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

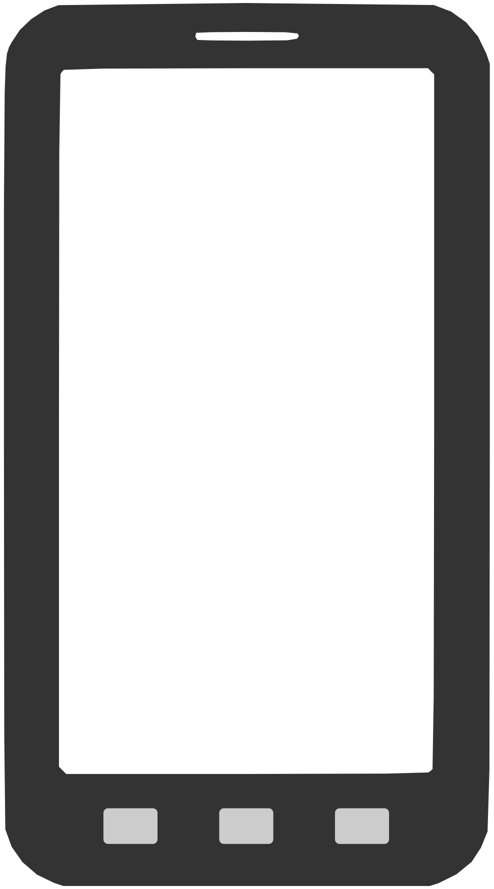
TECHNISCHE UNIVERSITÄT DARMSTADT

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

EC SPRIDE

Motivation

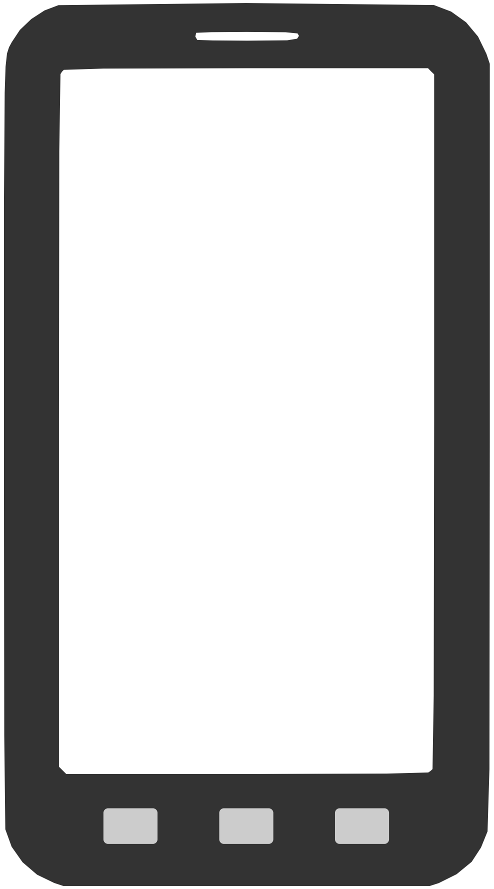
Smartphones are **awesome!**



Motivation

Smartphones are **awesome!**

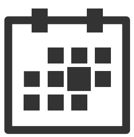
Contacts 

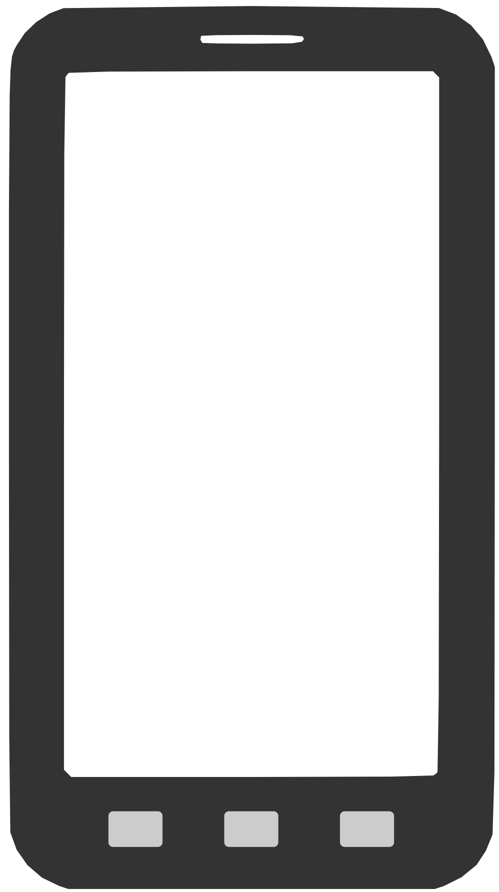


Motivation

Smartphones are **awesome!**

Contacts 

Calendar 



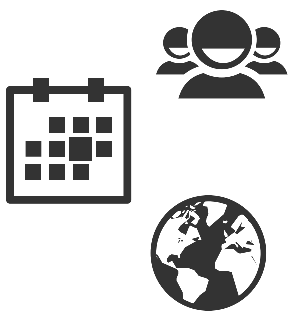
Motivation

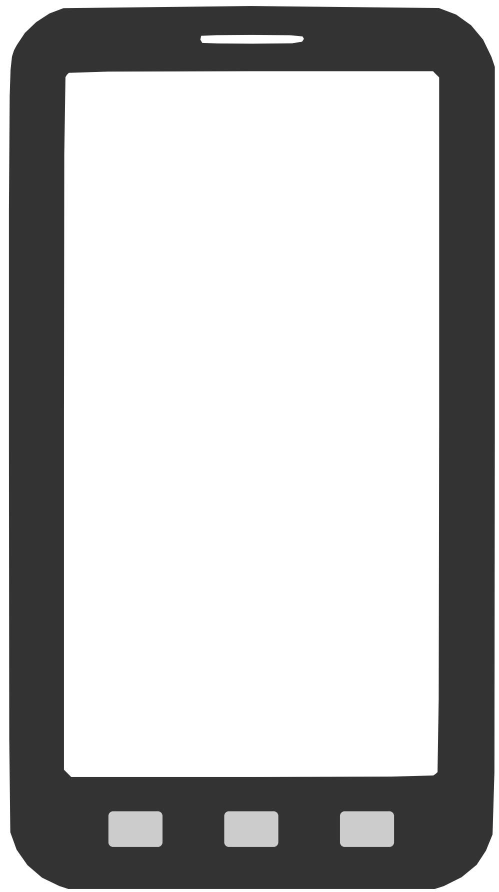
Smartphones are **awesome!**

Contacts

Calendar

Location





Motivation

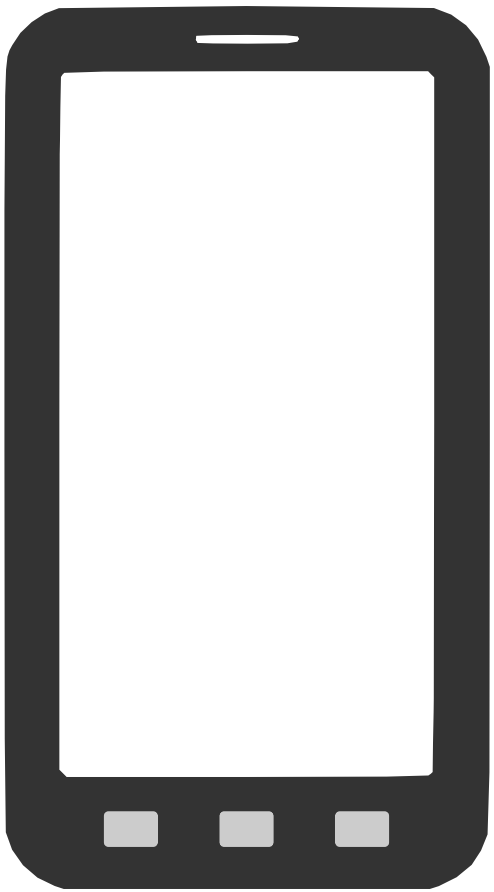
Smartphones are **awesome!**

Contacts

Calendar

Location

Banking

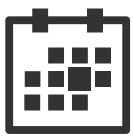


Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Smartphones are **awesome!**

