Ad-hoc Secure Two-Party Computation on Mobile Devices using Hardware Tokens

Daniel Demmler, Thomas Schneider, Michael Zohner (TU Darmstadt)
Motivation

Smartphones are awesome!
Motivation

Smartphones are **awesome**!

Contacts
Motivation

Smartphones are **awesome!**

Contacts

Calendar
Motivation

Smartphones are **awesome**!

Contacts
Calendar
Location
Motivation

Smartphones are **awesome**!

Contacts
Calendar
Location
Banking
Motivation

Smartphones are awesome!

Contacts
Calendar
Location
Banking
Messaging
Motivation

Smartphones are awesome!

Contacts
Calendar
Location
Banking
Messaging
Games
Motivation

Smartphones are awesome!

Contacts
Calendar
Location
Banking
Messaging
Games
...

$
Motivation

Smartphones are **awesome**!

Smartphones are **limited**...

- Contacts
- Calendar
- Location
- Banking
- Messaging
- Games
- ...

22.08.2014 | TU Darmstadt / CASED / EC-SPRIDE / ENCRYPTO | USENIX Security 2014 | 10
Motivation

Smartphones are **awesome**!

Smartphones are **limited**...

Contacts  
Calendar  
Location  
Banking  
Messaging  
Games  

...  

Computation
**Motivation**

Smartphones are **awesome**!

- Contacts
- Calendar
- Location
- Banking
- Messaging
- Games
  
...  

Smartphones are **limited**...

- Computation
- Memory
Motivation

Smartphones are **awesome**!

Contacts
Calendar
Location
Banking
Messaging
Games
...

Smartphones are **limited**...

Computation
Memory
Communication
Motivation

Smartphones are **awesome!**

- Contacts
- Calendar
- Location
- Banking
- Messaging
- Games

...  

Smartphones are **limited**...

- Computation
- Memory
- Communication
- Battery Life
Generic Secure Computation

Function $f \equiv$ Boolean Circuit $C$

Here: Two-Party scenario

Private Inputs: $x, y$

Passive Adversary Model
Generic Secure Computation

Function $f \equiv$ Boolean Circuit $C$

Here: Two-Party scenario

Private Inputs: $x, y$

Passive Adversary Model

Yao’s GC [Yao86] & GMW [GMW87]

$O(|f|)$ symmetric crypto

$O(t \cdot |f|)$ communication

$\Rightarrow$ Too much for mobile devices

Fairplay [MNPS04], FastGC [HEKM11]

Mobile Yao [HCE11]
Secure Computation Applications

Finding shared contacts

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>481516</td>
<td>666777</td>
</tr>
<tr>
<td>101010</td>
<td>101010</td>
</tr>
<tr>
<td>123456</td>
<td>234201</td>
</tr>
<tr>
<td>234201</td>
<td>000911</td>
</tr>
</tbody>
</table>
Secure Computation Applications

Finding shared contacts

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>481516</td>
<td>666777</td>
</tr>
<tr>
<td>101010</td>
<td>101010</td>
</tr>
<tr>
<td>123456</td>
<td>234201</td>
</tr>
<tr>
<td>234201</td>
<td>000911</td>
</tr>
</tbody>
</table>

Scheduling a meeting

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
<th>=</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Secure Computation Applications

Scheduling a meeting with location information
Our Setting
Our Setting
GMW Protocol

Goldreich-Micali-Wigderson, 1987

XOR sharing to mask values: \[ x = x_A \oplus x_B \]

Local evaluation of XOR gates

Interactive evaluation of AND gates

Using pre-computed Multiplication Triples

HOW TO PLAY ANY MENTAL GAME

or

A Completeness Theorem for Protocols with Honest Majority

[Extended Abstract]

Oded Goldreich

Shai Halevi

Avi Wigderson

Dept. of Computer Science

MIT

Inst. for Math.

and Comp. Sci.

Technion

Hebrew University

Haifa, Israel

Cambridge, MA 02139

Jerusalem, Israel

Abstract

We present a polynomial-time algorithm that, given as input a description of a game with incomplete information and any number of players, produces a protocol for playing the game that takes no partial information, provided the majority of the players is honest.

Our algorithm automatically solves all the multi-party protocol problems addressed in complexity-based cryptography during the last 10 years. It actually is a complete solution for the class of distributed protocols with honest majority. Such completeness theorems are optimal in the sense that, if the majority of the players is not honest, some protocol problems have no efficient solutions.

1. Introduction

Before discussing how to "make playables" a general game with incomplete information (which we do in section 8) let us address the problem of making playables a special class of games, the Turing machine games ("T-machines for short").

Informally, a parties, respectively and individually owning secret inputs \( x_1, \ldots, x_n \), would like to

compute \( y = f(x_1, \ldots, x_n) \) for some function \( f \) after an agreed-upon protocol, say an RNC circuit that implements \( f \). However, if any one of the parties \( x_i \) is malicious, it can compute extra information about \( y \) besides what can be learned from the inputs. Moreover, even if all \( x_i \) are honest, it can compute extra information about \( y \) besides what can be learned from the inputs.

To ensure that the protocol is "playable" (i.e., the output is not revealed to any of the parties), the players must agree on the following:

1. A function \( f \) they want to compute.
2. A protocol that will compute \( f \) without revealing anything about \( x_i \) to any of the other parties.

We call such a protocol a "playable protocol" for \( f \).

2. Preliminary Definitions

2.1 Notation and Conventions for Probabilistic Algorithms

We emphasize the number of inputs received by an algorithm as follows. If algorithm \( A \) receives only input \( x \), we write \( A(\cdot, x) \) and so on.

