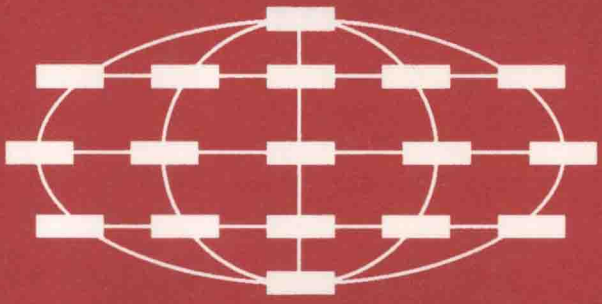
Victor Malyshkin (Ed.)

LNCS 3606

Parallel Computing Technologies

8th International Conference, PaCT 2005
Krasnoyarsk, Russia, September 2005
Proceedings



Springer

Victor Malyshkin (Ed.)

Parallel Computing Technologies

8th International Conference, PaCT 2005
Krasnoyarsk, Russia, September 5-9, 2005
Proceedings

Springer

Volume Editor

Victor Malyshkin
Russian Academy of Sciences
Institute of Computational Mathematics and Mathematical Geophysics
Supercomputer Software Department
pr. Lavrentiev 6, ICM MG RAS, 630090 Novosibirsk, Russia
E-mail: malysh@ssd.sscc.ru

Library of Congress Control Number: 2005930458

CR Subject Classification (1998): D, F.1-2, C, I.6

ISSN 0302-9743
ISBN-10 3-540-28126-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28126-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11535294 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3606

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Preface

The PaCT 2005 (Parallel Computing Technologies) conference was a four-day conference held in Krasnoyarsk, September 5–9, 2005. This was the Eighth international conference in the PaCT series. The conferences are held in Russia every odd year. The first conference, PaCT '91, was held in Novosibirsk (Academgorodok), September 7 – 11, 1991. The next PaCT conferences were held in Obninsk (near Moscow), August 30 – September 4, 1993, in St. Petersburg, September 12–15, 1995, in Yaroslavl, September, 9–12 1997, in Pushkin (near St. Petersburg) September, 6–10 1999, in Academgorodok (Novosibirsk), September 3–7, 2001, and in Nizhni Novgorod, September 15–19, 2003. The PaCT proceedings are published by Springer in the LNCS series.

PaCT 2005 was jointly organized by the Institute of Computational Mathematics and Mathematical Geophysics of the Russian Academy of Sciences (RAS), the Institute of Computational Modeling also of the RAS and the State Technical University of Krasnoyarsk.

The purpose of the conference was to bring together scientists working on theory, architecture, software, hardware and the solution of large-scale problems in order to provide integrated discussions on Parallel Computing Technologies.

The conference attracted about 100 participants from around the world. Authors from 20 countries submitted 78 papers. Of those submitted, 38 papers were selected for the conference as regular ones; there was also 1 invited paper. In addition there were a number of posters presented. All the papers were internationally reviewed by at least three referees. The demo session was organized for the participants.

PaCT 2007 is planned to be held in Irlutsk, near lake Baikal, in September as usual.

Many thanks to our sponsors: the Russian Academy of Sciences, the Russian Fund for Basic Research, the Russian State Committee of Higher Education, and IBM, for their financial support. Organizers highly appreciated the help of the Association Antenne-Provence (France).

June 2005 Victor Malyshkin

Organization

PaCT 2005 was organized by the Supercomputer Software Department, Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch, Russian Academy of Sciences (SB RAS) in cooperation with the Institute of Computational Modelling, SB RAS (Krasnoyarsk) and the State Technical University of Krasnoyarsk.

