FORMALLY REASONING ABOUT SECURITY ISSUES IN P2P PROTOCOLS: A CASE STUDY

Andrés Aristizábal[1] Hugo López[1] Frank Valencia[2]
Camilo Rueda[1]

Pontificia Universidad Javeriana - Cali, Colombia

CNRS-LIX École Polytechnique de Paris, France.

Third Taiwanese-French Conference on Information Technology
Nancy, France
March 2006

Process Calculi for Security Protocols

- Formalisms that allow the specification and verification of concurrent and distributed systems. Systems are represented as processes that may evolve in time.
- Probably the most popular process is Milner's π, a calculus based in names to model distributed/mobile systems.
- Process calculi have been extended to represent security protocols. Security properties are verified by means of behavioral equivalences (e.g. Milner's π, Abadi-Gordon's SPi calculus).

SPL, a process calculi focused in communication protocols, presents some interesting properties in the model for security analysis using intuitive proof techniques.

Motivation

P2P Systems

New kind of computer organization, intended to use distributed computing to achieve specific goals.

- Offer a way to put together disperse resources available in open networks such as Internet
- Ideal for lengthy or infinite computations.
- Robust to faults or intentional attacks.

Given the concurrent nature of P2P systems, it is natural to try modelling them using process calculi

Contributions

Our first approach to formally reason about security concerns in protocols for P2P systems using Process Calculi.

- Using SPL, a process calculus tailored to the analysis of security protocols with associated proof techniques
- A case of Study: The MUTE protocol. A content-sharing protocol that preserves the anonymity of peers involved in the network.
- We derive a property of secrecy: data transmitted over the network remains undisclosed for outsider attackers unless the keys are published.

Plan

- Preliminaries: SPL Process Calculus.
- A characterisation of P2P systems based on several searching protocols.
- Formalization and Verification of Security Properties in MUTE
- Ongoing and Future Work.

SPL

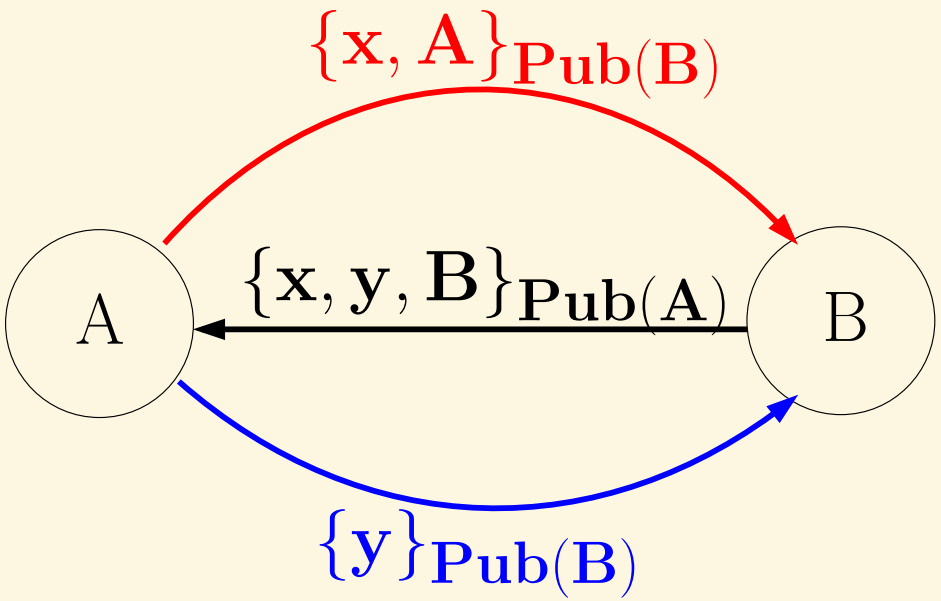
- Process calculus specially designed to reason about security protocols (Winskel & Crazzolara 2004).
- Concise and precise protocol descriptions.
- Reasoning techniques based on events, present dependencies between processes.
- Well-founded and general set of proof principles.

Syntax

Variables	x, Y	*(names)*
	$Pub(v) \mid Priv(v) \mid Key(\vec{v}) \mid \chi,$	*(keys)*
	$v \mid k \mid (M, M') \mid \{M\}_k \mid \psi$	*(messages)*
Processes		
Binding	$\mathbf{P} ::= out\ new(\vec{x})\ M.\mathbf{P}$	*(secret generation)*
	$\mid in\ pat\ \psi\ M.\mathbf{P}$	*(pattern-matching Input)*
Replication	$\mid \ \|_{i \in I}\ \mathbf{P_i}$	*(Process Composition)*
	$\mid !\mathbf{P}$	*(Infinite Behavior)*

- Asynchronous, process oriented language. Every output is persistent in the network.

