Proceedings of the Federated Conference on Computer Science and Information Systems pp. 407–416 DOI: 10.15439/2018F43 ISSN 2300-5963 ACSIS, Vol. 15

# Group Anonymity in Security Protocols

Ferucio Laurenţiu Ţiplea and Cosmin Vârlan
Department of Computer Science, "Al.I.Cuza" University of Iaşi
Iaşi, Romania, e-mail: {ferucio.tiplea@uaic.ro, vcosmin@info.uaic.ro}

***Abstract*—Group anonymity, as an instance of information hiding, means that an agent is not identifiable within a group of agents with respect to an observer. In this paper we define group anonymity in security protocols by taking into account two types of observers: honest agents, as local observers of the protocol execution, and intruders (active or passive), as global observers of the protocol execution. It is shown that an action may be group anonymous in a protocol under a passive intruder but not in the same protocol under an active intruder, and vice versa. In case of basic-term actions, group anonymity in a protocol under an active intruder implies group anonymity in the same protocol under a passive intruder. A broad spectrum of relationships between group anonymity for various types of actions is developed, as well as relationships between group anonymity, minimal anonymity, and role interchangeability. Finally, the decidability and complexity status of the decision problems induced by these concepts is completely discussed. Thus, it is shown that group anonymity and role interchangeability are undecidable in unrestricted protocols. Group anonymity is complete for NEXPTIME when it is restricted to basic-term actions and bounded security protocols, and it is complete for NP when it is restricted to basic-term actions and 1-session bounded security protocols.**

***Index Terms*—Security protocol, anonymity, decision problem, epistemic logic**

## I. Introduction

Over the last two decades there has been a growing interest in methods for anonymous communication and in developing techniques for reasoning about information hiding properties in security protocols [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]. This is mainly due to the applications of the anonymous communication to various fields such as e-voting, e-commerce, e-mail, e-cash and so on.

Information hiding embraces many forms such as *anonymity*, *unlinkability*, *indistinguishability*, *role interchangeability*, *undetectability*, *unobservability*, and *identity management*. In an effort to standardize the terminology on information hiding, Pfitzmann and Hansen have written and maintained since 2000 a consolidated proposal of terminology on information hiding properties [23].

Anonymity is a prominent information hiding property to which a lot of research has been dedicated. Using a multi-agent system and epistemic logic based framework, Halpen and O'Neill [11] classified the anonymity into:

- minimal anonymity – an action performed by an agent is not always seen by an observer;
- group anonymity – an agent who performed some action is not "identifiable", by an observer, within a set of agents. Sender and recipient anonymity in [23] are instances of group anonymity.

*Contribution:* In two earlier papers [24], [26], we have investigated the minimal anonymity in security protocols. In this paper we do the same for group anonymity. Using the framework developed in [24], [26], we define group anonymity for security protocols. As we are using six types of send actions and six types of receive actions, our approach covers a large spectrum of group anonymity concepts met in the literature on information hiding properties. Several basic relationships between all these concepts of group anonymity are established in the paper.

Anonymity is a property that depends on the observer of the protocol execution. We consider two types of observers: honest agents and the intruder. An honest agent is a local observer of the protocol execution; he can only record information about the actions which involve him. The intruder is a global observer of the protocol execution who can record all actions in the protocol execution. Honest agents and the intruder as observers may have incomparable deductive powers due to the fact that honest agents may know secret information unknown to the intruder, while the intruder may have more information about the actions performed in the protocol. This is clearly reflected in the results obtained in the paper.

As we prove in our paper, group anonymity highly depends on the intruder type: passive or active. Thus, we show that there are group anonymous actions in protocols under passive intruders which are not group anonymous if the intruder is active. That is, an active intruder may destroy the group anonymity property of an action. More interestingly is that the converse holds true as well: there are group anonymous actions in protocols under active intruders which are not group anonymous if the intruder is passive. That is, an active intruder may induce some degree of anonymity in security protocols.

The relationships between minimal anonymity, group anonymity, and role interchangeability are also discussed. Thus, we show that any group anonymous exclusive action is minimally anonymous (in the same security protocol), and any role interchangeable observable action is group anonymous (in the same security protocol).

As decision problem, group anonymity is proven undecidable in unrestricted security protocols. If we restrict group anonymity to basic-term actions and bounded security protocols, then it is complete for NEXPTIME. If one more restriction is added by requiring just 1-session protocol executions, then group anonymity becomes complete for NP.

*Related work:* The seminal work that marked the development of a formal study of anonymity-related properties is that of David Chaum [1], [2], [3] who proposed a method by which an agent can send a message to some other agent without revealing his identity. Since then, several formalisms for anonymity have been proposed. Thus, [4] proposes a formalization of anonymity in the CSP framework. [6] and [11] focus on anonymity in security protocols and multi-agent systems, respectively, by using an epistemic logic framework. These two papers have greatly influenced the research on anonymity based on an epistemic logic formalism [12], [15], [20], [24], [25], [26]. The roots of our paper can be traced back to these two papers too. While [6], [11] have just offered the basis for an epistemic logic based approach to anonymity, [24], [26] have proposed a rich inference system to reason about anonymity in security protocols. Moreover, based on this, many results on minimal anonymity, such as decidability and complexity results, have been developed in [24], [26]. Our paper continues along the same line, by offering a large spectrum of results on group anonymity in security protocols.

A rather different but very interesting approach to anonymity was proposed by Hughes and Shmatikov [8] by using *function views*. The *cryptographic protocol logic* (CPL) proposed in [27] came as an ambitious general framework for formalizing a very large class of security properties. While CPL seems very expressive, the model checking problem for it is undecidable and not too much about decidable fragments and proof systems for the core CPL is known.

*Structure of the paper:* The paper is organized into seven sections. The formal model we use in this paper for security protocols is recalled in Section 2. The group anonymity concepts studied in this paper are introduced in Section 3, while Section 4 presents basic properties of these concepts. Section 5 is dedicated to the study of the decidability status of the decision problems induced by our group anonymity concepts, while Section 6 discusses complexity issues. We conclude in Section 7.

Due to the space limitation, the proofs of the results were not included in the final version of the paper.

## II. Modeling Security Protocols

The formalism used in this paper to model security protocols is precisely the one in [28], [29], [30], [24], [26]. Therefore, we will only recall its basic notations and terminology (for more details the reader is referred to the papers cited above).

A *security protocol signature* is a 3-tuple $\mathcal{S} = (\mathcal{A}, \mathcal{K}, \mathcal{N})$ consisting of a finite set $\mathcal{A}$ of *agent names* (or shortly, *agents*) and two at most countable sets $\mathcal{K}$ and $\mathcal{N}$ of *keys* and *nonces* (numbers once used), respectively. It is assumed that:

- $\mathcal{A}$ contains a special element denoted by $I$ and called the *intruder*. All the other elements are called *honest agents* and $Ho$ denotes their set;
- $\mathcal{K} = \mathcal{K}_0 \cup \mathcal{K}_1$, where $\mathcal{K}_0$ is the set of *short-term keys* and $\mathcal{K}_1$ is a finite set of *long-term keys*. The elements of $\mathcal{K}_1$ are of the form $K_A^e$ ($A$'s public key), or $K_A^d$ ($A$'s private key), or $K_{AB}$ (shared key by $A$ and $B$), where $A$ and $B$ are distinct agents;
- some honest agents $A$ may be provided from the beginning with some *secret information* $Secret_A \subseteq \mathcal{K}_0 \cup \mathcal{N}$, not known to the intruder. $Secret_A$ does not contain long-term keys because they will never be communicated by agents during the runs;
- the intruder is provided from the beginning with a set of nonces $\mathcal{N}_I \subseteq \mathcal{N}$ and a set of short-term keys $\mathcal{K}_{0,I} \subseteq \mathcal{K}_0$. It is assumed that no elements in $\mathcal{N}_I \cup \mathcal{K}_{0,I}$ can be generated by honest agents.

The set of *basic terms* is $\mathcal{T}_0 = \mathcal{A} \cup \mathcal{K} \cup \mathcal{N}$. The set $\mathcal{T}$ of *terms* is defined inductively: every basic term is a term; if $t_1$ and $t_2$ are terms, then $(t_1, t_2)$ is a term (meaning concatenation of messages); if $t$ is a term and $K$ is a key, then $\{t\}_K$ is a term (meaning $t$ encrypted by $K$). We extend the construct $(t_1, t_2)$ to $(t_1, \ldots, t_n)$ as usual by letting $(t_1, \ldots, t_n) = ((t_1, \ldots, t_{n-1}), t_n)$, for all $n \geq 3$. Sometimes, parenthesis will be omitted. Given a term $t$, $Sub(t)$ is the set of all *sub-terms* of $t$ (defined as usual). This notation is extended to sets of terms by union.

