Security in Communicating Hierarchical Transaction–based Timed Automata*

Damas P. Gruska[1], Andrea Maggiolo–Schettini[2], and Paolo Milazzo[2]

[1] Institute of Informatics, Comenius University – *gruska@fmph.uniba.sk*
[2] Dipartimento di Informatica, Università di Pisa – {*maggiolo,milazzo*}*@di.unipi.it*

Abstract. Communicating Hierarchical Transaction-based Timed Automata have been introduced to model systems performing long–running transactions. For these automata here a security concept is introduced, which is based on a notion of opacity and on the assumption that and attacker can not only observe public system activities, but also cause abortion of some of them. Different intruder capabilities as well different kinds of opacity are defined and the resulting security properties are investigated.

Keywords: communicating hierarchical timed automata, timing attacks, opacity, information flow, security, long–running transactions

1 Introduction

Opacity is one of the strongest security concepts as, with its help, many other security properties can be expressed (see [BKMR06]). Its origin can be traced to a concept of non-interference (see [GM82]), which assumes the absence of any information flow between private and public system activities. More precisely, systems are considered to be secure if from observations of their public activities no information about private activities can be deduced. This approach has found many reformulations for different formalisms, computational models and nature or "quality" of observations. All reformulations try to capture important aspects of system behaviour with respect to possible attacks against systems security, and often are tailored to some types of attacks.

Timing attacks have a particular position among attacks against systems security. They represent a powerful tool for "breaking" "unbreakable" systems, algorithms, protocols, etc. For example, by carefully measuring the amount of time required to perform private key operations, attackers may be able to find fixed Diffie-Hellman exponents, factor RSA keys, and break other cryptosystems (see [Ko96]). This idea was developed in [DKL98] where a timing attack against smart card implementation of RSA was conducted. In [HH99], a timing attack on the RC5 block encryption algorithm, in [SWT01] the one against the popular SSH protocol and in [FS00] the one against web privacy are described.

To perform different kinds of timing attacks attackers might exploit different capabilities. For example, for some attacks it is enough if an attacker can

* Work supported by European Science Foundation program AutoMathA and by the grants VEGA 1/3105/06 and APVV-20-P04805.

only observe the system to be attacked. For other attacks an attacker has to communicate with the system via public actions, either directly or by means of an embedded auxiliary system. Some attacks exploit the brute force of an attacker who can interrupt some system activities (by resetting system components, breaking communication links, etc). Particularly sensitive to such type of attacks are systems performing so called long–running transactions (LRTs). A LRT is composed by atomic activities that should be executed completely. Atomicity means that they are either successfully executed or no effect is observed if their execution fails. Partial executions of a LRT are not desirable, and, if they occur, they must be compensated for. Therefore, all the activities A_i in a LRT have a compensating activity B_i that can be invoked to recover from the effects of a successful execution of A_i if some failure occurs later. Hence from the computational point of view the system is robust with respect to abortion of some of its activities. However these abortions may lead to some information flow between classified and public system activities.

In [LMMT06] we have introduced Communicating Hierarchical Transaction–based Timed Automata (CHTTAs) to model LRTs. In this paper we investigate information flow based attacks for systems described with CHTTAs and attackers that can not only passively observe public system activities but also actively cause abortion of system activities. We model information flow by the notion of opacity for which we give different definition depending on the assumed capabilities of attackers. It is our purpose to use the introduced concepts to study security of LRTs. In this paper we discuss the subject briefly and we leave a complete treatment for a future work.

The paper is organized as follows. In Section 2 we recall CHTTAs. In Section 3 we study opacity, reformulating it for intruders who can also abort system activities. In Section 4 we discuss the application of our study to LRTs. In Section 5 we conclude.

2 Communicating Hierarchical Timed Automata

Let us assume a finite set of communication channels $\mathcal{C}$ with a subset $C_{Pub} \subseteq \mathcal{C}$. As usual, we denote with $a!$ the action of sending a signal on channel a and with $a?$ the action of receiving a signal on a.

Let us assume a finite set X of positive real variables called *clocks*. A *valuation* over X is a mapping $v : X \to \mathbb{R}^{\geq 0}$ assigning real values to clocks. Let V_X denote the set of all valuations over X. For a valuation v and a time value $t \in \mathbb{R}^{\geq 0}$, let $v + t$ denote the valuation such that $(v + t)(x) = v(x) + t$, for each clock $x \in X$.

The set of *constraints* over X, denoted $\Phi(X)$, is defined by the following grammar, where ϕ ranges over $\Phi(X)$, $x \in X$, $c \in \mathbb{Q}$ and $\sim \in \{<, \leq, =, \neq, >, \geq\}$:

$$\phi ::= x \sim c \,|\, \phi \wedge \phi \,|\, \neg\phi \,|\, \phi \vee \phi \,|\, true$$

We write $v \models \phi$ when *the valuation v satisfies the constraint ϕ*. Formally, $v \models x \sim c$ iff $v(x) \sim c$, $v \models \phi_1 \wedge \phi_2$ iff $v \models \phi_1$ and $v \models \phi_2$, $v \models \neg\phi$ iff $v \not\models \phi$, $v \models \phi_1 \vee \phi_2$ iff $v \models \phi_1$ or $v \models \phi_2$, and $v \models true$.

