

Towards an integrated formal analysis for security and trust

Fabio Martinelli

Istituto di Informatica e Telematica - C.N.R., Pisa, Italy

1 Abstract

We aim on defining an integrated framework for the (automated) analysis for security and trust in complex and dynamic scenarios.

In the last years, several formal languages for describing distributed and communicating protocols have been refined to model at a high level of abstraction some specific features of cryptographic protocols.

Cryptography is usually modeled by representing encryptions as terms of an algebra, e.g., $E(m, k)$ may represent the encryption of a message m with a key k . Usually, the so-called perfect encryption abstraction is adopted: encryptions are considered as injective functions which can be inverted only by knowing the correct information, i.e. the decryption key. For instance, common inference rules for modeling the behavior of the encryption and decryption (in a shared-key schema) are the followings:

$$\frac{m \quad k}{E(m, k)} \quad \frac{E(m, k) \quad k}{m} \quad (1)$$

which should be read as: from a message m and a key k we can build the encryption $E(m, k)$; from an encryption $E(m, k)$ and a decryption key k we can obtain the encrypted message m .

The long standing tradition of modeling the specific features of cryptographic functions as term-rewriting rules met the powerful verification techniques developed for process algebras. As a matter of fact, several formal languages for describing communication protocols, for instance CSP [7], have been exploited for representing cryptographic protocols without changes in syntax or semantics: the inference rules have been given at the meta-level of the verification. Instead others, like the π -calculus [1] and the CCS [8, 10], have been effectively refined: the π -calculus have been equipped with two pattern matching constructs for modeling message splitting and shared-key decryption, respectively; the CCS has been equipped with a term-rewriting construct that permits to infer new messages from others, i.e.:

$$[m_1 \quad m_n \vdash_r x].P$$

which denotes a process that tries to deduce a message m from the messages in m_1, \dots, m_n and when it succeeds it substitutes this message for x in the process specification P . The language is called Crypto-CCS ([8]).

Formally, we may consider a set of relations among closed messages as: $\vdash_r \subseteq \mathcal{P}^{fin}(\mathcal{M}) \times \mathcal{M}$, where r is the name of the rule. Given a set \mathcal{R} of inference rules, we consider the deduction relation $\mathcal{D}^{\mathcal{R}} \subseteq \mathcal{P}^{fin}(\mathcal{M}) \times \mathcal{M}$. Given a finite set of closed messages, say ϕ , then $(\phi, M) \in \mathcal{D}^{\mathcal{R}}$ if M can be derived by iteratively applying the rules in \mathcal{R} . For the sake of simplicity, we assume that \vdash_r (for each $r \in \mathcal{R}$) and $\mathcal{D}^{\mathcal{R}} \subseteq \mathcal{P}^{fin}(\mathcal{M}) \times \mathcal{M}$ are decidable. Such inference systems allow us to cope with the variety of different crypto-systems that can be found in the literature.

However, when one analyzes a security protocol, usually assumes that public keys, digital certificates, and generally speaking credentials are already given, and does not check how these are formatted/managed. Such a limited view seems not enough for dynamic, fully interconnected systems, where access control policies may change and typically may also depend on credentials presented by users.

Similarly, when one wishes to formally analyze (e.g., see [2]) access control systems, the authentication mechanisms (usually a security protocol) is given for “secure”, without further specification.

The interplay between security protocols and access control mechanisms/policies is crucial. A good analysis framework should take an holistic point of view.

As a matter of fact, it is worthy noticing that the idea proposed by CryptoCCS of using inference constructs is also useful to model access control mechanisms based on credentials in distributed systems (e.g., see [12, 4]).

Example 1. Indeed, consider a set of credentials, i.e. (signed) messages containing information about access rights. Assume that $\{A, ob_1, +\}_{pr(C)}$ means that the user C (via the signature with its private key $pr(C)$) asserts A has the right to access the object ob_1 and may grant this access to other users (this is denoted through the symbol $+$). A rule like:

$$\frac{\{A, ob_1, +\}_{pr(C)} \quad pr(C) \quad \{grant \ B, ob_1\}_{pr(A)}}{\{B, ob_1, +\}_{pr(C)}} (acc_C)$$

may be used by the controller C to issue other access right credentials, after receiving an indication by A , i.e. the signed message $\{grant \ B, ob_1\}_{pr(A)}$.