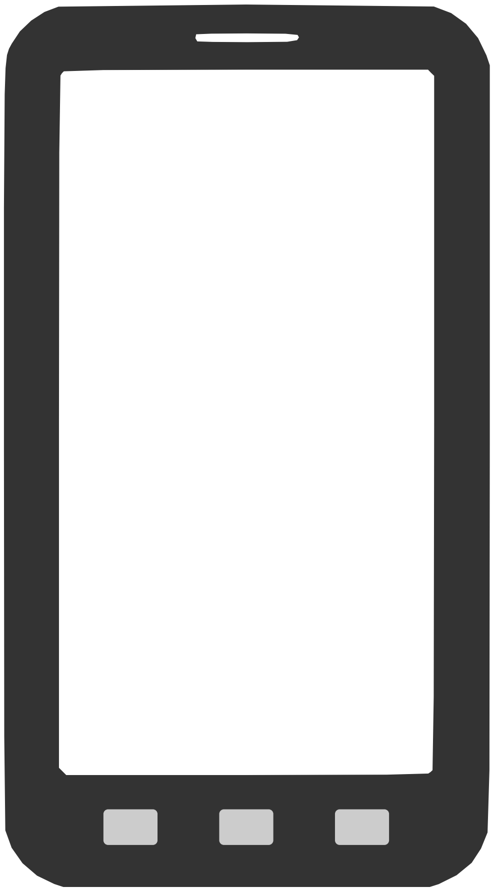
Contacts 

Calendar 

Location 

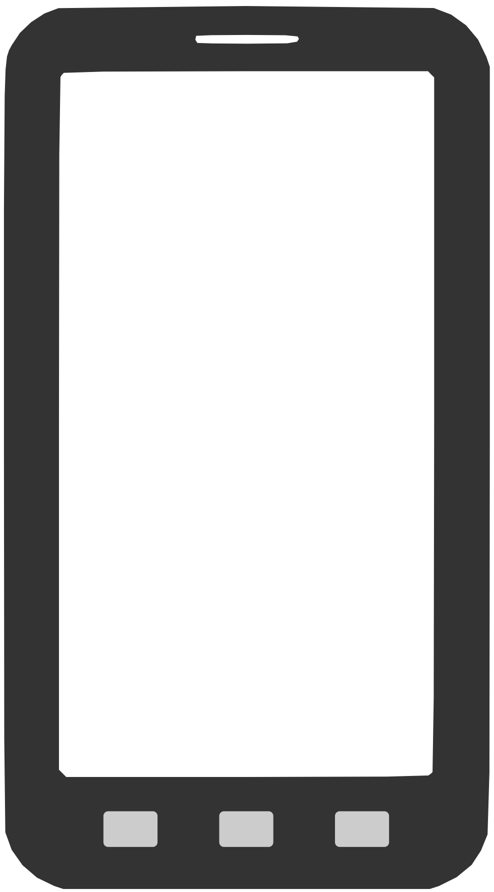
Banking 

Messaging 



Motivation

Smartphones are **awesome!**



Contacts

Calendar

Location

Banking

Messaging

Games

Motivation

Smartphones are **awesome!**

Contacts

Calendar

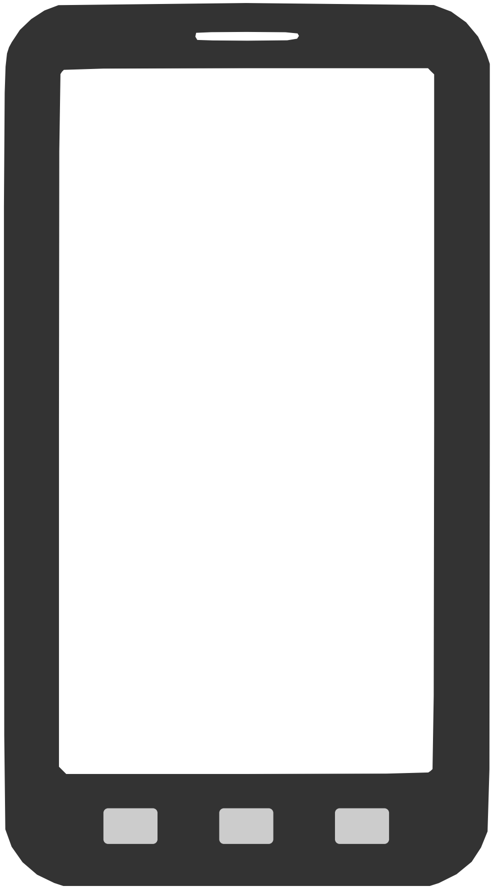
Location

Banking

Messaging

Games

…



Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Smartphones are **awesome!**

