Detecting Man-in-the-Middle Attacks by Precise Timing

Benjamin Aziz
e-Science Centre
STFC Rutherford Appleton Laboratory
Didcot, United Kingdom
Email: benjamin.aziz@stfc.ac.uk

Geoff Hamilton
School of Computing
Dublin City University
Dublin, Ireland
Email: Geoff.Hamilton@computing.dcu.ie

Abstract

Man-in-the-middle attacks are one of the most popular and fundamental attacks on distributed systems that have evolved with advances in distributed computing technologies and have assumed several shapes ranging from simple IP spoofing to complicated attacks on wireless communications, which have safety-critical applications such as remote wireless passport verification. This paper proposes a static analysis algorithm for the detection of man-in-the-middle attacks in mobile processes using a solution based on precise timing.

1. Introduction

Man-in-the-Middle (MiM) attacks are one of the most popular and challenging threats in computing systems and there is a large body of research dedicated to the detection and analysis of different forms of these attacks (examples include [1], [14], [17], [20], [24]). A MiM attack is defined as an attack in which the intruder is able to read and write messages communicated between two parties without either party being conscious of this fact. The attack appears in many shapes and forms and the sophistication of these forms has evolved with the evolution of modern computing systems. In 2004, Citibank's Citibusiness service was the victim of a *phishing* 1.0 attack (a form of MiM attacks targeted for Web users) in which a fake web page constructed to resemble the original service's page was used to trick the users into believing they are communicating with the authentic service, but in reality, compromising their account details. In safety-critical systems, the problem of MiMs becomes even more urgent.

In this paper, we are interested in analysing the ability of mobile systems, such as wireless sensor networks, in detecting MiM attacks in a timely fashion. Wireless sensor networks rely on a class of protocols known as *distance-bounding* protocols [6], [11], [16], [22] for estimating entity distances. In these protocols, the distance between two entities, measured using precise timing, is required in defining some of the security properties of the protocol. For example, the *authenticated* entity needs to be at some distance (not more) from the *authenticating* entity. This class of protocols is susceptible to a whole new class of attacks [8], [12], [23] including MiM attacks [10], for which some solutions have been proposed in [13], [19], [21]. An article in recent years by Bruce Schneier (`http://www.schneier.com/blog/archives/2006/04/rfid_cards_and.html`) highlights MiM threats on RFID-enabled passports in cross-border immigration controls and suggests precise timing mechanisms as a means of detecting such attacks.

Distance-bounding protocols have the requirement that a message arrives at its destination in a timely fashion. For example, consider the following scenario:

$$@T_1 : \mathit{Alice} \xrightarrow{m_1} @T_2 : \mathit{Carol}$$

$$@T_3 : \mathit{Carol} \xrightarrow{m_2} @T_4 : \mathit{Bob}$$

in which *Alice* sends a message m_1 to *Bob* at time T_1. The message is intercepted by *Carol* at time T_2, altered to m_2 and then forwarded to *Bob* at time T_3. Finally, *Bob* receives m_2 at time T_4. If Bob is time-sensitive, he would have two parameters: the first is T_{exp}; the time at which he expects to receive the message and the second is *Diff*; the maximum difference he is willing to tolerate between the actual time of receipt and T_{exp}. Therefore, in the above scenario, one would expect *Bob* to time-out and reject m_2 as inauthentic if $|T_4 - T_{exp}| > \mathit{Diff}$. Otherwise, if Carol succeeds in modifying m_1 in a timely fashion, i.e. such that $|T_4 - T_{exp}| \leq \mathit{Diff}$, then *Bob* is likely to accept the message as authentic (if all other non-time related criteria are satisfied).

Our approach in tackling this problem is formal; it is based on designing a static analysis for capturing name substitutions in a version of the π-calculus [18] extended with the notion of *timers* [5]. This approach follows from earlier, well-established, works on security analyses for mobile systems and cryptographic systems (see [3], [4]). The results of the analysis are used to define a name integrity property, which itself forms the basis for defining a MiM attack property. We demonstrate the applicability of the analysis in a simple example of a distance bounding protocol.

The rest of the paper is structured as follows. In Section 2, we give an overview of a timed process algebra and its operational semantics. In Section 2.1, we define a non-standard semantics for the language and show its soundness with respect to the operational semantics. An approximation for the semantics is defined in Section 3, over which a definition of the MiM property is given in Section 4. Finally, we show the applicability of the analysis in Section 5 and conclude the paper in Section 6.

2. TPi: A Process Algebra with Timers

The process algebra we use throughout the paper, called *TPi*, is defined according to the following syntax of processes, $P, Q \in \mathcal{P}$, inspired from the calculus of [5]:

$$P, Q ::= \\ \overline{x}\langle y\rangle.P \mid \mathtt{timer}^t(x(y).P, Q) \mid \ P \mid Q \ \mid \ !P \ \mid (\nu x)P \ \mid \mathbf{0}$$

The syntax corresponds to that of the standard synchronous π-calculus except for the fact that input actions are placed within a timer, $\mathtt{timer}^t(x(y).P, Q)$, where $t \in \mathbb{N}$ represents time. The input action, $x(y).P$, can synchronise with suitable output actions as long as $t > 0$. Otherwise, when $t = 0$, the timer behaves as Q. Names constitute the set $\mathcal{N}$.