Thus, we may also consider the inference rules as an abstract mechanism to express security policies usually defined using other mathematical models and logics (e.g., see [5, 12]). Moreover, it is also possible to encode with inference systems the mechanisms for reasoning about trust proposed in [6]. Having a unique language will allow us to model the interplay between security protocols that use the trust relationships among different users, and the ways in which these relationships are created (that often rely on security/interaction protocols).

The fact that we can both model cryptography and some form of credential/trust management with the inference construct of CryptoCCS allows us to use the software tools and methodologies already developed for security protocols analysis to the more general case where credentials are explicitly managed. In particular, in [11] a software tool for automated security protocols analysis has been defined and in [9] has been extended to cope with a huge class of inference systems.

It is worthy noticing that the CryptoCCS has been previously defined to set up a uniform framework for the analysis of security properties and information flow (non-interference) with the same machinery (e.g., see [3]).

To sum up, the flexibility of the inference construct as a modeling tool may allow us to study and analyze uniformly several aspects of network/system security and trust.

Research overview. This line of research is supported by the project (ready to start) "Model-based design and validation for web services" funded by CSP and we are currently working on a project proposal called "Trusted e-Services for Dynamic Coalitions" that should be funded by CNR. In this proposal we try to uniformly investigate issues of Business processes (Web services), GRID, Autonomous agents, Mobile ad hoc networks. Many of these aspects are covered by the Information Security Group of IIT-CNR. These projects continue our long-term research goal of promoting a cross fertilization between trust and security. In this line of thought, we started together Theo Dimitrakos, a workshop on Formal Aspects in Security and Trust (see www.iit.cnr.it/FAST2003 - the next edition will be affiliated with 18th IFIP WCC2004).

References

- [1] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 1999.
- [2] P. A. Bonatti and P. Samarati. Logics for authorizations and security. In *Logics for Emerging Applications of Databases*, LNCS. Springer-Verlag, 2003.
- [3] R. Focardi, R. Gorrieri, and F. Martinelli. Non interference for the analysis of cryptographic protocols. In *Proceedings of 27th International Colloquium in Automata, Languages and Programming*, volume 1853 of *Lectures Notes in Computer Science*, pages 354–372, 2000.
- [4] R. Gorrieri and F. Martinelli. Process algebraic frameworks for the specification and analysis of cryptographic protocols. In *MFCS*, LNCS 2747. Springer-Verlag, 2003.
- [5] Halpern and van der Meyden. A logic for SDSI's linked local name spaces. In *PSCFW: Proceedings of The 12th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1999.
- [6] A. Jsang, E. Gray, and M. Kinatader. Analysing topologies of transitive trust. In *Proc. of the 1st workshop on Formal Aspects in Security and Trust (FAST2003)*, 2003.
- [7] G. Lowe. Breaking and fixing the Needham Schroeder public-key protocol using FDR. In *Proceedings of Tools and Algorithms for the Construction and the Analysis of Systems*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer Verlag, 1996.
- [8] F. Martinelli. Languages for description and analysis of authentication protocols. In P. Degano and U. Vaccaro, editors, *Proceedings of 6th Italian Conference on Theoretical Computer Science*, pages 304–315. World Scientific, 1998.

- [9] F. Martinelli. Symbolic semantics and analysis for crypto-ccs with (almost) generic inference systems. In *Proceedings of the 27th international Symposium in Mathematical Foundations of Computer Sciences(MFCS'02)*, volume 2420 of *LNCS*, pages 519–531, 2002.
- [10] F. Martinelli. Analysis of security protocols as *open* systems. *Theoretical Computer Science*, 290(1):1057–1106, 2003.
- [11] F. Martinelli, M. Petrocchi, and A. Vaccarelli. PaMoChSA: A tool for verification of security protocols based on partial model checking. 2001. Tool Demo at the 1st International School on Formal Methods for the Design of Computer, Communication and Software Systems: Process Algebras.
- [12] P. Samarati and S. D. C. di Vimercati. Access control: Policies, models, and mechanisms. In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design*, LNCS 2171. Springer-Verlag, 2001.