Program Committee

V. Malyshkin	Chairman (Russian Academy of Sciences)
F. Arbab	(Centre for MCS, The Netherlands)
O. Bandman	(Russian Academy of Sciences)
F. Cappello	(INRIA, France)
T. Casavant	(University of Iowa, USA)
A. Chambarel	(University of Avignon, France)
P. Degano	(State University of Pisa, Italy)
D. Etiemble	(Université Paris Sud, Orsay, France)
B. Goossens	(University of Perpignan, France)
S. Gorlatch	(University of Muenster, Germany)
A. Hurson	(Pennsylvania State University, USA)
Yu. Karpov	(St.-Petersburg State Technical University, Russia)
B. Lecussan	(State University of Toulouse, France)
J. Li	(University of Tsukuba, Japan)
T. Ludwig	(Ruprecht-Karls-Universität Heidelberg, Germany)
G. Mauri	(University of Milan, Italy)
G. Papadopoulos	(University of Cyprus, Cyprus)
M. Raynal	(IRISA, Rennes, France)
B. Roux	(L3M, France)
V. Shaidurov	(Russian Academy of Sciences)
G. Silberman	(IBM, USA)
P. Sloot	(University of Amsterdam. The Netherlands)
C. Trinitis	(LRR, Munich, Germany)
M. Valero	(Universitat Politècnica de Catalunya, Spain)
V. Vshivkov	(Russian Academy of Sciences)

Organizing Committee

V. Malyshkin	Co-chairman (Novosibirsk)
V. Shaidurov	Co-chairman (Krasnoyarsk)
S. Achasova	Secretary (Novosibirsk)
O. Bandman	Publication Chair (Novosibirsk)
S. Isaev	Member (Krasnoyarsk)
F. Kazakov	Member (Krasnoyarsk)
N. Kuchin	Member (Novosibirsk)
A. Legalov	Member (Krasnoyarsk)
A. Malyshev	Member (Krasnoyarsk)
Yu. Medvedev	Member (Novosibirsk)
S. Nechaev	Member (Novosibirsk)
O. Nechaeva	Member (Novosibirsk)
G. Sadovskaya	Member (Krasnoyarsk)
E. Veysov	Member (Krasnoyarsk)

Referees

G. Acher
M. Alt
F. Arbab
T. Bair
S. Bandini
O. Bandman
H. Bischof
C. Blanchet
C. Bodei
N. Busi
E. Caron
Y. Caniou
T. Casavant
D. Chaly
A. Chambarel
D. Clarke
D. Defour
P. Degano
F. Desprez
J. Duennweber
D. Etiemble
P. Faber
G. Fedak
K. Fuerlinger
A. Giersch
A. Glebovsky
B. Goossens
A. Gorlatch
M. Gorodnichev
T. Hérault
A. Hurson
A. Iamnitchi
E. Jeannot
Y. Karpov
T. Klostermann
T. Klug
B. Lecussan
E. Kuzmin
R. Leshchinskiy
J. Li
O. Lodygensky
T. Ludwig
A. Maggiolo-Schettini
N. Malyshkin
V. Malyshkin
G. Mauri
H. Mehammed
J. Mueller
A. Nepomniaschaja
L. Pagli
G. Papodopoulos
M. Raynal
L. Ricci
Y. Robert
L. Rosaz
B. Roux
V. Shaidurov
G. Silberman
V. Sokolov
A. Solopov
A. Starita
D. Stodden
E. Timofeev
P. Trifonov
C. Trinitis
A. Tsigulin
M. Valero
V. Valkovskii
L. Vanneschi
I. Virbitskaite
V. Vshivkov
M. Walter
J. Weidendorfer
J. Zola

Lecture Notes in Computer Science

For information about Vols. 1–3516

please contact your bookseller or Springer

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3638: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 269 pages. 2005.

Vol. 3633: C. Bauzer Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005. (Subseries LNAI).

Vol. 3623: M. Liśkiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory. XV, 576 pages. 2005.

Vol. 3621: V. Shoup (Ed.), Advances in Cryptology – CRYPTO 2005. XI, 568 pages. 2005.

Vol. 3619: X. Lu, W. Zhao (Eds.), Networking and Mobile Computing. XXIV, 1299 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005. (Subseries LNBI).

Vol. 3608: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures. XIV, 446 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005. (Subseries LNAI).

Vol. 3606: V. Malyshkin (Ed.), Parallel Computing Technologies. XII, 470 pages. 2005.

Vol. 3602: R. Eigenmann, Z. Li, S.P. Midkiff (Eds.), Languages and Compilers for High Performance Computing. IX, 486 pages. 2005.

Vol. 3598: H. Murakami, H. Nakashima, H. Tokuda, M. Yasumura, Ubiquitous Computing Systems. XIII, 275 pages. 2005.