The structural operational semantics of *TPi* are given in terms of the structural congruence, $\equiv$, and labelled transition, $\xrightarrow{\mu}$, relations as shown in Figure 1. The definition of $\equiv$ is standard, except for rule (6), which deals with expired timers. The labels, $\mu \in \{\xrightarrow{\overline{x}\langle y\rangle}, \xrightarrow{\overline{x}(y)}, \xrightarrow{x(z)}, \xrightarrow{\tau}\}$, express free and bound outputs, inputs and silent actions, respectively. Again, most of the rules for $\xrightarrow{\mu}$ are straightforward and their explanation can be found elsewhere [2, §3.2.2] except for rule (14), where a *time-stepping* function, $\eth : \mathcal{P} \rightarrow \mathcal{P}$, expresses the ticking of activated timers:

$$\eth(P) = \begin{cases} \mathtt{timer}^t(x(y).P, Q), & \\ \qquad\quad \text{if } P = \mathtt{timer}^{t+1}(x(y).P, Q) & \\ \eth(Q) \mid \eth(R), & \text{if } P = Q \mid R \\ (\nu x)\eth(Q), & \text{if } P = (\nu x)Q \\ P, & \text{otherwise} \end{cases}$$

2.1. A Name-Substitution Semantics

In this section, we define a non-standard semantics for *TPi* such that it is possible to express the meaning of processes in terms of name substitutions resulting from message passing (note here that we exclude other substitutions, such as those due to α-conversions or renaming of bound names). For example, in:

$$!((\nu y)\overline{x}\langle y\rangle.\mathbf{0}) \mid \ !\mathtt{timer}^{t+1}(x(u).\mathbf{0}, \mathbf{0})$$

we would like to have a meaning that captures the set of substitutions, $\{y_1/u_1, y_2/u_2 \ldots\}$, where y_i is a labelled copy of the fresh name, y, and u_i is a labelled instance of the input parameter, u, assuming that $t+1 > 0$. (Note: other labelling schemes are also possible as long as they maintain bound name uniqueness).

First however, we need to introduce the notion of *tags* defined as the set, $\ell, \ell' \in \mathcal{L}$. The set $\mathcal{L}$ is then used to tag messages of output actions: $\overline{x}\langle y\rangle.P$ becomes $\overline{x}\langle y^{\ell}\rangle.P$. This tagging is performed uniquely, i.e. no two messages will be assigned the same tag even if the two messages have the same name. This will help distinguish every message in the non-standard interpretation. Additionally, we define the following two functions involving tags:

$$\begin{array}{l} \mathit{value\_of} \ : \ \mathcal{L} \rightarrow \mathcal{N} \\ \mathit{tags\_of} \ : \ \mathcal{P} \rightarrow \wp(\mathcal{L}) \end{array}$$

where $\mathit{value\_of}(\ell) = y$ signifies that ℓ was assigned to the message y and $\mathit{tags\_of}(P) = \{\ell 1, \ldots, \ell n\}$ signifies the set of tags used in P. Naturally, *value_of* is non-injective and we sometimes write $\mathit{value\_of}(\{\ell, \ell' \ldots\})$ to mean $\{\mathit{value\_of}(\ell), \mathit{value\_of}(\ell') \ldots\}$.

Next, we define the environment, $\phi_{\mathcal{S}} : \mathcal{N} \rightarrow \wp(\mathcal{L})$, such that $\ell \in \phi_{\mathcal{S}}(x)$ implies that the message tagged with ℓ replaces the input parameter, x, at runtime. From $\phi_{\mathcal{S}}$, a semantic domain, $D_{\perp} : \mathcal{N} \rightarrow \wp(\mathcal{L})$, is formed with the following ordering:

$$\forall \phi_{\mathcal{S}1}, \phi_{\mathcal{S}2} \in D_{\perp} : \quad \phi_{\mathcal{S}1} \sqsubseteq_{D_{\perp}} \phi_{\mathcal{S}2} \quad \Leftrightarrow \quad \forall x \in \mathcal{N} : \ \phi_{\mathcal{S}1}(x) \subseteq \phi_{\mathcal{S}2}(x)$$

where the bottom element, $\perp$, denotes the null environment, $\phi_{\mathcal{S}0}$, which maps every name in $\mathcal{N}$ to $\emptyset$. From the above definition of $D_{\perp}$ then, we can assign a meaning to process P as a function $\mathcal{S}(\![P]\!)\ \rho\ \phi_{\mathcal{S}} \in D_{\perp}$, defined over the structure of P as shown in Figure 2.

