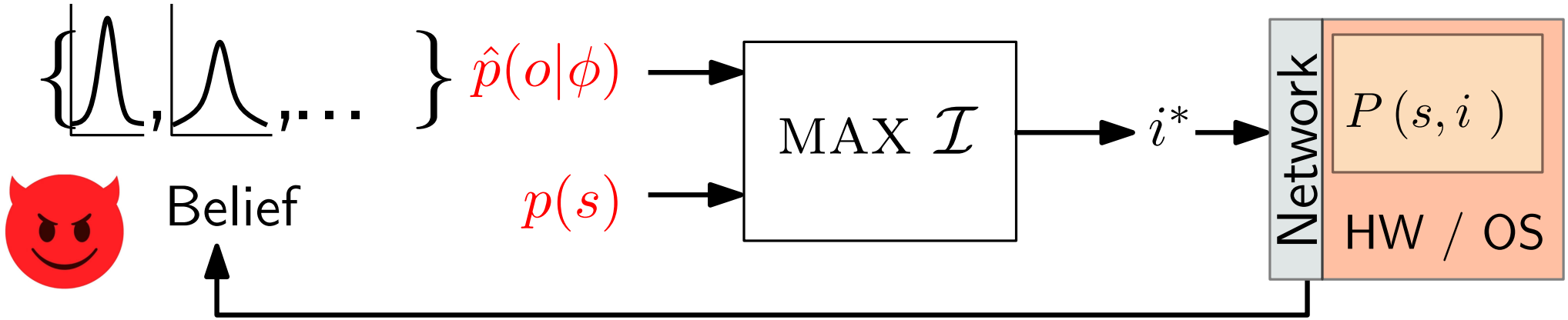
3. Online Attack Synthesis

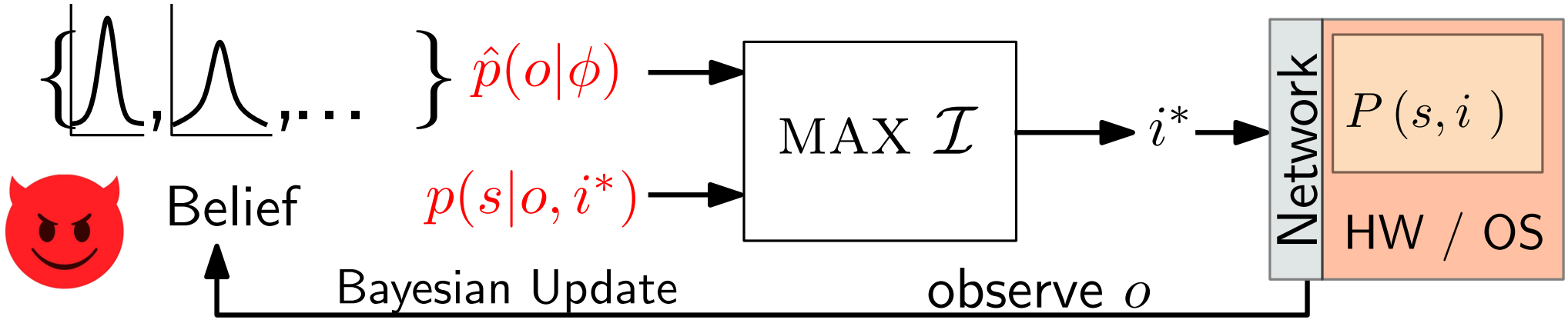


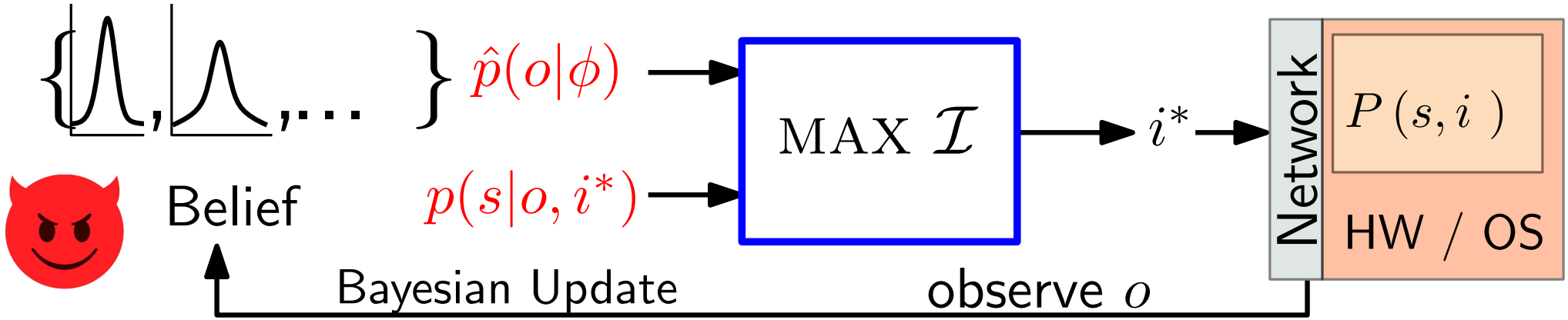
$\hat{p}(o|\phi)$

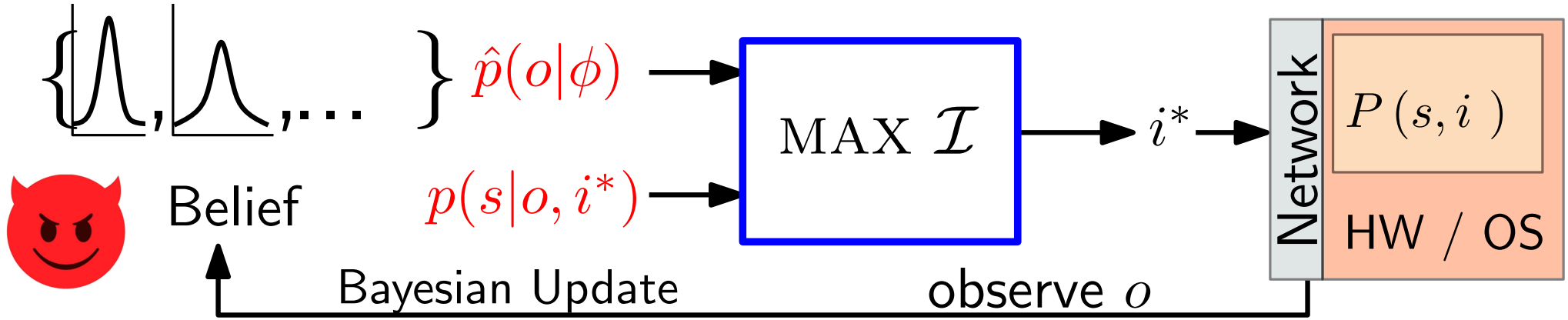
Belief $p(s)$


$\hat{p}(o|\phi)$
$p(s)$
MAX $\mathcal{I}$
i^*
Belief


$\hat{p}(o|\phi)$
$p(s)$
MAX $\mathcal{I}$
i^*
Network
$P(s,i)$
HW / OS
Belief


$\hat{p}(o|\phi)$
$p(s|o,i^*)$
MAX $\mathcal{I}$
i^*
Network
$P(s,i)$
HW / OS
Belief
Bayesian Update
observe o

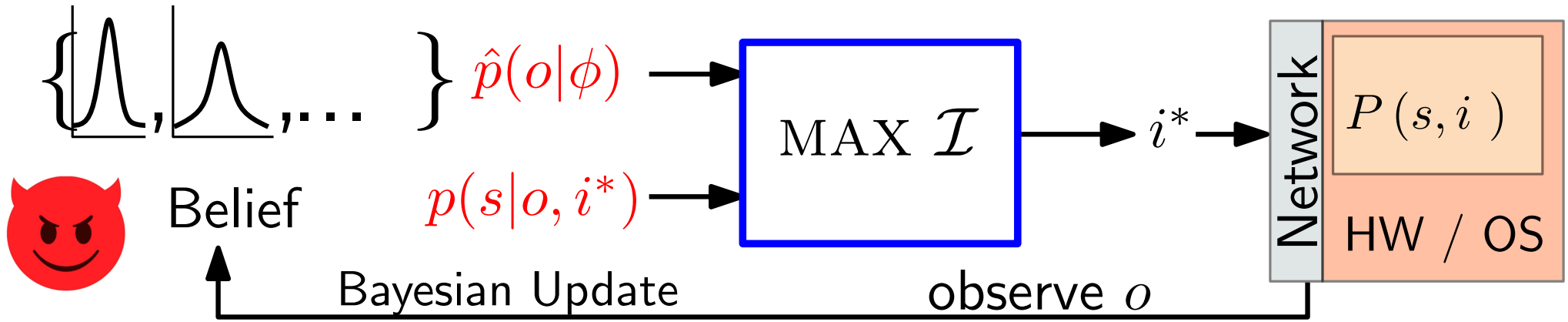

$\hat{p}(o|\phi)$
$p(s|o,i^*)$
MAX $\mathcal{I}$
i^*
Network
$P(s,i)$
HW / OS
Belief
Bayesian Update
observe o



$$\mathcal{I}(s;\phi(s,i)|i) = -\sum_{i=1}^{n} p(\phi(s,i)|i)\log_2 p(\phi(s,i)|i)$$

Expected info gain given attacker input

Observation constraint probabilities

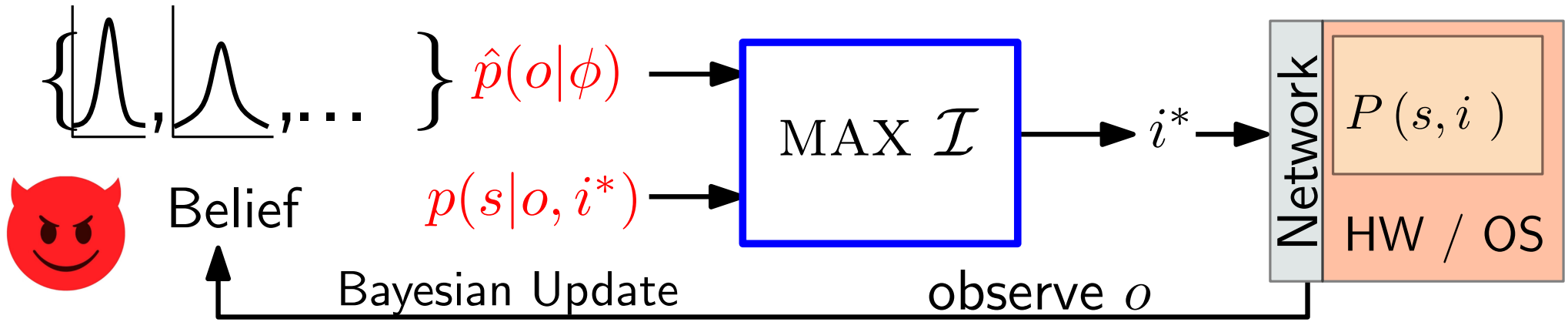


$$\mathcal{I}(s;\phi(s,i)|i) = -\sum_{i=1}^{n} p(\phi(s,i)|i)\log_2 p(\phi(s,i)|i)$$

Expected info gain given attacker input

Observation constraint probabilities

$$p(\phi(s,i)|i) = \sum_{s\in S} \qquad \phi(s,i)$$



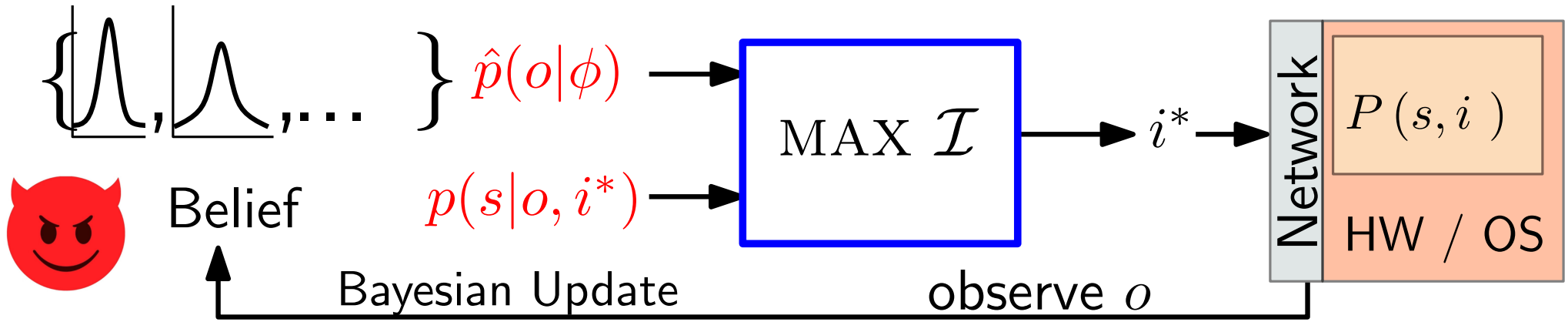
$$\mathcal{I}(s;\phi(s,i)|i) = -\sum_{i=1}^{n} p(\phi(s,i)|i)\log_2 p(\phi(s,i)|i)$$

Expected info gain given attacker input

Observation constraint probabilities

$$p(\phi(s,i)|i) = \sum_{s\in S} \qquad \phi(s,i)$$

Model Counting



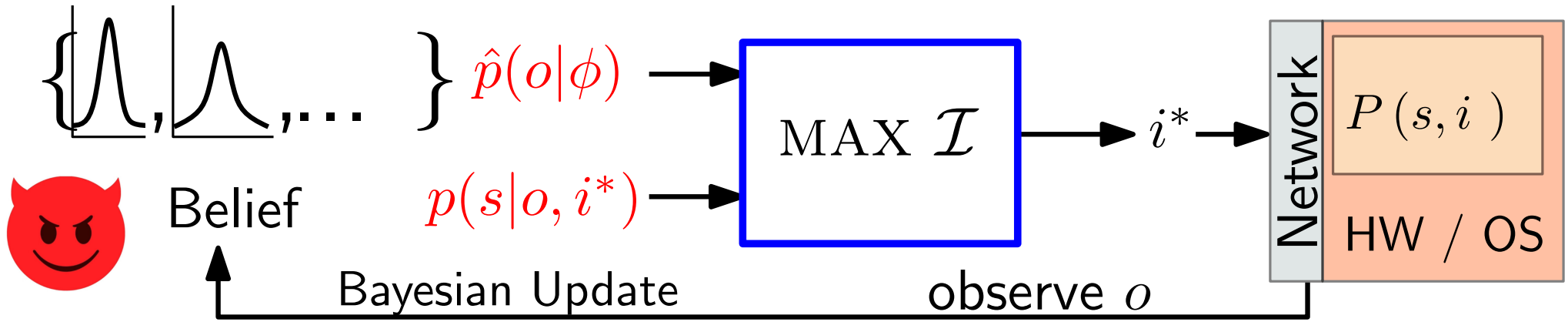
$$\mathcal{I}(s;\phi(s,i)|i) = -\sum_{i=1}^{n} p(\phi(s,i)|i)\log_2 p(\phi(s,i)|i)$$

Expected info gain given attacker input

Observation constraint probabilities

$$p(\phi(s,i)|i) = \sum_{s\in S} p(s)\phi(s,i)$$

Weighted Model Counting



$$\mathcal{I}(s;\phi(s,i)|i) = -\sum_{i=1}^{n} p(\phi(s,i)|i)\log_2 p(\phi(s,i)|i)$$

Expected info gain given attacker input

Observation constraint probabilities

$$p(\phi(s,i)|i) = \boxed{\sum_{s\in S} p(s)\phi(s,i)}$$ BARVINOK

1. Offline Static Analysis

2. Offline Dynamic Analysis

3. Online Attack Synthesis

Implementation

1. Offline Static Analysis

NASA Symbolic PathFinder (SPF)

