Software Side-Channel Analysis: Attack Synthesis

Lucas Bang

Dissertation Defense

Committee:
Tevfik Bultan (chair)
Ömer Eğecioğlu
Ben Hardekopf
Publications during PhD

- Aydin, Eiers, Bang, Brennan, Gavrilov, Yu, Bultan. [FSE 2018 (accepted)] “Parameterized Model Counting for String and Numeric Constraints.”

Submitted papers

Publications during PhD

- Aydin, Eiers, Bang, Brennan, Gavrilov, Yu, Bultan. [FSE 2018 (accepted)] “Parameterized Model Counting for String and Numeric Constraints.”

Submitted papers

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
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.
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.
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.

- Panama
- Granada
- Kuwait
Secret Data

Program
private s = getBufferSize();
private s = getBufferSize();

public int compare(int i){
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}
private s = getBufferSize();

public int compare(int i){
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}

\[ s \leq i \Rightarrow o = 1 \]
private s = getBufferSize();

public int compare(int i) {
    if (s <= i) {
        log.write("too large"); // 1 s
    } else {
        some computation; // 2 s
    }
    return 0;
}

s \leq i \Rightarrow o = 1

s > i \Rightarrow o = 2
private s = getBufferSize();

public int compare(int i)
{
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}
```java
private s = getBufferSize();

public int compare(int i){
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}
```

$s \leq i \Rightarrow o = 1$

$s > i \Rightarrow o = 2$
private s = getBufferSize();

public int compare(int i){
if(s <= i)
    log.write("too large"); // 1 s
else
    some computation; // 2 s
    return 0;
}

\[ s \leq i \Rightarrow o = 1 \]
\[ s > i \Rightarrow o = 2 \]
private s = getBufferSize();

public int compare(int i){
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}

\[ s \leq i \Rightarrow o = 1 \]
\[ s > i \Rightarrow o = 2 \]
```java
private s = getBufferSize();

public int compare(int i) {
    if (s <= i) {
        log.write("too large"); // 1 s
    } else {
        some computation; // 2 s
        return 0;
    }
}
```

- $s \leq i \Rightarrow o = 1$
- $s > i \Rightarrow o = 2$
private s = getBufferSize();

public int compare(int i){
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}

1 ≤ s ≤ 8

s ≤ i ⇒ o = 1
s > i ⇒ o = 2
private s = getBufferSize();

public int compare(int i){
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}
public int compare(int i) {
    if (s <= i) {
        log.write("too large"); // 1s
    } else {
        some computation; // 2s
    }
    return 0;
}
```java
private s = getBufferSize();

public int compare(int i){
    if(s <= i)
        log.write("too large"); // 1 s
    else
        some computation; // 2 s
    return 0;
}
```

- $o = 1 \Rightarrow s \leq 4$
- $o = 2 \Rightarrow s > 4$
private s = getBufferSize();

public int compare(int i){
  if(s <= i)
    log.write("too large"); // 1 s
  else
    some computation; // 2 s
  return 0;
}

1 ≤ s ≤ 8

o = 1 ⇒ s ≤ 4

o = 2 ⇒ s > 4

Attacker can binary search on s using i and o.
Is my code vulnerable to side-channel attacks?

```java
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?

```java
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?

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

1. Static Offline Analysis

Synthesized Attack
Is my code vulnerable to side-channel attacks?

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

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

Synthesized Attack
Is my code vulnerable to side-channel attacks?

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

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

Automated Analysis

Synthesized Attack
Side Channels and Searching: Entropy
Side Channels and Searching: Entropy

secret \( s \in S \)

\( i \in I \)
Side Channels and Searching: Entropy

\[ i \in I \]

\[ S \]

secret \( s \in S \)
Side Channels and Searching: Entropy

\[ \text{secret } s \in S \]

\[ i \in I \]

\[ o_1 \]

\[ o_2 \]

\[ S \]
Side Channels and Searching: Entropy

secret $s \in S$

$i \in I$

$S$

$o_1$

$o_2$
Side Channels and Searching: Entropy

secret $s \in S$

$i \in I$

$S$

$o_1$

$o_2$
Side Channels and Searching: Entropy

\[ i \in I \]

\[ S \]

\[ \text{secret } s \in S \]

Good outcome, very unlikely.
Side Channels and Searching: Entropy

\[ i \in I \]

\[ s \in S \]

Bad outcome, very likely.

