Timing Leaks and Coarse-Grained Clocks

Panagiotis Vasilikos, Hanne Riis Nielson, Flemming Nielson
Technical University of Denmark, Lyngby
{*panva,hrni,fnie*}@dtu.dk

Boris Köpf
Microsoft Research, Cambridge
boris.koepf@microsoft.com

***Abstract*—Timing-based side-channel attacks have matured from an academic exercise to a powerful attack vector in the hand of real-world adversaries. A widely deployed countermeausure against such attacks is to reduce the accuracy of the clocks that are available to adversaries. While a number of high-profile attacks show that this mitigation can be side-stepped, there has not been a principled analysis of the degree of security it provides until now.**

In this paper, we perform the first information-flow analysis with respect to adversaries with coarse-grained clocks. To this end, we define an adversary model that is parametric in the granularity of the clock and connect it with a system model based on timed automata. We present algorithms for translating such a system to an information-theoretic channel, which enables us to analyze the leakage using standard techniques from quantitative information-flow analysis.

We use our techniques to derive insights about the effect of reducing clock resolution on security. In particular, (1) we show that a coarse-grained clock might leak more than a fine-grained one, (2) we give a sufficient condition for when increasing the grain of the clock we achieve better security, and (3) we show that the attack techniques used in the literature form a strict hierarchy in terms of the information an adversary can extract using them.

Finally, we illustrate the expressiveness of our development on a case study of a system that uses RSA signatures.

***Index Terms*—timing channels, timed automata, quantitative information flow**

I. Introduction

A timing channel is a mechanism which reveals information through the timing behaviour of a system. Timing channels in computer systems allow adversaries to obtain confidential information and hence pose an important threat to the system's security. In particular, information conveyed by timing channels has been used by adversaries to recover cryptographic keys, where the timing channel is built by measuring cryptographic or cache-dependent operations, and by malicious websites which correlate this information with the internal state of a victim who visits the website [1]–[7].

Defeating timing channels is important but challenging. There are two main approaches to defeating timing channels. The first approach relies on closing or mitigating the channel. Examples of this approach include:

- constant-time software [8], which is a coding discipline that forbids that branching decisions, memory access patterns, and variable-latency instructions depend on secrets.
- input blinding, which decorrelates the payload of cryptographic algorithms from the execution time, making it provably difficult for the adversary to recover the cryptographic key [9], [10].
- predictive mitigation techniques, which group timing observations into epochs of increasing duration, thereby provably reducing the amount of leaked information [11].

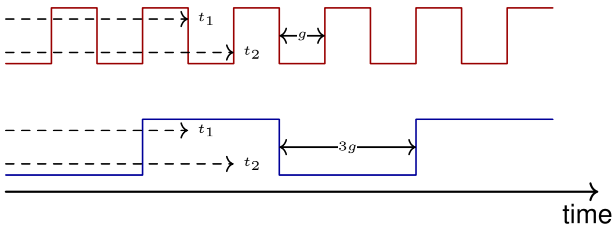


Fig. 1: The countermeasure of decreasing clock resolution.

The other approach relies on reducing the adversary's ability to make precise timing observations, e.g., by decreasing an adversary's clock resolution or removing the clock entirely. A clock is a counter that is being increased every after a fixed period. Increasing this period results in a clock with lower resolution, changing the perception of the adversary regarding the timing behaviour of the system, and hence makes it more difficult for it to build a timing channel.

This countermeasure is attractive because it does not imply performance overhead, while it works for adversaries that run on the same platform (i.e. not for remote attackers where one cannot control the clock). In particular, it has been proposed in the literature for mitigation of interrupt-related timing channels [12] and deployed in all major browser implementations after the first browser-based side-channel attacks [5].

Fig. 1 shows an example of the countermeasure. The clock on the top is being increased with a period of g, and it is able to distinguish between two events that arrive at times t_1 and t_2 resp. since at time t_1 the clock has been increased 4 times, while at t_2, 5 times. By increasing this period by a factor of $2 \cdot g$ (Fig. 1 bottom) the times of the two events become indistinguishable since in both cases the clock will be increased only once.

Unfortunately, low-resolution clocks are not an effective defense against many kinds of attacks. Several authors have successfully side-stepped this defense by building their own high-resolution clocks from primitives such as low-resolution clocks and simple counter processes [2], [13]–[15]. Therefore a principled analysis of this countermeasure and its security guarantees is needed.

Our approach. In this paper, we perform the first information-flow analysis w.r.t. adversaries with low-resolution clocks.

We define clock resolution based on the period of the clock, which we call *grain*. Our analysis relies on a notion of adversaries with clocks that is parametric in the clock's grain and on the number of timing observations they can make. We connect this adversary to a victim modeled as a timed automata [16], [17] system with either deterministic or stochastic time semantics. We show that the resulting model is expressive enough to capture the essential aspects of state-of-the-art attacks [2], [13]–[15], such as the clock-edge, and the one-pad, and we present a novel timing technique, which we call the co-prime technique.

In order to facilitate an information-flow analysis of the model, we present a novel algorithm that, given a system of timed automata and an adversary with a clock, constructs a channel matrix that represents the information that the adversary can extract from the system. The main challenge of the construction is computing the probabilities of the adversary's timing observations, since (stochastic) timed automata semantics are defined based on general probability measures. The construction enables us to leverage state-of-the-art techniques for quantitative information-flow analysis for formalizing and computing leakage to adversaries with fine-grained clocks. Using this approach, we derive the following insights:

- While it is well-known that coarse-grained clocks can be bypassed using attack techiniques such as the clock-edge and the one-pad, we show here that a coarse-grained clock might leak even more than a fine-grained clock without using any attack technique.
- We provide sufficient conditions for when a coarse-grained clock leaks less than a fine-grained one. In particular, we show that when the system is deterministic and we increase the grain of the clock by a multiple, the corresponding timing channel leaks less information.
- We show that the different techniques to construct fine-grained clocks, namely the one-pad technique, the clock-edge technique, and the co-prime technique, form a strict hierarchy in terms of the amount of information the adversary can extract.

Finally, we illustrate the expressiveness of our development on a stochastic system which implements an RSA encryption scheme to achieve data integrity.

Summary of contributions In summary, we perform 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 use to derive quantitative and qualitative insights into the effectiveness of existing attacks and countermeasures.

Organization of the paper. In Section 2, we recall some basics of timed automata, and in Section 3 we give the models of timed automata systems and adversaries with clocks. In Section 4, we recall the basics of quantitative information flow, and we present our algorithm for constructing timing channels of systems. In Section 5, we present our theoretical insights for the case of timing channels of deterministic systems, and the timing-techniques one-pad, clock-edge and co-prime. In Section 6, we perform our case study. Section 7, and 8 discusses related work and conclusions respectively. Finally, Appendix includes proofs of our results and some more detailed calculations of our case study.

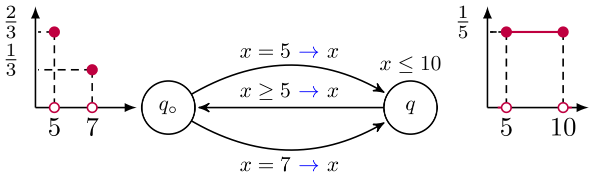


Fig. 2: An example of a (stochastic) timed automaton.

II. Timed Automata

In this section we recall some basics of timed automata.

We describe time with the set of non-negative real numbers $\mathbb{R}_{\geq 0}$. Timed automata [16], [17] are finite automata extended with real-valued variables called *dense clocks*, that are used to record the elapse of time.

Dense clocks are increased simultaneously, have infinite precision and can be reset. The transitions of the automaton are guarded with constraints over dense clocks, restricting the possible timing behaviour of the automaton.

Formally now, let $\mathbf{X}$ be a finite set of *dense clocks* taking values from $\mathbb{R}_{\geq 0}$. A *valuation* is a mapping $\delta : \mathbf{X} \mapsto \mathbb{R}_{\geq 0}$. Let now $\mathcal{D}(\mathbf{X})$ be the set of valuations over $\mathbf{X}$, then for a valuation $\delta \in \mathcal{D}(\mathbf{X})$ we have the following: if $t \in \mathbb{R}_{\geq 0}$ we write $\delta + t$ for the valuation that assigns $\delta(x) + t$ for all $x \in \mathbf{X}$ and if $X \in 2^{\mathbf{X}}$ is a set of dense clocks we write $\delta[X \mapsto 0]$ for the valuation that assigns the value 0 to all the dense clocks in X and leaves the rest of the them unchanged. A *guard* b is a finite conjunction of constraints of the form $x \,\mathsf{op}\, n$ where $\mathsf{op} \in \{<, \leq, =, \geq, >\}$, or the trivial constraint $\mathtt{tt}$, and x is a dense clock. Let now $\mathcal{G}(\mathbf{X})$ be the set of guards over $\mathbf{X}$. For a valuation δ and a guard b we write $\delta \models b$ whenever δ satisfies b in the usual way.

Definition 1. (**Timed Automaton**) A timed automaton $\mathsf{TA} = (\mathsf{Q}, q_\circ, \mathsf{E}, \mathsf{I})$ is quadruple where

- Q is a finite set of locations.
- $q_\circ$ is the initial location of the automaton.
- $\mathsf{E} \subseteq \mathsf{Q} \times \mathcal{G}(\mathbf{X}) \times 2^{\mathbf{X}} \times \mathsf{Q}$ is a finite set of guarded edges.
- $\mathsf{I} : \mathsf{Q} \mapsto \mathcal{G}(\mathbf{X})$ is a labelling function that imposes an invariant on each location.

The edges are annotated with actions and take the form $(q_s, b \rightarrow X, q_t)$ where $q_s \in \mathsf{Q}$ is the source location and $q_t \in \mathsf{Q}$ is the target location. The action $b \rightarrow X$ consists of a guard b that has to be satisfied in order for the dense clock variables X to reset. To cater for special cases we shall allow to omit the guard b when it is equal to $\mathtt{tt}$ and to omit the dense clock resets when X is empty.

The semantics of a timed automaton are given by a transition system whose configurations have the form $\langle q, \delta \rangle \in \textbf{Config} \subseteq \mathsf{Q} \times \mathcal{D}(\mathbf{X})$ and the transitions are described by an initial delay that increases the values of all the dense clocks followed by an action. Therefore, whenever $e = (q_s, b \rightarrow X, q_t)$ is in E we have the transition :

$$\langle q_s, \delta \rangle \xrightarrow{t,e} \langle q_t, \delta' \rangle \text{ if } \begin{cases} t \geq 0, \\ \delta + t \models \mathsf{I}(q_s) \wedge b, \\ \delta' = (\delta + t)[X \mapsto 0], \\ \delta' \models \mathsf{I}(q_t) \end{cases}$$

where t corresponds to the initial delay. The rule ensures that after the delay t the invariant and the guard are satisfied with the valuation $\delta + t$, and then updates the valuation $\delta + t$ to δ' by resetting the dense clocks in X. Finally, it ensures that the invariant is satisfied in the resulting configuration that uses the valuation δ'. The initial configuration of the automaton have all the dense clocks initialised to 0, and has the form $\langle q_\circ, \lambda x.0 \rangle$ where $\lambda x.0 \models \mathsf{I}(q_\circ)$, and we write $\gamma_{q_\circ}$ for $\langle q_\circ, \lambda x.0 \rangle$.

Definition 2. (**Run**). A *run* of a configuration $\gamma = \langle q, \delta \rangle \in \textbf{Config}$ is a sequence (possibly infinite)

$$\langle q_0, \delta_0 \rangle \xrightarrow{t_1,e_1} \dots \xrightarrow{t_n,e_n} \langle q_n, \delta_n \rangle \xrightarrow{t_{n+1},e_{n+1}} \dots$$

where $\langle q_0, \delta_0 \rangle = \langle q, \delta \rangle$ and we write $\mathsf{Runs}(\gamma)$ for the set of runs of γ.

Finally, a *run* of a timed automaton is described by a run $\rho \in \mathsf{Runs}(\gamma_{q_\circ})$ of the initial configuration $\gamma_{q_\circ}$ and we write $\mathsf{Runs}(\mathsf{TA})$ for $\mathsf{Runs}(\gamma_{q_\circ})$.

We now give our notion of determinism for timed automata

Definition 3. (**Deterministic timed automaton**). A timed automaton TA is deterministic whenever $\mathsf{Runs}(\mathsf{TA}) = \{\rho\}$ (for some run ρ).

Example 1. Fig. 2 depicts a timed automaton with two locations $q_\circ$ (the initial), q and a dense clock x. The automaton moves from $q_\circ$ to q after delaying 5 or 7 time units, and from q to $q_\circ$ after delaying for a time between 5 and 10. The dense clock x is reset after each move. Formally, we have that $\mathsf{TA} = (\mathsf{Q}, q_\circ, \mathsf{E}, \mathsf{I})$, where $\mathsf{Q} = \{q_\circ, q\}$, $\mathsf{E} = \{e_1, e_2, e_3\}$ where $e_1 = (q_\circ, x = 5 \rightarrow x, q)$, $e_2 = (q_\circ, x = 7 \rightarrow x, q)$, $e_3 = (q, x \geq 5 \rightarrow x, q_\circ)$ and the invariant mapping is $\mathsf{I} = [q_\circ \mapsto \mathsf{tt}, q \mapsto x \leq 10]$. A prefix of an example run of the automaton is

$$\begin{aligned} &\langle q_\circ, [x \mapsto 0] \rangle \xrightarrow{5,e_1} \langle q, [x \mapsto 0] \rangle \xrightarrow{5.5,e_3} \\ &\quad \langle q_\circ, [x \mapsto 0] \rangle \xrightarrow{7,e_2} \langle q, [x \mapsto 0] \rangle \xrightarrow{7.97,e_3} \langle q_\circ, [x \mapsto 0] \rangle \dots \end{aligned}$$

We define stochastic semantics for timed automata based on [18], [19]. Stochastic timed automata are stochastic processes, where at each transition the automaton first chooses randomly a delay and then an edge. We start by defining some auxiliary operators.