In the rules of this semantics, ρ is a multiset of processes in parallel with the interpreted process along with the standard $\{\!|-|\!\} : \mathcal{P} \rightarrow \wp(\mathcal{P})$ and $\uplus : \wp(\mathcal{P}) \times \wp(\mathcal{P}) \rightarrow \wp(\mathcal{P})$ operators over ρ. The meaning of ρ is given in $(\mathcal{R}0)$ using the special union, $\cup_{\phi_{\mathcal{S}}}$, defined as:

$$\forall x \in \mathcal{N} : \ (\phi_{\mathcal{S}1} \cup_{\phi_{\mathcal{S}}} \phi_{\mathcal{S}2})(x) = \phi_{\mathcal{S}1}(x) \cup \phi_{\mathcal{S}2}(x)$$

We discuss next a few interesting rules. Communications are dealt with in rule, $(\mathcal{S}2)$, for input actions. The rule uses the equivalence of two names, $\overset{\phi_{\mathcal{S}}}{\sim}$, parameterised by $\phi_{\mathcal{S}}$ to determine matching channel names. This is defined for any two names, x and y as:

$$x \overset{\phi_{\mathcal{S}}}{\sim} y \quad \Leftrightarrow \\ (\mathit{value\_of}(\phi_{\mathcal{S}}(x)) \cap \mathit{value\_of}(\phi_{\mathcal{S}}(y)) \neq \emptyset) \ \vee \ (x = y)$$

Rules of the $\equiv$ *relation:*
(1) $(\mathcal{P}/\equiv, |, \mathbf{0})$ is a commutative monoid
(2) $(\nu x)\mathbf{0} \equiv \mathbf{0}$
(3) $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$
(4) $!P \equiv P \,|!P$
(5) $(\nu x)(P \mid Q) \equiv (P \mid (\nu x)Q)$ if $x \notin fn(Q)$
(6) $\mathtt{timer}^0(x(z).P, Q) \equiv Q$

Rules of the $\xrightarrow{\mu}$ *relation:*
(7) $\overline{x}\langle y\rangle.P \xrightarrow{\overline{x}\langle y\rangle} P$
(8) $\mathtt{timer}^{t+1}(x(z).P, Q) \xrightarrow{x(z)} P$
(9) $P \xrightarrow{\overline{x}\langle y\rangle} Q \;\;\Rightarrow\;\; (\nu y)P \xrightarrow{\overline{x}(y)} Q$ if $x \neq y$
(10) $P \xrightarrow{\overline{x}\langle y\rangle} P', Q \xrightarrow{x(z)} Q' \;\Rightarrow\; P \mid Q \xrightarrow{\tau} P' \mid Q'[y/z]$
(11) $P \xrightarrow{\overline{x}(y)} P', Q \xrightarrow{x(z)} Q' \;\Rightarrow\; P \mid Q \xrightarrow{\tau} (\nu y)(P' \mid Q'[y/z])$
(12) $P \xrightarrow{\mu} Q \;\;\Rightarrow\;\; (\nu x)P \xrightarrow{\mu} (\nu x)Q$ if $x \neq fn(\mu)$
(13) $P \xrightarrow{\mu} P' \;\;\Rightarrow\;\; P \mid Q \xrightarrow{\mu} P' \mid Q$
(14) $P \xrightarrow{\tau} \eth(P)$

Figure 1. The structural operational semantics of *TPi*.

$$
\begin{array}{lll}
(\mathcal{S}1) & \mathcal{S}(\llbracket \overline{x}\langle y^{\ell}\rangle.P \rrbracket)\ \rho\ \phi_{\mathcal{S}} & = \phi_{\mathcal{S}} \\
(\mathcal{S}2) & \mathcal{S}(\llbracket \mathtt{timer}^{t+1}(x(y).P,Q) \rrbracket)\ \rho\ \phi_{\mathcal{S}} & = \Big(\bigcup_{\phi_{\mathcal{S}}}{}_{\overline{x'}\langle z^{\ell}\rangle.P' \in \rho:\ x \overset{\phi_{\mathcal{S}}}{\sim} x'} \mathcal{R}(\llbracket (\biguplus_{R \in \rho} \{\!| \eth(R) |\!\}) \uplus \{\!| P |\!\} \uplus \{\!| P' |\!\} \rrbracket)\ update(\phi_{\mathcal{S}}, y, \ell)\Big) \cup_{\phi_{\mathcal{S}}} \\
& & \quad \mathcal{R}(\llbracket (\biguplus_{R \in \rho} \{\!| \eth(R) |\!\}) \uplus \{\!| \mathtt{timer}^{t}(x(y).P,Q) |\!\} \rrbracket)\ \phi_{\mathcal{S}} \\
(\mathcal{S}3) & \mathcal{S}(\llbracket \mathtt{timer}^{0}(x(y).P,Q) \rrbracket)\ \rho\ \phi_{\mathcal{S}} & = \mathcal{R}(\llbracket \{\!| Q |\!\} \uplus \rho \rrbracket)\ \phi_{\mathcal{S}} \\
(\mathcal{S}4) & \mathcal{S}(\llbracket P \mid Q \rrbracket)\ \rho\ \phi_{\mathcal{S}} & = \mathcal{R}(\llbracket \{\!| P |\!\} \uplus \{\!| Q |\!\} \uplus \rho \rrbracket)\ \phi_{\mathcal{S}} \\
(\mathcal{S}5) & \mathcal{S}(\llbracket !P \rrbracket)\ \rho\ \phi_{\mathcal{S}} & = snd(fix\ \mathcal{F}(0, \bot))
\end{array}
$$