secret \( s \in S \)
Side Channels and Searching: Entropy

\[ i \in I \]

\[ S \]

secret \( s \in S \)
Side Channels and Searching: Entropy

secret $s \in S$

$i \in I$

$S$

$s$
Side Channels and Searching: Entropy

\[ i \in I \]

secret \( s \in S \)
Side Channels and Searching: Entropy

\[ i \in I \]

\[ S \]

\[ p(s \in \cdot) \]

\[ \text{secret } s \in S \]
Side Channels and Searching: Entropy

\[ p(s \in S) = \overline{\text{secret } s \in S} \]

\[ i \in I \]
Side Channels and Searching: Entropy

\[ i \in I \]

\[ \text{secret } s \in S \]
Side Channels and Searching: Entropy

$\text{secret } s \in S$

$i \in I$

$p_1$

$p_2$

$p_3$

$p_4$

Quantify expected information gain measured in bits.
Side Channels and Searching: Entropy

Let $i \in I$ be a set of indexes, $S$ be a set of secrets, $s \in S$ be a secret, and $p_j$ be the probability of $j$th event. Quantify expected information gain measured in bits.

$$\frac{1}{p_j}$$
Side Channels and Searching: Entropy

secret $s \in S$

$i \in I$

Quantify expected information gain measured in bits.

$$\log_2 \frac{1}{p_j}$$
Side Channels and Searching: Entropy

Quantify expected information gain measured in bits.

$$\sum_{j=1}^{n} p_j \log_2 \frac{1}{p_j}$$
Quantify expected information gain measured in bits.

\[ H = \sum_{j=1}^{n} p_j \log_2 \frac{1}{p_j} \]
Side Channels and Searching: Entropy

Quantify expected information gain measured in bits.

\[ H(i) = \sum_{j=1}^{n} p_j \log_2 \frac{1}{p_j} \]
$\max \mathcal{H}(i) \Rightarrow \text{Binary Search}$

$o = 1 \Rightarrow s \leq i \quad \quad o = 2 \Rightarrow s > i$
 Binary Search

\[ \text{o} = 1 \implies s \leq i \]

\[ \text{o} = 2 \implies s > i \]
Password Checker
Constraints

\[
\begin{align*}
\max H(i) &\Rightarrow \text{Binary Search} \\
o & = 2 \Rightarrow s > i \\
\end{align*}
\]
\[
\max \mathcal{H}(i) \Rightarrow \text{Binary Search}
\]

\[
o = 1 \Rightarrow s \leq i \quad o = 2 \Rightarrow s > i
\]
$$\max \mathcal{H}(i) \Rightarrow \text{Binary Search}$$

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

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

any program constraints
\[
\max H(i) \implies \text{Binary Search}
\]

\[
o = 1 \implies s \leq i \quad \quad o = 2 \implies s > i
\]

\[
\max H(i) \implies \text{Optimal Search}
\]

any program constraints
Symbolic Execution

Execute program on **symbolic** rather than concrete inputs. Maintain **path constraints**, PCs, $\phi_j$ over symbolic inputs.

For branch instructions:

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

\[
\begin{align*}
\phi & \leftarrow \phi \land c \\
\phi & \leftarrow \phi \land \neg c
\end{align*}
\]

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

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

Maintain **path constraints**, PCs, $\phi_j$ over symbolic inputs.

For branch instructions:

\[
\text{if}(c) \text{ then } s_1; \text{ else } s_2;
\]

\[
\phi \leftarrow \phi \land c
\]

\[
\phi \leftarrow \phi \land \neg c
\]

$\phi_j(s, i)$ characterizes the relation between $s$, $i$, and $o_j$
\[ p(s \in \text{ } \text{ } ) = \frac{\text{ }}{\text{ }} \]
\[ p(s \in \phi) = \frac{\#\phi(i)}{|\phi|} \]
\[
\{ \phi_j(s, i) \} \xrightarrow{\text{Model Counter}} \{ \#\phi_j(i) \}
\]

\[
p(s \in \phi) = \frac{\#\phi}{\#\phi(i)}
\]
\[ \{ \phi_j(s, i) \} \rightarrow \text{MODEL COUNTER} \rightarrow \{ \#\phi_j(i) \} \]

\[ \#\phi(i) = \#\phi(i) \]

\#\phi(i) \text{ is the number of satisfying solutions (models) for } \phi(s, i) \text{ for a given } i. \]

\[ p(s \in \phi) = \frac{\#\phi}{\#\phi(i)} \]
$\{ \phi_j (s, i ) \} \xrightarrow{\text{Model Counter}} \{ \#\phi_j (i) \}$

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

$$p(s \in \phi) = \frac{\mid \phi \mid}{|S|}$$

$$p(i) = \frac{\#\phi(i)}{|S|}$$
#ϕ(i) is the number of satisfying solutions (models) for ϕ(s, i) for a given i.

\[
p(s \in \phi) = \frac{|\phi|}{|S|}
\]

\[p(i) = \frac{#ϕ(i)}{|S|}\]

\[
H(i) = \sum_{j=1}^{n} p_j(i) \log_2 \frac{1}{p_j(i)}
\]
$\mathcal{H}(i)$ is a symbolic expression that measures the expected information an attacker gains when making input $i$. 
$H(i)$ is a symbolic expression that measures the expected information an attacker gains when making input $i$.

