Security Protocols Checker

Autor
Ștefan Stan

Coordonator științific
Lect. dr. Cosmin-Nicolae Vârlan

Facultatea de Informatică
Universitatea Alexandru Ioan Cuza din Iași

3 Iulie 2015

Cuprins

Cuvinte cheie

protocol de securitate, model checking,
integritate, confidențialitate,
Java, analiză sintactică, analiză semantică

Obiective

- definirea, încărcarea în memorie și rularea protocoalelor de securitate descrise la nivel teoretic;
- verificarea automată a proprietăților de securitate:
  - **Integritate**
    protecția informației spre a nu fi **modificată** de către surse neautorizate;
  - **Confidențialitate**
    protecția informației spre a nu fi **accesată** de către surse neautorizate.

Aplicații similare

Scyther

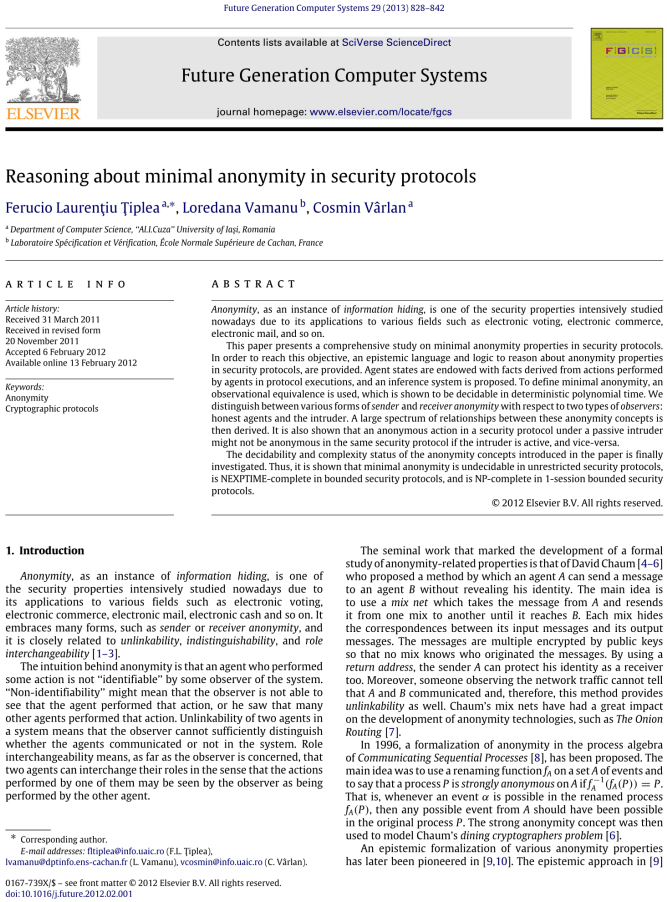
- Cas Cremers - *Scyther - Semantics and Verification of Security Protocols*
- Verifică **integritatea** și **confidențialitatea** unei instanțe de protocol
- Arată atacurile posibile în manieră grafică

Sursa de inspirație

Reasoning about minimal anonymity in security protocols

Autori:

- Prof. dr. Ferucio Laurențiu Țiplea
- Lect. dr. Cosmin Vârlan
- Loredana Vamanu



Future Generation Computer Systems 29 (2013) 828–842

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Reasoning about minimal anonymity in security protocols

Ferucio Laurențiu Țiplea [a,*], Loredana Vamanu [b], Cosmin Vârlan [a]

[a] Department of Computer Science, "Al.I.Cuza" University of Iași, Romania
[b] Laboratoire Spécification et Vérification, École Normale Supérieure de Cachan, France

ARTICLE INFO

Article history:
Received 31 March 2011
Received in revised form
20 November 2011
Accepted 6 February 2012
Available online 13 February 2012

Keywords:
Anonymity
Cryptographic protocols

ABSTRACT

Anonymity, as an instance of *information hiding*, is one of the security properties intensively studied nowadays due to its applications to various fields such as electronic voting, electronic commerce, electronic mail, and so on.

This paper presents a comprehensive study on minimal anonymity properties in security protocols. In order to reach this objective, an epistemic language and logic to reason about anonymity properties in security protocols, are provided. Agent states are endowed with facts derived from actions performed by agents in protocol executions, and an inference system is proposed. To define minimal anonymity, an observational equivalence is used, which is shown to be decidable in deterministic polynomial time. We distinguish between various forms of *sender* and *receiver anonymity* with respect to two types of *observers*: honest agents and the intruder. A large spectrum of relationships between these anonymity concepts is then derived. It is also shown that an anonymous action in a security protocol under a passive intruder might not be anonymous in the same security protocol if the intruder is active, and vice-versa.

