USING PROCESS CALCULI TO MODEL AND VERIFY SECURITY PROPERTIES IN REAL LIFE COMMUNICATION PROTOCOLS – Thesis Summary

Andrés Alberto ARISTIZÁBAL Hugo Andrés LÓPEZ

July 10, 2006

Thesis Information

- Work carried out within the AVISPA research group. Pontificia Universidad Javeriana - Cali.
- Supervisors:
  - Prof. Camilo RUEDA. Dean of the Department of Sciences and Engineering of Computing, Pontificia Universidad Javeriana, Cali, Colombia.
  - Dr. Frank D. Valencia, CNRS Associate Research Scientist at Laboratoire d'Informatique (LIX), École Polytechnique de Paris, France.
- Submitted: January 25, 2006. Defended: February 2, 2006. Grade 5.0/5.0.
- Two papers published with partial results of this dissertation [1], [2].
- Laureated for the achievements obtained, May 2006.

Security of information has always been one of the main concerns in social behavior. The assurance of a personal secret which cannot be revealed to someone unauthorized, and the notion of trust have been relevant concerns since the beginnings of commerce and wars. The emergence of global communications, electronic processing, and distributed computation have increased the relevance of these concerns. To overcome these security risk, a wide variety of (automated) tools have been developed, including firewalls, access control mechanisms and cryptographic-based software. These mechanisms by themselves, however, are not enough to provide security warranties; the open nature of the communications, and the inherent vulnerabilities of distributed systems makes essential to provide higher levels of assurance for principals involved in a privacy-sensitive communication process. As a response to this problem, a set of methods known as *security protocols* have arisen: they define a precise set of steps that principals have to follow in order to establish secure communication between parties involved.

Security protocols have been widely used since its appearance, being at the heart of a huge amount of computer applications. However, we can never be confident over the security of a system unless we have some assurance of their effectiveness. As an example, one of the classical methods dates from 1978 when Roger Nedham and Michael Schröeder designed a protocol to prevent the disclosure of identities in an authentication process over untrusted networks such as the internet [3]. The protocol, apparently correct, was rapidly adopted in industrial and military applications until Gavin Lowe showed a flaw where messages in transit can be discovered and manipulated using a well defined set of steps [4]. With these results, one can question: How to ensure the correctness of a protocol?

Formal methods constitute an analytical approach for software and hardware design, that intends the reduction of errors by relying on solid mathematical models. One of the major benefits of formal methods

is that they offer reasoning techniques that cover every possible state of a design, and the inclusion of well-defined proof techniques which ensure the accuracy and correctness of a design. The generality of formal methods contrasts with the ad-hoc spirit present in other approaches, such as empirical analysis and simulations. *Process calculi* constitute a particular class of formal languages, specially oriented to the analysis of concurrent systems. The main idea underlying process calculi is the abstraction of real systems in terms of basic units known as *processes*. The calculi provide precise elements to describe systems as a combination of processes, as well as offer tools to study the behavior of systems over time.

Consequently, process calculi appear as convenient tools to give a formal flavor to complex, concurrent computing systems. Several process calculi have been proposed over the last twenty years [5, 6, 7, 8, 9, 10]: although they differ on particular aspects for understanding communications, all of them agree on the basic principles given above. Following an interesting evolution, in the last ten years process calculi have *particularized* in specific domain areas. In this way, for instance, several process calculi tailored for modeling biological phenomena or artistic applications have been proposed [11, 12, 13, 14]. Similarly, security has been a particular active area in this recent evolution: diverse process calculi, offering alternatives to the problem of modeling and verifying secure communications, have been proposed. Instances of these calculi include the π and the Spi calculus [7, 15], the CSP process algebra [9], and more recently, the secure protocol language (SPL) [16].

Our thesis explores the use of a process calculi in the analysis and verification of security protocols, providing an analysis of recently proposed models and tools, as well as contrasting their applicability in the modeling and verification of real world communicating systems. In particular, we focus on the study of communication protocols in Peer-to-Peer (P2P) systems. These systems, usually operating over open and distributed networks, take advantage of vast communication networks to accomplish diverse tasks in a very flexible manner. However, the inherent ubiquity of P2P communication systems makes them prone to serious security vulnerabilities, such attacks outside and inside their trusted networks. Therefore, formally studying security vulnerabilities in P2P communication networks is a relevant task, both from the practical and theoretical standpoint. Very surprisingly, we find that little work tackles this problem [17, 18] .

