HAL
open science

Computationally Sound Compositional Logic for Security Protocols

Anupam Datta, Ante Derek, John C. Mitchell, Arnab Roy, Vitaly Shmatikov, Mathieu Turuani, Bogdan Warinschi

To cite this version:

Anupam Datta, Ante Derek, John C. Mitchell, Arnab Roy, Vitaly Shmatikov, et al.. Computationally Sound Compositional Logic for Security Protocols. Workshop on Formal and Computational Cryptography - FCC 2006, Véronique Cortier et Steve Kremer, Jul 2006, Venice/Italy. inria-00080593

HAL Id: inria-00080593

https://inria.hal.science/inria-00080593v1

Submitted on 19 Jun 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computationally Sound Compositional Logic for Security Protocols

Anupam Datta[1], Ante Derek[1], John C. Mitchell[1], Arnab Roy[1], Vitaly Shmatikov[2], Mathieu Turuani[3], and Bogdan Warinschi[3]

[1] Stanford University
[2] University of Texas at Austin
[3] LORIA-INRIA Nancy

Abstract. We have been developing a cryptographically sound formal logic for proving protocol security properties without explicitly reasoning about probability, asymptotic complexity, or the actions of a malicious attacker. The approach rests on a probabilistic, polynomial-time semantics for a protocol security logic that was originally developed using nondeterministic symbolic semantics. This workshop presentation will discuss ways in which the computational semantics lead to different reasoning methods and report our progress to date in several directions. One significant difference between the symbolic and computational settings results from the computational difference between efficiently recognizing and efficiently producing a value. Among the more recent developments are a compositional method for proving cryptographically sound properties of key exchange protocols, and some work on secrecy properties that illustrates the computational interpretation of inductive properties of protocol roles.

1 Introduction

As is well-known to the Workshop on Formal and Computational Cryptography (FCC 2006) audience, there are two important but historically independent foundations for reasoning about cryptographic protocols, one based on logic and symbolic computation, and one based on computational complexity theory. While the symbolic approach has been asuccessful basis for formal logics and automated tool, the computational approach yields more insight into the strength and vulnerabilities of protocols. We have developed a cryptographically sound formal logic for proving protocol security properties without explicitly reasoning about probability, asymptotic complexity, or the actions of a malicious attacker. The approach rests on a probabilistic, polynomial-time semantics for a protocol security logic that was originally developed using nondeterministic symbolic semantics. The goal of this work is to show that formal reasoning, based on an abstract treatment of cryptographic primitives, can be used to reason about probabilistic polynomial-time protocols in the face of probabilistic polynomial-time attacks. The semantics we explore in this work reflects accepted modelling

approaches used in the field of cryptography, particularly [5, 24]. The computational semantics brings forward some interesting distinctions that were not available in the coarser symbolic model, and also raises a number of interesting technical problems and challenges.

Protocol Composition Logic [11, 7, 8] uses a modal operator similar to Floyd-Hoare logic. Intuitively, the formula $\psi\,[P]_X\,\phi$ means that if ψ is true at some point in the execution of a protocol (in the presence of a malicious attacker), then ϕ will be true after agent X performs the sequence P of actions. The pre- and post-conditions may describe actions taken by various principals and characterize the information that is available to or hidden from them.

Semantics and Soundness Theorem Our central organizing idea is to interpret formulas as operators on probability distributions on traces. Informally, representing a probability distribution by a set of equi-probable traces (each tagged by the random sequence used to produce it), the meaning of a formula ϕ on a set T of traces is the subset $T' \subseteq T$ in which ϕ holds. This interpretation yields a probability: the probability that ϕ holds is the ratio $|T'|/|T|$. Conjunction and disjunction are simply intersection and union. There are several possible interpretations for implication, and it is not clear at this point which will prove most fruitful in the long run. In the present paper, we interpret $\phi \implies \psi$ as the union of $\neg\phi$ and the composition of ψ with ϕ; the latter is also the conditional probability of ψ given ϕ. This interpretation supports a soundness proof for a sizable fragment of the protocol logic, and resembles the probabilistic interpretation of implication in [22]. Since the logic does not mention probability explicitly, we consider a formula "true" if it holds with asymptotically overwhelming probability.

