Computationally Sound Symbolic Analysis of Anonymity in the Presence of Active Adversaries

Yusuke Kawamoto\(^1\) and Hideki Sakurada\(^2\)

\(^1\) Graduate School of Information Science and Technology, University of Tokyo
\(^2\) NTT Communication Science Laboratories, NTT Corporation

Abstract

Anonymity is an important security policy required in many cryptographic protocols. In anonymous communication channels, parties should be able to send and receive messages without revealing any relationships between the messages and the identities of parties. In anonymous authentication protocols, parties should be able to authenticate themselves without revealing their identities.

Anonymity of cryptographic protocols can be analyzed in two approaches. The first is a computational approach: a message is a bit string and an adversary is a probabilistic polynomial-time interactive Turing machine. The computational analysis of cryptographic protocols is based on computational complexity theory. In computational models, anonymity is defined using indistinguishability between two probability ensembles of networks: with overwhelming probability, no adversary can distinguish between two networks in one of which the identities of parties have been switched.

The second is a symbolic approach: a message is abstracted into a symbolic expression, called a Dolev-Yao term [4], and an adversary can perform only a fixed set of operations on Dolev-Yao terms. The symbolic analysis of cryptographic protocols assumes perfectly secure cryptographic schemes. In symbolic models, for example in the applied pi-calculus, anonymity is defined as the observational equivalence between two symbolic processes in one of which the identities of parties have been switched.

Recently, many researches have related the two approaches and shown the computational soundness [1, 7]: symbolic security proofs imply strong security guarantees based on computational complexity theory.

The soundness results in the previous studies are divided into two categories: mapping soundness and policy soundness. Mapping soundness [7] claims that a computational execution trace corresponds to a symbolic execution trace with overwhelming probability. This soundness can deal with trace properties in the presence of an active adversary. On the other hand, policy soundness claims that a symbolic security policy corresponds to a computational security policy that is defined using only the computational execution traces corresponding to symbolic execution traces. This kind of soundness can deal with the security policies, such as anonymity, beyond trace properties. For example, the symbolic
anonymity defined using Abadi-Rogaway pattern [1] corresponds to the computational anonymity defined using the computational indistinguishability with respect to the computational execution traces corresponding to symbolic execution traces. However, this cannot deal with fully active adversaries unless it is combined with mapping soundness.

By combining these two kinds of soundness, Comon-Lundh and Cortier [2] proved that in the case of symmetric encryption, the observational equivalence between two processes in the applied pi-calculus implies the computational indistinguishability between the computational executions of the two processes. By extending [2] and [6], we obtained the computational soundness of observational equivalence in the case of ring signature [5, 3].

In this work, we provide a symbolic model for public key encryption and ring signature in the applied pi-calculus, prove its computational soundness in the presence of active adversaries, and analyze the anonymity of an example protocol in the computationally sound symbolic model.

In addition, we mention the relationships between the security of cryptographic primitives used in protocols and the class of protocols to which we can apply our computational soundness result. Specifically, we deal with two levels of security both for public key encryption and for ring signature. As regards encryption, we consider IND-CCA2 and rerandomizable IND-RCCA security. In the case of rerandomizable IND-RCCA, we need to restrict the class of protocols so that every plaintext encrypted by an honest party is concatenated with an honestly generated nonce. As regards ring signature, we compare strong unforgeability with existential unforgeability, and full anonymity with basic anonymity. In the case of existential unforgeability, we need to restrict the class of protocols so that every signature produced by an honest party is concatenated with an honestly generated nonce.

References