Maximizing $H(i)$ 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]
private s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1 s
    else
        log.write("too many bytes"); // 2s
    return 0;
}
private s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1 s
    else
        log.write("too many bytes"); // 2s
    return 0;
}
```
private s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1 s
    else
        log.write("too many bytes"); // 2s
    return 0;
}
```
```java
private int s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1 s
    else
        log.write("too many bytes"); // 2s
    return 0;
}
```
private s = getMaxBytes();

public int compare(int i) {
  if (s <= i) {
    some computation; // 1s
  } else {
    log.write("too many bytes"); // 2s
    return 0;
  }
}
private s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1s
    else
        log.write("too many bytes"); // 2s
    return 0;
}
private s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1 s
    else
        log.write("too many bytes"); // 2 s
    return 0;
}
private s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1 s
    else
        log.write("too many bytes"); // 2s
    return 0;
}
private s = getMaxBytes();

public int compare(int i) {
    if(s <= i) {
        some computation; // 1 s
    } else {
        log.write("too many bytes"); // 2s
    }
    return 0;
}
private s = getMaxBytes();

public int compare(int i) {
    if (s <= i) {
        some computation; // 1 s
    } else {
        log.write("too many bytes"); // 2 s
    }
    return 0;
}
private s = getMaxBytes();

public int compare(int i){
    if(s <= i)
        some computation; // 1 s
    else
        log.write("too many bytes"); // 2s
    return 0;
}
Attacker Belief

$s$?
Attacker Belief

\[ s? \]
Attacker Belief

\[ s? \]

Input Choice

\[ i^* \]
Attacker Belief

Input Choice

Observation Noise

$s < i$

$s > i$

15 - 5
Attacker Belief

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ s \leq 5 \quad s > 5 \]
Attacker Belief

\[ s? \]

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ s \leq 5 \quad s > 5 \]

\[ t = 4.12 \]
Attacker Belief

\[ s? \]

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ s \leq 5 \quad s > 5 \]

\[ t = 4.12 \]

1 2 3 4 5 6 7 8

\[ \frac{1}{3} \quad \frac{1}{8} \]
Attacker Belief

\[ s? \]

\[ \frac{1}{8} \]

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ s \leq 5 \]

\[ s > 5 \]
Attacker Belief: $s$?

Input Choice: $i^* = 5$

Observation Noise:
- $s \leq 5$
- $s > 5$
  
  $t = 2.3$
Attacker Belief

\[ s? \]

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ s \leq 5 \quad s > 5 \]
Attacker Belief

\[ p(s|o, i^*) \]

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ s \leq 5 \]

\[ s > 5 \]
Attacker Belief

Input Choice

Observation Noise

$s$?  

$i^* = 5$

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

$p(o|s, i)$

$s \leq 5$  

$s > 5$
Attacker Belief

Input Choice

Observation Noise

$s$?

\[ i^* = 5 \]

\[ p(s|o, i^*) \]

\[ p(o|s, i) \]

\[ p(o|s, i) \]

\[ s \leq 5 \]

\[ s > 5 \]
Attacker Belief

$s$?

Input Choice

$i^* = 5$

Observation Noise

$s \leq 5$

$s > 5$

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

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

$p(o|s, i)$
Attacker Belief

Input Choice

Observation Noise

\[ i^* = 5 \]

\[ p(s | o, i^*) \]

\[ p(o | s, i^*) \]

\[ p(o | s, i) \]

\[ s \leq 5 \]

\[ s > 5 \]
Attacker Belief

\[ s? \]

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(o|s, i) \]
Attacker Belief

Input Choice

Observation Noise

\[ i^* = 5 \]

\[ s \leq 5 \]
\[ s > 5 \]

\[ p(s \mid o, i^*) \]

\[ p(o \mid s, i) \]

Bayes’ Rule
Attacker Belief

\[ s? \]

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ p(o|s,i) \]

\[ p(s,o,i^*) \]

Bayes' Rule

\[ p(s|o,i^*) \]

\[ s \leq 5 \]

\[ s > 5 \]
Attacker Belief

$s$?