Let $B \subseteq X$; with $v[B]$ we denote the valuation resulting after resetting all clocks in B. More precisely, $v[B](x) = 0$ if $x \in B$, $v[B](x) = v(x)$, otherwise. Finally, with $\mathbf{0}$ we denote the valuation with all clocks reset to 0, namely $\mathbf{0}(x) = 0$ for all $x \in X$.

Definition 1. *A Transaction-based Timed Automaton (TTA) is a tuple* $A = (\Sigma, X, S, Q, q_0, \delta)$*, where:*

- $\Sigma \subseteq \{a!, a? \mid a \in C\}$ *is a finite set of labels;*
- X *is a finite set of clocks;*
- S *is a finite set of superstates;*
- $Q = L \cup S \cup \{\odot, \otimes\}$*, where* L *is a finite set of basic states and* $\odot$ *and* $\otimes$ *represent the special states commit and abort, respectively;*
- $q_0 \in L$ *is the initial state;*
- $\delta \subseteq (L \times \Sigma \cup \{\tau\} \times \Phi(X) \times 2^X \times Q) \cup (S \times \{\boxdot, \boxtimes\} \times Q)$ *is the set of transitions.*

A TTA is said to be flat *when* $S = \emptyset$*.*

Superstates are states that can be refined to automata (*hierarchical composition*). Note that from superstates in S only transitions with labels in $\{\boxdot, \boxtimes\}$ can be taken. We assume that $\odot$ and $\otimes$ are the final states of a TTA.

We now introduce CHTTAs as an extension of TTAs allowing superstate refinement and parallelism.

Definition 2. *Let* $\Sigma_{Pub} = \{a!, a? \mid a \in C_{Pub}\}$ *and* $\mathcal{A} = \{A^1, \ldots, A^n\}$ *be a finite set of TTAs, with* $A^i = (\Sigma^i, X^i, S^i, Q^i, q_0^i, \delta^i)$ *and such that there exists* m *(*$m < n$*) such that* A^j *is flat if and only if* $j \geq m$*. A* Communicating Hierarchical Transaction-based Timed Automaton *(*$CHTTA_{\mathcal{A}}^{\Sigma_{Pub}}$*) is given by:*

$$CHTTA_{\mathcal{A}}^{\Sigma_{Pub}} \ ::= \ \langle A^i, \mu \rangle \quad \mid \quad CHTTA_{\mathcal{A}}^{\Sigma_{Pub}} || CHTTA_{\mathcal{A}}^{\Sigma_{Pub}}$$

where μ *is a* hierarchical composition function $\mu : S^i \to CHTTA_{\{A^{i+1}, \ldots, A^n\}}^{\Sigma_{Pub}}$.

Parallelism allows concurrent execution of automata. Hierarchical composition allows refining superstates. Automata executed in parallel may communicate by synchronizing transitions labeled with a sending and a receiving action on the same channel. The set Σ_{Pub} contains sending and receiving actions on public channels. These actions may belong to the alphabets of TTAs in $\mathcal{A}$. Communications performed using non public channels are only allowed between components inside the same superstate or at top–level. Communication performed by using public channels have no restrictions.

Note that, by definition of $\mathcal{A}$ and μ, cyclic nesting is avoided. In the following, if it does not give rise to ambiguity, we may write CHTTA instead of $\text{CHTTA}_{\mathcal{A}}^{\Sigma_{Pub}}$. Finally, if A is a flat TTA, in $\langle A, \mu \rangle$ μ is an empty function.

Example 1. In Figure 1 we show an example of CHTTA. Superstates of the CHTTA are depicted as boxes and basic states as circles; initial states are represented as vertical segments. Transitions are labeled arrows in which labels τ and constraints *true* are omitted. Containment in boxes represents hierarchical

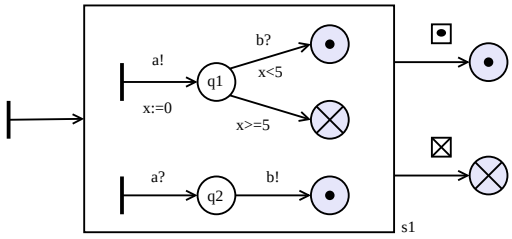

Fig. 1. Example of CHTTA.

composition, while parallel composition is represented by juxtapositions. The CHTTA in the figure is formally defined as $\langle(\emptyset, \emptyset, \{s_1\}, \{q_0, s_1, \odot, \otimes\}, q_0, \delta), \mu\rangle$, where $\delta = \{(q_0, \tau, true, \emptyset, s_1), (s_1, \boxdot, \odot), (s_1, \boxtimes, \otimes)\}$, and $\mu(s_1) = A_1||A_2$. A_1 and A_2 are defined as $A_1 = \langle(\{a!, b?\}, \{x\}, \emptyset, \{q_0, q_1, \odot, \otimes\}, q_0, \delta_1)$ and $A_2 = \langle(\{a?, b!\}, \emptyset, \emptyset, \{q_0, q_2, \odot, \otimes\}, q_0, \delta_2)$, with $\delta_1 = \{(q_0, a!, true, \{x\}, q_1), (q_1, b?, x < 5, \emptyset, \odot), (q_1, \tau, x \geq 5, \emptyset, \otimes)\}$ and $\delta_2 = \{(q_0, a?, true, \emptyset, q_2), (q_2, b!, true, \emptyset, \odot)\}$.