Let $\mathsf{TA} = (\mathsf{Q}, q_\circ, \mathsf{E}, \mathsf{I})$ be a timed automaton, then for a configuration $\gamma \in \textbf{Config}$ and an edge $e \in \mathsf{E}$ we define

$$\mathsf{Int}(\gamma, e) = \left\{ t \in \mathbb{R}_{\geq 0} \mid \exists \gamma' \in \textbf{Config} : \gamma \xrightarrow{t,e} \gamma' \right\}$$

to be the set of delays such that the edge e can be taken from γ after such a delay, and we write

$$\mathsf{Int}(\gamma) = \bigcup_{e \in \mathsf{E}} \mathsf{Int}(\gamma, e)$$

for the set of all possible delays that γ can perform.[1] Finally, for a configuration $\gamma \in \textbf{Config}$ we write

$$\mathsf{Enab}(\gamma) = \{ e \mid e \in \mathsf{E} : \mathsf{Int}(\gamma, e) \neq \emptyset \}$$

for the set of enabled edges of γ (i.e the ones that at least one transition is possible by taking them). We are now ready to give the definition of stochastic timed automaton.

Definition 4. (**Stochastic Timed Automata**.) Given a timed automaton $\mathsf{TA} = (\mathsf{Q}, q_\circ, \mathsf{E}, \mathsf{I})$, a *stochastic timed automaton* $\mathsf{STA} = (\mathsf{TA}, (\mu_\gamma)_{\gamma \in \textbf{Config}}, (\kappa_\gamma)_{\gamma \in \textbf{Config}})$ is a timed automaton equipped with the families $(\mu_\gamma)_{\gamma \in \textbf{Config}}$, $(\kappa_\gamma)_{\gamma \in \textbf{Config}}$ where

- for each configuration $\gamma \in \textbf{Config}$, $\mu_\gamma : \mathcal{B}(\mathbb{R}_{\geq 0}) \mapsto [0, 1]$ is a probability measure over the Borel[2] σ-algebra $\mathcal{B}(\mathbb{R}_{\geq 0})$ such that $\mu_\gamma(\mathsf{Int}(\gamma)) = 1$.
- for each configuration $\gamma \in \textbf{Config}$, $\kappa_\gamma : \mathsf{E} \mapsto [0, 1]$ is a probability distribution over the set of edges such that for all $e \in \mathsf{E} : \kappa_\gamma(e) > 0$ iff $e \in \mathsf{Enab}(\gamma)$, and we also have that $\sum_{e \in \mathsf{Enab}(\gamma)} \kappa(e) = 1$

For a run of the automaton now, at each transition (in the run) from a configuration $\langle q, \delta \rangle$, first, a delay t is chosen according to $\mu_{\langle q, \delta \rangle}$ and then an edge according to $\kappa_{\langle q, \delta + t \rangle}$.

Example 2. Back to Example 1, a stochastic version of the automaton is depicted in Fig. 2, where for configurations of the initial location $q_\circ$ the delay is chosen according to a (discrete) probability distribution that is expressed with a probability mass function (Fig. 2 left) that assigns $\frac{2}{3}$ and $\frac{1}{3}$ to the delays 5 and 7 resp., while for the configurations of the location q the delay is chosen according to a (continuous) uniform probability distribution that is expressed using the density function (Fig. 2 right) over the interval $[5, 10]$. In both cases after the delay has been chosen, there is exactly one enabled edge, that is chosen with probability 1.

III. Timed Systems and Adversaries with Clocks

In this section we give our models of systems of timed automata and adversaries with clocks. Systems of timed automata, or simply *timed systems*, will be used to model the timing behaviour of a system that operates on some secret provided by a victim. An adversary then tries to infer information of the victim's secret by measuring the timing behaviour of the system using a clock.

[1] As in [18], [19] we shall assume that for all $\gamma \in \textbf{Config}$, $\mathsf{Int}(\gamma) \neq \emptyset$, that is that the automaton is *deadlock-free*.

[2] The Borel set $\mathcal{B}(\mathbb{R}_{\geq 0})$ is the smallest σ-algebra generated by the open sets of $\mathbb{R}_{\geq 0}$.

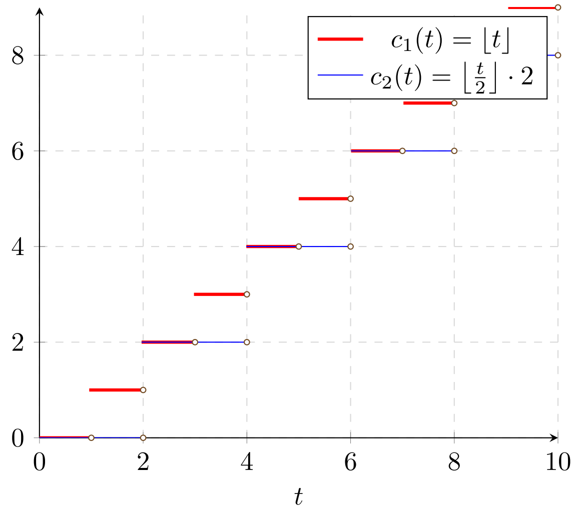


Fig. 3: Two clocks c_1 and c_2 with grains 1 and 2 respectively.

A. Timed Systems

Let I be a finite set of *secret* inputs of the victim. A *timed system* S is either a family of deterministic timed automata $(\mathsf{Q}_i, q^i_\circ, \mathsf{E}_i, \mathsf{I}_i)_{i \in I}$, or a family of stochastic timed automata $((\mathsf{Q}_i, q^i_\circ, \mathsf{E}_i, \mathsf{I}_i), (\mu_\gamma)^i_{\gamma \in \textbf{Config}}, (\kappa_\gamma)^i_{\gamma \in \textbf{Config}})_{i \in I}$, where for each $i \in I$, the automaton $\mathsf{TA}_i = (\mathsf{Q}_i, q^i_\circ, \mathsf{E}_i, \mathsf{I}_i)$ describes the timing behaviour of the system on input $i \in I$. In the first case the system will be called *deterministic* and in the second *stochastic*. We will also write $\mathsf{Q} = \bigcup_{i \in I} \mathsf{Q}_i$, $\mathsf{E} = \bigcup_{i \in I} \mathsf{E}_i$ and $\mathsf{Runs}(\mathsf{S}) = \bigcup_{i \in I} \mathsf{Runs}(\mathsf{TA}_i)$.

Example 3. Consider the input set $I = \{i_1, i_2\}$ and the stochastic system S, where for the input i_1 the behaviour of S is described by the stochastic timed automaton from Example 2. For input i_2 the behaviour of S is given by a variation of the stochastic timed automaton of Example 2, where now the probability measure over the delays for the configurations of the initial location is given by a discrete uniform distribution that is described by a probability mass function that assigns $\frac{1}{2}$ to both delays 5 and 7.

B. Clocks

The main tool of the adversary for building a timing channel from a timed system is a *clock*. A clock is a counter that is being increased after a constant period g, discretizing in this way the time domain $\mathbb{R}_{\geq 0}$ in buckets of size g.

This fixed period g between two consecutive increments of the clock is called its *grain* or *granularity*. The grain of the clock is the smallest time unit that it can measure. A point in time where a clock is being incremented is called a *clock-edge*, and this increment is equal to the clock's grain g.

Formally, a clock is given by the following definition

Definition 5. (**Clock.**) A *clock* $c : \mathbb{R}_{\geq 0} \mapsto \mathbb{N}$ with *granularity* or *grain* $g \in \mathbb{N}_{>0}$ is a step function over the time domain $\mathbb{R}_{\geq 0}$, where at time $t \in \mathbb{R}_{\geq 0}$ the value of c is

$$c(t) = \left\lfloor \frac{t}{g} \right\rfloor \cdot g$$

and $\lfloor . \rfloor$ is the floor function.

A fine-grained clock gives more precise measurements, and intuitively one could think that an adversary with such a clock is a bigger threat in comparison to one with a coarse-grained clock; however, we will show later that this is not always the case.

Example 4. Consider the clocks c_1 and c_2 of grain 1 and 2 respectively, given in Fig. 3. For the time points 0, 1.3, 2.5, 3.6... the value of c_1 is 0, 1, 2, 3... ,while for c_2 we have 0, 0, 2, 2,... For every clock-edge of c_2 we have two clock-edges of c_1. The clock-edges of c_1 happen at the time points 1, 2, 3,... while the ones of c_2 at the time points 2, 4, 6,...

We now give two facts, which will be later used in the proofs of our two main theorems (Theorem 3 and Theorem 4 resp.).

Fact 1. Let c_1 and c_2 be two clocks with grains g_1 and g_2 respectively. If g_2 is a multiple of g_1 then $\forall t_1, t_2 \in \mathbb{R}_{\geq 0} : (c_1(t_1) = c_1(t_2) \Rightarrow c_2(t_1) = c_2(t_2))$

Fact 2. Let c be a clock with grain g and $n \in \mathbb{N}$, then for $t_1, t_2 \in \mathbb{R}_{\geq 0}$: $c(t_1 + n) = c(t_2 + n) \Leftrightarrow c(t_1 + (n \bmod g)) = c(t_2 + (n \bmod g))$.

C. Adversaries with Clocks

Let S be a (deterministic or stochastic) system. We model the *view* of an adversary on the runs of a system as a function $\mathsf{view}_c : \mathsf{Runs}(\mathsf{S}) \mapsto O$ that maps runs to a finite set of sequence of observations $O \subseteq \mathbb{N}^+$, obtained by making timing measurements using a clock c.

In particular, for the system we assume a finite set of *public edges* $\mathsf{E}_{\text{pub}} \subseteq \mathsf{E}$ such that whenever the system performs a transition using this edge the adversary makes a timing observation using his clock (to be explained shortly).

We consider adversaries that make k (positive integer) number of timing observations and we also assume that each run of the system visits *at least* k times the public edges in E_{pub} (not necessarily all of them).

For a run $\rho = \gamma_0 \xrightarrow{t_1, e_1} \gamma_1 .... \xrightarrow{t_n, e_n} \gamma_n ... \in \mathsf{Run}(\mathsf{S})$, let $j_1, ..., j_k$ to be the unique ordered sequence of indices of the first k public edges appearing in ρ, we then have that the view of the adversary on ρ is given by

$$\mathsf{view}_c(\rho) = (c(t'_1), c(t'_2), ..., c(t'_k))$$

where for $i \in \{1, ..., k\}$, $t'_i = t_1 + ... + t_{j_i}$ is the time moment, when the adversary performs his i-th observation. We sometimes refer to the sequence t'_1,...,t'_k as the *k-time sequence* of ρ.

To wrap up everything from above we give the definition of an *attack scenario*.

Definition 6. (**Attack Scenario**). An attack scenario is a quadruple $\mathsf{AS} = (\mathsf{S}, \mathsf{E}_{\mathsf{pub}}, c, k)$ where S is a system, $\mathsf{E}_{\mathsf{pub}}$ are the public edges of it, c is the clock of the adversary, and k is the number of his timing observations.

Example 5. Consider the attack scenario $\mathsf{AS} = (\mathsf{S}, \mathsf{E}_{\mathsf{pub}}, c, k)$ where S is the stochastic system given in Example 3 over the input set $I = \{i_1, i_2\}$, $\mathsf{E}_{\mathsf{pub}} = \mathsf{E}$ is the set of observable edges of the system (i.e all the edges are observable), the adversary is using a clock c with grain $g = 5$, and performs k=2 timing observations.

Consider the following prefix of a run of the system that corresponds to the input i_1

$$\langle q_\circ, [x \mapsto 0] \rangle \xrightarrow{5, e_1} \langle q, [x \mapsto 0] \rangle \xrightarrow{5.5, e_3}$$
$$\langle q_\circ, [x \mapsto 0] \rangle \xrightarrow{7, e_2} \langle q, [x \mapsto 0] \rangle \xrightarrow{7.97, e_3} \langle q_\circ, [x \mapsto 0] \rangle ...$$

Since all of the edges are observable and the adversary makes $k = 2$ timing observations, the 2-time sequence of ρ is $t'_1 = 5$ and $t'_2 = t'_1 + 5.5 = 10.5$. For the values of the clock c at those two time points we have $c(t'_1) = 5$ and $c(t'_2) = 10$, and therefore the view of the adversary on ρ is

$$\mathsf{view}_c(\rho) = (c(t'_1), c(t'_2)) = (5, 10)$$

IV. Quantifying Leakage in Timed Systems

In this section, we give an algorithm that given an attack scenario constructs the corresponding timing channel. Based on the timing channel, one can then quantify the leakage of the timed system using standard measures from quantitative information-flow. We start by recalling some basics of quantitative information-flow.

A. Quantitative Information Flow

For the rest of this subsection we assume a random variable $\mathcal{I}$ with range the *secret* input space I of a timed system, a probability distribution $p_{\mathcal{I}}$ on I, and a random variable $\mathcal{O}$ with range the *public* set of timing observations O of the adversary. Our definitions are based on [20], [21] and all the logarithms have base two.

The threat of the adversary guessing the secret input with one try, before making any timing observation, is given by the min-vulnerability of $\mathcal{I}$ defined as

$$\mathsf{V}(p_{\mathcal{I}}) = \ \max_{i \in I} p_{\mathcal{I}}(i)$$

Min-vulnerability expresses that the adversary will choose for his guess the input that is more probable.

The relationship between the input space and the observations of the adversary is given by a timing channel $\mathsf{TC} : I \times O \mapsto [0, 1]$, that is a probability transition matrix, where for $i \in I$ and $o \in O$, $\mathsf{TC}(i, o)$ is the probability of $\mathcal{O} = o$ conditioned on $\mathcal{I} = i$ (i.e the conditional probability).

The expected probability of the adversary guessing the secret input, given his timing channel, is given by the conditional min-vulnerability of $\mathcal{I}$ and the timing channel TC, by

$$\mathsf{V}(p_{\mathcal{I}}, \mathsf{TC}) = \sum_{o \in O} \max_{i \in I} p_{\mathcal{I}}(i) \cdot \mathsf{TC}(i, o)$$

The min-vulnerability and the conditional min-vulnerability, can be turned into entropies by taking the negative logarithm of them [21].