where, $\mathcal{F} = \lambda f \lambda(j, \phi).f\ (if\, \phi = \mathcal{R}(\llbracket (\biguplus_{i=0}^{j} \{\!| (P)\sigma |\!\})\ \uplus\ \rho \rrbracket)\ \phi_{\mathcal{S}}\ then\ j, \phi\ else\ (j+1), (\mathcal{R}(\llbracket (\biguplus_{i=0}^{j} \{\!| (P)\sigma |\!\})\ \uplus\ \rho \rrbracket)\ \phi_{\mathcal{S}}))$
and $\sigma = [bn_i(P)/bn(P)][tags\_of_i(P)/tags\_of(P)],\ bn_i(P) = \{x_i \mid x \in bn(P)\},\ tags\_of_i(P) = \{\ell_i \mid \ell \in tags\_of(P)\}$

$$
\begin{array}{lll}
(\mathcal{S}6) & \mathcal{S}(\llbracket (\nu n)P \rrbracket)\ \rho\ \phi_{\mathcal{S}} & = \mathcal{R}(\llbracket \{\!| P |\!\} \uplus \rho \rrbracket)\ \phi_{\mathcal{S}} \\
(\mathcal{S}7) & \mathcal{S}(\llbracket \mathbf{0} \rrbracket)\ \rho\ \phi_{\mathcal{S}} & = \phi_{\mathcal{S}} \\
(\mathcal{R}0) & \mathcal{R}(\llbracket \rho \rrbracket)\ \phi_{\mathcal{S}} & = \bigcup_{\phi_{\mathcal{S}}}{}_{P \in \rho} \mathcal{S}(\llbracket P \rrbracket)\ (\rho \backslash \{\!| P |\!\})\ \phi_{\mathcal{S}}
\end{array}
$$

Figure 2. The definition of $\mathcal{S}(\llbracket P \rrbracket)\ \rho\ \phi_{\mathcal{S}}$.

For each synchronisation, the value of $\phi_{\mathcal{S}}$ is updated with the tag of the communicated message using the *update* operator defined as:

$\forall \phi_{\mathcal{S}} \in D_{\bot}, y \in \mathcal{N}, \ell \in \mathcal{L}$:
$update(\phi_{\mathcal{S}}, y, \ell) = \phi_{\mathcal{S}}[y \mapsto \phi_{\mathcal{S}}(y) \cup \{\ell\}]$

Rule $(\mathcal{S}2)$ also considers the case where no communications take place. In either case, all active timers are decremented some using the time-stepping defined in the previous section. Rule $(\mathcal{S}5)$ deals with replicated processes using a fixed-point calculation of the higher order functional, $\mathcal{F}$. The rule allows for as many copies of P to be spawned and the number of each copy is used to subscript its bound names and tags in order to maintain their uniqueness. As a result, the interpretation of restricted names in rule $(\mathcal{S}6)$ drops the ν operator in ρ.

The following soundness theorem states that name substitutions in the structural operational semantics are captured in the non-standard semantics.

Theorem[**Soundness of the non-standard semantics**]
$\forall P, Q, x, y\ :\ \ P \xrightarrow{\mu}{}^{*} Q[x/y] \ \ \Rightarrow\ \ x \in value\_of(\phi'_{\mathcal{S}}(y))$
where, $\phi'_{\mathcal{S}} = \mathcal{S}(\llbracket P \rrbracket)\ \rho\ \phi_{\mathcal{S}}$

Proof: The proof is by induction on the rules of the structural operational semantics in Figure 1. The most interesting cases are rules (10) and (11), where we need to show that if a process, P, exhibits a transition, $P \xrightarrow{\overline{x}\langle y\rangle} P'$, then this will eventually yield a process, $\overline{x}\langle y^{\ell}\rangle.P'' \in \rho$ during the non-standard interpretation. The same can be shown for $Q \xrightarrow{x(z)} Q'$ and $R \xrightarrow{\overline{x}(y)} R'$. From rule $(\mathcal{S}2)$, we can then show that $P \mid Q$ and $R \mid Q$ will capture substitutions in the $\phi_{\mathcal{S}}$ environment in each case □ □

3. An Approximated Semantics

The computation of the non-standard semantics of the previous section is not guaranteed to terminate due to the infinite size of $D_{\bot}$ as a result of the presence of replication in processes. Therefore, we need to approximate the meaning of processes by introducing the α_k approximation, which limits the number of copies of fresh names and tags that can be captured by the semantics.

