Reasoning about Aggregation of Information in Timing Attacks

Itsaka Rakotonirina
INRIA Nancy Grand-Est

Boris Köpf
Microsoft Research
Choose a letter: A or B.
Choose a letter: A or B.

Does it occur more than 10 times in TARATATATARATATATA?
Choose a letter: A or B.
Does it occur more than 10 times in TARATATATARATATATA?
Choose a letter: A or B.

Does it occur more than 10 times in TARATATATARATATATA?

No.
Choose a letter: A or B.

Does it occur more than 10 times in TARATATATARATATATA?

Q: Which letter was chosen?

No.
Timing attacks

1996  on RSA (Kocher)
1998  on RSA (Dhem et al.)
2005  on AES (Bernstein)
2007  on AES (Acıçmez et al.)
2013  Lucky Thirteen (AlFardan, Paterson)
2014  Flush+Reload (Yarom, Falkner)
2016  on ECDH (Kaufmann et al.)
2018  Spectre (Kocher et al.)
2018  Meltdown (Lipp et al.)
2019  RIDL (van Schaik et al.)
2019  ZombieLoad (Schwarz et al.)
Timing attacks

1996 on RSA (Kocher)
1998 on RSA (Dhem et al.)
2005 on AES (Bernstein)
2007 on AES (Acııçmez et al.)
2013 Lucky Thirteen (AlFardan, Paterson)
2014 Flush+Reload (Yarom, Falkner)
2016 on ECDH (Kaufmann et al.)
2018 Spectre (Kocher et al.)
2018 Meltdown (Lipp et al.)
2019 RIDL (van Schaik et al.)
2019 ZombieLoad (Schwarz et al.)

A long-term secret, and queries to an oracle
\(O : public\ input \rightarrow execution\ time\ of\ a\ program\)

Remote measurements
Timing attacks

1996 on RSA (Kocher)
1998 on RSA (Dhem et al.)
2005 on AES (Bernstein)
2007 on AES (Acııçmez et al.)
2013 Lucky Thirteen (AlFardan, Paterson)
2014 Flush+Reload (Yarom, Falkner)
2016 on ECDH (Kaufmann et al.)
2018 Spectre (Kocher et al.)
2018 Meltdown (Lipp et al.)
2019 RIDL (van Schaik et al.)
2019 ZombieLoad (Schwarz et al.)

A long-term secret, and queries to an oracle
\( O : \text{public input} \mapsto \text{execution time of a program} \)

Remote measurements

Exploit timing variations, and not the absolute execution time

Differential measurements
Timing attacks

1996 on RSA (Kocher)
1998 on RSA (Dhem et al.)
2005 on AES (Bernstein)
2007 on AES (Acıçmez et al.)
2013 Lucky Thirteen (AlFardan, Paterson)
2014 Flush+Reload (Yarom, Falkner)
2016 on ECDH (Kaufmann et al.)
2018 Spectre (Kocher et al.)
2018 Meltdown (Lipp et al.)
2019 RIDL (van Schaik et al.)
2019 ZombieLoad (Schwarz et al.)

A long-term secret, and queries to an oracle 
\[ O : \text{public input} \mapsto \text{execution time of a program} \]

Remote measurements

Exploit timing variations, and not the absolute execution time

Differential measurements

The secret is recovered chunk by chunk

Compositionality
## Timing attacks

### Attacker model

<table>
<thead>
<tr>
<th>Remote measurements</th>
<th>Differential measurements</th>
<th>Compositionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>A long-term secret, and queries to an oracle $O: public input \mapsto execution\ time\ of\ a\ program$</td>
<td>Exploit timing variations, and not the absolute execution time</td>
<td>The secret is recovered chunk by chunk</td>
</tr>
</tbody>
</table>

### Under what hypotheses?

- Exploit timing variations, and not the absolute execution time
- A long-term secret, and queries to an oracle
- Remote measurements
Contributions

- A model of timing attacks
capturing the essence of compositional attacks

- Core hypotheses giving rise to efficient attacks
under the form of independence properties

- Generic attack descriptions + cost analyses
A model for timing leakage
**Program**

