Linking Unlinkability

Mayla Brusó, Konstantinos Chatzikokolakis, Sandro Etalle, Jerry Den Hartog

Linking Unlinkability

Mayla Brusó[1], Konstantinos Chatzikokolakis[2], Sandro Etalle[1,3], and Jerry den Hartog[1]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
[2] CNRS & École Polytechnique, Paris, France
[3] University of Twente, Enschede, The Netherlands

Abstract. Unlinkability is a privacy property of crucial importance for several systems (such as RFID or voting systems). Informally, unlinkability states that, given two events/items in a system, an attacker is not able to infer whether they are related to each other. However, in the literature we find several definitions for this notion, which are apparently unrelated and shows a potentially problematic lack of agreement. This paper sheds new light on unlinkability by comparing different ways of defining it and showing that in many practical situations the various definitions coincide. It does so by (a) expressing in a unifying framework four definitions of unlinkability from the literature (b) demonstrating how these definitions are different yet related to each other and to their dual notion of "inseparability" and (c) by identifying conditions under which all these definitions become equivalent. We argue that the conditions are reasonable to expect in identification systems, and we prove that they hold for a generic class of protocols.

1 Introduction

Unlinkability is a privacy property which holds when an attacker cannot identify the link between two or more items in a system. This property is fundamental in the context of identification systems. For instance, a person who buys an item with an EPC tag at a shop may expect that the protocol used to identify tags prevents his/her tracking.

In this paper we use Radio Frequency Identification (RFID) systems as a case study for our protocol analysis. RFID systems are a wireless technology for automatic identification consisting of a set of tags, readers and a backend. Tags usually offer very limited resources, while backends and readers have standard computational resources. An identification protocol allows tags to authenticate to a backend exchanging information through a reader. One of the main issues raised by the widespread use of RFID is that of privacy. The problem is that anyone in the neighbourhood of a tag may access it wirelessly, and the resource limitation of RFID tags makes it difficult to use full-fledged cryptographic algorithms. The ease of access paves the way to misuse: an attacker could exploit tags to follow the movements of people or goods. The attacker does not even need to break anonymity, since a tag sending the expiry date of a product already allows a certain degree of tracking.

These privacy concerns lead to the definition of *unlinkability* (sometimes called *untraceability* or *privacy*). In the case of RFID systems, unlinkability [17, 21, 4, 5, 22, 8, 26] is satisfied if an attacker is not able to infer whether two sessions have been executed by the same agent. In the RFID literature, unlinkability is usually defined either in a

computational setting in terms of games [10, 17, 21, 4, 22] or in a symbolic setting [12, 13, 2, 3, 7]. [12] proposes a definition of untraceability in a trace-based model. [2, 3] formalize protocols in the applied pi calculus and define weak and strong unlinkability in terms of trace and observational equivalence respectively. Finally, [7] proposes a definition of unlinkability in the applied pi calculus, inspired by the unlinkability games of the computational setting.

As most definitions are different in model and strength, there is no agreement in the literature on the concept of unlinkability. The goal of this paper is to create a better understanding of this notion by comparing the strength of different definitions and determining whether the differences have a practical impact on real world systems. Our contribution is threefold. First, we express four trace-based definitions of unlinkability from the literature in a unifying model. We start with the one of weak unlinkability from [12, 3]. By strengthening it, we obtain the definition of strong unlinkability from [3]. Then, we express two game-based notions [10, 17, 4, 20, 7], and we give a definition that capture them both. We also investigate *inseparability*, a notion dual to unlinkability, which requires that the attacker cannot infer that two messages are *not* linked. Second, we identify a set of conditions and demonstrate that, when they hold, all the above forms of unlinkability and inseparability coincide. Last, we prove that these conditions are satisfied by a generic class of simple identification protocols from [7].

These results help us to understand the essence of these privacy properties. Working in an abstract setting, we can concentrate on their inherent nature – the inability to distinguish certain traces – without dealing with the complications of a concrete model. As a result, the definitions and the conditions under which they coincide become intuitive, while the results can be transferred to a concrete trace model as we do in Section 6.

Plan of the paper. Section 2 briefly introduces epistemic logic. Section 3 presents our abstract trace model. Section 4 states several definitions of privacy using epistemic logic. Section 5 presents several conditions and shows that all the privacy properties coincide under them. Section 6 presents the class of RFID single-step protocols and shows that they satisfy all the conditions stated in Section 5. Section 7 lists the related work. Section 8 and provides conclusions.

2 Preliminaries

In this section we briefly introduce epistemic logic with public announcements, a logic modeling agent knowledge, that we later use to formalize privacy properties. Only the basic concepts are stated here, we refer the reader to [18] for more details.

Let P be a set of propositional constants (atoms). The set $\mathfrak{L}(P)$ of epistemic formulas $\varphi, \psi, \ldots$ over A is given by:

$$\varphi, \psi ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid K\varphi \mid [\varphi]\psi$$

with $p \in P$. The formula $K\varphi$ means "the attacker knows φ", while $[\varphi]\psi$ means "after φ is revealed, ψ holds". The semantics is given in terms of Kripke structures. A Kripke structure M is a tuple $(S, f, \sim)$ where S is a set of possible states, $f : S \to 2^P$ is a function assigning to each state a set of atoms that hold in that state, and $\sim$ is an

equivalence relation on S. Intuitively, $s_1 \sim s_2$ means that from the attacker's point of view, the two states are indistinguishable. The semantics of the logic is given by:

$M, s \vDash p$ iff $p \in f(s)$;
$M, s \vDash \varphi \wedge \psi$ iff $M, s \vDash \varphi$ and $M, s \vDash \psi$;
$M, s \vDash \varphi \vee \psi$ iff $M, s \vDash \varphi$ or $M, s \vDash \psi$;
$M, s \vDash \neg\varphi$ iff $s \nvDash \varphi$;
$M, s \vDash K\varphi$ iff $s' \vDash \varphi$ for all s' such that $s' \sim s$.
$M, s \vDash [\varphi]\psi$ iff ($M, s \vDash \varphi$ implies $M|\varphi, s \vDash \psi$)

Intuitively, $M, s \vDash \varphi$ means that M satisfies φ at state s (we write $s \vDash \varphi$ when M is clear from the context). The interesting case is the knowledge operator K: the attacker knows φ at state s iff φ is satisfied in all states that are indistinguishable from s from the attacker's point of view. For the $[\varphi]$ operator, let $M|\varphi$ be a Kripke structure obtained by M by restricting only to states satisfying φ, i.e. having state space $S' = \{s \in S \mid M, s \vDash \varphi\}$. Intuitively, revealing φ (in a state where φ holds) restricts the model to a smaller one where φ always holds. Then $[\varphi]\psi$ is true if ψ holds in the restricted model.

3 A trace-based model

In a system, agents exchange messages according to a protocol with a specific purpose. To capture one or more protocol runs in our model, we introduce the concept of *transactions*. A transaction starts when the attacker gains access to an agent and lasts until the attacker loses it. During the transaction the attacker can passively eavesdrop or actively forge messages. We allow the attacker to execute an arbitrary number of protocol sessions within a transaction, while knowing that the agent participating in the transaction does not change. However, when a new transaction starts, the agent involved can be either the same as before, or a different one, and the attacker's goal is to distinguish these two cases. The attacker's intentions are captured by the concept of a *strategy*. For example, the attacker might passively eavesdrop a session, then build a new message and send it to the agent executing another transaction, and so on. An attacker strategy and a mapping of transactions to agent identities, completely defines a trace.

Definition 1. *A system is a tuple* $(A, \Sigma, \mathrm{T}, \sim)$ *where:*

- $A = \{a_1, a_2, \ldots\}$ *is a (possibly infinite) set of agents; we assume an ordered set of transactions* $\{p_1, p_2, \ldots\}$*, and we denote by* $\Pi_n = \{p_1, \ldots, p_n\} \to A$ *the set of assignments of* n *transactions to agents;*
- Σ *is a set of strategies; each strategy* $\sigma \in \Sigma$ *has a length* $|\sigma|$*; we denote the set of transactions involved in the strategy by* $Dom_\sigma = \{p_1, \ldots, p_{|\sigma|}\}$*;*
- $\mathrm{T} = \{(\pi, \sigma) \mid \sigma \in \Sigma, \pi \in \Pi_{|\sigma|}\}$ *is a set of traces; each trace* $\tau \in \mathrm{T}$ *is a tuple* (π, σ) *where* σ *is a strategy and* π *is a mapping of transactions to agents;*
- $\sim$ *is an equivalence relation on* T *such that* $(\pi_1, \sigma_1) \sim (\pi_2, \sigma_2) \Rightarrow \sigma_1 = \sigma_2$.

A trace $\tau \in \mathrm{T}$ is a complete execution of the system, and is determined by a strategy σ, chosen by the attacker, and a mapping π, which the attacker does not control, that defines which agent participates in each transaction. A protocol is an abstract object

Table 1. Unlinkability and Inseparability

Property	*Epistemic formula*	*The attacker cannot infer*
WU	$\neg Klink(p, p')$	that two messages p, p' are linked
SU	$\neg K(anyLink)$	the existence of linked messages
GB-1	$[\pi_a \vee \pi_{a'}]\neg K\pi_a$	the mapping π_a even when $\pi_a \vee \pi_{a'}$ is revealed
GB-2	$[\pi_{a,a} \vee \pi_{a_1,a_2}]\neg K\pi_{a,a}$	the mapping $\pi_{a,a}$ even when $\pi_{a,a} \vee \pi_{a_1,a_2}$ is revealed
WI	$\neg K(unlink(p, p'))$	that two messages p, p' are unlinked
SI	$\neg KanyUnlink$	the existence of unlinked messages

that describes the behaviour of the agents in a system. Each protocol generates the set T of all the possible traces that can be obtained under any attacker strategy $\sigma \in \Sigma$. The relation $\sim$ is crucial for defining privacy properties. Consider two traces $\tau_1 = (\pi_1, \sigma)$ and $\tau_2 = (\pi_2, \sigma)$, produced when the attacker chooses the strategy σ and interacts with two different sets of agents. If $\tau_1 \sim \tau_2$, it means that when using the strategy σ, the attacker cannot distinguish between the sets of the agents.

We sometimes use π_τ to emphasize that it belongs to the trace τ. For a trace $\tau = (\pi, \sigma)$, we write Dom_τ for Dom_σ, $|\tau|$ for $|\sigma|$ and A_τ, A_π for the image of π (i.e. the set of agents involved in the trace). For a mapping $\pi \in \Pi_n$ we define $|\pi| = n$ and we denote $\Pi = \cup_{n\geq 1}\Pi_n$. Since transactions are ordered, we write mappings as sequences of agents, e.g. $\pi = (a_4, a_1, a_2)$ assigns agent a_4 to the first transaction, a_1 to the second and a_2 to the third ones. We extend $\sim$ to mappings as follows:

$$\pi \sim \pi' \quad \text{iff} \quad |\pi| = |\pi'| \text{ and } (\pi, \sigma) \sim (\pi', \sigma)\ \forall \sigma \in \Sigma \text{ s.t. } |\sigma| = |\pi|$$

Note that we keep our model abstract and do not explicitly define the messages in the protocol, the exact strategies Σ and the relation $\sim$. We assume that these are produced by a concrete protocol model (such as the one given in Section 6).