Input Choice

\[ i^* = 5 \]

Observation Noise

\[ p(s|o, i^*) \]

\[ p(o|s, i) \]

Bayes’ Rule

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]

\[ p(s|o, i^*) \]
Attacker Belief

Input Choice

Observation Noise

$s$?

$i^* = 5$

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

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

$p(o | s, i)$

Bayes’ Rule

Weighted Model Counting

$s \leq 5$

$s > 5$
Attacker Belief: $s$?

Input Choice: $i^* = 5$

Observation Noise: $s \leq 5$ and $s > 5$

Bayes' Rule:

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

Weighted Model Counting:

$max \mathcal{H}(i)$

Diagram:

- Column: Values from 1 to 8 with probability 1/8 for each.
- Row: Values from 1 to 8 with probability 1/8 for each.
- Cells: Green bars indicating the belief distribution.
- Arrows: Directed relationships showing the flow of information.
- Formulas: Expressions for $p(o|s, i)$ and $p(s|o, i^*)$. 
16 - 1
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
Symbolic Execution

\[ P(s, i) \]

Source Code

\[ \{ \phi_j(s, i) \} \]

path constraints
$P(s, i)$

Source Code

**Symbolic Execution**

$\{\phi_j(s, i)\} \rightarrow \{w_j = (s_j, i_j)\}$

path constraints

PC models (witnesses)
Each PC characterizes an observable program behavior

\[(s, i) \models \phi_j \quad (s', i') \models \phi_j\]
Each PC characterizes an observable program behavior

\[(s, i) \models \phi_j \quad \quad (s', i') \models \phi_j\]
Each PC characterizes an observable program behavior

\[(s, i) \models \phi_j \quad \quad (s', i') \models \phi_j\]

\[P(s, i) \quad ? \quad ? \quad P(s', i')\]

\(\phi_j (s, i)\) 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.

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

\[ \{ w_j = (s_i, i_j) \} \times 1000 \]
Characterize effect of noise on each class of program behaviors using the witness for that behavior.

\[ \{ w_j = (s_i, i_j) \} \times 1000 \]

\[ P(s, i) \]

\[ \{ p(o|s_j, i_j), \ldots \} \]
Characterize effect of noise on each class of program behaviors using the witness for that behavior.

\[ \{ w_j = (s_i, i_j) \} \times 1000 \]

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

\[
\{w_j = (s_i, i_j )\} \times 1000 \rightarrow \text{Network} \rightarrow \begin{cases} P(s, i) \end{cases} \rightarrow \begin{cases} p(o|s_j, i_j) \end{cases}
\]

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

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

\[
\{w_j = (s_i, i_j)\} \times 1000
\]

Merging via Hellinger Distance

(a) \( d_H = 0.068 \)  
(b) \( d_H = 0.491 \)  
(c) \( d_H = 0.978 \)
Characterize effect of noise on each class of program behaviors using the witness for that behavior.

\[ \{ w_j = (s_i, i_j) \} \times 1000 \]

\[ P(s, i) \]

\[ \text{HW / OS} \]

\[ \{ p(o|s_j, i_j) \} \]

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

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

\[ \{ w_j = (s_i, i_j ) \} \times 1000 \]

\[ \text{Network} \]

\[ P(s, i ) \]

\[ \{ \text{\ldots} \} \]

\[ p(o|s_j, i_j ) \]

\[ d_H(\hat{p}(o|\phi_1), \hat{p}(o|\phi_2)) < \tau \Rightarrow \text{let } \phi' = \phi_1 \lor \phi_2 \]

**Observation constraint** \( \phi' \):

Disjunction over path constraints which characterizes inputs that are observationally indistinguishable via side-channel observation.
\{ \mathcal{A}, \mathcal{B}, \ldots \} \hat{p}(o|\phi)

Belief $p(s)$
\{\mathcal{M}, \mathcal{A}, \ldots \} \xrightarrow{\hat{p}(o|\phi)} p(s) \xrightarrow{\text{Belief}} \text{MAX } I \xrightarrow{i^*}
\[ \{ \text{Belief} \} \rightarrow \hat{p}(o|\phi) \rightarrow \text{MAX } I \rightarrow i^* \rightarrow P(s,i) \rightarrow \text{HW/OS} \]
\[
\{ \text{Belief} \} \quad \hat{p}(o|\phi) \quad p(s|o, i^*) \quad \text{MAX } I \quad i^* \quad P(s, i) \quad \text{HW / OS}
\]

