Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information and Computation 196 (2005) 127–155

Information and Computation

www.elsevier.com/locate/ic

Secrecy and group creation

Luca Cardelli[a], Giorgio Ghelli[b], Andrew D. Gordon[a,*]

[a] *Microsoft Research, Roger Needham Building, 7 Thomson Avenue, Cambridge, UK*
[b] *Dipartimento di Informatica, Università di Pisa, Via Buonarroti 2, Pisa, Italy*

Received 21 April 2001; revised 6 April 2004

Abstract

We add an operation of group creation to the typed π-calculus, where a group is a type for channels. Creation of fresh groups has the effect of statically preventing certain communications, and can block the accidental or malicious leakage of secrets. Intuitively, no channel belonging to a fresh group can be received by processes outside the initial scope of the group, even if those processes are untyped. We formalize this intuition by adapting a notion of secrecy introduced by Abadi, and proving a preservation of secrecy property.
© 2004 Elsevier Inc. All rights reserved.

Keywords: π-Calculus; Secrecy; Security types

1. Introduction

Group creation is a natural extension of the sort-based type systems developed for the π-calculus. However, group creation has an interesting and subtle connection with secrecy. We start from the untyped π-calculus, where an operation to create fresh communication channels can be interpreted as creating fresh secrets. Under this interpretation, though, secrets can be leaked. We then introduce the notion of groups, which are types for channels, together with an operation for creating fresh groups. We explain how a fresh secret belonging to a fresh group can never be communicated to

[*] Corresponding author.
E-mail address: adg@microsoft.com (A.D. Gordon).

doi:10.1016/j.ic.2004.08.003

anybody who does not know the group in the first place. In other words, our type system prevents secrets from being leaked. Crucially, groups are not values, and cannot be communicated; otherwise, this secrecy property would fail.

1.1. Leaking secrets

Consider the following configuration, where P is a private subsystem (a player) running in parallel with a potentially hostile adversary O (an opponent):

$$O \mid P$$

Suppose that the player P wants to create a fresh secret x. For example, x could be a private communication channel to be used only between subsystems of P. In the π-calculus this can be done by letting P evolve into a configuration $(\nu x)P'$, which means: create a new channel x to be used in the scope of P'.

$$O \mid (\nu x)P'$$

The channel x is intended to remain private to P'. This privacy policy is going to be violated if the system then evolves into a situation such as the following, where p is a public channel known to the opponent ($p(y)$ is input of y on p, and $\overline{p}\langle x\rangle$ is output of x on p):

$$p(y).O' \mid (\nu x)(\overline{p}\langle x\rangle \mid P'')$$

In this situation, the name x is about to be sent by the player over the public channel p and received by the opponent. In order for this communication to happen, the rules of the π-calculus, described in Section 2, require first an enlargement (extrusion) of the scope of x (otherwise x would escape its lexical scope). We assume that x is different from p, y, and any other name in O', so that the enlargement of the scope of x does not cause name conflicts. After extrusion, we have:

$$(\nu x)(p(y).O' \mid \overline{p}\langle x\rangle \mid P'')$$

Now, x can be communicated over p into the variable y, while keeping x entirely within the scope of (νx). This results in:

$$(\nu x)(O'\{y{\leftarrow}x\} \mid P'')$$

where the opponent has acquired the secret.

1.2. Preventing leakage

The private name x has been leaked to the opponent by a combination of two mechanisms: the output instruction $\overline{p}\langle x\rangle$, and the extrusion of (νx). Can we prevent this kind of leakage of information? We have to consider that such a leakage may arise simply because of a mistake in the code of the player P, or because P decides to violate the privacy policy of x, or because a subsystem of P acts as a spy for the opponent.