Z3 Constraint Solver

2. Offline Dynamic Analysis

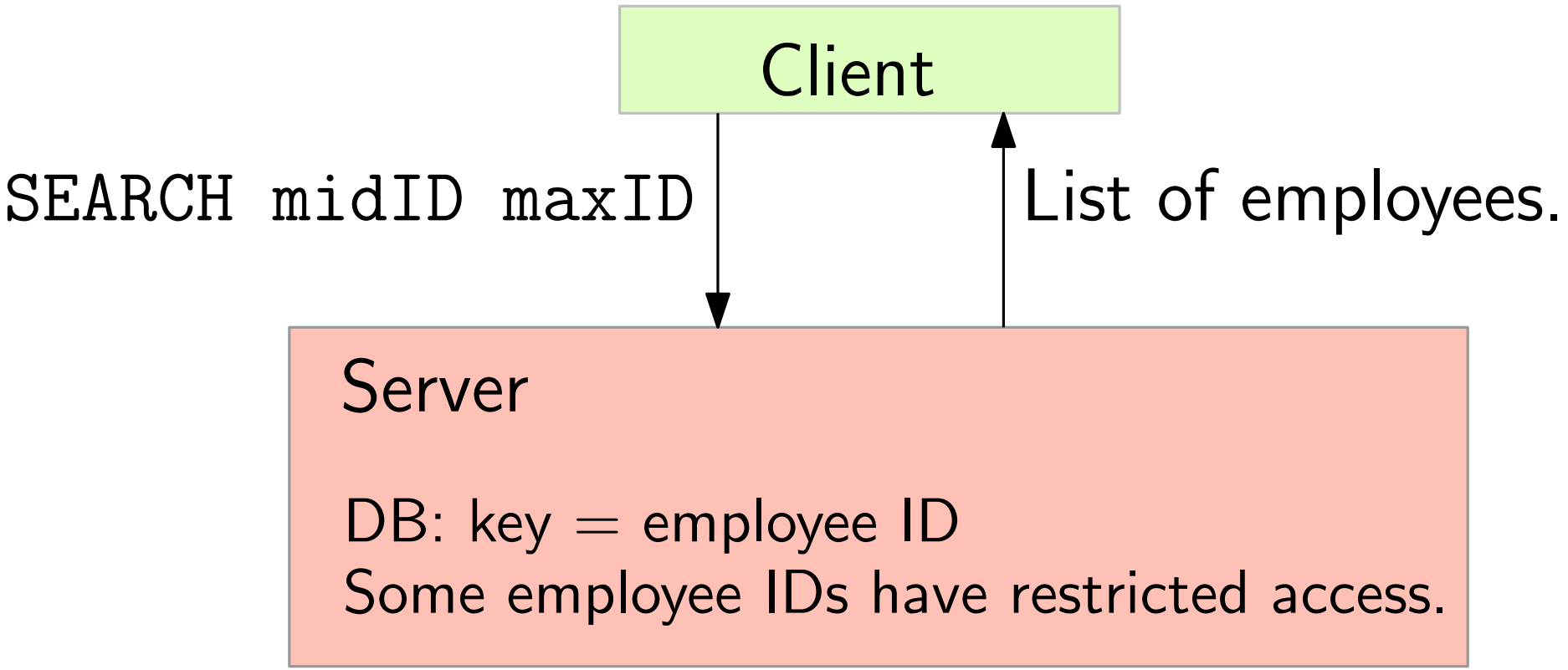
3. Online Attack Synthesis

Barvinok Weighted Symbolic Model Counting

Mathematica Symbolic Entropy Computation Numeric Maximization

Case Study: LawDB

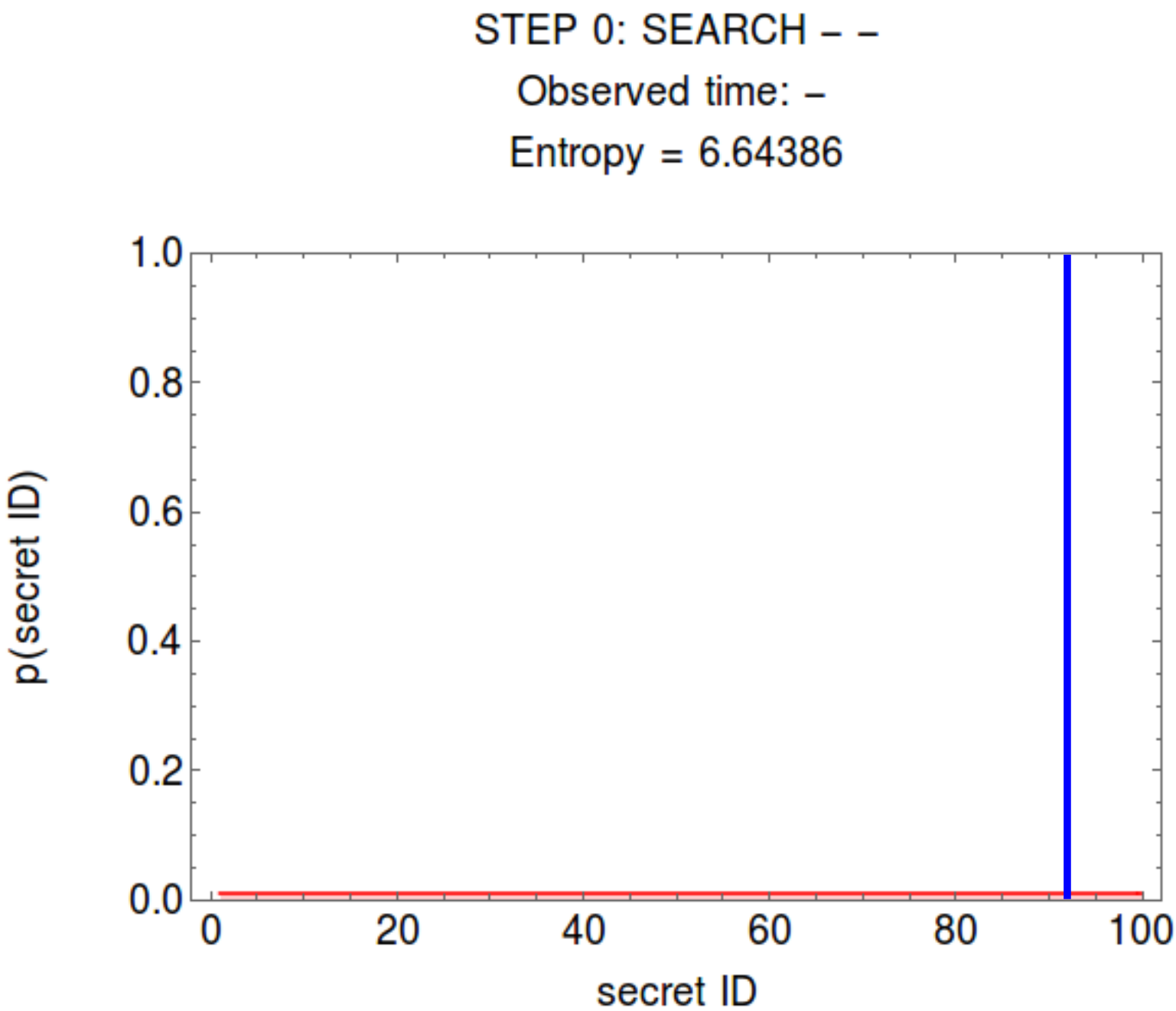
From DARPA Space-Time Analysis for Cybersecurity (STAC)



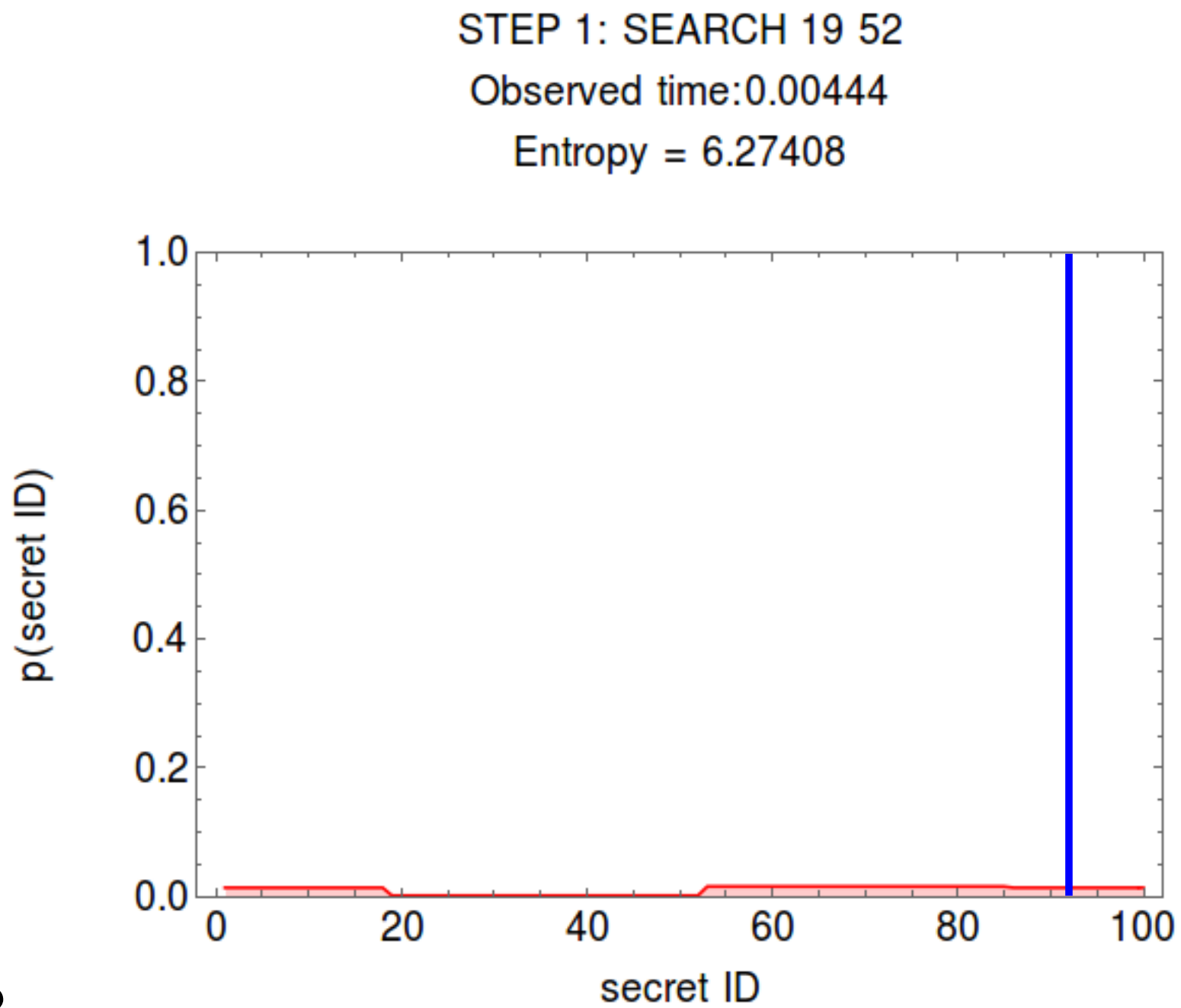
Writes to log file depending on whether

$$ID_{res} \in [minID, maxID]$$

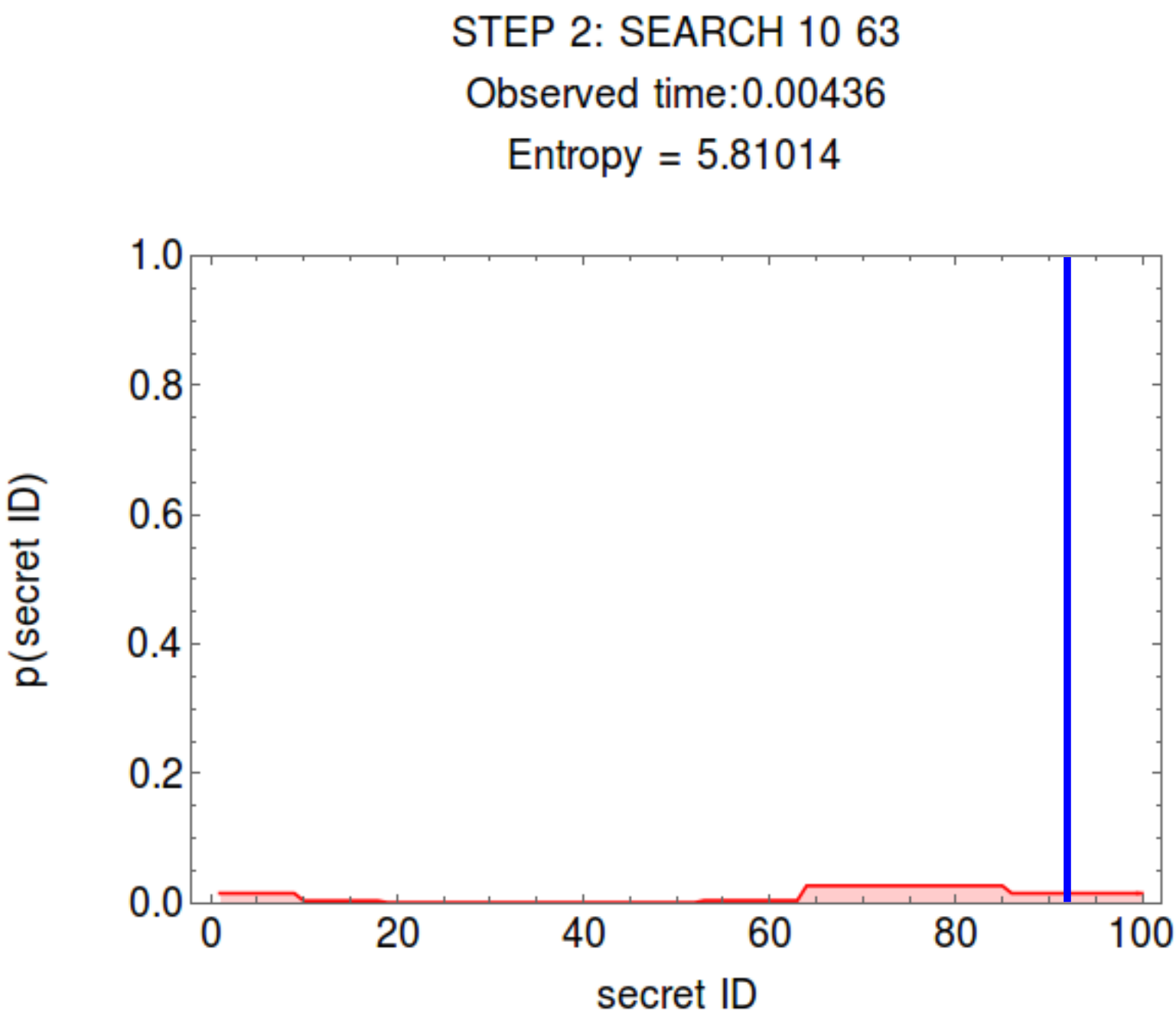
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



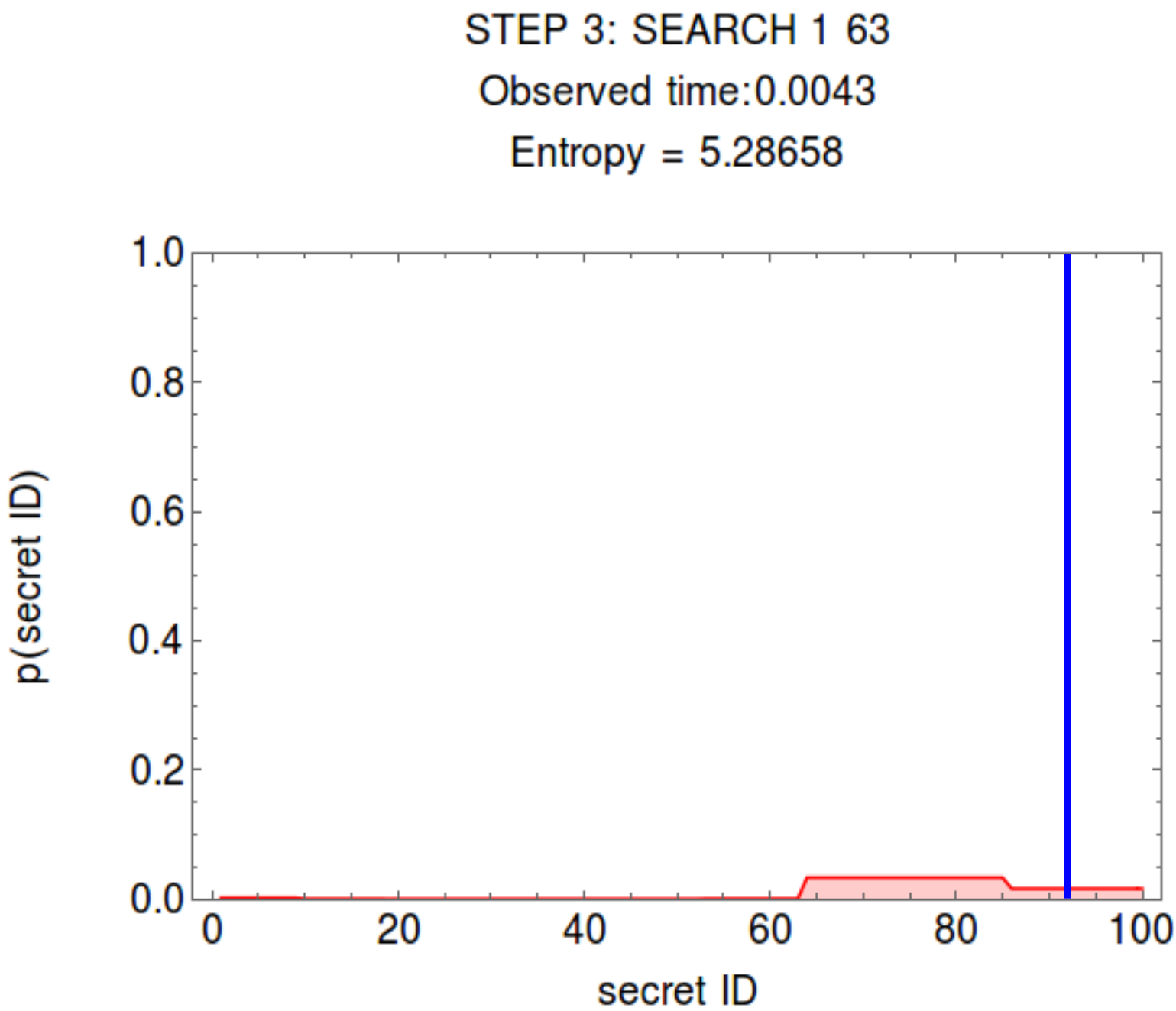
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