Vol. 3597: S. Shimojo, S. Ichii, T.W. Ling, K.-H. Song (Eds.), Web and Communication Technologies and Internet-Related Social Issues - HSI 2005. XIX, 368 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005. (Subseries LNAI).

Vol. 3595: L. Wang (Ed.), Computing and Combinatorics. XVI, 995 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005. (Subseries LNBI).

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005. (Subseries LNAI).

Vol. 3586: A.P. Black (Ed.), ECOOP 2005 - Object-Oriented Programming. XVII, 631 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005. (Subseries LNAI).

Vol. 3583: R.W. H. Lau, Q. Li, R. Cheung, W. Liu (Eds.), Advances in Web-Based Learning – ICWL 2005. XIV, 420 pages. 2005.

Vol. 3582: J. Fitzgerald, I.J. Hayes, A. Tarlecki (Eds.), FM 2005: Formal Methods. XIV, 558 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005. (Subseries LNAI).

Vol. 3580: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming. XXV, 1477 pages. 2005.

Vol. 3579: D. Lowe, M. Gaedke (Eds.), Web Engineering. XXII, 633 pages. 2005.

Vol. 3578: M. Gallagher, J. Hogan, F. Maire (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2005. XVI, 599 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005. (Subseries LNAI).

Vol. 3576: K. Etessami, S.K. Rajamani (Eds.), Computer Aided Verification. XV, 564 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005. (Subseries LNAI).

Vol. 3574: C. Boyd, J.M. González Nieto (Eds.), Information Security and Privacy. XIII, 586 pages. 2005.

Vol. 3573: S. Etalle (Ed.), Logic Based Program Synthesis and Transformation. VIII, 279 pages. 2005.

Vol. 3572: C. De Felice, A. Restivo (Eds.), Developments in Language Theory. XI, 409 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005. (Subseries LNAI).

Vol. 3570: A. S. Patrick, M. Yung (Eds.), Financial Cryptography and Data Security. XII, 376 pages. 2005.

Vol. 3569: F. Bacchus, T. Walsh (Eds.), Theory and Applications of Satisfiability Testing. XII, 492 pages. 2005.

Vol. 3568: W.-K. Leow, M.S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, E.M. Bakker (Eds.), Image and Video Retrieval. XVII, 672 pages. 2005.

Vol. 3567: M. Jackson, D. Nelson, S. Stirk (Eds.), Database: Enterprise, Skills and Innovation. XII, 185 pages. 2005.

Vol. 3566: J.-P. Banâtre, P. Fradet, J.-L. Giavitto, O. Michel (Eds.), Unconventional Programming Paradigms. XI, 367 pages. 2005.

Vol. 3565: G.E. Christensen, M. Sonka (Eds.), Information Processing in Medical Imaging. XXI, 777 pages. 2005.

Vol. 3564: N. Eisinger, J. Małuszyński (Eds.), Reasoning Web. IX, 319 pages. 2005.

Vol. 3562: J. Mira, J.R. Álvarez (Eds.), Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Part II. XXIV, 636 pages. 2005.

Vol. 3561: J. Mira, J.R. Álvarez (Eds.), Mechanisms, Symbols, and Models Underlying Cognition, Part I. XXIV, 532 pages. 2005.

Vol. 3560: V.K. Prasanna, S. Iyengar, P.G. Spirakis, M. Welsh (Eds.), Distributed Computing in Sensor Systems. XV, 423 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005. (Subseries LNAI).

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005. (Subseries LNAI).

Vol. 3557: H. Gilbert, H. Handschuh (Eds.), Fast Software Encryption. XI, 443 pages. 2005.

Vol. 3556: H. Baumeister, M. Marchesi, M. Holcombe (Eds.), Extreme Programming and Agile Processes in Software Engineering. XIV, 332 pages. 2005.

Vol. 3555: T. Vardanega, A.J. Wellings (Eds.), Reliable Software Technology – Ada-Europe 2005. XV, 273 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005. (Subseries LNAI).

Vol. 3553: T.D. Hämäläinen, A.D. Pimentel, J. Takala, S. Vassiliadis (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XV, 476 pages. 2005.