Belief Update

Bayesian Update

observe \( o \)
\[
\{ \mathbf{\mu}, \mathbf{\Sigma}, \ldots \} \rightarrow \hat{p}(o|\phi) \\
p(s|o, i^*) \rightarrow \text{MAX } \mathcal{I} \rightarrow i^* \\
\downarrow \text{Belief} \qquad \uparrow \text{Bayesian Update} \qquad \text{observe } o \\
\downarrow \text{Network} \quad \begin{array}{c} P(s, i) \\ \text{HW / OS} \end{array}
\]
\[ \mathcal{I}(s; \phi(s, i)|i) = - \sum_{i=1}^{n} p(\phi(s, i)|i) \log_2 p(\phi(s, i)|i) \]

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

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

Expected info gain
given attacker input

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

Observation constraint probabilities

Model Counting
\[ I(s; \phi(s, i)|i) = - \sum_{i=1}^{n} p(\phi(s, i)|i) \log_2 p(\phi(s, i)|i) \]

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

Expected info gain given attacker input

Observation constraint probabilities

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

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

\text{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

Python Profiler Client  Intel NUC Server

3. Online Attack Synthesis

Barvinok Weighted Symbolic Model Counting  Mathematica Symbolic Entropy Computation Numeric Maximization

$P(s, i)$
Case Study: LawDB

From DARPA Space-Time Analysis for Cybersecurity (STAC)

DB: key = employee ID

Some employee IDs have restricted access.

Server

DB: key = employee ID
Some employee IDs have restricted access.

Client

SEARCH midID maxID

List of employees.

Writes to log file depending on whether

\[ ID_{res} \in [\text{minID}, \text{maxID}] \]
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 0: SEARCH

Observed time: 

Entropy = 6.64386
1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92

STEP 1: SEARCH 19 52

Observed time: 0.00444

Entropy = 6.27408
\[ 1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92 \]

**STEP 2: SEARCH 10 63**

*Observed time:* 0.00436

*Entropy: 5.81014*
\[ 1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92 \]

STEP 3: SEARCH 1 63

Observed time: 0.0043

Entropy = 5.28658
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

**STEP 4: SEARCH 63 85**

Observed time: 0.00733

Entropy = 3.53218
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 5: SEARCH 70 73

Observed time: 0.00447

Entropy = 3.19249
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 6: SEARCH 67 74

Observed time: 0.00427

Entropy = 2.74012
\[1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92\]

STEP 7: SEARCH 63 74

Observed time: 0.00452

Entropy = 2.41548
\[ 1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92 \]

**STEP 8: SEARCH 63 70**

Observed time: 0.00435

Entropy = 2.07286

![Secret ID Distribution Graph](image)
1 ≤ ID ≤ 100  \hspace{0.5cm} ID_1 = 64 \hspace{0.5cm} ID_2 = 85 \hspace{0.5cm} ID_{res} = 92

STEP 9: SEARCH 74 75

Observed time: 0.00431
Entropy = 2.46103
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 10: SEARCH 74 75

Observed time: 0.00435

Entropy = 2.39414
\[ 1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92 \]

STEP 11: SEARCH 63 100

Observed time: 0.00732

Entropy = 4.19456
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 12: SEARCH 74 100

Observed time: 0.00743

Entropy = 4.73142

![Graph showing the distribution of secret IDs with peaks at ID1 and ID2, and a large peak at IDres.](image-url)
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

**STEP 13: SEARCH 78 100**

Observed time: $0.00733$

Entropy = $4.70767$
$1 \leq ID \leq 100$  \hspace{1cm} ID_1 = 64 \hspace{1cm} ID_2 = 85 \hspace{1cm} ID_{res} = 92$

STEP 14: SEARCH 86 100

Observed time: 0.00728

Entropy = 4.68363
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 15: SEARCH 87 99

Observed time: 0.00716

Entropy = 4.37901
\[1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92\]

STEP 16: SEARCH 87 95
Observed time: 0.00727
Entropy = 3.83405

\[\text{p(secret ID)}\] vs secret ID graph
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 17: SEARCH 91 95

Observed time: 0.00731

Entropy = 3.87438
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 18: SEARCH 92 95

Observed time: 0.0072

Entropy = 2.9822
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

STEP 19: SEARCH 92 94

Observed time: 0.00729

Entropy = 2.98878

$p(\text{secret ID})$

0.0
0.2
0.4
0.6
0.8
1.0

secret ID

0
20
40
60
80
100
$1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92$

**STEP 20: SEARCH 92 93**

Observed time: 0.00735

Entropy = 2.22644
\[1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92\]

STEP 21: SEARCH 92 92

Observed time: 0.00739

Entropy = 0.767476
\[ 1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92 \]