- **k**: Long-term secret constant across all invocations of the program
- **m**: Public input chosen by the attacker
- **O**: Observation e.g. timing as a real number
A simple example

```plaintext
for i = 0 to n - 1 do
    if k[i] ≠ m[i] then g()
done
```
A simple example

\[ t(k,m) = \sum_{i=1}^{n} k[i] \oplus m[i] = \text{nb of bits where } k \text{ and } m \text{ differ} \]

Hamming distance

```
for i = 0 to n - 1 do
  if k[i] ≠ m[i] then g()
done
```
Aggregation of information

potential values of the long-term secret

\[ t(k,m) = \sum_{i=1}^{3} k[i] \oplus m[i] \]

Hamming distance
Aggregation of information

Hamming distance

\[ t(k,m) = \sum_{i=1}^{3} k[i] \oplus m[i] \]

Potential values of the long-term secret

\[ 000 \rightarrow o = t(k,000) \in \{0,1,2,3\} \]
Aggregation of information

potential values of the long-term secret

\[ t(k,m) = \sum_{i=1}^{3} k[i] \oplus m[i] \]

Hamming distance

000 \quad \leftrightarrow \quad o = t(k,000) \in \{0,1,2,3\}

\Rightarrow k \in \{ k' \mid t(k',000) = o \}
Aggregation of information

potential values of the long-term secret

\[ t(k, m) = \sum_{i=1}^{3} k[i] \oplus m[i] \]

Hamming distance

\[ 000 \quad \rightarrow \quad o = t(k, 000) \in \{0,1,2,3\} \]

\[ \Rightarrow \quad k \in \{ k' \mid t(k',000) = o \} \]
Aggregation of information

Hamming distance

\[ t(k,m) = \Sigma_{i=1}^{3} k[i] \oplus m[i] \]

Potential values of the long-term secret

\[ t(k,000) \in \{0,1,2,3\} \]

\[ 000 \rightarrow o = t(k,000) \in \{0,1,2,3\} \]

\[ \Rightarrow k \in \{ k' \mid t(k',000) = o \} \]

\[ 001 \rightarrow o' = t(k,001) \]
Aggregation of information

Hamming distance

$$t(k,m) = \sum_{i=1}^{3} k[i] \oplus m[i]$$

potential values of the long-term secret

$$k$$

$$t(k,000) \in \{0,1,2,3\}$$

$$000 \mapsto o = t(k,000) \in \{0,1,2,3\}$$

$$k \in \{ k' \mid t(k',000) = o \}$$

$$001 \mapsto o' = t(k,001)$$
Aggregation of information

potential values of the long-term secret

\[
t(k,m) = \sum_{i=1}^{3} k[i] \oplus m[i]
\]

Hamming distance

\[
\begin{align*}
000 & \quad \mapsto \quad o = t(k,000) \in \{0,1,2,3\} \\
\Rightarrow & \quad k \in \{ k' \mid t(k',000) = o \} \\
001 & \quad \mapsto \quad o' = t(k,001) \\
010 & \quad \mapsto \quad o'' = t(k,010)
\end{align*}
\]
Aggregation of information

Hamming distance

\[ t(k,m) = \Sigma_{i=1}^{3} k[i] \oplus m[i] \]

potential values of the long-term secret

\[ t(k,000) \in \{0,1,2,3\} \]

\[ t(k,001) \]

\[ t(k,010) \]

\[ t(k,010) \]

\[ 000 \quad \quad \quad \Rightarrow \quad \quad \quad \quad \quad o = t(k,000) \in \{0,1,2,3\} \]

\[ k \in \{ k' \mid t(k',000) = o \} \]

\[ 001 \quad \quad \quad \Leftarrow \quad \quad \quad \quad \quad o' = t(k,001) \]

\[ 010 \quad \quad \quad \Leftarrow \quad \quad \quad \quad \quad o'' = t(k,010) \]
Aggregation of information

potential values of the long-term secret

k
Aggregation of information

potential values of the long-term secret

Compute this equivalence relation over the set of secrets

static approach (security bounds)
Aggregation of information

Compute this equivalence relation over the set of secrets

- **static approach**
  - (security bounds)

Given an oracle to $t(k, \cdot)$, retrieve the class enclosing $k$

- **dynamic approach**
  - (attacks)
A more practical model for timing leakage
Program

Long-term secret

Two public inputs

\( k \)

\( m_1, m_2 \)

\( O_1 - O_2 \)

Difference of timings
Differential measurements

Less powerful attacker, but...