An Authentication Example: NSL



Alice

$Init(A, B) \equiv$ $out\ new(x)\{x, A\}_{Pub(B)}.$
$in\{x, y, B\}_{Pub(A)}.$
$out\{y\}_{Pub(B)}$

Bob

$Resp(B) \equiv$ $in\{x, Z\}_{Pub(B)}.$
$out\ new(y)\{x, y, B\}_{Pub(Z)}.$
$in\{y\}_{Pub(B)}$

Cryptographic Model

SPL relies on the Dolev-Yao threat model

- Perfect Cryptography.
- A powerful attacker: overhear, intercept, introduce and synthesise messages over the network.
- The model must include this spy to verify security properties.

Operational Semantics

Labelled Transition Semantics

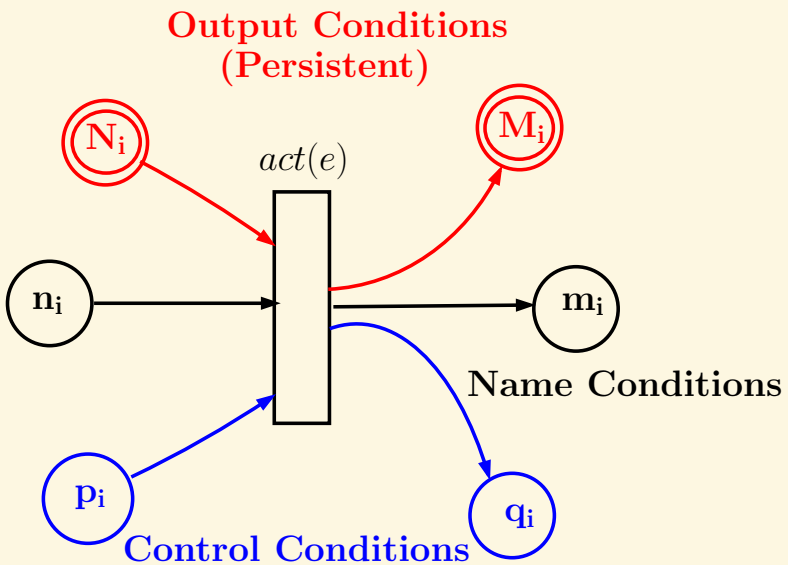
Based on configurations of the form $\langle p, s, t \rangle$ where p is a set of process, s a set of used names and t a set of closed messages, three reaction rules are declared:

- Output Processes
- Pattern-Matching inputs.
- Parallel Composition.

Transition Semantics do not provide causal dependencies between events.

Event-Based Semantics

- Persistent Coloured Petri-nets can provide an appropriate model where input and output conditions can be modelled, making verification of threats easier.



- Events: Messages, generated names and Processes.
- Actions: input / output processes. Parallel Composition

Transition and Event-based semantics are equivalent

General Proof Principles

General set of theorems available for the verification of security properties. They can be grouped as general results:

1. Concerning casual dependencies.
2. Guaranteeing the freshness of generated names.
3. Declaring relationship between messages.

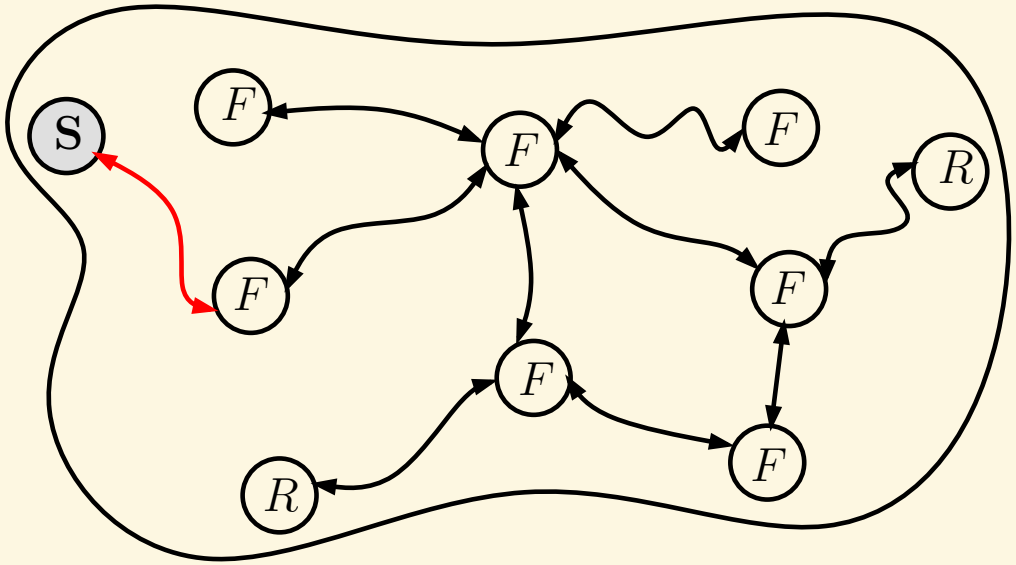
Outline

The MUTE protocol

- Content-Sharing Protocol: Users request for an specific keyword and the network spreads the query until someone responds with the appropriate information.
- Security based on key distribution methods.
- Hides information about the data transmitted for outsider attackers.