4 Unlinkability definitions

In this section we express several definitions of unlinkability from the literature in our trace-based model: the weak unlinkability of [12, 3], the strong unlinkability of [3], and two game-based definitions [10, 17, 4, 20, 7]. Finally, we introduce the notion of inseparability, which does not appear in the literature, but arises as a natural dual notion to unlinkability. Our purpose is not a technical comparison between definitions expressed in different models, but a comparison between the ideas behind each definition.

Table 4 gathers all the resulting notions.

4.1 Kripke structure

To express unlinkability using epistemic logic, first we have to define a Kripke structure $M = (\mathrm{T}, f, \sim)$ corresponding to a system $(A, \Sigma, \mathrm{T}, \sim)$. T is the set of states and the attacker's indistinguishability relation $\sim$ is provided directly by the system. The set of atomic propositions P and the assignment function $f : \mathrm{T} \to P$ are built as follows:

$$P = \Pi \cup \{link(p, p') \mid p, p' \in Dom_\tau, p \neq p', \tau \in \mathrm{T}\}$$
$$f((\pi, \sigma)) = \{\pi\} \cup \{link(p, p') \mid \pi(p) = \pi(p')\}$$

We use two types of propositions: $\pi \in \Pi$ denotes that the mapping of the trace is π, while $link(p, p')$ denotes that the transactions p, p' are linked, i.e. they are mapped to the same agent in a given trace. Note that $link(p, p')$ holds iff $p \neq p'$, which we implicitly assume in all the definitions.

4.2 Weak Unlinkability

The first definition is the one of weak unlinkability of [12, 3]. Although presented in different models, they are similar in nature, both requiring that, given a trace where two messages are linked, an equivalent trace must exist where those messages are unlinked.

Definition 2 (Weak unlinkability). *A protocol generating the set of traces* T *guarantees weak unlinkability iff*

$$\forall \tau \in \mathrm{T}, p, p' \in Dom_\tau : \tau \vDash \neg K(link(p, p'))$$

This definition imposes that the attacker does not know whether any two given transactions are linked to each other. This implies that for all traces τ and all pairs of distinct transactions, there must exist an equivalent trace $\tau' \sim \tau$ in which the transactions are mapped to two different agents. So the above definition can be written as:

$$\forall \tau \in \mathrm{T}, p, p' \in Dom_\tau : \exists \tau' \in \mathrm{T}, \tau' \sim \tau : \tau' \vDash \neg link(p, p')$$

which corresponds exactly to the one of [12, 3]. The weakness of this definition lies in the fact that it does not completely prevent the attacker from obtaining knowledge about linked transactions. For example, in a system satisfying weak unlinkability, the attacker could still know that p_1 is linked to either p_2 or p_3, but without knowing which one.

4.3 Strong Unlinkability

[3] also defines a strong version of unlinkability by requiring that a system is equivalent to one where each agent executes one session only. Their definition, in a simplified form and without entering into details, requires that $!\,T \approx !\,T_s$ where $T = \nu m.\ init.\ !main$ and $T_s = \nu m.\ init.\ main.\ T$ represents an agent executing an initialization phase ($init$) and an unbounded number (denoted by !) of protocol sessions ($main$), while T_s is an agent executing one session. $\approx$ denotes observational equivalence in [3], while we use trace equivalence here because it is directly expressable in our model. To capture this definition in our framework, we not only require that the attacker is not able to infer the link between two given transactions, but also the *existence* of linked transactions.

Definition 3 (Strong unlinkability). *We say that a protocol generating the set of traces* T *guarantees strong unlinkability iff*

$$\forall \tau \in \mathrm{T} : \tau \vDash \neg K(anyLink) \forall p, p' \in Dom_\tau$$

where $anyLink = \bigvee_p \bigvee_{p'} link(p, p')$.

$anyLink$ holds for a trace if there exists at least a linked transaction. Thus, strong unlinkability holds iff the attacker does not know whether there exists a link at all. For this to hold, each trace must be equivalent to one with no linked transactions:

$$\forall \tau \in \mathrm{T} : \exists \tau' \in \mathrm{T}, \tau' \sim \tau : \forall p, p' \in Dom_\tau : \tau' \vDash \neg link(p, p')$$

This formulation corresponds exactly to the definition of Arapinis et al. since τ' is a trace that can be produced by the process $!T_s$, where no agent executes more than one transaction.

4.4 Game-based definitions of privacy

In the game-based definitions, privacy is defined as the result of a game between an attacker (whose goal is to distinguish between the actions of different agents) and a challenger. We refer to two different types of game-based definition of privacy: the first is related to the definition of [21], variations of which can be found also in [10, 17, 4, 22], while the second corresponds to the definition given by [10, 20]. We demonstrate that in our model, these two classes of definitions are equivalent to each other and to a third *simpler* definition of game-base unlinkability based on trace equivalence.
Both types of games consist of three phases. In the first game, which we call two-agents game unlinkability, during the first phase the attacker can interact with all the agents of the system. In the second phase, the attacker is asked to select two agents a, a'. The challenger selects an agent $x \in \{a, a'\}$, and gives x to the attacker, hiding its identity. The attacker can interact with all the agents, including a and a', and, in the final phase, she wins the game if she can infer whether x is a or a' with non-negligible probability. We use π_x to denote a partial mapping from transactions to agents, where some transactions are mapped to a variable x, while all the others are known to the attacker; Π_x is the set of all the partial mappings. π_a is a mapping obtained from π_x by mapping to an agent a all transactions previously mapped to the variable x. We formalize this game by requiring that the attacker cannot infer whether she is given a mapping π_a or $\pi_{a'}$. Thus, a protocol generating the set of traces T guarantees *two-agents game unlinkability* iff

$$\forall \tau \in \mathrm{T}, a, a' \in A, \pi_x \in \Pi_x : \tau \vDash [\pi_a \vee \pi_{a'}] \neg K \pi_a \tag{1}$$