Closer to the models used in actual attack research

Compositionality
Compositionality for differential measurements
Compositional attacks

recovering $k$?

with oracle to execution time $m \mapsto t(k, m)$
Compositional attacks

```plaintext
for i = 0 to n – 1 do
  if k[i] ≠ m[i] then g()
done
```

recovering k?

*with oracle to execution time* \( m \mapsto t(k,m) \)

\( t(k,0) - t(k,2^i) \)
Compositional attacks

recovering $k$?

with oracle to execution time $m \mapsto t(k,m)$

```plaintext
for i = 0 to n - 1 do
  if $k[i] \neq m[i]$ then $g()$
done
```

if $t(k,0) < t(k,2^i)$
then $K := K \cap \{ k | k[i] = 1 \}$
else $K := K \cap \{ k | k[i] = 0 \}$

Exploiting the $i^{th}$ iteration
Sequential composition

1. $x = m$
2. $\textbf{for } i = 0 \textbf{ to } n - 1 \textbf{ do}$
3. $x = f_i(k,x)$
4. $\textbf{if } \text{Test}_i(k,x) = 1 \textbf{ then } g()$
5. $\textbf{done}$
Sequential composition

```
1  x = m
2  for i = 0 to n - 1 do
3      x = f_i(k,x)
4      if Test_i(k,x) = 1 then g()
5  done
```

**Goal:** writing this code under the form

```
p = p_0; p_2; ... ; p_{n-1}
```
Sequential composition

Goal: writing this code under the form

\[ p = p_0; p_2; \ldots; p_{n-1} \]

\( p_i \) computes \( f_i : K \times M \rightarrow M \) with execution time \( Test_i : K \times M \rightarrow \{0,1\} \)
Sequential composition

\[ p_{\text{comp}} = p_1 ; p_2 \]
Sequential composition

\[ p_{\text{comp}} = p_1 \circ p_2 \]

\( p_\ell \) computes \( f_\ell : K \times M \rightarrow M \) with execution time \( t_\ell : K \times M \rightarrow O \)
Sequential composition

$p_{\text{comp}} = p_1 ; p_2$

$p_\ell$ computes $f_\ell : K \times M \rightarrow M$ with execution time $t_\ell : K \times M \rightarrow O$

$f_{\text{comp}} = f_2 \circ f_1$

States are composed
Sequential composition

composition of public values, i.e. \( (f \circ g)(k,m) = f(k, g(k,m)) \)

\[ f_{\text{comp}} = f_2 \circ f_1 \]

States are composed

\[ p_{\text{comp}} = p_1 ; p_2 \]

\( p_\ell \) computes \( f_\ell : K \times M \rightarrow M \) with execution time \( t_\ell : K \times M \rightarrow O \)
Sequential composition

Composition of public values, i.e. \((f \circ g)(k,m) = f(k, g(k,m))\)

- States are composed:
  \[ f_{\text{comp}} = f_2 \circ f_1 \]

- Timings are summed:
  \[ t_{\text{comp}} = t_1 + (t_2 \circ f_1) \]

\[ p_{\text{comp}} = p_1 ; p_2 \]

\( p_\ell \) computes \( f_\ell : K \times M \to M \) with execution time \( t_\ell : K \times M \to O \)
Key hypothesis: independence
Key hypothesis: independence

**Hypotheses**

- $t, t'$ timing functions

**Theorem**

$$\text{Leak}(t+t') = \text{Leak}(t) \cap \text{Leak}(t')$$
**Key hypothesis: independence**

--- **Hypotheses** ---

- $t, t'$ timing functions

--- **Theorem** ---

$$\text{Leak}(t + t') = \text{Leak}(t) \cap \text{Leak}(t')$$

$\text{Leak}(t) = $ the equivalence relation on secrets characterising timing leakage
Key hypothesis: independence

--- Hypotheses ---

- \( t, t' \) timing functions
- \( X \) distribution of public inputs
- for all secrets \( k, k' \), the distributions \( t(k, X) \) and \( t'(k', X) \) are independent

--- Theorem ---

\[
\text{Leak}(t + t') = \text{Leak}(t) \cap \text{Leak}(t')
\]

\( \text{Leak}(t) \) = the equivalence relation on secrets characterising timing leakage
Randomised compositional attack
Randomised compositional attack