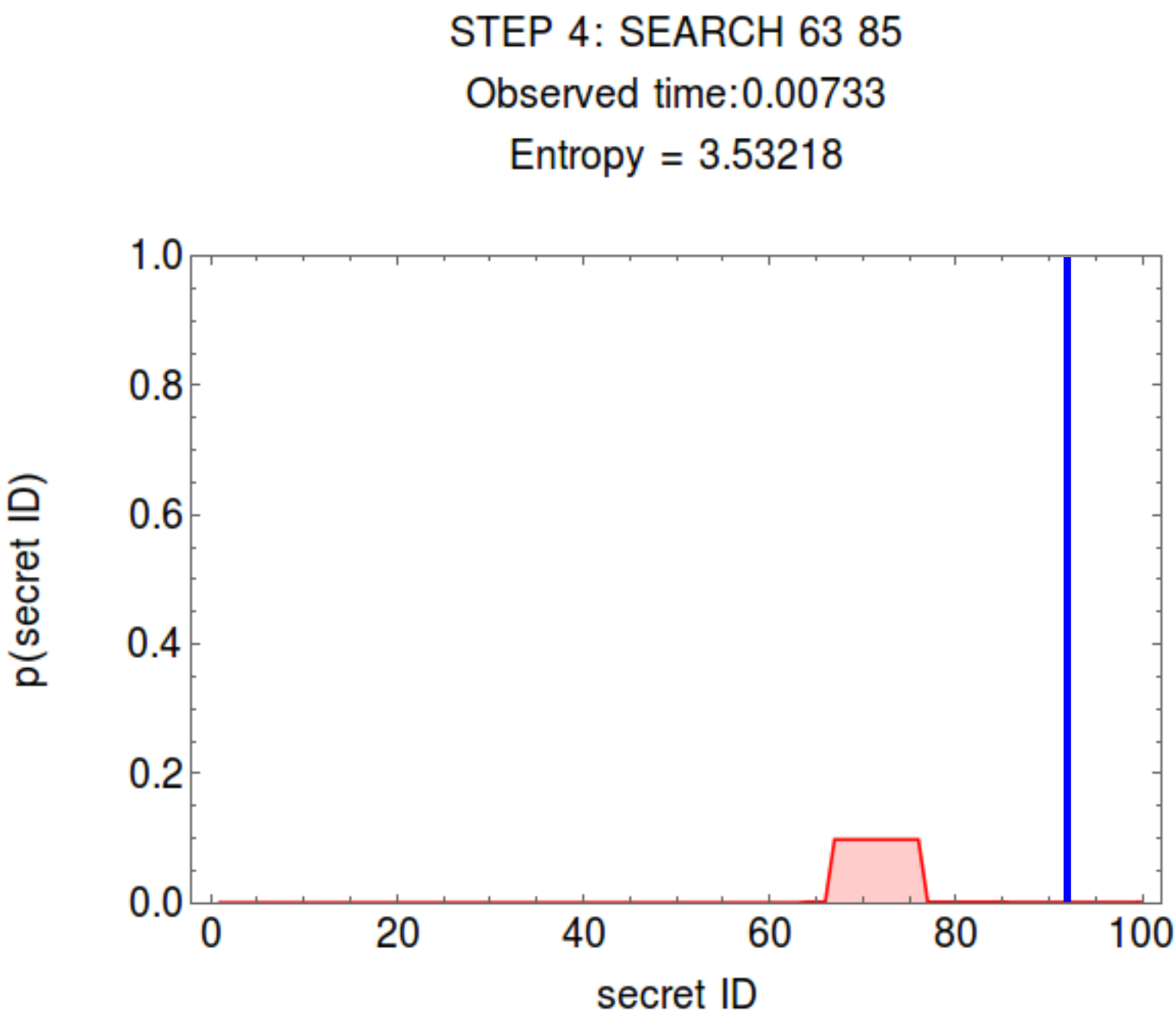
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



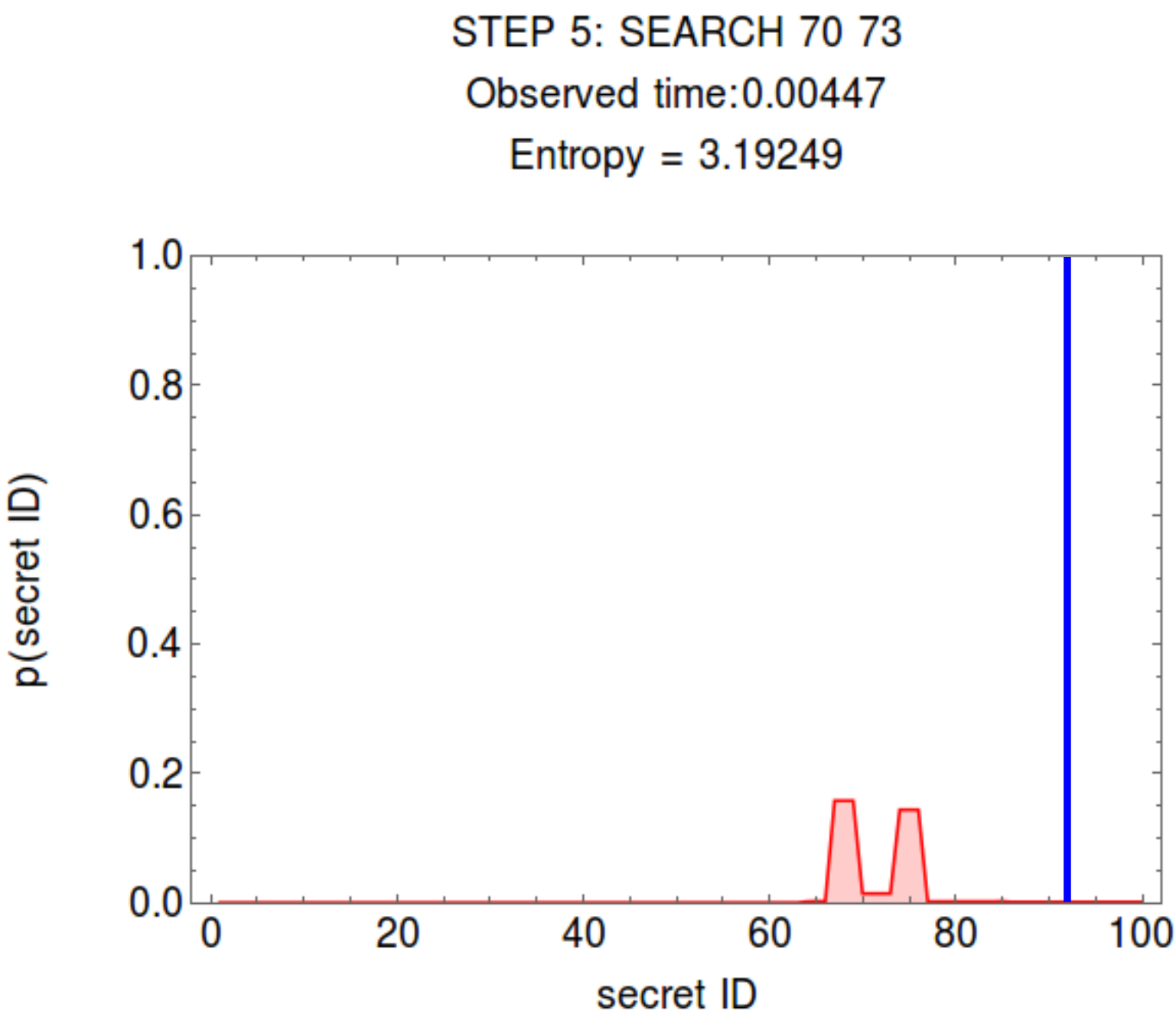
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



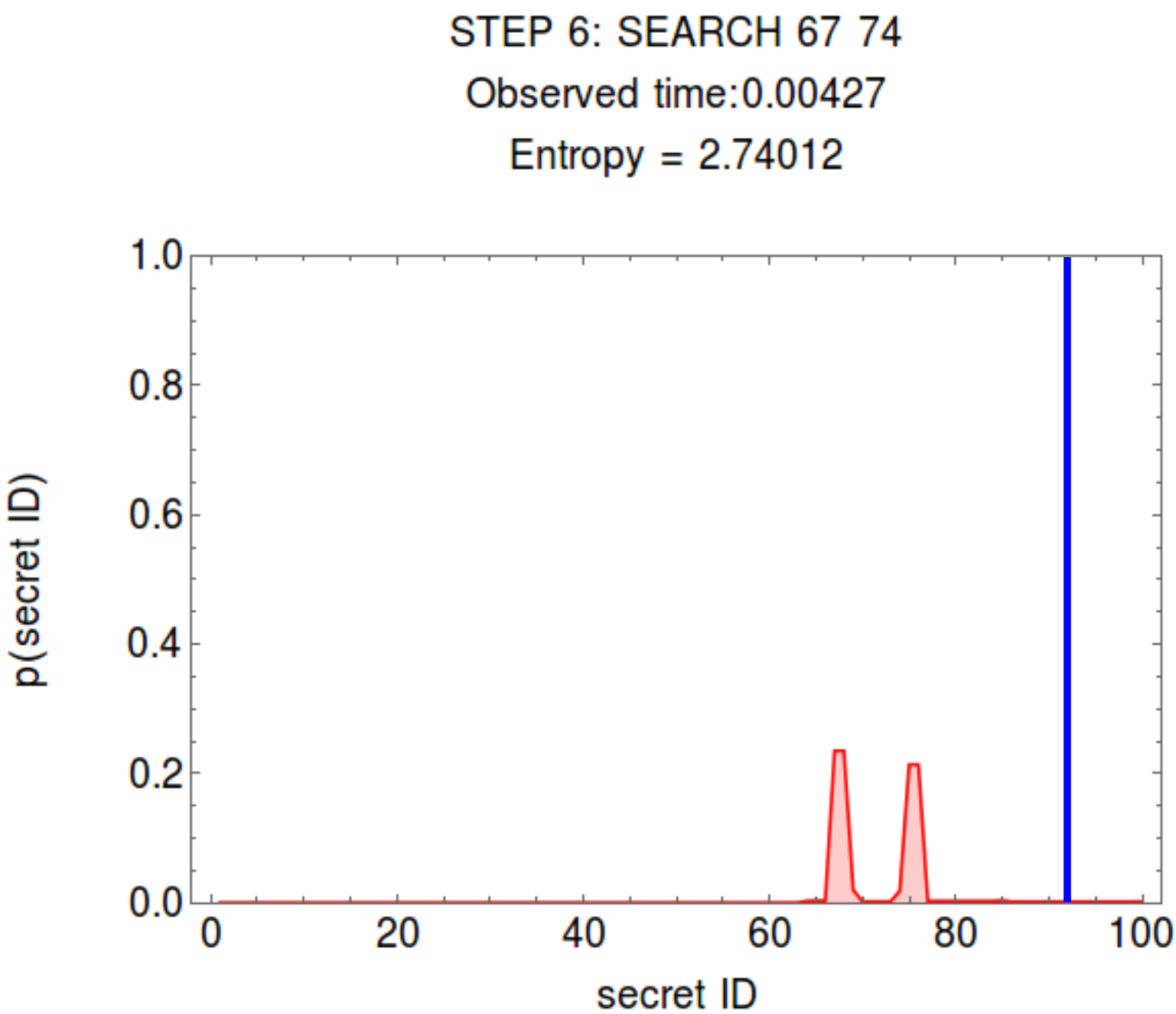
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



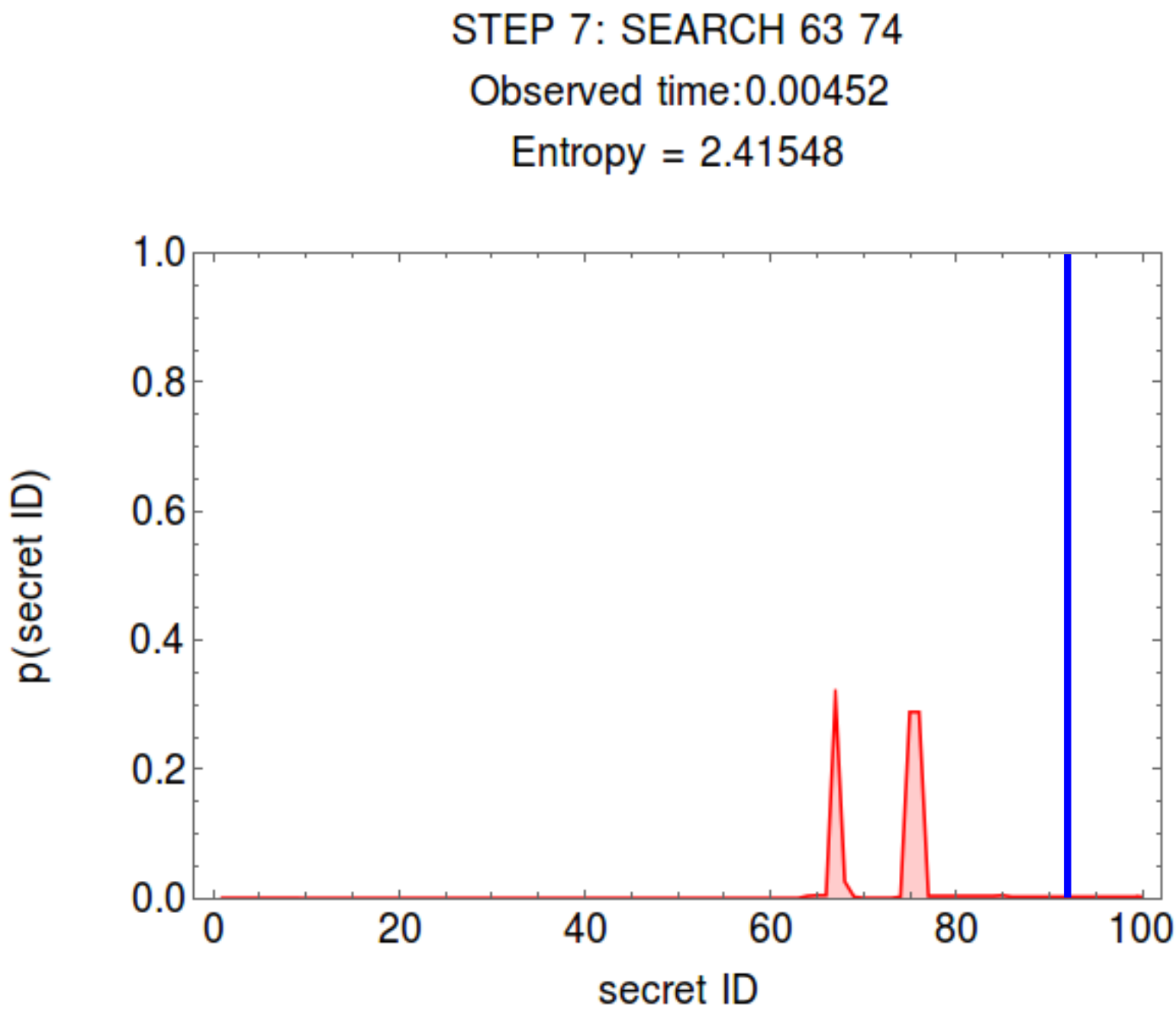
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