RV will stand for "random variable" in this paper we only consider RVs that assume values in
Secure Computation Phases

Setup Phase

\[ |f| \]
Secure Computation Phases

Setup Phase

Online Phase

\[ f(x, y) \]
Secure Computation Phases

Init Phase

Setup Phase

Online Phase

\[ f(x, y) \]
Secure Computation Phases

Init Phase

Setup Phase

Online Phase

\[ f(x, y) \]
Multiplication Triple Generation

Multiplication Triple (MT):

\[(c_A \oplus c_B) = (a_A \oplus a_B)(b_A \oplus b_B)\]

Shares intended for only one party: \(a_A, b_A, c_A \iff a_B, b_B, c_B\)
Multiplication Triple Generation

Multiplication Triple (MT): \((c_A \oplus c_B) = (a_A \oplus a_B)(b_A \oplus b_B)\)

Shares intended for only one party: \(a_A, b_A, c_A \leftrightarrow a_B, b_B, c_B\)

\[
c_A = ((a_A \oplus a_B)(b_A \oplus b_B)) \oplus c_B
\]
Multiplication Triple Generation

Multiplication Triple (MT): \((c_A \oplus c_B) = (a_A \oplus a_B)(b_A \oplus b_B)\)

Shares intended for only one party: \(a_A, b_A, c_A \leftrightarrow a_B, b_B, c_B\)
**Multiplication Triple Sets**

What if $|f|$ is unknown in the init phase?

Generate MT sets of size $2^i$ MTs from random seeds $s_i$

Build $x$ MTs from $\log_2(x)$ MT sets
Android Apps for Secure Computation

Alice

SC App

MT Set Service

G&D MSC Service

Bob

SC App

Wi-Fi Direct
Benchmarks – General

Giesecke & Devrient Mobile Security Card SE 1.0
- microSD JavaCard
- Memory: 75 KB EEPROM / 1750 Byte RAM
- AES: 16 KB/s

Samsung Galaxy S3
- 4x 1.4 GHz ARM CPU
- 1 GB RAM, 16 GB flash storage

Interactive OT: 11 000 MT/s
Init Phase on Smartcard: 5 800 MT/s
Benchmarks – Applications

Scheduling for a week with 392 time slots

<table>
<thead>
<tr>
<th>Scheduling</th>
<th>392 / 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Init</td>
<td>1.3 s</td>
</tr>
<tr>
<td>Setup</td>
<td>0.003 s</td>
</tr>
<tr>
<td>Online</td>
<td>1.3 s</td>
</tr>
<tr>
<td>Ad-Hoc</td>
<td>3.82 s</td>
</tr>
<tr>
<td>[HCE11]</td>
<td>3.82 s</td>
</tr>
<tr>
<td>[HEK12]</td>
<td>---</td>
</tr>
</tbody>
</table>
# Benchmarks – Applications

Scheduling for a week with 392 time slots (16 bit coordinates)

<table>
<thead>
<tr>
<th>Location Scheduling</th>
<th>Scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>f</td>
</tr>
<tr>
<td>$Init$</td>
<td>0.37 s</td>
</tr>
<tr>
<td>$Setup$</td>
<td>1.3 s</td>
</tr>
<tr>
<td>$Online$</td>
<td>0.003 s</td>
</tr>
<tr>
<td>$Ad-Hoc$</td>
<td>1.3 s</td>
</tr>
<tr>
<td>[HCE11]</td>
<td>3.82 s</td>
</tr>
<tr>
<td>[HEK12]</td>
<td>---</td>
</tr>
</tbody>
</table>
Benchmarks – Applications

Scheduling for a week with 392 time slots (16 bit coordinates)
512 contacts with 32 bit each

<table>
<thead>
<tr>
<th></th>
<th>Scheduling</th>
<th>Location Scheduling</th>
<th>Common Contactcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>f</td>
<td>/ d(f)$</td>
<td>392 / 1</td>
</tr>
<tr>
<td>Init</td>
<td>0.37 s</td>
<td>48.5 s</td>
<td>137.9 s</td>
</tr>
<tr>
<td>Setup</td>
<td>1.3 s</td>
<td>1.8 s</td>
<td>2.5 s</td>
</tr>
<tr>
<td>Online</td>
<td>0.003 s</td>
<td>0.82 s</td>
<td>1.9 s</td>
</tr>
<tr>
<td>Ad-Hoc</td>
<td>1.3 s</td>
<td>2.6 s</td>
<td>4.4 s</td>
</tr>
<tr>
<td>[HCE11]</td>
<td>3.82 s</td>
<td>---</td>
<td>1 468.0 s</td>
</tr>
<tr>
<td>[HEK12]</td>
<td>---</td>
<td>---</td>
<td>4.95 s</td>
</tr>
</tbody>
</table>
## Comparison with Related Work

<table>
<thead>
<tr>
<th></th>
<th>$f$ unknown in init phase</th>
<th>low ad-hoc communication</th>
<th>low ad-hoc computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yao's Garbled Circuits</td>
<td>✓</td>
<td>✘</td>
<td>✘</td>
</tr>
<tr>
<td>Token-Based Yao</td>
<td>✘</td>
<td>✓</td>
<td>✘</td>
</tr>
<tr>
<td>GMW</td>
<td>✓</td>
<td>✘</td>
<td>✘</td>
</tr>
<tr>
<td><strong>Our Approach</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion

Mobile Secure Computation is becoming practical

(But requires a smartcard)

Trusted hardware enables secure offline pre-computation

Outlook:

Active Security

Multiple Hardware Tokens
Thanks!

Questions?

Contact: http://encrypted.org