Definition[**The α_k-approximation function**]
Define $\alpha_k\ :\ (\mathcal{N} \cup \mathcal{L})\ \rightarrow\ (\mathcal{N}^{\sharp} \cup \mathcal{L}^{\sharp})$ as follows, where $\mathcal{N}^{\sharp} = \mathcal{N} \backslash \{x_i \mid i > k\}$ and $\mathcal{L}^{\sharp} = \mathcal{L} \backslash \{\ell_i \mid \ell > k\}$:

$$\forall u \in (\mathcal{N} \cup \mathcal{L}) : \alpha_k(u) = \begin{cases} u_k, & \text{if } u = u_i \ \wedge \ i > k \\ u, & \text{otherwise} \end{cases}$$

And we write, $\alpha_k(\{u, u', \ldots\})$, to mean $\{\alpha_k(u), \alpha_k(u'), \ldots\}$. The α_k approximation function leads naturally to the appearance of the abstract environment, $\phi_{\mathcal{A}} : \mathcal{N}^\sharp \to \wp(\mathcal{L}^\sharp)$ and the abstract semantic domain, $D^\sharp_\perp$ with the following ordering:

$$\forall \phi_{\mathcal{A}1}, \phi_{\mathcal{A}2} \in D^\sharp_\perp : \phi_{\mathcal{A}1} \sqsubseteq_{D^\sharp_\perp} \phi_{\mathcal{A}2} \Leftrightarrow \forall x \in \mathcal{N}^\sharp : \phi_{\mathcal{A}1}(x) \subseteq \phi_{\mathcal{A}2}(x)$$

Based on $D^\sharp_\perp$, we can interpret processes as a new function, $\mathcal{A}([\![P]\!]) \ \rho \ \phi_{\mathcal{A}} \in D^\sharp_\perp$, defined as follows:

$$\mathcal{A}([\![P]\!]) \ \rho \ \phi_{\mathcal{A}} = \texttt{let } \mathit{update} = \mathit{update}^{\mathcal{A}}_{\alpha_k} \texttt{ in let } \phi_{\mathcal{S}} = \phi_{\mathcal{A}} \texttt{ in } \mathcal{S}([\![P]\!]) \ \rho \ \phi_{\mathcal{S}}$$

which uses the same algorithm for $\mathcal{S}([\![P]\!]) \ \rho \ \phi_{\mathcal{S}}$ defined in Figure 2 but replacing $\phi_{\mathcal{S}}$ and *update* with their abstract siblings. The $\mathit{update}^{\mathcal{A}}_{\alpha_k}$ operator is defined for all $\phi_{\mathcal{A}} \in D^\sharp_\perp, y \in \mathcal{N}, \ell \in \mathcal{L}$ as follows:

$$\mathit{update}^{\mathcal{A}}_{\alpha_k}(\phi_{\mathcal{A}}, y, \ell) = \phi_{\mathcal{A}}[\alpha_k(y) \mapsto \phi_{\mathcal{A}}(\alpha_k(y)) \cup \{\alpha_k(\ell)\}]$$

The following termination result can be shown to hold.

Theorem[**Termination of the Abstract Semantics**]
For any process, P, the computation of $\mathcal{A}([\![P]\!]) \ \{\!|\,|\!\} \ \perp_{D^\sharp_\perp}$ terminates.

Proof: The proof relies on two requirements: First, to show that $D^\sharp_\perp$ is finite. This is true from the definition of α_k. The second is to show that the abstract meaning of a process is monotonic with respect to the number of copies of a replicated process:

$$\mathcal{R}([\![(\biguplus_{i=0}^{j} \{\!|(P)\sigma|\!\}) \uplus \rho]\!]) \ \phi_{\mathcal{A}} \sqsubseteq_{D^\sharp_\perp} \mathcal{R}([\![(\biguplus_{i=0}^{j+1} \{\!|(P)\sigma|\!\}) \uplus \rho]\!]) \ \phi_{\mathcal{A}}$$

This latter requirement is proved by showing that the extra copy of P can "only" induce more communications □ □

4. Man-in-the-Middle Analysis

In our analysis of the MiM attacks, we refer to the usual finite lattice of security levels, $(\mathcal{S}, \sqsubseteq_{\mathcal{S}}, \sqcap_{\mathcal{S}}, \sqcup_{\mathcal{S}}, \top_{\mathcal{S}}, \perp_{\mathcal{S}})$, and based on it define $\zeta : \mathcal{N} \to \mathcal{S}$ as a mapping from names to their security levels. Now, we can define the *name integrity* property as follows.

