Security of Low Level IoT Protocols *

Damas P. Gruska[1] and M. Carmen Ruiz[2]

[1] Institute of Informatics, Comenius University, Bratislava, Slovakia
gruska@fmph.uniba.sk
[2] Universidad de Castilla-La Mancha, Albacete, Spain
mcarmen.ruiz@uclm.es

Abstract. Application of formal methods in security is demonstrated. Formalism for description of security properties of low level IoT protocols is proposed. It is based on security property called infinite step opacity. We prove some of its basic properties as well as we show its relation to other security notions. Finally, complexity issues of verification and security enforcement are discussed. As a working formalism timed process algebra is used.

Keywords: formal methods, security, protocol, opacity, process algebra, information flow

1 Introduction

The Internet of Things (IoT) is finally a reality. Smart devices, smart phones, smart cars, smart homes, smart industries, in short a smart world. There are multiple predictions which declare that there will be at least tens of billions connected devices by 2020, almost everything from personal items such as pacemakers to aircraft black boxes could all become part of this interconnected world. But each device connected increases privacy and security concerns surrounding the IoT. These concerns range from hackers stealing our data and even threatening our lives to how corporations can easily uncover private data we carelessly give them. So that, IoT transforms our whole world bringing great benefits but also new risks due to the fact that we need to worry about protecting more and more devices.

The great revolution brought about by IoT involves the emergence of new protocols, networks, sensors, devices and, of course, new security requirements. Nevertheless, all these changes are occurring at such a speed that new protocols come into operation before they are properly evaluated. Due to this lack of prior study motivated by the need for their early use, malfunctions arise when the protocols are already in use, causing the appearance of new versions or "patches".

* This study was funded in part by the grant VEGA 1/0778/18, the Spanish Ministry of Science and Innovation and the European Union FEDER Funds under grant CAS18/00106 and TIN2015-65845-C3-2-R and by the JCCM regional project SBPLY/17/180501/000276

A nice example how a network of IoT can be jeopardized in several ways can be found in [20]. We came to this issue even in our own works. In [13] we propose a new packet format and a new Bluetooth Low Energy (BLE) mesh topology, with two different configurations: Individual Mesh and Collaborative Mesh. To include user devices is a security challenge due to the physical accessibility to sensors, actuators and objects, and the openness of the systems, and the fact that most devices will communicate wirelessly. Other examples could be found in [4] where we deploy a collaborative mesh network based on BLE and long range wide-area network (LoRaWAN) technologies and in [22] where it is presented a BLE wearable which is aimed at enhancing working conditions and efficiency in Industry 4.0 scenarios. In both cases we feel some lack of basic security requirements. Therefore, from our experience, we can say that IoT security can not be an afterthought or an add-on and needs to be addressed from the earliest stages of development.

Simulations and tests represent the common validation techniques but we advocate the use of formal methods for the evaluations for several reasons: simulation results depend on the simulator and may vary from those obtained in real scenarios. At this point, it would be easy to argue that the results obtained by these methods will also depend on the formalism used, but in defense of formal methods, we should say that the underlying mathematical models in simulators are often not clear and are not accessible to users while formal methods are supported by rigorous mathematical basis which allow users to model all the required properties, abstracting from the details that are not relevant. Moreover, simulation and testing can only show the presence of errors, not their absence but to rule out errors we must consider all possible executions. This can be made by means of formal methods that provide correct results that cover the full behavior of the models. At present there are few works where formal methods are used in IoT and they are focused on the field of automotive.

In this paper, we focus on formal methods applied to the field of security of IoT protocols. Here communication issues are considered to be weakest point. Many new, general or proprietary protocols are often vulnerable to different types of attacks. Timing attacks represent a powerful tool for breaking "unbreakable" systems, algorithms, protocols, etc. In the literature we can find examples how strong security algorithms or its components (fixed Diffie-Hellman exponents, RSA keys, RC5 block encryption, SSH protocol or web privacy, see [17, 1, 12, 23, 2] could be compromised by timing attacks. All these attacks employ so called information flow. Information flow based security properties (see [5]) are based on an absence of information flow between private and public data or activities. Systems are considered to be secure if from observations of their public data or activities no information about private data or activities can be obtained. This approach has found many reformulations and among them opacity could be considered as the most general one (for an overview of various types of opacity of discrete event systems see [14]).

Opacity could be, in general, formulated in the following way. Let us assume a (security) property. This could be an execution of actions in a particular classified order, time stamps of actions, execution of a private action etc. Let predicate ϕ over process's traces expresses such the property. An observer cannot deduce the validity of the property ϕ just by observing system's behaviour (traces) if there are two traces w, w' such that $\phi(w)$ holds and $\phi(w')$ does not hold and the traces w, w' cannot be distinguished by the observer.