Our work intends to give concrete contributions in this context by studying two P2P communication protocols using formal languages. Firstly, we present a comparative analysis on the expressiveness of a representative set of process calculi. This analysis led to the selection of Crazzolara and Winkel's SPL as the more appropriate formalism for carrying out security analysis over dynamic networks. Despite of its simplicity, this process calculi counts with the necessary primitives to model security protocols, Communication in SPL is inherently asynchronous, and occurs in the context of a public, monotonic pool of messages that preserves transmitted information. This basic conceptual construct for persistent information allows to represent an spy capable to overhear the messages over the network for an infinite period of time. A fundamental issue that contributed to the selection of SPL were its associated tools for verification. In fact, SPL provides an event-based semantics, where processes can be decomposed in atomic actions by means of models related to persistent Petri Nets. Such a semantics faithfully captures casual dependencies between events which, in turn, are the basic information for discovering attacks and flaws in protocols' specifications. The generality of this approach is generalized as a set of proof principles for SPL; they formalize intuitions underlying the nature of the event based semantics, allowing for the definition of different kinds of proofs without further language extensions.

With this selection, we proceed with our analysis by selecting two different cases of study of P2P systems. The first one pertains to the analysis of MUTE [19], a system conceived for sharing resources over a dynamic network. Common services for file sharing such as those for music clearly relate to this system. We focused on the security analysis of the searching protocol underlying MUTE, which can be regarded as a general method of search in P2P systems. Here, the information is perceived as *secret* if its shared among trusted peers. We formalise this model verifying the property of secrecy over outsider attackers. Also, we extend the protocol in order to be resistant to a wider class of attackers that can masquerade as trusted peers, the so-called *insider attackers*. We proposed the inclusion of a mid-layer mechanism to control the distributed keys, verifying its correctness by means of the inherent proof techniques of SPL.

The second system analyzed is oriented to the dynamic reconfiguration of applications in collaborative

environments. We use a cutting-edge system as a valid case of study to achieve this affirmation. The Friends Troubleshooting Network (FTN) protocol proposed by Wang et al [20] is intended to resolve the problem of automatic reconfiguration of applications in a fully distributed system without compromising the identities of the agents involved in the protocol, neither their own secrets. In this case we follow a two-fold approach: Firstly, we extend the basic syntactic structure of SPL with some notions of concurrency to formalize an SPL model for the FTN protocol, basically by providing a set of encodings to model concepts such as Exclusive Non-deterministic choice, scheduling and clustering. Secondly, we propose a new protocol that maintains the main functionality of the FTN in a model which is more succinct and less complex than the proposed previously. In order to do so, we exploit the idea of a layered encryption protocol [21], providing proofs of its correctness in terms of the secrecy of the messages transmitted and the integrity of the transmissions.

Summing up, this thesis presents direct contributions in the state of the art in process calculi by giving a concrete idea of its applicability in the modelling new kind of concurrent behaviors such the ones exemplified by P2P systems. We support this contribution by modelling a set of cases of study with a simple yet powerful process calculi, extending it in order to provide a broader set of constructions for concurrent behaviors. Also, security flaws of current protocols for P2P systems have been found, proposing new protocols designed to overcome previous security threats, providing proofs of correctness by means of the process language. We expect this research work will motivate the developers to continuing using formal approaches on the design of critical tasks such as those of specifications of security applications, taking advantages of the their solid foundations and their vast variety of (automatic or semi-automatic) formal frameworks [22, 23, 24].

References

[1] A. Aristizábal, H. A. López, and C. Rueda, "Using a declarative process language for P2P protocols," *The Association for Logic Programming Newsletter Digest*, vol. 18, November 2005.

[2] A. Aristizábal, H. A. López, F. D. Valencia, and C. Rueda, "Formally reasoning about security issues in p2p protocols: A case study.," in *Third Taiwanese-French Conference on Information Technology (TFIT)* (S. Cruz-Lara and Y.-K. Tsay, eds.), INRIA Technical Reports, pp. 577 – 598, INRIA, 2006.