Configurations of CHTTAs are pairs $tc = (c, \nu)$ where c, the *untimed configuration*, represents the currently active states, and ν, the *composed valuation*, represents the current clock valuations. The configuration of a CHTTA without parallel components, when the currently active state is a basic state, is a pair (q, v) with q the currently active state, and v the automaton clock valuation. We represent with $q.c$ the configuration where q is a superstate and c is the untimed configuration of $\mu(q)$, and with $v.\nu$ the composed valuation where v is the clock valuation of the automaton having q as superstate and ν is the composed valuation of the clocks of $\mu(q)$. We denote with $c_1; c_2$ the untimed configuration of the parallel composition of two CHTTAs having c_1 and c_2 as untimed configurations. Analogously, we denote with $\nu_1; \nu_2$ the composed valuation of the parallel composition of two CHTTAs having ν_1 and ν_2 as composed valuations. Formally, the set of configurations $Conf(A)$ of a CHTTA A is inductively defined as follows:

- if $A = \langle(\Sigma, X, S, Q, q_0, \delta), \mu\rangle$, then $Conf(A) = \{(Q \backslash S) \times V_X\} \cup \{(q.c, v.\nu) \mid q \in S \wedge v \in V_x \wedge (c, \nu) \in Conf(\mu(q))\}$;
- if $A = A_1||A_2$ then $Conf(A) = \{(c_1; c_2, \nu_1; \nu_2) \mid (c_1, \nu_1) \in Conf(A_1) \wedge (c_2, \nu_2) \in Conf(A_2)\}$.

For a composed valuation ν and a time value $t \in \mathbb{R}^{\geq 0}$, let $\nu + t$ denote the composed valuation such that $(v + t)(x) = v(x) + t$, for each valuation v in ν.

The initial configuration of A, denoted $Init(A) \in Conf(A)$, is the configuration (c, ν) such that each state occurring in c is an initial state and each valuation occurring in ν is $\mathbf{0}$.

We give a semantics of CHTTAs as a labeled transition system where states are pairs (A, tc) with $A \in \text{CHTTA}_{\mathcal{A}}^{\Sigma_{Pub}}$ and $tc \in Conf(A)$, and labels are in $\Lambda = \mathbb{R}^{>0} \cup \bigcup_i \Sigma^i \cup \{\tau\}$. In order to simplify the semantics we introduce a notion of structural equivalence for pairs (A, tc), accounting for commutativity and associativity of parallelism. The relation $\approx$ is the least equivalence relation satisfying $(A_1||A_2, tc_1; tc_2) \approx (A_2||A_1, tc_2; tc_1)$ and $(A_1||(A_2||A_3), tc_1; (tc_2; tc_3)) \approx$

$((A_1||A_2)||A_3, (tc_1; tc_2); tc_3)$. Moreover, given an untimed parallel configuration $c = c_1; \ldots; c_n$ we use the following notations: $c \approx \odot$ if $\forall i.c_i = \odot$, and $c \approx \otimes$ if $\exists i.c_i = \otimes \wedge \forall i \neq j.c_j \in \{\odot, \otimes\}$.

Definition 3 (Semantics of CHTTAs). *Given $A \in CHTTA_{\mathcal{A}}^{\Sigma_{Pub}}$, the semantics of a A is the least labeled transition relation $\xrightarrow{\alpha}$ over $\{A\} \times Conf(A)$ closed with respect to structural equivalence and satisfying the following rules:*

$$\frac{t \in \mathbb{R}^{>0}}{(A,(c,\nu)) \xrightarrow{t} (A,(c,\nu+t))} \qquad \frac{(q,\alpha,\phi,B,q') \in \delta \quad v \models \phi \quad q' \notin S}{(\langle A,\mu\rangle,(q,v)) \xrightarrow{\alpha} (\langle A,\mu\rangle,(q',v[B]))} \tag{T,C1}$$

$$\frac{(q,\alpha,\phi,B,q') \in \delta \quad v \models \phi \quad q' \in S \quad Init(\mu(q')) = (c,\nu)}{(\langle A,\mu\rangle,(q,v)) \xrightarrow{\alpha} (\langle A,\mu\rangle,(q'.c,v[B].\nu))} \tag{C2}$$

$$\frac{(\mu(q),(c,\nu)) \xrightarrow{\alpha} (\mu(q),(c',\nu')) \quad \alpha \in \Sigma_{Pub} \cup \{\tau\}}{(\langle A,\mu\rangle,(q.c,v.\nu)) \xrightarrow{\alpha} (\langle A,\mu\rangle,(q.c',v.\nu'))} \tag{C3}$$

$$\frac{(A_1,(c_1,v)) \xrightarrow{\alpha} (A_1,(c_1',v')) \quad \alpha \in \Sigma_{Pub} \cup \{\tau\}}{(A_1||A_2,(c_1;c_2,v)) \xrightarrow{\alpha} (A_1||A_2,(c_1';c_2,v'))} \tag{P1}$$