In previous work [11, 7, 8] over a symbolic semantic model, the atomic formula $\mathsf{Has}(X, m)$ means that m is in the set of values "derivable," by a simple fixed algorithm, from information visible to X. The simple fixed algorithm is central to what is called the Dolev-Yao model, after [10] and much subsequent work by others. In replacing the symbolic semantics with a computational semantics based on probabilistic polynomial time, we replace the predicate Has with two predicates, $\mathsf{Possess}$ and Indist. Intuitively, $\mathsf{Possess}(X, m)$ means that there is an algorithm that computes the value of m from information available to X with high probability, while $\mathsf{Indist}(X, m)$ means that X cannot feasibly distinguish m from a random value chosen according to the same distribution.

The proof system used in this work is based on one developed in [7, 8, 2]. The axioms embody different forms of reasoning employed by cryptographers including complexity-theoretic reductions, information-theoretic arguments and asymptotic calculations. The soundness proofs of the axioms are carried out using similar mathematical techniques. We emphasize that this is a one-time mathematical effort. Protocol proofs can now be carried out symbolically using the proof system without explicitly reasoning about probability and complexity. Another advantage of this approach is that the axioms and rules used in a proof identify the specific properties of the cryptographic primitives that are relevant in order to guarantee the property of the protocol. This information can be

useful for protocol design since primitives that provide weaker properties often have more efficient constructions.

Application to key exchange We have developed a compositional method for proving cryptographically sound security properties of key exchange protocols, based on this logic. Since reasoning about an unbounded number of runs of a protocol involves induction-like arguments about properties preserved by each run, we formulate a specification of *secure key exchange* that is closed under general composition with steps that use the key. We present formal proof rules based on this game-based condition, and prove that the proof rules are sound over the computational semantics. We demonstrate the applicability of the proof method by formalizing and proving some security properties of the ISO-9798-3 key exchange protocol [20] and its composition with a canonical secure sessions protocol. The security proof for ISO-9798-3 relies on the Decisional Diffie-Hellman assumption and the use of CMA-secure signatures, while the security of the secure sessions protocol relies on the use of a CPA-secure symmetric encryption scheme [15]. The fact that these two protocols compose securely when executed one after the other follows from the *sequential composition theorem* [8]. In order to model and prove security for these protocols, we had to extend the computational model and logic [9] to include a number of additional cryptographic primitives: signatures, symmetric encryption, as well as codify the Decisional Diffie-Hellman assumption. A central concept in many compositional proof methods [1, 12, 13, 6, 8] is an *invariant*. In developing compositional security proofs of complex protocols [17], we require that each protocol component respects the invariants of the other components in the system [8]. It appears that standard cryptographic security definitions for key exchange like *key indistinguishability* [5, 3] are *not* invariant in the manner needed for an inductive proof of composition. Even if a key exchange protocol, run by itself in isolation, produces a key that is indistinguishable from a random value chosen from the same distribution, key indistinguishability is generally lost as soon as the key is used to encrypt a message of a known form or with partially known possible content. Moreover, some situations allow one agent to begin transmitting encrypted data before the other agent finishes the last step of the key exchange, rendering key indistinguishability false at the point that the key exchange protocol finishes. (This appears to be the case for SSL [14]; see [21] for a discussion of data transfer before the key exchange *finished* messages are received.) Furthermore, some key exchange protocols even use the generated key during the protocol, preventing key indistinguishability. Fortunately, many protocols that use keys do not require key indistinguishability to provide meaningful security guarantees. In particular, semantic security [16] does *not* require that the keys used remain indistinguishable from random.

To circumvent the technical problems we encountered in working with key indistinguishability, we propose an alternative notion that is parameterized by the security goal of the application in which the resulting key is used. As concrete examples, we consider cases where the key is used for encryption or MAC. The security definition for key exchange requires that the key produced is "good" for that application, i.e. an adversary interacting with the encryption scheme using

this key cannot win the security game for that scheme (for example, the IND-CPA game for encryption). The resulting definition for key exchange is invariant under composition with the application protocol which uses the key.