Many security properties are special cases of opacity (see, for example, [8, 9]). In [10] process opacity is introduced in such a way that instead of traces produced by processes we concentrate on reachable states and their properties. We assume an intruder who wants to discover whether given process reaches a state for which some given predicate holds. In this way we could express some new security properties. On the other hand some security flaws, particularly important for IoT protocols, are not covered by this state-based security property neither by its variant called an initial state opacity, studied in [11].

In this paper we define infinite state opacity for timed process algebras, we show its properties and relation to other security notions. The property is based on a similar concept defined for discrete event structures (see [14]) but some modifications are needed. In this way we obtain the security property with some practical value, since timed process algebra, as our working formalism, can be used for description of many communication protocols. Moreover, there are well developed techniques and software tools for process algebra formal verification and they can be employed for security checking. We start with CCS (Milners's Calculus of Communicating Systems, see [19]) to which we add means to express time behaviour. Then we formalize infinite step opacity for such process algebra, called TPA.

The organization of the paper is the following. In Section 2 we describe the our working formalism, a special timed process algebra TPA. In next Section we present and study infinite step opacity, we show its properties, relations to other security notions and some complexity issues. In Section 4 we discuss obtained results and plans for future work.

2 TPA

In this section we introduce an extension of CCS (Milners's Calculus of Communicating Systems, see [19]). This extension will be called Timed Process Algebra, TPA for short, and it enriches CCS by the special time action t. This action will express elapsing of time. TPA represents a simplification of Timed Security Process Algebra (tSPA) introduced in [3]. An explicit idling operator ι used in tSPA is omitted since we allow implicit idling of processes. Processes can idle either by performing t actions or they can also idle if there is no internal communication possible. We also do not need division of actions between high and low level ones. TPA also strictly preserves *time determinancy* contrary to tCryptoSPA (see [6])..

First we assume a set of atomic actions A such that $\tau, t \notin A$. For every $a \in A$ there exists $\overline{a} \in A$ and $\overline{\overline{a}} = a$. We define $Act = A \cup \{\tau\}, At = A \cup \{t\}, Actt =$

$Act \cup \{t\}$. Let us suppose that $a, b, \ldots$ range over A, $u, v, \ldots$ range over Act, and $x, y \ldots$ range over $Actt$. We consider the signature $\Sigma = \bigcup_{n \in \{0,1,2\}} \Sigma_n$, where

$$\begin{aligned}
\Sigma_0 &= \{Nil\} \\
\Sigma_1 &= \{x. \mid x \in A \cup \{t\}\} \\
&\quad \cup \{[S] \mid S \text{ is a function form } A \text{ to } A\} \\
&\quad \cup \{\backslash M \mid M \subseteq A\} \\
\Sigma_2 &= \{|, +\}
\end{aligned}$$

we will write unary action operators in prefix form, the unary operators $[S], \backslash M$ in postfix form, and the rest of operators in infix form. For S, $S : A \to A$ are such that $\overline{S(a)} = S(\bar{a})$.

TPA terms over the signature Σ are defined as follows:

$$P \ ::= \ X \ \mid \ op(P_1, P_2, \ldots P_n) \ \mid \ \mu X P$$

where $X \in Var$, Var is a set of process variables, $P, P_1, \ldots P_n$ are TPA terms, $\mu X-$ is the binding construct, $op \in \Sigma$.

TPA terms without t action are CCS terms. We will use an usual definition of opened and closed terms. Closed terms which are t-guarded (each occurrence of X is within some subterm $t.A$) will be called processes.

A structural operational semantics of terms will be defined by labeled transition systems. The set of states is given by the set of terms, and actions represent labels. The transition relation $\to$ connects terms. We write $P \xrightarrow{x} P'$ instead of $(P, x, P') \in \to$ and $P \overset{x}{\not\to}$ if there is no P' such that $P \xrightarrow{x} P'$. By $P \xrightarrow{x} P'$ we indicate that the term P can perform action x to become P', $P \xrightarrow{x}$ means that P can perform the action x. The transition relation $\to$ is defined as for CCS with these new inference rules:

$$\frac{}{Nil \xrightarrow{t} Nil} \quad N$$

$$\frac{}{u.P \xrightarrow{t} u.P} \quad I$$

$$\frac{P \xrightarrow{t} P', Q \xrightarrow{t} Q', P \mid Q \overset{\tau}{\not\to}}{P \mid Q \xrightarrow{t} P' \mid Q'} \quad Par$$

$$\frac{P \xrightarrow{t} P', Q \xrightarrow{t} Q'}{P + Q \xrightarrow{t} P' + Q'} \quad ND$$

Rules N, I allow arbitrary idling for Nil process and for processes prefixed by an action from the set A. Processes cannot idle if there is a possibility of an internal communication (Par). Time determinancy is given by the last rule (ND).