Vol. 3552: H. de Meer, N. Bhatti (Eds.), Quality of Service – IWQoS 2005. XVIII, 400 pages. 2005.

Vol. 3551: T. Härder, W. Lehner (Eds.), Data Management in a Connected World. XIX, 371 pages. 2005.

Vol. 3548: K. Julisch, C. Kruegel (Eds.), Intrusion and Malware Detection and Vulnerability Assessment. X, 241 pages. 2005.

Vol. 3547: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. XIII, 588 pages. 2005.

Vol. 3546: T. Kanade, A. Jain, N.K. Ratha (Eds.), Audio- and Video-Based Biometric Person Authentication. XX, 1134 pages. 2005.

Vol. 3544: T. Higashino (Ed.), Principles of Distributed Systems. XII, 460 pages. 2005.

Vol. 3543: L. Kutvonen, N. Alonistioti (Eds.), Distributed Applications and Interoperable Systems. XI, 235 pages. 2005.

Vol. 3542: H.H. Hoos, D.G. Mitchell (Eds.), Theory and Applications of Satisfiability Testing. XIII, 393 pages. 2005.

Vol. 3541: N.C. Oza, R. Polikar, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. XII, 430 pages. 2005.

Vol. 3540: H. Kalviainen, J. Parkkinen, A. Kaarna (Eds.), Image Analysis. XXII, 1270 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005. (Subseries LNAI).

Vol. 3538: L. Ardissono, P. Brna, A. Mitrovic (Eds.), User Modeling 2005. XVI, 533 pages. 2005. (Subseries LNAI).

Vol. 3537: A. Apostolico, M. Crochemore, K. Park (Eds.), Combinatorial Pattern Matching. XI, 444 pages. 2005.

Vol. 3536: G. Ciardo, P. Darondeau (Eds.), Applications and Theory of Petri Nets 2005. XI, 470 pages. 2005.

Vol. 3535: M. Steffen, G. Zavattaro (Eds.), Formal Methods for Open Object-Based Distributed Systems. X, 323 pages. 2005.

Vol. 3534: S. Spaccapietra, E. Zimányi (Eds.), Journal on Data Semantics III. XI, 213 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005. (Subseries LNAI).

Vol. 3532: A. Gómez-Pérez, J. Euzenat (Eds.), The Semantic Web: Research and Applications. XV, 728 pages. 2005.

Vol. 3531: J. Ioannidis, A. Keromytis, M. Yung (Eds.), Applied Cryptography and Network Security. XI, 530 pages. 2005.

Vol. 3530: A. Prinz, R. Reed, J. Reed (Eds.), SDL 2005: Model Driven. XI, 361 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005. (Subseries LNAI).

Vol. 3527: R. Morrison, F. Oquendo (Eds.), Software Architecture. XII, 263 pages. 2005.

Vol. 3526: S. B. Cooper, B. Löwe, L. Torenvliet (Eds.), New Computational Paradigms. XVII, 574 pages. 2005.

Vol. 3525: A.E. Abdallah, C.B. Jones, J.W. Sanders (Eds.), Communicating Sequential Processes. XIV, 321 pages. 2005.

Vol. 3524: R. Barták, M. Milano (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. XI, 320 pages. 2005.

Vol. 3523: J.S. Marques, N. Pérez de la Blanca, P. Pina (Eds.), Pattern Recognition and Image Analysis, Part II. XXVI, 733 pages. 2005.

Vol. 3522: J.S. Marques, N. Pérez de la Blanca, P. Pina (Eds.), Pattern Recognition and Image Analysis, Part I. XXVI, 703 pages. 2005.

Vol. 3521: N. Megiddo, Y. Xu, B. Zhu (Eds.), Algorithmic Applications in Management. XIII, 484 pages. 2005.

Vol. 3520: O. Pastor, J. Falcão e Cunha (Eds.), Advanced Information Systems Engineering. XVI, 584 pages. 2005.

Vol. 3519: H. Li, P. J. Olver, G. Sommer (Eds.), Computer Algebra and Geometric Algebra with Applications. IX, 449 pages. 2005.

Vol. 3518: T.B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005. (Subseries LNAI).