The decidability and complexity status of the anonymity concepts introduced in the paper is finally investigated. Thus, it is shown that minimal anonymity is undecidable in unrestricted security protocols, is NEXPTIME-complete in bounded security protocols, and is NP-complete in 1-session bounded security protocols.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Anonymity, as an instance of *information hiding*, is one of the security properties intensively studied nowadays due to its applications to various fields such as electronic voting, electronic commerce, electronic mail, electronic cash and so on. It embraces many forms, such as *sender* or *receiver anonymity*, and it is closely related to *unlinkability*, *indistinguishability*, and *role interchangeability* [1–3].

The intuition behind anonymity is that an agent who performed some action is not "identifiable" by some observer of the system. "Non-identifiability" might mean that the observer is not able to see that the agent performed that action, or he saw that many other agents performed that action. Unlinkability of two agents in a system means that the observer cannot sufficiently distinguish whether the agents communicated or not in the system. Role interchangeability means, as far as the observer is concerned, that two agents can interchange their roles in the sense that the actions performed by one of them may be seen by the observer as being performed by the other agent.

The seminal work that marked the development of a formal study of anonymity-related properties is that of David Chaum [4–6] who proposed a method by which an agent A can send a message to an agent B without revealing his identity. The main idea is to use a *mix net* which takes the message from A and resends it from one mix to another until it reaches B. Each mix hides the correspondences between its input messages and its output messages. The messages are multiple encrypted by public keys so that no mix knows who originated the messages. By using a *return address*, the sender A can protect his identity as a receiver too. Moreover, someone observing the network traffic cannot tell that A and B communicated and, therefore, this method provides *unlinkability* as well. Chaum's mix nets have had a great impact on the development of anonymity technologies, such as *The Onion Routing* [7].

In 1996, a formalization of anonymity in the process algebra of *Communicating Sequential Processes* [8], has been proposed. The main idea was to use a renaming function f_A on a set A of events and to say that a process P is *strongly anonymous* on A if $f_A^{-1}(f_A(P)) = P$. That is, whenever an event α is possible in the renamed process $f_A(P)$, then any possible event from A should have been possible in the original process P. The strong anonymity concept was then used to model Chaum's *dining cryptographers problem* [6].

An epistemic formalization of various anonymity properties has later been pioneered in [9,10]. The epistemic approach in [9]

* Corresponding author.
E-mail addresses: ftiplea@info.uaic.ro (F.L. Țiplea), lvamanu@dptinfo.ens-cachan.fr (L. Vamanu), vcosmin@info.uaic.ro (C. Vârlan).

0167-739X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2012.02.001

Protocoale de securitate

Signatura unui protocol - $\mathcal{S} = (\mathcal{A}, \mathcal{K}, \mathcal{N})$

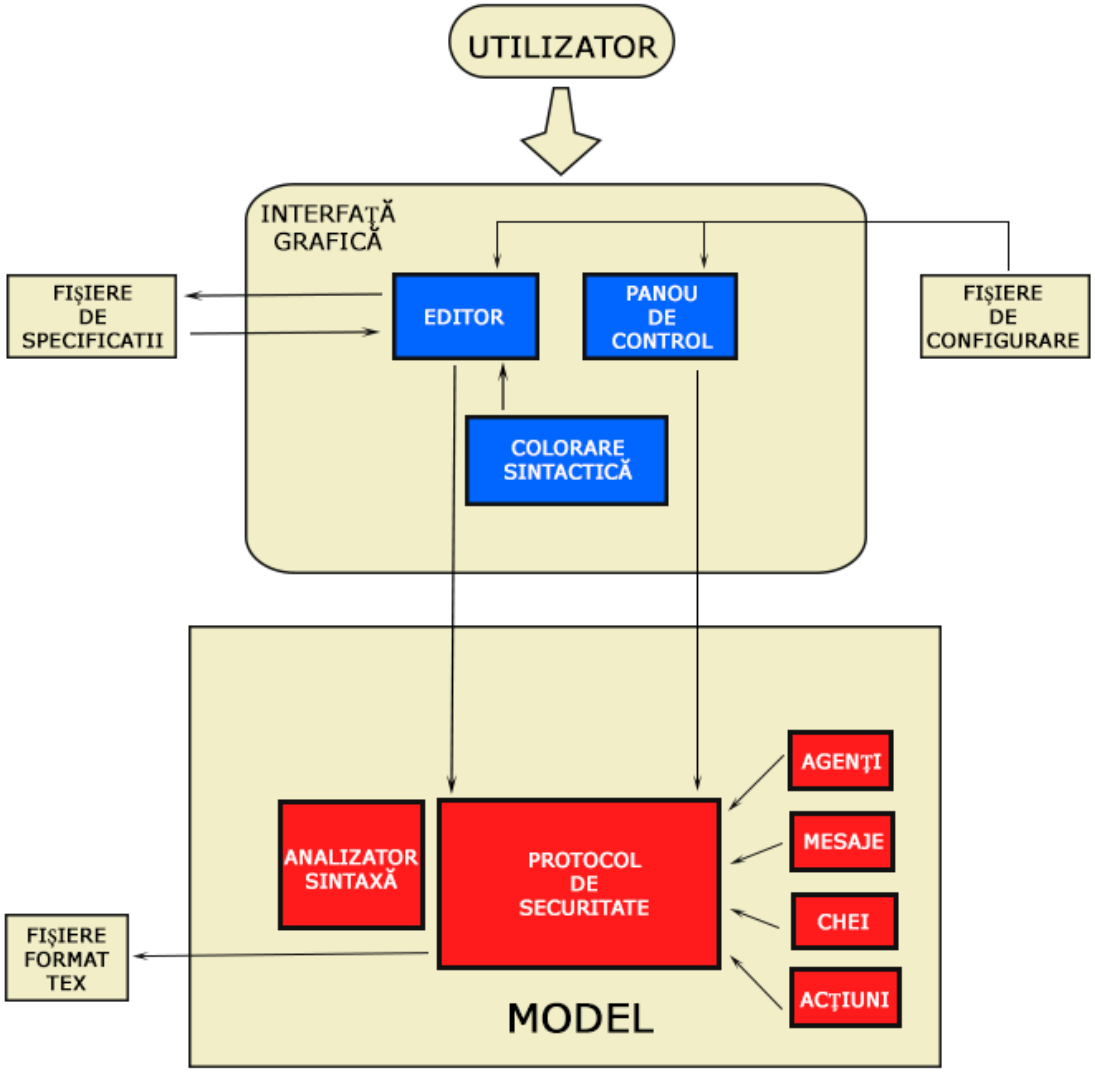
$\mathcal{A}$ - mulțime finită de *agenți*; include *intrusul* - *I*
$\mathcal{K}$, $\mathcal{N}$ - două mulțimi cel mult numărabile de *chei* și, respectiv, *nonce*-uri.