For measuring the leakage of a timing channel we have the min-leakage [21]

$$\mathsf{L}_{\mathsf{min}}(p_{\mathcal{I}}, \mathsf{TC}) = \log \frac{\mathsf{V}(p_{\mathcal{I},\mathsf{TC}})}{\mathsf{V}(p_{\mathcal{I}})}$$

and the min-capacity

$$\mathsf{C}_{\mathsf{min}}(\mathsf{TC}) = \sup_{p_{\mathcal{I}}} \mathsf{L}_{\mathsf{min}}(p_{\mathcal{I}}, \mathsf{TC})$$

The min-capacity is the worst-case leakage and it is realised over a uniform prior $p_{\mathcal{I}}$ [20]. Based on min-leakage we can order timing channels as

Definition 7. (**Ordering on Channels**) Given a random variable $\mathcal{I}$ with range I, two random variables $\mathcal{O}_1$, $\mathcal{O}_2$ with range O_1 and O_2 resp., and the timing channels $\mathsf{TC}_1 : I \times O_1 \mapsto [0, 1]$ and $\mathsf{TC}_2 : I \times O_2 \mapsto [0, 1]$, we write

$$\mathsf{TC}_1 \preceq \mathsf{TC}_2 \quad \text{if} \quad \forall p_{\mathcal{I}} : \mathsf{L}_{\mathsf{min}}(p_{\mathcal{I}}, \mathsf{TC}_1) \leq \mathsf{L}_{\mathsf{min}}(p_{\mathcal{I}}, \mathsf{TC}_2)$$

A special case of a timing channel is a *deterministic* timing channel. A timing channel $\mathsf{TC} : I \times O \mapsto [0, 1]$ is deterministic whenever $\forall i \in I : \exists o \in O : \mathsf{TC}(i, o) = 1$ (i.e each row of the channel contains exactly one 1).

Recall [20], [21] that a deterministic channel $\mathsf{TC} : I \times O \mapsto [0, 1]$ gives rise to an equivalence relation (or partition) on I, given by

$$i_1 \equiv_{\mathsf{TC}} i_2 \quad \text{iff} \quad \exists o \in O : \mathsf{TC}(i_1, o) = 1 = \mathsf{TC}(i_2, o)$$

Two secrets are indistinguishable to the the adversary if and only if they give the same observation through the timing channel TC. For example if $\equiv_{\mathsf{TC}}$ is equal to $\top = I \times I$ then $\equiv_{\mathsf{TC}}$ describes no leakage since all the secrets are related. On the other hand if $\equiv_{\mathsf{TC}}$ is equal to the identity relation $\bot = \{(i, i) \mid i \in I\}$ we have that everything is leaked since each secret produces a unique observable. In any other case where the equivalence relation $\equiv_{\mathsf{TC}}$ is such that $\bot \subset \equiv_{\mathsf{TC}} \subset \top$, we have partial information about the secret.

Deterministic channels can be ordered based on their equivalence relation by partition refinement.

Definition 8. (**Partition Refinement**). Given deterministic channels $\mathsf{TC}_1 : I \times O_1 \mapsto [0, 1]$ and $\mathsf{TC}_2 : I \times O_2 \mapsto [0, 1]$ we write $\mathsf{TC}_1 \sqsubseteq \mathsf{TC}_2$ if the partition of TC_1 is refined by the partition of TC_2.

The following theorem from [20], [21] shows that the leakage ordering corresponds to the partition refinement ordering.

Theorem 1. Given deterministic channels $\mathsf{TC}_1 : I \times O_1 \mapsto [0, 1]$ and $\mathsf{TC}_2 : I \times O_2 \mapsto [0, 1]$ we have

$$\mathsf{TC}_1 \sqsubseteq \mathsf{TC}_2 \text{ iff } \mathsf{TC}_1 \preceq \mathsf{TC}_2$$

B. Timing Channels of Deterministic Systems

We now show how one can construct the timing channel of an attack scenario where the system is deterministic.

TABLE I: Algorithm for constructing the timing channel $\mathsf{TC}(\mathsf{AS})$ of an attack scenario $\mathsf{AS} = (\mathsf{S}, \mathsf{E}_{\text{pub}}, c, k)$

Step 1. For each $i \in I$ let $O_i = \{\mathsf{view}_c(\rho) \mid \rho \in \mathsf{Runs}(\mathsf{TA}_i)\}$.

Step 2. Let $\mathcal{I}$ to be a random variable with range the input set I and $\mathcal{O}$ to be a random variable with range $O = \bigcup_i O_i$. For the timing channel $\mathsf{TC}(\mathsf{AS}) : I \times O \mapsto [0, 1]$, and for $i \in I$, and $o \in O$:

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

if S is stochastic and $o \in O_i$, then set $\mathsf{TC}(\mathsf{AS})(i, o) = \mathsf{P}_{\gamma_{q^i_\circ}}(\mathsf{view}_c^{-1}(o))$.

Otherwise, set $\mathsf{TC}(\mathsf{AS})(i, o) = 0$.

Let $\mathsf{AS} = (\mathsf{S}, \mathsf{E}_{\text{pub}}, c, k)$ be an attack scenario where $\mathsf{S} = (\mathsf{TA}_i)_{i \in I}$ is a deterministic system. We construct the timing channel $\mathsf{TC}(\mathsf{AS})$ of AS using the algorithm given in TABLE I. The first step of the algorithm computes for each input $i \in I$, the set of its possible observations O_i. Since S is deterministic, $O_i = \{o\}$ is a singleton. All the observations of the system are described by the set $O = \bigcup_{i \in I} O_i$ and taking random variables $\mathcal{I}$ and $\mathcal{O}$ over the input set I and the observations O (resp.), we have the deterministic timing channel $\mathsf{TC}(\mathsf{AS}) : I \times O \mapsto [0, 1]$, that for input i and its unique observation o it returns 1, otherwise it returns 0. Notice that our algorithm is independent of our choice of min-leakage to be used as the measure for quantifying leakage.

C. Probability Measure for Stochastic Timed Automata

To explain the construction for the case of S being stochastic we need to define a probability measure on the runs of stochastic timed automata. We will then use this measure to compute the probabilities of the timing observations of an adversary.

Let $\mathsf{STA} = (\mathsf{TA}, (\mu_\gamma)_{\gamma \in \mathbf{Config}}, (\kappa_\gamma)_{\gamma \in \mathbf{Config}})$ be a stochastic timed automaton, we define a probability measure over the set of $\mathsf{Runs}(\gamma)$ for each $\gamma \in \mathbf{Config}$ as in [18], [19]. We start by giving some helpful definitions.

For an edge $e \in \mathsf{E}$ we write $\mathsf{source}(e) = q_s$ for its source location, and $\mathsf{target}(e) = q_t$ for its target location. A *path* $e_1....e_n$ $(n \geq 1)$ is a sequence of edges such that for all $i \in \{2, .., n\}$ we have that $\mathsf{source}(e_i) = \mathsf{target}(e_{i-1})$. For a path $\pi = e_1...e_n$, a configuration $\gamma \in \mathbf{Config}$ and a Borel set $\mathcal{C}$ of $\mathbb{R}^n_{\geq 0}$ $(n \geq 1)$ (i.e $\mathcal{C} \in \mathcal{B}(\mathbb{R}^n_{\geq 0})$) we define the set of $\mathcal{C}$-*constrained cylinders* of π as

$$\mathsf{Cyl}_{\mathcal{C}}(\gamma, \pi) = \\ \left\{ \gamma_0 \xrightarrow{t_1, e_1} \gamma_1 ... \gamma_{n-1} \xrightarrow{t_n, e_n} \gamma_n ... \in \mathsf{Runs}(\gamma) \mid (t_1, .., t_n) \in \mathcal{C} \right\}$$

that is the set of all runs of γ that go through the path $\pi = e_1...e_n$ and the time delays $t_1, ..., t_n$ satisfy the constrain $\mathcal{C}$.

For a path $\pi = e_1...e_n$, a configuration $\gamma \in \mathbf{Config}$ and a Borel set $\mathcal{C}$ of $\mathbb{R}^n_{\geq 0}$ $(n \geq 1)$ we define inductively the probability measure P_γ. For the base case where $\pi = e$ we have that

$$\mathsf{P}_\gamma(\mathsf{Cyl}_{\mathcal{C}}(\gamma, e)) = \int_{t \in \mathsf{Int}(\gamma, e)} \kappa_{\gamma + t}(e) \cdot 1_{\mathcal{C}}(t) \mathrm{d}\mu_\gamma(t)$$

where $\gamma + t$ is γ with its valuation having its dense clocks increased by t and $1_{\mathcal{C}} : \mathbb{R}_{\geq 0} \mapsto \{0, 1\}$ is the indicator function defined as

$$1_{\mathcal{C}}(t) = \begin{cases} 1 & \text{if } t \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases}$$

The domain of integration[3] is over all possible delays $t \in \mathsf{Int}(\gamma, e)$ that γ could make by choosing e and would result to a configuration $\gamma + t$. The function which is integrated then is the probability $\kappa_{\gamma + t}(e)$ of choosing e from $\gamma + t$, multiplied by $1_{\mathcal{C}}(t)$, ensuring that t satisfies $\mathcal{C}$.

For the inductive case, where $\pi = e_1...e_n$, we have

$$\mathsf{P}_\gamma(\mathsf{Cyl}_{\mathcal{C}}(\gamma, e_1...e_n)) = \\ \int_{t_1 \in \mathsf{Int}(\gamma, e_1)} \kappa_{\gamma + t_1}(e_1) \cdot \mathsf{P}_{\gamma'}(\mathsf{Cyl}_{\mathcal{C}^{t_1}}(\gamma', e_2...e_n)) \mathrm{d}\mu_\gamma(t_1)$$

where $\gamma \xrightarrow{t_1, e_1} \gamma'$, and

$$\mathcal{C}^{t_1} = \left\{ (t_2, ..., t_n) \in \mathbb{R}^{n-1}_{\geq 0} \mid (t_1, ..., t_n) \in \mathcal{C} \right\}$$

The explanation is similar to the base case, where now we also integrate over the probability of the constrained cylinder of the remaining path $e_2....e_n$, starting at the resulting configuration γ'.

The following theorem from [18], [19] shows that P_γ is a well-defined probability measure.

Theorem 2. *For a stochastic timed automaton STA and for each configuration $\gamma \in$ **Config**, P_γ is a probability measure over $(\mathsf{Runs}(\gamma), \mathcal{F})$ where $\mathcal{F}$ is the σ-algebra generated by the constrained cylinders of γ.*

Example 6. Consider the stochastic automaton of Example 2. For the constraint

$$\mathcal{C} = \left\{ (t_1, t_2) \in \mathbb{R}^2_{\geq 0} \mid (5 \leq t_1 < 10) \wedge (10 \leq t_1 + t_2 < 15) \right\}$$

and the path $\pi = e_1 e_3$ we want to compute the probability $P_{\gamma_{q_\circ}}(\mathsf{Cyl}_{\mathcal{C}}(\gamma_{q_\circ}, \pi))$ that is equal to

$$\int_{t_1 \in \mathsf{Int}(\gamma_{q_\circ}, e_1)} \kappa_{\gamma_{q_\circ} + t_1}(e_1) \cdot \mathsf{P}_{\gamma'}(\mathsf{Cyl}_{\mathcal{C}^{t_1}}(\gamma', e_3)) \, d\mu_{\gamma_{q_\circ}}(t_1) \quad (1)$$

Next, for $\gamma_{q_\circ} = \langle q_\circ, [x \mapsto 0] \rangle$ we have the discrete probability distribution $\mu_{\gamma_{q_\circ}}$ over the all possible delays of γ, $\mathsf{Int}(\gamma_{q_\circ}) = \{5, 7\}$, defined by the probability mass function p, where $p(5) = \frac{2}{3}$, and $p(7) = \frac{1}{3}$. We also have that $\mathsf{Int}(\gamma_{q_\circ}, e_1) = \{5\}$ and for the resulting (after a delay) configurations $\gamma_{q_\circ} + 5$ the probability of taking the edge e_1 is $\kappa_{\gamma_\circ + 5}(e_1) = 1$. Therefore (1) is equal to

$$p(5) \cdot \mathsf{P}_{\gamma'}(\mathsf{Cyl}_{\mathcal{C}^{t_1}}(\gamma', e_3)) \quad (2)$$

and since $t_1 \in \mathsf{Int}(\gamma_{q_\circ}, e_1) = \{5\}$ we have that

$$\mathcal{C}^{t_1} = \mathcal{C}^5 = \{t_2 \in \mathbb{R}_{\geq 0} \mid 10 \leq t_2 + 5 < 15\} = [5, 10)$$

Next, for the resulting configuration $\gamma' = \langle q, x \mapsto 0 \rangle$ we have the continuous uniform probability distribution $\mu_{\gamma'}$ over the

[3] Whenever μ_γ is discrete then the integration becomes summation instead.

all possible delays of γ', $\mathsf{Int}(\gamma') = [5, 10]$, defined by the probability density function $f(t) = \frac{1}{5} \cdot 1_{[5,10]}(t)$. For the resulting (after a delay) configurations $\gamma' + t$ he probability of taking the edge e_3 is $\kappa_{\gamma'+t}(e_3) = 1$. Therefore for $\mathsf{P}_{\gamma'}(\mathsf{Cyl}_{\mathcal{C}^5}(\gamma', e_3))$ we have

$$\begin{aligned}
\mathsf{P}_{\gamma'}(\mathsf{Cyl}_{\mathcal{C}^5}(\gamma', e_3)) &= \int_{t_2 \in \mathsf{Int}(\gamma', e_3)} \kappa_{\gamma'+t_2}(e_3) \cdot 1_{\mathcal{C}^5}(t_2) d\mu_{\gamma'}(t_2) \\
&= \int_{t_2 \in [5,10]} f(t) \cdot 1_{\mathcal{C}^5}(t_2) dt_2 \\
&= \int_{t_2 \in [5,10]} \frac{1}{5} \cdot 1_{[5,10]}(t_2) \cdot 1_{[5,10)}(t_2) dt_2 \\
&= \frac{1}{5} \cdot \int_{t_2 \in [5,10)} dt_2 = 1 \quad (3)
\end{aligned}$$

and thus using (1), (2) and (3) we have

$$\mathsf{P}_{\gamma_{q_\circ}}(\mathsf{Cyl}_{\mathcal{C}}(\gamma_{q_\circ}, e_1 e_3)) = \frac{2}{3} \cdot 1 = \frac{2}{3}$$

D. Timing Channels of Stochastic Systems

We now show how one can construct the timing channel of an attack scenario where the system is stochastic.