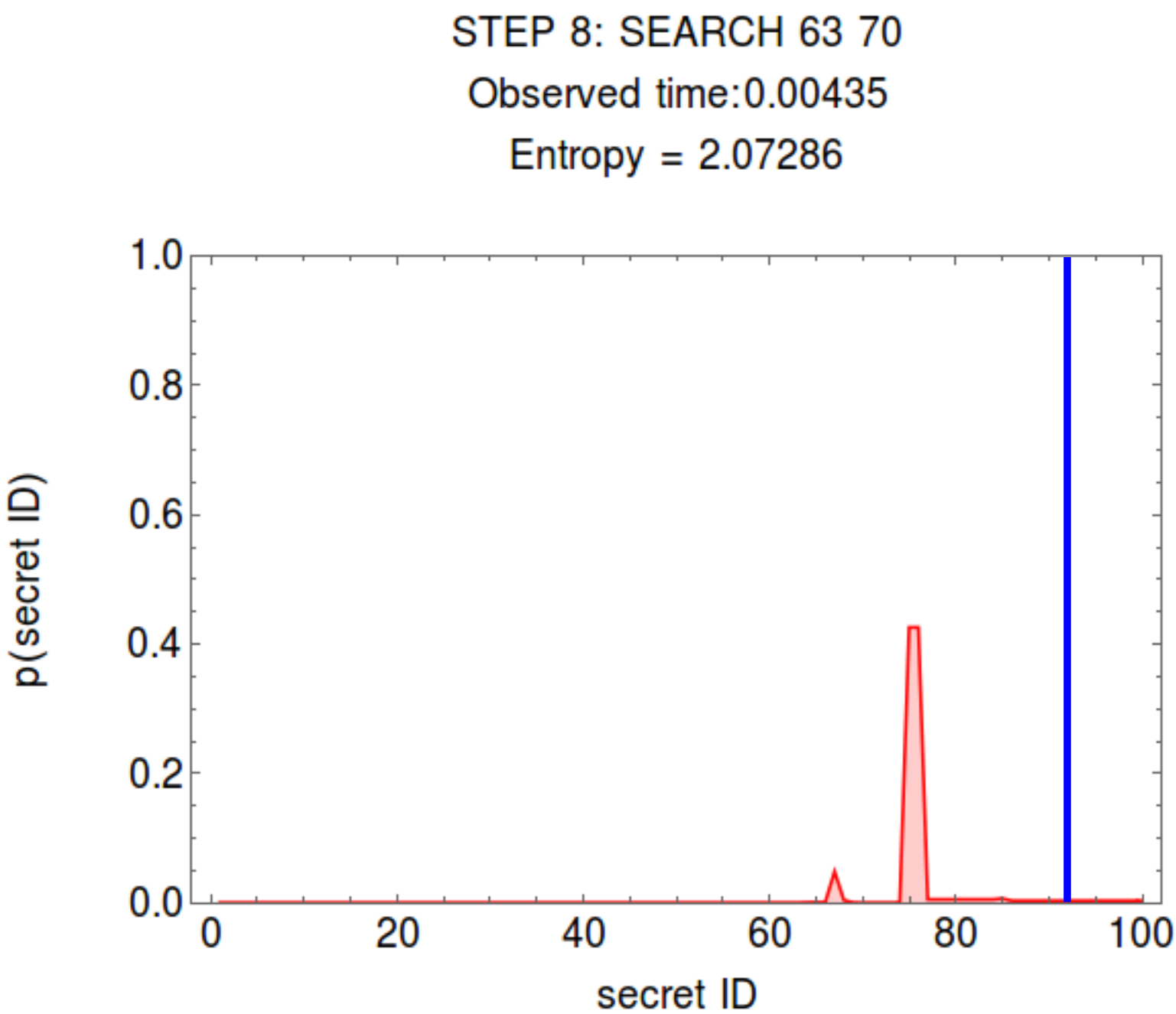
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



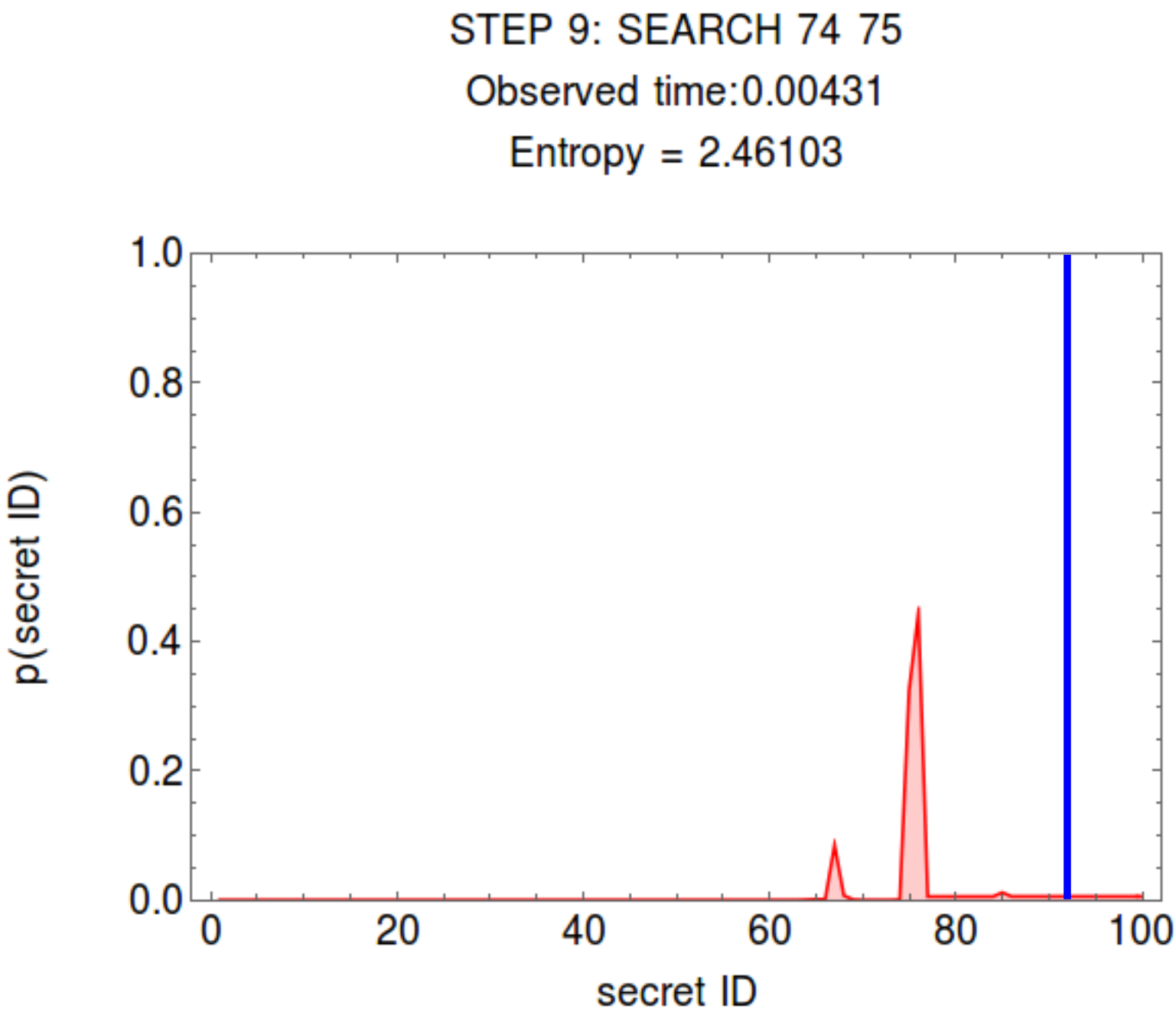
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



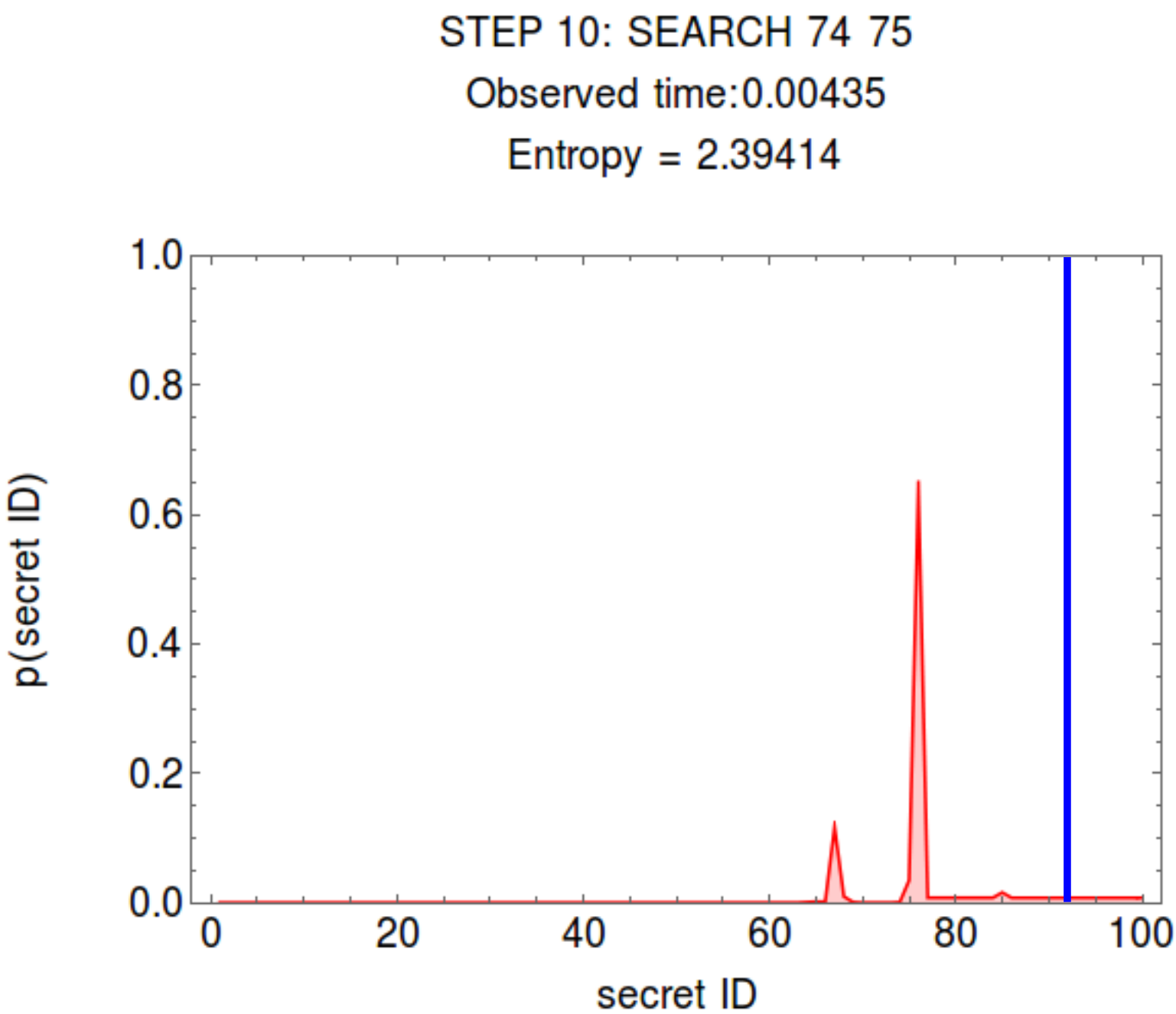
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



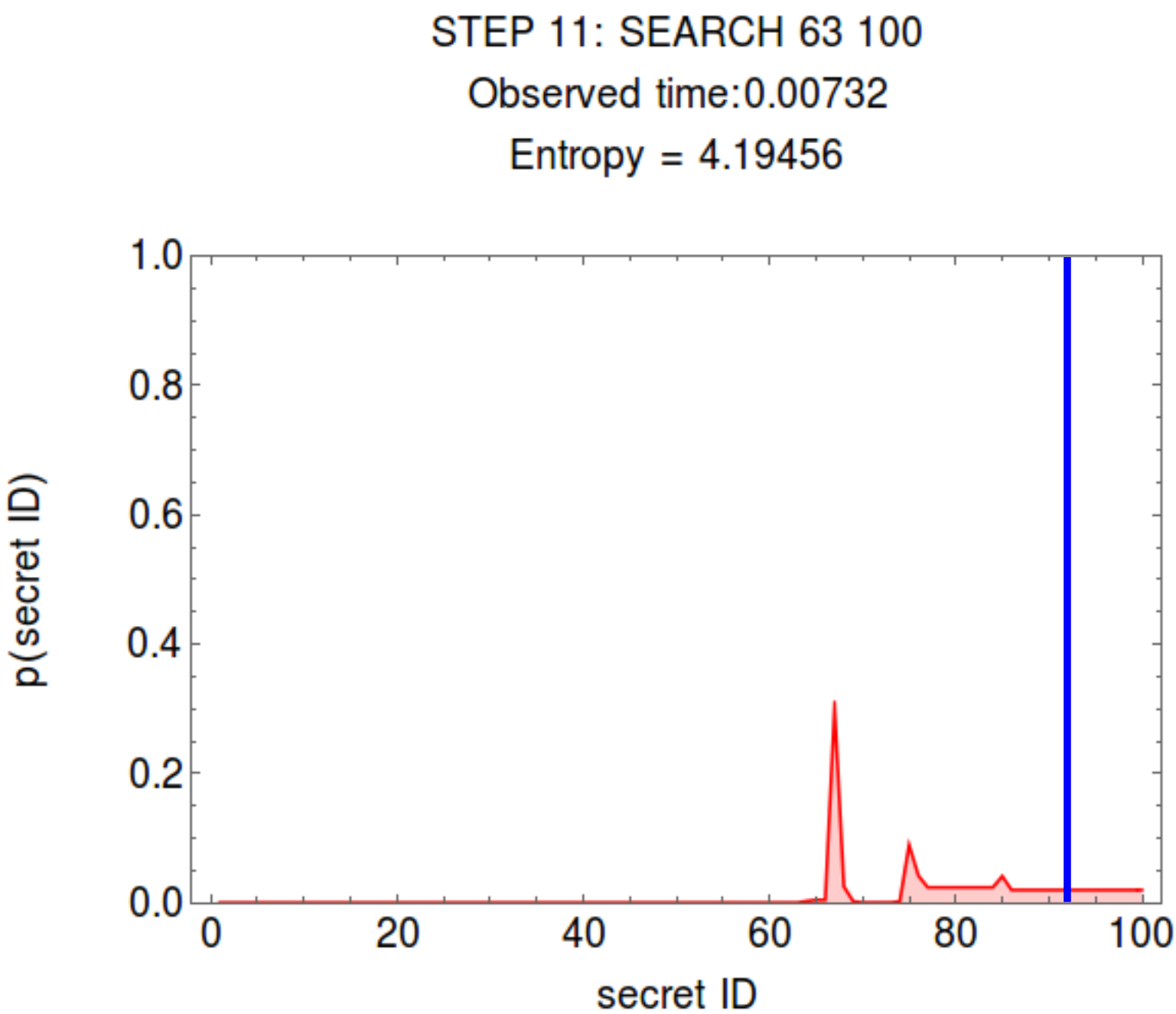
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



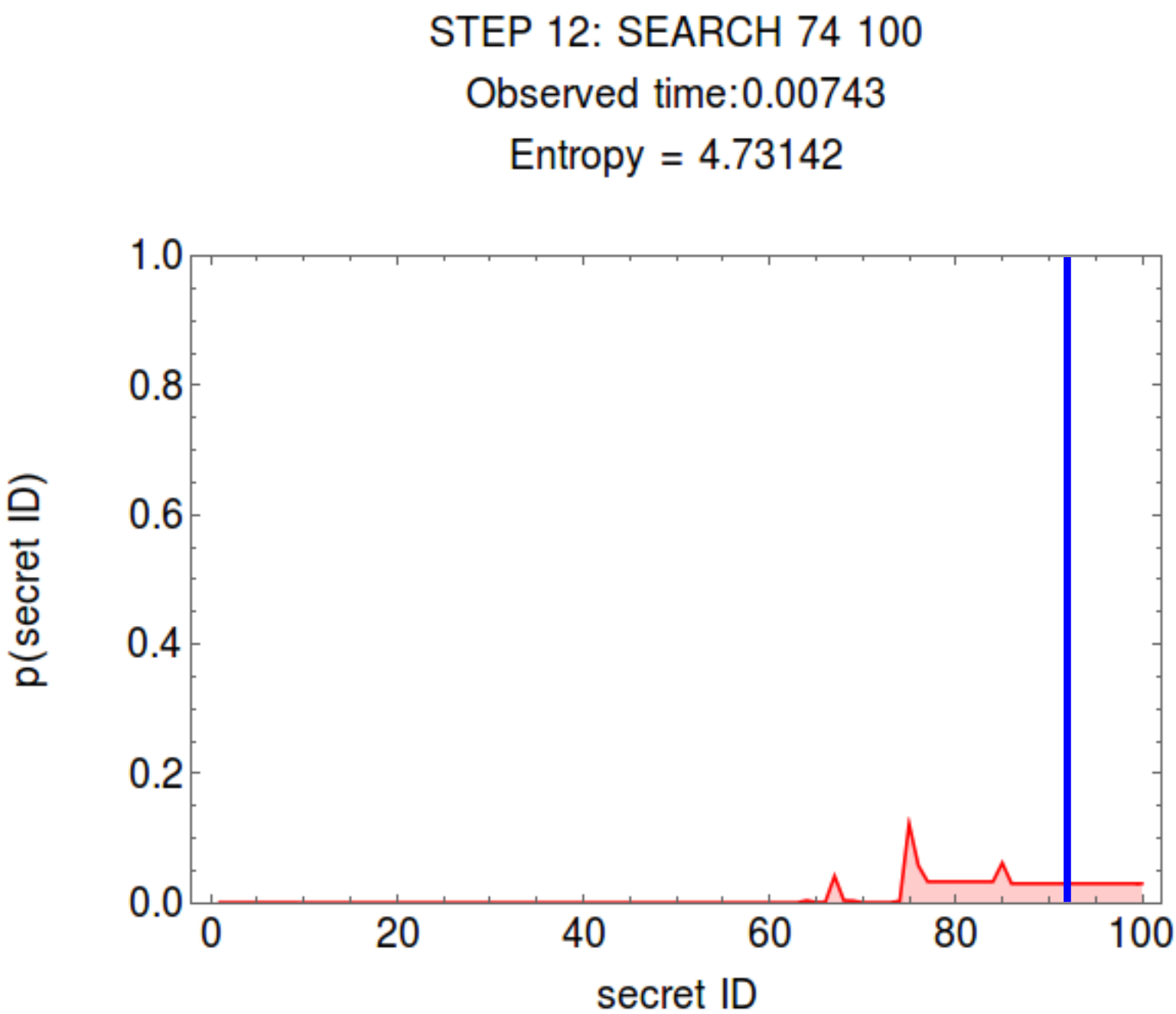
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