Although the only forbidden knowledge concerns π_a, (1) is equivalent to $\tau \vDash [\pi_a \vee \pi_{a'}] \neg K \pi_{a'}$, thus the attacker cannot know $\pi_{a'}$ either. This property implies the equivalence of the mappings π_a and $\pi_{a'}$ under all strategies, thus we can express (1) as:

$$\forall a, a' \in A, \pi_x \in \Pi_x : \pi_a \sim \pi_{a'}$$

The second game, which we call three-agents game unlinkability, can be found in a computational [10, 20] and a formal setting [7]. Only the second phase differs from the

previous game: the attacker selects three agents a, a_1, a_2 and the challenger gives her two agents x, y, that are set to either $x = y = a$ or $x = a_1, y = a_2$; the attacker wins if she can infer whether x and y are linked. We now use $\pi_{x,y}$ as a partial mapping, $\Pi_{x,y}$ as a set of all the partial mappings and $\pi_{a,b}$ as a complete mapping obtained from $\pi_{x,y}$. We require that the attacker cannot infer whether she is given a mapping $\pi_{a,a}$ or π_{a_1,a_2}. A protocol generating the set of traces T guarantees three-agents game unlinkability iff

$$\forall \tau \in \mathrm{T}, a, a_1, a_2 \in A, \pi_{x,y} \in \Pi_{x,y} : \tau \vDash [\pi_{a,a} \vee \pi_{a_1,a_2}] \neg K \pi_{a,a} \tag{2}$$

In terms of equivalence of traces, (2) can be restated as follows:

$$\forall \tau \in \mathrm{T}, a, a_1, a_2 \in A, \pi_{x,y} \in \Pi_{x,y} : \pi_{a,a} \sim \pi_{a_1,a_2}$$

It is easy to see that both the games require all mappings to be equivalent. Thus, we give a definition of game-based unlinkability which unifies these two notions.

Definition 4 (Game-based unlinkability). *We say that a protocol generating the set of traces* T *guarantees game-based unlinkability iff*

$$\forall \pi, \pi' \in \Pi, |\pi| = |\pi'| : \pi \sim \pi' \tag{3}$$

Each of the referenced works uses a variant of either (1) or (2), while [10] mentions both, referring to the first as untraceability and to the second as unlinkability, but does not explore the relation between them. Instead, we can prove that they reduce to Def. 4:

Theorem 1. *A protocol satisfies game-based unlinkability iff it satisfies two-agents game unlinkability, which it does iff it satisfies three-agents game unlinkability.*

From now on we will only use the definition of game-based unlinkability. However, by Theorem 1, all the results that hold for game-based unlinkability hold also for the other game-based notions. Finally, as one may expect, we can show that strong unlinkability and game-based unlinkability are both stronger than weak unlinkability.

Theorem 2. *Strong unlinkability and game-based unlinkability imply both weak unlinkability.*

Note that strong unlinkability has already been proven to imply weak unlinkability by [3] in their model in the applied pi calculus.

4.5 Inseparability

In some situations we want to hide the existence of *unlinked* transactions. In fact, an attacker might be interested in changes in the system rather than in tracking agents. For example, consider a high security location protected by a guard who authenticates himself using an RFID tag. The attacker might want to be alerted when a *new* guard appears. The definitions of weak and strong unlinkability impose no condition when two messages are unlinked, thus we introduce the concept of *inseparability*, which requires that the attacker does not know that two transactions are *not* linked.

Definition 5 (Inseparability). *We say that a protocol generating the set of traces* T *guarantees weak inseparability iff* $\forall \tau \in \mathrm{T}, p, p' \in Dom_\tau : \tau \vDash \neg K(unlink(p, p'))$ *and strong inseparability iff* $\forall \tau \in \mathrm{T} : \tau \vDash \neg K(anyUnlink)$ *where* $unlink(p, p') = \neg link(p, p')$ *and* $anyUnlink = \bigvee_p \bigvee_{p' \neq p} unlink(p, p')$.

As expected, strong inseparability is stronger than weak inseparability. On the other hand, somewhat surprisingly, game-based unlinkability is stronger than strong inseparability, although game-based unlinkability is incomparable to strong unlinkability, which is incomparable to strong inseparability. The reason is that game-based unlinkability enforces all traces to be equivalent to one performed by a single agent.

Theorem 3. *Game-based unlinkability implies strong inseparability, which in turn implies weak inseparability.*

The above properties are in general different (the implications not covered here do not hold in general), as shown by the following examples; in the next section we investigate conditions under which some or all of the properties become equivalent.

4.6 RFID systems: protocols where the properties do not coincide

In this section we list some examples of RFID protocols that guarantee only some of the properties described in Section 4. They are variations of the OSK protocol (see Section 6, although understanding the protocol is not needed to follow the examples) that satisfies all the properties. Here we introduce features that cause some properties to fail. These features are artificial and unlikely to be present in realistic protocols. In fact, in the next section, we identify some conditions under which all properties coincide.