For an attack scenario $\mathsf{AS} = (\mathsf{S}, \mathsf{E_{pub}}, c, k)$ of a stochastic system $\mathsf{S} = (\mathsf{TA}_i, (\mu_\gamma)^i_{\gamma \in \textbf{Config}} (\kappa_\gamma)^i_{\gamma \in \textbf{Config}})_{i \in I}$ we construct the corresponding timing channel using TABLE I. The construction follows the same logic as the one for deterministic systems, where for each input $i \in I$ we need to enumerate all the possible observations (**Step 1** of TABLE I) of the adversary and then compute its probabilities (**Step 2** of TABLE I).

For deterministic systems, this process is straightforward, since each input is associated with exactly one observation, and consequently this observation has probability 1. However, this is not the case for stochastic systems.

Starting with **Step 1**, for an input $i \in I$, the set of possible observations $O_i = \{\mathsf{view}_c(\rho) \mid \rho \in \mathsf{Runs}(\mathsf{TA}_i)\}$ could turn to be infinite. To deal with this case (when needed) we assume that the clock c of the adversary has a limit (i.e this models that the clock has finite capacity). Now let g be the grain of c and $l = m.g$ (m is a natural number) its capacity. The modified clock $c_l : \mathbb{R}_{\geq 0} \mapsto \mathbb{N}$ is given by

$$c_l(t) = \min \left\{ l, \left\lfloor \frac{t}{g} \right\rfloor \cdot g \right\}$$

The modified clock c_l now behaves as the clock c for time points $t < l$, whereas for values greater or equal to l its value becomes constant. Here notice that the algorithm from TABLE I remains unchanged, but only the definition of the clock changes, so we can bound the set O_i.

Next, for **Step 2** we compute the probability of an observation o. First, we have that the runs which can result to the observation o, are described by the preimage $\mathsf{view}_c^{-1}(o)$, and consequently the probability of the observation o is equal to $\mathsf{P}_{\gamma_{q^i_\circ}}(\mathsf{view}_c^{-1}(o))$. To show that $\mathsf{view}_c^{-1}(o)$ is measurable our goal is to express it as a union of disjoint constrained cylinders. This will also give us a more algorithmic approach for computing the probability of o. We start by providing some auxiliary sets and operators.

For the rest, we fix an input $i \in I$ and let TA be its corresponding timed automaton where we omit the subscript i. Let

$$\begin{aligned}\textbf{Paths} = \textstyle\bigcup_{i=k}^{\infty} \{e_1 ... e_i \mid |\{j \mid e_j \in \mathsf{E_{pub}}\}| = k \,\wedge \\ \mathsf{source}(e_1) = q_\circ \wedge e_i \in \mathsf{E_{pub}}\}\end{aligned}$$

be the set of paths that start at the initial location $q_\circ$, contain exactly k public edges and the last edge is public. Each path in this set represents one or more (prefixes of) runs that could result to a k-sequence of timing observations.

Example 7. For the attack scenario of Example 5, we have that $k = 2$ and $\textbf{Paths} = \{e_1 e_3, e_2 e_3\}$

Now for a sequence of observations $o = (z_1, ..., z_k) \in O_i$ and a path $\pi = e_1, ..., e_n \in \textbf{Paths}$ we want to specify a constraint that describes the set of possible delays that could result to this particular sequence of observations taking this path. We thus define

$$\begin{aligned}&\mathcal{C}_{e_1 ... e_n}(z_1, ..., z_k) = \\ &\textstyle\bigcap_{i=1}^{k} \{(t_1, ..., t_n) \in \mathbb{R}^n_{\geq 0} \mid \{j \mid e_j \in \mathsf{E_{pub}}\} = \{j_1 ..., j_k\} \\ &\qquad\qquad \Rightarrow\; t_1 + ... + t_{j_i} \in c^{-1}(z_i)\}\end{aligned}$$

and notice here that $c^{-1}(z)$ (for $z \in \{z_1, ..., z_k\}$) is an interval.[4]

Example 8. For the path $\pi = e_1 e_3 \in \textbf{Paths}$ from Example 7 and the observation $o = (5, 10) \in O$ we have the constraint

$$\begin{aligned}\mathcal{C}_\pi(o) = &\{(t_1, t_2) \in \mathbb{R}^2_{\geq 0} \mid t_1 \in c^{-1}(5)\} \cap \\ &\{(t_1, t_2) \in \mathbb{R}^2_{\geq 0} \mid t_1 + t_2 \in c^{-1}(10)\}\end{aligned}$$

and since $c^{-1}(5) = [5, 10)$ and $c^{-1}(10) = [10, 15)$ we have that

$$\begin{aligned}\mathcal{C}_\pi(o) = \{(t_1, t_2) \in \mathbb{R}^2_{\geq 0} \mid\; &(5 \leq t_1 < 10) \wedge \\ &(10 \leq t_1 + t_2 < 15)\}\end{aligned}$$

Finally, this allows us to express $\mathsf{view}_c^{-1}(o)$ as a union of disjoint constrained cylinders as

$$\mathsf{view}_c^{-1}(o) = \bigcup_{\pi \in \textbf{Paths}} \mathsf{Cyl}_{\mathcal{C}_\pi(o)}(\gamma_{q_\circ}, \pi)$$

and thus the probability of the observation $o \in O_i$ is

$$\mathsf{P}_{\gamma_{q_\circ}}(\mathsf{view}_c^{-1}(o)) = \sum_{\pi \in \textbf{Paths}} \mathsf{P}_{\gamma_{q_\circ}}(\mathsf{Cyl}_{\mathcal{C}_\pi(o)}(\gamma_{q_\circ}, \pi))$$

The next example illustrates the construction of a timing channel for the case of the system being stochastic.

Example 9. We will now compute the timing channel of the attack scenario from Example 5.

The set of possible observations of the adversary is $O = \{(5, 10), (5, 15)\} = O_{i_1} = O_{i_2}$ and from Example 7, we have that **Paths**$=\{e_1 e_3, e_2 e_3\}$.

Next, let $\pi_1 = e_1 e_3$, and $\pi_2 = e_2 e_3$. For the input i_1 and the observation $o = (5, 10)$ we have that

[4] The same holds whenever we have a clock c_l with limit l.

$$\begin{aligned}\mathsf{view}_c^{-1}((5,10)) = {} & \mathsf{Cyl}_{\mathcal{C}_{\pi_1}(5,10)}(\gamma_{q_\circ^{i_1}}, \pi_1) \cup \\ & \mathsf{Cyl}_{\mathcal{C}_{\pi_2}(5,10)}(\gamma_{q_\circ^{i_1}}, \pi_2)\end{aligned}$$

and thus

$$\begin{aligned}\mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{view}_c^{-1}((5,10))) = {} & \mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{Cyl}_{\mathcal{C}_{\pi_1}(5,10)}(\gamma_{q_\circ^{i_1}}, \pi_1)) + \\ & \mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{Cyl}_{\mathcal{C}_{\pi_2}(5,10)}(\gamma_{q_\circ^{i_1}}, \pi_2))\end{aligned}$$

Note that in Example 6, we calculated the probability $\mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{Cyl}_{\mathcal{C}_{\pi_1}(5,10)}(\gamma_{q_\circ^{i_1}}, \pi_1)) = \frac{2}{3}$ and working similarly we can show that $\mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{Cyl}_{\mathcal{C}_{\pi_2}(5,10)}(\gamma_{q_\circ^{i_1}}, \pi_2)) = \frac{1}{5}$ and therefore we get that

$$\mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{view}_c^{-1}((5,10))) = \frac{2}{3} + \frac{1}{5} = \frac{13}{15}$$

We work similarly for the observation $o = (5,15)$ and we get

$$\begin{aligned}\mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{view}_c^{-1}((5,15))) = {} & \mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{Cyl}_{\mathcal{C}_{\pi_1}(5,15)}(\gamma_{q_\circ^{i_1}}, \pi_1)) + \\ & \mathsf{P}_{\gamma_{q_\circ^{i_1}}}(\mathsf{Cyl}_{\mathcal{C}_{\pi_2}(5,15)}(\gamma_{q_\circ^{i_1}}, \pi_2)) \\ = {} & 0 + \frac{2}{15} = \frac{2}{15}\end{aligned}$$

We repeat the process for the input i_2, and we obtain the following timing channel

$$\mathsf{TC}(\mathsf{AS})(i,o) = \begin{cases} \dfrac{13}{15} & \text{if } i = i_1 \text{ and } o = (5,10) \\[2ex] \dfrac{2}{15} & \text{if } i = i_1 \text{ and } o = (5,15) \\[2ex] \dfrac{8}{10} & \text{if } i = i_2 \text{ and } o = (5,10) \\[2ex] \dfrac{2}{10} & \text{otherwise} \end{cases}$$

V. Analysis of Timing Channels in Deterministic Systems

In this section, we start by analyzing the relationship between clock grain and leakage for deterministic systems. Next, we present timing techniques that have been used to bypass a low-resolution clock, we present a new timing technique, and we show how those techniques can be modelled in our framework. We finish by showing a result on the hierarchy of those techniques in terms of how much information can be extracted from the adversary.

A. Relating Clock Grain and Leakage

In our first result, we show that, contrary to popular belief a coarse-grained clock might leak more information than a fine-grained clock.

Proposition 1. There exists deterministic system S and attack scenarios AS_1, AS_2 of S with clocks c_1, c_2 resp., and grains g_1, g_2 with $g_1 < g_2$ and $\mathsf{TC}(\mathsf{AS}_1) \preceq \mathsf{TC}(\mathsf{AS}_2)$.

Note that Proposition 1 does not talk about the well-known bypassing techniques [2], [13]–[15] that have been used to side-step the defense of a coarse-grained clock; instead it shows that the security offered by a coarse-grained clock could be worse than the one offered by a fine-grained clock, even when bypassing techniques are not in use.

Proposition 1 follows from the following example

Example 10. Consider a deterministic system S, whose input set is $I = \{i_1, i_2\}$ and the system is given by two automata TA_{i_1} and TA_{i_2} who both have a single edge leaving their initial location $e_1 = (q_\circ, x = 2, q)$ (for TA_{i_1}), and $e_2 = (q'_\circ, x = 3, q')$ (for TA_{i_2}) controlled by a dense clock x.

For TA_{i_1} we have the run $\rho_1 = \gamma_{q_\circ} \xrightarrow{2,e_1} \gamma_1 \ldots$ and for TA_{i_2} we have the run $\rho_2 = \gamma_{q'_\circ} \xrightarrow{3,e_2} \gamma'_1 \ldots$. Next consider the two attack scenarios $\mathsf{AS}_1 = (S, \{e_1, e_2\}, c_1, 1)$ and $\mathsf{AS}_2 = (\mathsf{S}, \{e_1, e_2\}, c_2, 1)$ where the edges of interest are observable, the clock c_1 has grain $g_1 = 2$, the clock c_2 has grain $g_2 = 3$, and the adversary makes one timing observation. Following the algorithm from TABLE I, for AS_1 we have $O_{i_1} = \{\mathsf{view}_{c_1}(\rho_1)\} = \{c_1(2)\} = \{2\}$ and $O_{i_2} = \{\mathsf{view}_{c_1}(\rho_2)\} = \{c_1(3)\} = \{2\}$, whereas for AS_2 we have $O_{i_1} = \{\mathsf{view}_{c_2}(\rho_1)\} = \{c_2(2)\} = \{0\}$ and $O_{i_2} = \{\mathsf{view}_{c_2}(\rho_2)\} = \{c_2(3)\} = \{3\}$ and thus we get the timing channels

$$\mathsf{TC}(\mathsf{AS}_1)(i,o) = 1 \quad \mathsf{TC}(\mathsf{AS}_2)(i,o) = \begin{cases} 1 & \text{if } i = i_1 \\ & \text{and } o = 0 \\ 1 & \text{if } i = i_2 \\ & \text{and } o = 3 \\ 0 & \text{otherwise} \end{cases}$$

We then have that $\equiv_{\mathsf{TC}(\mathsf{AS}_1)} = \top$, whereas $\equiv_{\mathsf{TC}(\mathsf{AS}_2)} = \bot$, and thus $\mathsf{TC}(\mathsf{AS}_1) \sqsubseteq \mathsf{TC}(\mathsf{AS}_2)$. Next using Theorem 1 we get that $\mathsf{TC}(\mathsf{AS}_1) \preceq \mathsf{TC}(\mathsf{AS}_2)$ showing that the attack scenario where the clock has grain $g_2 = 3$ leaks more than the scenario where the clock has grain $g_1 = 2$.

Although, we showed that, in general, increasing the grain of the clock does not increase security, with our next theorem we provide sufficient coditions for when this actually happens.

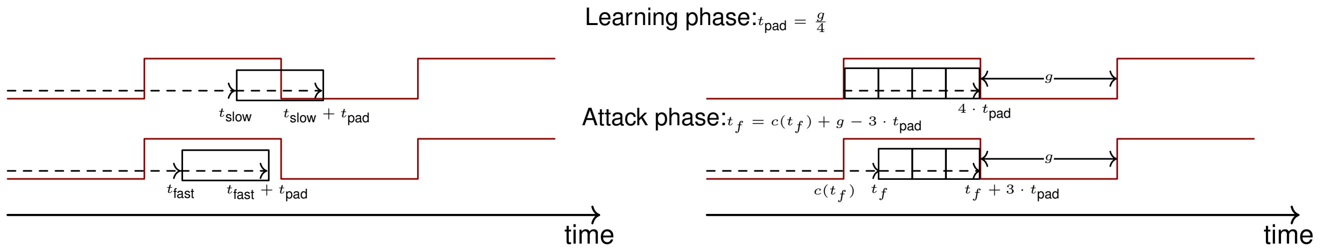
Theorem 3. (**Multiple-g security.**) Let $\mathsf{AS}_1 = (\mathsf{S}, \mathsf{E}_{\text{pub}}, c_1, k)$ and $\mathsf{AS}_2 = (\mathsf{S}, \mathsf{E}_{\text{pub}}, c_2, k)$ be two attack scenarios such that S is deterministic and the clocks c_1, c_2 have grains g_1, g_2 (resp.) and g_1 is a multiple of g_2. We then have that

$$\mathsf{TC}(\mathsf{AS}_1) \preceq \mathsf{TC}(\mathsf{AS}_2)$$

In particular, Theorem 3 shows that whenever the system is deterministic and we increase the grain of the clock to a multiple of it, the new low-resolution clock gives better (or at least the same) security. Theorem 3 is proved using Fact 1, and Theorem 1 (Appendix C).