In general, rules with negative premises could cause some problems (for details see [7]). But since in TPA derivations of τ actions are independent of derivations of t action, our system lacks such problems.

We write $P \xrightarrow{s}$ instead of $P \xrightarrow{x_1}\xrightarrow{x_2} \ldots \xrightarrow{x_n}$ where $s = x_1.x_2.\ldots.x_n, x_i \in Actt$. Sequence of actions s will be called a trace of P. The empty sequence of actions will be be indicated by ϵ. Let $Succ(P) = \{P' | P \xrightarrow{s} P', s \in Actt^*\}$. i.e. $Succ(P)$ is the set of all successors of P If the set of all successors of a process is finite we say that this process is finite state process.

To define transitions which hide τ action, a relation $\overset{x}{\Rightarrow}$ was defined for CCS ([19]). We use this idea to define transitions ($\overset{x}{\Rightarrow}_M$) which hide actions from $M, M \subseteq Actt$. We will write $P \overset{x}{\Rightarrow}_M P'$ iff $P \xrightarrow{s_1}\xrightarrow{x}\xrightarrow{s_2} P'$ for $s_1, s_2 \in M^\star$ and similarly for $P \overset{s}{\Rightarrow}_M$. We will write $P \overset{x}{\Rightarrow}_M$ iff $P \overset{x}{\Rightarrow}_M P'$ for some P'. If $x \in M$ then $P \overset{\widehat{x}}{\Rightarrow}_M P'$ denotes the same as $P \overset{\epsilon}{\Rightarrow}_M P'$. The relation $\overset{x}{\Rightarrow}_M$ will be used in our definitions of security properties only for actions (or sequence of actions) not belonging to M. We will extend of $\overset{s}{\Rightarrow}_M$ for sequences of actions s similarly to $\xrightarrow{s}$. Let $Tr_w(P) = \{s \in (A \cup \{t\})^\star | \exists P'.P \overset{s}{\Rightarrow}_{\{\tau\}} P'\}$. We say that $Tr_w(P)$ is the set of weak timed traces of process P Two processes are weakly timed trace equivalent ($P \simeq_w Q$) iff $Tr_w(P) = Tr_w(Q)$. Now we are ready to define M-bisimulation which is an extension of bisimulation and which ignores actions from the set M.

Definition 1. *A relation $\Re \subseteq TPA \times TPA$ is called a* M-bisimulation *if it is symmetric and the following holds: if $(P, Q) \in \Re$ and $P \xrightarrow{x} P', x \in Actt$ then there exists a process Q' such that $Q \overset{\widehat{x}}{\Rightarrow}_M Q'$ and $(P', Q') \in \Re$. Two processes P, Q are called to be M-bisimilar, abbreviated $P \approx_M Q$, if there exists a M-bisimulation relating P and Q.*

Note that $\approx_{\{\tau\}}$, i.e. in the case that $M = \{\tau\}$ corresponds to weak bisimulation. Standard CCS bisimulation will be denoted by $\sim$ (see [19]).

3 Opacity

To motivate security concepts introduced later we start this section with some examples of security properties. At the beginning we mention property called Strong Nondeterministic Non-Interference (SNNI, for short, see [3]). Let us assume that actions are divided into two groups, namely low level actions (called public) L and high level actions (called private) H i.e. $L \cup H = A$. SNNI property is based on an absence of information flow between low and high level actions. Process P has SNNI property whenever an observer cannot learn whether a high level action was performed if only low level actions can be observed. This means that process has SNNI property (denoted by $P \in SNNI$) if $P \setminus H$ behaves like P for which all high level actions are hidden (i.e. replaced by action τ) for a possible intruder observing P. The hiding is expressed by binary operator $P/M, M \subseteq A$, for which it holds if $P \xrightarrow{a} P'$ then $P/M \xrightarrow{a} P'/M$ whenever $a \notin M \cup \bar{M}$ and

$P/M \xrightarrow{\tau} P'/M$ whenever $a \in M \cup \bar{M}$. Formally SNNI property can be defined as follows.