Property[**Name integrity**]
We say that a name, x, has the *integrity* property with respect to a $\phi_{\mathcal{A}}$ environment if

$$\forall n \in \mathit{value\_of}(\phi_{\mathcal{A}}(x)) : \zeta(x) \sqsubseteq \zeta(n) \qquad \square$$

The predicate $\mathit{integrity}(x, \phi_{\mathcal{A}})$ indicates that x upholds the above property with respect to $\phi_{\mathcal{A}}$. A MiM attack is defined as an attack in which the intruder is capable of breaching the integrity of names of two processes.

Property[**Man-in-the-Middle Attack**]
A context, C (a process with a hole) succeeds in launching a MiM attack on two processes, P and Q, if the result of the abstract interpretation, $\mathcal{A}([\![C(P \mid Q)]\!]) \ \{\!|\,|\!\} \ \perp_{D^\sharp_\perp} = \phi_{\mathcal{A}}$ proves that, $\exists x \in \mathit{bn}(P), y \in \mathit{bn}(Q) : \neg(\mathit{integrity}(x, \phi_{\mathcal{A}}) \vee \mathit{integrity}(y, \phi_{\mathcal{A}}))$ □

5. Example: Distance-bounding Protocols

We discuss here the application of our analysis to a simplified model of the RFID distance-bounding protocol defined in [11]. The one-way authentication protocol consists of the following steps between a verifier, *Vr*, and a prover, *Pr*, starting at time, T_0:

$$\begin{array}{llll} @T_0 : & & \mathit{Vr} \to \mathit{Pr} : & N_{\mathit{Vr}} \\ \mathit{for}(i = 1; i \leq n; \mathit{inc}(i)) \ \{ & & & \\ & @T_i : & \mathit{Vr} \to \mathit{Pr} : & C_i \\ & @(T_i + \delta) : & \mathit{Pr} \to \mathit{Vr} : & R_i^{C_i} \quad \} \end{array}$$

where $n > 0, T_i, \delta \in \mathbb{N}$ are natural numbers such that T_i is a point in time and δ is a very short time gap (ideally $T_i + \delta < T_{i+1}$). Also, $\mathit{inc} : \mathbb{N} \to \mathbb{N}$ is the increment function, N_{Vr} is a fresh nonce and C_i, $R_i^{C_i}$ are challenge values and their corresponding responses. For the sake of brevity, we refer the reader for a full description of the protocol to [11, §3.1]. Here, we give in Figure 3 a non-cryptographic *TPi*-based specification of the protocol for the specific case of $n = 3$. The specification allows *Vr* to send a fresh nonce N_{Vr} to *Pr*. *Vr* then uses the internal channel x to simulate time waitings of T_1, T_2 and T_3 since no inputs can be performed over x and these will time-out. However, their continuations will output challenges Ci to *Pr*. *Pr* itself waits on these challenges and then replies with the expected responses Ri. During this protocol, the intruder I is capable of interfering with all communications over *c*, since it knows the name of this channel. The protocol itself is defined as the parallel composition of the three processes.

Applying the abstract interpretation, $\mathcal{A}([\![\mathit{Prot}]\!]) \ \{\!|\,|\!\} \ \perp_{D^\sharp_\perp}$, with $k = 1$, we obtain the following substitutions for $i = 1 \ldots n$:

$$Ci' \in \mathit{value\_of}(\phi_{\mathcal{A}}(ui)) \text{ and } Ri' \in \mathit{value\_of}(\phi_{\mathcal{A}}(ri))$$

Now, assuming that the intruder's challenges and responses have lower security levels than the prover's and verifier's input parameters, i.e. $\zeta(Ci') \sqsubseteq \zeta(ui)$ and $\zeta(Ri') \sqsubseteq \zeta(ri)$, then it can be seen that I achieves the MiM property