The length of a term is defined as usual, by taking into consideration that pairing and encryption are operations. Thus, $|t| = 1$ for any $t \in \mathcal{T}_0$, $|(t_1, t_2)| = |t_1| + |t_2| + 1$, for any terms $t_1$ and $t_2$, and $|\{t\}_K| = |t| + 2$, for any term $t$ and key $K$.

The *perfect encryption assumption* we adopt [31] states that a message encrypted with a key $K$ can be decrypted only by an agent who knows the corresponding inverse $K^{-1}$ of $K$, and the only way to compute $\{t\}_K$ is by encrypting $t$ with $K$.

There are two types of actions, send and receive. A *send action* is of the form $A!B : (M)t$, and a *receive action* is of the form $A?B : t$. In both cases, $A$ is assumed an honest agent who *performs the action*, $A \neq B$, $t \in \mathcal{T}$ is the *term of the action*, and $M \subseteq Sub(t) \cap (\mathcal{N} \cup \mathcal{K}_0)$ is the *set of new terms of the action*. $M(a)$ denotes $M$, if $a = A!B : (M)t$, and the empty set, if $a = A?B : t$; $t(a)$ stands for the term of $a$. When $M = \emptyset$ we will simply write $A!B : t$. For a sequence of actions $w = a_1 \cdots a_l$ and an agent $A$, define the *restriction of $w$ to $A$*, denoted $w|_A$, as being the sequence obtained from $w$ by removing all actions not performed by $A$. The notations $M(a)$ and $t(a)$ are extended to sequences of actions by union.

A *security protocol* (or simply, *protocol*) is a triple $\mathcal{P} = (\mathcal{S}, \mathcal{C}, w)$, where $\mathcal{S}$ is a security protocol signature, $\mathcal{C}$ is a subset of $\mathcal{T}_0$, called the set of *constants* of $\mathcal{P}$, and $w$ is a non-empty sequence of actions, called the *body* of the protocol, such that no action in $w$ contains the intruder. Constants are publicly known elements in the protocol that cannot be re-instantiated (as it will be explained below). As usual, $\mathcal{C}$ does not include private keys, elements in $Secret_A$ for any honest agent $A$, or elements in $\mathcal{N}_I$, $\mathcal{K}_{0,I}$ and $M(w)$. Any non-empty sequence $w|_A$, where $A$ is an agent, is called a *role* of the protocol. A role specifies the actions a participant should perform in a protocol, and the order of these actions.

An example of a security protocol is given in Figure 1. In this example, the server $S$ wants to get an opinion from its clients regarding the network services provided by it (the clients are $A$ and $B$ in our example). Therefore, $S$ generates a fresh short term key $K$ and sends it to $A$ and $B$. These agents compose some messages (their opinions) $t$ and $t'$ and send them, encrypted by $K$, to $H$. When $H$ has collected all the messages (opinions), it forwards them to $S$.

$$\begin{array}{lcl}
S\,!\,A & : & (\{K\})\{K,H\}_{K_{SA}} \\
A\,?\,S & : & \{K,H\}_{K_{SA}} \\
S\,!\,B & : & \{K,H\}_{K_{SB}} \\
B\,?\,S & : & \{K,H\}_{K_{SB}} \\
A\,!\,H & : & \{\{t\}_K,S\}_{K_{AH}} \\
H\,?\,A & : & \{\{t\}_K,S\}_{K_{AH}} \\
B\,!\,H & : & \{\{t'\}_K,S\}_{K_{BH}} \\
H\,?\,B & : & \{\{t'\}_K,S\}_{K_{BH}} \\
H\,!\,S & : & \{\{t\}_K,\{t'\}_K\}_{K_{SH}} \\
S\,?\,H & : & \{\{t\}_K,\{t'\}_K\}_{K_{SH}}
\end{array}$$

Fig. 1. A running example

Instantiations of a protocol are given by *substitutions*, which are functions $\sigma$ that map agents to agents, nonces to arbitrary terms, short-term keys to short-term keys, and long-term keys to long-term keys. Moreover, for long-term keys, $\sigma$ should satisfy $\sigma(K_A^e) = K_{\sigma(A)}^e$, $\sigma(K_A^d) = K_{\sigma(A)}^d$, and $\sigma(K_{AB}) = K_{\sigma(A)\sigma(B)}$, for any distinct agents $A$ and $B$. Substitutions are homomorphically extended to terms, actions, and sequences of actions. A substitution $\sigma$ is called *suitable for an action* $a = AxB : y$ if $\sigma(A)$ is an honest agent, $\sigma(A) \neq \sigma(B)$, and $\sigma$ maps distinct nonces from $M(a)$ into distinct nonces, distinct keys into distinct keys, and it has disjoint ranges for $M(a)$ and $Sub(t(a)) - M(a)$. $\sigma$ is *suitable for a sequence of actions* if it is suitable for each action in the sequence, and $\sigma$ is *suitable for a subset* $C \subseteq \mathcal{T}_0$ if it is the identity on $C$.

An *event* of a protocol $\mathcal{P} = (\mathcal{S}, \mathcal{C}, w)$ is any triple $e_i = (u, \sigma, i)$, where $u = a_1 \cdots a_l$ is a role of $\mathcal{P}$, $\sigma$ is a substitution suitable for $u$ and $\mathcal{C}$, and $1 \leq i \leq l$. $\sigma(a_i)$ is the *action of the event* $e_i$. As usual, $act(e_i)$ ($t(e_i)$, $M(e_i)$) stands for the the action of $e_i$ (term of $e_i$, set of new terms of $e_i$). The *local precedence relation* on events is defined by $(u, \sigma, i) \rightarrow (u', \sigma', i')$ if and only if $u' = u$, $\sigma' = \sigma$, and $i' = i + 1$, provided that $i < |u|$. $\xrightarrow{+}$ is the transitive closure of $\rightarrow$. Given an event $e$, ${}^\bullet e$ stands for the *set of all local predecessors of* $e$, i.e., ${}^\bullet e = \{e' | e' \xrightarrow{+} e\}$.

Given $X$ a set of terms, $analz(X)$ stands for the least set which includes $X$, contains $t_1$ and $t_2$ whenever it contains $(t_1, t_2)$, and contains $t$ whenever it contains $\{\{t\}_K\}_{K^{-1}}$ or $\{t\}_K$ and $K^{-1}$. By $synth(X)$ we denote the least set which includes $X$, contains $(t_1, t_2)$, for any terms $t_1, t_2 \in synth(X)$, and contains $\{t\}_K$, for any term $t$ and key $K$ in $synth(X)$. Moreover, $\overline{X}$ stands for $synth(analz(X))$.

A *state* of a protocol $\mathcal{P}$ is an indexed set $s = (s_A | A \in \mathcal{A})$, where $s_A$ is $A$'s (local) state, for any agent $A$. The traditional approach to security protocols defines agent states as sets of messages (all messages sent and received by the agent during some computation). This approach is quite sufficient if one wants to reason about confidentiality [28], [29], [30]. However, this is not enough to reason about anonymity properties, where more information about the actions performed by agents in the protocol are needed. One way to solve this is to add *facts* to agent states [6], [24], [26]. A fact is a sentence of the form $P(t_1, \ldots, t_i)$, where $P$ is a predicate symbol and $t_1, \ldots, t_i$ are message terms (facts beginning by the same predicate symbol $P$ will also be called *$P$-facts*). Using the approach in [24], [26], we will use six classes of facts which are illustrated on the protocol in Figure 1:

1) *sent*-facts. Each agent $X$ who sends a message $t$ to some agent $Y$ records a fact $sent(X, t, Y)$. For instance, when the first action of the protocol in Figure 1 will be performed, $S$ records $sent(S, \{K,H\}_{K_{SA}}, A)$;
2) *rec*-facts. Two cases are to be considered here:
   a) *passive intruder*. If an action $X\,?\,Y : t$ was performed by $X$, then $X$ may safely record a fact $rec(X, t, Y)$ because he knows that the message he received is from $Y$;
   b) *active intruder*. If an action $X\,?\,Y : t$ was performed by $X$, then $X$ might not be sure whether $t$ comes from $Y$ or from the intruder. In such a case $X$ records a fact $rec(X, t, (Y, I))$ which tells him that $t$ may be from $Y$ or from $I$.