Smartphones are **limited…**

Contacts

Calendar

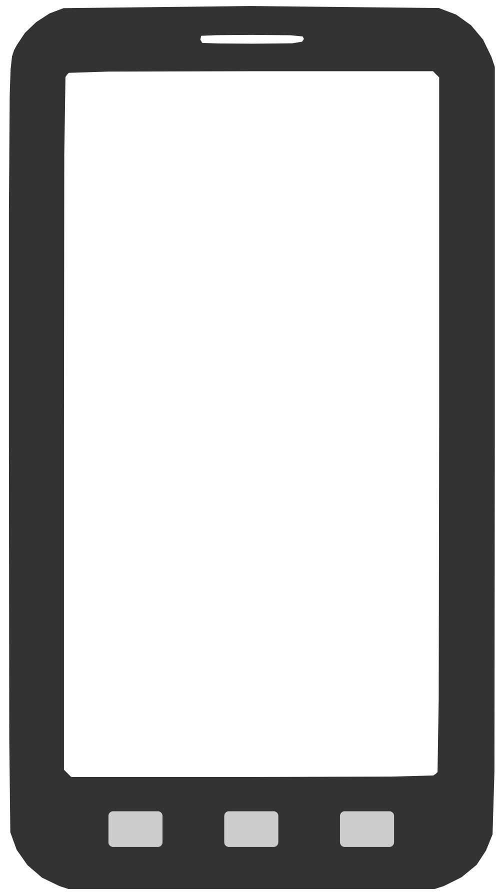
Location

Banking

Messaging

Games

…



Motivation

Smartphones are **awesome!**

Contacts

Calendar

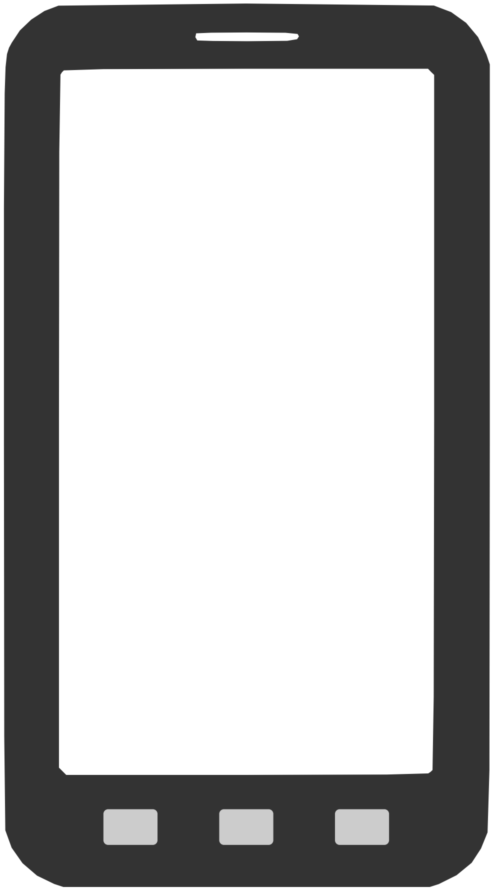
Location

Banking

Messaging

Games

…



Smartphones are **limited…**

Computation

TECHNISCHE UNIVERSITÄT DARMSTADT

Motivation

Smartphones are **awesome!**

Contacts

Calendar

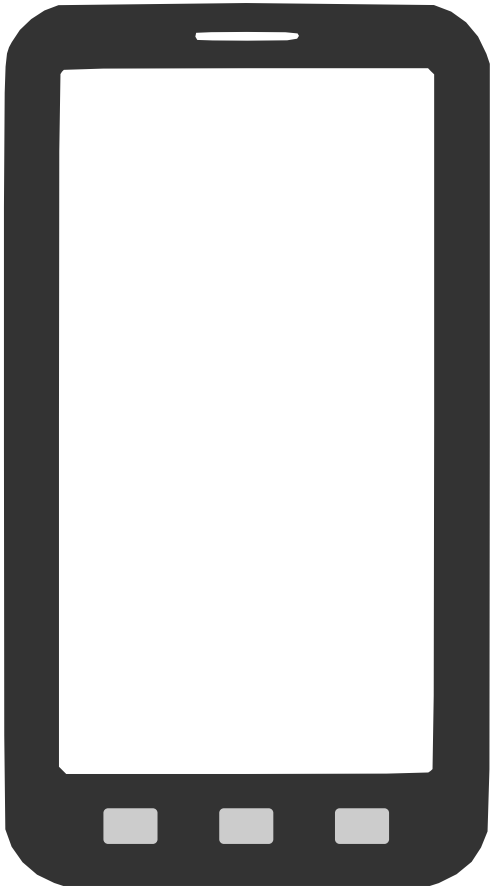
Location

Banking

Messaging

Games

…



Smartphones are **limited…**

Computation

Memory

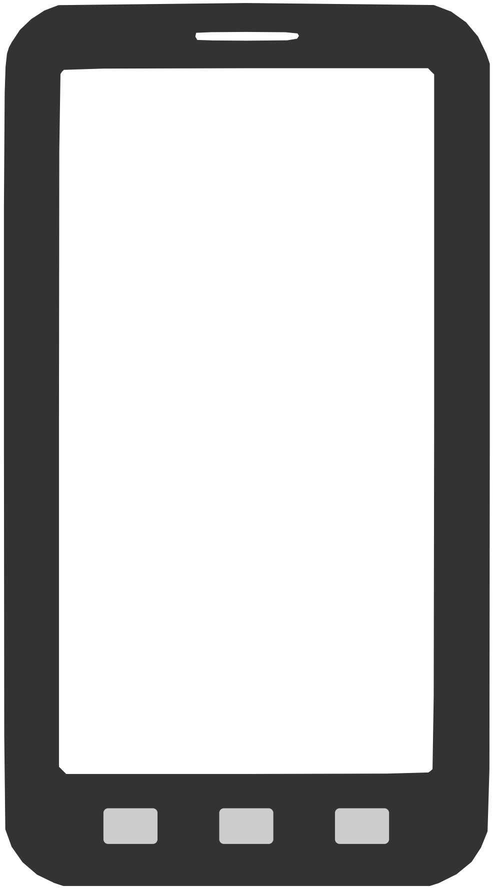