$$\frac{(A_1,(c_1,v)) \xrightarrow{a!} (A_1,(c_1',v')) \quad (A_2,(c_2,v')) \xrightarrow{a?} (A_2,(c_2',v''))}{(A_1||A_2,(c_1;c_2,v)) \xrightarrow{\tau} (A_1||A_2,(c_1';c_2',v''))} \tag{P2}$$

$$\frac{c \approx \odot \quad (q,\boxdot,q') \in \delta \quad q' \notin S}{(\langle A,\mu\rangle,(q.c,v.\nu)) \xrightarrow{\tau} (\langle A,\mu\rangle,(q',v))} \qquad \frac{c \approx \odot \quad (q,\boxdot,q') \in \delta \quad q' \in S \quad Init(\mu(q')) = (c',\nu')}{(\langle A,\mu\rangle,(q.c,v.\nu)) \xrightarrow{\tau} (\langle A,\mu\rangle,(q'.c',v.\nu'))} \tag{Com1,2}$$

$$\frac{c \approx \otimes \quad (q,\boxtimes,q') \in \delta \quad q' \notin S}{(\langle A,\mu\rangle,(q.c,v.\nu)) \xrightarrow{\tau} (\langle A,\mu\rangle,(q',v))} \qquad \frac{c \approx \otimes \quad (q,\boxtimes,q') \in \delta \quad q' \in S \quad Init(\mu(q')) = (c',\nu')}{(\langle A,\mu\rangle,(q.c,v.\nu)) \xrightarrow{\tau} (\langle A,\mu\rangle,(q'.c',v.\nu'))} \tag{Ab1,2}$$

where $A = (\Sigma, X, S, Q, q_0, \delta)$ except for rule (T) where A is any CHTTA.

Rule (T) allows the elapsing of time for a generic CHTTA A. We note that the time t is the same for any TTA composing A. Rules (C1) and (C2) describe the behavior of a flat TTA. From a configuration (q, v), the step is performed due to a transition (q, α, ϕ, B, q') such that the condition ϕ is satisfied by v. After the step, the flat TTA is in the configuration composed by state q' and where clocks in B are reset. If q' is a superstate (rule (C2)), then the CHTTA $\mu(q')$ becomes active inside q'. The synchronization step is described by rule (P2). The relation $\approx$ allows CHTTAs that are not neighbors in the parallel composition to communicate. Rules (C3) and (P1) allow expanding the step of a TTA which is a component of a CHTTA. Rule (C3) deals with the hierarchical composition and rule (P1) deals with the parallel composition. The label of the step is either τ or a public channel. Hence, thanks to rule (P2), communication between TTAs in parallel is allowed both for private and public channels, while for TTAs in different superstates the communication is allowed only if the channel is public. Moreover, we note that the step we are expanding cannot be a time step. Hence, since time steps can be performed only by the root, the time elapsed is the same for each TTA composing the CHTTA we are considering.

Each execution of a superstate terminates with either a commit or an abort state. Rules (Com1) and (Com2) deal with the case in which the commit of the

superstate takes the TTA to a basic state or to a superstate, respectively, and rules (Ab1) and (Ab2) deal with the case in which the abort of the superstate takes the TTA to a basic state or to a superstate, respectively.

Given a string $w = \alpha_1 \dots \alpha_m$, we will write $(A,(c,\nu)) \stackrel{w}{\Longrightarrow} (A,(c',\nu'))$ to denote the existence of a sequence of steps $(A,(c,\nu)) \stackrel{\alpha_1}{\longrightarrow} \dots \stackrel{\alpha_m}{\longrightarrow} (A,(c',\nu'))$. We denote with $|w| = m$ the length of w and with $w[i] = \alpha_i$ the $i-$th label.

Definition 4 (Language accepted by a CHTTA). *With $\mathcal{L}(A,\Sigma_V)$ we denote* the language accepted by a CHTTA A w.r.t. a set of visible actions $\Sigma_V \subseteq \Sigma_{Pub}$. *Namely, $\mathcal{L}(A,\Sigma_V) = \{w \in (\{\tau\} \cup \Sigma_V \cup \mathbb{R}^{>0})^* \mid (A,Init(A)) \stackrel{w}{\Longrightarrow} (A,(\odot,\nu'))$ or $(A,Init(A)) \stackrel{w}{\Longrightarrow} (A,(\otimes,\nu'))\}$. By $\mathcal{L}^p(A,\Sigma_V)$ we will denote the set of all prefixies of elements from $\mathcal{L}(A,\Sigma_V)$ i.e $\mathcal{L}^p(A,\Sigma_V) = \{w|$ such that $w.w' \in \mathcal{L}(A,\Sigma_V)$ for some $w'\}$.*

3 Information Flow in CHTTAs

In this section we will formalize a notion of attacks on system security that are based on an information flow between invisible (private) and visible (public) system activities. We assume that an attacker is just an eavesdropper who can see a part of the system behaviour and tries to deduce from this observation some classified information. In the case of timing attacks, time of occurrences of observed events plays a crucial role, namely, timing of actions represents a fundamental information.