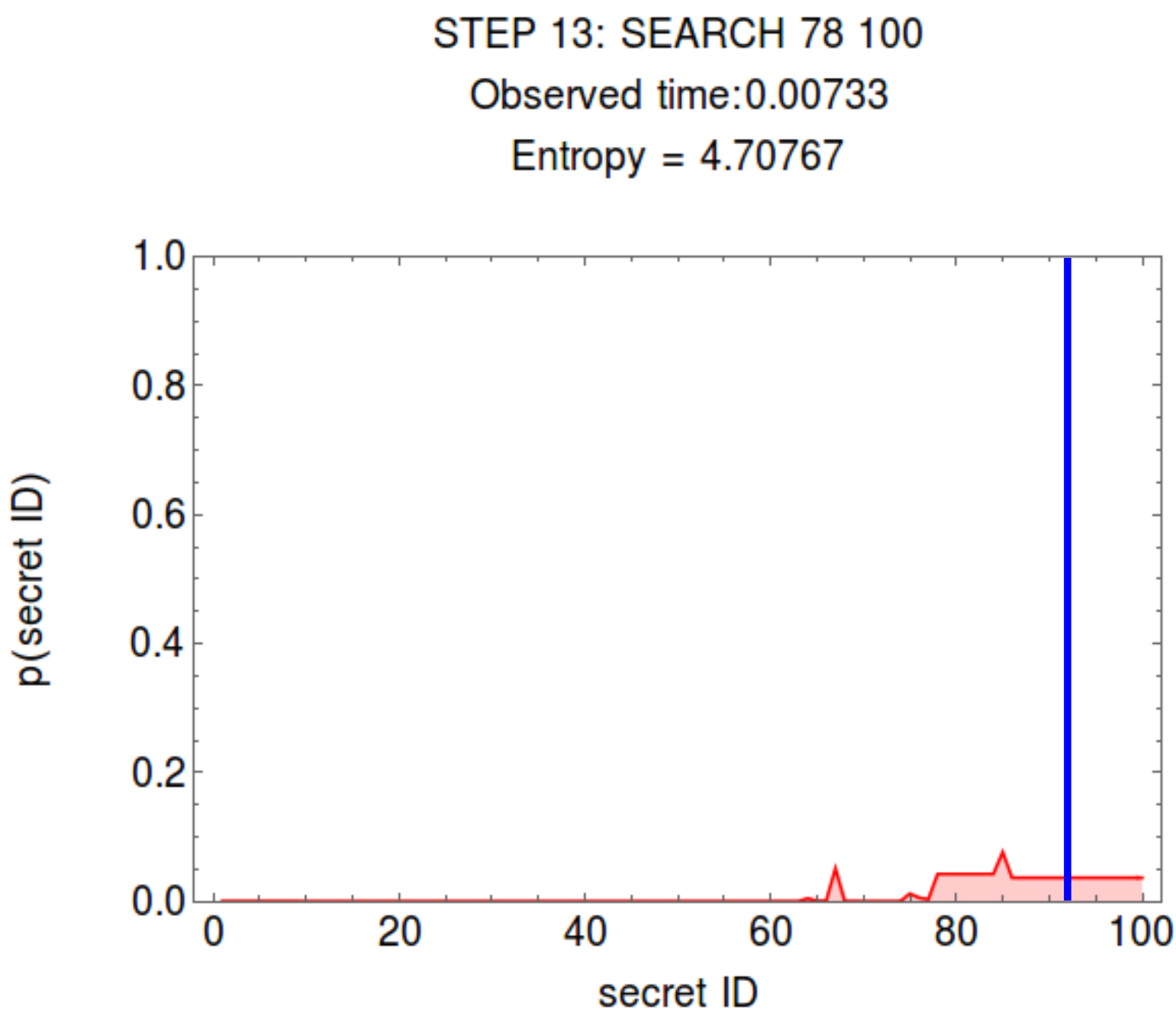
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



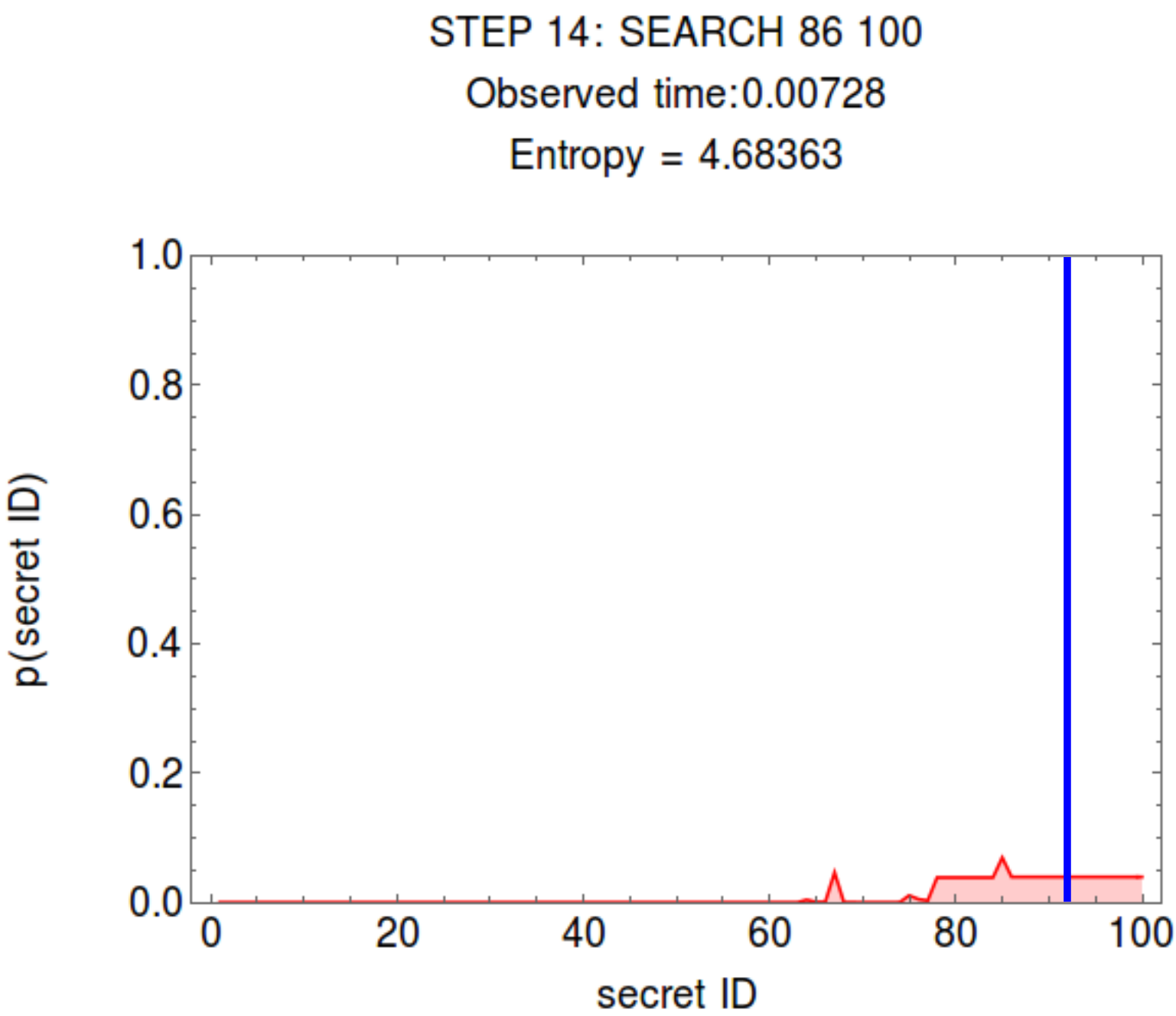
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



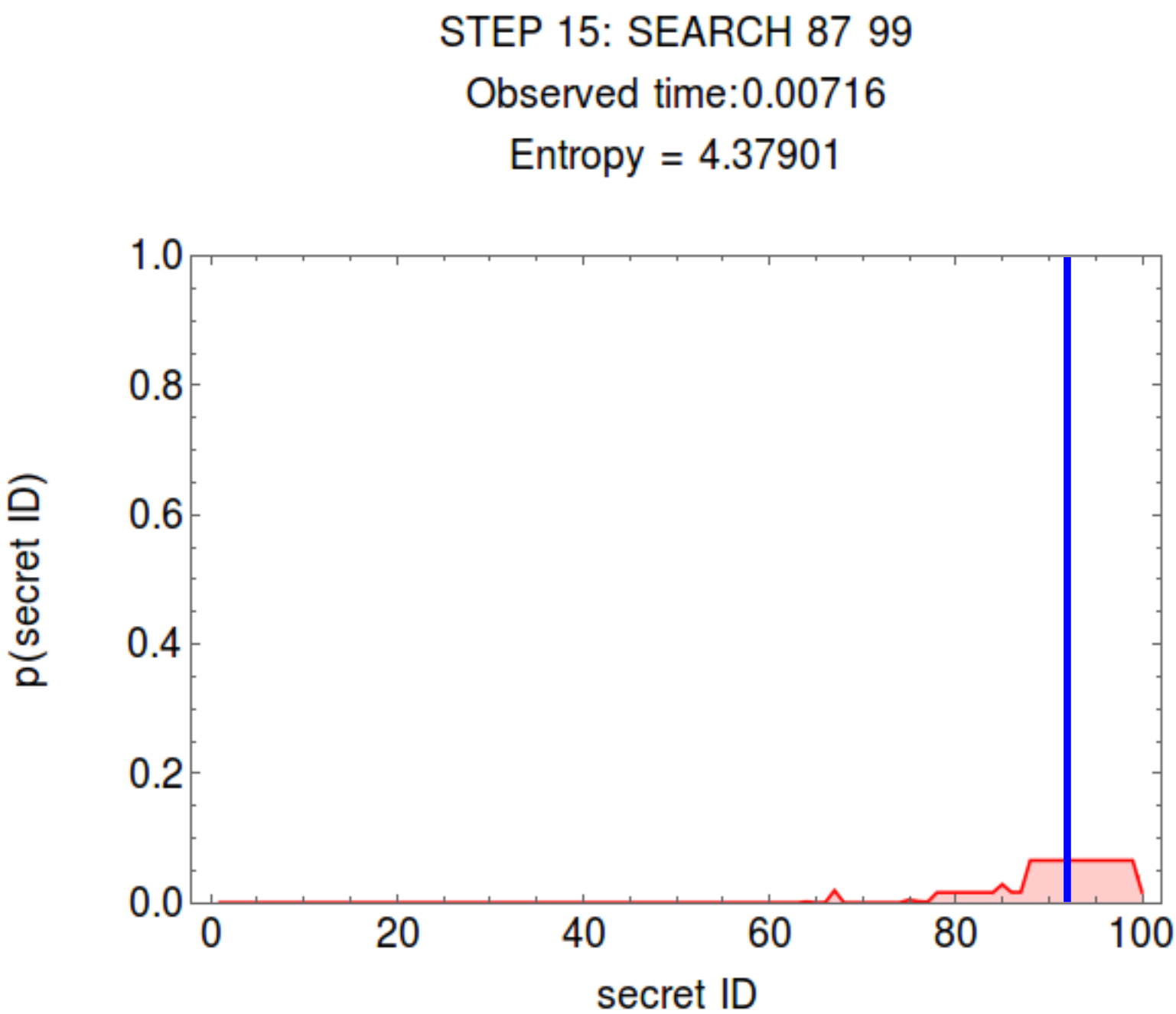
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



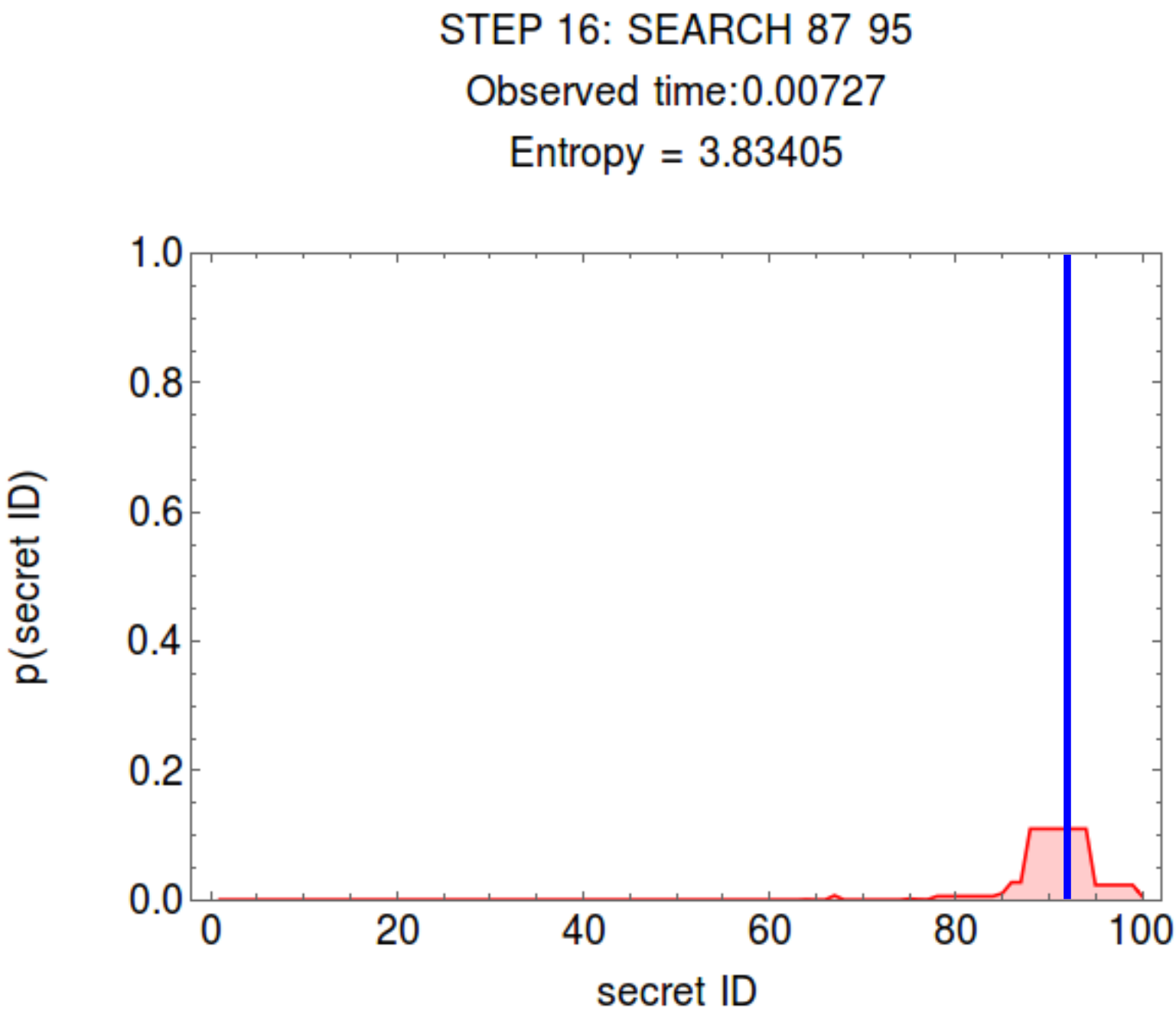
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