EC SPRIDE

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Smartphones are **awesome!**

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



Smartphones are **limited…**

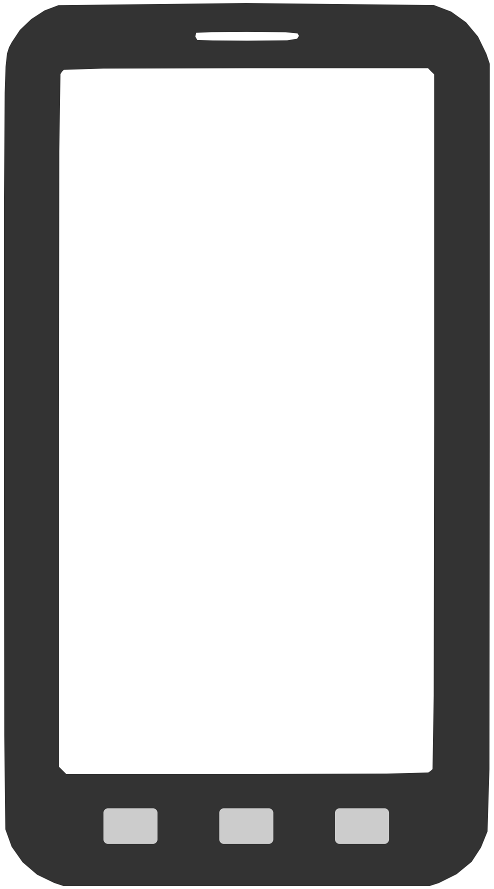
- Computation
- Memory
- Communication

EC SPRIDE

Motivation

Smartphones are **awesome!**

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



Smartphones are **limited…**

- Computation
- Memory
- Communication
- Battery Life

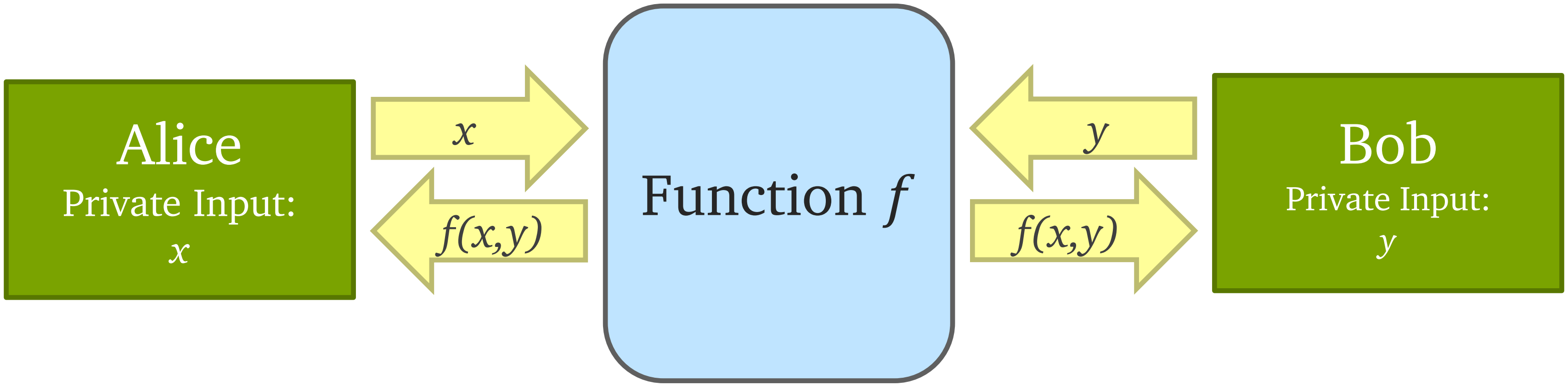
Generic Secure Computation

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]