To formalize the attacks we do not divide actions into public and private ones at the system description level, as it is done for example in [GM04,BG04], but we use a more general concept of observation. This concept was recently exploited in [BKR04] and [BKMR06] in a framework of Petri Nets and transition systems, respectively, where opacity is defined with the help of observations. First we reformulate a notion of observation function.

Definition 5 (Observation). *Let $\Lambda = \mathbb{R}^{>0} \cup \bigcup_i \Sigma^i \cup \{\tau\}$ and Θ be a set of elements called observables. Any function $obs : \Lambda^\star \to \Theta^\star$ is an observation function. It is called static /dynamic /orwellian / m-orwellian ($m \geq 1$) if the following conditions hold respectively (below we assume $w = x_1 \dots x_n$):*

- *static if there is a mapping $obs' : \Lambda \to \Theta \cup \{\epsilon\}$ such that for every $w \in \Lambda^\star$ it holds $obs(w) = obs'(x_1) \dots obs'(x_n)$,*
- *dynamic if there is a mapping $obs' : \Lambda^\star \to \Theta \cup \{\epsilon\}$ such that for every $w \in \Lambda^\star$ it holds $obs(w) = obs'(x_1).obs'(x_1.x_2) \dots obs'(x_1 \dots x_n)$,*
- *orwellian if there is a mapping $obs' : \Lambda \times \Lambda^\star \to \Theta \cup \{\epsilon\}$ such that for every $w \in \Lambda^\star$ it holds $obs(w) = obs'(x_1,w).obs'(x_2,w) \dots obs'(x_n,w)$,*
- *m-orwellian if there is a mapping $obs' : \Lambda \times \Lambda^\star \to \Theta \cup \{\epsilon\}$ such that for every $w \in \Lambda^\star$ it holds $obs(w) = obs'(x_1,w_1).obs'(x_2,w_2) \dots obs'(x_n,w_n)$ where $w_i = x_{max\{1,i-m+1\}}.x_{max\{1,i-m+1\}+1} \dots x_{min\{n,i+m-1\}}$.*

In the case of the static observation function each action is observed independently from its context. In case of the dynamic observation function an observation of an action depends on the previous ones, in case of the orwellian and m-orwellian observation function an observation of an action depends on the all and m previous actions in the sequence, respectively. The static observation function is the special case of m-orwellian one for $m = 1$. Note that from the practical point of view the m-orwellian observation functions are the most interesting ones. An observation expresses what an observer - eavesdropper can see from a system behaviour and we will alternatively use both the terms (observation - observer) with the same meaning.

Now suppose that we have some security property. This might be an execution of one or more classified actions, an execution of actions in a particular classified order which should be kept hidden, etc. Suppose that this property is expressed by a predicate ϕ over sequences. We would like to know whether the observer can deduce the validity of the property ϕ just by observing a sequence from $\mathcal{L}^p(A, \Sigma_V)$. The observer cannot deduce the validity of ϕ if there are two sequences $w, w' \in \mathcal{L}^p(A, \Sigma_V)$ such that $\phi(w), \neg\phi(w')$ and the sequences cannot be distinguished by the observer i.e. $obs(w) = obs(w')$. We formalize this concept by the notion of opacity.

Definition 6 (Opacity). *Given a CHTTA A, a predicate ϕ over $\mathcal{L}^p(A, \Sigma_V)$ is opaque w.r.t. the observation function obs if for every sequence w, $w \in \mathcal{L}^p(A, \Sigma_V)$ such that $\phi(w)$ holds, there exists a sequence $w', w' \in \mathcal{L}^p(A, \Sigma_V)$ such that $\neg\phi(w')$ holds and $obs(w) = obs(w')$. The set of CHTTAs for which the predicate ϕ is opaque with respect to obs will be denoted by Op^{ϕ}_{obs}.*

The notion of opacity is rather general. With its help many other security properties can be defined (anonymity, non-interference etc.) [BKMR06]. On the other side opacity, is undecidable even for the simplest possible observation function, namely for the constant one, and for finite state processes.

Theorem 1. *Opacity for CHTTA is undecidable.*

Proof. Let us consider an instance of the Post Correspondence Problem with (u_i, v_i) for $i = 1, \ldots, n$. Let A be automaton with only two states and transitions $\delta = \{(q_0, \tau, true, \emptyset, \odot)\} \cup \{(q_0, u_i, true, \emptyset, q_0),\ (q_0, v_i, true, \emptyset, q_0) \mid i \in 1, \ldots, n\}$ and let $obs(w) = \epsilon$ for every $w \in \mathcal{L}^p(A, \Sigma_V)$. We define $\phi(i_1 \ldots i_m)$ to be true iff $u_{i_1} \ldots u_{i_m} \neq v_{i_1} \ldots v_{i_m}$. It is easy to see that the Post Corresponding Problem has a solution if ϕ is opaque with respect to obs. □

Hence there is the need of formalizing a variant of opacity which is decidable but still practically useful, i.e. such that with its help basic security notions could be still expressed.