--- Inputs -----------------------------

**independent** blocks $p_1 = (f_1, t_1), ..., p_n = (f_n, t_n)$

**oracle** to $t(k^*, .)$ execution time of $(p_1; ...; p_n)$

*for some $k^*$*
Randomised compositional attack

--- Inputs --------------------------

**independent** blocks $p_1 = (f_1, t_1), \ldots, p_n = (f_n, t_n)$

**oracle** to $t(k^*, .)$ execution time of $(p_1; \ldots; p_n)$
  
  *for some $k^*$

--- Output --------------------------

equivalence class of $k^*$ in $\text{Leak}(t)$
Randomised compositional attack

**Inputs**
- **independent** blocks $p_1 = (f_1, t_1), \ldots, p_n = (f_n, t_n)$
- **oracle** to $t(k^*, .)$ execution time of $(p_1; \ldots; p_n)$ for some $k^*$

**Output**
equivalence class of $k^*$ in $\text{Leak}(t)$

**Algorithm**
1. $K := $ set of all secrets
2. $M := $ sample of $r$ random messages
3. for $i = 1$ to $n$ do
   - $K := K \cap \text{Attack}(\overline{t}_i | K \times M)$
4. done
5. return $K$
Randomised compositional attack

Inputs

- independent blocks $p_1 = (f_1, t_1), ..., p_n = (f_n, t_n)$
- oracle to $t(k^*, .)$ execution time of $(p_1; ...; p_n)$ for some $k^*$

Output

- equivalence class of $k^*$ in $\text{Leak}(t)$

Algorithm

$K := \text{set of all secrets}$

$M := \text{sample of } r \text{ random messages}$

for $i = 1$ to $n$ do

$K := K \cap \text{Attack}(\bar{t}_i | K \times M)$

done

return $K$

Timing attack on $\bar{t}_i = t_i \circ f_{i-1} \circ ... \circ f_1$ with oracle to $t(k^*, .)$
Applications
Cost analysis

for simple bit-serial operations, $n$ bits

<table>
<thead>
<tr>
<th>Method</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brute force</td>
<td>$O(2^n)$ measurements</td>
</tr>
<tr>
<td>Random attack</td>
<td>$O(n \log(n/\varepsilon))$ random measurements (to guarantee proba of success $1 - \varepsilon$)</td>
</tr>
</tbody>
</table>
## Cost analysis

For simple bit-serial operations, $n$ bits

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bruteforce</td>
<td>$O(2^n)$ measurements</td>
</tr>
<tr>
<td>Random. attack</td>
<td>$O(n \log(n/\varepsilon))$ random measurements</td>
</tr>
</tbody>
</table>

(to guarantee proba of success $1 - \varepsilon$)
Explaining documented attacks

as instances of the randomised attack VS independent blocks
Explaining documented attacks
as instances of the randomised attack VS independent blocks

1998 on RSA (Dhem et al.)

**Targets**: implem. of modular exponentiation with Montgomery multiplications

**Exploits**: timing variations of squaring operations

**Extracts**: all bits of the secret exponent but one
Explaining documented attacks
as instances of the randomised attack VS independent blocks

1998 on RSA (Dhem et al.)

**Targets:** implem. of modular exponentiation with Montgomery multiplications

**Exploits:** timing variations of squaring operations

**Extracts:** all bits of the secret exponent but one

**Decomposition:**
1 block = 1 multiplication
Explaining documented attacks
as instances of the randomised attack VS independent blocks

2007 on AES (Aciicmez et al.)

Targets: implem. of AES with precomputed tables

Exploits: timing variations due to cache

Extracts: all bits of the encryption key

Decomposition:
1 block = 1 table lookup
Conclusion
Conclusion

A formal model for reasoning about timing attacks
Conclusion

A formal model for reasoning about timing attacks

Compositionality results
Conclusion

A formal model for reasoning about timing attacks

- Compositionality results
- Generic description of attacks / cost analysis
Conclusion

A formal model for reasoning about timing attacks

- Compositionality results
- Generic description of attacks / cost analysis
- Captures several documented attacks
Conclusion

A formal model for reasoning about timing attacks

- Compositionality results
- Generic description of attacks / cost analysis
- Captures several documented attacks
- Future: use as a basis for automating attack synthesis