B. Timing Techniques

Several timing techniques have successfully side-stepped the defence provided by a coarse-grained clock, by building their own fine-grained clocks from primitives such as coarse-grained clocks and simple counter processes [2], [13], [14]. We now explain how techniques such as the one-pad and the clock-edge [13], [14] work, and we present a new timing technique called the co-prime.


Learning phase:$t_{\text{pad}} = \frac{g}{4}$
Attack phase:$t_f = c(t_f) + g - 3 \cdot t_{\text{pad}}$
t_{slow}
$t_{\text{slow}} + t_{\text{pad}}$
t_{fast}
$t_{\text{fast}} + t_{\text{pad}}$
g
$4 \cdot t_{\text{pad}}$
$c(t_f)$
t_f
$t_f + 3 \cdot t_{\text{pad}}$
time
time

(a) The one-pad technique.

(b) An example of the clock-edge technique. The learning phase on top and the attack phase bottom.

Fig. 4: Padding timing techniques.

We study those techniques in a setting as in [13], [14], where the adversary tries to measure the timing of a function f that is sent to the victim in a piece of malicious code where he performs his timing technique.

1) The One-Pad Technique: In many cases the adversary wants only to distinguish between two different executions of f, t_{slow} and t_{fast}, where t_{fast} is smaller than t_{slow}. Using the *one-pad* technique the adversary exploits the fact that the time between two clock-edges is constant and equal to g. He then chooses to perform a *constant time* operation called padding.

If now t_{pad} is the time of the padding operation, the padding is chosen in such a way such that the short duration t_{slow} plus the duration of the padding t_{pad} always crosses the next clock-edge i.e $t_{\text{slow}}+t_{\text{pad}} \geq c(t_{\text{slow}})+g$, while $t_{\text{fast}}+t_{\text{pad}} < c(t_{\text{fast}})+g$.

After the end of the padding operation, if the value of the adversary's clock is above the next clock-edge, he infers that the slow event has happened, otherwise it is the case of the fast event. Fig. 4 (a) depicts a scenario of the one-pad technique.

2) The Clock-Edge Technique: The one-pad technique is good enough for distinguishing between two different execution times, however, sometimes an adversary requires a mechanism that gives more precise measurements. For those cases, the *clock-edge* technique can be used. The fact that a clock c with grain g is being increased with a constant rate i.e every g, gives the attacker the ability to express the duration of a sequence of his operations as a portion to g. In particular, similarly to the one-pad technique the adversary here adds a padding for one or more times.

The technique begins with a *learning phase* (Fig. 4 (b) top), where the adversary performs his padding operation between two consecutive clock-edges. Assuming that a number of m operations have occurred between the two edges, and using that the duration of m padding operations is equal to g, the attacker derives that the duration of his padding operation is $t_{\text{pad}} = \frac{g}{m}$.

The *attack phase* (Fig. 4 (b) bottom) of the technique then begins with the adversary aligning the operation f that he wants to measure to a clock-edge and right after the execution of f completes, he inspects the value of his clock $c(t_f)$. Immediately after, he starts performing his padding operator until he observes the next clock-edge. If at this point he observes n paddings the duration t_f of f can be approximated by the number $c(t_f) + g - n \cdot t_{\text{pad}} \approx t_f$.[5]

TABLE II: Algorithm for constructing padding timing techniques.

$\mathsf{TA}(t_f, t_{pad}, m)$	$=$	let $q_\circ$, q_1, ..., q_{m+1} be fresh nodes	
		and x a fresh dense clock	
		$\mathsf{Q} =$	$\{q_\circ, q_1, ..., q_{m+1}\}$
		$\mathsf{E} =$	$\{(q_\circ, x = t_f \rightarrow x, q_1),$
			$(q_1, x = t_{\text{pad}} \rightarrow x, q_2), ...,$
			$(q_m, x = t_{\text{pad}} \rightarrow x, q_{m+1})\}$
		$\mathsf{I} =$	$[q_\circ \mapsto \mathsf{tt}][q_1 \mapsto \mathsf{tt}]...[q_{m+1} \mapsto \mathsf{tt}]$
		in $(\mathsf{Q}, q_\circ, \mathsf{E}, \mathsf{I})$	

3) The Co-Prime Technique: Looking at the one-pad and the clock-edge technique one could think of the following questions: Do we always need a padding operation with duration less than g in order to perform fine-grained measurements? What happens if we do not stop at the next clock-edge and we continue performing the padding operator for a couple of more times?

For the co-prime technique, the adversary may not necessarily use a padding with timing less than the grain of the clock, and he may also not stop at the next clock-edge. In particular, in the co-prime technique, the adversary performs a padding operation for g (the grain of the clock) times, and the timing of his padding t_{pad} is co-prime with g. Why this technique works becomes clear in the next two subsections.

C. Modelling Timing Techniques

We use the algorithm in TABLE II to model the essential aspects of the timing techniques: one-pad, clock-edge and co-prime.

Recall that we assume an adversary who tries to measure the timing of a deterministic function f that is sent to the victim in a piece of malicious code where he performs his timing technique. The timing of f varies depending on the internal state of the victim. The algorithm in TABLE II takes as an input the timing of the operation t_f, the timing of the padding t_{pad} and the number m of padding operations the adversary performs.

[5] In the clock-edge techniques presented in [13], [14] the padding operation corresponds to the increment of a counter.

The resulting timed automaton consists of a single dense clock x and $m+2$ locations which are being constructed in the third line. Line four constructs the edges of the automaton. The first edge corresponds to executing the operation f and thus we delay exactly t_f time expressed by the guard $x = t_f$. The next m edges correspond to the execution of the padding and similarly as for the first edge we now wait for exactly t_{pad} time. All the invariants are set to true.

Let now g be the granularity of the adversary's clock c, and $t_{\text{pad}} \in \mathbb{N}$ be the execution time of his padding operator. Assume also that the function f the adversary wants to measures takes an input from the victim's set $I = \{i_1, .., i_n\}$ and let $t_{i_1}, ..., t_{i_n}$ be the execution times of f on the inputs $i_1, ..., i_n$ (resp.).

The attack scenario of the one-pad technique can then be described by

$$\mathsf{AS}_{\text{1-pad}} = ((\mathsf{TA}(t_i, t_{\text{pad}}, 1))_{i \in I}, \mathsf{E}, c, 2)$$

where E are the edges of the system and they are observable, and the adversary makes $k = 2$ observations (one before and one after its padding operation).

Similarly for the (attack phase of the) clock-edge technique we have

$$\mathsf{AS}_{\text{clock-edge}} = ((\mathsf{TA}(t_i, t_{\text{pad}}, m))_{i \in I}, \mathsf{E}, c, m+1)$$

where

$$m = \min\ \{n \in \mathbb{N} \mid n \cdot t_{\text{pad}} \geq g\}$$

Again E are the edges of the system and they are all observable. The number m of paddings is ensuring that independently of f's timing, the padding will cross the next clock-edge.

Finally, if g is co-prime with t_{pad}, the attack scenario of the co-prime technique is

$$\mathsf{AS}_{\text{co-prime}} = ((\mathsf{TA}(t_i, t_{\text{pad}}, g))_{i \in I}, \mathsf{E}, c, g+1)$$

D. A Hierarchy of Timing Techniques

We now compare the power of the timing techniques, one-pad, clock-edge and co-prime in terms of how much information the adversary can extract using them, and we explain in more details why the co-prime technique works. We start with an example that illustrates the construction (TABLE II) of the attack scenarios of the clock-edge and the co-prime technique and shows that the co-prime technique can distinguish more.

Example 11. Assume that we want to distinguish between two timing behaviours of an operation f that takes the inputs i_1 and i_2 and runs for time $t_{i_1} = 8$ and $t_{i_2} = 9$ respectively. Assume also that we have a clock c with grain $g = 10$ and a padding with timing t_{pad}=2.

For the clock-edge technique we need to add our padding for

$$\begin{array}{rcl} m & = & \min\ \{n \in \mathbb{N} \mid n \cdot t_{\text{pad}} \geq g\} \\ & = & \min\ \{n \in \mathbb{N} \mid n \cdot 2 \geq 10\} \\ & = & 5 \end{array}$$

Using the algorithm of TABLE II we get the system $\mathsf{S} = (\mathsf{TA}(t_i, 2, 5))_{i \in I}$ and for each $i \in I$ we have a timed automaton $\mathsf{TA}(t_i, 2, 5) = (\mathsf{Q}, q_\circ, \mathsf{E}, \mathsf{I})$ where $\mathsf{Q} = \{q_\circ, q_1, ..., q_6\}$ and E contains the edges $e_1 = (q_\circ, x = t_i \rightarrow x, q_1)$, $e_2 = (q_1, x = 2 \rightarrow x, q_2)$,...,$e_6 = (q_5, x = 2 \rightarrow x, q_6)$, and $\mathsf{I} = \lambda q.\mathsf{tt}$.

We then have two runs $\rho_1 = \gamma_1 \xrightarrow{8,e_1} \gamma_2 \xrightarrow{2,e_2} ... \xrightarrow{2,e_6} \gamma_7$ and $\rho_2 = \gamma'_1 \xrightarrow{9,e'_1} \gamma'_2 \xrightarrow{2,e'_2} ... \xrightarrow{2,e'_6} \gamma'_7$ for i_1 and i_2 respectively.

The view of the adversary on ρ_1 is

$$\begin{array}{rcl} \mathsf{view}_c(\rho_1) & = & (c(8), c(10), c(12), c(14), c(16), c(18)) \\ & = & (0, 10, 10, 10, 10, 10) \end{array}$$

and for ρ_2 we have

$$\begin{array}{rcl} \mathsf{view}_c(\rho_2) & = & (c(9), c(11), c(13), c(15), c(17), c(19)) \\ & = & (0, 10, 10, 10, 10, 10) \end{array}$$

Therefore the runs are indistinguishable to the adversary.

Consider now the same scenario where the padding of the adversary has duration t_{pad}=19 which is *strictly greater* than the grain $g = 10$ of the clock, and let g be the number of paddings that we want to add. We will construct the attack scenario of the co-prime technique.

Using TABLE II we get the system $\mathsf{S} = (\mathsf{TA}(t_i, 19, 10))_{i \in I}$ and for each $i \in I$ we have a timed automaton $\mathsf{TA}(t_i, 19, 10) = (\mathsf{Q}, q_\circ, \mathsf{E}, \mathsf{I})$ where $\mathsf{Q} = \{q_\circ, q_1, ..., q_{11}\}$ and E contains the edges $e_1 = (q_\circ, x = t_i \rightarrow x, q_1)$, $e_2 = (q_1, x = 19 \rightarrow x, q_2)$,...,$e_{11} = (q_{10}, x = 19 \rightarrow x, q_{11})$, and $\mathsf{I} = \lambda q.\mathsf{tt}$.

We then have two runs $\rho_1 = \gamma_1 \xrightarrow{8,e_1} \gamma_2 \xrightarrow{19,e_2} ... \xrightarrow{19,e_{11}} \gamma_{12}$ and $\rho_{i_2} = \gamma'_1 \xrightarrow{9,e'_1} \gamma'_2 \xrightarrow{19,e'_2} ... \xrightarrow{19,e'_{11}} \gamma'_{12}$ (for i_1 and i_2 respectively.

The view of the adversary on ρ_1 is

$$\begin{array}{rcl} \mathsf{view}_c(\rho_1) & = & (c(8), c(27), c(46), c(65), ..., c(179), c(198)) \\ & = & (0,20,40,60,...,170,190) \end{array}$$

and for ρ_2 we have

$$\begin{array}{rcl} \mathsf{view}_c(\rho_2) & = & (c(9), c(28), c(47), c(66), ..., c(180), c(199)) \\ & = & (0,20,40,60...,180,190) \end{array}$$

and thus the two runs become distinguishable at the 10th observation, because for ρ_1 the adversary observes 170 and for ρ_2 he observes 180.

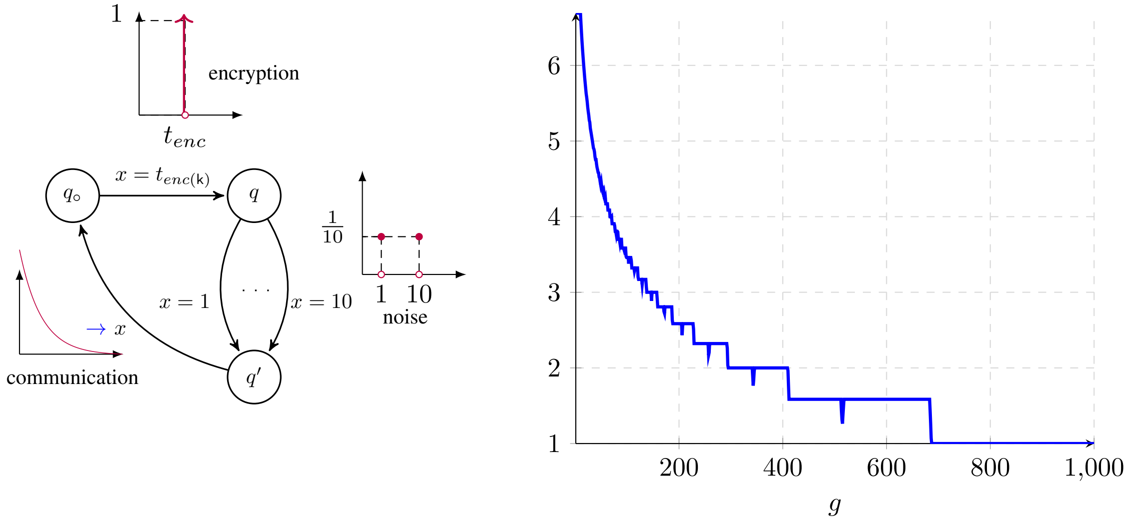
To understand better why the co-prime technique works observe that in general a timing technique is distinguishing two timings t_1, t_2, when observing differences in the sequences

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

and

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

The question therefore is how to choose the appropriate number m of paddings for making the two sequences distinguishable.