There are two problems with the undecidability of opacity: on one side it is a rather powerful notion of observation functions (both dynamic and orwellian ones consider a potentially infinite memory to store actions and subsequently to compute observations) and, on the other side, it is the power of predicate ϕ that

itself might be difficult to compute. We overcome these obstacles by expressing both an opacity function and predicate ϕ by CHTTAs. We can model the observation function by a deterministic automaton O that, via a communication mechanism, communicates with the original process and produces observables. In this way we have that $A \stackrel{w}{\Rightarrow}$ iff $(A|O) \stackrel{obs(w)}{\Rightarrow}$. By means of the automaton O we can model also observations which are unprecise with respect to timing of activities, namely, as an example, the case when an observer cannot measure time with absolute accuracy.

Predicates ϕ and $\neg\phi$ will be also modeled by deterministic automata E_ϕ and $E_{\neg\phi}$. Roughly speaking, these take as an input a sequence w of automaton A and on the output produce the same sequence followed by a new special action *true* if $\phi(w)$ or $\neg\phi(w)$, respectively. By ϕ we will denote also the set of sequences satisfying ϕ. From now on we will assume that $O, E_\phi, E_{\neg\phi}$ are deterministic CHTTAs. Note that by O we can still model the most interesting set of observational functions, namely m-orwellian ones. Now with help of O, E_ϕ and $E_{\neg\phi}$ we define the reduced opacity (r-opacity) property.

Definition 7. *Let A be a CHTTA. We say that A is r-opaque with respect to observation automaton O and automata E_ϕ and $E_{\neg\phi}$ iff*

$$\mathcal{L}^p(((A||E_\phi)||O), \Sigma_V) \subseteq \mathcal{L}^p(((A||E_{\neg\phi})||O), \Sigma_V).$$

The set of CHTTAs that are r-opaque with respect to $O, E_\phi, E_{\neg\phi}$ will be denoted by r-Op_O^ϕ.

Note that the most of reasonable observations and security requirements over sequences can be expressed by timed automata $O, E_\phi, E_{\neg\phi}$ and r–Op_O^ϕ.

Now we prove that r-opacity is decidable. First we recall from [LMMT06] the following theorem which claims that for any CHTTA there is a flat automaton which can perform the same sequences of actions. As a consequence we have that the class of CHTTAs is equivalent to the class of Timed automata.

Theorem 2. *Let A be a CHTTA; it holds that $(A, (c_0, v_0)) \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} (A, (c_n, v_n))$ is a sequence of steps of A iff $(A', (c_0, v'_0)) \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} (A', (c_n, v'_n))$ is a sequence of steps of A' where $A' = Flat(A, \Sigma_V)$.*

Theorem 3. *The property of r-opacity for deterministic CHTTAs is decidable.*

Proof. The main idea. According to Theorem 2, for both the automata $((A||E_\phi)||O)$ and $((A||E_{\neg\phi})||O)$ there exist corresponding flat automata. Moreover it can be shown that these flat automata are deterministic since automata $A, E_\phi, E_{\neg\phi}, O$ are deterministic. The rest of the proof follows from the fact that language inclusion is decidable for deterministic timed automata. □

Till now we have omitted the discussion about abortions as a tool for performing timing attacks. Suppose that some abortion could be provoked by an intruder. This means that $\boxtimes$ becomes an input non-public action and to distinguish different occurrences of such actions we will use indexes. More precisely,

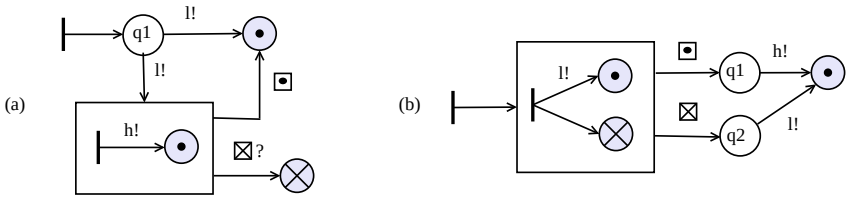

Fig. 2. Examples of abortion–opened CHTTAs.

we assume that there might be actions that cannot be aborted by the intruder and actions that can be aborted. It is a task of a designer of system A to identify those "weak" places and replace $\boxtimes$ by $\boxtimes_i?$. We will call such resulting automaton an *abortion-opened* automaton and we will denote it by A_a. The intruder forces an abortion of a corresponding activity by performing $\boxtimes_i!$. Note that for actions $\boxtimes_i?, \boxtimes_i!$ only the rule P2 from Definition 3 will be applied. Hence we model every intruder as an automaton I that can perform only transitions labeled by $\boxtimes_i!$. We will call an intruder *trivial* either if it cannot abort any action or it can always abort any action. Now we can define r-opacity with respect to some intruder I.

Definition 8. *Let A be a CHTTA. We say that A is r-opaque with respect to observation automaton O, intruder I and automata E_ϕ and $E_{\neg\phi}$ iff $(A_a||I)$ is r-opaque with respect to observation automaton O and automata E_ϕ and $E_{\neg\phi}$ for every abortion-opened CHTTA A_a obtained from A.*

The set of CHTTAs which are r-opaque with respect to $O, I, E_\phi, E_{\neg\phi}$ will be denoted by $r\text{-}Op^\phi_{IO}$.

The relationship between $r\text{-}Op^\phi_O$ and $r\text{-}Op^\phi_{IO}$ is stated in the following theorem.