   If the second action in the protocol in Figure 1 has been performed in some computation, and the intruder was active, then $A$ records the fact $rec(A, \{K,H\}_{K_{SA}}, (S, I))$;
3) *shared_key*-facts. In the first action of the protocol, the agent $S$ generates a short-term key $K$ and sends it to $A$. Therefore, $K$ acts as a short term *shared key* between $S$ and $A$. We use $shared\_key(Z, X, Y, K)$ to mean that $Z$ randomly generated a short term key $K$ to be used by $X$ and $Y$ as a shared-key. In our protocol in Figure 1, $shared\_key(S, S, A, K)$ is the fact to be recorded by $A$ when the first action of the protocol is performed;
4) *gen*-facts. The message in the first action of the protocol in our running example is *generated by S for A* because it is encrypted by the long term shared key $K_{SA}$; denote this by $gen(S, \{K,H\}_{K_{SA}}, A)$ and record it in $S$'s state. Similarly, $A$ will record the fact $gen(A, \{t\}_K, S)$ in his state when fifth action of the protocol is performed;
5) *auth*-facts. A message $t$ encrypted by $X$'s private key $K_X^d$ is authenticated by $X$. The fact $auth(X, (t, \{t\}_{K_X^d}))$ denotes this;
6) *hop*-facts. These are facts of the form $hop(A, C, B, t)$ whose meaning is that $B$ can only received $t$ from $A$ via $C$ (examples of hope facts can be found in [26]).

According to our discussion above, an agent $A$ state is a pair $s_A = (s_{A,m}, s_{A,f})$, where $s_{A,m}$ is a set of messages and $s_{A,f}$ is a set of facts. Intuitively, $s_{A,m}$ represents the set of all messages the agent $A$ sent or received in some computation from the initial state to the state $s_A$, and $s_{A,f}$ represents the set of facts which give information about the actions the agent $A$ performed in that computation.

The protocol computation rule in [28], [29], [30] has to be changed accordingly [24], [26]. Given two states $s$ and $s'$ and an action $a$, we write $s[a\rangle s'$ if and only if:

1) if $a$ is of the form $A!B : (M)t$, then:
   a) $t \in \overline{s_{A,m} \cup M}$ and $M \cap Sub(s) = \emptyset$;
   b) $s'_{A,m} = s_{A,m} \cup M \cup \{t\}$, $s'_{I,m} = s_{I,m} \cup \{t\}$, and $s'_{C,m} = s_{C,m}$ for any $C \in \mathcal{A} - \{A, I\}$;
   c) the facts in $s'$ are obtained as follows:
      i) add $sent(A, t, B)$ to $s_{A,f}$ and $s_{I,f}$;
      ii) if some term $t_1 = \{t'\}_{K_{AC}}$ or $t_1 = \{t'\}_{K^e_C}$ or $t_1 = \{t'\}_K$ has been built by $A$ in order to build $t$, where $K$ is a short-term shared key by $A$ and some agent $C$ and $A$ owns this key, then add $gen(A, t_1, C)$ to $s_{A,f}$;
      iii) if some term $t_1 = (t', \{t'\}_{K^d_A})$ has been built by $A$ in order to build $t$, then add $auth(A, t_1)$ to $s_{A,f}$;
      iv) if some short-term key $K$ has been generated by $A$ to be used as a shared key by two agents $C$ and $D$, and $K$ is a part of $t$, then add $shared\_key(A, C, D, K)$ to $s_{A,f}$;
      v) $s'_{C,f} = s_{C,f}$, for any $C \in \mathcal{A} - \{A, I\}$;
2) if $a$ is of the form $A?B : t$, then:
   a) $t \in \overline{s_{I,m}}$;
   b) $s'_{A,m} = s_{A,m} \cup \{t\}$ and $s'_{C,m} = s_{C,m}$, for all $C \in \mathcal{A} - \{A\}$;
   c) the facts in $s'$ are obtained as follows:
      i) add $rec(A, t, (B, I))$ to $s_{A,f}$ and $s_{I,f}$;
      ii) if $A$ received a key $K$ as part of $t$ and he knows that $K$ was generated by some agent $C$ to be shared by $A$ with another agent $D$, then add $shared\_key(C, A, D, K)$ to $s_{A,f}$;
      iii) if $A$ received a message $t'$ as part of $t$ and he knows that this message comes from some agent $C$ via another agent $D$, then add $hop(C, D, A, t')$ to $s_{A,f}$;
      iv) $s'_{C,f} = s_{C,f}$, for any $C \in \mathcal{A} - \{A, I\}$.

In the case of a passive intruder (2a) should be "$t \in \overline{s_{B,m}}$" and (2ci) above should be "add $rec(A, t, B)$ to $s_{A,f}$ and $s_{I,f}$". If we remove (1c) and (2c) from the computation rule above, we obtain the standard computation rule in [28], [29], [30].

At each point in the evolution of a protocol, each agent may derive new facts from the facts he owns at that point. The derivation process is guided by deduction rules. In order to present these rules we need first two basic concepts. A message $t$ is called *decomposable* [24], [26] over an agent state $s = (s_m, s_f)$ if $t \in \mathcal{T}_0$, or $t = (t_1, t_2)$ for some messages $t_1$ and $t_2$, or $t = \{t'\}_K$ for some message $t'$ and key $K$ with $K^{-1} \in analz(s_m)$, or $gen(A, t, B) \in s_f$ for some honest agents $A$ and $B$ ("$gen(A, t, B)$" covers the case when $A$ generates $t$ for $B$ by encrypting some message by $B$'s public key. $A$ does not know $B$'s corresponding private key but knows how he built $t$ and, from this point of view, we may say that $t$ is decomposable). The function $trace(t, s)$ [24], [26], where $t$ is a message and $s = (s_m, s_f)$ is an agent state, is given by:

- $trace(t, s) = \{t\}$, if $t \in \mathcal{T}_0$;
- $trace(t, s) = \{t\} \cup trace(t_1, s) \cup trace(t_2, s)$, if $t = (t_1, t_2)$ for some terms $t_1$ and $t_2$;
- $trace(t, s) = \{t\}$, if $t$ is not decomposable over $s$;
- $trace(t, s) = \{t\} \cup trace(t', s)$, if $t = \{t'\}_K$ is an encrypted but decomposable message over $s$.

The deduction rules (Table I) we use are those from [24] with slight modifications [26] (a rule with one or two indexes specifies the current state where the rule should be applied; for instance, $(RShR)_{A,C}$ means that the rule $(RShR)$ should be applied in $A$'s or $C$'s current state. A rule with no indexes means that the rule can be applied in any state). We discuss just one of the rules in Table I, namely $(RShR)$ (for a complete discussion about them, the reader is referred to [26]). According to this rule, if $A$ received a message $\{t\}_K$ encrypted by a short-term key distributed by $C$ to him and to $B$, then surely $B$ received the key from $C$.

Given a set $M$ of messages and a set $F$ of facts, denote by $Analz(M, F)$ the set of all facts that can be inferred from $F$ and $M$. If $s = (s_m, s_f)$ is an agent state, then $Analz(s)$ stands for $Analz(s_m, s_f)$.

## III. Defining Anonymity

Anonymity in a security protocol is a property that has to be defined w.r.t. an observer of the protocol execution. In our approach, the observer is either an honest agent (as in [11], [24], [26]) or the intruder (passive [6], [10], or active [24], [26]). Honest agents as observers are limited to observing some of the actions performed by the agents who interact with him, while passive intruders as observers are capable to observe the entire protocol execution. On the other side, honest agents may have more deductive power than passive intruders because they may know secret keys unknown to intruders. Therefore, from the anonymity point of view, honest agents and passive intruder as observers have incomparable powers.

While [6], [10] have considered only passive intruders as observers, in [24], [26] active intruders were taken into consideration too. This is because an action may be anonymous w.r.t. a passive intruder but not w.r.t. an active one, and vice-versa (see [26] and Theorem 5 in this paper).

Observers draw conclusions about protocol executions by analyzing their current states. If two current states are "equivalent", then the conclusions should be equivalent. We formalize this as follows. Given a pair of agent states $(s, s')$ define the binary relation $\sim_{s,s'}$ on message terms by [24], [26]:

- $t \sim_{s,s'} t$, for any $t \in \mathcal{T}_0$;
- $t \sim_{s,s'} t'$, for any term $t$ undecomposable over $s$ and any term $t'$ undecomposable over $s'$;
- $(t_1, t_2) \sim_{s,s'} (t'_1, t'_2)$, for any terms $t_1$, $t_2$, $t'_1$, and $t'_2$ with $t_1 \sim_{s,s'} t'_1$ and $t_2 \sim_{s,s'} t'_2$;
- $\{t\}_K \sim_{s,s'} \{t'\}_K$, for any terms $t$ and $t'$ and any key $K$ with $t \sim_{s,s'} t'$ and $K^{-1} \in analz(s_m) \cap analz(s'_m)$.