STEP 22: SEARCH 92 92

Observed time: 0.00715

Entropy = 0.170871
\[ 1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92 \]

**STEP 23: SEARCH 92 92**

Observed time: 0.00746

Entropy = 0.026079
1 \leq ID \leq 100 \quad ID_1 = 64 \quad ID_2 = 85 \quad ID_{res} = 92

STEP 24: SEARCH 92 92

Observed time: 0.00721
Entropy = 0.026084
<table>
<thead>
<tr>
<th>ID Range</th>
<th># Employees</th>
<th>Offline Analysis</th>
<th>Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-100</td>
<td>3</td>
<td>57s</td>
<td>2m38s</td>
</tr>
<tr>
<td>1-10000</td>
<td>4</td>
<td>2m21s</td>
<td>2m43s</td>
</tr>
<tr>
<td>1-10000</td>
<td>5</td>
<td>6m30s</td>
<td>3m08s</td>
</tr>
</tbody>
</table>
| 1-10000    | 10          | 42m09s           | 4m31s  | 77

2m21s = 2 minutes 21 seconds
3m08s = 3 minutes 8 seconds
42m09s = 42 minutes 9 seconds
57s = 57 seconds
2m38s = 2 minutes 38 seconds
2m43s = 2 minutes 43 seconds
5m30s = 5 minutes 30 seconds
4m31s = 4 minutes 31 seconds
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

```java
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

```java
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

```java
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

```java
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

```java
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

```java
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

<table>
<thead>
<tr>
<th>Phase 0</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>csja</td>
<td>ciub</td>
<td>ciqi</td>
<td>ciqa</td>
</tr>
<tr>
<td>fzgk</td>
<td>cnte</td>
<td>ciji</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>daaz</td>
<td>cwcs</td>
<td>cimq</td>
<td>ciqz</td>
<td>ciqg</td>
</tr>
<tr>
<td>zgap</td>
<td>ctdo</td>
<td>citz</td>
<td>ciqu</td>
<td>ciqa</td>
</tr>
<tr>
<td>uaak</td>
<td>cved</td>
<td>ciqz</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>bnza</td>
<td>cvfo</td>
<td>ciaz</td>
<td>ciqc</td>
<td>ciqa</td>
</tr>
<tr>
<td>ecjq</td>
<td>ceyu</td>
<td>ciok</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>zmna</td>
<td>ciil</td>
<td>cida</td>
<td>ciqe</td>
<td></td>
</tr>
<tr>
<td>tzar</td>
<td></td>
<td>cijw</td>
<td>ciqr</td>
<td></td>
</tr>
<tr>
<td>zmna</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vzsc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qyas</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asvr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cmxq</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Phase 0:** ε

**Phase 1:** csja
cnte
cwcs
czte

crdt

cvfo
ceyu
cil

ciw

**Phase 2:**

- ciub
- ciji
- cimq
- citz
- ciok
- cida
- cijw
- cqw

**Phase 3:**

- ciqi
- ciqz
- ciqz
- ciqu
- ciqz
- ciqe
- ciqr

**Phase 4:**

- ciqa
- ciqa
- ciqg
- ciqa
- ciqa
- ciqa
- ciqa

---

22 - 2
<table>
<thead>
<tr>
<th>Phase 0</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>csja</td>
<td>ciub</td>
<td>ciqi</td>
<td>ciqa</td>
</tr>
<tr>
<td>fzgk</td>
<td>cnte</td>
<td>ciij</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>daaz</td>
<td>cwcs</td>
<td>cimq</td>
<td>ciqz</td>
<td>ciqg</td>
</tr>
<tr>
<td>zgap</td>
<td>ctdo</td>
<td>citz</td>
<td>ciqu</td>
<td>ciqa</td>
</tr>
<tr>
<td>uaak</td>
<td>cved</td>
<td>ciqz</td>
<td>ciqz</td>
<td></td>
</tr>
<tr>
<td>bnza</td>
<td>cvfo</td>
<td>ciaz</td>
<td>ciqc</td>
<td></td>
</tr>
<tr>
<td>ecjq</td>
<td>ceyu</td>
<td>ciok</td>
<td>ciqz</td>
<td></td>
</tr>
<tr>
<td>zmna</td>
<td>ciil</td>
<td>cida</td>
<td>ciqe</td>
<td></td>
</tr>
<tr>
<td>tzar</td>
<td></td>
<td>cijw</td>
<td>ciqr</td>
<td></td>
</tr>
<tr>
<td>zmna</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vzsc</td>
<td></td>
<td>cigz</td>
<td>ciqk</td>
<td></td>
</tr>
<tr>
<td>qyas</td>
<td></td>
<td>cisu</td>
<td>ciqd</td>
<td></td>
</tr>
<tr>
<td>asvr</td>
<td></td>
<td>cisp</td>
<td>ciqd</td>
<td></td>
</tr>
<tr>
<td>cmxq</td>
<td></td>
<td>cine</td>
<td>ciqr</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ciqk</td>
<td>ciqz</td>
<td></td>
</tr>
</tbody>
</table>