Example 1 (System with a bounded number of tags). Consider a system with a bounded number of tags. If the attacker observes a number of sessions greater than the number of tags, she knows that there exist some linked sessions, although she cannot point to any specific message, i.e. weak unlinkability holds, but strong unlinkability does not. Still, all the traces of equal length produced by the OSK protocol are equivalent, thus game-based unlinkability holds. As a consequence, also strong inseparability holds. □

Example 2 (System with several "types" of tags). Consider a system with two distinguishable types of tags $Type_1$ and $Type_2$, for example because they differ in technical characteristics. Weak inseparability and game-based unlinkability are violated since the attacker can distinguish tags of different type. Instead, strong unlinkability holds: the adversary cannot know the existence of linked transactions since all transactions of the same type could come from different tags. Together with the previous example, this shows that strong unlinkability and the game-based definition are incomparable. However, if the number of tags of $Type_2$ is bounded, we have a situation similar to the previous example (although the total number of tags might still be unbounded); again, strong unlinkability is violated while weak unlinkability holds. □

Example 3 (Protocol outputs depending on past sessions I). Consider a variation of the OSK protocol where the reader beeps when the session it is executing is linked to a previous session, but only if at least two tags appeared before it. This protocol satisfies weak unlinkability: the beep does not allow the attacker to point to any two specific linked sessions. Despite this, the attacker knows that the session that made the reader

beep must be linked to a past session. Consider the observation where the reader beeps at the third session. The beep tells him that the third session is either linked to the first or the second one, and the first two sessions are not linked, violating strong unlinkability and weak inseparability. Since not all mappings are equivalent to each other due to the beep, game-based unlinkability is also violated. □

Example 4 (Protocol outputs depending on past sessions II). Consider a variation of the OSK protocol where the reader beeps when the third tag of a trace first appears. The protocol satisfies weak unlinkability, but violates strong unlinkability: if the reader beeps after four or more sessions, there must be a link. Game-based unlinkability is violated, since not all the traces are equivalent. It also breaks weak inseparability, since a beep during the third session means that the first three sessions are unlinked. □

Example 5 (Protocol outputs depending on past sessions III). Consider a variation of the OSK protocol where the reader beeps when it sees at least two tags and one link. The protocol satisfies weak unlinkability but violates strong unlinkability and game-based unlinkability. Strong inseparability is also violated, because a beep means that at least two tags run some session. However, the attacker cannot link two specific sessions, thus weak inseparability holds. □

5 Conditions under which the properties coincide

In this section we identify a (large) class of protocols C and we demonstrate for these protocols that all the notions of unlinkability and inseparability are equivalent: if a protocol in C satisfies any of them, then it satisfies all of them. The class C is given by all the protocols satisfying the five conditions that we identify in the next section, where we argue that most RFID protocols actually satisfy them, at least in their abstract form.

5.1 Conditions

Condition Unbounded number of agents. As we showed in Example 1, a system with a bounded number of agents cannot satisfy strong unlinkability, since observing a greater number of transactions reveals that at least two transactions are linked. Thus, protocols in C contain an unbounded number of agents.

Definition 6. *A protocol has an unbounded number of agents iff* $\forall n > 0 \exists \tau \in \mathrm{T} : |A_\tau| = n$.

Clearly, an unbounded number of agents is not realistic. However, identification systems have usually a wide number of agents, thus an attacker cannot usually communicate with all of them in a limited amount of time, and does not know the total number of agents. This is why, at an abstract level, this condition is often assumed in the literature.

Condition Renaming. As shown in Example 2, having distinguishable types of agents can be problematic. However, in real systems agents are usually identical in functionality, differing only in the secret information. As a result, we can expect that replacing *all* transactions of an agent with a new one will not have a visible effect.

Definition 7. *Let π be a mapping, $a \in A_\pi$ and $a' \notin A_\pi$. The renaming of a to a' in π, denoted by $\pi[a'/a]$, is a mapping such that*

$$\pi[a'/a](p) = \begin{cases} a' & \text{if } \pi(p) = a \\ \pi(p) & \text{otherwise} \end{cases}$$

A protocol satisfies the condition Renaming iff $\pi \sim \pi[a'/a] \forall \pi \in \Pi, a' \notin A_\pi$.

In other words, only the positions in which an agent appears in the trace matters. In the rest of the paper, we assume that this condition holds and we write all mappings in a normalized form, sorting the agents by their order of appearance: the first agent is always a_1, the next agent different from a_1 will be a_2 and so on. For example, we write $(a_1, a_1, a_2, a_3, a_1, a_2)$ instead of $(a_5, a_5, a_3, a_1, a_5, a_3)$. Thus, the number of possible traces is always finite for any given length, even when the number of agents is infinite.

Condition Swapping. Consider $\pi_1 = (\ldots, a_1, a_2, \ldots)$ and $\pi_2 = (\ldots, a_3, a_4, \ldots)$, two mappings where the k-th and $k+1$-st transactions involve different agents. Now assume that $\pi_1 \sim \pi_2$ and consider the mappings $\pi_1' = (\ldots, a_2, a_1, \ldots)$ and $\pi_2' = (\ldots, a_4, a_3, \ldots)$ that differ from π_1 and π_2 only for the k-th and $k+1$-st agents. We require that different agents act in an independent way and the execution of the one should not affect the execution of the other. Thus, π_1' and π_2' should also be indistinguishable.

Definition 8. *Let π be a mapping. The swapping of π at position $k < |\pi|$, denoted by $sw_k(\pi)$ is a new mapping such that:*

$$sw_k(\pi)(p_i) = \begin{cases} \pi(p_{k+1}) & \text{if } p_i = p_k \\ \pi(p_k) & \text{if } p_i = p_{k+1} \\ \pi(p_i) & \text{otherwise} \end{cases}$$

A protocol satisfies the condition Swapping iff $\pi \sim \pi' \Rightarrow sw_k(\pi) \sim sw_k(\pi')$ for all π, π', k such that $\pi(p_k) \neq \pi(p_{k+1})$ and $\pi'(p_k) \neq \pi'(p_{k+1})$.

