

Formal Methods and Cryptography

Michael Backes¹, Birgit Pfizmann², and Michael Waidner³

¹ Saarland University, Saarbrücken, Germany, backes@cs.uni-sb.de

² IBM Research, Rueschlikon, Switzerland, bpf@zurich.ibm.com

³ IBM Software Group, Somers, NY, USA, wmi@zurich.ibm.com

Abstract. Security-critical systems are an important application area for formal methods. However, such systems often contain cryptographic subsystems. The natural definitions of these subsystems are probabilistic and in most cases computational. Hence it is not obvious how one can treat cryptographic subsystems in a sound way within formal methods, in particular if one does not want to encumber the proof of an overall system by probabilities and computational restrictions due only to its cryptographic subsystems.

We survey our progress on integrating cryptography into formal models, in particular our work on reactive simulatability (RSIM), a refinement notion suitable for cryptography. We also present the underlying system model which unifies a computational and a more abstract presentation and allows generic distributed scheduling. We show the relation of RSIM and other types of specifications, and clarify what role the classical Dolev-Yao (term algebra) abstractions from cryptography can play in the future.

1 Secure Channels as an Example of Cryptography within Larger Systems

Imagine you are using formal methods to prove the correctness of a distributed system with respect to an overall specification. This system uses SSL/TLS for secure communication between its components; this is a widely used standard for cryptographically protecting messages on otherwise insecure channels [10]. What does the use of SSL mean for the overall proof?

Clearly, the nicest solution would be if you would not need to bother that SSL is a cryptographic subsystem, but could simply abstract from it by a secure channel, as if that channel were realized by a dedicated and protected wire. Most formal methods for distributed systems have a notion of secure channels as a basic communication mechanism, or can easily specify one. Essentially, a unidirectional secure channel correctly delivers messages from one specific sender to one specific recipient and to no other party. Hence simply using such a secure-channel abstraction for SSL would be perfect for the overall system proof. However, is this abstraction sound? SSL is realized with cryptographic primitives, such as Diffie-Hellman key exchange and symmetric encryption and authentication. These systems are not perfectly unbreakable; e.g., an adversary with

sufficient computing time can deterministically insert forged messages or learn the messages sent by honest participants. Hence the simple secure-channel abstraction that we discussed is certainly not sound in the standard absolute sense. Nevertheless, it seems “essentially” correct in the sense that for adversaries with reasonably bounded computational power, and if one ignores very small probabilities, the differences seem to disappear. Hence the two main questions are:

- Can we rigorously define a soundness notion in which a cryptographic realization like SSL can be a refinement of a non-cryptographic specification like simple secure channels?
- Does SSL have features that differ so much from the secure-channel abstraction that they are not abstracted away by this potential soundness notion, and if yes, can the secure-channel abstraction be slightly modified to accommodate these features?

As an answer to the first question, we introduced in [15] the notion of *reactive simulatability (RSIM)*, which we survey below. It is not only applicable to the example of secure channels, but very broadly for abstractions of cryptographic systems. As to the second question, SSL indeed has such features (imperfections). For instance, an adversary who observes the underlying communication lines can easily see who communicates when with whom, and even the length of the communicated messages, because typical encryption leaves this length more or less unchanged. The adversary can also suppress messages.

Similar to this example, the overall approach when proving a system with cryptographic subsystems is usually as follows: First, find a good “natural” abstraction of the cryptographic subsystem. Secondly, investigate whether the natural abstraction has to be extended by certain imperfections such as leaking traffic patterns and message lengths. Thirdly, prove the real cryptographic system sound with respect to this modified abstraction in the RSIM sense. During the third step, one often also changes the cryptographic implementation a little because typical classical realizations concentrated on specific security properties and did not aim at realizing an overall abstraction.

