Timing Leaks and Coarse-Grained Clocks

Panagiotis Vasilikos
Flemming Nielson
Hanne Riis Nielso

Boris Köpf

DTU Technical University of Denmark

Microsoft Research Cambridge
Timing-Channel Attacks

Common countermeasures that refine the victim’s system:
- Constant-time software bucketing
- Randomized delays...

The drawback:
- Performance overhead

Victim

secret

\[ \begin{array}{c}
  i_1 \\
  i_2
\end{array} \]

0101

1101

observation

Adversary

time

\[ t_1 \]

\[ t_2 \]
Timing-Channel Attacks

Common countermeasures that *refine the victim’s system*:

- constant-time software
- bucketing
- randomized delays
- ...
Timing-Channel Attacks

Common countermeasures that refine the victim’s system:

- constant-time software
- bucketing
- randomized delays
- ...

The drawback: performance overhead
Reducing Clock Resolution

A countermeasure which **configures** the clock

![Diagram showing clock resolution configuration](image.png)
Reducing Clock Resolution

A countermeasure which **configures the clock**

About it:

- **no** performance overhead
- **local** scope (i.e., does not work for remote adversaries)
- has been **deployed in** major **browser** implementations
Reducing Clock Resolution

A countermeasure which **configures the clock**

---

About it:

- **no** performance overhead
- **local** scope (i.e. does not work for remote adversaries)
- has been **deployed in major browser implementations**

The drawback: it can be **bypassed**, using **timing techniques**
This work’s contributions

We propose

- The first **information theoretic framework** for adversaries with coarse-grained clocks.
This work’s contributions

We propose

- The first information theoretic framework for adversaries with coarse-grained clocks.

Based on this we derive the following:

- A coarse-grained clock might leak more information than a fine-grained one.
This work’s contributions

We propose

- The first information theoretic framework for adversaries with coarse-grained clocks.

Based on this we derive the following:

- A coarse-grained clock might leak more information than a fine-grained one.
- Conditions under which a coarse-grained clock imply better security.
This work’s contributions

We propose

- The first information theoretic framework for adversaries with coarse-grained clocks.

Based on this we derive the following:

- A coarse-grained clock might leak more information than a fine-grained one.

- Conditions under which a coarse-grained clock imply better security.

- A new timing technique.

- The timing techniques form a strict hierarchy in terms of information leakage.
The victim is described by:

- a finite set of secrets \( I \), and
- the family of timed automata \( S = (TA_i)_{i \in I} \)
Modelling Part: the Victim

The victim is described by:
- a finite set of secrets $I$, and
- the family of timed automata $S = (TA_i)_{i \in I}$

Guarded edges over real-valued variables

\[ q_s \xrightarrow{x \geq 2 \land y < 3} q_t \]

Transitions

\[ \langle q_s, [x \mapsto 3.4, y \mapsto 0] \rangle \xrightarrow{1.32, e_1} \langle q_t, [x \mapsto 0, y \mapsto 1.32] \rangle \]

Computations of the victim

\[ \rho = \langle q_0, \delta_0 \rangle \xrightarrow{e_1} \ldots \xrightarrow{e_n} \langle q_n, \delta_n \rangle \xrightarrow{e_{n+1}, e_{n+1}} \ldots \]
The victim is described by:

- a finite set of secrets $I$, and
- the family of timed automata $S = (TA_i)_{i \in I}$
Modelling Part: the Victim

The victim is described by:
- a finite set of secrets $I$, and
- the family of timed automata $S = (\text{TA}_i)_{i \in I}$

The system $S$ can be either
- deterministic (i.e. for each $i$, we have a unique computation),
- or stochastic (i.e. at each transition we first choose randomly a delay and then an edge)
The adversary is described by:

- a clock $c$ of grain $g$: $c(t) = \left\lfloor \frac{t}{g} \right\rfloor \cdot g$
The adversary is described by:

- a clock $c$ of grain $g$: $c(t) = \lfloor \frac{t}{g} \rfloor \cdot g$
- a finite set $E_{\text{pub}}$ of public edges
Modelling Part: the Adversary

The adversary is described by:

- a clock \( c \) of grain \( g \): \( c(t) = \left\lfloor \frac{t}{g} \right\rfloor \cdot g \)
- a finite set \( E_{\text{pub}} \) of public edges
- a finite number of observations \( k \)

\[ \rho = \langle q_0, \delta_0 \rangle \xrightarrow{t_1,e_1} \ldots \xrightarrow{t_{j_1},e_{j_1}} \langle q_n, \delta_n \rangle \xrightarrow{t_{j_1+1},e_{j_1+1}} \ldots \xrightarrow{t_{j_k},e_{j_k}} \ldots \]
Modelling Part: the Adversary

The adversary is described by:

- a clock \( c \) of grain \( g \): \( c(t) = \left\lfloor \frac{t}{g} \right\rfloor \cdot g \)
- a finite set \( E_{\text{pub}} \) of public edges
- a finite number of observations \( k \)

The view of the adversary on the computation \( \rho \) is

\[ \text{view}_c(\rho) = (c(t_{j_1}), ..., c(t_{j_k})) \]
A Counterexample on the Security of Coarse-Grained Clocks

Take deterministic function $f$ with inputs $i_1$, $i_2$ and timings 2, 3 resp.

- Scenario (a), the adversary has a clock $c$ of grain 2. In both cases of $i_1$ and $i_2$ the adversary sees $c(2) = 2 = c(3)$.

Proposition 1
Increasing the grain of the clock may result to more information leakage.

Theorem 2 (Multiple-g Security)
In deterministic systems, increasing the grain $g$ to a multiple $g' = mg$ results always to less or equal information leakage.
A Counterexample on the Security of Coarse-Grained Clocks

Take deterministic function $f$ with inputs $i_1$, $i_2$ and timings 2, 3 resp.

- Scenario (a), the adversary has a clock $c$ of grain 2. In both cases of $i_1$ and $i_2$ the adversary sees $c(2) = 2 = c(3)$.
- Scenario (b), the adversary has a clock $c$ of grain 3. In the case of $i_1$ the adversary sees $c(2) = 0$, whereas for $i_2$ it sees $c(3) = 3$ and thus $c(3) \neq c(2)$. 

Proposition 1: Increasing the grain of the clock may result to more information leakage.

Theorem 2 (Multiple-g Security): In deterministic systems, increasing the grain $g$ to a multiple $g' = mg$ results always to less or equal information leakage.
A Counterexample on the Security of Coarse-Grained Clocks

Take deterministic function $f$ with inputs $i_1$, $i_2$ and timings 2, 3 resp.

- Scenario (a), the adversary has a clock $c$ of grain 2. In both cases of $i_1$ and $i_2$ the adversary sees $c(2) = 2 = c(3)$.
- Scenario (b), the adversary has a clock $c$ of grain 3. In the case of $i_1$ the adversary sees $c(2) = 0$, whereas for $i_2$ it sees $c(3) = 3$ and thus $c(3) \neq c(2)$.

**Proposition 1**

Increasing the grain of the clock may result to more information leakage.
A Counterexample on the Security of Coarse-Grained Clocks

Take deterministic function $f$ with inputs $i_1, i_2$ and timings 2, 3 resp.

- Scenario (a), the adversary has a clock $c$ of grain 2. In both cases of $i_1$ and $i_2$ the adversary sees $c(2) = 2 = c(3)$.
- Scenario (b), the adversary has a clock $c$ of grain 3. In the case of $i_1$ the adversary sees $c(2) = 0$, whereas for $i_2$ it sees $c(3) = 3$ and thus $c(3) \neq c(2)$.