Termi

$\mathcal{T}_0 = \mathcal{A} \cup \mathcal{K} \cup \mathcal{N}$ - Mulțimea termilor de bază
Mulțimea $\mathcal{T}$ a termilor este definită inductiv:

- $\mathcal{T}_0 \in \mathcal{T}$;
- $t_1, t_2 \in \mathcal{T} \Rightarrow (t_1, t_2) \in \mathcal{T}$;
  $(t_1, ..., t_n) = ((t_1, ..., t_{n-1})\ t_n) \Rightarrow (t_1, ..., t_n) \in \mathcal{T}$, n≥3;
- $t \in \mathcal{T}, K \in \mathcal{K} \Rightarrow \{t\}_K \in \mathcal{T}$;

Arhitectura aplicației



Descrierea unui protocol

Definirea specificației

agent_1 : IK $\text{info}_{11}, \ldots, \text{info}_{1m}$

...

agent_n : IK $\text{info}_{n1}, \ldots, \text{info}_{np}$

acțiune_1

...

acțiune_k

Informație

agent|ticket|nonce|cheie

Acțiune

agent_1 ! agent_2 : (termiGenerați) mesaj

agent_1 ? agent_2 : mesaj

Descrierea specificației protocolului

Specificație protocol:

$$A \; ! \; B \;\; : (\{N_{1_A}\}) \; \{N_{1_A}\}K^e_B$$

$$B \; ? \; A : \{N_{1_A}\}K^e_B$$

$$B \; ! \; A \;\; : \{N_{1_A}\}K^e_A$$

$$A \; ? \; B : \{N_{1_A}\}K^e_A$$

Rulare - Security Protocols Checker

Agenți comuni celor două instanțe de protocol: B

Instanța 1	Instanța 2
$A \; ! \; B \; : (\{N_{1_A}\}) \; \{N_{1_A}\}K^e_B$ A generated $\{N_{1_A}\}$ for B A sent $\{N_{1_A}\}K^e_B$ to B I received $\{N_{1_A}\}K^e_B$	$I!B{:}\{N_{1_A}\}K^e_B$ I trimite același mesaj care s-a trimis în instanța numărul 1, în aceeași acțiune. Deoarece mai tarziu va descoperi $[N_{1_A}]$ din cealaltă instanță de protocol
$B \; ? \; A \; : \{N_{1_A}\}K^e_B$ B received $\{N_{1_A}\}K^e_B$ from A B received N_{1_A} from A	$B?I{:}\{N_{1_A}\}K^e_B$ B received $\{N_{1_A}\}K^e_B$ from I B received N_{1_A} from I