Extend $\sim_{s,s'}$ to facts by $P(t_1, \ldots, t_i) \sim_{s,s'} P(t'_1, \ldots, t'_i)$ if $t_j \sim_{s,s'} t'_j$ for any $1 \leq j \leq i$.

Two agent states $s = (s_m, s_f)$ and $s' = (s'_m, s'_f)$ are called *observationally equivalent* [24], [26], denoted $s \sim s'$, if:

TABLE I
DEDUCTION RULES

| | | |
|---|---|---|
| $(S1)\ \dfrac{sent(A,t,B)}{sent(A,t), sent(A,B), sent(t,B)}$ | $(R1)\ \dfrac{rec(A,t,x)}{rec(A,t), rec(A,x), rec(t,x)}$ | $(RS)\ \dfrac{rec(A,t,B)}{sent(B,t,A)}$ |
| $(S2)\ \dfrac{sent(A,B)}{sent(A)}$ | $(R2)\ \dfrac{rec(A,x)}{rec(A)}$ | $(RGS)_A\ \dfrac{rec(A,t),\ gen(B,t,A)}{sent(B,t,A)}$ |
| $(S3)\ \dfrac{sent(A,t)}{sent(A), sent(t)}$ | $(R3)\ \dfrac{rec(A,t)}{rec(A), rec(t)}$ | $(RAS)\ \dfrac{rec(t),\ auth(A,t)}{sent(A,t)}$ |
| $(S4)\ \dfrac{sent(t,B)}{sent(t)}$ | $(R4)\ \dfrac{rec(t,x)}{rec(t)}$ | $(SGS)_{A,B}\ \dfrac{sent(A,t),\ gen(A,t,B)}{sent(A,t,B)}$ |
| $(S5)\ \dfrac{sent(A,t,B),\ t' \in trace(t,s)}{sent(A,t',B)}$ | $(R5)\ \dfrac{rec(A,t,x),\ t' \in trace(t,s)}{rec(A,t',x)}$ | $(RA)\ \dfrac{rec(t, \{t\}_{K_A^d})}{auth(A,(t,\{t\}_{K_A^d}))}$ |
| $(RG)_A\ \dfrac{rec(A,\{t\}_{K_{AB}}),\ \neg gen(A,\{t\}_{K_{AB}},B)}{gen(B,\{t\}_{K_{AB}},A)}$ | $(RG')_A\ \dfrac{rec(A,\{t\}_K),\ shared\_key(C,A,B,K),\ \neg gen(A,\{t\}_K,B)}{gen(B,\{t\}_K,A)}$ | |
| $(RGR)_{A,B}\ \dfrac{rec(A,t,(B,I)),\ gen(B,t,A)}{rec(A,t,B)}$ | $(SGR)_{B,C}\ \dfrac{sent(A,t,B),\ gen(C,t,B), hop(C,A,B,t)}{rec(A,t,C)}$ | |
| | $(RShR)_{A,C}\ \dfrac{rec(A,\{t\}_K),\ shared\_key(C,A,B,K),\ \neg gen(A,\{t\}_K,B)}{rec(B,K,C)}$ | |

- $analz(s_m) \cap \mathcal{T}_0 = analz(s'_m) \cap \mathcal{T}_0$;
- $(\forall \varphi \in Analz(s))(\exists \varphi' \in Analz(s'))(\varphi \sim_{s,s'} \varphi')$;
- $(\forall \varphi' \in Analz(s'))(\exists \varphi \in Analz(s))(\varphi' \sim_{s',s} \varphi)$.

That is, $s$ and $s'$ are observationally equivalent if the agent can derive the same meaningful information from any of these two states. In other words, these two states are *indistinguishable*.

Two protocol states $s$ and $s'$ are *observationally equivalent w.r.t. an agent* $A$, denoted $s \sim^A s'$, if $s_A \sim s'_A$.

It was shown in [24], [26] that the observational equivalence on agent states is an equivalence relation decidable in $\mathcal{O}(f^4 l^4)$ time complexity, where $f$ is the maximum number of facts and $l$ is the maximum length of the messages in the states.

We use a fragment of the epistemic logic in [32], [11] to reason about anonymity. Its syntax is

$$\varphi ::= p \,|\, \varphi \wedge \varphi \,|\, \neg\varphi \,|\, \mathtt{K}_A\varphi$$

where $p$ ranges over a countable set $\Phi$ of atomic propositions, $A$ ranges over a non-empty finite set $\mathcal{A}$ of agent names, and $\varphi$ in $\mathtt{K}_A\varphi$ does not contain any $\mathtt{K}$ operator. Denote by $\mathcal{L}(\Phi, \mathcal{A})$ the set of all formulas defined as above. As usual we use $\mathtt{P}_A\varphi$ as an abbreviation for $\neg\mathtt{K}_A\neg\varphi$.

Let $\mathcal{P}$ be a security protocol. The *truth value of a formula* $\varphi \in \mathcal{L}(\Phi, \mathcal{A})$ *in* $\mathcal{P}$ is defined as follows:

- $\mathcal{P} \models \varphi$ iff $(\mathcal{P}, s) \models \varphi$, for any reachable state $s$ in $\mathcal{P}$;
- $(\mathcal{P}, s) \models p$ iff $(\mathcal{P}, s_A) \models p$, for some agent $A \in \mathcal{A} - \{I\}$;
- $(\mathcal{P}, s) \models \neg\varphi$ iff $(\mathcal{P}, s) \not\models \varphi$;
- $(\mathcal{P}, s) \models \varphi \wedge \psi$ iff $(\mathcal{P}, s) \models \varphi$ and $(\mathcal{P}, s) \models \psi$;
- $(\mathcal{P}, s) \models \mathtt{K}_A\varphi$ iff $(\mathcal{P}, s'_A) \models \varphi$, for any reachable state $s'$ with $s' \sim^A s$;
- for any formula $\varphi$ without $\mathtt{K}$ operators and any $A \in \mathcal{A}$, $(\mathcal{P}, s_A) \models \varphi$ is defined as follows:
  - if $\varphi = p$ then $(\mathcal{P}, s_A) \models \varphi$ iff $p \in Analz(s_A)$;
  - if $\varphi = \varphi_1 \wedge \varphi_2$ then $(\mathcal{P}, s_A) \models \varphi$ iff $(\mathcal{P}, s_A) \models \varphi_1$ and $(\mathcal{P}, s_A) \models \varphi_2$;
  - if $\varphi = \neg\varphi_1$ then $(\mathcal{P}, s_A) \models \varphi$ iff $(\mathcal{P}, s_A) \not\models \varphi_1$.

We shall simply write $s \models \varphi$ ($s_A \models \varphi$) instead of $(\mathcal{P}, s) \models \varphi$ ($(\mathcal{P}, s_A) \models \varphi$), whenever $\mathcal{P}$ is understood from the context.

The formula $\mathtt{K}_A\varphi$ means "agent $A$ knows $\varphi$". It holds in a reachable state $s$ if it holds in any reachable state that is observationally equivalent to $s$ w.r.t. $A$. $\mathtt{P}_A\varphi$ means "agent $A$ thinks that $\varphi$ is possible". It holds in a state $s$ if it holds in some reachable state observationally equivalent to $s$ w.r.t. $A$.

Anonymity in security protocols will be defined for *actions* performed by agents, w.r.t. some observer. By an *action* we will understand a $sent$-fact (also called $sent$*-action*), or a $rec$-fact that does not contain terms of the form $(B, I)$ (also called $rec$*-action*). Therefore, the $sent$-actions are of the form $sent(A, t, B)$, $sent(A, t)$, $sent(A, B)$, $sent(A)$, $sent(t)$, or $sent(t, B)$, while the $rec$-actions are of the form $rec(A, t, B)$, $rec(A, t)$, $rec(A, B)$, $rec(A)$, $rec(t)$, or $rec(t, B)$. By $act$ we will denote a generic action of the one of the forms above.

Each action, except for $sent(t)$, $sent(t, B)$, $rec(t)$, and $rec(t, B)$, is performed by exactly one agent, namely, the first argument of the corresponding $sent$- or $rec$-fact. These actions are also called *mono-agent actions*. The actions $sent(t)$, $sent(t, B)$, $rec(t)$, and $rec(t, B)$ may be performed by more than one agent; they will be called *multi-agent actions*. If $act$ is a mono-agent action performed by some agent $A$, then we also write $act(A)$ just to specify the agent who performs the action. If $act$ is a multi-agent action, such as $sent(t)$, $sent(t, B)$, $rec(t)$, or $rec(t, B)$, we also write $act(t)$ just to specify the message term involved in the action.