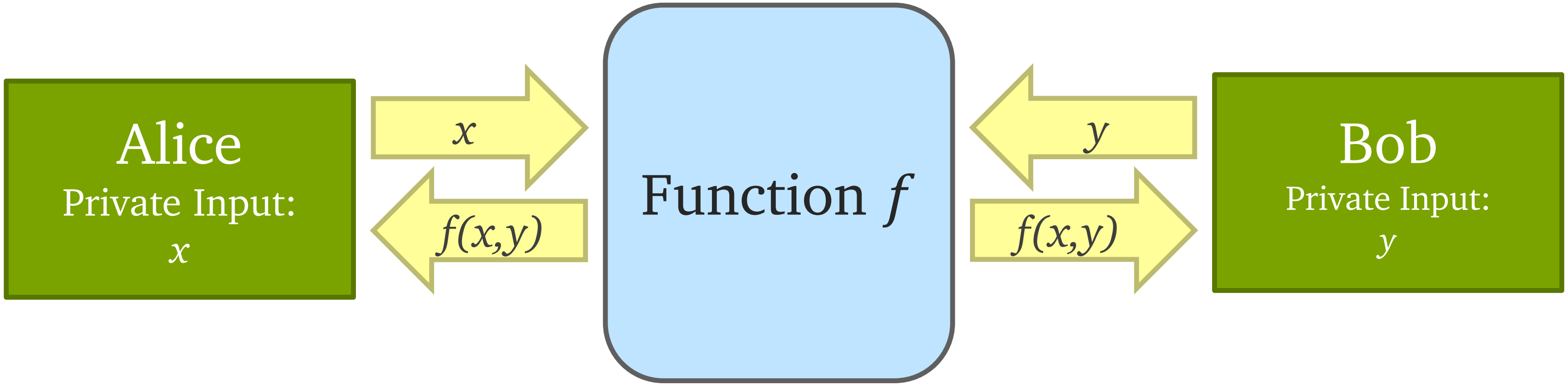
$O(|f|)$ symmetric crypto

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

$\Rightarrow$ Too much for mobile devices

Fairplay [MNPS04], FastGC [HEKM11]

Mobile Yao [HCE11]