2 Reactive Simulatability

Reactive simulatability (RSIM) is a notion for comparing two systems, typically called real and ideal system [15,16]. In terms of the formal-methods community one might call RSIM an implementation or refinement relation, specifically geared towards the preservation of what one might call secrecy properties compared with functional properties. In Figure 1, the ideal system is called TH (trusted host), and the protocol machines of the real system are called M_1, \dots, M_n . The ideal or real system interacts with arbitrary so-called honest users, collectively denoted by a machine H, and an adversary A, who is often given more power than the honest users. In real systems A typically controls the network and can manipulate messages on the bitstring level. The option for A and H to communicate directly corresponds to known- and chosen-message attacks.

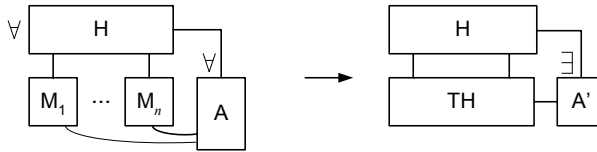


Fig. 1. Overview of reactive simulatability (RSIM).

Reactive simulatability between the real and ideal system essentially means that for every adversary A on the real system there exists an equivalent adversary A' on the ideal system, such that arbitrary honest users H cannot distinguish whether they interact with the real system and A , or with the ideal system and A' . Indistinguishability of families of random variables, here the two families of views of the honest users, is a well-known cryptographic notion from [18]. There exist stronger universal and blackbox version of this definition [15], depending mainly on the quantifier order.

The RSIM notion is based on simulatability definitions for secure (one-step) function evaluation [11, 12, 5, 14, 8]. It is also highly related to the observational equivalence notions for a probabilistic π -calculus from [13]. A notion very similar to RSIM was later also called UC [9].

3 System Model

In principle, RSIM can be defined over many system models by regarding the boxes in Figure 1 as (possibly probabilistic) I/O automata, Turing machines, CSP or π -calculus processes, etc.. Even large parts of most theorems and proofs about RSIM are on this box-level and could be instantiated in many ways. For the rigorous definitions, we have used a system model with probabilistic I/O automata, and with a well-defined computational realization by Turing machines for computational aspects. Two important aspects are:

- It is usually not sufficient that individual transitions of the automata are polynomial-time (“transaction poly”), and not even that the runtime of each automaton is polynomial in the entire inputs so far (“weakly poly”), because these notions do not compose. One usually needs overall polynomial-time restrictions in the initial inputs.
- We allow generic distributed scheduling for the asynchronous case. This means that for each part of the distributed computation that may be scheduled separately, we can designate an arbitrary other machine as the scheduler. This allows us to define adversarial scheduling with realistic information as well as, e.g., adversary-scheduled secure channels, local distributed algorithms not under control of the adversary, and specific fair schedulers.

4 Individual Security Properties

Reactive simulatability is great once it has been proved for a pair of an abstraction and a cryptographic realization, because then arbitrary larger systems can

be proved with the abstraction and the results are also true with the realization. However, it is a strong property, and sometimes the consequences for the realization are not desired. Hence it is important to also have notions of individual security properties, such as the integrity of certain messages or the absence of information flow between certain parties. These properties can be given similar links between abstract formulations usual in formal methods and cryptographic realizations; see, e.g., [15, 2, 3].

5 Dolev-Yao Models

In the past, formal methods have usually abstracted from cryptographic operations by term algebras called Dolev-Yao models or symbolic cryptography. Justifying these abstractions is non-trivial because the term algebras are used as initial models; in particular it is assumed that terms for which no explicit equations and derivation rules for an adversary exist in the algebra are perfectly secret from the adversary. We have shown that a Dolev-Yao model of several important core cryptographic primitives, with small extensions similar to those we mentioned for SSL above, can indeed be implemented soundly in the sense of RSIM under standard cryptographic assumptions, see in particular [4, 17]. However, we have also shown recently that extending these results to other primitives like hashing or XOR is not possible, at least not in the same very strong sense and with similar generality as our positive results.