*Definition 1:* Let $\mathcal{P}$ be a security protocol, $G$ a nonempty set of agents, $T$ a finite set of message terms, and $X$ an observer (agent) not in $G$.

1) A mono-agent action $act(A)$ of $\mathcal{P}$ is *anonymous within* $G$ *w.r.t.* $X$ if $\mathcal{P} \models \psi(act(A), G, X)$, where $\psi(act(A), G, X) = (\mathtt{P}_X act(A) \Rightarrow \bigwedge_{C \in G} \mathtt{P}_X act(C))$.
2) A multi-agent action $act(t)$ of $\mathcal{P}$ is *anonymous within* $T$ *w.r.t.* $X$ if $\mathcal{P} \models \psi(act(t), T, X)$, where $\psi(act(t), T, X) = (\mathtt{P}_X act(t) \Rightarrow \bigwedge_{t' \in T} \mathtt{P}_X act(t'))$.
3) An action $sent(A, t)$ is *role interchangeable* within $G \times T$ w.r.t. $X$ if the following property holds:
$$\mathcal{P} \models \mathtt{P}_X sent(A, t) \;\Rightarrow\; \bigwedge_{C \in G,\, t' \in T} (\mathtt{P}_X sent(C, t') \;\Rightarrow\; \mathtt{P}_X (sent(A, t') \wedge sent(C, t))).$$

A few explanations about these concepts are in order:

- Anonymity of $act(A)$ within $G$ w.r.t. $X$ in $\mathcal{P}$ means that, whenever $X$ thinks that $act(A)$ is possible at some state $s$ then, for any $C \in G$, $X$ thinks that $act(C)$ is possible at some state $s'$ observationally equivalent to $s$.
- Role interchangeability simply means that, from the observer's point of view, two actions may be interchanged between two distinct agents.

Sender (receiver) anonymity within a set of senders (receivers), as defined in [23], is a special case of anonymity of a mono-agent $sent$-action ($rec$-action) within a set of agents.

The anonymity concepts introduced in Definition 1(1)(2) are also called *group anonymity* concepts, and the sets $G$ and $T$ in these definitions are called *anonymity sets*. Thus, group anonymity says that the agent who performs an action or the message which purports an action is not identifiable within a set (group) of agents or messages, respectively.

We want to emphasize that the anonymity of an action which contains messages, such as $sent(A, t)$, should not be confused with the secrecy of $t$. The anonymity of $sent(A, t)$ within $G$ w.r.t. $X$ means that $X$ is not sure whether $A$ sent the message $t$ because he was able to deduce that any member of $G$ sent at some point in the protocol the message $t$ (although $X$ might knew the message $t$).

*Example 2:* Figure 2 presents a sequence of inferences in the state $s$ of the protocol in Figure 1, obtained by playing all actions of the protocol (the right hand side column indicates the inference process). It is not difficult to see that:

| | | |
|---|---|---|
| 1. | $shared\_key(S, S, A, K)$ | $\in s_S$ |
| 2. | $shared\_key(S, S, B, K)$ | $\in s_S$ |
| 3. | $rec(S, \{\{t\}_K, \{t'\}_K\}_{K_{SH}}, (H, I))$ | $\in s_S$ |
| 4. | $rec(S, \{\{t\}_K, \{t'\}_K\}_{K_{SH}})$ | $3, R1$ |
| 5. | $rec(S, \{t\}_K)$ | $4, R5$ |
| 6. | $rec(S, \{t'\}_K)$ | $4, R5$ |
| 7. | $\neg gen(S, \{t\}_K, A)$ | $\in s_S$ |
| 8. | $\neg gen(S, \{t\}_K, B)$ | $\in s_S$ |
| 9. | $\neg gen(S, \{t'\}_K, A)$ | $\in s_S$ |
| 10. | $\neg gen(S, \{t'\}_K, B)$ | $\in s_S$ |
| 11. | $gen(A, \{t\}_K, S)$ | $5, 1, 7, (RG')_S$ |
| 12. | $gen(B, \{t\}_K, S)$ | $5, 2, 8, (RG')_S$ |
| 13. | $gen(A, \{t'\}_K, S)$ | $6, 1, 9, (RG')_S$ |
| 14. | $gen(B, \{t'\}_K, S)$ | $6, 2, 10, (RG')_S$ |
| 15. | $sent(A, \{t\}_K, S)$ | $5, 11, (RGS)_S$ |
| 16. | $sent(B, \{t\}_K, S)$ | $5, 12, (RGS)_S$ |
| 17. | $sent(A, \{t'\}_K, S)$ | $6, 13, (RGS)_S$ |
| 18. | $sent(B, \{t'\}_K, S)$ | $6, 14, (RGS)_S$ |

Fig. 2. Examples of inferences by the rules in Table I

- $sent(A, t)$ is anonymous within $\{A, B\}$ w.r.t. $S$ (that is, $S$ cannot clearly identify whether $A$ or $B$ sent $t$);
- $sent(A, t)$ is anonymous within $\{t, t'\}$ w.r.t. $S$ (that is, $S$ cannot clearly identify whether $A$ sent $t$ or $t'$);
- $sent(A, t)$ is role interchangeable within $\{A, B\} \times \{t, t'\}$ w.r.t. $S$ (that is, from $S$'s point of view, $A$ could have send $t$ and $B$ could have send $t'$, or vice versa).

## IV. Relating anonymity concepts

### A. Basic properties of group anonymity

An action $act$ of a security protocol is called a *basic-term action* if all terms in the action are basic terms. For instance, $sent(A, N_A, B)$, where $N_A$ is a nonce, is a basic-term action, whereas the action $sent(A, \{N_A\}_K, B)$ is not. From definitions we obtain:

*Lemma 3: For any basic-term action act, any agent* $X$*, and any protocol states* $s$ *and* $s'$*, the following property holds: if* $s' \sim^X s$ *then* $s'_X \models act$ *if and only if* $s_X \models act$*.*

*Proposition 4:* A basic-term action $act(x)$ is anonymous within a group $G$ of basic terms w.r.t. $X$ in a protocol $\mathcal{P}$ if and only if, for any reachable state $s$ in $\mathcal{P}$, $s_X \models act(x)$ implies $(\forall y \in G)(s_X \models act(y))$.

**Proof.** Assume that $act(x)$ is anonymous within a group $G$ w.r.t. $X$ in $\mathcal{P}$, and let $s$ be a reachable state in $\mathcal{P}$ such that $s_X \models act(x)$.

The anonymity of $act(x)$ within $G$ leads to the fact that for any $y \in G$ there exists a reachable state $s'$, observationally equivalent to $s$ w.r.t. $X$, such that $s'_X \models act(y)$. Lemma 3 leads then to $s_X \models act(y)$. As a conclusion, we obtain

$$s_X \models act(x) \;\Rightarrow\; (\forall y \in G)(s_X \models act(y))$$

The converse is obtained in a similar way. ■

Anonymity highly depends on the intruder type, passive or active. This is shown by Theorem 5 below: there are group anonymous actions in protocols under passive intruders which are not group anonymous if the intruder is active, and vice versa, there are group anonymous actions in protocols under active intruders which are not group anonymous if the intruder is passive.

*Theorem 5:*

1) There are protocols $\mathcal{P}$, actions $act(x)$, groups $G$ of agents or message terms, and observers $X$ such that $act(x)$ is anonymous within $G$ w.r.t. $X$ in $\mathcal{P}$ under a passive intruder, but $act(x)$ is not anonymous within $G$ w.r.t. $X$ in $\mathcal{P}$ under an active intruder.
2) There are protocols $\mathcal{P}$, actions $act(x)$, groups $G$ of agents or message terms, and observers $X$ such that $act(x)$ is anonymous within $G$ w.r.t. $X$ in $\mathcal{P}$ under an active intruder, but $act(x)$ is not anonymous within $G$ w.r.t. $X$ in $\mathcal{P}$ under a passive intruder.
3) For any protocol $\mathcal{P}$, basic-term action $act(x)$, group $G$ of agents or basic terms, and observer $X$, if $act(x)$ is anonymous within $G$ w.r.t. $X$ in $\mathcal{P}$ under an active intruder, then $act(x)$ is anonymous within $G$ w.r.t. $X$ in $\mathcal{P}$ under a passive intruder.

The following results relate the group anonymity concepts for various actions.