Theorem 4. $r\text{-}Op^\phi_{IO} \subseteq r\text{-}Op^\phi_O$.

Proof. Let $A \in r\text{-}Op^\phi_{IO}$. That means that $(A_a||I)$ is r-opaque for every abortion-opened CHTTA A_a obtained from A. Hence $(A_a||I) \in r\text{-}Op^\phi_O$ for any A_a and hence also for such A_a that intruder I cannot abort any action of A_a, i.e. $(A_a||I)$ and A perform the same sequences of actions, and therefore we get $A \in r\text{-}Op^\phi_O$ what proves that $r\text{-}Op^\phi_{IO} \subseteq r\text{-}Op^\phi_O$. □

Note that the inclusion from Theorem 4 is proper if the intruder is non-trivial and predicate ϕ expresses, for example, the property that a sequence contains the private action h (see Fig. 2 a)).

If we consider only intruders that can be represented by deterministic automata we obtain a similar property as the one holding for r-opacity (see Theorem 3) and its proof is also similar.

Theorem 5. *The property of r-opacity for deterministic CHTTAs and deterministic intruders is decidable.*

As an extreme case we might consider a situation when at any time any activity can be aborted. This might be modeled by replacing every $\boxtimes$ by $\boxtimes?$ and by automaton which can at any time perform $\boxtimes!$. Instead of this we can model this type of attacks simply by putting $\boxtimes$ among public and visible actions. We say that action x is visible with respect to observation function iff $obs(u.x.v) \neq obs(u.v)$.

Definition 9. *Let A be a CHTTA. We say that A is ar-opaque with respect to observation automaton O for which $\boxtimes$ is visible and automata E_ϕ and $E_{\neg\phi}$ iff*

$$\mathcal{L}^p(((A||E_\phi)||O), \Sigma_V \cup \{\boxtimes\}) \subseteq \mathcal{L}^p(((A||E_{\neg\phi})||O), \Sigma_V \cup \{\boxtimes\}).$$

The set of CHTTAs which are ar-opaque with respect to $O, E_\phi, E_{\neg\phi}$ will be denoted by $ar\text{-}Op_O^\phi$.

Theorem 6. $ar\text{-}Op_O^\phi \subseteq r\text{-}Op_{IO}^\phi$.

Proof. Sketch. If $A \in ar\text{-}Op_O^\phi$ then all possible abortions are visible and despite this fact there is no information flow. Hence $A \in r\text{-}Op_{IO}^\phi$ since here only some of abortions are visible by I. □

Note that again the inclusion in Theorem 6 is proper for nontrivial intruders (see Fig. 2 b)).

As regards timing of actions it is not clear from the above mentioned security concepts whether possible information flow is due to time information contained in observations or not. In other words, whether there is a danger of timing attack or not. To formalize this concept, let us assume an untimed version obs_t of observation obs i.e. $obs_t(w) = obs_t(w_t) = obs(w_t)$, where w_t is obtained from w by removing all timing information $x \in \mathbb{R}^{>0}$. By $\mathcal{L}^p(A, \Sigma_V)_t$ we will denote sequences from $\mathcal{L}^p(A, \Sigma_V)$ from which timing information is removed.

Now we can formalize a notion of being opened to timing attacks.

Definition 10 (Opening for Timing Attacks). *Let A be CHTTA. We say that A is opened to timing attacks with respect to predicate ϕ over $\mathcal{L}^p(A, \Sigma_V)$ and the observation function obs if for every sequence w, $w \in \mathcal{L}^p(A, \Sigma_V)$ such that $\phi(w)$ holds, there exists a sequence $w' \in \mathcal{L}^p(A, \Sigma_V)$ such that $\neg\phi(w')$ holds and $obs_t(w) = obs_t(w')$ and there exists a sequence $w \in \mathcal{L}^p(A, \Sigma_V)$ such that $\phi(w)$ holds, but there is not a sequence $w' \in \mathcal{L}^p(A, \Sigma_V)$ such that $\neg\phi(w')$ holds and $obs(w) = obs(w')$.*

Now we define a restricted version of the above mentioned notion. Since observations that do not see elapsing of time cannot be directly modeled by a timed automaton, we use untimed sequences.

Definition 11 (Restricted Opening for Timing Attacks). *Let A be a CHTTA. We say that A is opened to timing attacks with respect to observation automaton O and automata E_ϕ and $E_{\neg\phi}$ iff*

$$\mathcal{L}^p(((A||E_\phi)||O), \Sigma_V)_t \subseteq \mathcal{L}^p(((A||E_{\neg\phi})||O), \Sigma_V)_t.$$

but $A \notin r\text{-}Op^\phi_O$.

This notion could be formulated similarly as for the other two types of opacity, $r\text{-}Op^\phi_{IO}$ and $ar\text{-}Op^\phi_O$.

4 Long–Running Transactions as CHTTAs

A *long–running transaction* (LRT) is a composition of *atomic activities* that are either successfully executed (*committed*) or no effect is observed if their execution fails (*aborted*). Partial executions of a long–running transaction are not desirable, and, if they occur, they must be compensated for. Hence, all the activities A_i in a long–running transaction have a compensating activity B_i that can be invoked to repair from the effects of a successful execution of A_i if some failure occurs later. Compensations are assumed to be activities that always complete their execution successfully (they always commit and can never abort).