Definition 2. *We say that process P has Strong Nondeterministic Non-Interference property iff $P \setminus H \simeq_w P/H$.*

Strong Nondeterministic Non-Interference property belongs to a group of so called language based properties. These properties are focused on traces of actions instead of system's states. Hence such approach is not appropriate for systems which can have sensitive, say secure, states and an intruder tries to learn whether such state is reached. In such cases state based security properties are more appropriate. They do not assume division of actions into the high and low level ones but a more general concept of observations and predicates are exploited.

Let us have a predicate over system's states. This could be capability to idle, deadlock, capability to execute only traces with a given time length, capability to perform at the same time actions form a given set, incapacity to idle (to perform t action) etc. Suppose that an intruder tries to learn whether a state for which the predicate holds has been reached. We do not assume any restrictions for predicates except that they are consistent with some behavioral equivalence. The formal definition follows.

Definition 3. *Predicate ϕ over processes is consistent with respect to relation $\cong$ if $P \cong P'$ implies that $\phi(P) \Leftrightarrow \phi(P')$.*

Behavioral equivalence $\cong$ could be bisimulation ($\sim$), weak bisimulation ($\approx_{\{\tau\}}$) etc. We can also define predicates by means $\cong$ (denoted as $\phi^Q_{\cong}$) Let us assume process Q and equivalence relation $\cong$. We define that $\phi^Q_{\cong}(P)$ holds iff $P \cong Q$.

Now we will assume intruders which can observe only some processes activities i.e. only some actions. Hence we suppose that there is a set of observable actions which can be observed and a set of hidden i.e. non-observable actions (what does not mean classified). We model such observations by relations $\overset{s}{\Rightarrow}_M$.

3.1 Infinite Step Opacity

Now we will define security property called *infinite step opacity* for TPA. It is based on a similar concept defined for discrete event structures (see [14]) but some modifications are needed. Since many protocols could be described by means of timed process algebra formalism we obtain the security property with some practical value added. Moreover, to verify this property we could exploit many software tools and techniques for process algebra formal verification.

Definition 4 (Infinite Step Opacity). *Let us assume TPA process P, a predicate ϕ over processes is infinite step opaque w.r.t. the set M if whenever $P \overset{x_1}{\Rightarrow}_M P_1 \overset{x_2}{\Rightarrow}_M P_2 \ldots \overset{x_n}{\Rightarrow}_M P_n$ for $x_i \in Actt \setminus M$, $1 \le i \le n$ and $\phi(P_i)$ holds then there exist also processes P'_i, such that $P \overset{x_1}{\Rightarrow}_M P'_1 \overset{x_2}{\Rightarrow}_M P'_2 \ldots \overset{x_n}{\Rightarrow}_M P'_n$ and $\phi(P'_i)$ does not hold. By ISO^{ϕ}_M we will denote the set of processes for which the predicate ϕ is infinite step opaque w.r.t. to the M.*

Infinite step opacity is depicted on Fig. 1. Note that contrary to Strong Nondeterministic Non-Interference property which express that an execution of secrete action can be detected by an intruder here we express that an intruder cannot detect a presence of a state satisfying ϕ during a particular step of computation.

$$\begin{array}{c} P \xRightarrow{x_1}_M \ldots \xRightarrow{x_i}_M P_i \xRightarrow{x_{i+1}}_M \quad \ldots P_n \\ \phi(P_i) \\ \\ \neg\phi(P_i') \\ P \xRightarrow{x_1}_M \ldots \xRightarrow{x_i}_M P_i' \xRightarrow{x_{i+1}}_M \quad \ldots P_n' \end{array}$$

Fig. 1. Infinite Step Opacity

Example 1. Let us assume a system P which could be a communication protocol, description of an interface mechanism, power supply management, memory management etc. Suppose that the system could enter into a sensitive phase i.e. into a state, from which an intruder can obtain some sensitive information about this system, for example, previous communications, activities, stored private values etc. To do so the intruder usually cannot try all states because doing so she or he could be easily detected or it is just costly etc.

Many attacks use information about cash memory usage. Recently this techniques was used by attacks called Meltdown and Spectre. They exploit critical vulnerabilities in practically all today's processors which allow to obtain secrets stored in the memory of other running programs. This could be passwords stored in a password manager or browser, personal photos, emails, instant messages and even business-critical files. (see [18, 16]). A difference in time which is needed to load some value from the cash memory or from the main memory could help an attacker to obtain information whether that value was previously used or not and consequently he or she could exploit that information.