**Proposition 1**
Increasing the grain of the clock may result to more information leakage.

**Theorem 2 (Multiple-g Security)**
In deterministic systems, increasing the grain $g$ to a multiple $g' = m \cdot g$ results always to less or equal information leakage.
Leakage Analysis Part: Quantitative Information Flow

The common set-up contains
The common set-up contains

- a probability distribution $p$ on the set of secrets $I$
The common set-up contains

- a probability distribution \( p \) on the set of secrets \( I \)
- a set of outputs \( O \)
Leakage Analysis Part: Quantitative Information Flow

The common set-up contains
- a probability distribution \( p \) on the set of secrets \( I \)
- a set of outputs \( O \)
- an information-channel \( TC \)

\[
\begin{array}{ccc}
\text{TC} & o_1 & \ldots & o_m \\
i_1 & \frac{1}{2} & \ldots & \frac{1}{2} \\
\ldots & 1 & \ldots \\
i_n & 0 & \ldots & 1 \\
\end{array}
\]

The initial uncertainty of the adversary is \( H(p) \) (e.g., Shannon-, min-, \( g \)-entropy) and his posterior uncertainty is \( H(p; TC) \) (e.g., conditional Shannon-, min-, \( g \)-entropy).

The leakage is defined as:

\[
\text{Leakage}(p; TC) = H(p) - H(p; TC)
\]
Leakage Analysis Part: Quantitative Information Flow

The common set-up contains

- a probability distribution $p$ on the set of secrets $I$
- a set of outputs $O$
- an information-channel $TC$

$$
\begin{array}{ccc}
TC & o_1 & \ldots & o_m \\
i_1 & \frac{1}{2} & \ldots & \frac{1}{2} \\
\ldots & 1 & \ldots \\
i_n & 0 & \ldots & 1
\end{array}
$$

The initial uncertainty of the adversary is

$$H(p) \quad (\text{e.g. } H(p) \text{ could be Shannon-, min-, g-entropy})$$
Leakage Analysis Part: Quantitative Information Flow

The common set-up contains

- a probability distribution $p$ on the set of secrets $I$
- a set of outputs $O$
- an information-channel $TC$

<table>
<thead>
<tr>
<th>$TC$</th>
<th>$o_1$</th>
<th>...</th>
<th>$o_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_1$</td>
<td>$\frac{1}{2}$</td>
<td>...</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>...</td>
<td>1</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$i_n$</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

The initial uncertainty of the adversary is

$$H(p)$$

(e.g. $H(p)$ could be Shannon-, min-, g-entropy)

and his posterior uncertainty is

$$H(p, TC)$$

(e.g. $H(p, TC)$ could be conditional Shannon-, min-, g-entropy)
Leakage Analysis Part: Quantitative Information Flow

The common set-up contains

- a probability distribution \( p \) on the set of secrets \( I \)
- a set of outputs \( O \)
- an information-channel TC

\[
\begin{array}{c|c|c}
TC & o_1 & \cdots & o_m \\
\hline
i_1 & \frac{1}{2} & \cdots & \frac{1}{2} \\
\vdots & 1 & \cdots & \vdots \\
i_n & 0 & \cdots & 1 \\
\end{array}
\]

The initial uncertainty of the adversary is

\[ H(p) \quad \text{(e.g. } H(p) \text{ could be Shannon-, min-, g-entropy)} \]

and his posterior uncertainty is

\[ H(p, TC) \quad \text{(e.g. } H(p, TC) \text{ could be conditional Shannon-, min-, g-entropy)} \]

The leakage is defined

\[ \text{Leakage}(p, TC) = H(p) - H(p, TC) \]
Step 1. For each $i \in I$ let $O_i = \{\text{view}_c(\rho) | \rho \in \text{Runs}(\text{TA}_i)\}$.