*Theorem 6:* Let $\mathcal{P}$ be a security protocol with the property that for any agents $A, X \in \mathcal{A} - \{I\}$, any message $t$, and any reachable state $s$, if $s_X \models (rec(t) \wedge auth(A,t))$ then there exists $B \in \mathcal{A} - \{A, I\}$ such that $s_X \models sent(A,t,B)$. Then, the following properties hold in $\mathcal{P}$ ($G$ is a set of agents, $T$ is a set of messages, and $X$ is an observer):

1) If $\bigwedge_{B\in\mathcal{A}-\{A,I\}} \psi(sent(A,t,B),G,X)$ holds in $\mathcal{P}$, then $\psi(sent(A,t),G,X)$ holds in $\mathcal{P}$;
2) If $\bigwedge_{t\in\mathcal{T}} \psi(sent(A,t,B),G,X)$ holds in $\mathcal{P}$, then $\psi(sent(A,B),G,X)$ holds in $\mathcal{P}$;
3) If $\bigwedge_{B\in\mathcal{A}-\{A,I\}} \psi(sent(A,B),G,X)$ holds in $\mathcal{P}$, then $\psi(sent(A),G,X)$ holds in $\mathcal{P}$;
4) If $\bigwedge_{t\in\mathcal{T}} \psi(sent(A,t),G,X)$ holds in $\mathcal{P}$, then $\psi(sent(A),G,X)$ holds in $\mathcal{P}$;
5) If $\bigwedge_{B\in\mathcal{A}-\{I\}} \psi(sent(t,B),T,X)$ holds in $\mathcal{P}$, then $\psi(sent(t),T,X)$ holds in $\mathcal{P}$.

The hypothesis in Theorem 6 is quite natural: if an agent $X$ receives a message authenticated by some agent $A$, then he draw the conclusion that $A$ sent that message to some other agent $B$.

Figure 3 pictorially represents the implications in Theorem 6. Moreover, it is not difficult to find examples of protocols where $\psi(sent(A,B),G,X)$ holds but $\psi(sent(A,t),G,X)$ does not hold, and vice versa. That is, $\psi(sent(A,B),G,X)$ and $\psi(sent(A,t),G,X)$ are incomparable.

$$\begin{tikzcd}
 & \psi(sent(A,B),G,X) \arrow[dr, "\forall B"] & \\
\psi(sent(A,t,B),G,X) \arrow[ur, "\forall t"] \arrow[dr, "\forall B"'] & & \psi(sent(A),G,X) \\
 & \psi(sent(A,t),G,X) \arrow[ur, "\forall t"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
\psi(sent(t,B),T,X) \arrow[r, "\forall B"] & \psi(sent(t),T,X)
\end{tikzcd}$$

Fig. 3. Relationships between group anonymity concepts

Under the *sender identifiability restriction*, Theorem 6 holds for $rec$-actions too.

*Definition 7:* Let $\mathcal{P}$ be a security protocol. An action $rec(A,t)$ ($rec(A)$, $rec(t)$, resp.) is *sender identifiable* if, for any $X$ and reachable state $s$ of $\mathcal{P}$ with $s_X \models rec(A,t)$ ($s_X \models rec(A)$, $s_X \models rec(t)$, resp.) there exists $B$ such that $s_X \models rec(A,t,B)$ ($s_X \models rec(A,B)$, $s_X \models rec(t,B)$, resp.).

It is obvious that $rec(A,t)$, $rec(A)$, and $rec(t)$ are all sender identifiable in any protocol $\mathcal{P}$ under a passive intruder. Similar to Theorem 6 we obtain the following result.

*Theorem 8:* The following properties hold in any security protocol $\mathcal{P}$ ($G$ is a set of agents, $T$ is a set of messages, and $X$ is an observer but not $I$):

1) If $\bigwedge_{B\in\mathcal{A}-\{A,I\}} \psi(rec(A,t,B),G,X)$ holds in $\mathcal{P}$ and $rec(A,t)$ is sender identifiable, then $\psi(rec(A,t),G,X)$ holds in $\mathcal{P}$;
2) If $\mathcal{P} \models \bigwedge_{t\in\mathcal{T}} \psi(rec(A,t,B),G,X)$ holds in $\mathcal{P}$, then $\psi(rec(A,B),G,X)$ holds in $\mathcal{P}$;
3) If $\bigwedge_{B\in\mathcal{A}-\{A,I\}} \psi(rec(A,B),G,X)$ holds in $\mathcal{P}$ and $rec(A)$ is sender identifiable, then $\psi(rec(A),G,X)$ holds in $\mathcal{P}$;
4) If $\bigwedge_{t\in\mathcal{T}} \psi(rec(A,t),G,X)$ holds in $\mathcal{P}$ and $rec(A)$ is sender identifiable, then $\psi(rec(A),G,X)$ holds in $\mathcal{P}$;
5) If $\bigwedge_{B\in\mathcal{A}-\{I\}} \psi(rec(t,B),T,X)$ holds in $\mathcal{P}$ and $rec(t)$ is sender identifiable, then $\psi(rec(t),T,X)$ holds in $\mathcal{P}$.

### B. Minimal and group anonymity

An action $act$ of a protocol $\mathcal{P}$ is *minimally anonymous w.r.t.* $X$ [24], [26] if $\mathcal{P} \models (act \Rightarrow \neg \mathbb{K}_X act)$. Using a multi-agent framework, it has been shown [11] that any *exclusive action* which is anonymous within a group of agents is also minimally anonymous (w.r.t. the same observer). The exclusiveness of an action means that no two different agents can perform the action. This result holds in our framework too.

*Definition 9:* Let $\mathcal{P}$ be a security protocol and $A$ an honest agent. An action $act(A)$ performed by $A$ is *locally exclusive* if $s_B \models \neg(act(A) \wedge act(A'))$, for any reachable state $s$ of $\mathcal{P}$, any honest agent $B$, and any honest agent $A' \neq A$.

*Proposition 10:* If a locally exclusive action $act(A)$ of a security protocol $\mathcal{P}$ is anonymous within $G$ w.r.t. an honest agent $H$ and $|G| \geq 3$, then $act(A)$ is minimally anonymous w.r.t. $H$.

If the agent $H$ in Proposition 10 is replaced by the intruder, then Proposition 10 might not hold. This is because if an action $act(A)$ is in some state of $H$, then the action still may be anonymous within some set $G$ w.r.t. $H$, but it is definitely not minimally anonymous w.r.t. $H$.

If the local exclusiveness of an action fails to hold, then the conclusion of Proposition 10 may fail too.

### C. Role interchangeability and group anonymity

As it was remarked in [12], [20] using a multi-agent framework, role interchangeability implies group anonymity, under certain conditions. We recall this result here in our security protocol framework.

*Definition 11:* Let $\mathcal{P}$ be a security protocol, $G \subseteq \mathcal{A} - \{I\}$ a set of agents, and $T \subseteq \mathcal{T}$ a finite set of message terms, and $X$ an observer. We say that an action $sent(A,t)$ is $(G \times T)$-*observable w.r.t.* $X$ if the following property holds

$$s_X \models (sent(A,t) \ \Rightarrow \ \bigwedge_{C \in G} \bigvee_{t' \in T} sent(C,t'))$$

for any reachable state $s$ in $\mathcal{P}$.

One can easily prove now the following result:

*Proposition 12:* Let $\mathcal{P}$ be a security protocol, $G \subseteq \mathcal{A} - \{I\}$ a set of agents, $T \subseteq \mathcal{T}$ a finite set of message terms, and $X$ an observer. If $X$ is not in $G$ and $sent(A,t)$ is role interchangeable within $G \times T$ and $(G \times T)$-observable w.r.t. $X$, then $sent(A,t)$ is anonymous within $G$ w.r.t. $X$.

Role interchangeability can similarly be formulated for actions like $sent(A,B)$, $rec(A,t)$, or $rec(A,B)$.

## V. Deciding Group Anonymity

In this section we establish several undecidability results for the anonymity concepts defined so far. The proofs are based on the undecidability of the halting problem for counter machines and various reduction techniques.

Each action has a *type* which is a tuple. For instance, $sent(A,t,B)$ has type $(s,a,m,a)$ and $rec(A,t)$ has type $(r,a,m)$, where $s$ stands for "*sent*", $r$ stands for "*rec*", $a$ for "agent", and $m$ for "message".