Vol. 3517: H.S. Baird, D.P. Lopresti (Eds.), Human Interactive Proofs. IX, 143 pages. 2005.

Table of Contents

On Evaluating the Performance of Security Protocols*

Chiara Bodei[1], Mikael Buchholtz[3], Michele Curti[1], Pierpaolo Degano[1], Flemming Nielson[3], Hanne Riis Nielson[3], and Corrado Priami[2]

[1] Dipartimento di Informatica, Università di Pisa,
Largo B.Pontecorvo, 3, I-56127 Pisa, Italy
{chiara, curtim, degano}@di.unipi.it
[2] Dipartimento di Informatica e Telecomunicazioni,
Università di Trento, Via Sommarive, I-1438050 Povo (TN), Italy
priami@science.unitn.it
[3] Informatics and Mathematical Modelling, Technical University of Denmark,
Richard Petersens Plads bldg 321, DK-2800 Kongens Lyngby, Denmark
{mib, nielson, riis}@imm.dtu.dk

Abstract. We use an enhanced operational semantics to infer quantitative measures on systems describing cryptographic protocols. System transitions carry enhanced labels. We assign rates to transitions by only looking at these labels. The rates reflect the distributed architecture running applications and the use of possibly different crypto-systems. We then map transition systems to Markov chains and evaluate performance of systems, using standard tools.

1 Introduction

Cryptographic protocols are used in distributed systems for authentication and key exchange, and must therefore guarantee security. The mechanisms used are always the result of a judicious balance between their cost and benefits. Performance costs, in terms of time overhead and resource consumption, must be carefully evaluated when choosing security mechanisms.

Here, we extend a preliminary idea introduced in [6] for the development of a single, formal design methodology that supports designers in analysing the performance of protocols, with a semi-mechanizable procedure. We provide a general framework, where quantitative aspects, symbolically represented by parameters, can be formally estimated. By changing only these parameters on the architecture and the algorithm chosen, one can compare different implementations of the same protocol or different protocols. This allows the designer to choose among different alternatives, based on an evaluation of the trade-off between security guarantees and their price.

We are mainly interested in evaluating the cost of each cryptographic operation and of each message exchange. Here, "cost" means any measure of quantitative properties such as speed, availability, etc.

* Supported in part by the EU IST-2001-32072 project DEGAS.

V. Malyshkin (Ed.): PaCT 2005, LNCS 3606, pp. 1–15, 2005.

Usually protocols are described through informal narrations. These narrations include only a list of the messages to be exchanged, leaving it unspecified which are the actions to be performed in receiving these messages (inputs, decryptions and possible checks on them). This can lead, in general, to an inaccurate estimation of costs. The above motivates the choice of using the process algebra LySA [3,5], a close relative of the π- [24] and Spi-calculus [1], that details the protocol narration, in that outputs and the corresponding inputs are made explicit and similarly for encryptions and the corresponding decryptions. Also, LySA is explicit about which keys are fresh and about which checks are to be performed on the received values. More generally, LySA provides us with a unifying framework, in which security protocols can be specified and statically analysed [3,5] through Control Flow Analysis. This analysis, fully automatic and always terminating, is strong enough to report known flaws on a wide range of protocols, and even to find new ones [4].

Technically, we give LySA (Sect. 2) an enhanced semantics, following [14], and then we associate rates to each transition, in the style of [26]. It suffices to have information about the activities performed by the components of a system in isolation, and about some features of the network architecture. We then mechanically derive Markov chains using these rates (Sect. 3). The actual performance evaluation is carried out using standard techniques and tools [33,31,32]. Significantly, quantitative measures, typically on cryptography, here live together with the usual qualitative semantics, where instead these aspects are usually abstracted away. Specifically, there exists a very early prototype, based on π-calculus, on which it is possible to run LySA, that we used for the case study presented here (Sect. 4), along with a standard mathematical tool such as Mathematica. Relative approaches are EMPA[8] and PEPA[19], to cite only a few.

In comparing different versions of the same protocol or different protocols, specified in LySA, our technique can be suitably integrated with the Control Flow one, to check security at the same stage.