$$\begin{array}{lcl}
\mathit{Vr} & \overset{\text{def}}{=} & (\nu\, N_{\mathit{Vr}})(\nu\, x)\ \ (\overline{c}\langle N_{\mathit{Vr}}\rangle.\mathtt{timer}^{T1}(x(d1).\mathbf{0},\overline{c}\langle C1\rangle. \\
 & & \mathtt{timer}^{\delta}(c(r1).\mathtt{timer}^{T2}(x(d2).\mathbf{0},\overline{c}\langle C2\rangle. \\
 & & \mathtt{timer}^{\delta}(c(r2).\mathtt{timer}^{T3}(x(d3).\mathbf{0},\overline{c}\langle C3\rangle. \\
 & & \mathtt{timer}^{\delta}(c(r3).\mathbf{0},\mathbf{0})\ ),\mathbf{0})\ ),\mathbf{0})\ )\ ) \\
\mathit{Pr} & \overset{\text{def}}{=} & \mathtt{timer}^{\infty}(c(n).\mathtt{timer}^{\infty}(c(u1).\overline{c}\langle R1\rangle.\mathtt{timer}^{\infty}(c(u2).\overline{c}\langle R2\rangle. \\
 & & \mathtt{timer}^{\infty}(c(u3).\overline{c}\langle R3\rangle.\mathbf{0},\mathbf{0}),\mathbf{0}),\mathbf{0}),\mathbf{0}) \\
I & \overset{\text{def}}{=} & (\nu\, N_I)\ \ (\mathtt{timer}^{\infty}(c(n').\overline{c}\langle N_I\rangle. \\
 & & \mathtt{timer}^{\infty}(c(u1').\overline{c}\langle C1'\rangle.\mathtt{timer}^{\infty}(c(r1').\overline{c}\langle R1'\rangle. \\
 & & \mathtt{timer}^{\infty}(c(u2').\overline{c}\langle C2'\rangle.\mathtt{timer}^{\infty}(c(r2').\overline{c}\langle R2'\rangle. \\
 & & \mathtt{timer}^{\infty}(c(u3').\overline{c}\langle C3'\rangle.\mathtt{timer}^{\infty}(c(r3').\overline{c}\langle R3'\rangle.\mathbf{0},\mathbf{0}),\mathbf{0}),\mathbf{0}),\mathbf{0}),\mathbf{0})) \\
\mathit{Prot} & \overset{\text{def}}{=} & I \mid (\mathit{Vr} \mid \mathit{Pr})
\end{array}$$

Figure 3. The definition of the RFID protocol in *TPi*.

above with respect to ui and ri. This is due mainly to the promptness with which I sends its challenges and responses to both the prover and the verifier processes.

5.1. A Note on Modelling the Intruder

One of the benefits of modelling the intruder as any other process in the specification of the system, rather than for example hardcoding its behaviour directly into the semantics of the language, is that it is possible to capture any class of intruders ranging from the most passive (modelled as the process **0**) to the most general as envisioned by the Dolev-Yao model [9], [7].

In the previous example, we defined the process I in a manner sufficient to demonstrate the MiM attack. However, similar results could have been obtained by a more general, Dolev-Yao, intruder. This general intruder could be specified as follows, where $\prod$ denotes the parallel composition of multiple processes.:

$$I \overset{\text{def}}{=} (\nu\, i)\ (\bar{i}\langle\kappa_{init}\rangle \mid\ !\,\mathtt{timer}^{\infty}(i(\kappa).(\prod_{\forall x,y\in\kappa} \overline{x}\langle y\rangle.\bar{i}\langle\kappa\rangle \mid \prod_{\forall x\in\kappa}\mathtt{timer}^{\infty}(x(z).\bar{i}\langle\kappa\cup\{z\}\rangle,\mathbf{0}) \mid (\nu\, \mathit{net})\bar{i}\langle\kappa\cup\{\mathit{net}\}\rangle),\mathbf{0})$$

In this specification, κ denotes a set of names representing the knowledge of the intruder, (ν *net*) allows for the intruder to create fresh data at any time, and i is a channel used for the intruder's internal communications. The initial subprocess, $\bar{i}\langle\kappa_{init}\rangle$, outputs the set of names, κ_{init}, representing an instantiation of the intruder's initial knowledge (in general, $\kappa_{init} = \mathit{fn}(P)$, for the analysed process, P). The specification then allows the intruder to build its knowledge, κ, by repeatedly inputting over names in its knowledge. The inputted name is then passed as part of the new knowledge to the next instance of the intruder. The intruder can also perform output actions. These are either free output actions sending messages over channels already in κ, or bound output actions that create a copy of the name *net* and send it over the internal channel i. This allows the intruder to learn *net* without the need to output it first to external processes. The *learning* behaviour is interpreted as the standard union, $\cup$, over κ.

6. Conclusion and Future Work

We have presented in this paper a static analysis for detecting MiM attacks in real-time systems using precise timing. The analysis, designed for a stochastic process algebraic language, captures name substitutions occurring among processes as a result of their communications. The results of the analysis are then used to define a name integrity property and a notion of MiM attacks.

There are several directions for expanding this work. For example, other security properties of protocols with some notion of time could be investigated, such as the minimum/maximum speed at which authentication can be achieved in a real-time system. Also, time denotes cost, therefore, a slow protocol could be exploited by an intruder to mount a denial of resources attack [15].

7. Acknowledgements

This work is funded by the EU Framework 7 Project FP7-214859 CONSEQUENCE (Context-aware Data-centric Information Sharing) and the EU Framework 7 Integrated Project FP7-216917 MASTER (Managing Assurance, Security and Trust for sERvices).

References

[1] N. Asokan, Valtteri Niemi, and Kaisa Nyberg. Man-in-the-Middle in Tunnelled Authentication Protocols. In Bruce Christianson, Bruno Crispo, James A. Malcolm, and Michael Roe, editors, *Security Protocols Workshop*, volume 3364 of *Lecture Notes in Computer Science*, pages 28–41. Springer, 2003.