We define usability of keys obtained through a key exchange protocol Σ with respect to a class of applications S via a two-phase experiment. The experiment involves a two-phase adversary $\mathcal{A} = (\mathcal{A}_e, \mathcal{A}_c)$. In the *key exchange phase*, the honest parties run sessions of the protocol following the standard execution model: each principal executes multiple sessions of the protocol (as both initiator and responder) with other principals; the communication between parties is controlled by the adversary $\mathcal{A}_e$. At the end of the key exchange phase, the adversary selects a challenge session sid among all sessions executed by the honest parties, and outputs some state information St representing the information $\mathcal{A}_e$ was able to gather during its execution. Let k be the key locally output by the honest party executing the session sid. At this point, the experiment enters its second phase—the *challenge phase* where the goal of the adversary is to demonstrate an attack against a scheme $\Pi \in S$ which uses the key k. After $\mathcal{A}_e$ receives as input St, it starts interacting with Π according to the game used for defining security of the application protocols in S. For example, if S is a set of encryption schemes, then the relevant game may be IND-CPA, IND-CCA1, or IND-CCA2 [15]. Since the specific task we treat in this paragraph is secure sessions, we formalize the case when the game defines IND-CPA security. Thus, in playing the game, $\mathcal{A}_c$ has access to a left-right encryption oracle under k, and in addition, it receives as input the state information from $\mathcal{A}_e$. The advantage of the adversary is defined as for the standard IND-CPA game with the difference that the probability is taken over the random coins of the honest parties (used in the execution of the protocol), the coins of the two adversaries, and the coins used for encryption in the challenge phase. The keys obtained by running the key exchange protocol are usable for the schemes in S if this advantage is bounded above by a negligible function of the security parameter, for *all* encryption schemes in S. The universal quantification over schemes is used to capture the fact that the security property is guaranteed for all encryption schemes which satisfy the IND-CPA condition. We note that in this definition the adversary $\mathcal{A}_c$ is not allowed to interact with the key exchange protocol in the challenge phase.

Secrecy A secrecy property asserts that no 'useful' information about some data that is used in the protocol is revealed to others. There are protocols, such as exchanges of signed messages, that provide authentication but not secrecy of sent data, and similarly there are ways to generate a secret between two parties (such as Diffie-Hellman key exchange) without receiving any guarantee about the identity of the other party. In many protocols, however, the two properties turn out to be interdependent, with authentication relying on secrecy of signing keys or other material, for example. If a protocol generates a fresh value, called a *nonce*, and sends it in an encrypted message, then under ordinary circumstances only agents that have the decryption key can obtain the nonce. However, many protocols have steps that receive a message encrypted with one key, and send some of its parts out encrypted with a different key. Since network protocols

are executed asynchronously by independent agents, some potentially malicious, we need to show that even after arbitrarily many steps of independent protocol sessions, a secret remains so.

In as-yet-unpublished work, we formalize reasoning about secrecy by introducing axioms and rules for showing that individual receive-send protocol steps respect secrecy of message parts, and an induction rule for reasoning about arbitrarily many simultaneous protocol sessions. These proof principles are shown sound for the probabilistic polynomial-time semantics of protocol execution and attack. At a high level, the basic ideas are similar to the "rank function method" [23] and work in the strand space approach [25], both formulated for symbolic execution models.

The proof system is sufficient to prove authentication and secrecy properties of the Kerberos V5 [19] protocol. Our concise, modular proof provides assurance about the correctness of Kerberos, assuming that the symmetric encryption scheme is IND-CCA secure and provides ciphertext integrity [4]. Our notion of 'usefulness' of the secrets (the session keys) in the protocol is whether they are good for use as a key in the second stage of the two-phase experiment described previously.

Although our formal proof system is sufficient to prove interesting security properties of a nontrivial, practical protocol, some intriguing technical questions remain. As an intermediate step before producing our most recent computationally sound system, we also developed an unpublished proof system for secrecy over the symbolic model. The symbolically sound proof system appears to be more powerful, but we do not know whether this is inherent in the change from symbolic to computational semantics, or a gap that we will be able to close through further work on the present system.