Two usual ways of composing transactional activities (including compensations) are sequentially and in parallel. For this reason, given activities $A_1, \ldots, A_n \in \text{CHTTA}^{\Sigma_{Pub}}_{\mathcal{A}}$ and compensations $B_1, \ldots, B_n \in \text{CHTTA}^{\Sigma_{Pub}}_{\mathcal{A}}$, we can define a language for LRTs as follows:

$$T \;::=\; A_i \text{⇂} B_i \;\;\Big|\;\; T \cdot T \;\;\Big|\;\; T||T\,.$$

The LRT $A \text{⇂} B$ denotes the association of the atomic activity A with compensation B. Given two LRTs T_1 and T_2, with $T_1 \cdot T_2$ we denote their sequential composition and with $T_1||T_2$ their parallel composition.

In the sequential composition of n transactional activities $A_1 \text{⇂} B_1 \cdot \ldots \cdot A_n \text{⇂} B_n$, either the entire sequence $A_1, \ldots, A_n$ is executed or the compensated sequence $A_1, \ldots, A_i, B_i, \ldots, B_1$ is executed for some $i < n$. The first case means that all activities in the sequence completed successfully, and the second one stands for the abort of activity A_{i+1}; hence, all the activities already completed $(A_1, \ldots, A_i)$ are recovered by executing the compensating activities $(B_i, \ldots, B_1)$.

In the parallel composition of n transactional activities $A_1 \text{⇂} B_1 || \ldots || A_n \text{⇂} B_n$, all the atomic activities are assumed to be executed concurrently, and the whole transaction terminates when all of them complete. If some A_i aborts, then compensation activities should be invoked for the activities that completed successfully. In this latter case, the result of the whole transaction is "abort".

In [LMMT06] the sequential and parallel compositions of LRTs have been formulated as compositions of CHTTAs. For both compositions the two properties of *correct completion* and *correct compensation* have been proved, namely it has been shown that for each pattern the corresponding CHTTAs reach commit and abort states, and activate compensations accordingly to the semantics of the composition. As a consequence, CHTTAs can be used as a model of LRTs, and the problem of assessing opacity of a predicate on the execution of a LRT

can be reduced to the problem of assessing opacity of the same predicate on the CHTTA modeling the LRT. Now, it would be interesting to check whether the various notions of opacity we have defined are preserved by the composition operations of LRTs. We leave this study as future work.

5 Conclusions and further work

In this paper we have presented a formal model which can express robustness of systems with respect to timing attacks and intruders which can not only observe system activities but also interrupt some or all of them.

Further study will concern more efficient decision algorithms and decidability results also for some classes of nondeterministic automata. Moreover we plan to study some specific classes of observation functions or predicates over sequences of actions and compositionality results for the studied security properties.

References

[BKR04] Bryans J., M. Koutny and P. Ryan: Modelling non-deducibility using Petri Nets. Proc. of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models, 2004.

[BKMR06] Bryans J., M. Koutny, L. Mazare and P. Ryan: Opacity Generalised to Transition Systems. In Proceedings of the Formal Aspects in Security and Trust, LNCS 3866, Springer, Berlin, 2006

[BG04] Busi N. and R. Gorrieri: Positive Non-interference in Elementary and Trace Nets. Proc. of Application and Theory of Petri Nets 2004, LNCS 3099, Springer, Berlin, 2004.

[DKL98] Dhem J.-F., F. Koeune, P.-A. Leroux, P. Mestre, J.-J. Quisquater and J.-L. Willems: A practical implementation of the timing attack. Proc. of the Third Working Conference on Smart Card Research and Advanced Applications (CARDIS 1998), LNCS 1820, Springer, Berlin, 1998.

[FS00] Felten, E.W., and M.A. Schneider: Timing attacks on web privacy. Proc. 7^{th} ACM Conference on Computer and Communications Security, 2000.

[GM82] Goguen J.A. and J. Meseguer: Security Policies and Security Models. Proc. of IEEE Symposium on Security and Privacy, 1982.

[GM04] Gorrieri R. and F. Martinelli: A simple framework for real-time cryptographic protocol analysis with compositional proof rules. to appear at Science of Computer Programing.

[HH99] Handschuh H. and Howard M. Heys: A timing attack on RC5. Proc. Selected Areas in Cryptography, LNCS 1556, Springer, Berlin, 1999.

[Ko96] Kocher P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. Proc. Advances in Cryptology - CRYPTO'96, LNCS 1109, Springer, Berlin, 1996.

[LMMT06] Lanotte R., A. Maggiolo–Schettini, P. Milazzo and A. Troina: Modeling Long-running Transactions with Communicating Hierarchical Timed Automata. Proc. of Formal Methods for Open Object-Based Distributed Systems (FMOODS'06), LNCS 4037, Springer, Berlin, 2006.

[SWT01] Song. D., D. Wagner, and X. Tian: *Timing analysis of Keystrokes and SSH timing attacks.* Pro.10th USENIX Security Symposium, 2001.