Rulare - Security Protocols Checker

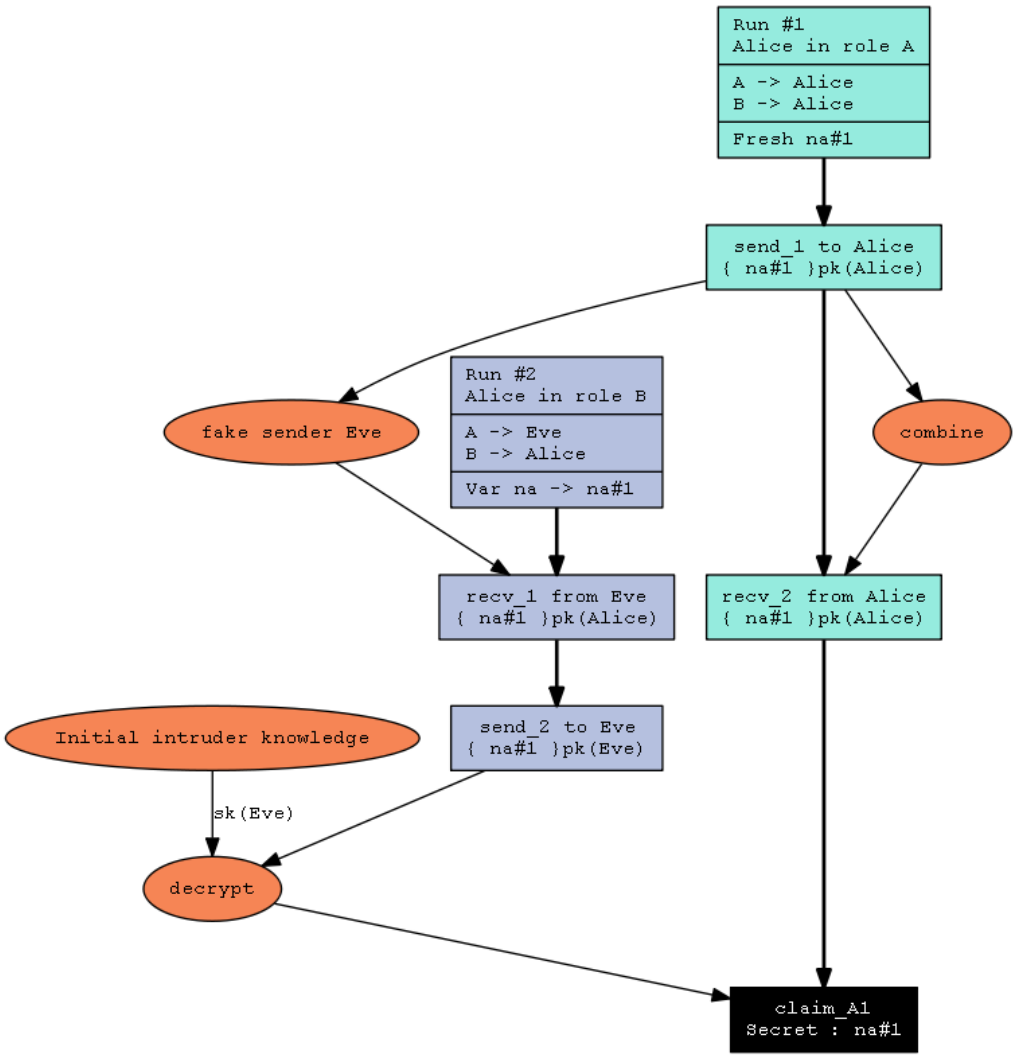
$B\ !\ A\ : \{N_{1_A}\}K^e_A$
B sent $\{N_{1_A}\}K^e_A$ to A
I received $\{N_{1_A}\}K^e_A$

$A\ ?\ B : \{N_{1_A}\}K^e_A$
A received $\{N_{1_A}\}K^e_A$ from B
A received N_{1_A} from B

$B!I{:}\{N_{1_A}\}K^e_I$
B sent $\{N_{1_A}\}K^e_I$ to I
I received $\{N_{1_A}\}K^e_I$

$I?B{:}\{N_{1_A}\}K^e_I$
I received $\{N_{1_A}\}K^e_I$ from B
I received N_{1_A} from B

Rulare - Scyther



[Id 4] Protocol prot3, role A, claim type Secret

Corectarea specificației

Specificație corectă:

$$A \; ! \; B \; : (\{N_{1_A}\}) \; \{A, N_{1_A}\} K^e_B$$

$$B \; ? \; A : \{A, N_{1_A}\} K^e_B$$

$$B \; ! \; A \; : \{N_{1_A}\} K^e_A$$

$$A \; ? \; B : \{N_{1_A}\} K^e_A$$

Concluzii

- Utilitar ce încarcă în memorie și rulează protocoale de securitate descrise în fișiere de specificație;
- Ofera mijloacele necesare pentru verificarea **integrității** și a **confidențialității**;
- Verifică dacă, la nivel teoretic, proprietățile sunt asigurate de către un protocol;