For simplicity, let us assume property ϕ of a (sub)system which says, that some private action h could be performed within two time units. Formally, $\phi(Q)$ holds iff $Q \xrightarrow{h}$ or $Q \xrightarrow{t.h}$ or $Q \xrightarrow{t.t.h}$. Attacker cannot see private actions from the set M i.e. she or he could see only actions from the set $Act \setminus M$ and tries to discover, whether the system has reached a state which satisfy property ϕ. The system is infinite step opaque, i.e. $P \in ISO_M^{\phi}$ if the attacker cannot learn whether system is or was in the such sensitive state.

3.2 Properties

In this subsection we present same basic properties of infinite step opacity. We start with two "inclusion" properties reflecting power of predicate ϕ as well as capability of an observer to see actions, what is expressed by a set M.

Proposition 1. *Let $\phi_1 \Rightarrow \phi_2$. Then $ISO_M^{\phi_2} \subseteq ISO_M^{\phi_1}$.*

Proof. Let $P \in ISO_M^{\phi_2}, P \overset{x_1}{\Rightarrow}_M P_1 \overset{x_2}{\Rightarrow}_M P_2 \ldots \overset{x_n}{\Rightarrow}_M P_n$ for $x_i \in (Actt \setminus M)$ and $\phi_1(P_i)$ holds. Since $\phi_1 \Rightarrow \phi_2$ and $P \in ISO_M^{\phi_2}$ then we know that there exist processes P'_i, $1 \leq i \leq n$ such that $P \overset{x_1}{\Rightarrow}_M P'_1 \overset{x_2}{\Rightarrow}_M P'_2 \ldots \overset{x_n}{\Rightarrow}_M P'_n$ and $\neg\phi_2(P'_i)$ holds. Again since $\neg\phi_2 \Rightarrow \neg\phi_1$ we have $P \in ISO_M^{\phi_1}$.

Proposition 2. *Let $M_1 \subseteq M_2$. Then $ISO_{M_1}^{\phi} \subseteq ISO_{M_2}^{\phi}$.*

Proof. Let $P \in ISO_{M_1}^{\phi}$ and $P \overset{x_1}{\Rightarrow}_{M_2} P_1 \overset{x_2}{\Rightarrow}_{M_2} P_2 \ldots \overset{x_n}{\Rightarrow}_{M_2} P_n$ for $x_i \in (Actt \setminus M_2)$ and $\phi(P_i)$ holds. Since $M_1 \subseteq M_2$ it hold also $x_i \in (Actt \setminus M_1)$ and since $P \in ISO_{M_1}^{\phi}$ then we know that there exist processes P'_i, $1 \leq i \leq n$ such that $P \overset{x_1}{\Rightarrow}_M P'_1 \overset{x_2}{\Rightarrow}_M P'_2 \ldots \overset{x_n}{\Rightarrow}_M P'_n$ and $\neg\phi(P'_i)$ holds. This means that $P \in ISO_{M_2}^{\phi}$.

An equivalence of systems is the fundamental concept in the process algebra theory. The following propositions guarantees that under some conditions infinite state opacity could be extended to all processes which are equivalent (in the sensense of bisimulation) to a process which is already infinity state opaque.

Proposition 3. *Let us assume the predicate ϕ which is consistent with respect to relation $\cong$ such that $\sim \subseteq \cong$. Let $P \in ISO_M^{\phi}$ and $P \sim Q$. Then also $Q \in ISO_M^{\phi}$.*

Proof. $Q \overset{x_1}{\Rightarrow}_M Q_1 \overset{x_2}{\Rightarrow}_M Q_2 \ldots \overset{x_n}{\Rightarrow}_M Q_n$ for $x_i \in (Actt \setminus M)$ and $\phi(Q_i)$ holds. Since $P \sim Q$ process P can perform exactly the same sequence of actions and for all processes we have $Q_i \sim P_i$ and since according the assumption $\sim \subseteq \cong$ holds and ϕ is consistent with respect to relation $\cong$ we have that $\phi(P_i)$ holds. Now since $P \in ISO_M^{\phi}$ we know that there exist P'_i, $1 \leq i \leq n$ for which $P \overset{x_1}{\Rightarrow}_M P'_1 \overset{x_2}{\Rightarrow}_M P'_2 \ldots \overset{x_n}{\Rightarrow}_M P'_n$ and $\neg\phi(P'_i)$ holds. Again, we can repeat the previous arguments to show that there exist processes Q'_i, $1 \leq i \leq n$ for which $Q \overset{x_1}{\Rightarrow}_M Q'_1 \overset{x_2}{\Rightarrow}_M Q'_2 \ldots \overset{x_n}{\Rightarrow}_M Q'_n$ and $\neg\phi(Q'_i)$ holds. Hence $Q \in ISO_M^{\phi}$.