Secure Computation Applications

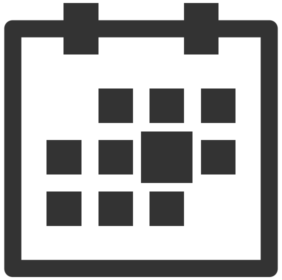
Finding shared contacts

Alice	Bob
481516	666777
101010	**101010**
123456	**234201**
234201	000911

Secure Computation Applications

Finding shared contacts

Alice	Bob
481516	666777
101010	**101010**
123456	**234201**
234201	000911

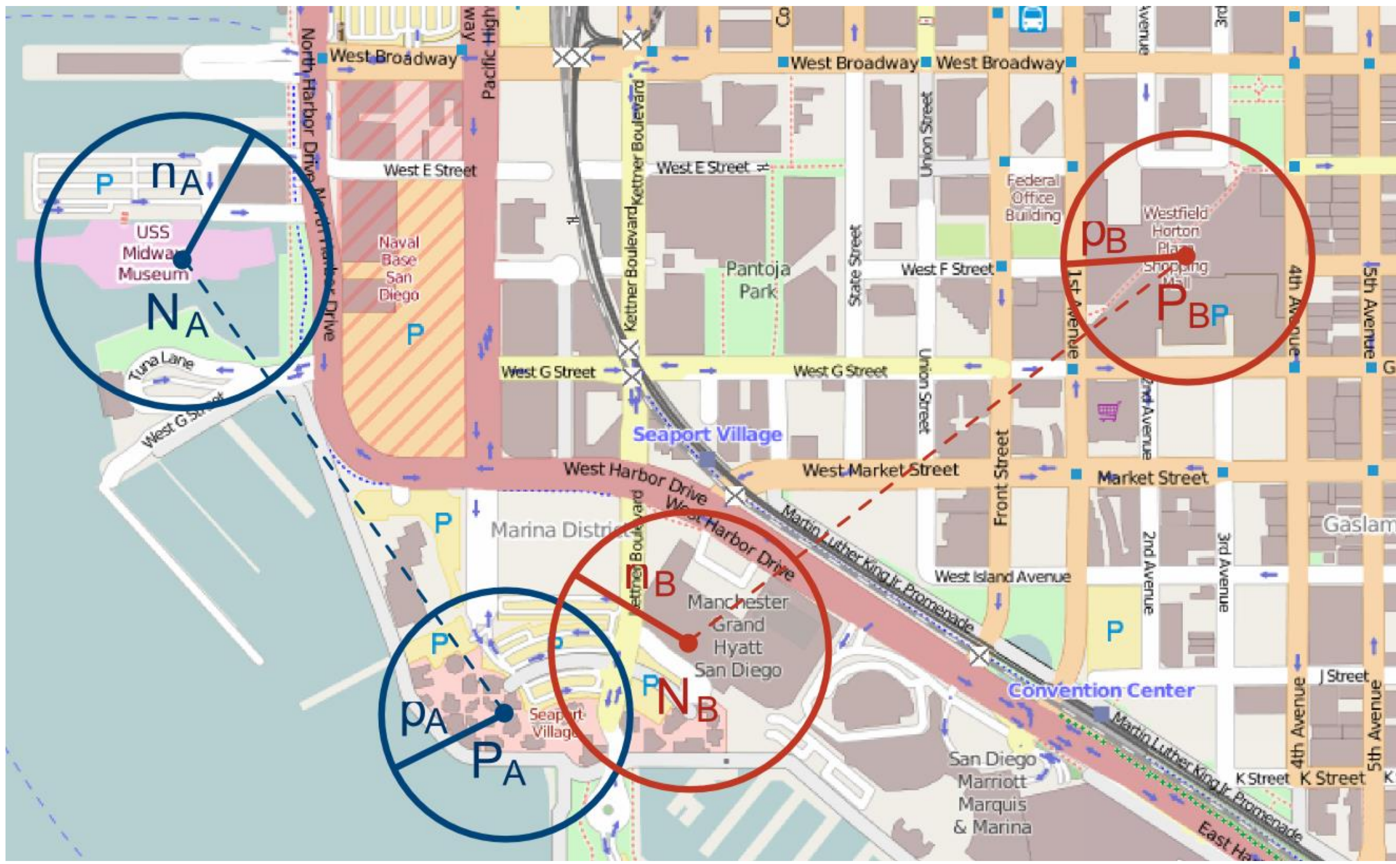


Scheduling a meeting

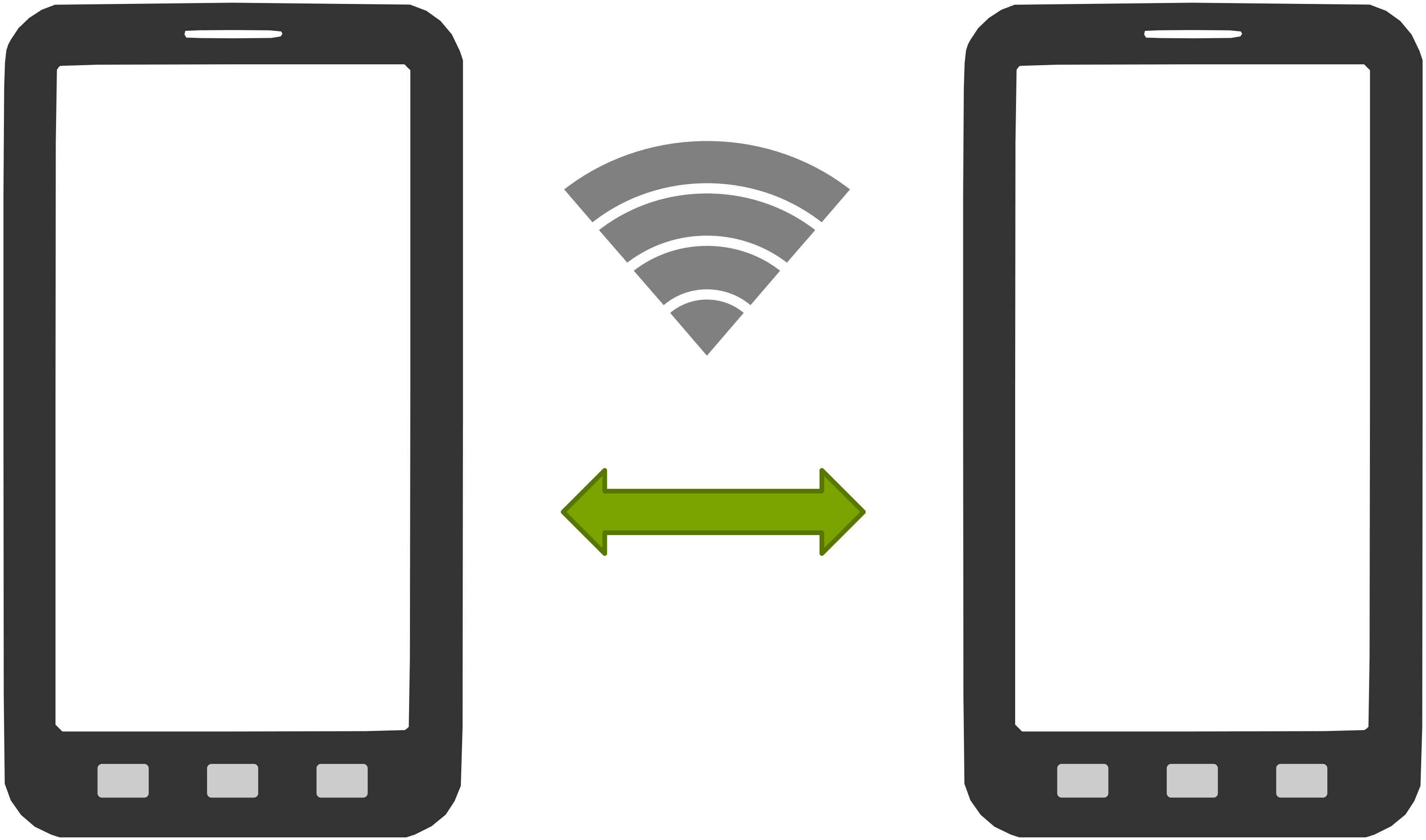
Alice	Bob	=
0	1	0
0	**0**	**0**
1	**1**	**1**
1	0	0

