A Formal Model of Identity Mixer*

Jan Camenisch\(^2\), Sebastian Mödersheim\(^1\), and Dieter Sommer\(^2\)

\(^1\) DTU Informatics, Denmark
\hspace{2cm} samo@imm.dtu.dk

\(^2\) IBM Research – Zurich, Switzerland
\hspace{1cm} \{jca,dso\}@zurich.ibm.com

Abstract. Identity Mixer is an anonymous credential system developed
at IBM that allows users for instance to prove that they are over 18 years
old without revealing their name or birthdate. This privacy-friendly tech-
nology is realized using zero-knowledge proofs. We describe a formal
model of Identity Mixer that is well-suited for automated protocol veri-
fication tools in the spirit of black-box cryptography models.

1 Introduction

Due to the spreading use of electronic communication means, users increasingly
disperse their personal information widely. Users have lost control over their
data, as it is most often not clear who receives and stores which information
and how organizations handle this information, particularly to whom they pass
it on. This situation is aggravated by the increasing easiness to store, distribute,
and profile these data. While on the one hand protecting users’ privacy is very
important, on the other hand, many transactions require authentication, au-
thorization, and accountability. There is seemingly a partial conflict of goals of
properly identifying users while protecting their privacy.

The Identity Mixer system developed by IBM Research – Zurich solves this
contradiction by employing particular non-interactive zero-knowledge proofs and
suitable signature and encryption schemes. For instance, using Identity Mixer to
issue electronic identity credentials, a user is able to prove being at least 18 years
old or living in a certain town—without revealing their name or their precise age
or any other details. The system’s main goal is to provide strong authentication
of users and at the same time to protect the users’ privacy by minimizing the
amount of the users’ information being revealed in an interaction.

Identity Mixer is an implementation of the Camenisch-Lysyanskaya anony-
mous credential system [13], extended by a number additional features aimed
at enabling its use in practice. These features were put forth in a number of
later publications [9,15]. The cryptography behind Identity Mixer is well under-
stood and the basic system got proved secure [13]. However, the extended system

* This work was partially supported by the EU-funded projects AVANTSSAR and
PrimeLife (grant agreements 216471 and 216483). The authors enjoyed discussions
with Alberto Calvi, Luca Viganò, and Greg Zaverucha. Thank you!

© Springer-Verlag Berlin Heidelberg 2010
as implemented has never been proved secure. Indeed, proving security of the complex and dynamic system resulting from the combination of the many cryptographic building blocks is a challenging task in general. Subtle mistakes in the design can easily lead to vulnerabilities that can be exploited without breaking the cryptography. Such mistakes are often hard to find due to the complicated behavior of distributed systems. Automated verification with model-checking methods based on perfect cryptography models can help here to discover many such mistakes and increase the confidence in systems when the verification is successful, e.g., [14, 17]. The goal of this paper is to provide a formal model for Identity Mixer, in particular the zero-knowledge proofs it uses, in a way feasible for automated protocol verification tools.

While a lot of verification tools exist for protocol verification, the area of zero-knowledge-proof based system has been started only recently. Backes, Maffei, and Unruh [7] provide a first attempt to integrate this “cryptographic primitive” into the verification tool ProVerif [11]. Their model considers non-interactive zero-knowledge proofs as terms and a set of function symbols representing verification operations that a verifier can apply to a received proof term. Algebraic properties ensure that the term resulting from the operations evaluate to true or false according to whether the zero-knowledge proof term indeed proves the desired statement. While this gives a highly declarative model of the zero-knowledge proofs as an abstract primitive, this is hardly feasible in automated verification tools due to the extensive use of algebraic properties. In fact, even the properties for the boolean combinations induce an undecidable unification problem, and it is no wonder that tools that allow for such properties easily run into infinite loops. As a consequence, Backes et al. [7] uses a very restricted re-encoding of the algebraic theory in ProVerif, and the authors have eventually moved to another approach altogether, namely security types [6]. While this is a valuable complementary approach, the question remains whether we cannot use at all the existing methods and tools of protocol verification that were so successful on closely related tasks.

We show that there is indeed a feasible way to model zero-knowledge proofs in standard black-box protocol verification tools. In a nutshell, the idea to avoid the difficulties that arise when employing algebraic properties to model statements that are proved is to use pattern matching. This applies when an honest agent receives a zero-knowledge proof term. Instead of expressing the verification of this proof by verification operations, we show how to transform the desired properties into a pattern that describes the set of zero-knowledge proof terms that the receiver will accept. This can be done by a simple matching (or unification in case of symbolic representations) without any algebraic reasoning.

Contributions. The first contribution of this paper is a model of zero-knowledge proofs that is feasible for automated verification. In fact, the specifications (except for privacy goals, which are not considered in this paper) can directly be run in existing tools without requiring extensions. The second contribution is a formalization of Identity Mixer in this abstract model, both allowing for verification, and also as an overview that abstracts from the underlying cryptography