(a) The stochastic timed automaton of the sensor when it operates on key k.

(b) The effect of increasing the grain g on the min-capacity.

Fig. 5: Case Study:RSA

However, Fact 2 shows that the two sequences are indistinguishable if and only if

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

and

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

are different.

The co-prime technique exploits this fact and rephrases the question of the padding techniques to: (1) what padding is needed, and (2) how many times I need to add it so by the end of the timing technique the padding times generate the entire $\mathbb{Z}_g$ set (answer: (1) t_{pad} needs to be co-prime with g, and (2) it needs to be added g times).

We finish by showing that the timing-techniques of the one-pad, clock-edge and the co-prime form a strict hierarchy in terms of the amount of information the adversary can extract. In particular, we show that the co-prime techinique achieves the most information leakage among the other techniques, whereas the one-pad achieves the least leakage.

Theorem 4. Let f be a function that runs on the input set I and its timing behaviour is described by the family $(t_i)_{i \in I}$. Let also c be a clock with grain g, a padding with time t_{pad}, and $\mathsf{AS}_{\text{1-pad}}$, and $\mathsf{AS}_{\text{clock-edge}}$ the corresponding attack scenarios of the one-pad and the clock-edge techniques. Consider also another padding with duration t'_{pad} that is co-prime with g, and let $\mathsf{AS}_{\text{co-prime}}$ be the corresponding co-prime attack scenario. We then have that

$$\mathsf{TC}(\mathsf{AS}_{\text{1-pad}}) \preceq \mathsf{TC}(\mathsf{AS}_{\text{clock-edge}}) \preceq \mathsf{TC}(\mathsf{AS}_{\text{co-prime}})$$

Theorem 4 is proved using Fact 2, and Theorem 1 (Appendix D).

VI. Analysis of Timing Channels in Stochastic Systems: A Case Study

In this section, we analyse timing channels of stochastic systems. In particular, we perform a case study, which consists of two parts. The first part is the modelling of our case study as a system of timed automata. In the second part, we consider different adversaries (with respect to their clock), we compute their timing channels, and we derive our insights about the relation between clock grain and leakage in stochastic systems.

A. Modelling the Case Study

We consider a scenario of a distributed system that consists of a sensor and a controller. We are interested in modelling the behaviour of the sensor. For details of the case study, see Appendix E.

In particular, the sensor constantly computes some data and communicates it to the controller. For ensuring data integrity, the sensor always encrypts (signs) the data with his RSA private key. The RSA encryption is implemented using the modular exponentiation algorithm which computes $\mathsf{x}^{\mathsf{k}} \bmod \mathsf{n}$ for the secret key k, some data x and the constant modulus n. The implementation is given by the following piece of code

```
m := (1 * 1) mod n;
for (j = 0; j < len(k); j++) {
    m := (m * m) mod n;
    if (k[j] == 1) then
      m := (m * x) mod n;
}
```

where the secret bits of the key are stored in the array k[]. Due to the conditional execution of the modular multiplication operation $\mathsf{m} = \mathsf{m} * \mathsf{x} \bmod \mathsf{n}$ the running time of this program reveals information about the entries of k.

To decrease the correlation between the encryption time and the secret bits of the key, the sensor adds noise to the encryption time by delaying for some additional period after each encryption, and then it communicates the data to the controller.

In our model, we assume that the sensor performs each modular multiplication in 1 time unit. We also assume that the size of the secret key k is 1024-bits, and thus the timed needed for one encryption is

$$t_{enc(\mathsf{k})} = 1025 + \mathsf{Ham}(\mathsf{k})$$

where $\mathsf{Ham}(\mathsf{k})$ is the Hamming weight of k (i.e the number of non-zero bits).

For the noise added by the sensor, we assume that the sensor chooses randomly to wait for $t \in \{1, 2, ..., 10\}$, with respect to a uniform distribution.

We model the timing behaviour of the sensor as a stochastic timed system with input space the 2^{1024} keys and for each key, we have a stochastic timed automaton as depicted in Fig. 5 (a).

The automaton consists of three locations and one dense clock x that is used to control the transitions between them. Starting at the initial location $q_\circ$, the automaton performs an encryption with respect to a Dirac's distribution on the time point $t_{enc(\mathsf{k})}$, modeling in that way that an encryption takes exactly $t_{enc(\mathsf{k})}$ time units. Next, it moves to location q, where we have 10 different edges leaving q, one for each possible delay, modelling the additional noise added by the sensor. A delay is chosen uniformly and the automaton moves to location q'. At location q' the automaton communicates the message to the controller with respect to an exponential distribution of parameter $\lambda = 6$ time units.

Finally, on the side of the controller, we assume an adversary who runs malicious code that uses the clock of the controller and measures the time needed for the sensor to send its data, trying to infer bits of the sensor's private key.

B. Analysing the Leakage in the Case Study

Using TABLE I, we constructed the timing channel for 1000 different attack scenarios (for different clocks), where we have as observable edges the ones that model the communication of the message. We assumed an adversary that performs one timing observation and uses a clock with grain $g = 1, 2, ..., 1000$ and with some limit $l > 15000$. The details of the construction can be found in the Appendix E.

We then computed the min-capacity for each one of those channels. The graph in Fig. 5 (b) shows the effect of increasing the grain of the clock on the information leaked by the channel. The maximum leakage is around 6.7-bits for grain $g = 1$, whereas we have 1-bit leakage for the attack scenarios where the grain is above 678. We can also see from the graph that increasing the grain of the clock does not always give us less information leakage. In particular, for $g = 514$, we have around 1.43-bits leaked, whereas for $g = 520$ we have around 1.58-bits, which leads to the following proposition

Proposition 2. There exists stochastic system S and attack scenarios AS_1, AS_2 of S with clocks c_1, c_2 resp., and grains g_1, g_2 with $g_1 < g_2$, and $\mathsf{C}_{\min}(\mathsf{TC}(\mathsf{AS}_1)) < \mathsf{C}_{\min}(\mathsf{TC}(\mathsf{AS}_2))$.

Proposition 2 shows also for the case of stochastic systems that the security offered by a coarse-grained clock could be worse than the one offered by a fine-grained clock, even when bypassing timing techniques are not used.

Finally, our experiment shows that increasing the grain of the clock to a multiple of it results to a channel with less (or equal) leakage, however a proof that this holds for general stochastic systems (i.e Theorem 3 generalizes to stochastic systems) is still elusive.

VII. Related Work

There is an extensive work on formally quantifying and providing bounds on the leakage of timing-channels for cryptographic implementations [9], [10], [22], [23], remote network adversaries [11], [24] and language-based settings [25]–[27]. The main novelty of our approach compared to those is the modelling of coarse-grained clock adversaries, and the novel algorithms that we give for constructing timing channels for systems of timed automata.

Clocks of certain granularity and their defence power have been studied a lot in practice using empirical ways. Schwarz et al. [14] provided a wide range of techniques that can be used to build fine-grained clocks in Javascript, including similar techniques to the one-pad and the clock-edge. Wei-Ming Hu [28], [29] proposed the concept of fuzzy time. Instead of increasing the grain of the clock, fuzzy time modifies its functionality by randomly changing its grain within a certain period. Vattikonda et al. [30] proposed fuzzy time for mitigating timing channels in hypervisors, and also evaluated the impact of this countermeasure on the usability of the system. Fuzzy time has also been proposed and implemented in Firefox by Kohlbrenner et al. [13] for defeating timing channels in browsers. They also showed that this mitigation is effective against timing techniques such as the clock-edge.

Mantel et al. [12] proposed an information-theoretic framework for comparing the effectiveness of different countermeasures on the bandwidth of interrupt-related channels, that is a special case of timing channels. In their analysis, they include the countermeasure of coarse-grained clocks and fuzzy time. For coarse-grained clocks, they perform a case-study where they show how increasing the grain of the clock reduces the capacity of the channel. Our approach is more general, while we also showed that increasing the grain of a clock might result to more leakage, and we provided formal proofs for when this is not the case.

VIII. Conclusions

We performed the first principled information-flow analysis of timing leaks w.r.t. adversaries with clocks of reduced resolution, where we derived novel insights into the effectiveness of existing attacks and countermeasures.

In particular, we introduced a model of timed automata systems which is general enough to cater for scenarios where the victim's timing behaviour is stochastic or deterministic, and a model of adversary that is parametric on the clock's granularity and the number of timing observations.

We provided novel algorithms for transforming such a model into an information-theoretic channel, allowing us to measure the leakage conveyed by it using existing techniques from quantitative information-flow.

Based on that, we showed that a coarse-grained clock might leak more than a fine-grained clock, and we provided sufficient conditions for when one can achieve better security by increasing the grain of the clock. For the techniques that have bypassed this countermeasure, we showed that they form a strict hierarchy in terms of the information an adversary can extract using them, and we introduced a new timing technique.

As future work, we are interested in extending our model so it can encompass for clocks with fuzzy time as in [12], [13], [28], [29]. As another direction, we want to approximate the security offered by a coarse-grained clock using techniques from statistical model checking and automate our analysis using the model-checker for (stochastic) timed automata UPPAAL [31]. This approach is particularly interesting since it can allow for estimating the trade-off between security and safety properties of a system.

Acknowledgements Part of this work was carried out at the IMDEA Software Institute, supported by a grant from the Intel Corporation, Ramón y Cajal grant RYC-2014-16766, Spanish projects TIN2015-70713-R DEDETIS and TIN2015-67522-C3-1-R TRACES, and Madrid regional project S2013/ICE-2731 N-GREENS.

References

[1] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, 1996, pp. 104–113. [Online]. Available: https://doi.org/10.1007/3-540-68697-5_9

[2] D. X. Song, D. A. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on SSH," in *10th USENIX Security Symposium, August 13-17, 2001, Washington, D.C., USA*, 2001. [Online]. Available: http://www.usenix.org/publications/library/proceedings/sec01/song.html

[3] B. B. Brumley and N. Tuveri, "Remote timing attacks are still practical," in *Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, 2011, pp. 355–371. [Online]. Available: https://doi.org/10.1007/978-3-642-23822-2_20

[4] P. Vila and B. Köpf, "Loophole: Timing attacks on shared event loops in chrome," in *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017.*, 2017, pp. 849–864. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/vila

[5] Y. Oren, V. P. Kemerlis, S. Sethumadhavan, and A. D. Keromytis, "The spy in the sandbox: Practical cache attacks in javascript and their implications," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, 2015, pp. 1406–1418. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813708

[6] E. W. Felten and M. A. Schneider, "Timing attacks on web privacy," in *CCS 2000, Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece, November 1-4, 2000.*, 2000, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/352600.352606

[7] M. Andrysco, D. Kohlbrenner, K. Mowery, R. Jhala, S. Lerner, and H. Shacham, "On subnormal floating point and abnormal timing," in *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, 2015, pp. 623–639. [Online]. Available: https://doi.org/10.1109/SP.2015.44

[8] J. B. Almeida, M. Barbosa, G. Barthe, F. Dupressoir, and M. Emmi, "Verifying constant-time implementations," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, 2016, pp. 53–70. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/almeida

[9] D. Chaum, R. L. Rivest, and A. T. Sherman, Eds., *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982*. Plenum Press, New York, 1983.

[10] B. Köpf and M. Dürmuth, "A provably secure and efficient countermeasure against timing attacks," *IACR Cryptology ePrint Archive*, vol. 2009, p. 89, 2009. [Online]. Available: http://eprint.iacr.org/2009/089

[11] D. Zhang, A. Askarov, and A. C. Myers, "Predictive mitigation of timing channels in interactive systems," in *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011*, 2011, pp. 563–574. [Online]. Available: https://doi.org/10.1145/2046707.2046772

[12] H. Mantel and H. Sudbrock, "Comparing countermeasures against interrupt-related covert channels in an information-theoretic framework," in *20th IEEE Computer Security Foundations Symposium, CSF 2007, 6-8 July 2007, Venice, Italy*, 2007, pp. 326–340. [Online]. Available: https://doi.org/10.1109/CSF.2007.14

[13] D. Kohlbrenner and H. Shacham, "Trusted browsers for uncertain times," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, 2016, pp. 463–480. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kohlbrenner

[14] M. Schwarz, C. Maurice, D. Gruss, and S. Mangard, "Fantastic timers and where to find them: High-resolution microarchitectural attacks in javascript," in *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers*, 2017, pp. 247–267. [Online]. Available: https://doi.org/10.1007/978-3-319-70972-7_13

[15] J. C. Wray, "An analysis of covert timing channels," in *IEEE Symposium on Security and Privacy*, 1991, pp. 2–7. [Online]. Available: https://doi.org/10.1109/RISP.1991.130767

[16] L. Aceto, A. Ingolfsdottir, K. G. Larsen, and J. Srba, *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press, 2007.

[17] R. Alur and D. L. Dill., "A theory of timed automata." *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, 1994.