[2] B. Aziz. *A Static Analysis Framework for Security Properties in Mobile and Cryptographic Systems*. PhD thesis, School of Computing, Dublin City University, Dublin, Ireland, 2003.

[3] Benjamin Aziz and Geoff Hamilton. A privacy analysis for the π-calculus: The denotational approach. In *Proceedings of the* 2^{nd} *Workshop on the Specification, Analysis and Validation for Emerging Technologies*, number 94 in Datalogiske Skrifter, Copenhagen, Denmark, July 2002. Roskilde University.

[4] Benjamin Aziz, Geoff Hamilton, and David Gray. A static analysis of cryptographic processes: The denotational approach. *Journal of Logic and Algebraic Programming*, 64(2):285–320, August 2005.

[5] Martin Berger and Kohei Honda. The two-phase commitment protocol in an extended pi-calculus. *Electronic Notes in Theoretical Comp. Science*, 39(1), 2000.

[6] Stefan Brands and David Chaum. Distance-bounding protocols. In Tor Helleseth, editor, *Proceedings of EUROCRYPT'93, Workshop on the Theory and Application of of Cryptographic Techniques*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359, Lofthus, Norway, May 1993. Springer.

[7] Iliano Cervesato. The dolev-yao intruder is the most powerful attacker. In J. Halpern, editor, *Proceedings of the* 16^{th} *Annual Symposium on Logic in Computer Science*, pages 246–265, Boston, MA, U.S.A., June 2001. IEEE Computer Society Press.

[8] Jolyon Clulow, Gerhard P. Hancke, Markus G. Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In Levente Buttyán, Virgil D. Gligor, and Dirk Westhoff, editors, *ESAS*, volume 4357 of *Lecture Notes in Computer Science*, pages 83–97. Springer, 2006.

[9] Danny Dolev and A. Yao. On the security of public key protocols. In *Proceedings of the* 22^{nd} *Annual Symposium on Foundations of Computer Science*, pages 350–357, October 1981.

[10] Ratan K. Guha, Zeeshan Furqan, and Shahabuddin Muhammad. Discovering man-in-the-middle attacks in authentication protocols. *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7, Oct. 2007.

[11] Gerhard P. Hancke and Markus G. Kuhn. An RFID distance bounding protocol. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 67–73, Athens, Greece, September 2005. ACM Press.

[12] Gerhard P. Hancke and Markus G. Kuhn. Attacks on time-of-flight distance bounding channels. In *WiSec '08: Proceedings of the first ACM conference on Wireless network security*, pages 194–202, New York, NY, USA, 2008. ACM.

[13] Chong Hee Kim, Gildas Avoine, Franois Koeune, François-Xavier Standaert, and Olivier Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In *The 11th International Conference on Information Security and Cryptology - ICISC 2008*, pages 98–115. Springer-Verlag, 2008.

[14] Dennis Kügler. "Man in the Middle" Attacks on Bluetooth. In Rebecca N. Wright, editor, *Financial Cryptography*, volume 2742 of *Lecture Notes in Computer Science*, pages 149–161. Springer, 2003.

[15] Catherine Meadows. A cost-based framework for analysis of denial of service networks. *Journal of Computer Security*, 9(1/2):143–164, 2001.

[16] Catherine Meadows, Radha Poovendran, Dusko Pavlovic, LiWu Chang, and Paul Syverson. Distance bounding protocols: Authentication logic analysis and collusion attacks. *Advances in Information Security: Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, 30, 2006.

[17] Ulrike Meyer and Susanne Wetzel. A man-in-the-middle attack on UMTS. In *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, pages 90–97, New York, NY, USA, 2004. ACM.

[18] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes (parts I & II). *Information and Computation*, 100(1):1–77, September 1992.

[19] Ventzislav Nikov and Marc Vauclair. Yet another secure distance-bounding protocol. Cryptology ePrint Archive, Report 2008/319, 2008. http://eprint.iacr.org/.

[20] Dimitrios N. Serpanos and Richard J. Lipton. Defense against man-in-the-middle attack in client-server systems. In *ISCC*, pages 9–14. IEEE Computer Society, 2001.

[21] Vitaly Shmatikov and Ming-Hsiu Wang. Secure verification of location claims with simultaneous distance modification. In Iliano Cervesato, editor, *ASIAN*, volume 4846 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2007.

[22] Dave Singelée and Bart Preneel. Location verification using secure distance bounding protocols. In *Proceedings of the 2005 IEEE International Workshop on Wireless and Sensor Networks Security*, pages 834–840, Washington DC, USA, November 2005. IEEE Computer Society.

[23] Yu-Ju Tu and Selwyn Piramuthu. RFID Distance Bounding Protocols. In *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

[24] Haidong Xia and José Carlos Brustoloni. Hardening web browsers against man-in-the-middle and eavesdropping attacks. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 489–498, New York, NY, USA, 2005. ACM.