Secure Computation Applications

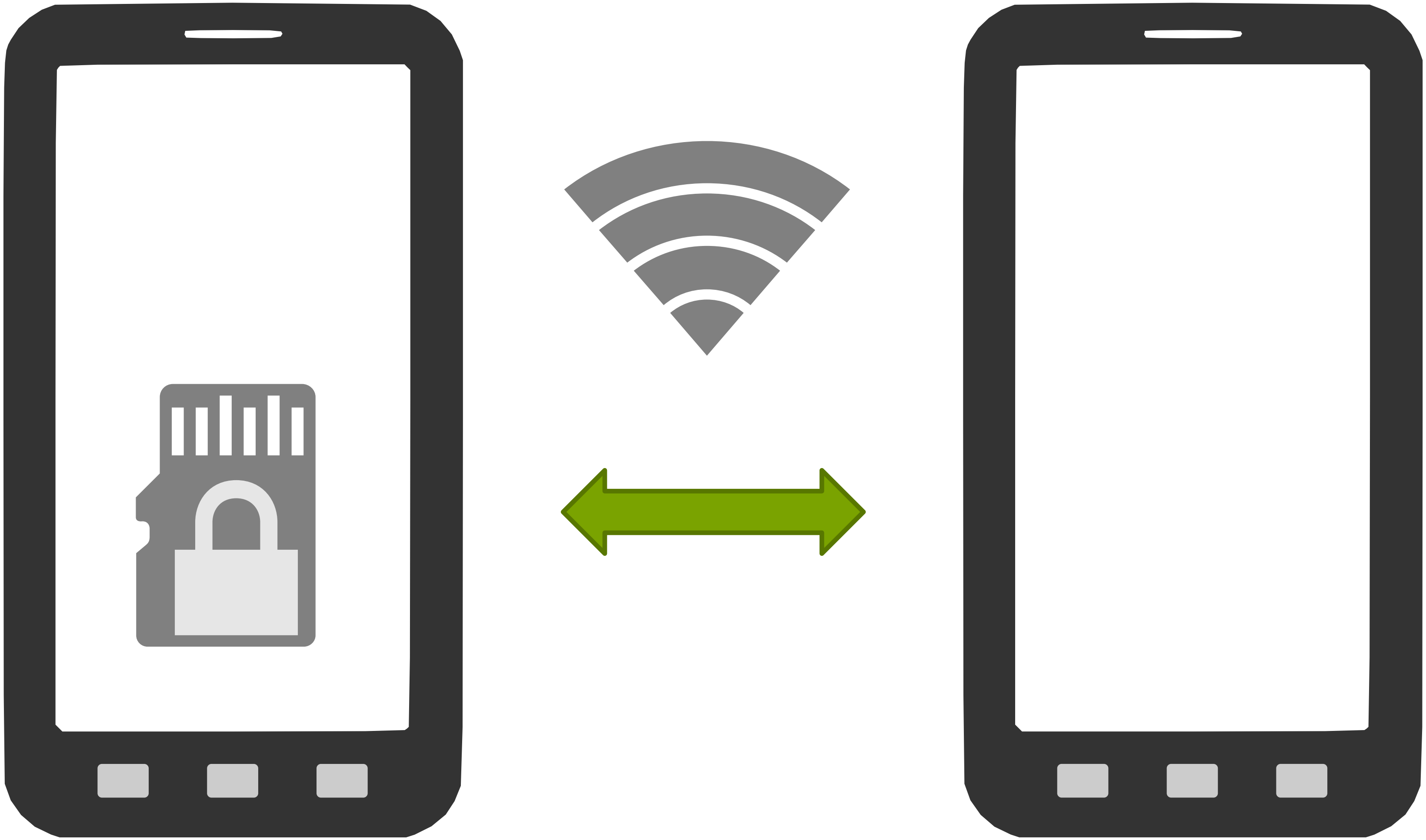
Scheduling a meeting with location information



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 — Dept. of Computer Sc., Technion, Haifa, Israel

Silvio Micali — Lab. for Computer Sc., MIT, Cambridge, MA 02139

Avi Wigderson — Inst. of Math. and CS, Hebrew University, Jerusalem, Israel

Abstract

We present a polynomial-time algorithm that, given as a input the description of a game with incomplete information and any number of players, produces a protocol for playing the game that leaks 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 completeness theorem* for the class of distributed protocols with honest majority. Such completeness theorem is optimal in the sense that, if the majority of the players is not honest, some protocol problems have no efficient solution[C].

1. Introduction

Before discussing how to "make playable" a general game with incomplete information (which we do in section 6) let us address the problem of making playable a special class of games, the *Turing machine games* (*Tm-games* for short).

Informally, n parties, respectively and individually owning secret inputs $x_1, \ldots, x_n$, would like to *correctly* run a given Turing machine M on these x_i's while keeping the maximum possible *privacy* about them. That is, they want to compute $y = M(x_1, \ldots, x_n)$ without revealing more about the x_i's than it is already contained in the value y itself. For instance, if M computes the sum of the x_i's, every single player should not be able to learn more than the sum of the inputs of the other parties. Here M may very well be a probabilistic Turing machine. In this case, all players want to agree on a single string y, selected with the right probability distribution, as M's output.

The correctness and privacy constraint of a Tm-game can be easily met with the help of an extra, trusted party P. Each player i simply gives his secret input x_i to P. P will privately run the prescribed Turing machine, M, on these inputs and publically announce M's output. Making a Tm-game playable essentially means that the correctness and privacy constraints can be satisfied by the n players themselves, without invoking any extra party. Proving that Tm-games are playable retains most of the flavor and difficulties of our general theorem.

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 one input we write "$A(\cdot)$", if it receives two inputs we write $A(\cdot,\cdot)$ and so on.

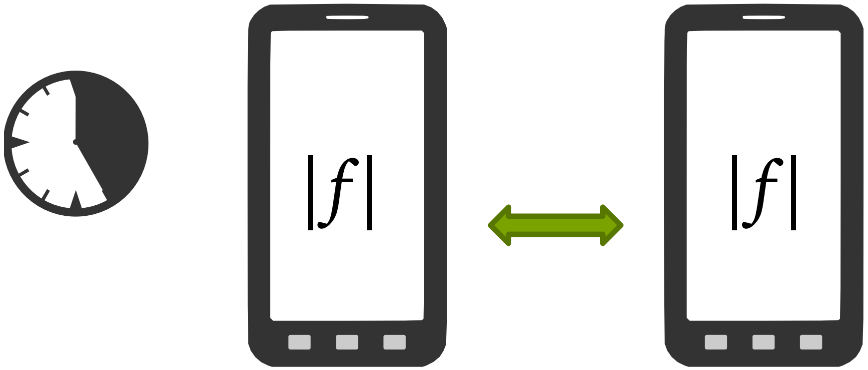
RV will stand for "random variable"; in this paper we only consider RVs that assume values in

Work partially supported by NSF grants DCR-8509905 and DCR-8413577, an IBM post-doctoral fellowship and an IBM faculty development award. The work was done when the first author was at the Laboratory for Computer Science at MIT; and the second author at the mathematical Sciences Research Institute at UC-Berkeley.