Each action type $\tau$ induces two decision problems:

1) the *group anonymity problem for type $\tau$ actions w.r.t. an honest agent*, abbreviated $GAP(\tau)$, which is the problem to decide, given a security protocol $\mathcal{P}$, a type $\tau$ action $act$, a non-empty set $G$ of honest agents or messages, and an honest agent $H$ not in $G$, whether $act$ is anonymous within $G$ w.r.t. $H$ in $\mathcal{P}$;
2) the *group anonymity problem for type $\tau$ actions w.r.t. the intruder*, abbreviated $GAP_I(\tau)$, which is the problem to decide, given a security protocol $\mathcal{P}$, a type $\tau$ action $act$, a non-empty set $G$ of honest agents or messages, whether $act$ is anonymous within $G$ w.r.t. the intruder.

Now, we can prove the following theorem.

*Theorem 13:*

1) *$GAP(\tau)$ is undecidable in unrestricted security protocols, for any action type $\tau$.*
2) *$GAP_I(\tau)$ is undecidable in unrestricted protocols, for any sent-action type $\tau$.*

In Section IV-B it has been shown that group anonymity implies minimal anonymity in case of exclusive actions. However, exclusiveness is undecidable.

*Theorem 14: Local exclusiveness problems is undecidable in unrestricted security protocol.*

Role interchangeability is an undecidable problem too. It is the problem to decide, given a security protocol $\mathcal{P}$, an action $sent(A,t)$, a group $G \subseteq \mathcal{A} - \{I\}$ of agents, a finite set $T \subseteq \mathcal{T}$ of message terms, and an honest observer $H$, whether $sent(A,t)$ is role interchangeable within $G \times T$ w.r.t. $H$.

*Theorem 15: Role interchangeability is undecidable in unrestricted security protocols.*

## VI. Complexity of Group Anonymity

The group anonymity problem is undecidable in unrestricted security protocols. Clearly, if we focus on bounded security protocols then group anonymity is decidable. In this section we study the complexity of this problem. Recall first a few concepts regarding bounded protocols [30].

Let $\mathcal{P} = (\mathcal{S}, \mathcal{C}, w)$ be a security protocol, $T \subseteq \mathcal{T}_0$ a finite set, and $k \geq 1$. A $(T,k)$*-run* of $\mathcal{P}$ is any run with the property that all terms in the run are built up upon $T$ and all messages communicated in the course of the run have length at most $k$. When for $\mathcal{P}$ only $(T,k)$-runs are considered we say that it is a *protocol under* $(T,k)$*-runs* or a $(T,k)$*-bounded protocol*, and denote this by $(\mathcal{P},T,k)$. A *bounded protocol* is a $(T,k)$-bounded protocol, for some finite set $T \subseteq \mathcal{T}_0$ and $k \geq 1$.

A *1-session* $(T,k)$*-run* of $\mathcal{P}$ is any $(T,k)$-run of $\mathcal{P}$ obtained by applying each role at most once (not necessarily in its entirety), under the same substitution (i.e., all its events are defined by using the same substitution). Therefore, any 1-session $(T,k)$-run has length at most $|w|$. When for the protocol $\mathcal{P}$ only 1-session $(T,k)$-runs are considered we say that it is a *1-session* $(T,k)$*-bounded protocol.* A *1-session bounded protocol* is a 1-session $(T,k)$-bounded protocol, for some finite set $T \subseteq \mathcal{T}_0$ and $k \geq 1$.

In [30] it has been shown that the number of distinct events in a $(T,k)$-run of a protocol $\mathcal{P}$ is exponential in $poly(size(\mathcal{P}))$, where $size(\mathcal{P}) = |w| + k \log |T|$ and $poly$ is a polynomial. For 1-session $(T,k)$-runs, the number of events in each such run is at most the length of the protocol's body. Although the term $log|T|$ is not necessary to define the size of a 1-session $(T, k)$-bounded protocol, we will use the same protocol size as defined above just for the sake of uniformity with the results in [30].

The following technical lemma [26] will be very useful in estimating the time complexity of our algorithms.

*Lemma 16 ([26]):* Let $\mathcal{P} = (\mathcal{S}, \mathcal{C}, w)$ be a $(T, k)$-bounded protocol, $s$ be the last state of some run of length $n$ of $\mathcal{P}$, $A$ an agent, $t$ be a message of length at most $k$ over $T$, and $\varphi$ a fact whose terms have length at most $k$. Then,

1) it is decidable in $\mathcal{O}(nk^2)$ time whether $t$ is derivable from $s_{A,m}$ (i.e., $t \in \overline{s_{A,m}}$);
2) it is decidable in $\mathcal{O}(n^3k^6)$ time whether $\varphi \in Analz(s_A)$;
3) it is decidable in $\mathcal{O}(n^3k^6|w|)$ time whether $\varphi \in Analz(s_B)$ for some agent $B$.

The state space of a bounded security protocol is finite and so we are able to decide whether an action $act(x)$ is anonymous within some group $G$ w.r.t. some observer $X$. An obvious algorithm for deciding this would search the state space twice: first, the algorithm detects a state $s$ with $s_X \models act(x)$ and then, for each $y \in G$, the algorithm searches for a state $s'$ with $s' \sim^X x$ and $s'_X \models act(y)$. As the number of events of a bounded security protocol is exponential w.r.t. the size of the protocol [30], this algorithm has a very high time complexity (w.r.t. the size of the protocol).

If we restrict the group anonymity problem to basic-term actions (Section IV-A) then Proposition 4 shows that only one search through the state space would suffice.

*Theorem 17:* $GAP(\tau)$ and $GAP_I(\tau)$ are in $NEXPTIME$ for any $\tau$ if they are restricted to basic-term actions of type $\tau$ and bounded security protocols. Moreover, except for $GAP_I(\tau)$ where $\tau$ is a rec-action type, all the other group anonymity problems restricted as above are complete for $NEXPTIME$.

If we restrict more bounded security protocols by allowing only 1-session runs, then we obtain the following results.

*Theorem 18:* $GAP(\tau)$ and $GAP_I(\tau)$ are in $NP$ for any $\tau$ if they are restricted to basic-term actions of type $\tau$ and 1-session bounded security protocols. Moreover, except for $GAP_I(\tau)$ where $\tau$ is a rec-action type, all the other group anonymity problems restricted as above are complete for $NP$.

## VII. Conclusions

Employing the epistemic logic framework developed in [24], [26], this paper proposes an approach to group anonymity in security protocols. This is formulated with respect to an honest agent and with respect to the intruder. A large spectrum of relationships between, and properties of, anonymity concepts were provided. One of the most interesting properties states that a group anonymous action in a security protocol under a passive intruder might not be group anonymous in the same security protocol if the intruder is active, and vice-versa.

It is shown that group anonymity is undecidable in unrestricted security protocols. Clearly, it becomes decidable in bounded security protocols. More precisely, group anonymity is complete for NEXPTIME if it restricted to basic-term actions and bounded security protocols, and it is complete for NP if it restricted to basic-term actions and 1-session bounded security protocols. These results show how difficult is to prove group anonymity for bounded security protocols. In practice, one has to design decision tools for group anonymity to work on restricted classes of protocols in order to obtain feasible results. We are not aware of any approaches for "practical classes of security protocols". However, there are tools such as MCMAS [33], [34] and PRISM [35] capable to check epistemic formulas on "not very complex" security protocols met in practice.

## References

[1] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, Feb 1981. doi: 10.1145/358549.358563. [Online]. Available: http://dx.doi.org/10.1145/358549.358563

[2] ——, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, Oct 1985. doi: 10.1145/4372.4373. [Online]. Available: http://dx.doi.org/10.1145/4372.4373

[3] ——, "The dining cryptographers problem: Unconditional sender untraceability," *Journal of Cryptology*, vol. 1, no. 1, pp. 65–76, 1988. doi: 10.1007/BF00206326. [Online]. Available: http://dx.doi.org/10.1007/BF00206326

[4] S. Schneider and A. Sidiropoulos, "CSP and anonymity." in *4th European Symposium on Research in Computer Security (ESORICS'96)*, ser. Lecture Notes in Computer Science, E. Bertino, H. Kurth, G. Martella, and E. Montolivo, Eds., no. 1146, 1996. doi: 10.1007/3-540-61770-1_38 pp. 198–218. [Online]. Available: http://dx.doi.org/10.1007/3-540-61770-1_38

[5] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, pp. 66–92, Nov 1998. doi: 10.1145/290163.290168. [Online]. Available: http://dx.doi.org/10.1145/290163.290168

[6] P. F. Syverson and S. G. Stubblebine, "Group principals and the formalization of anonymity," in *World Congress on Formal Methods'99*, 1999. doi: 10.1007/3-540-48119-2_45 pp. 814–833. [Online]. Available: http://dx.doi.org/10.1007/3-540-48119-2_45

[7] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," *In 13th USENIX Security Symposium*, 2004.