In the context of larger systems, we see Dolev-Yao models as a tool on a middle level, useful for proving protocols that use standard cryptography in a blackbox way, but still rather explicitly. Within proofs of overall systems, we believe that even more abstract specifications of cryptographic subsystems, such as entire secure channels or entire secure payments, are more suitable.

6 Conclusion

An overview of all our own results on relating cryptography and formal methods, including concrete abstractions from cryptography that are sound in the RSIM sense, can be found at <http://www.zurich.ibm.com/security/models/>. The papers listed there also contain more references to related literature by others.

A particular area of where we can see formal methods for cryptography gaining more industrial relevance is web services security, at least if the current trend continues to make web services security specification, like all web services specifications, highly extensible and configurable, so that one may not be able to prove all standards-compatible realizations in advance. A first analysis of how formal-methods considerations and cryptographic considerations play together in this context can be found in [1]. For purely formal considerations based on Dolev-Yao models we also refer to [7, 6].

Acknowledgments. This work is partially supported by the European Commission through the IST Programme under Contract IST-4-026764-NOE ReSIST.

References

1. M. Backes, S. Mödersheim, B. Pfitzmann, and L. Viganò. Symbolic and cryptographic analysis of the Secure WS-ReliableMessaging Scenario. In *Proc. 9th FOSSACS*, volume 3921 of *LNCS*, pages 428–445. Springer, 2006.
2. M. Backes and B. Pfitzmann. Intransitive non-interference for cryptographic purposes. In *Proc. 24th IEEE Symp. on Security & Privacy*, pages 140–152, 2003.
3. M. Backes and B. Pfitzmann. Relating symbolic and cryptographic secrecy. *IEEE Transactions on Dependable and Secure Computing*, 2(2):109–123, 2005.
4. M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *Proc. 10th ACM CCS*, pages 220–230, 2003.
5. D. Beaver. Secure multiparty protocols and zero knowledge proof systems tolerating a faulty minority. *Journal of Cryptology*, 4(2):75–122, 1991.
6. K. Bhargavan, R. Corin, C. Fournet, and A. Gordon. Secure sessions for web services. In *ACM Workshop on Secure Web Services (SWS)*. ACM Press, to appear, 2004.
7. K. Bhargavan, C. Fournet, A. Gordon, and R. Pucella. TulaFale: A security tool for web services. In *Proc. 2nd Intern. Symp. on Formal Methods for Components and Objects (FMCO03)*, volume 3188 of *LNCS*, pages 197–222. Springer, 2004.
8. R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 3(1):143–202, 2000.
9. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. 42nd IEEE FOCS*, pages 136–145, 2001.
10. T. Dierks and C. Allen. The TLS Protocol Version 1.0, 1999. Internet RFC 2246.
11. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game – or – a completeness theorem for protocols with honest majority. In *Proc. 19th ACM STOC*, pages 218–229, 1987.
12. S. Goldwasser and L. Levin. Fair computation of general functions in presence of immoral majority. In *Proc. CRYPTO '90*, volume 537 of *LNCS*, pages 77–93. Springer, 1990.
13. P. Lincoln, J. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic poly-time framework for protocol analysis. In *Proc. 5th ACM CCS*, pages 112–121, 1998.
14. S. Micali and P. Rogaway. Secure computation. In *Proc. CRYPTO '91*, volume 576 of *LNCS*, pages 392–404. Springer, 1991.
15. B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In *Proc. 7th ACM CCS*, pages 245–254, 2000.
16. B. Pfitzmann and M. Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proc. 22nd IEEE Symp. on Security & Privacy*, pages 184–200, 2001.
17. C. Sprenger, M. Backes, D. Basin, B. Pfitzmann, and M. Waidner. Cryptographically sound theorem proving. In *Proc. 19th IEEE CSFW*, 2006. To appear.
18. A. C. Yao. Theory and applications of trapdoor functions. In *Proc. 23rd IEEE FOCS*, pages 80–91, 1982.