3.3 Relation to Process Opacity

In this subsection we present a relation between infinite step opacity and process opacity. The formal definition of the later one (see [10]) follows.

Definition 5. *Let us assume process P, a predicate ϕ over processes is process opaque w.r.t. the set M if whenever $P \overset{s}{\Rightarrow}_M P'$ for $s \in (Actt \setminus M)^*$ and $\phi(P')$ holds then there exists P'' such that $P \overset{s}{\Rightarrow}_M P''$ and $\phi(P'')$ does not hold. By POp_M^{ϕ} we will denote the set of processes for which the predicate ϕ is process opaque w.r.t. to the M.*

Example 2. Let the set M contains only actions h_1 and h_2 and let the predicate ϕ says that both actions h_1 and h_2 cannot be performed. Then $P \in POp_M^{\phi}$ but $P' \notin POp_M^{\phi}$ where $P = l.Nil + l.h_1.Nil + l.h_2.Nil + l.(h_1.Nil + h_2.Nil)$ and $P' = l.Nil + l.h_1.Nil + l.h_2.Nil$. When action l is performed this process always reaches a state which satisfies ϕ and hence it cannot be considered to be safe with respect to predicate ϕ and set M. On the other side these two processes cannot be distinguished by any opacity property since the property ϕ is not a trace property (as it is property SNNI).

$$P \quad \overset{s}{\Longrightarrow}_M \quad \phi(P')$$

$$P \quad \overset{s}{\Longrightarrow}_M \quad \neg\phi(P'')$$

Fig. 2. Process opacity

Process opacity is depicted on Fig. 2.

Let ϕ is consistent with respect to $\cong$ and $\cong$ is such that it a subset of the trace equivalence (see [19]). Then we have if $P \cong P'$ then $P \in POp^{\phi}_{M} \Leftrightarrow P' \in POp^{\phi}_{M}$.

The relation between process opacity and infinite step opacity is expressed by the following proposition.

Proposition 4. *For every M and ϕ it holds $ISO^{\phi}_{M} \subseteq PO^{\phi}_{M}$. Moreover there exist M and ϕ such that $ISO^{\phi}_{M} \subset PO^{\phi}_{M}$.*

Proof. Sketch. It can be easily proved that the first part of the proposition directly follows from Definitions 4 and 5. What distinguish infinite step opacity from process opacity is a stronger requirement of the former that visible computational trace s has to be emulated even when it does not ends in a state satisfying ϕ.

On the other side, we can obtain a stronger security property than infinite step opacity by requiring both process opacity for ϕ and its negation. The corresponding proposition is the following.

Proposition 5. *For every M and total ϕ it holds $PO^{\phi}_{M} \cap PO^{\neg\phi}_{M} \subseteq ISO^{\phi}_{M}$. Moreover there exist M and ϕ such that $PO^{\phi}_{M} \cap PO^{\neg\phi}_{M} \subset ISO^{\phi}_{M}$.*

Proof. The main idea. Since ϕ is total any step reached be a process satisfy either ϕ or $\neg\phi$. The rest could be directly easily proved directly from Definitions 4 and 5.

As a consequence of Propositions 4 and 5 we obtain the following corollary.

Corollary 1. *For every M and total ϕ it holds $PO^{\phi}_{M} \cap PO^{\neg\phi}_{M} = ISO^{\phi}_{M} \cap ISO^{\neg\phi}_{M}$.*

3.4 Timing Attacks vs Non-timing Attacks

For attackers who can observe systems in real time, time attacks represent powerful tools to compromise such systems. On the other side these techniques are useless for off-line systems and hence these could be considered safe with respect timing attacks, i.e. with respect to attackers who cannot observe elapsing of (real) time. By the presented formalism we have a way how to distinguish these two cases.

Definition 6 (Immunity with respect to Timinig Attacks). *Process P is safe with respect to timing attack, ϕ and $M, t \notin M$ iff $P \notin ISO_M^{\phi}$ but $P \in ISO_{\{M \cup t\}}^{\phi}$.*

In case that an attacker can observe time behavior of systems but only with a limited accuracy we can model that behavior with a larger time granularity (bigger time units) or we could change Definition 4 in such a way that we would allow that actions x_i to be from $(Actt \setminus M) \cup \{t\}$.

3.5 Complexity

Unfortunately, infinite step opacity is undecidable in general. This fact is implied by undecidability of process opacity (see [10]) and Propositions 4 and 5.