The set of outputs is $O = \bigcup_i O_i$

Given an attack scenario $\text{AS} = (S, E_{\text{pub}}, c, k)$ we construct the timing channel $\text{TC}(\text{AS})$:

- **Step 1** is the output enumeration
Leakage Analysis Part: Attack Scenarios to Channels

**Step 1.** For each \( i \in I \) let \( O_i = \{ \text{view}_c(\rho) \mid \rho \in \text{Runs}(TA_i) \} \).

The set of outputs is \( O = \bigcup_i O_i \).

**Step 2.** Construct the timing channel \( TC(AS) : I \times O \mapsto [0, 1] \), and for \( i \in I \), and \( o \in O \):

- if \( S \) deterministic and \( o \in O_i \), then set \( TC(AS)(i, o) = 1 \).
- if \( S \) is stochastic and \( o \in O_i \), then set \( TC(AS)(i, o) = P_{\gamma q_i}(\text{view}^{-1}_c(o)) \).
- Otherwise, set \( TC(AS)(i, o) = 0 \).

Given an attack scenario \( AS = (S, E_{\text{pub}}, c, k) \) we construct the timing channel \( TC(AS) \):

- **Step 1** is the output enumeration
- **Step 2** is the actual construction, and for stochastic systems, it is based on the probability measure \( P_{\gamma} \).
Leakage Analysis Part: Attack Scenarios to Channels

**Step 1.** For each $i \in I$ let $O_i = \{\text{view}_c(\rho) \mid \rho \in \text{Runs}(TA_i)\}$.

The set of outputs is $O = \bigcup_i O_i$

**Step 2.** Construct the timing channel $TC(AS) : I \times O \mapsto [0,1]$, and for $i \in I$, and $o \in O$:

if $S$ deterministic and $o \in O_i$, then set $TC(AS)(i, o) = 1$.

if $S$ is stochastic and $o \in O_i$, then set $TC(AS)(i, o) = P_{\gamma q_o}(\text{view}_c^{-1}(o))$.

Otherwise, set $TC(AS)(i, o) = 0$.

Given an attack scenario $AS = (S, E_{\text{pub}}, c, k)$ we construct the timing channel $TC(AS)$:

- **Step 1** is the output enumeration
- **Step 2** is the actual construction, and for stochastic systems, it is based on the probability measure $P_{\gamma}$.

We showed that for an observation $o \in O$, the set $\text{view}_c^{-1}(o)$ is measurable with $P_{\gamma}$.
Timing Techniques

The set-up

- The victim runs a deterministic function $f$
- The adversary performs a constant-time operation for one, or more times after the execution of $f$, while it also makes queries to its clock.
Timing Techniques: The One-Pad

The **one-pad** technique

\[ t_{slow} \]

\[ t_{fast} \]
Timing Techniques: The One-Pad

The **one-pad** technique

\[
t_{\text{slow}} \quad t_{\text{slow}} + t_{\text{pad}}
\]

\[
t_{\text{fast}} \quad t_{\text{fast}} + t_{\text{pad}}
\]

Very effective on cache side-channel attacks.
Timing Techniques: The One-Pad

The one-pad technique

Very effective on cache side-channel attacks
The clock-edge technique (the learning phase)
The clock-edge technique (the learning phase)
The clock-edge technique (the learning phase)
The clock-edge technique (the learning phase)
The clock-edge technique (the learning phase)
The **clock-edge** technique (the **learning** phase)
The **clock-edge** technique (the **learning** phase)

The adversary learns:

\[ t_{pad} = \frac{g}{4} \]
The clock-edge technique (the attack phase)
The clock-edge technique (the attack phase)
The clock-edge technique (the attack phase)
The clock-edge technique (the attack phase)
The clock-edge technique (the attack phase)
The clock-edge technique (the attack phase)
The clock-edge technique (the attack phase)

The adversary learns

\[ c(t_f) + g = t_f + 3 \cdot t_{\text{pad}} \]
The clock-edge technique (the attack phase)

The adversary learns

\[ c(t_f) + g = t_f + 3 \cdot t_{pad} \]
\[ \Leftrightarrow t_f = c(t_f) + g - 3 \cdot t_{pad} \]
Basic idea behind the timing techniques:

- Distinguish $t_1$, $t_2$, when

$$
(c(t_1), c(t_1 + t_{pad}), ..., c(t_1 + m \cdot t_{pad})) 
\neq

(c(t_2), c(t_2 + t_{pad}), ..., c(t_2 + m \cdot t_{pad}))
$$
Basic idea behind the timing techniques:

- Distinguish $t_1$, $t_2$, when

$$ (c(t_1), c(t_1 + t_{pad}), ..., c(t_1 + m \cdot t_{pad})) $$

$$ \neq $$

$$ (c(t_2), c(t_2 + t_{pad}), ..., c(t_2 + m \cdot t_{pad})) $$

Question (1): How many times should I add my padding?

Question (2): Does the time $t_{pad}$ need to be fast?
We showed that:

\[(c(t_1), c(t_1 + t_{pad}), ..., c(t_1 + m \cdot t_{pad})) \neq (c(t_2), c(t_2 + t_{pad}), ..., c(t_2 + m \cdot t_{pad}))\]

iff

\[(c(t_1), c(t_1 + (t_{pad} \mod g)), ..., c(t_1 + (m \cdot t_{pad} \mod g))) \neq (c(t_2), c(t_2 + (t_{pad} \mod g)), ..., c(t_2 + (m \cdot t_{pad} \mod g)))\]
We showed that:

\[(c(t_1), c(t_1 + t_{pad}), ..., c(t_1 + m \cdot t_{pad})) \neq (c(t_2), c(t_2 + t_{pad}), ..., c(t_2 + m \cdot t_{pad}))\]

iff

\[(c(t_1), c(t_1 + (t_{pad} \mod g)), ..., c(t_1 + (m \cdot t_{pad} \mod g))) \neq (c(t_2), c(t_2 + (t_{pad} \mod g)), ..., c(t_2 + (m \cdot t_{pad} \mod g)))\]

**Question (1):** How many times should I add my padding?

**Answer:** \(g\) times.

**Question (2):** Does the time \(t_{pad}\) need to be fast?

**Answer:** Not always. \(t_{pad}\) needs to be co-prime with \(g\).
Timing Techniques: a Hierarchy

Theorem 2

\[ \text{TC}(\text{AS}_{1\text{-pad}}) \preceq \text{TC}(\text{AS}_{\text{clock-edge}}) \preceq \text{TC}(\text{AS}_{\text{co-prime}}) \]
Limitations and Solutions to them

Scalability issues:
- The set of outputs $O$ can be large
- the number of observations $k$ can be large
- the stochastic case involves calculations of the form

$$P(...) = \int_{t_1 \in C_1} \ldots \int_{t_n \in C_n} d\mu_n(t_n) \ldots d\mu_1(t_1)$$
Limitations and Solutions to them

Scalability issues:
- The set of outputs $O$ can be large
- The number of observations $k$ can be large
- The stochastic case involves calculations of the form

$$P(...) = \int_{t_1 \in C_1} \ldots \int_{t_n \in C_n} d\mu_n(t_n) \ldots d\mu_1(t_1)$$

However, there are cases where we can do better:
- In deterministic systems we have that $\min\text{-leakage} = \log|O|$. Any over-approximation $O^\# \supseteq O$ gives us a direct upper bound on the information leakage i.e. $\log|O| \leq \log|O^\#|$
- In stochastic systems with independent observations, calculating the channel for a single observation can be used to give us bounds on the information leakage for $k$ observations.
Conclusions

We performed the first principled information-flow analysis of timing leaks w.r.t. adversaries with clocks of reduced resolution. Our analysis relies on a novel translation of timed automata to information-theoretic channels, which we used to derive the following:

- A coarse-grained clock might leak more information than a fine-grained one.
- A sufficient condition for when increasing the grain we achieve better security.
- A new timing technique.
- The timing techniques form a strict hierarchy in terms of information leakage.
Questions?