

SpringerBriefs in Computer Science

Series editors

Stan Zdonik, Brown University, Providence, Rhode Island, USA

Shashi Shekhar, University of Minnesota, Minneapolis, Minnesota, USA

Xindong Wu, University of Vermont, Burlington, Vermont, USA

Lakhmi C. Jain, University of South Australia, Adelaide, South Australia, Australia

David Padua, University of Illinois Urbana-Champaign, Urbana, Illinois, USA

Xuemin (Sherman) Shen, University of Waterloo, Waterloo, Ontario, Canada

Borko Furht, Florida Atlantic University, Boca Raton, Florida, USA

V.S. Subrahmanian, University of Maryland, College Park, Maryland, USA

Martial Hebert, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

Katsushi Ikeuchi, University of Tokyo, Tokyo, Japan

Bruno Siciliano, Università di Napoli Federico II, Napoli, Italy

Sushil Jajodia, George Mason University, Fairfax, Virginia, USA

Newton Lee, Newton Lee Laboratories, LLC, Tujunga, California, USA

More information about this series at <http://www.springer.com/series/10028>

Niklas Büscher • Stefan Katzenbeisser

Compilation for Secure Multi-party Computation



Springer

Niklas Büscher
Security Engineering Group
Technische Universität Darmstadt
Darmstadt, Germany

Stefan Katzenbeisser
Security Engineering Group
Technische Universität Darmstadt
Darmstadt, Germany

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-319-67521-3 ISBN 978-3-319-67522-0 (eBook)
<https://doi.org/10.1007/978-3-319-67522-0>

Library of Congress Control Number: 2017954354

© The Author(s) 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

In memory of Helmut Veith.

Preface

An ever-growing amount of data is processed by online services every day. Unfortunately, data owners are at risk to lose control over their possibly sensitive data once it is deployed at an untrusted third party. A solution to this dilemma is Secure Multi-party Computation (MPC), which has emerged as a generic approach to realize Privacy-Enhancing Technology in a cryptographically sound manner, as it allows to perform arbitrary computations between two or more parties over encrypted data. In recent years, numerous protocols and optimizations have made MPC practical for relevant real world scenarios. This development has led to a significant improvement in the performance and size of applications that can be realized with MPC.

A major obstacle in the past was to generate MPC applications by hand. Recently special compilers have been developed to build all kinds of applications. In this book, we summarize our research on the compiler CBMC-GC for MPC over Boolean circuits. We show and explain how efficient MPC applications can be created automatically from ANSI-C, which bridges the areas of cryptography, compilation and hardware synthesis. Moreover, we give an insight into the requirements for creating efficient applications for MPC, and thus we hope that this work can be of interest not only to researchers in the area of MPC but also developers realizing practical applications with MPC.

The authors wish to thank Andreas Holzer, Martin Franz and Helmut Veith, with whom we started the research on compilers for MPC. Moreover, we wish to thank our students Alina Weber and David Kretzmer who contributed ideas and implementations to the later parts of this book. This work has been co-funded by the DFG as part of project S5 within the CRC 1119 “CROSSING” and by the DFG within the RTG 2050 “Privacy and Trust for Mobile Users”.

Darmstadt, Germany
August 2017

Niklas Büscher
Stefan Katzenbeisser

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Classification of MPC Compilers	3
1.3	Outline of the Book	4
2	Background	5
2.1	Boolean Circuits	5
2.2	Secure Computation	6
2.2.1	Oblivious Transfer	7
2.2.2	Yao's Garbled Circuits Protocol	7
2.2.3	Goldreich-Micali-Wigderson (GMW) Protocol	9
2.3	Benchmarking Applications for MPC Compilers	10
3	Compiling ANSI-C Code into Boolean Circuits	15
3.1	Motivation and Overview	15
3.2	Background: Bounded Model Checking	17
3.3	CBMC-GC's Compilation Chain	18
3.3.1	Input Language and Circuit Mapping	18
3.3.2	C Parser and Type Checking	19
3.3.3	GOTO Conversion	20
3.3.4	Loop Unrolling	21
3.3.5	Conversion into Single Static Assignment Form	23
3.3.6	Expression Simplification	24
3.3.7	Circuit Instantiation	24
3.4	Complexity of Operations in MPC	26
4	Compiling Size-Optimized Circuits for Constant-Round MPC Protocols	29
4.1	Motivation and Overview	29
4.2	Circuit Minimization for MPC	32
4.3	Building Blocks for Boolean Circuit Based MPC	33
4.4	Gate-Level Circuit Minimization	35

4.5	Evaluation	39
4.5.1	Evaluation of Circuit Minimization Techniques	39
4.5.2	Compiler Comparison	41
5	Compiling Parallel Circuits	43
5.1	Motivation and Overview	43
5.2	Parallel Circuit Evaluation	44
5.3	Compiler Assisted Parallelization Heuristics	46
5.3.1	Fine-Grained Parallelization (FGP)	46
5.3.2	Coarse-Grained Parallelization (CGP)	48
5.4	Evaluation of Parallelization in Yao's Garbled Circuits	52
5.4.1	UltraSFE	52
5.4.2	Evaluation Methodology	53
5.4.3	Circuit Garbling (Offline)	55
5.4.4	Full Protocol (Online)	58
6	Compiling Depth-Optimized Circuits for Multi-Round MPC	
	Protocols	61
6.1	Motivation and Overview	61
6.2	Compilation Chain for Low-Depth Circuits	62
6.2.1	Preprocessing Reductions	63
6.2.2	Sequential Arithmetics and Carry-Save Networks (CSNs)	65
6.2.3	Optimized Building Blocks	67
6.2.4	Gate Level Minimization Techniques	71
6.3	Experimental Evaluation	71
6.3.1	Benchmarked Functionalities and Their Parameters	72
6.3.2	Compiler Comparison	73
6.3.3	Evaluation of the Optimizations Techniques	74
6.3.4	Protocol Runtime	75
7	Towards Scalable and Optimizing Compilation for MPC	79
7.1	Motivation and Overview	79
7.2	Adapted Compilation Chain	80
7.2.1	Compilation Architecture	80
7.2.2	Global Constant Propagation	81
7.2.3	Implementation	83
7.3	Experimental Evaluation	84
7.3.1	Description of Experiments	84
7.3.2	Compilation Results	85
A	CBMC-GC Manual	87
	References	89