**Attack Synthesis vs. PW Checker**

22 - 3
### Attack Synthesis vs. PW Checker

<table>
<thead>
<tr>
<th>Phase 0</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>csja</td>
<td>ciub</td>
<td>ciqi</td>
<td>ciqa</td>
</tr>
<tr>
<td>fzgk</td>
<td>cnte</td>
<td>ciji</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>daaz</td>
<td>cwcs</td>
<td>cimq</td>
<td>ciqz</td>
<td>ciqg</td>
</tr>
<tr>
<td>zgap</td>
<td>ctdd</td>
<td>citz</td>
<td>ciqu</td>
<td>ciqa</td>
</tr>
<tr>
<td>uaak</td>
<td>cved</td>
<td>ciqz</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>bnza</td>
<td>cvfo</td>
<td>ciaz</td>
<td>ciqc</td>
<td>ciqa</td>
</tr>
<tr>
<td>ecjq</td>
<td>ceyu</td>
<td>ciok</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>zmna</td>
<td>ciil</td>
<td>cida</td>
<td>ciqe</td>
<td>ciqa</td>
</tr>
<tr>
<td>tzar</td>
<td></td>
<td>cijw</td>
<td>ciqr</td>
<td>ciqa</td>
</tr>
<tr>
<td>zmna</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vzsc</td>
<td></td>
<td>cigz</td>
<td>ciqk</td>
<td>ciqa</td>
</tr>
<tr>
<td>qyas</td>
<td></td>
<td>cisu</td>
<td>ciqd</td>
<td>ciqa</td>
</tr>
<tr>
<td>asvr</td>
<td></td>
<td>cisp</td>
<td>ciqd</td>
<td>ciqa</td>
</tr>
<tr>
<td>cmxq</td>
<td></td>
<td>cine</td>
<td>ciqr</td>
<td>ciqa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ciqk</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
</tbody>
</table>
## Attack Synthesis vs. PW Checker

<table>
<thead>
<tr>
<th>Phase 0</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>csja</td>
<td>ciub</td>
<td>ciqi</td>
<td>ciqa</td>
</tr>
<tr>
<td>fzgk</td>
<td>cnte</td>
<td>ciij</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>daaz</td>
<td>cwcs</td>
<td>cimq</td>
<td>ciqz</td>
<td>ciqg</td>
</tr>
<tr>
<td>zgap</td>
<td>ctdo</td>
<td>citz</td>
<td>ciqu</td>
<td>ciqa</td>
</tr>
<tr>
<td>uaak</td>
<td>cved</td>
<td>ciqz</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>bnza</td>
<td>cvfo</td>
<td>ciaz</td>
<td>ciqc</td>
<td></td>
</tr>
<tr>
<td>ecjq</td>
<td>ceyu</td>
<td>ciok</td>
<td>ciqz</td>
<td></td>
</tr>
<tr>
<td>zmna</td>
<td>ciil</td>
<td>cida</td>
<td>ciqe</td>
<td></td>
</tr>
<tr>
<td>tzar</td>
<td></td>
<td>cijw</td>
<td>ciqr</td>
<td></td>
</tr>
<tr>
<td>zmna</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vzsc</td>
<td></td>
<td></td>
<td>ciqk</td>
<td></td>
</tr>
<tr>
<td>qyas</td>
<td></td>
<td></td>
<td>ciqd</td>
<td></td>
</tr>
<tr>
<td>asvr</td>
<td></td>
<td></td>
<td>ciqk</td>
<td></td>
</tr>
<tr>
<td>cmxq</td>
<td></td>
<td></td>
<td>ciqr</td>
<td></td>
</tr>
</tbody>
</table>