Definition of roles in P2P protocols

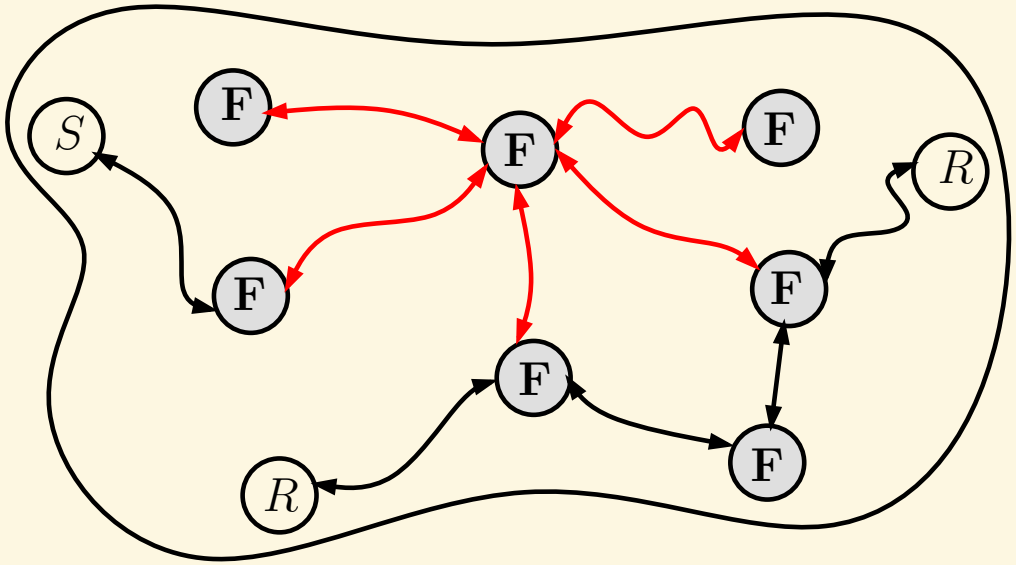
- Initiator
- Forwarder
- Responder



$$Init(A) \equiv (\|_{B \in ngb(A)}\ out\ new(m)(\{m\}_{Key(A,B)}, A, B)) \cdot (\|_{Y \in ngb(A)}\ in\ (\{m, n\}_{key(Y,A)}, Y, A))$$

Definition of roles in P2P protocols

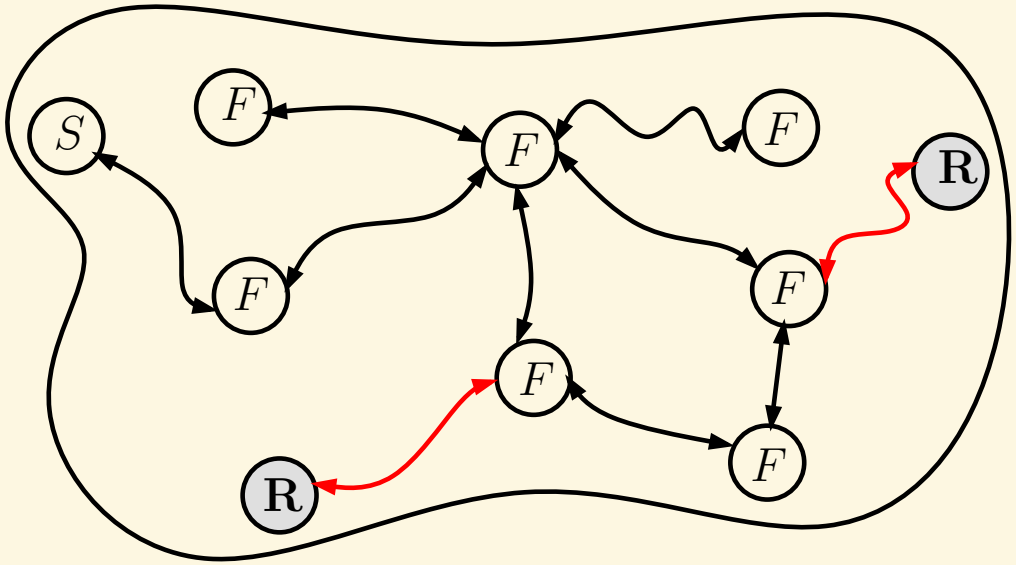
- Initiator
- Forwarder
- Responder



$$Interm(A) \equiv \|_{Y \in ngb(A)} in(\{m\}_{key(Y,A)}, Y, A) \cdot \|_{B \in ngb(A)-\{Y\}} out(\{m\}_{key(A,B)}, A, B)$$