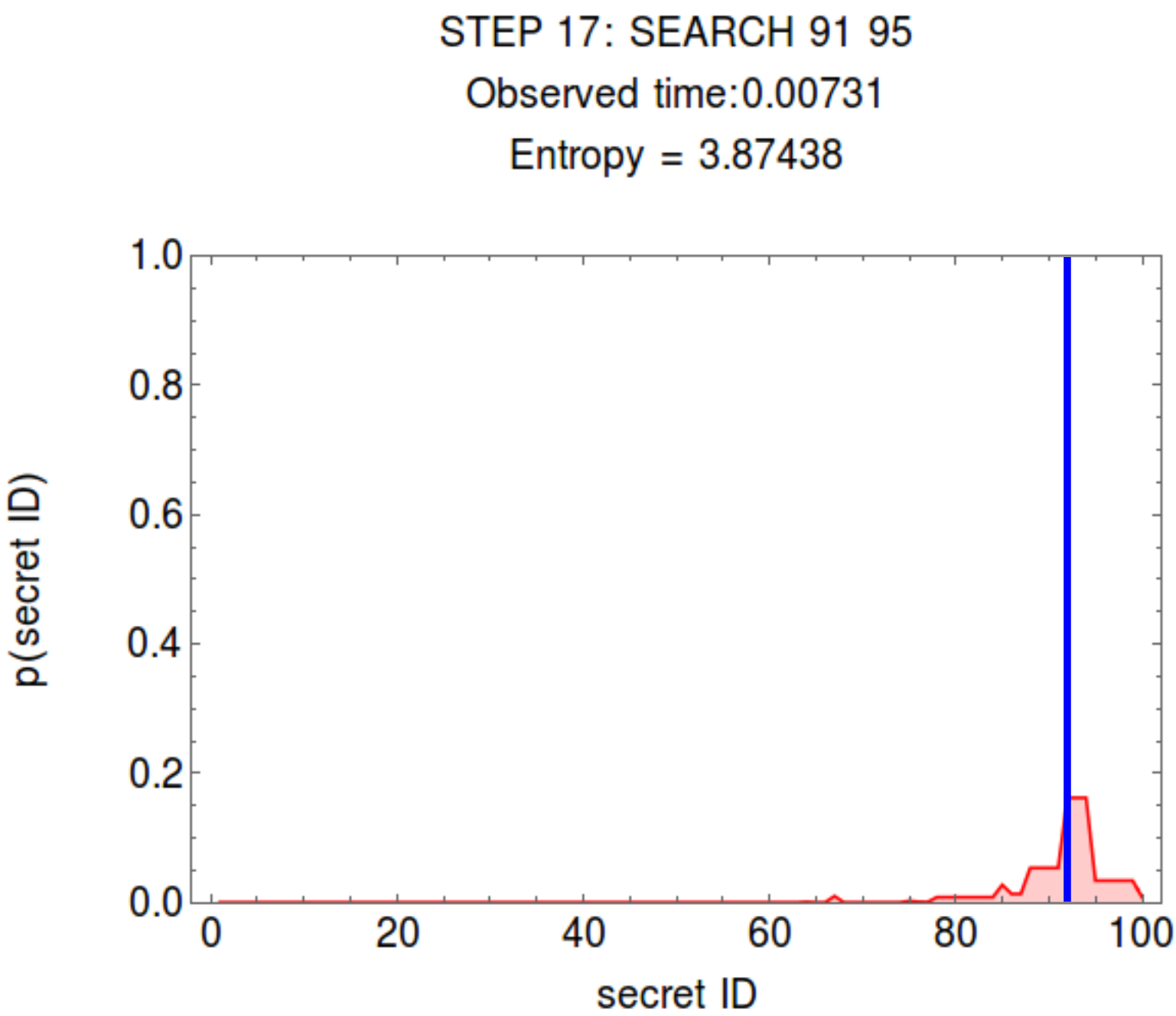
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



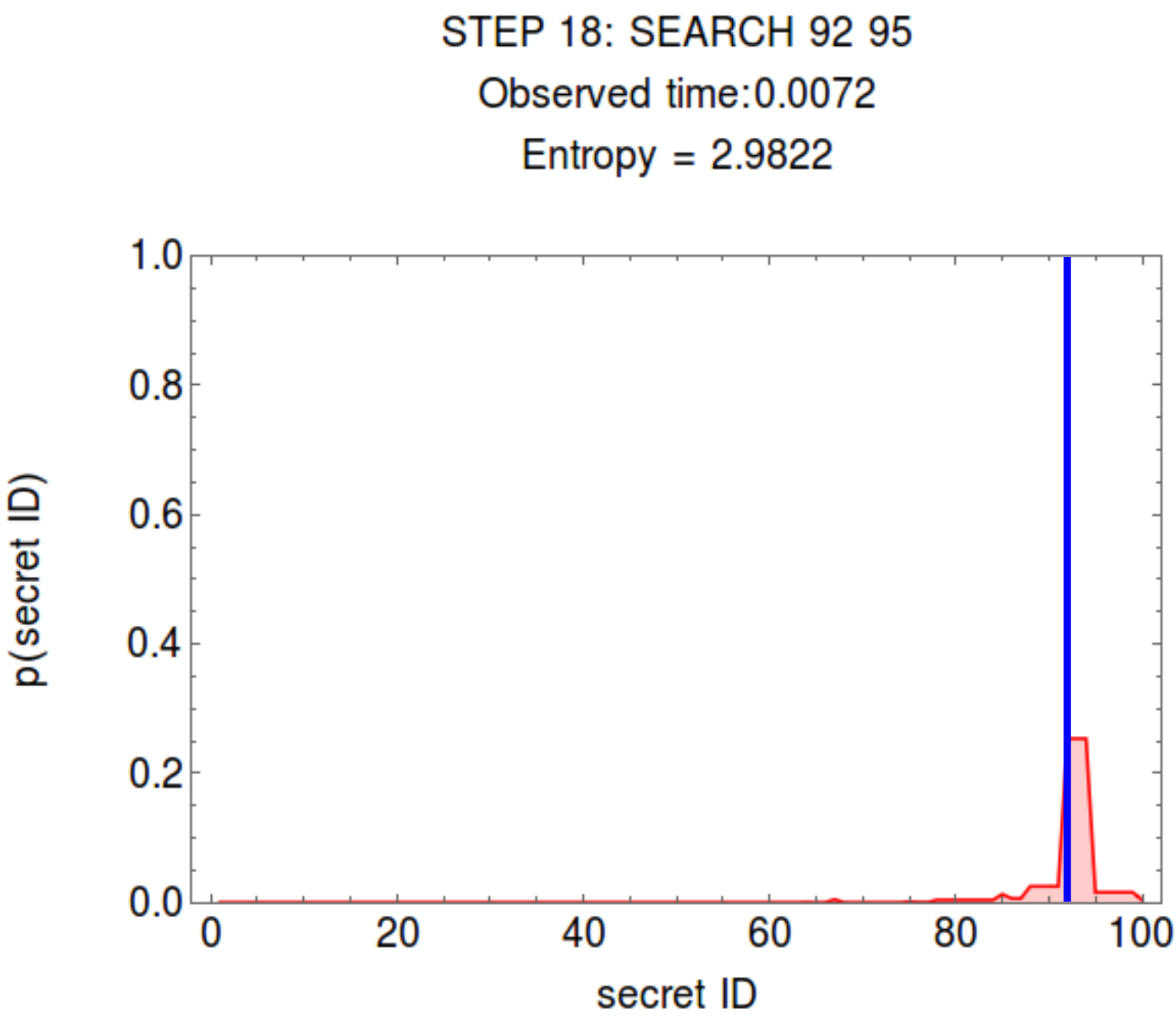
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



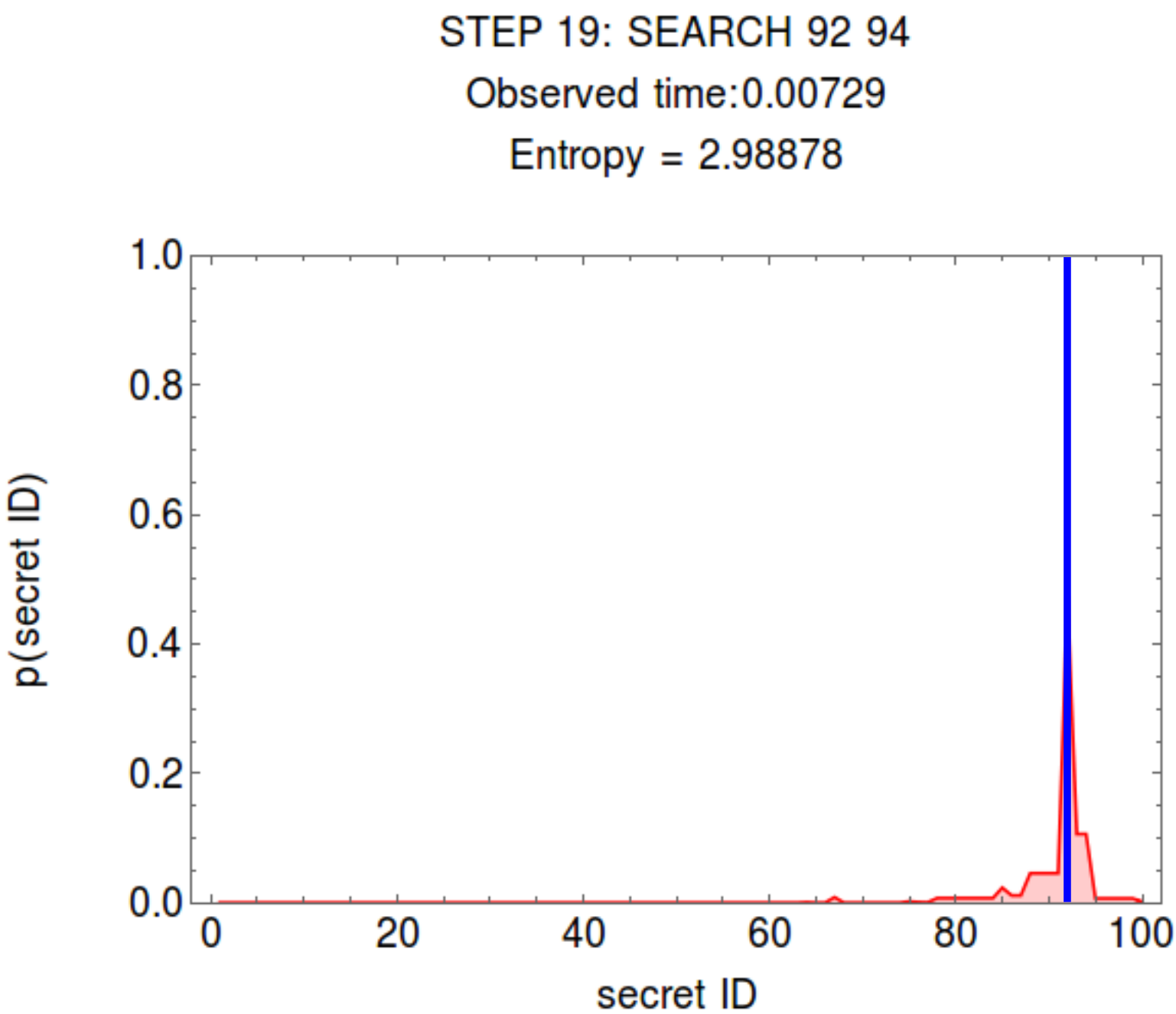
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



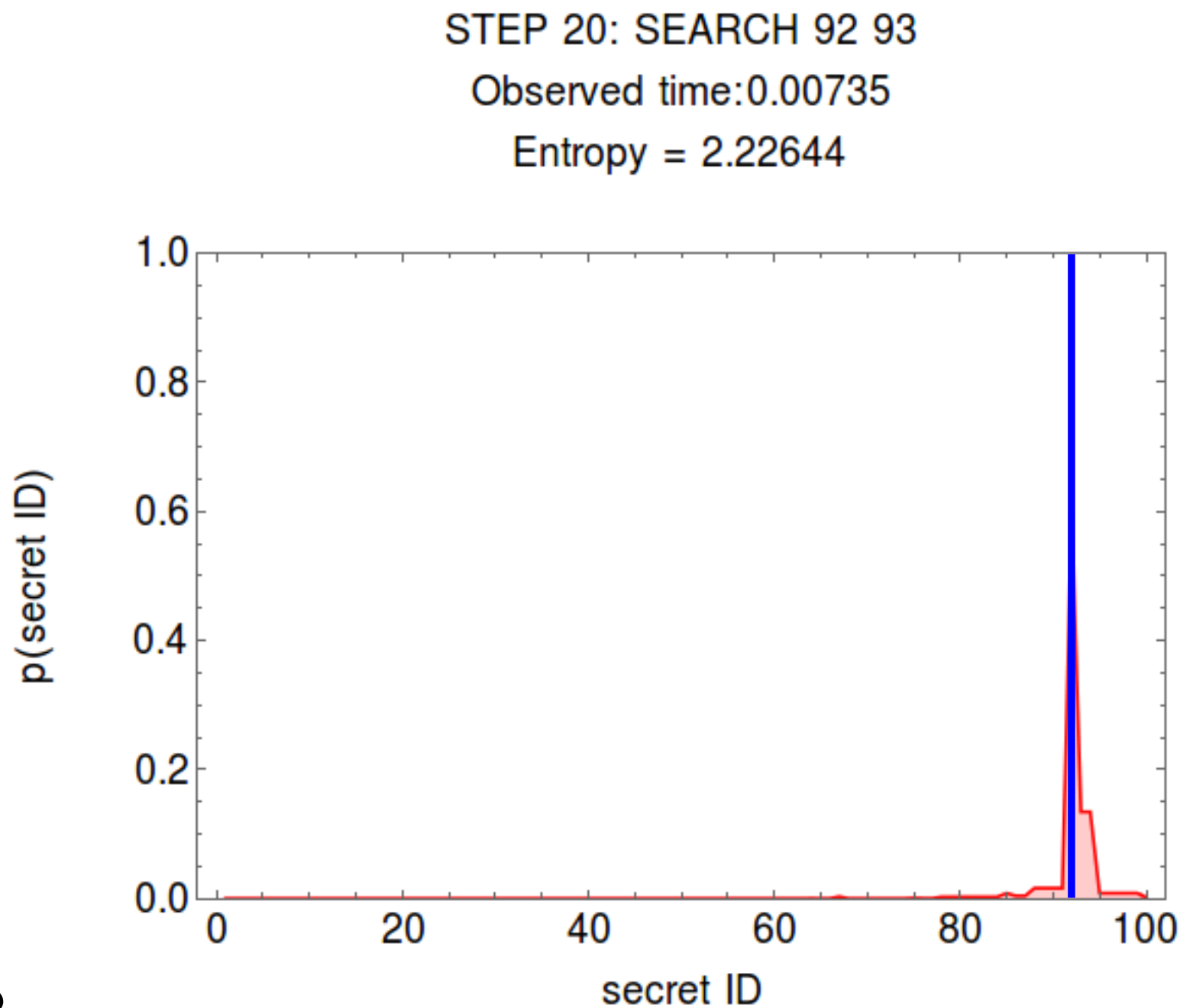
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