22 - 5
### Attack Synthesis vs. PW Checker

<table>
<thead>
<tr>
<th>Phase 0</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>csja</td>
<td>ciub</td>
<td>ciqi</td>
<td>ciqa</td>
</tr>
<tr>
<td>fzgk</td>
<td>cnte</td>
<td>ciji</td>
<td>ciqz</td>
<td>ciqa</td>
</tr>
<tr>
<td>daaz</td>
<td>cwcs</td>
<td>cimq</td>
<td>ciqz</td>
<td>ciqg</td>
</tr>
<tr>
<td>zgap</td>
<td>ctdo</td>
<td>citz</td>
<td>ciqu</td>
<td>ciqa</td>
</tr>
<tr>
<td>uaak</td>
<td>cved</td>
<td>ciqz</td>
<td>ciqz</td>
<td></td>
</tr>
<tr>
<td>bnza</td>
<td>cvfo</td>
<td>ciaz</td>
<td>ciqc</td>
<td></td>
</tr>
<tr>
<td>ecjq</td>
<td>ceyu</td>
<td>ciok</td>
<td>ciqz</td>
<td></td>
</tr>
<tr>
<td>zmna</td>
<td>ciil</td>
<td>cida</td>
<td>ciqe</td>
<td></td>
</tr>
<tr>
<td>tzar</td>
<td></td>
<td>cijw</td>
<td>ciqr</td>
<td></td>
</tr>
<tr>
<td>zmna</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vzsc</td>
<td></td>
<td></td>
<td>ciqk</td>
<td></td>
</tr>
<tr>
<td>qyas</td>
<td></td>
<td></td>
<td>ciqd</td>
<td></td>
</tr>
<tr>
<td>asvr</td>
<td></td>
<td></td>
<td>ciqd</td>
<td></td>
</tr>
<tr>
<td>cmxq</td>
<td></td>
<td></td>
<td>ciqr</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ciqz</td>
<td></td>
</tr>
</tbody>
</table>

22 - 6
## Attack Synthesis vs. PW Checker

<table>
<thead>
<tr>
<th>Phase 0</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>csja</td>
<td>ciub</td>
<td>ciqi</td>
<td>ciqa</td>
</tr>
<tr>
<td>fzgk</td>
<td>cnte</td>
<td>ciij</td>
<td>$\rightarrow$</td>
<td>ciqz</td>
</tr>
<tr>
<td>daaz</td>
<td>cwcs</td>
<td>cimq</td>
<td>$\rightarrow$</td>
<td>ciqz</td>
</tr>
<tr>
<td>zgap</td>
<td>ctdo</td>
<td>citz</td>
<td></td>
<td>ciqu</td>
</tr>
<tr>
<td>uaak</td>
<td>cved</td>
<td>ciqz</td>
<td>$\rightarrow$</td>
<td>ciqz</td>
</tr>
<tr>
<td>bnza</td>
<td>cvfo</td>
<td>ciaz</td>
<td></td>
<td>ciqc</td>
</tr>
<tr>
<td>ecjq</td>
<td>ceyu</td>
<td>ciok</td>
<td>$\rightarrow$</td>
<td>ciqz</td>
</tr>
<tr>
<td>zmna</td>
<td>ciil</td>
<td>cida</td>
<td></td>
<td>ciqe</td>
</tr>
<tr>
<td>tzar</td>
<td></td>
<td></td>
<td></td>
<td>ciqr</td>
</tr>
<tr>
<td>zmna</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vzsc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qyas</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asvr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cmxq</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

22 - 7
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($H$) | $|H|$ | $|\Phi|$ | $|T|$ | Vulnerable? | 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 | 106.13|
| 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($H$) | $|\mathbb{H}|$ | $|\Phi|$ | $|T|$ | Vuln? | 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($H$) | $|H|$  | $|\Phi|$ | $|T|$ | 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($H$) | $|H|$  | $|\Phi|$ | $|T|$ | 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 |

24 - 4
## Space/Time Analysis for Cybersecurity Benchmark

| Benchmark | Dim($H$) | $|H|$ | $|\Phi|$ | $|T|$ | Vuln? | 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 |

24 - 5
# Space/Time Analysis for Cybersecurity Benchmark

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

| Benchmark   | Dim($H$) | |H| | |Φ| | |T| | Vuln? | 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|

24 - 8
Space/Time Analysis for Cybersecurity Benchmark

<table>
<thead>
<tr>
<th>STAC–1</th>
<th>Number of Attack Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC–3</td>
<td>Attack Synthesis Time</td>
</tr>
<tr>
<td>STAC–11A</td>
<td>Number of Attack Steps</td>
</tr>
<tr>
<td>STAC–11B</td>
<td>Attack Synthesis Time</td>
</tr>
<tr>
<td>STAC–4</td>
<td>Number of Attack Steps</td>
</tr>
<tr>
<td>STAC–12</td>
<td>Attack Synthesis Time</td>
</tr>
</tbody>
</table>

\[ |S|, \text{ Secret Domain Size} \]
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

Symbolic Execution
Model Counting
Entropy

\[ H(i) \]

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.”