In practice, the condition Swapping simply implies that the agents are independent of each other. As a consequence, the order of appearance of agents does not affect the knowledge of the attacker. Note that this condition is violated in the Example 3. The mappings (a_1, a_1, a_2) and (a_1, a_2, a_3) produce the same observations $(\_, \_, \_)$. However, by swapping the second and third transactions, we obtain the mappings (a_1, a_2, a_1) and (a_1, a_2, a_3), which produce different observations: $(\_, \_, beep)$ and $(\_, \_, \_)$. This happens because the execution of one agent depends on the previous executions of other agents.

Conditions: Extension I and II. We now introduce two last conditions. The first states that two equivalent mappings should preserve their equivalence when extended with a new transaction mapped to a new agent. The underlying idea is that adding a new agent should not make two traces distinguishable, if they could not be distinguished before.

Definition 9. *Let π be a mapping. The extension of π with a new agent $a \notin A_\pi$, denoted by $extn(\pi)$, is a mapping of length $|\pi| + 1$ such that*

$$extn(\pi)(p_i) = \begin{cases} \pi(p_i) & i \leq |\pi| \\ a & i = |\pi| + 1 \end{cases}$$

A protocol satisfies the condition Extension I if $\pi \sim \pi' \Rightarrow extn(\pi) \sim extn(\pi')$ for all mappings π, π'.

Note that this condition is violated by the protocol in the Example 4. The mappings (a_1, a_1, a_1), (a_1, a_1, a_2) are equivalent (produce no beep), while their extensions are not, since (a_1, a_1, a_1, a_2) does not make the reader beep while (a_1, a_1, a_2, a_3) does.

For the second extension condition, consider two equivalent mappings π_1, π_2 of length n. We extend these mappings with a new transaction p_{n+1}, which is mapped to the last agent appearing in each mapping. Since the attacker had access to these agents during the transaction p_n, and she could not distinguish the two mappings, she does not gain any new knowledge by querying the same agents in the new transactions.

Definition 10. *Let π be a mapping. The extension of π with the last appearing agent, denoted by $extl(\pi)$ is a mapping of length $|\pi| + 1$ such that*

$$extl(\pi)(p_i) = \begin{cases} \pi(p_i) & i \leq |\pi| \\ \pi(p_{|\pi|}) & i = |\pi| + 1 \end{cases}$$

A protocol satisfies the condition Extension II if $\pi \sim \pi' \Rightarrow extl(\pi) \sim extl(\pi')$ for all mappings π, π'.

This condition is violated by the protocol in the Example 5. In fact, the traces with mappings (a_1, a_1) and (a_1, a_2) are not distinguishable, while their extensions are distinguishable because the first trace produces no beep while the second beeps.

5.2 Equivalence results

In this section we demonstrate that under the conditions stated in Section 5.1, all the definitions of unlinkability coincide. Moreover, we prove that, under a smaller set of conditions, the definitions of inseparability coincide as well as all the strong definitions.

Theorem 4 (Unification of unlinkability). *If a protocol guarantees all the conditions of Section 5.1 then all the unlinkability properties (weak unlinkability, strong unlinkability, game-based unlinkability) coincide.*

The intuition is that, under these conditions, all the definitions require all the equal length mappings to be equivalent in particular to a mapping where all the transactions are not linked.

Theorem 5 (Unification of inseparability). *Under the conditions Renaming and Extension II, a protocol satisfies weak inseparability iff it satisfies strong inseparability.*

Again, under these conditions, both properties require all the mappings of the same length to be equivalent, in particular to one where all the transactions are linked.

Theorem 6 (Unification of strong properties). *Under the conditions Unbounded number of agents and Renaming, all the strong properties (strong unlinkability, game-based unlinkability, strong inseparability) coincide.*

It is worth noting that this result uses weaker assumptions than the previous ones; indeed, the conditions Swapping and Extension I and II are not needed. An unbounded number of agents is required to guarantee the existence of completely unlinked traces (for strong unlinkability), while the condition Renaming implies that agents are not observationally different (for strong inseparability and game-based unlinkability).

Finally, we can state the result we were aiming at.

Corollary 1. *If a protocol guarantees all the conditions of Section 5.1 then all the forms of unlinkability and inseparability coincide.*

This result shows that all the privacy definitions of Section 4 coincide under a set of conditions.

6 RFID systems: single-step protocols

In this section, we show that the conditions stated in the previous section are satisfied by a generic class of "single-step" protocols [7]. To this end, we express such protocols in the applied pi calculus, using the model of [3], which defines an instantiation of our model, i.e. it provides a concrete set of traces and an equivalence relation between them.

Single-step protocols consist of a single message from a tag to a reader. A protocol of this class is shown in Figure 1. Each tag is initialized with an internal state S_0, which contains a secret s that is unique to that tag. At each run of the protocol, the tag computes an output function $O(S)$ on its current state S, and sends the result to the reader. Then, the tag computes an update function $U(S)$ on its current state S. O and U can be arbitrary functions, using any cryptographic operation that can be modelled by an equational theory in the applied pi calculus. As discussed in [7], two well-known protocols from the literature, namely the OSK protocol [21] and the basic hash protocol of [27], fall in this class.

6.1 Modelling single-step protocols

We model single-step protocols in the applied pi calculus [1], a language for describing concurrent processes and their interaction. It extends the pi calculus [19] adding the possibility to model cryptographic primitives using a signature and an equational theory. A detailed description of the calculus is available in [1]; here, we only assume a basic understanding of the calculus.