[3] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[4] G. Lowe, "An attack on the needham-schroeder public-key authentication protocol," *Inf. Process. Lett.*, vol. 56, no. 3, pp. 131–133, 1995.

[5] G. D. Plotkin, "A structural approach to operational semantics," tech. rep., University of Aarhus, 1981.

[6] R. Milner, *Communication and concurrency*. Hertfordshire, UK, UK: Prentice Hall International (UK) Ltd., 1995.

[7] R. Milner, *Communicating and Mobile systems. The Pi Calculus*. Cambridge University Press, 1999.

[8] L. Cardelli and A. D. Gordon, "Mobile ambients," in *Foundations of Software Science and Computation Structures: First International Conference, FOSSACS '98*, Springer-Verlag, Berlin Germany, 1998.

[9] C. A. R. Hoare, "Communicating Sequential Processes," *Commun. ACM*, vol. 26, no. 1, pp. 100–106, 1983.

[10] V. S. M. Rinard and P. Panangaden, "The semantic foundations of concurrent constraint programming," in *POPL '91*, pp. 333–352, jan 1991.

[11] J. Krivine and V. Danos, "Formal molecular biology done in CCS-R," in *BioConcur 2003, Workshop on Concurrent Models in Molecular Biology*, 2003.

[12] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Y. Shapiro, "Bioambients: an abstraction for biological compartments.," *Theor. Comput. Sci.*, vol. 325, no. 1, pp. 141–167, 2004.

[13] L. Cardelli, "Brane calculi," in *CMSB* (V. Danos and V. Schachter, eds.), vol. 3082 of *Lecture Notes in Computer Science*, pp. 257–278, Springer, 2004.

[14] C. Rueda and F. D. Valencia, "On validity in modelization of musical problems by ccp," *Soft. Computing*, vol. 8, no. 9, pp. 641–648, 2004.

[15] M. Abadi and A. D. Gordon, "A calculus for cryptographic protocols: The spi calculus.," *Inf. Comput.*, vol. 148, no. 1, pp. 1–70, 1999.

[16] F. Crazzolara and G. Winskel, "Events in security protocols," in *ACM Conference on Computer and Communications Security*, pp. 96–105, 2001.

[17] T. Chothia, "Analysing the mute anonymous file-sharing system using the pi-calculus," in *26th Conference on Formal Methods for Networked and Distributed Systems*, LNCS, 2006.

[18] J. Borgström, U. Nestmann, L. O. Alima, and D. Gurov, "Verifying a structured peer-to-peer overlay network: The static case," in *Proc. Global Computing 2004* (TBD, ed.), vol. TBD of *Lecture Notes in Computer Science*, p. TBD, Springer, 2004.

[19] J. Rohrer and M. Roth., "Mute: Simple, anonymous file sharing," 2005. Available at `http://mute-net.sourceforge.net/howAnts.shtml`.

[20] H. J. Wang, Y.-C. Hu, C. Yuan, Z. Zhang, and Y.-M. Wang, "Friends troubleshooting network: Towards privacy-preserving, automatic troubleshooting.," in *IPTPS* (G. M. Voelker and S. Shenker, eds.), vol. 3279 of *Lecture Notes in Computer Science*, pp. 184–194, Springer, 2004.

[21] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Communications of the ACM (USA)*, vol. 42, no. 2, pp. 39–41, 1999.

[22] F. Crazzolara and G. Milicia, "A framework for the development of protocols," in *Proceeding of the 3rd International Conference on Application of Concurrency to System Design (ACSD 2003)*, pp. 239–240, 2003. Tool Demo.

[23] B. Blanchet, M. Abadi, and C. Fournet, "Automated Verification of Selected Equivalences for Security Protocols," in *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, (Chicago, IL), pp. 331–340, IEEE Computer Society, June 2005.

[24] L. Viganò, "Automated security protocol analysis with the avispa tool," in *XXI Mathematical Foundations of Programming Semantics (MFPS'05)*, no. 155 in ENTCS, pp. 61–86, Elsevier, 2006.