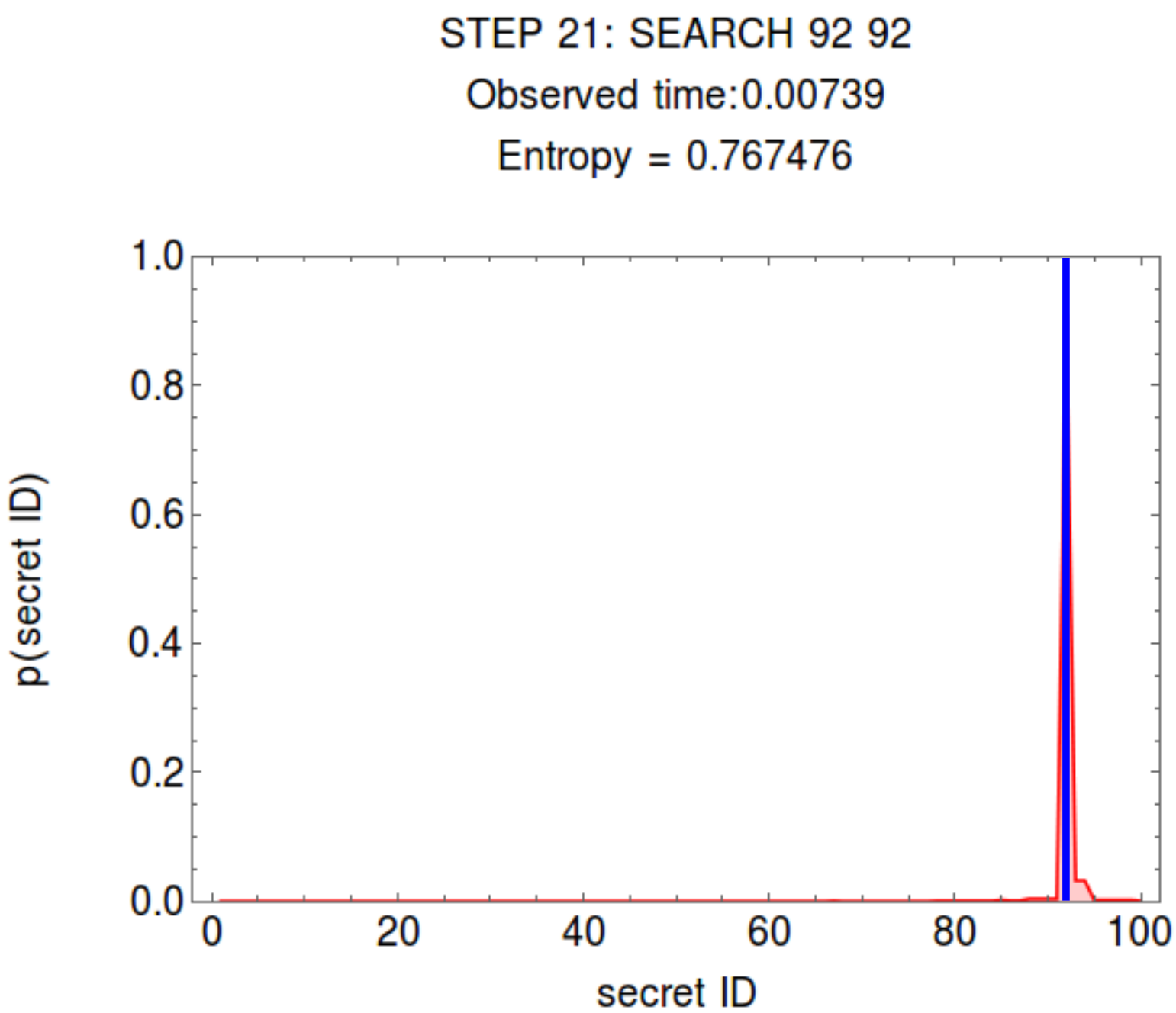
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



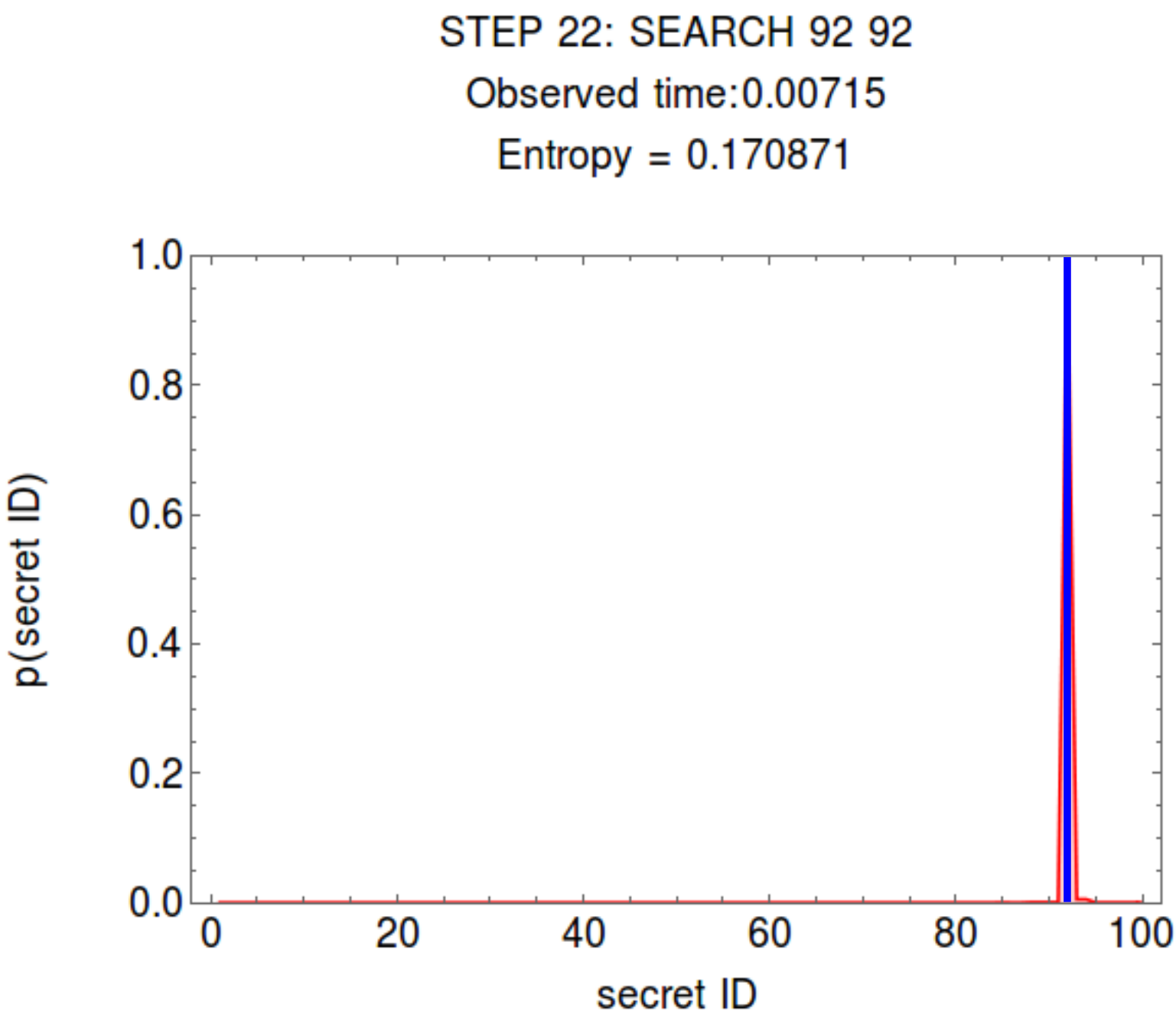
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



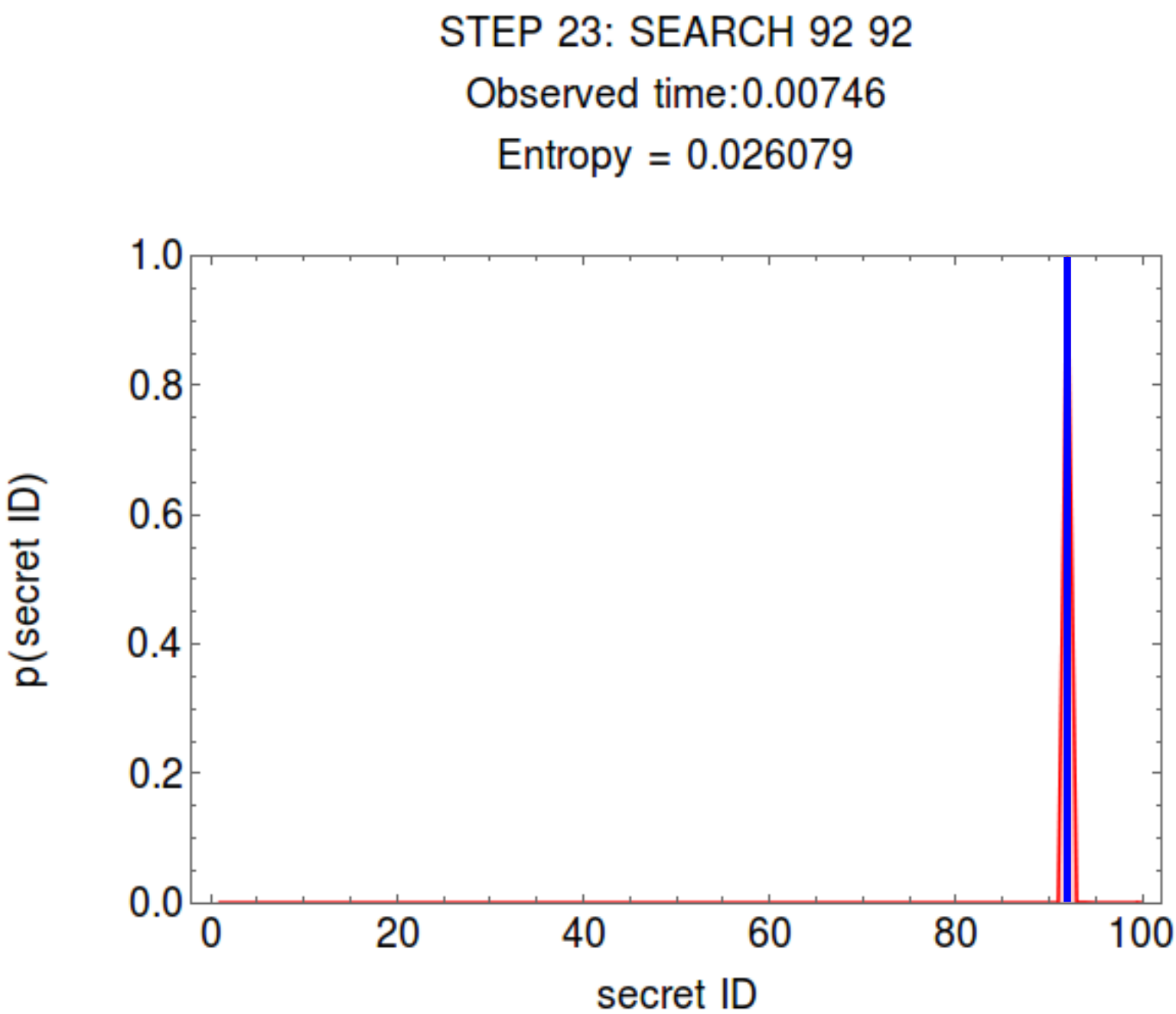
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



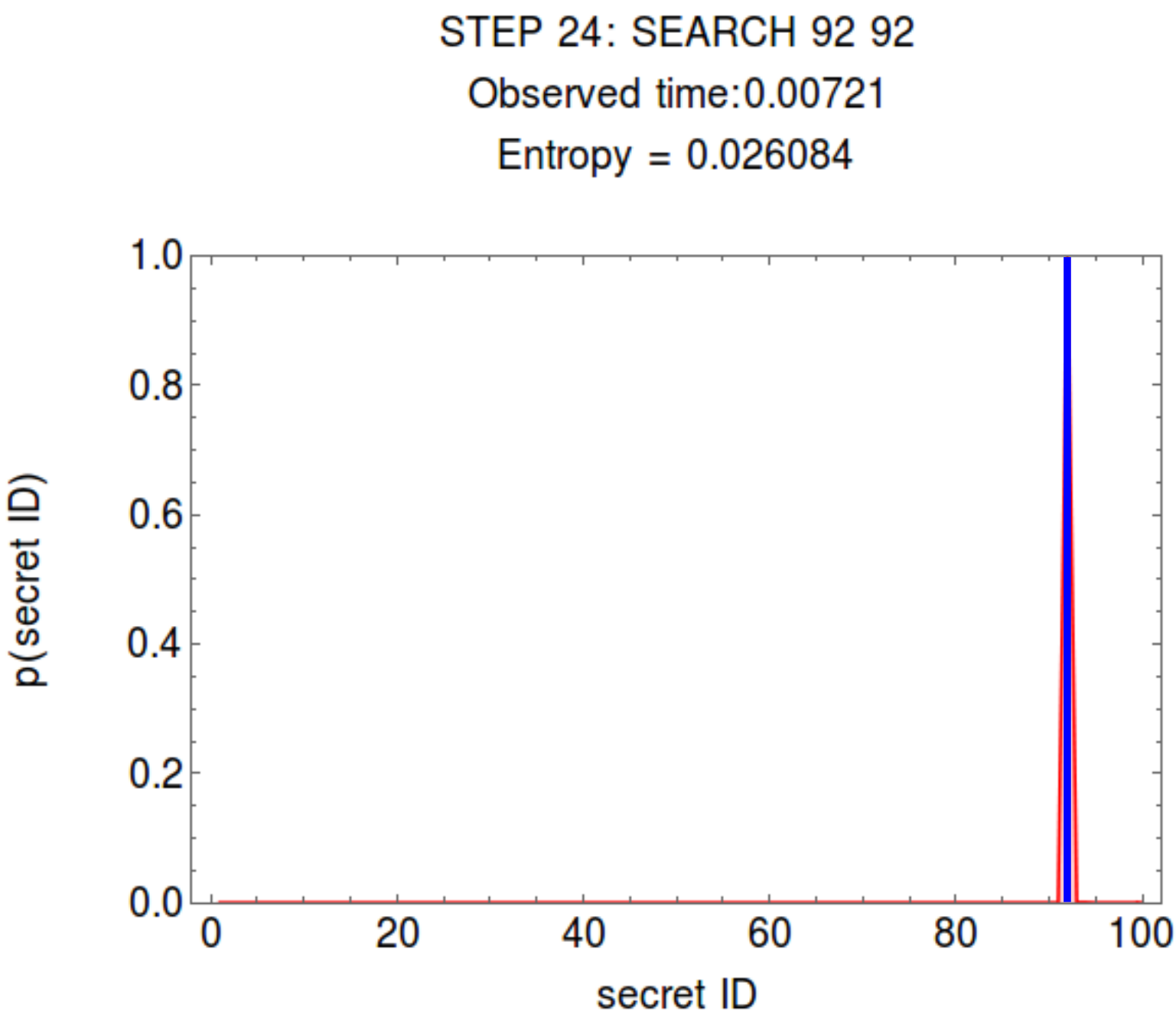
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$



ID Range	# Employees	Offline Analysis	Attack	
			time (m)	# steps
1-100	3	57s	2m38s	25
1-10000	4	2m21s	2m43s	45
1-10000	5	6m30s	3m08s	48
1-10000	10	42m09s	4m31s	77

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: california

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: california

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: harmonicas

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: harmonicas

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: harmonicas

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: harmonicas

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: harmonicas

Segment Oracle Side Channel Attacks

```
Boolean compare(String pw, String input){
 for(int i = 0; i < pw.length, i++)
  if(pw[i] != input[i])
   return false;
 return true;
}
```

pw: harveymudd

input: harmonicas

Attacker can brute-force individual characters!

String Analysis for Side Channels with Segmented Oracles [IEEE Foundations of Software Engineering 2016]

Attack Synthesis vs. PW Checker

Attack Synthesis vs. PW Checker

Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
ε	csja	ciub	ciqi	ciqa
fzgk	cnte	ciij	ciqz	ciqa
daaz	cwcs	cimq	ciqz	ciqg
zgap	ctdo	citz	ciqu	ciqa
uaak	cved	ciqz	ciqz	
bnza	cvfo	ciaz	ciqc	
ecjq	ceyu	ciok	ciqz	
zmna	ciil	cida	ciqe	
tzar		cijw	ciqr	
zmna		⋮	⋮	
maau		cigz	ciqk	
vzsc		cisu	ciqd	
qyas		cisp	ciqd	
asvr		cine	ciqr	
cmxq		ciqk	ciqz	

Attack Synthesis vs. PW Checker

Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
ε	csja	ciub	ciqi	ciqa
fzgk	cnte	ciij	ciqz	ciqa
daaz	cwcs	cimq	ciqz	ciqg
zgap	ctdo	citz	ciqu	ciqa
uaak	cved	ciqz	ciqz	
bnza	cvfo	ciaz	ciqc	
ecjq	ceyu	ciok	ciqz	
zmna	ciil	cida	ciqe	
tzar		cijw	ciqr	
zmna		⋮	⋮	
maau		cigz	ciqk	
vzsc		cisu	ciqd	
qyas		cisp	ciqd	
asvr		cine	ciqr	
cmxq		ciqk	ciqz	

Attack Synthesis vs. PW Checker

Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
ε	csja	ciub	ciqi	ciqa
fzgk	cnte	ciij	ciqz	ciqa
daaz	cwcs	cimq	ciqz	ciqg
zgap	ctdo	citz	ciqu	ciqa
uaak	cved	ciqz	ciqz	
bnza	cvfo	ciaz	ciqc	
ecjq	ceyu	ciok	ciqz	
zmna	ciil	cida	ciqe	
tzar		cijw	ciqr	
zmna		⋮	⋮	
maau		cigz	ciqk	
vzsc		cisu	ciqd	
qyas		cisp	ciqd	
asvr		cine	ciqr	
cmxq		ciqk	ciqz	