Our framework can be extended [7] to estimate the cost of security attacks. The typical capabilities of the Dolev-Yao attacker [16] go beyond the ones a legitimate principal has. The needed model includes a set of the possible extra actions in which the attacker exploits its computational power and its capability of guessing (see also [10] and [23]). It would be interesting to deal with timing attacks as well, even though this may considerably complicate our model.

2 LySA and Its Enhanced Semantics

The LySA calculus [3,5] is based on the π- [24] and Spi-calculus [1], but differs from these essentially in two aspects: (i) the absence of channels: there is only one global communication medium to which all processes have access; (ii) the tests associated with input and decryption are naturally expressed using pattern matching. Below, we assume that the reader is familiar with the basics of process calculi.

Syntax. The syntax consists of terms $E \in \mathcal{E}$ and processes $P \in \mathcal{P}$,

$$E ::= a \mid x \mid \{E_1, \cdots, E_k\}_{E_0}$$
$$P ::= 0 \mid out.P \mid in.P \mid P_1 \mid P_2 \mid (\nu\, a)P \mid dec \text{ in } P \mid A(y_1, \ldots, y_n)$$

where we introduced the following abbreviations: • $out \triangleq \langle E_1, \cdots, E_k \rangle$, • $in \triangleq (E'_1, \cdots, E'_j; x_{j+1}, \cdots, x_k)$, • $dec \triangleq$ decrypt E as $\{E_1, \cdots, E_j; x_{j+1}, \cdots, x_k\}_{E_0}$.
Intuitively, the process 0 or *nil* represents the null inactive process. The operator | describes parallel composition of processes. The operator (νa) acts as a static declaration for the name a in the process P the restriction prefixes. Restriction is therefore used to create new names such as nonces or keys. The process $\langle E_1, \cdots, E_k \rangle.P$ sends $E_1, \cdots, E_k$ on the net and then continues like P. The process $(E_1, \cdots, E_j; x_{j+1}, \cdots, x_k).P$ receives the tuple $E'_1, \cdots, E'_k$ and continues as $P[E_{j+1}/x_{j+1}, \ldots, E_k/x_k]$, provided that $E_i = E'_i$ for all $i \in [1, j]$. The intuition is that the matching succeeds when the first j values E'_i pairwise correspond to the values E_i, and the effect is to bind the remaining $k - j$ values to the variables $x_{j+1}, \cdots, x_k$. Note that, syntactically, a semi-colon separates the components where matching is performed from those where only binding takes place. The same simple form of patterns is also used for decryption (see [9] for a more flexible choice). In fact, the process decrypt E as $\{E_1, \cdots, E_j; x_{j+1}, \cdots, x_k\}^{\text{in}}_{E_0}$ in P decrypts $E = \{E'_1, \cdots, E'_k\}_{E'_0}$ with the key E_0. Whenever $E_i = E'_i$ for all $i \in [0, j]$, the process behaves as $P[E_{j+1}/x_{j+1}, \ldots, E_k/x_k]$. Finally, an agent is a static definition of a parameterised process. Each agent identifier A has a unique defining equation of the form $A(\tilde{y}) = P$, where $\tilde{y}$ denotes a tuple $y_1, \ldots, y_n$ of distinct names occurring free in P.

Working Example. Consider the following basic Kerberos key agreement protocol [22] that is part of our case study. We assume that the AES algorithm [12] is the crypto-system used here.

$$\begin{array}{ll} & 1.\ A \rightarrow S :\ A, B \\ & 2.\ S \rightarrow A :\ \{B, T, L, K_{AB}\}_{K_A}, \{A, T, L, K_{AB}\}_{K_B} \\ (Kerberos) & 3.\ A \rightarrow B :\ \{A, T, L, K_{AB}\}_{K_B}, \{A, T\}_{K_{AB}} \\ & 4.\ B \rightarrow A :\ \{T, T\}_{K_{AB}} \end{array}$$

Intuitively, principal A asks the Key Distribution Center S for a session key to share with B. S generates the key K_{AB}, a timestamp T and lifetime L and produces an encryption of these components for A and another one for B, including the identity of the other principal. Both encryptions are sent to A, that can decrypt the first and forward the second to B, along with another encryption that A obtains by encoding (A, T) with the new key. B can decrypt the first encryption so to obtain K_{AB} then B decrypts the second encryption, and uses K_{AB} to encrypt (T, T) as a replay to A. To simplify, we use $\{T, T\}_{K_{AB}}$ rather than the usual $\{T + 1\}_{K_{AB}}$.