Tags are modelled as processes in the calculus, using a public channel c to communicate with the reader. To model the state of a tag, we use an internal channel w. The content of the state is available to the tag by a sub-process $\overline{w}\langle S\rangle$ running in parallel to

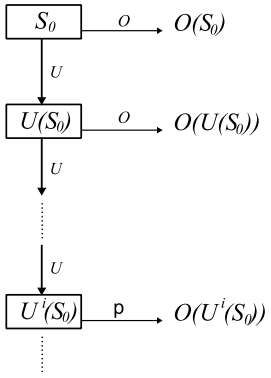

Fig. 1. A single-step protocol.

it, so the tag can read the state by an input on w. A tag execution can be modelled as follows:

$$TagExec \triangleq c(\_).\ w(x).\ \nu\tilde{\rho}.\ \overline{c}\langle O(x)\rangle.\ \overline{w}\langle U(x)\rangle$$

The tag is first triggered by an input on the public channel c (without reading a value). Then, it reads the current state x by an input on w and outputs $O(x)$ on a public channel. $\nu\tilde{\rho}$ denotes the possibility of generating fresh nonces for the output. Finally, the tag outputs $U(x)$ on w, updating its state with the new value.

A complete tag starts with an initial state S_0, containing the tag secret s, and can execute an unbounded number of sessions.

$$Tag \triangleq \nu s.\nu w.(\overline{w}\langle S_0\rangle \mid !TagExec)$$

Note that the secret is private to the tag, thus we use a freshly generated name s. We also restrict w so that only the tag can access its state. Finally, the complete system P consists of an unbounded number of tags: $P \triangleq !Tag$. Here the reader only triggers tags. Since c is public, the tag can be triggered by any external process, so we can simply omit the reader altogether.

6.2 Instantiating our trace model

The system P can perform labelled transitions, according to the semantics of the applied pi calculus. We denote by $\stackrel{\alpha}{\Longrightarrow}$ a sequence of internal transitions, followed by the visible transition α, followed again by internal transitions. A trace tr is a sequence

$$tr = P \stackrel{\alpha_1}{\Longrightarrow} P_1 \stackrel{\alpha_2}{\Longrightarrow} P_2 \stackrel{\alpha_3}{\Longrightarrow} \ldots \stackrel{\alpha_q}{\Longrightarrow} P_q$$

Two traces are equivalent, denoted by $tr_1 \sim_{tr} tr_2$ if they contain the same transitions, and all intermediate processes are *statically* equivalent according to the definition of [1], which states that the processes provide the attacker with the same static knowledge. We refer to [3] for the formal definition of $\sim_{tr}$.

To instantiate our trace model, we need do define a set of agents A, a set of strategies Σ such that a strategy σ together with a mapping π give rise to a trace $\tau = (\pi, \sigma)$, and an equivalence relation $\sim$ between traces. The agents $A = \{a_i | i \in \mathbb{N}\}$ correspond to the tags of the system. In the applied pi calculus model, we use replication to denote an unbounded number of tags. We identify the tags by their secret s, which is unique for each tag. When a_i is spawned we denote its secret by s_i.

Since tags in single-step protocols have no input, the only thing that the attacker can decide is how many transactions she will run, and how many protocol executions she will trigger in each transaction. Thus, a strategy σ is a sequence $\sigma = (\sigma_1, \ldots, \sigma_k)$ such that k is the number of transactions the attacker performs, and σ_i the number of executions that she triggers in the transaction p_i. A mapping π determines which tag will participate in each transaction. Given a strategy σ and a mapping π, we can define a unique trace $tr(\pi, \sigma)$ starting from P. We also define trace equivalence as follows:

$$(\pi, \sigma) \sim (\pi, \sigma') \quad \text{iff} \quad tr(\pi, \sigma) \sim_{tr} tr(\pi, \sigma')$$

Now that we have a concrete trace model for single-step protocols, we can show that they satisfy all the conditions of Section 5.1.

Theorem 7. *Single-step protocols satisfy all the conditions of Section 5.1, thus all the unlinkability and inseparability properties coincide.*

We can conclude that all the forms of unlinkability and inseparability defined in Section 4 coincide for the class of single-step protocols. As a consequence, if any of these properties is proven to hold for a single-step protocol, by Theorem 7 all the unlinkability and inseparability properties should be satisfied.

7 Related work

Our work makes direct use of several definitions of unlinkability from the literature. As explained in detail in Section 4, we express the notion of weak unlinkability of [12, 3], strong unlinkability of [2, 3], and game-based definitions of [10, 17, 21, 4, 22, 20]. While all these works have given their own definitions of privacy properties in a very specific context, ours provides a more general and abstract framework where all other definitions can be captured.

Epistemic models have been used in the past to formalize privacy. Similarly to our work, [15] gives general privacy definitions for a multiagent system using a modal logic of knowledge. The paper considers different levels of strength for unlinkability, providing some probabilistic definition as well. In [9], epistemic logic is used to give intuitive definitions of privacy in voting systems, with the applied pi calculus as the underlying model. Similarly, [11] proposes a framework in which protocols are expressed in a process language while security properties in a logic with both temporal and epistemic operators. The properties considered in the above works are quite different than the unlinkability properties that we consider in this paper. Moreover, the above works are involved with the mechanics of the corresponding formalisms, while we try to completely abstract from the concrete model, viewing a system as an abstract set of traces.

Several other definitions of unlinkability have also been studied in the literature. A logic approach has been followed in [25], where an axiom system is defined to reason about anonymity, and in [16] that expresses privacy properties using logic and models the system through other formalisms, like CSP, combining two different techniques. As in our work, logic is used to define in a natural way privacy properties, while having an abstract model applicable to any real system. However, while our work focuses on unlinkability, [15] and [25] study anonymity, namely a property that ensures that the identity of the agent which executes some action remains hidden from other observers. [23] proposes a terminology for anonymity, unlinkability, unobservability and pseudonimity. While it aims at clarifying terminology at an informal level, our work aims instead at comparing definitions of unlinkability in a unifying formal model.