Now we briefly sketch how to overcome this problem. First, to obtain decidability of infinite step opacity we need to restrict predicates ϕ. We will model predicates by special processes called tests. From now on, we will assume that action τ is not visible for an attacker, that is $\tau \in M$. Process called test of $'phi$ (denoted by T_ϕ) will communicates with processes P and produce a special action $\surd$ (indicating passing of the test) if the predicate ϕ holds for the processes P.

Definition 7. *We say that process T_ϕ tests predicate ϕ if $\phi(P)$ holds iff $(P|T_\phi) \setminus At \approx_t \surd.Nil$.*

We say that ϕ is the finitely definable predicate if T_ϕ is the finite state process. By means of test T_ϕ infinite step opacity could be expressed as M-bisimulation based property similarly to process opacity (see [10]). Hence due to the above sketched construction, we can obtain a decidable variant of infinite step opacity. We limit tests to be finite states processes but this limitation has no practical importance since majority of (if not all) practically important and useful properties (predicates) can be described by such tests.

Proposition 6. *Let both predicates ϕ and $\neg\phi$ are finitely definable. In this case infinite step opacity $ISOp_M^{\phi}$ is decidable in time $O((|A|)^3.k.l)$ for finite state processes, where k and l are numbers of states of P and the maximum of numbers of states of processes representing tests of predicates ϕ and $\neg\phi$, respectively.*

Proof. The main idea. We need to show that the relation $\approx_{M \cup \{t\}}$ can be decided in time $O((|A|)^3.k.l)$. For this we need a slight modification of the proof of complexity results for weak bisimulation (see [15]). For testing scenario see Fig. 3.

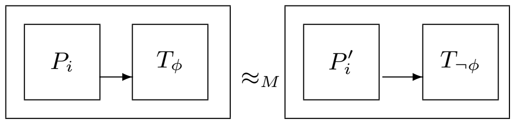

Fig. 3. Testing scenario

This methods allows us to reduce verification of security given in terms of infinite step opacity to checking bisimulation by some of existing techniques and software tools. Moreover, protocols for IoT could be defined by finite state systems as well as security properties expressed by ϕ and $\neg\phi$, hence Proposition 6 is fully applicable.

3.6 Enforcement

In case that a given process is not secure with respect to infinite step opacity for predicate ϕ and set M i.e. $P \notin ISO_M^\phi$ we can either modify its behavior. But this could be be costly, difficult or even impossible, in the case that it is already part of a hardware solution, proprietary firmware etc. Or we can use supervisory control (see [21]). This method restricts system's activities in such a way that the resulting system becomes secure with respect to infinite step opacity. A supervisor can see (some) set of process's actions and can control (disable or enable) some set of process's action. In such a way supervisor restricts resulting behaviour to guarantee system's security. This represents a trade-off between functionality and security. But in many cases it does not cause a real problem.

Let us assume some communication protocol which could reach (with some very low probability) a state which cannot be considered to be secure with respect to infinite step opacity. In such case the transmission of a packet is interrupted by supervisor and it should start from the begging. What has no influence on basic functionality of the protocol but the supervisor increases its security.

Sometimes this restriction could have even smaller influence on process's behavior. Let us assume that the system can perform action a and b in an arbitrary order but only a sequence $b.a$ could leak some classified information i.e. lead to some sensitive intermediate states. If the supervisor restricts such sequence, system becomes secure but this has no significant influence on overall system's functionality. In [11] we have proposed supervisory control for process opacity. This technique could be extended also for infinite state opacity as well.

4 Discussion and further work

We have presented application of formal methods in security, namely we have introduced the security property called infinite step opacity. We have formalized this property in the framework of timed process algebra called TPA. This security property is very general and can cover some already defined security properties.

Moreover, by suitable choice of process's predicates we obtain property which can be effectively checked.

One of the main features that makes it so valuable in systems for IoT devices is the fact that we can model process's security with respect to a limited number of attempts to perform an attack, with respect to limited time of attack, to limited time precision of an attacker and so on.

Moreover, we can use also different process algebras than TPA. We can enrich TPA by operators expressing such "parameters" as probability, distribution, networking architecture, network capacity or throughput, power consumption and so on. In such a way also other types of attacks, which exploit information flow through various covert channels, could be described. All these is particularly challenging for systems consisting of IoT devices.

Due to complexity issues, plan to define and examine compositional properties of infinite step opacity in such a way that bottom-up design of secure processes would be easier. We also consider to work with value-passing process algebra to simplify some notations and also to express new security features. In this way we could obtain new formalism(s) capable to express other aspects of behavior of low level protocols used in the field of IoT.

For the time being, we are testing the usefulness of this algebra in our own systems deployed in our laboratory: a Wireless Sensor and Actuator Networks using BLE and TCP/IP protocols, but we intend to study real systems in a short time.