[18] P. Carlier, "Verification of stochastic timed automata. (vérification des automates temporisés et stochastiques)," Ph.D. dissertation, University of Paris-Saclay, France, 2017. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01696130

[19] N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Größer, and M. Jurdzinski, "Stochastic timed automata," *Logical Methods in Computer Science*, vol. 10, no. 4, 2014. [Online]. Available: https://doi.org/10.2168/LMCS-10(4:6)2014

[20] M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *25th IEEE Computer Security Foundations Symposium, CSF 2012, Cambridge, MA, USA, June 25-27, 2012*, 2012, pp. 265–279. [Online]. Available: https://doi.org/10.1109/CSF.2012.26

[21] G. Smith, "On the foundations of quantitative information flow," in *Foundations of Software Science and Computational Structures, 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, 2009, pp. 288–302. [Online]. Available: https://doi.org/10.1007/978-3-642-00596-1_21

[22] B. Köpf and D. A. Basin, "An information-theoretic model for adaptive side-channel attacks," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, 2007, pp. 286–296. [Online]. Available: https://doi.org/10.1145/1315245.1315282

[23] M. Backes and B. Köpf, "Quantifying information flow in cryptographic systems," *Mathematical Structures in Computer Science*, vol. 25, no. 2, pp. 457–479, 2015. [Online]. Available: https://doi.org/10.1017/S0960129513000662

[24] J. Giles and B. E. Hajek, "An information-theoretic and game-theoretic study of timing channels," *IEEE Trans. Information Theory*, vol. 48, no. 9, pp. 2455–2477, 2002. [Online]. Available: https://doi.org/10.1109/TIT.2002.801405

[25] A. D. Pierro, C. Hankin, and H. Wiklicky, "Quantifying timing leaks and cost optimisation," in *Information and Communications Security, 10th International Conference, ICICS 2008, Birmingham, UK, October 20-22, 2008, Proceedings*, 2008, pp. 81–96. [Online]. Available: https://doi.org/10.1007/978-3-540-88625-9_6

[26] P. Malacaria, M. H. R. Khouzani, C. S. Pasareanu, Q. Phan, and K. S. Luckow, "Symbolic side-channel analysis for probabilistic programs," in *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, 2018, pp. 313–327. [Online]. Available: https://doi.org/10.1109/CSF.2018.00030

[27] G. Doychev, D. Feld, B. Köpf, L. Mauborgne, and J. Reineke, "Cacheaudit: A tool for the static analysis of cache side channels," in *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013*, 2013, pp. 431–446. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/doychev

[28] W. Hu, "Reducing timing channels with fuzzy time," in *IEEE Symposium on Security and Privacy*, 1991, pp. 8–20. [Online]. Available: https://doi.org/10.1109/RISP.1991.130768

[29] ——, "Reducing timing channels with fuzzy time," *Journal of Computer Security*, vol. 1, no. 3-4, pp. 233–254, 1992. [Online]. Available: https://doi.org/10.3233/JCS-1992-13-404

[30] B. C. Vattikonda, S. Das, and H. Shacham, "Eliminating fine grained timers in xen," in *Proceedings of the 3rd ACM Cloud Computing Security Workshop, CCSW 2011, Chicago, IL, USA, October 21, 2011*, 2011, pp. 41–46. [Online]. Available: https://doi.org/10.1145/2046660.2046671

[31] A. David, K. G. Larsen, A. Legay, M. Mikucionis, and D. B. Poulsen, "Uppaal SMC tutorial," *STTT*, vol. 17, no. 4, pp. 397–415, 2015. [Online]. Available: https://doi.org/10.1007/s10009-014-0361-y

APPENDIX
PROOFS AND CALCULATIONS

A. Proof of Fact 1

Proof: A clock c with grain g, defines an equivalence relation on $\mathbb{R}_{\geq 0}$ by for t_1, $t_2 \in \mathbb{R}_{\geq 0}$ we have that

$$t_1 \equiv_g t_2 \text{ iff } c(t_1) = c(t_2)$$

and observe that the equivalence classes of $\equiv_g$ are then described by the intervals $[0, g)$, $[g, 2g)$, $[2g, 3g)$.....

Now for two clocks c_1 and c_2 with grains g_1 and g_2 (resp.) where $g_2 = n \cdot g_1$ (for n being a positive integer) we have that $\equiv_{g_1}$ refines every equivalence class of $\equiv_{g_2}$ in exactly n equivalence classes (e.g for the first equivalence class $[0, g_2)$ of $\equiv_{g_2}$ we have the n equivalence classes $[0, g_1), ..., [(n-1) \cdot g_1, n \cdot g_1)$ of $\equiv_{g_1}$). Therefore for t_1, $t_2 \in \mathbb{R}_{\geq 0}$ we have that

$$t_1 \equiv_{g_1} t_2 \Rightarrow t_1 \equiv_{g_2} t_2$$

which give us that

$$c_1(t_1) = c_1(t_2) \Rightarrow c_2(t_1) = c_2(t_2)$$

as required. ■

B. Proof of Fact 2

Proof: Since $n \in \mathbb{N}$ is a natural number we have that $n = m \cdot g + (n \bmod g)$ for some integer m. We then have that $c(t_1 + n) = c(t_2 + n)$ iff

$$\begin{aligned}
& \left\lfloor \tfrac{t_1+n}{g} \right\rfloor \cdot g = \left\lfloor \tfrac{t_2+n}{g} \right\rfloor \cdot g \\
\Leftrightarrow\ & \left\lfloor \tfrac{t_1+m\cdot g+(n \bmod g)}{g} \right\rfloor \cdot g = \left\lfloor \tfrac{t_2+m\cdot g+(n \bmod g)}{g} \right\rfloor \cdot g \\
\Leftrightarrow\ & \left\lfloor \tfrac{t_1+(n \bmod g)}{g} + m \right\rfloor \cdot g = \left\lfloor \tfrac{t_2+(n \bmod g)}{g} + m \right\rfloor \cdot g \\
\Leftrightarrow\ & (\left\lfloor \tfrac{t_1+(n \bmod g)}{g} \right\rfloor + m) \cdot g = (\left\lfloor \tfrac{t_2+(n \bmod g)}{g} \right\rfloor + m) \cdot g \\
\Leftrightarrow\ & \left\lfloor \tfrac{t_1+(n \bmod g)}{g} \right\rfloor \cdot g + g \cdot m = \\
& \left\lfloor \tfrac{t_2+(n \bmod g)}{g} \right\rfloor \cdot g + g \cdot m \\
\Leftrightarrow\ & \left\lfloor \tfrac{t_1+(n \bmod g)}{g} \right\rfloor \cdot g = \left\lfloor \tfrac{t_2+(n \bmod g)}{g} \right\rfloor \cdot g \\
\Leftrightarrow\ & c(t_1 + (n \bmod g)) {=} c(t_2 + (n \bmod g))
\end{aligned}$$

and thus we have proved the required result. ■

C. Proof of Theorem 3

Proof. Let $\mathsf{AS}_1 = (\mathsf{S}, \mathsf{E}_{\text{pub}}, c_1, k)$ and $\mathsf{AS}_2 = (\mathsf{S}, \mathsf{E}_{\text{pub}}, c_2, k)$ be two attack scenarios such that S is deterministic and the clocks c_1, c_2 have grains g_1, g_2 (resp.), and g_1 is a multiple of g_2. We will prove that

$$\mathsf{TC}(\mathsf{AS}_1) \preceq \mathsf{TC}(\mathsf{AS}_2)$$

Since S is deterministic then also $\mathsf{TC}(\mathsf{AS}_1)$ and $\mathsf{TC}(\mathsf{AS}_2)$ are, and thus by Theorem 1 in order to prove that $\mathsf{TC}(\mathsf{AS}_1) \preceq \mathsf{TC}(\mathsf{AS}_2)$ it is sufficient to show that $\mathsf{TC}(\mathsf{AS}_1) \sqsubseteq \mathsf{TC}(\mathsf{AS}_2)$.

Let i_1, $i_2 \in I$, with $i_1 \equiv_{\mathsf{TC}(\mathsf{AS}_2)} i_2$. Let also ρ_1, $\rho_2 \in \mathsf{Runs}(\mathsf{S})$ be their corresponding runs, and t'_1,...,t'_k and t''_1,...,t''_k be the k-time sequence of ρ_1 and ρ_2 resp. Since, $i_1 \equiv_{\mathsf{TC}(\mathsf{AS}_2)} i_2$ we have that the view of the adversary on the runs of them is the same, that is

$$\begin{aligned}
\mathsf{view}_{c_2}(\rho_1) &= (c_2(t'_1), c_2(t'_2), ..., c_2(t'_k)) \\
&= \mathsf{view}_{c_2}(\rho_2) \\
&= (c_2(t''_1), c_2(t''_2), ..., c_2(t''_k)) \quad (1)
\end{aligned}$$

Next, using that the grain g_1 of the clock c_1, is a multiple of the grain g_2, of the clock c_2, Fact 1 and (1) we get that

$$\begin{aligned}
\mathsf{view}_{c_1}(\rho_1) &= (c_1(t'_1), c_1(t'_2), ..., c_1(t'_k)) \\
&= \mathsf{view}_{c_1}(\rho_2) \\
&= (c_1(t''_1), c_1(t''_2), ..., c_1(t''_k))
\end{aligned}$$

and this give us that $i_1 \equiv_{\mathsf{TC}(\mathsf{AS}_1)} i_2$ and thus we showed that $\mathsf{TC}(\mathsf{AS}_1) \sqsubseteq \mathsf{TC}(\mathsf{AS}_2)$ as required. □

D. Proof of Theorem 4

Proof: Let $\mathsf{S}_{\text{1-pad}}$, $\mathsf{S}_{\text{clock-edge}}$ and $\mathsf{S}_{\text{co-prime}}$ be the deterministic systems that correspond to the three scenarios $\mathsf{AS}_{\text{1-pad}}$, $\mathsf{AS}_{\text{clock-edge}}$ and $\mathsf{AS}_{\text{co-prime}}$ (resp.) Since $\mathsf{S}_{\text{1-pad}}$, $\mathsf{S}_{\text{clock-edge}}$, $\mathsf{S}_{\text{co-prime}}$ are deterministic, we also have that $\mathsf{TC}(\mathsf{AS}_{\text{1-pad}}) : I \times O_1 \mapsto [0, 1]$, $\mathsf{TC}(\mathsf{AS}_{\text{clock-edge}}) : I \times O_2 \mapsto [0, 1]$ and $\mathsf{TC}(\mathsf{AS}_{\text{co-prime}}) : I \times O_3 \mapsto [0, 1]$ are deterministic. Therefore, using Theorem 1, in order to prove that

$$\mathsf{TC}(\mathsf{AS}_{\text{1-pad}}) \preceq \mathsf{TC}(\mathsf{AS}_{\text{clock-edge}}) \preceq \mathsf{TC}(\mathsf{AS}_{\text{co-prime}})$$

it is sufficient to show that

$$\mathsf{TC}(\mathsf{AS}_{\text{1-pad}}) \sqsubseteq \mathsf{TC}(\mathsf{AS}_{\text{clock-edge}}) \sqsubseteq \mathsf{TC}(\mathsf{AS}_{\text{co-prime}})$$

that is that the partition of $\mathsf{TC}(\mathsf{AS}_{\text{1-pad}})$ is refined by the one of $\mathsf{TC}(\mathsf{AS}_{\text{clock-edge}})$, and the partition of $\mathsf{TC}(\mathsf{AS}_{\text{clock-edge}})$ is refined by the one of $\mathsf{TC}(\mathsf{AS}_{\text{co-prime}})$.

We will start by showing that

$$\mathsf{TC}(\mathsf{AS}_{\text{clock-edge}}) \sqsubseteq \mathsf{TC}(\mathsf{AS}_{\text{co-prime}})$$

Let i_1, $i_2 \in I$, with their timings t_{i_1}, t_{i_2}, such that

$$i_1 \equiv_{\mathsf{TC}(\mathsf{AS}_{\text{co-prime}})} i_2$$

which means that there exists $o \in O_3$ such that

$$\mathsf{TC}(\mathsf{AS}_{\text{co-prime}})(i_1, o) = 1 = \mathsf{TC}(\mathsf{AS}_{\text{co-prime}})(i_2, o) \quad (1)$$

Next, let ρ_{i_1} and ρ_{i_2} to be the runs of the automata of the system $\mathsf{S}_{\text{co-prime}}$ which correspond to i_1 and i_2 (resp.). Expanding (1) we have that

$$\begin{aligned}\mathsf{view}_c(\rho_{i_1}) &= (c(t_{i_1}), c(t_{i_1} + t'_{\text{pad}}), ..., c(t_{i_1} + g \cdot t'_{\text{pad}}))\\ &= o\\ &= \mathsf{view}_c(\rho_{i_2})\\ &= (c(t_{i_2}), c(t_{i_2} + t'_{\text{pad}}), ..., c(t_{i_2} + g \cdot t'_{\text{pad}})) \quad (2)\end{aligned}$$

Now since t'_{pad} is co-prime with g, t'_{pad} is a generator of the group $(\mathbb{Z}_g, +)$ and thus

$$\begin{aligned}\mathbb{Z}_g &= \{0, 1, .., g-1\}\\ &= \left\{0 \bmod g, t'_{\text{pad}} \bmod g, ..., (g-1) \cdot t'_{\text{pad}} \bmod g\right\} \quad (3)\end{aligned}$$

Therefore using (3) and (2) we have that

$$\forall z \in \mathbb{Z}_g : c(t_{i_1} + z) = c(t_{i_2} + z) \quad (4)$$

Now using (4) we will show that $i_1 \equiv_{\mathsf{TC}(\mathsf{AS}_{\text{clock-edge}})} i_2$. Let ρ'_{i_1} and ρ'_{i_2} to be the runs of the automata of the system $\mathsf{S}_{\text{clock-edge}}$ that correspond to i_1 and i_2 (resp.). We then have that

$$\mathsf{view}_c(\rho'_{i_1}) = (c(t_{i_1}), c(t_{i_1} + t_{\text{pad}}), ..., c(t_{i_1} + m \cdot t_{\text{pad}}))$$

and

$$\mathsf{view}_c(\rho'_{i_1}) = (c(t_{i_2}), c(t_{i_2} + t_{\text{pad}}), ..., c(t_{i_2} + m \cdot t_{\text{pad}}))$$

where m is the number of paddings needed for the clock-edge technique. Using Fact 2, we have that for proving $\mathsf{view}_c(\rho'_{i_1}) = \mathsf{view}_c(\rho'_{i_1})$ it is sufficient to show that

$$\begin{aligned}&\forall z \in \{0 \bmod g, t_{\text{pad}} \bmod g, ..., (m \cdot t_{\text{pad}}) \bmod g\} :\\ &\quad c(t_{i_1} + z) = c(t_{i_2} + z) \ (6)\end{aligned}$$

However, since

$$\{0 \bmod g, t_{\text{pad}} \bmod g, ..., (m \cdot t_{\text{pad}}) \bmod g\} \subseteq \mathbb{Z}_g$$

and by (4), we have that (6) holds and we have proved that

$$\mathsf{TC}(\mathsf{AS}_{\text{clock-edge}}) \sqsubseteq \mathsf{TC}(\mathsf{AS}_{\text{co-prime}})$$

Finally, we will show that $\mathsf{TC}(\mathsf{AS}_{\text{1-pad}}) \sqsubseteq \mathsf{TC}(\mathsf{AS}_{\text{clock-edge}})$. Let i_1, $i_2 \in I$, with times t_{i_1} and t_{i_2} (resp.), such that $i_1 \equiv_{\mathsf{TC}(\mathsf{AS}_2)} i_2$, which means that there exists $o \in O_2$ such that $\mathsf{TC}(\mathsf{AS}_{\text{clock-edge}})(i_1, o) = 1 = \mathsf{TC}(\mathsf{AS}_{\text{clock-edge}})(i_2, o)$. Next let ρ_{i_1}, ρ_{i_2} to be the runs of the automata of the system $\mathsf{S}_{\text{clock-edge}}$ that correspond to i_1 and i_2 (resp.), and ρ'_{i_1}, ρ'_{i_2} to be the runs of the automata of the system $\mathsf{S}_{\text{1-pad}}$ that correspond to i_1 and i_2 (resp.). By our assumptions we have that

$$\begin{aligned}\mathsf{view}_c(\rho_{i_1}) &= (c(t_{i_1}), c(t_{i_1} + t_{\text{pad}}), ..., c(t_{i_1} + m \cdot t_{\text{pad}}))\\ &= o\\ &= \mathsf{view}_c(\rho_{i_2})\\ &= (c(t_{i_2}), c(t_{i_2} + t_{\text{pad}}), ..., c(t_{i_2} + m \cdot t_{\text{pad}}))\end{aligned}$$

for m being the padding needed for the clock-edge technique, and thus we also have that

$$\begin{aligned}\mathsf{view}_c(\rho'_{i_1}) &= (c(t_{i_1}), c(t_{i_1} + t_{\text{pad}}))\\ &= (c(t_{i_2}), c(t_{i_2} + t_{\text{pad}}))\\ &= \mathsf{view}_c(\rho'_{i_2})\end{aligned}$$

which give us that $i_1 \equiv_{\mathsf{TC}(\mathsf{AS}_{\text{1-pad}})} i_2$.