Finally, other papers introduce the notion of unlinkability using approaches based on information theory. Examples are [14], [24], and [6] that give probabilistic descriptions of unlinkability, quantifying the linkability of items in the system. Our work does not provide any probabilistic definition, but this would be possible following the approach used in [15], that we leave as future work.

8 Conclusion and future work

In this paper we studied the privacy notion of unlinkability. We captured several definitions from the literature into a simple abstract model based on epistemic logic, obtaining natural and intuitive definitions in terms of the attacker's knowledge. We also identified inseparability, a notion dual to unlinkability, in weak and strong forms. Moreover, we showed that these privacy definitions are different in general, but do coincide in systems satisfying a set of conditions. Finally, we proved that the conditions hold for a class of identification protocols.

As future work, we plan at investigating probabilistic descriptions of unlinkability. We also plan at developing a more concrete model in the style of [11]. This work bridges the gap between operational semantics and epistemic logic, offering a combined framework where it is possible to easily model the behavior of a protocol in a process language with an operational semantics and reason about properties expressed in a rich epistemic temporal logic. This would allow to automatically verify privacy properties.

References

1. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: Proc. of POPL. pp. 104–115 (2001)
2. Arapinis, M., Chothia, T., Ritter, E., Ryan, M.: Untraceability in the applied pi-calculus. In: Proc. of ICITST. pp. 1–6. IEEE (2009)
3. Arapinis, M., Chothia, T., Ritter, E., Ryan, M.: Analysing unlinkability and anonymity using the applied pi calculus. In: Proc. of CSF. pp. 107–121. IEEE Computer Society (2010)
4. Avoine, G.: Adversary model for radio frequency identification. Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology, Lausanne, Switzerland (2005)
5. Avoine, G.: Cryptography in Radio Frequency Identification and Fair Exchange Protocols. Ph.D. thesis, EPFL, Lausanne, Switzerland (2005)

6. Berman, R., Fiat, A., Ta-Shma, A.: Provable unlinkability against traffic analysis. In: Proc. of Financial Cryptography. pp. 266–280. LNCS, Springer (2004)
7. Brusó, M., Chatzikokolakis, K., den Hartog, J.: Formal verification of privacy for rfid systems. In: Proc. of CSF. pp. 75–88. IEEE Computer Society (2010)
8. Burmester, M., Le, T.V., de Medeiros, B.: Provably secure ubiquitous systems: Universally composable rfid authentication protocols. In: Proc. of Securecomm. pp. 1–9 (2006)
9. Chadha, R., Delaune, S., Kremer, S.: Epistemic logic for the applied pi calculus. In: Proceedings of IFIP. LNCS, vol. 5522, pp. 182–197. Springer, Lisbon, Portugal (2009)
10. Chatmon, C., van Le, T., Burmester, M.: Secure anonymous RFID authentication protocols. Technical Report TR-060112, Florida State University, Tallahassee, Florida, USA (2006)
11. Dechesne, F., Mousavi, M.R., Orzan, S.: Operational and epistemic approaches to protocol analysis: Bridging the gap. In: Proc. of LPAR. LNCS, vol. 4790, pp. 226–241 (2007)
12. van Deursen, T., Mauw, S., Radomirovic, S.: Untraceability of rfid protocols. In: Proc. of WISTP. LNCS, vol. 5019, pp. 1–15. Springer (2008)
13. van Deursen, T., Radomirovic, S.: Algebraic attacks on rfid protocols. In: Proc. of WISTP. LNCS, vol. 5746, pp. 38–51. Springer (2009)
14. Franz, M., Meyer, B., Pashalidis, A.: Attacking unlinkability: The importance of context. In: Proc. of Privacy Enhancing Technologies. LNCS, vol. 4776, pp. 1–16. Springer (2007)
15. Halpern, J.Y., O'Neill, K.R.: Anonymity and information hiding in multiagent systems. In: Proc. of CSFW. pp. 75–88. IEEE Computer Society (2003)
16. Hughes, D., Shmatikov, V.: Information hiding, anonymity and privacy: A modular approach. Journal of Computer Security 12, 3–36 (2004)
17. Juels, A., Weis, S.A.: Defining strong privacy for rfid. In: Proc. of PerCom Workshops. pp. 342–347. IEEE Computer Society (2007)
18. Meyer, J.J.C., Hoek, W.v.d.: Epistemic Logic for AI and Computer Science. Cambridge University Press, New York, NY, USA (2004)
19. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, parts i and ii. I and II. Information and Computation 100, 1–77 (1989)
20. Nohl, K., Evans, D.: Privacy through noise: a design space for private identification. In: Annual Computer Security Applications Conference (ACSAC 2009) (2009)
21. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic approach to "privacy-friendly" tags. In: Proc. of RFID Privacy Workshop (2003)
22. Ouafi, K., Phan, R.C.W.: Privacy of recent rfid authentication protocols. In: Proc. of ISPEC. LNCS, vol. 4991, pp. 263–277. Springer (2008)
23. Pfitzmann, A., Köhntopp, M.: Anonymity, unobservability, and pseudonymity - a proposal for terminology. In: Workshop on Design Issues in Anonymity and Unobservability. pp. 1–9 (2000)
24. Steinbrecher, S., Kopsell, S.: Modelling unlinkability. In: Proc. of Privacy Enhancing Technologies. LNCS, vol. 2760, pp. 32–47. Springer (2003)
25. Syverson, P.F., Stubblebine, S.G.: Group principals and the formalization of anonymity. In: Proc. of World Congress on Formal Methods. LNCS, vol. 1708, pp. 814–833 (1999)
26. Vaudenay, S.: On privacy models for rfid. In: Proc. of ASIACRYPT. LNCS, vol. 4833, pp. 68–87. Springer (2007)
27. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost RFID systems. In: Proc. of SPC. LNCS, vol. 2802, pp. 201–212. Springer (2003)