[8] D. Hughes and V. Shmatikov, "Information hiding, anonymity and privacy: A modular approach," *Journal of Computer Security*, vol. 12, pp. 3–36, Jan 2004. doi: 10.3233/JCS-2004-12102. [Online]. Available: http://dx.doi.org/10.3233/JCS-2004-12102

[9] S. Mauw, J. Verschuren, and E. P. de Vink, "A formalization of anonymity and onion routing," in *In Proceedings of the 9th European Symposium on Research in Computer Security (ESORICS 2004)*. Springer, 2004. doi: 10.1007/978-3-540-30108-0_7 pp. 109–124. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30108-0_7

[10] F. D. Garcia, I. Hasuo, W. Pieters, and P. van Rossum, "Provable anonymity," in *Proceedings of the 2005 ACM Workshop on Formal Methods in Security Engineering (FMSE'05)*, 2005. doi: 10.1145/1103576.1103585 pp. 63–72. [Online]. Available: http://dx.doi.org/10.1145/1103576.1103585

[11] J. Y. Halpern and K. R. O'Neill, "Anonymity and information hiding in multi-agent systems," *Journal of Computer Security*, vol. 13, no. 3, pp. 483–514, 2005. doi: 10.1109/CSFW.2003.1212706. [Online]. Available: http://dx.doi.org/10.1109/CSFW.2003.1212706

[12] K. Mano, Y. Kawabe, H. Sakurada, and Y. Tsukada, "Role interchangeability and verification of electronic voting," in *The 2006 Symposium on Cryptography and Information Security*, Hiroshima, Japan, Jan 2006.

[13] T. Chothia, S. Orzan, J. Pang, and M. T. Dashti, “A framework for automatically checking anonymity with $\mu$crl,” in *In Proceedings TGC06, LNCS*, 2007. doi: 10.1007/978-3-540-75336-0_19 pp. 301–318. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75336-0_19

[14] J. Feigenbaum, A. Johnson, and P. Syverson, “A model of onion routing with provable anonymity,” in *In Proceedings of the 11th Financial Cryptography and Data Security Conference*. Springer-Verlag, 2007. doi: 10.1007/978-3-540-77366-5_9 pp. 57–71. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77366-5_9

[15] Y. Kawabe, K. Mano, H. Sakurada, and Y. Tsukada, “Theorem-proving anonymity of infinite-state systems,” *Inf. Process. Lett.*, vol. 101, pp. 46–51, Jan 2007. doi: 10.1016/j.ipl.2006.06.016. [Online]. Available: http://dx.doi.org/10.1016/j.ipl.2006.06.016

[16] X. Sun, H. Wang, and J. Li, “On the complexity of restricted k-anonymity problem,” in *Proceedings of the 10th Asia-Pacific Web Conference on Progress in WWW Research and Development*, ser. APWeb’08. Berlin, Heidelberg: Springer-Verlag, 2008. ISBN 3-540-78848-4, 978-3-540-78848-5 pp. 287–296.

[17] M. Edman and B. Yener, “On anonymity in an electronic society: A survey of anonymous communication systems,” *ACM Computing Surveys*, vol. 42, no. 1, 2009. doi: 10.1145/1592451.1592456. [Online]. Available: http://dx.doi.org/10.1145/1592451.1592456

[18] J. F. Groote and S. Orzan, “Parameterised anonymity,” in *Formal Aspects in Security and Trust*, P. Degano, J. Guttman, and F. Martinelli, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 177–191. ISBN 978-3-642-01464-2. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01465-9_12

[19] H. Comon-Lundh, Y. Kawamoto, and H. Sakurada, “Computational and symbolic anonymity in an unbounded network,” *JSIAM Letters*, vol. 1, pp. 28–31, 2009. doi: 10.14495/jsiaml.1.28. [Online]. Available: http://dx.doi.org/10.14495/jsiaml.1.28

[20] Y. Tsukada, K. Mano, H. Sakurada, and Y. Kawabe, “Anonymity, privacy, onymity, and identity: A modal logic approach,” in *2009 International Conference on Computational Science and Engineering*, 2009. doi: 10.1109/CSE.2009.251 pp. 42–51. [Online]. Available: http://dx.doi.org/0.1109/CSE.2009.251

[21] C. A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou, “Providing mobile users’ anonymity in hybrid networks,” in *European Symposium on Research in Computer Security (ESORICS 2010)*, ser. Lecture Notes in Computer Science, D. Gritzalis, B. Preneel, and T. Theoharidou, Eds., vol. 6345, 2010. doi: 10.1007/978-3-642-15497-3_33 pp. 540–557. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15497-3_33

[22] M. Backes, G. Doychev, M. Dürmuth, and B. Köpf, “Speaker recognition in encrypted voice streams,” in *European Symposium on Research in Computer Security (ESORICS 2010)*, ser. Lecture Notes in Computer Science, D. Gritzalis, B. Preneel, and T. Theoharidou, Eds., vol. 6345, 2010. doi: 10.1007/978-3-642-15497-3_31 pp. 508–523. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15497-3_31

[23] A. Pfitzmann and M. Hansen, “A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management,” http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, Aug 2010.

[24] F. L. Ţiplea, L. Vamanu, and C. Vârlan, “Complexity of anonymity for security protocols,” in *European Symposium on Research in Computer Security (ESORICS 2010)*, ser. Lecture Notes in Computer Science, D. Gritzalis, B. Preneel, and T. Theoharidou, Eds., vol. 6345, 2010. doi: 10.1007/978-3-642-15497-3_34 pp. 558–572. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15497-3_34

[25] I. Goriac, “An epistemic logic based framework for reasoning about information hiding,” in *Availability, Reliability and Security, International Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, 2011. doi: 10.1109/ARES.2011.49 pp. 286–293. [Online]. Available: http://dx.doi.org/10.1109/ARES.2011.49

[26] F. L. Ţiplea, L. Vamanu, and C. Vârlan, “Reasoning about minimal anonymity in security protocols,” *Future Generation Computer Systems*, vol. 29, pp. 828–842, March 2013. doi: 10.1016/j.future.2012.02.001. [Online]. Available: http://dx.doi.org/10.1016/j.future.2012.02.001

[27] S. Kramer, “Cryptographic protocol logic: Satisfaction for (timed) Dolev-Yao cryptography,” *The Journal of Logic and Algebraic Programming*, vol. 77, pp. 60–91, Sep 2008. doi: 10.1016/j.jlap.2008.05.005. [Online]. Available: http://dx.doi.org/10.1016/j.jlap.2008.05.005

[28] R. Ramanujam and S. P. Suresh, “A decidable subclass of unbounded security protocols,” in *Workshop on Issues in the Theory of Security (WITS’03)*, 2003, pp. 11–20.

[29] F. L. Ţiplea, C. Enea, and C. V. Bîrjoveanu, “Decidability and complexity results for security protocols,” in *Verfication of Infinite-state Systems with Applications to Security (VISSAS 2005)*, E. Clarke, M. Minea, and F. Tiplea, Eds. IOS Press, 2005, pp. 185–211.

[30] F. L. Ţiplea, C. Enea, C. V. Bîrjoveanu, and I. Boureanu, “Secrecy for bounded protocols with freshness check is NEXPTIME-complete,” *Journal of Computer Security*, vol. 16, pp. 689–712, Dec 2008. doi: 10.3233/JCS-2007-0306. [Online]. Available: http://dx.doi.org/10.3233/JCS-2007-0306

[31] D. Dolev and A. Yao, “On the security of public-key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983. doi: 10.1109/TIT.1983.1056650. [Online]. Available: http://dx.doi.org/10.1109/TIT.1983.1056650

[32] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning About Knowledge*. The MIT Press, 2003.

[33] A. Lomuscio, H. Qu, and F. Raimondi, “MCMAS: A model checker for the verification of multi-agent systems,” in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, A. Bouajjani and O. Maler, Eds. Springer Berlin Heidelberg, 2009, pp. 682–688. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02658-4_55

[34] ——, “MCMAS: An open-source model checker for the verification of multi-agent systems,” *International Journal on Software Tools for Technology Transfer*, pp. 1–22, 2015. doi: 10.1007/s10009-015-0378-x. [Online]. Available: http://dx.doi.org/10.1007/s10009-015-0378-x

[35] M. Kwiatkowska, G. Norman, and D. Parker, “PRISM 4.0: Verification of probabilistic real-time systems,” in *Proceedings of the 23rd International Conference on Computer Aided Verification*. Springer-Verlag, 2011. doi: 10.1007/978-3-642-22110-1_47 pp. 585–591. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22110-1_47