Therefore we can conclude that

$$\mathsf{TC}(\mathsf{AS}_{\text{1-pad}}) \sqsubseteq \mathsf{TC}(\mathsf{AS}_{\text{clock-edge}})$$

and this completes the proof. ■

E. Details of the Case Study

We show **Step 1**, and **Step 2** of the algorithm in TABLE I for an arbitrary key k, a clock c_l with $g \in [1, 1000]$ and limit $l > 15000$.

Let $e_1 = (q_\circ, x = t_{enc(\mathsf{k})}, q)$ be the edge that corresponds to the encryption, $e_2 = (q, x = 1, q')$,...,$e_{11} = (q, x = 10, q')$ the edges that correspond to the noise, and $e_{12} = (q', \rightarrow x, q_\circ)$ to be the edge of the communication.

For the initial configuration $\gamma_{q_\circ}$ we have the Dirac's distribution $\mu_{\gamma_{q_\circ}}$, where for a Borel set A we have

$$\mu_{\gamma_{q_\circ}}(A) = \delta_{t_{enc}(\mathsf{k})}(A) = \begin{cases} 1 & \text{if } t_{enc}(\mathsf{k}) \in A\\ 0 & \text{otherwise}\end{cases}$$

For a configuration γ_q of the location q, we have a discrete uniform probability μ_{γ_q} over the set $1, ..., 10$ given by the probability mass function

$$p(t) = \frac{1}{10} \cdot 1_{\{1,...,10\}}(t)$$

Finally, for the configuration $\gamma_{q'}$ of the location q' we have an exponential distribution $\mu_{\gamma_{q'}}$ given by the density function

$$f(t) = 6 \cdot \mathsf{exp}(-6 \cdot t) \cdot 1_{[0,+\infty)}(t)$$

Next, for any $t_1 \in \mathsf{Int}(\gamma_\circ) = \left\{t_{enc(\mathsf{k})}\right\}$, $t_2 \in \mathsf{Int}(\gamma_q,) = \{1, ..., 10\}$, and $t_3 \in \mathsf{Int}(\gamma_{q'}) = [0, +\infty)$ we have $\kappa_{\gamma_{q_\circ}+t_1}(e_1) = 1$, $\kappa_{\gamma_q+t_2}(e) = 1$ (if $t_2 = z$ and $e = e_{z+1}$) and $\kappa_{\gamma_{q'}+t_3}(e_{12}) = 1$ respectively.

Step 1, the possible observations of the adversary for input k, is given by the set

$$O_{\mathsf{k}} = \left\{c(t_{enc(\mathsf{k})} + 1), c(t_{enc(\mathsf{k})} + 1) + g, ..., l\right\}$$

Next we need to compute the probabilities of those outputs (**Step 2**). We start by computing the 1-observable (i.e $k = 1$) paths of the automaton and we get

$$\textbf{Paths} = \{e_1 e_2 e_{12}, e_1 e_3 e_{12}, ..., e_1 e_{11} e_{12}\}$$

Therefore for an observation $o \in O_{\mathsf{k}}$ we have that

$$\mathsf{view}_{c_l}(o) = \bigcup_{\pi \in \textbf{Paths}} \mathsf{Cyl}_{\mathcal{C}_{\pi(o)}}(\gamma_{q_\circ}, \pi)$$

and thus

$$\mathsf{P}_{\gamma_{q_\circ}}(\mathsf{view}_{c_l}^{-1}(o)) = \sum_{\pi \in \textbf{Paths}} \mathsf{P}_{\gamma_{q_\circ}}(\mathsf{Cyl}_{\mathcal{C}_\pi(o)}(\gamma_{q_\circ}, \pi))$$

For an observation $o \in O_{\mathsf{k}}$, and a path $\pi \in$ **Paths** we will show how we compute the probability

$$\mathsf{P}_{\gamma_{q_\circ}}(\mathsf{Cyl}_{\mathcal{C}_\pi(o)}(\gamma_{q_\circ}, \pi))$$

We distinguish the following cases $o < l$ and $o = l$.

Case (a). For $o < l$, and $\pi = e_1 e_z e_{12} \in$ **Paths** where $z \in \{2, ..., 11\}$ we have that

$$\mathcal{C}_\pi(o) = \left\{(t_1, t_2, t_3) \in \mathbb{R}^3_{\geq 0} \mid o \leq t_1 + t_2 + t_3 < o + g\right\}$$

and

$$\mathsf{P}_{\gamma_{q_\circ}}(\mathsf{Cyl}_{\mathcal{C}_\pi(o)}(\gamma_{q_\circ}, \pi))$$

is equal to

$$\int_{t_1 \in \mathsf{Int}(\gamma_{q_\circ}, e_1)} \kappa_{\gamma_{q_\circ} + t_1}(e_1) \cdot \mathsf{P}_{\gamma_q}(\mathsf{Cyl}_{\mathcal{C}_\pi^{t_1}(o)}(\gamma_q, \pi(1))) d\mu_{\gamma_{q_\circ}}(t_1)$$

where $\pi(1) = e_z e_{12}$. Since we integrate with respect to a Dirac's distribution over $t_{enc(\mathsf{k})}$, we have that the previous integral is equal to

$$\mathsf{P}_{\gamma_q}(\mathsf{Cyl}_{\mathcal{C}_\pi^{t_{enc(\mathsf{k})}}(o)}(\gamma_q, \pi(1))) \quad (1)$$

Next let $\mathcal{C}_\pi^{t_{enc(\mathsf{k})}}(o) = \mathcal{C}_1$ and then (1) is equal to

$$\begin{aligned}
&\int_{t_2 \in \mathsf{Int}(\gamma_q, e_z)} \kappa_{\gamma_q + t}(e_z) \cdot \mathsf{P}_{\gamma_{q'}}(\mathsf{Cyl}_{\mathcal{C}_1^{t_2}}(\gamma_{q'}, e_{12})) d\mu_{\gamma_q}(t_2) \\
&= p(z-1) \cdot \mathsf{P}_{\gamma_{q'}}(\mathsf{Cyl}_{\mathcal{C}_1^{z-1}}(\gamma_{q'}, e_{12})) \\
&= \frac{1}{10} \cdot \mathsf{P}_{\gamma_{q'}}(\mathsf{Cyl}_{\mathcal{C}_1^{z-1}}(\gamma_{q'}, e_{12})) \\
&= \frac{1}{10} \cdot \int_{t \in \mathsf{Int}(\gamma_{q'}, e_{12})} \kappa_{\gamma_{q'} + t}(e_{12}) \cdot 1_{\mathcal{C}_1^{z-1}}(t) d\mu_{\gamma_{q'}}(t) \\
&= \frac{1}{10} \cdot \int_{t \in [0, +\infty)} 6 \cdot \mathsf{exp}(-6 \cdot t) \cdot 1_{[0, +\infty)}(t) \cdot 1_{\mathcal{C}_1^{z-1}}(t) dt \quad (2)
\end{aligned}$$

We next distinguish between three subcases based on the constraint

$$\mathcal{C}_1^{z-1} = [o - (t_{enc(\mathsf{k})} + z - 1), o + g - (t_{enc(\mathsf{k})} + z - 1))$$

First, let

$$L = o - (t_{enc(\mathsf{k})} + z - 1)$$

and

$$U = o + g - (t_{enc(\mathsf{k})} + z - 1)$$

Now if $L < 0$, and $U > 0$, (2) is

$$\begin{aligned}
&\frac{1}{10} \cdot \int_{t \in [0, U)} 6 \cdot \mathsf{exp}(-6 \cdot t) dt \\
&= \frac{1}{10} \cdot (-\mathsf{exp}(-6 \cdot U) + 1)
\end{aligned}$$

Next, if $L \geq 0$, and $U > 0$, (2) is

$$\begin{aligned}
&\frac{1}{10} \cdot \int_{t \in [L, U)} 6 \cdot \mathsf{exp}(-6 \cdot t) dt \\
&= \frac{1}{10} \cdot (-\mathsf{exp}(-6 \cdot U) + \mathsf{exp}(-6 \cdot L))
\end{aligned}$$

Otherwise, when $U \leq 0$ (2) is equal to 0.

Case (b). For $o = l$, and $\pi = e_1 e_z e_{12} \in$ **Paths** where $z \in \{2, ..., 11\}$ we have that

$$\mathcal{C}_\pi(l) = \left\{(t_1, t_2, t_3) \in \mathbb{R}^3_{\geq 0} \mid t_1 + t_2 + t_3 \geq l\right\}$$

and

$$\mathsf{P}_{\gamma_{q_\circ}}(\mathsf{Cyl}_{\mathcal{C}_\pi(l)}(\gamma_{q_\circ}, \pi))$$

is equal to

$$\int_{t_1 \in \mathsf{Int}(\gamma_{q_\circ}, e_1)} \kappa_{\gamma_{q_\circ} + t_1}(e_1) \cdot \mathsf{P}_{\gamma_q}(\mathsf{Cyl}_{\mathcal{C}_\pi^{t_1}(l)}(\gamma_q, \pi(1))) d\mu_{\gamma_{q_\circ}}(t_1)$$

where $\pi(1) = e_z e_{12}$. Since we integrate with respect to a Dirac's distribution over $t_{enc(\mathsf{k})}$, we have that the previous integral is equal to

$$\mathsf{P}_{\gamma_q}(\mathsf{Cyl}_{\mathcal{C}_\pi^{t_{enc(\mathsf{k})}}(l)}(\gamma_q, \pi(1))) \quad (1)$$

Next, let $\mathcal{C}_\pi^{t_{enc(\mathsf{k})}}(l) = \mathcal{C}_1$ and then (1) is equal to

$$\begin{aligned}
&\int_{t_2 \in \mathsf{Int}(\gamma_q, e_z)} \kappa_{\gamma_q + t}(e_z) \cdot \mathsf{P}_{\gamma_{q'}}(\mathsf{Cyl}_{\mathcal{C}_1^{t_2}}(\gamma_{q'}, e_{12})) d\mu_{\gamma_q}(t_2) \\
&= p(z-1) \cdot \mathsf{P}_{\gamma_{q'}}(\mathsf{Cyl}_{\mathcal{C}_1^{z-1}}(\gamma_{q'}, e_{12})) \\
&= \frac{1}{10} \cdot \mathsf{P}_{\gamma_{q'}}(\mathsf{Cyl}_{\mathcal{C}_1^{z-1}}(\gamma_{q'}, e_{12})) \\
&= \frac{1}{10} \cdot \int_{t \in \mathsf{Int}(\gamma_{q'}, e_{12})} \kappa_{\gamma_{q'} + t}(e_{12}) \cdot 1_{\mathcal{C}_1^{z-1}}(t) d\mu_{\gamma_{q'}}(t) \\
&= \frac{1}{10} \cdot \int_{t \in [0, +\infty)} 6 \cdot \mathsf{exp}(-6 \cdot t) \cdot 1_{[0, +\infty)}(t) \cdot 1_{\mathcal{C}_1^{z-1}}(t) dt \quad (2)
\end{aligned}$$

Next, we distinguish between two subcases based on the constraint

$$\mathcal{C}_1^{z-1} = [l - (t_{enc(\mathsf{k})} + z - 1), +\infty)$$

First, let

$$L = l - (t_{enc(\mathsf{k})} + z - 1)$$

Now if $L < 0$, (2) is

$$\begin{aligned}
&\frac{1}{10} \cdot \int_{t \in [0, +\infty)} 6 \cdot \mathsf{exp}(-6 \cdot t) dt \\
&= \frac{1}{10}
\end{aligned}$$

Otherwise, if $L \geq 0$, (2) is

$$\begin{aligned}
&\frac{1}{10} \cdot \int_{t \in [L, +\infty)} 6 \cdot \mathsf{exp}(-6 \cdot t) dt \\
&= \frac{1}{10} \cdot \lim_{n \to \infty} [-\mathsf{exp}(-6 \cdot t)]_L^n \\
&= \frac{1}{10} \cdot \lim_{n \to \infty} -\mathsf{exp}(-6 \cdot n) + \mathsf{exp}(-6 \cdot L) \\
&= \frac{1}{10} \cdot \mathsf{exp}(-6 \cdot L)
\end{aligned}$$