The protocol specification in LySA is in Tab. 1, where the right column reports a concise explanation of the action on the left, in terms of the number of the message (called *msg*, while *enc* stands for an encrypted term) in the protocol narration. The whole system is given by the parallel composition (||) of the three processes A, B, S. Each part of the system performs a certain number of actions and then restarts.

Table 1. Specification of *Kerberos* Protocol

1 $Sys_1 = (\nu K_A)(\nu K_B)((A\|B)\|S)$	K_A, K_B long-term keys
2 $A = (\langle A, B\rangle).\, A'$	A sends msg (1)
4 $A' = (;\, v^A_{enc}, v^B_{enc}).\, A''$	A receives and checks msg (2)
5 $A'' =$ decrypt v^A_{enc} as $\{B;\, v_T, v_L, v_K\}_{K_A}$ in A'''	A decrypts the enc in msg (2)
6 $A''' = \langle v^B_{enc}, \{A, v_T\}_{v_K}\rangle.\, A''''$	A sends msg (3)
7 $A'''' = (;\, w^A_{enc}).\, A'''''$	A receives and checks msg (4)
8 $A''''' =$ decrypt w^A_{enc} as $\{v_T, v_T;\,\}_{v_K}$ in A	A decrypts the enc in msg (4)
9 $B = (;\, z^1_{enc}, z^2_{enc}).\, B'$	B receives and checks msg (3)
10 $B' =$ decrypt z^1_{enc} as $\{;\, z_A, z_T, z_L, z_K\}_{K_B}$ in B''	B decrypts the 1^{st} enc in msg (3)
11 $B'' =$ decrypt z^2_{enc} as $\{z_A, z_T;\,\}_{z_K}$ in B'''	B decrypts the 2^{nd} enc in msg (3)
12 $B''' = \langle\{z_T, z_T\}_{z_K}\rangle.\, B$	B sends msg (4)
13 $S = (;\, y^A, y^B).\, S'$	S receives and checks msg (1)
14 $S' = (\nu K_{AB})(\nu T)(\nu L)$	K_{AB} fresh session key
15 $(\langle\{y^B, T, L, K_{AB}\}_{K_A}, \{y^A, T, L, K_{AB}\}_{K_B}\rangle.\, S)$	S sends msg (2)

Enhanced Operational Semantics. Here, we give a concrete version of operational semantics, called *enhanced* in the style of [13,14]. Our enhanced semantics for LySA is a reduction semantics, built on top of the standard reduction semantics [3], where both processes and transitions are annotated with labels that will be helpful for computing costs.

Formally, each transition is enriched with an *enhanced label* θ which records both the action corresponding to the transition and its syntactic context. Actually, the label of a communication transition records the two actions (input and output) that lead to the transition. To facilitate the definition of our reduction semantics, for each given process, we annotate each of its sub-processes P with an encoding of the context in which P occurs. The encoding is a string of tags ϑ, that essentially record the syntactic position of P w.r.t. the parallel composition nesting. To do this, we exploit the abstract syntax tree of processes, built using the binary parallel composition as operator. We introduce a tag $\|_0$ ($\|_1$, resp.) for the left (for the right, resp.) branch of a parallel composition. Labels are defined as follows.

Definition 1. *Let* $\mathcal{L} = \{\|_0, \|_1\}$. *Then, the set of* context labels *is defined as* $\mathcal{L}^*$, *i.e. the set of all the string generated by* $\mathcal{L}$, *ranged over by* ϑ.

We choose to have tags concerned with the parallel structure of processes, i.e. linked to parallel composition "|". For our present purpose, this is the only necessary annotation (for other annotations, see [26,14]).

Technically, labelled processes are inductively obtained in a pre processing step, by using the function $\mathcal{T}$. This function (inductively) prefixes actions with context labels: $\mathcal{T}$ unwinds the syntactic structure of processes, until reaching a 0 or a constant. Given a process P, this transformation operates in linear time with the number of prefixes. Note that this pre-processing step can be