References

1. Dhem, J.F., Koeune, F., Leroux, P.A., Mestré, P., Quisquater, J.J., Willems, J.L.: A practical implementation of the timing attack. In: Quisquater, J.J., Schneier, B. (eds.) Smart Card Research and Applications. pp. 167–182. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
2. Felten, E.W., Schneider, M.A.: Timing attacks on web privacy. In: Proceedings of the 7th ACM Conference on Computer and Communications Security. pp. 25–32. CCS '00, ACM, New York, NY, USA (2000). https://doi.org/10.1145/352600.352606, http://doi.acm.org/10.1145/352600.352606
3. Focardi, R., Gorrieri, R., Martinelli, F.: Information flow analysis in a discrete-time process algebra. In: Proceedings 13th IEEE Computer Security Foundations Workshop. CSFW-13. pp. 170–184 (July 2000)
4. Garrido-Hidalgo, C., Hortelano, D., Roda-Sanchez, L., Olivares, T., Ruiz, M.C., Lopez, V.: Iot heterogeneous mesh network deployment for human-in-the-loop challenges towards a social and sustainable industry 4.0. IEEE Access **PP**, 1–1 (05 2018)
5. Goguen, J.A., Meseguer, J.: Security policies and security models. In: 1982 IEEE Symposium on Security and Privacy. pp. 11–11 (April 1982)
6. Gorrieri, R., Martinelli, F.: A simple framework for real-time cryptographic protocol analysis with compositional proof rules. Science of Computer Programming **50**(1), 23 – 49 (2004), 12th European Symposium on Programming (ESOP 2003)
7. Groote, J.F.: Transition system specifications with negative premises. Theoretical Computer Science **118**(2), 263 – 299 (1993)
8. Gruska, D.P.: Observation based system security. Fundam. Inf. **79**(3-4), 335–346 (Aug 2007)
9. Gruska, D.P.: Informational analysis of security and integrity. Fundam. Inf. **120**(3-4), 295–309 (Jul 2012)
10. Gruska, D.P.: Process opacity for timed process algebra. In: Voronkov, A., Virbitskaite, I. (eds.) Perspectives of System Informatics. pp. 151–160. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
11. Gruska, D., Ruiz, M.: Initial process security. In: Proceedings of 26th International Workshop on Concurrency, Specification and Programming. CS&P'17 (2017)
12. Handschuh, H., Heys, H.M.: A timing attack on rc5. In: Proceedings of the Selected Areas in Cryptography. pp. 306–318. SAC '98, Springer-Verlag, Berlin, Heidelberg (1999)
13. Hortelano, D., Olivares, T., Ruiz, M.C., Garrido-Hidalgo, C., Lpez, V.: From sensor networks to internet of things. bluetooth low energy, a standard for this evolution. Sensors **17**, 372 (02 2017)
14. Jacob, R., Lesage, J.J., Faure, J.M.: Overview of discrete event systems opacity: Models, validation, and quantification. Annual Reviews in Control **41**, 135 – 146 (2016)
15. Kanellakis, P.C., Smolka, S.A.: Ccs expressions, finite state processes, and three problems of equivalence. Information and Computation **86**(1), 43 – 68 (1990)
16. Kocher, P., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., Yarom, Y.: Spectre attacks: Exploiting speculative execution. CoRR **abs/1801.01203** (2018)

17. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology. pp. 104–113. CRYPTO '96, Springer-Verlag (1996)
18. Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., Kocher, P., Genkin, D., Yarom, Y., Hamburg, M.: Meltdown. CoRR **abs/1801.01207** (2018)
19. Milner, R.: Communication and Concurrency. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1989)
20. Nassi, B., Sror, M., Lavi, I., Meidan, Y., Shabtai, A., Elovici, Y.: Piping botnet - turning green technology into a water disaster. CoRR **abs/1808.02131** (2018), http://arxiv.org/abs/1808.02131
21. Ramadge, P.J.G., Wonham, W.M.: The control of discrete event systems. Proceedings of the IEEE **77**(1), 81–98 (Jan 1989)
22. Roda-Sanchez, L., Garrido-Hidalgo, C., Hortelano, D., Olivares, T., Ruiz, M.C.: Operable: An iot-based wearable to improve efficiency and smart worker care services in industry 4.0. Journal of Sensors **2018**, 6272793:1–6272793:12 (2018)
23. Song, D.X., Wagner, D., Tian, X.: Timing analysis of keystrokes and timing attacks on ssh. In: Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10. SSYM'01, USENIX Association (2001)