Computationally Sound Abstraction and Verification of Secure Multi-Party Computations*

Michael Backes
Saarland University
MPI-SWS

Matteo Maffei
Saarland University

Esfandiar Mohammadi
Saarland University

While Dolev-Yao models traditionally comprise only non-interactive cryptographic operations, recent cryptographic protocols rely on more sophisticated *interactive primitives*, with unique features that go far beyond the traditional goals of cryptography to solely offer secrecy and authenticity of communication.

Secure multi-party computation (SMPC) constitutes arguably one of the most prominent and most amazing such interactive primitive. Intuitively, in an SMPC, a number of parties $P_1, \dots, P_n$ wish to securely compute the value $F(d_1, \dots, d_n)$, for some well-known public function F, where each party P_i holds a private input d_i. This multi-party computation is considered secure if it does not divulge any information about the private inputs to other parties; more precisely, no party can learn more from the participation in the SMPC than the party could learn purely from the result of the computation already.

Recently, the effectiveness of SMPC as a building block of large-scale and practical applications has been demonstrated by the sugar-beet double auction that took place in Denmark, which built upon an SMPC. Given the complexity of SMPC and its potential role as a building block for larger cryptographic protocols, it is important to develop abstraction techniques to reason about SMPC-based cryptographic protocols and to offer support for the automated verification of their security.

Our contribution is threefold. We present an **abstraction of SMPC** within the applied π-calculus. This abstraction consists of a process that receives the inputs from the parties involved in the protocol over private channels, computes the result, and sends it to the parties again over private channels. This abstraction can be used to model and reason about larger cryptographic protocols that employ SMPC as a building block. Building upon an existing type-checker, we propose an automated **verification technique** for protocols based on our SMPC abstraction. Finally, we establish a **computational soundness result** for protocols built upon our abstraction of SMPC. This computational soundness result holds for protocols that use SMPCs which involve arbitrary arithmetic operations, encryption, and digital signatures. Furthermore, we prove that given a computationally sound Dolev-Yao model for a set operations, the extension that additionally includes SMPCs that use these operations is still computationally sound.

*A paper about this work appeared in the proceedings of the Conference on Foundations of Software Technology and Theoretical Computer Science 2010 (FSTTCS).