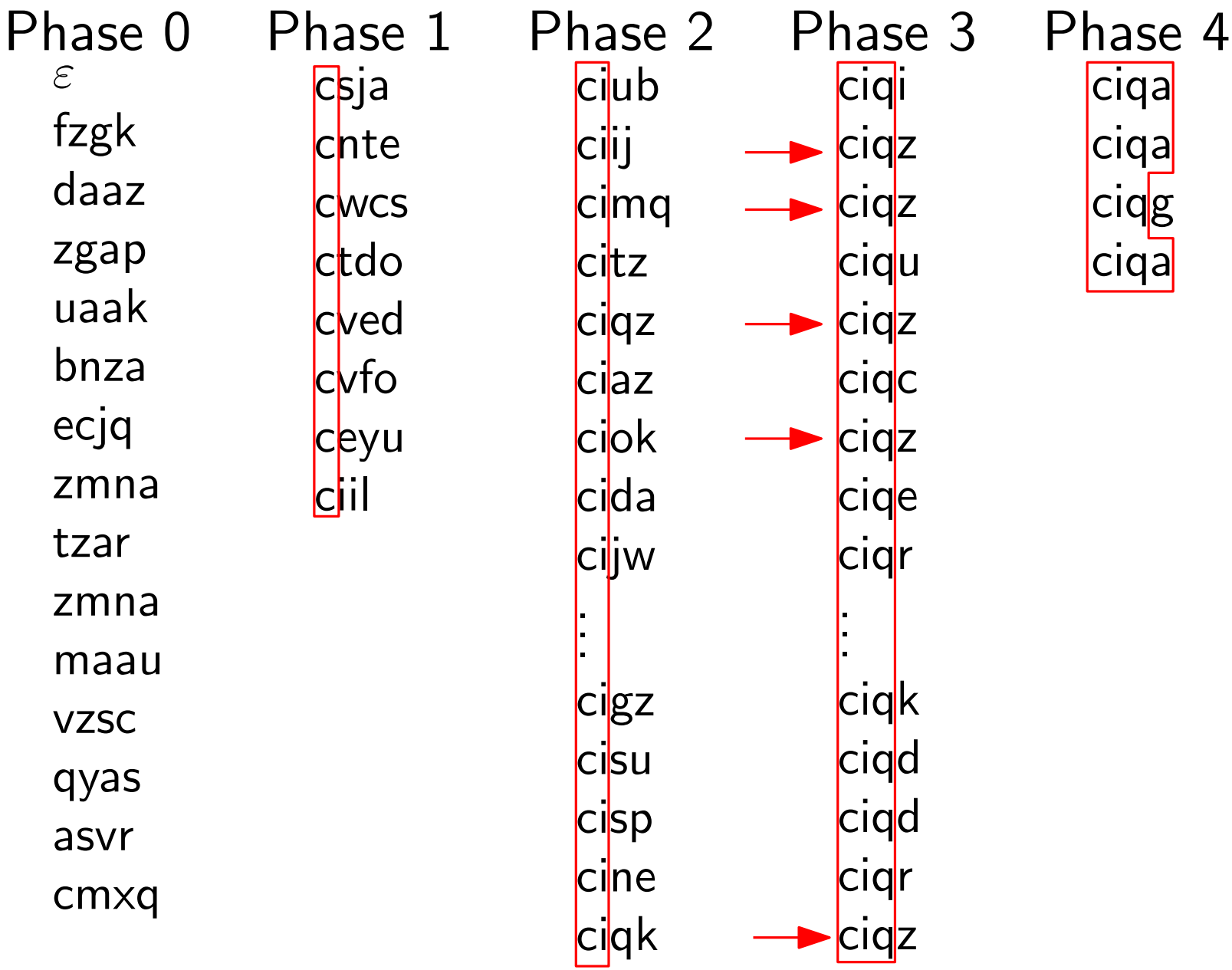
Attack Synthesis vs. PW Checker

Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
ε	csja	ciub	ciqi	ciqa
fzgk	cnte	ciij	ciqz	ciqa
daaz	cwcs	cimq	ciqz	ciqg
zgap	ctdo	citz	ciqu	ciqa
uaak	cved	ciqz	ciqz	
bnza	cvfo	ciaz	ciqc	
ecjq	ceyu	ciok	ciqz	
zmna	ciil	cida	ciqe	
tzar		cijw	ciqr	
zmna		⋮	⋮	
maau		cigz	ciqk	
vzsc		cisu	ciqd	
qyas		cisp	ciqd	
asvr		cine	ciqr	
cmxq		ciqk	ciqz	

Attack Synthesis vs. PW Checker

Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
ε	csja	ciub	ciqi	ciqa
fzgk	cnte	ciij	ciqz	ciqa
daaz	cwcs	cimq	ciqz	ciqg
zgap	ctdo	citz	ciqu	ciqa
uaak	cved	ciqz	ciqz	
bnza	cvfo	ciaz	ciqc	
ecjq	ceyu	ciok	ciqz	
zmna	ciil	cida	ciqe	
tzar		cijw	ciqr	
zmna		⋮	⋮	
maau		cigz	ciqk	
vzsc		cisu	ciqd	
qyas		cisp	ciqd	
asvr		cine	ciqr	
cmxq		ciqk	ciqz	

Attack Synthesis vs. PW Checker



Attack Synthesis vs. PW Checker

Offline time: 9m54s

Attack time: 19m03s

Attack steps: 79

```
int memcmp(s1, s2, n)
  CONST VOID *s1;
  CONST VOID *s2;
  size_t n;
  {
    unsigned char u1, u2;
    for ( ; n-- ; s1++, s2++) {
      u1 = * (unsigned char *) s1;
      u2 = * (unsigned char *) s2;
        if ( u1 != u2) {
          return (u1-u2);
        }
  }
  return 0;
}
```

```
int memcmp(s1, s2, n)
  CONST VOID *s1;
  CONST VOID *s2;
  size_t n;
  {
    unsigned char u1, u2;
    for ( ; n-- ; s1++, s2++) {
      u1 = * (unsigned char *) s1;
      u2 = * (unsigned char *) s2;
        if ( u1 != u2) {
          return (u1-u2);
        }
  }
  return 0;
}
```

Xbox OS, HMAC signatures compared with memcmp!
Allowed insecure kernel downgrade.

Space/Time Analysis for Cybersecurity Benchmark

Benchmark		Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds)			
							S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

	Benchmark	Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds)			
							S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

Benchmark		Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds) S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

Benchmark		Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds)			
							S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

Benchmark		Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds)			
							S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

Benchmark		Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds)			
							S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

Benchmark		Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds)			
							S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

Benchmark		Dim(H)	$\lvert\mathbb{H}\rvert$	$\lvert\Phi\rvert$	$\lvert\mathcal{T}\rvert$	Vuln?	Offline Phase Time (seconds)			
							S.E.	Noise Est.	Merging	Total
1	STAC-1(nv)	1	$2^{\{8,16,24,31\}}$	2	1	no	0.57	22.28	0.81	23.67
2	STAC-3(nv)	1	$2^{\{8,16,24,31\}}$	6	3	no	0.64	36.18	4.89	41.72
3	STAC-1(v)	1	$2^{\{8,16,24,31\}}$	2	2	yes	0.56	31.52	0.48	32.58
4	STAC-3(v)	1	$2^{\{8,16,24,31\}}$	6	4	yes	0.57	34.09	5.17	39.85
5	STAC-11A(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.58	25.65	1.32	27.56
6	STAC-11B(v)	1	$2^{\{8,16,24,31\}}$	3	2	yes	0.57	26.63	1.29	28.50
7	STAC-4(v)	1	26	10	2	yes	0.73	14.79	7.10	22.63
8	STAC-4(v)	2	702	27	3	yes	1.19	44.52	2.28	48.01
9	STAC-4(v)	3	18278	55	5	yes	2.67	100.55	64.94	168.17
10	STAC-12(v)	1	26	17	4	yes	0.94	26.30	18.57	45.83
11	STAC-12(v)	2	702	39	5	yes	0.99	57.46	48.67	107.13
12	STAC-12(v)	3	18278	77	6	yes	1.62	125.49	132.63	259.76
13	STAC-12(v)	4	475254	149	7	yes	3.06	258.48	293.57	555.13

Space/Time Analysis for Cybersecurity Benchmark

Number of Attack Steps

Attack Synthesis Time

STAC–1

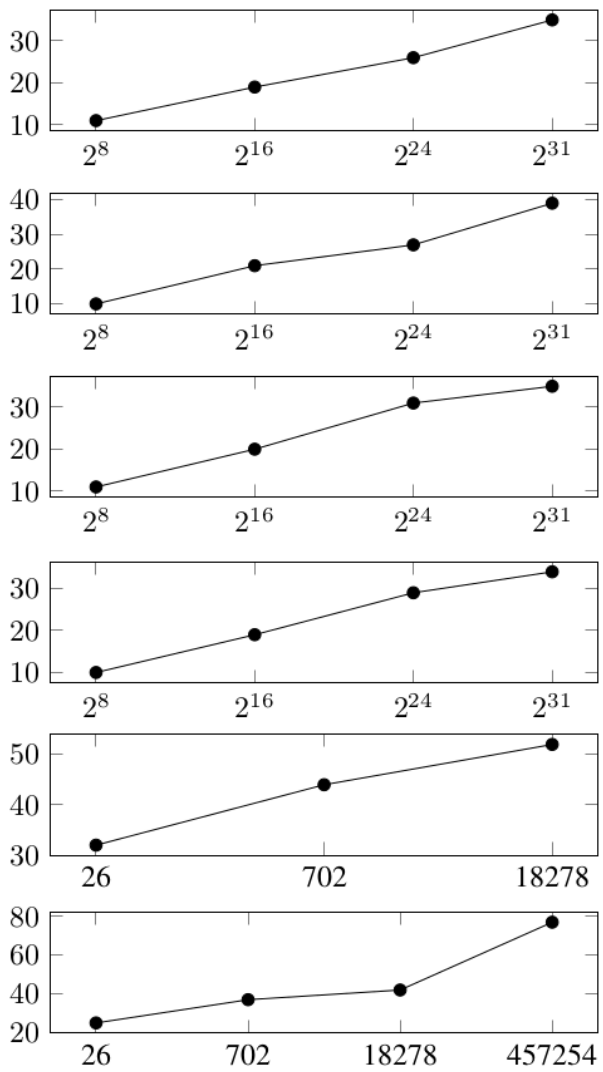
STAC–3

STAC–11A

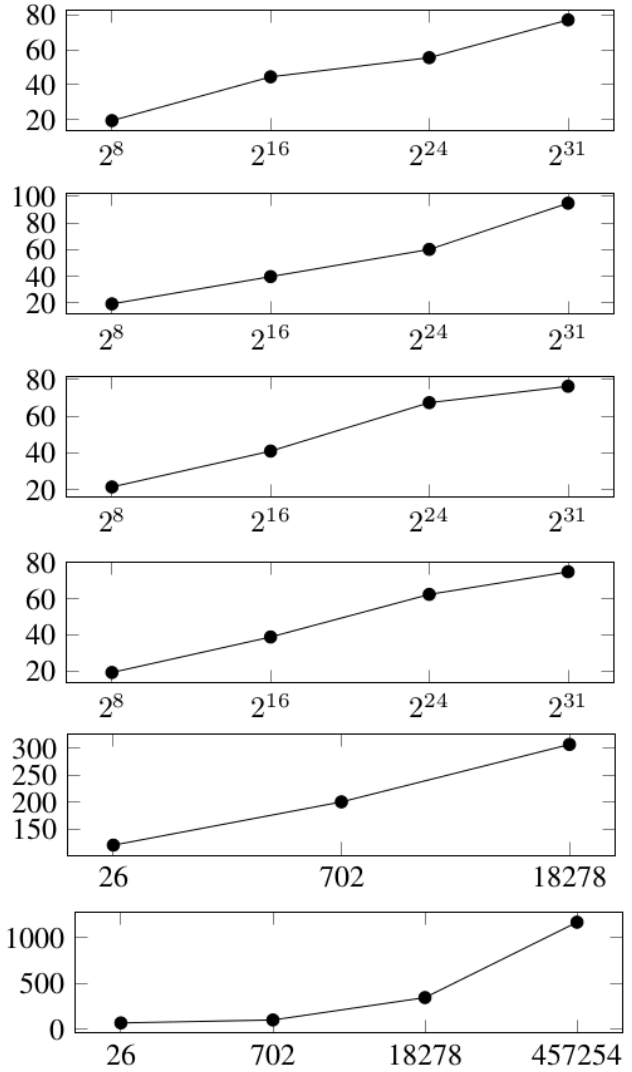
STAC–11B

STAC–4

STAC–12



$|\mathcal{S}|$, Secret Domain Size



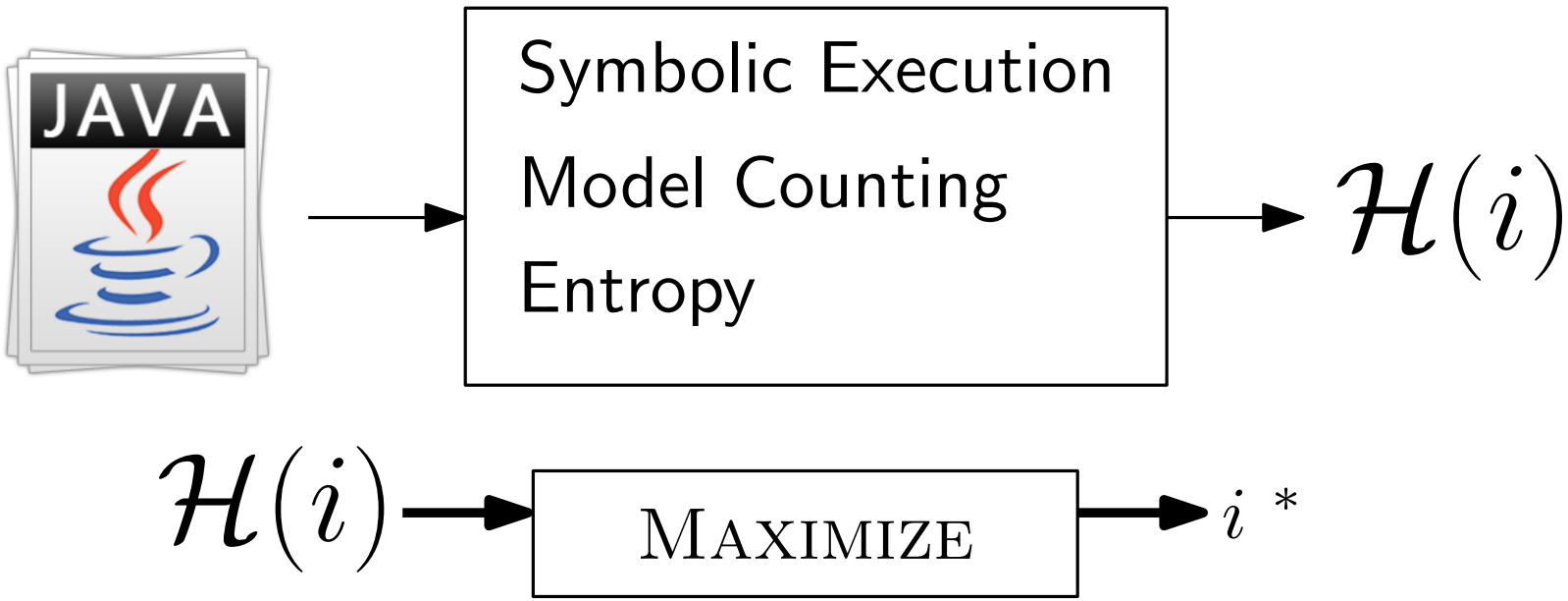
$|\mathcal{S}|$, Secret Domain Size

Closing Remark

```
Boolean compare(String pw, String input){
  for(int i = 0; i < pw.length, i++)
    if(pw[i] != input[i])
      return false;
  return true;
}
```

"Premature optimization is the root of all evil." - Donald Knuth

Summary



Contributions

- Quantifying side-channel leaks with model counting.
- Static offline attack synthesis.
- Dynamic online attack synthesis with noise.
- QIF and attack synthesis for segment oracles.

Publications during PhD

- Aydin, **Bang**, Bultan. [CAV 2015]
  "Automata-Based Model Counting for String Constraints."
- **Bang**, Aydin, Bultan. [FSE 2015]
  "Automatically Computing Path Complexity of Programs."
- **Bang**, Aydin, Phan, Pasareanu, Bultan. [FSE 2016]
  "String Analysis for Side Channels with Segmented Oracles."
- Phan, **Bang**, Pasareanu, Malacaria, Bultan. [CSF 17]
  "Synthesis of Adaptive Side-Channel Attacks."
- **Bang**, Rosner, Bultan. [Euro S&P 2018]
  "Online Synthesis of Adaptive Side-Channel Attacks Based On Noisy Observations."
- Aydin, Eiers, **Bang**, Brennan, Gavrilov, Yu, Bultan. [FSE 2018 (accepted)]
  "Parameterized Model Counting for String and Numeric Constraints."

Submitted papers

- Tsiskaridze, **Bang**, McMahan, Bultan, Sherwood.
  "Information Leakage in Arbiter Protocols."
- Saha, Kadron, Eiers, **Bang**, Bultan.
  "Attack Synthesis for Strings via Incremental Model Counting and Meta-Heuristics."