One specific issue is illustrated by the variant of the Needham-Schroeder protocol that is used in [18] to illustrate a limitation of the original rank function method and motivate an extension for reasoning about temporary secrets. We are able to give a straightforward formal proof of this example in our symbolically sound proof system, but we do not yet have computationally sound rules for carrying out that argument. A basic limitation in computationally sound reasoning is that it is not possible to do case analysis on the structure of messages sent by an attacker. In the symbolic model, however, every message is represented by a symbolic expression, and every expression has a syntactic form.

References

1. R. Alur and T. A. Henzinger. Computer-aided verification. an introduction to model building and model checking for concurrent systems. Draft, 1998.
2. M. Backes, A. Datta, A. Derek, J. C. Mitchell, and M. Turuani. Compositional analysis of contract signing protocols. In *Proceedings of 18th IEEE Computer Security Foundations Workshop*. IEEE, 2005. to appear.
3. M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authenticationand key exchange protocols. In *Proc. of the 30th Annual Symposium on the Theory of Computing*, pages 419–428. ACM, 1998.

4. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *ASIACRYPT*, pages 531–545, 2000.
5. M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '93)*, pages 232–249. Springer-Verlag, 1994.
6. R. Canetti, L. Cheung, D. Kaynar, M. Liskov, N. Lynch, O. Pereira, and R. Segala. Using probabilistic i/o automata to analyze an oblivious transfer protocol. Technical Report MIT-LCS-TR-1001, MIT CSAIL, 2005.
7. A. Datta, A. Derek, J. C. Mitchell, and D. Pavlovic. A derivation system for security protocols and its logical formalization. In *Proceedings of 16th IEEE Computer Security Foundations Workshop*, pages 109–125. IEEE, 2003.
8. A. Datta, A. Derek, J. C. Mitchell, and D. Pavlovic. A derivation system and compositional logic for security protocols. *Journal of Computer Security*, 2004. to appear.
9. A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, and M. Turuani. Probabilistic polynomial-time semantics for a protocol security logic. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP '05)*, Lecture Notes in Computer Science. Springer-Verlag, 2005.
10. D. Dolev and A. Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 2(29):198–208, 1983.
11. N. Durgin, J. C. Mitchell, and D. Pavlovic. A compositional logic for proving security properties of protocols. *Journal of Computer Security*, 11:677–721, 2003.
12. S. O. Ehmety and L. C. Paulson. Program composition in isabelle/unity. In *16th International Parallel and Distributed Processing Symposium (IPDPS 2002), Proceedings*. IEEE Computer Society, 2002.
13. S. O. Ehmety and L. C. Paulson. Mechanizing compositional reasoning for concurrent systems: some lessons. *Formal Aspects of Computing*, 17(1):58–68, 2005.
14. A. Freier, P. Karlton, and P. Kocher. The SSL protocol version 3.0. IETF Internet draft, November 18 1996.
15. O. Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.
16. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Science*, 28:270–299, 1984.
17. C. He, M. Sundararajan, A. Datta, A. Derek, and J. C. Mitchell. A modular correctness proof of IEEE 802.11i and TLS. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, 2005.
18. J. Heather. Strand spaces and rank functions: More than distant cousins. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, page 104, 2002.
19. J. Kohl and B. Neuman. The Kerberos network authentication service (version 5). IETF RFC 1510, September 1993.
20. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
21. J. C. Mitchell, V. Shmatikov, and U. Stern. Finite-state analysis of SSL 3.0. In *Proceedings of Seventh USENIX Security Symposium*, pages 201–216, 1998.
22. N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–87, 1986.
23. S. Schneider. Verifying authentication protocols with csp. *IEEE Transactions on Software Engineering*, pages 741–58, 1998.
24. V. Shoup. On formal models for secure key exchange (version 4). Technical Report RZ 3120, IBM Research, 1999.

25. F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7(1), 1999.