Definition of roles in P2P protocols

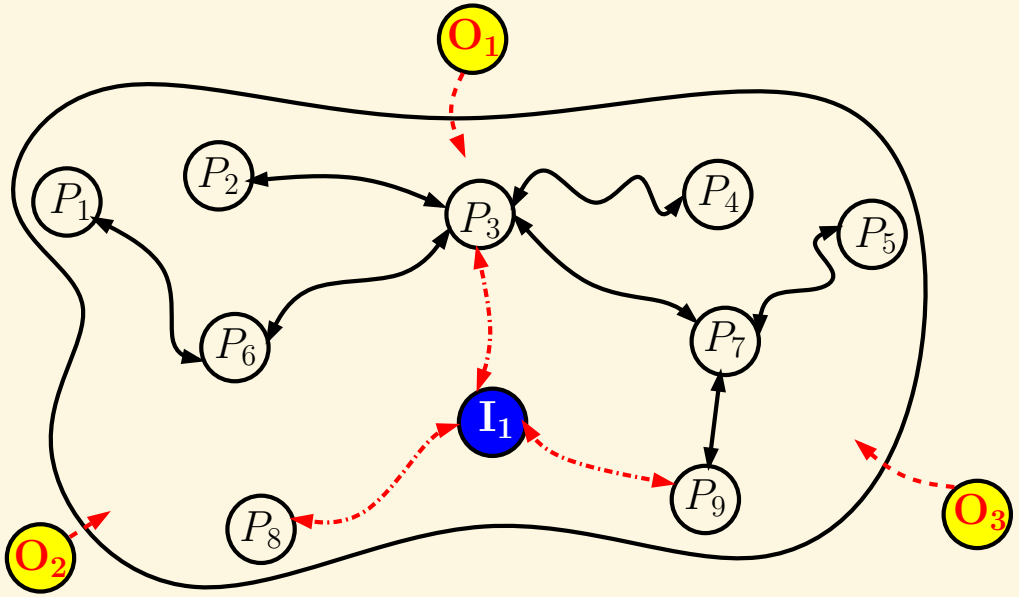
- Initiator
- Forwarder
- Responder



$$Resp(A) \equiv \quad \|_{Y \in ngb(A)}\ in(\{m\}_{Key(Y,A)}, Y, A) \\ .(\|_{B \in ngb(A)}\ out\ new(n)(\{m, n\}_{key(A,B)}, A, B))$$

Definition of roles in P2P protocols

- Initiator
- Forwarder
- Responder



$$MUTE \equiv !(Init(A) \parallel Interm(A) \parallel Resp(A))$$

In a pure P2P system agents acts together in a completely dynamic system, with multiple roles. Possible untrustful insider users and outsider attackers.

Secrecy Proofs Of MUTE

It is crucial to guarantee that the messages in transit over the network cannot be available for the disclosure of outsider attackers

Theorem

Given a run of MUTE, if the keys used to encrypt messages between peers are never leaked, and the privacy-sensitive data m always appears under those encryptions, then nobody outside the network can have access m.

Secrecy Proofs Of MUTE

It is crucial to guarantee that the messages in transit over the network cannot be available for the disclosure of outsider attackers

Theorem

Given a run of MUTE, if the keys used to encrypt messages between peers are never leaked, and the privacy-sensitive data m always appears under those encryptions, then nobody outside the network can have access m.

Proof Idea.

1. Formalize the secrecy property using SPL's general proof principles.
2. Suppose an event in which that property is not fulfilled, i.e.: m is accessed by an agent outside the network.
3. Search for an event that breaks the property using event dependencies, if the event can't be reached, the secrecy property is fulfilled

□

Outline

Summary

In this talk:

- We have presented SPL as a complete formalism well suited for security analysis.
- We have provided a characterisation of P2P searching protocols in the field of security analysis.
- We have used the inherent reasoning techniques of the formalism to ensure the correctness of the protocol.

Current Work – Part I

- Use the approach with other P2P protocols [$\surd$]
- Consider Insider attackers [$\surd$]
- Prove specific properties for P2P systems [$\surd$]

Current Work – Part II

Extension of the language to model and verify other systems [+]

- SPL remarks strong similarities with CCP in the way it treats the information in the network.
- Concurrent Constraint Programming is a formalism that generalizes logic Programming.
- Concurrent agents can interact each other by communication methods in a so-called *store*.
- Logical and Denotational description of processes.

Current Work – Part II

Extension of the language to model and verify other systems [+]

- Our approach will be to give a logic representation an attacker can inffer from the network.
- Use of an associated inference system to model and verify security properties.
- Take advantage of a wide set of tools based in this model with the aim of contruct automatic verification tools for security protocols.

THANK YOU!