218

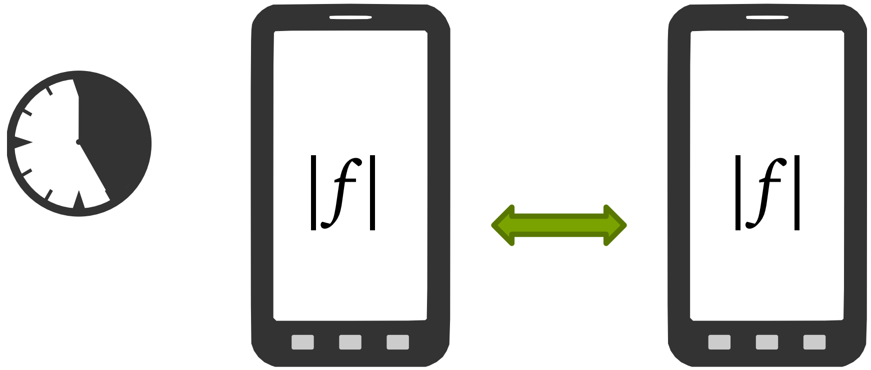
Secure Computation Phases

Setup Phase

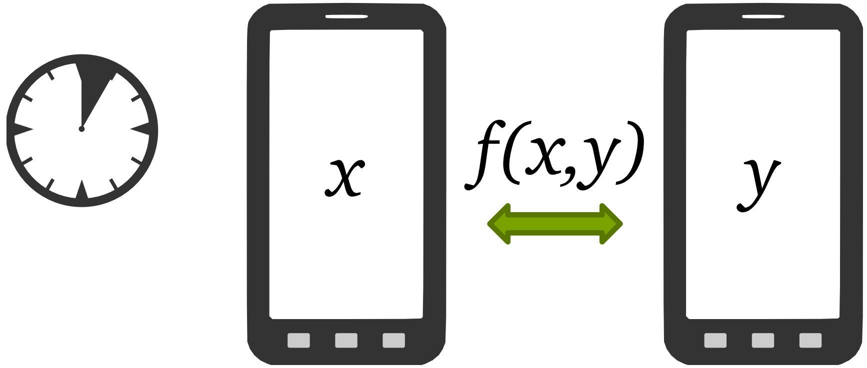


Secure Computation Phases

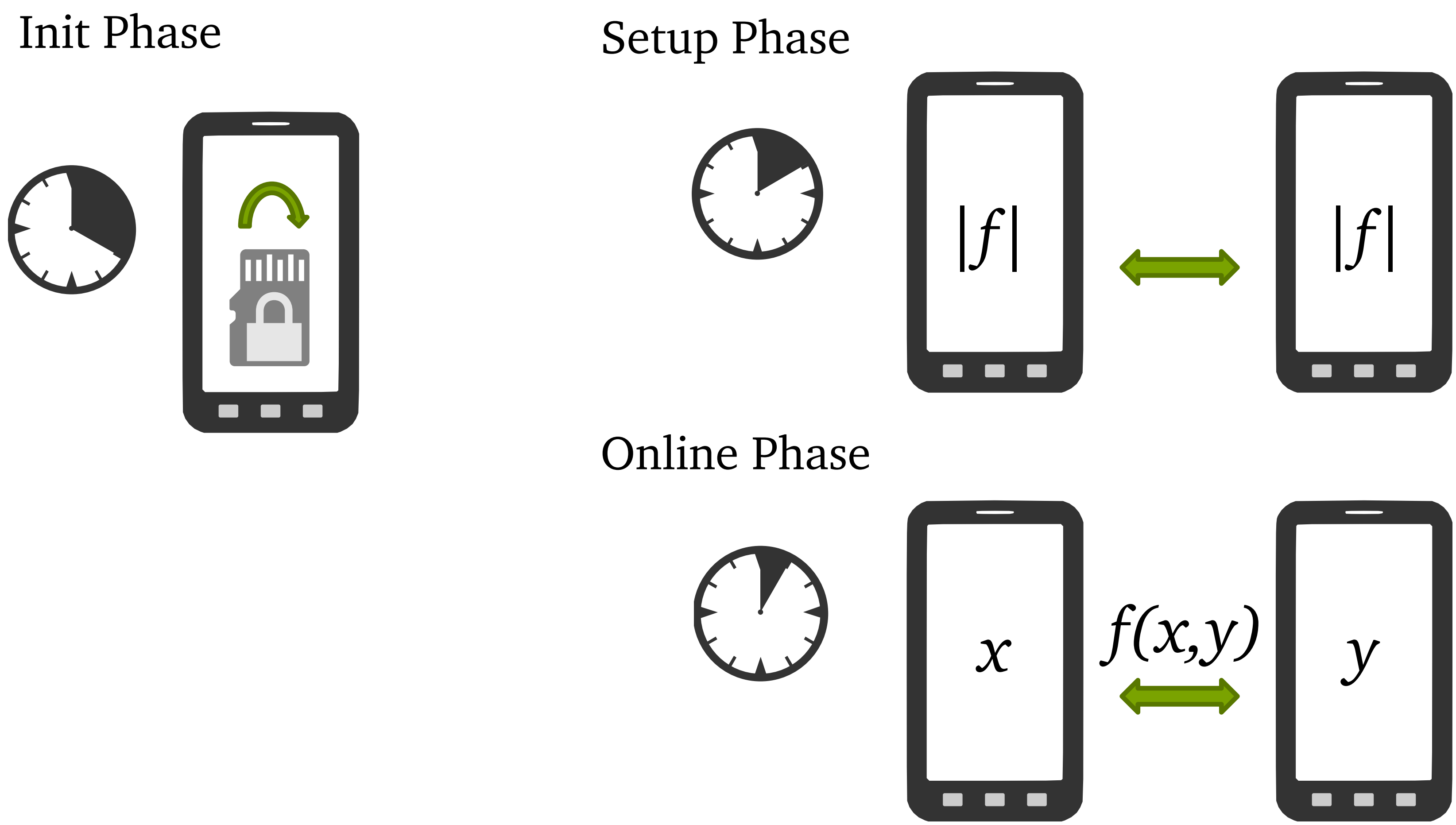
Setup Phase



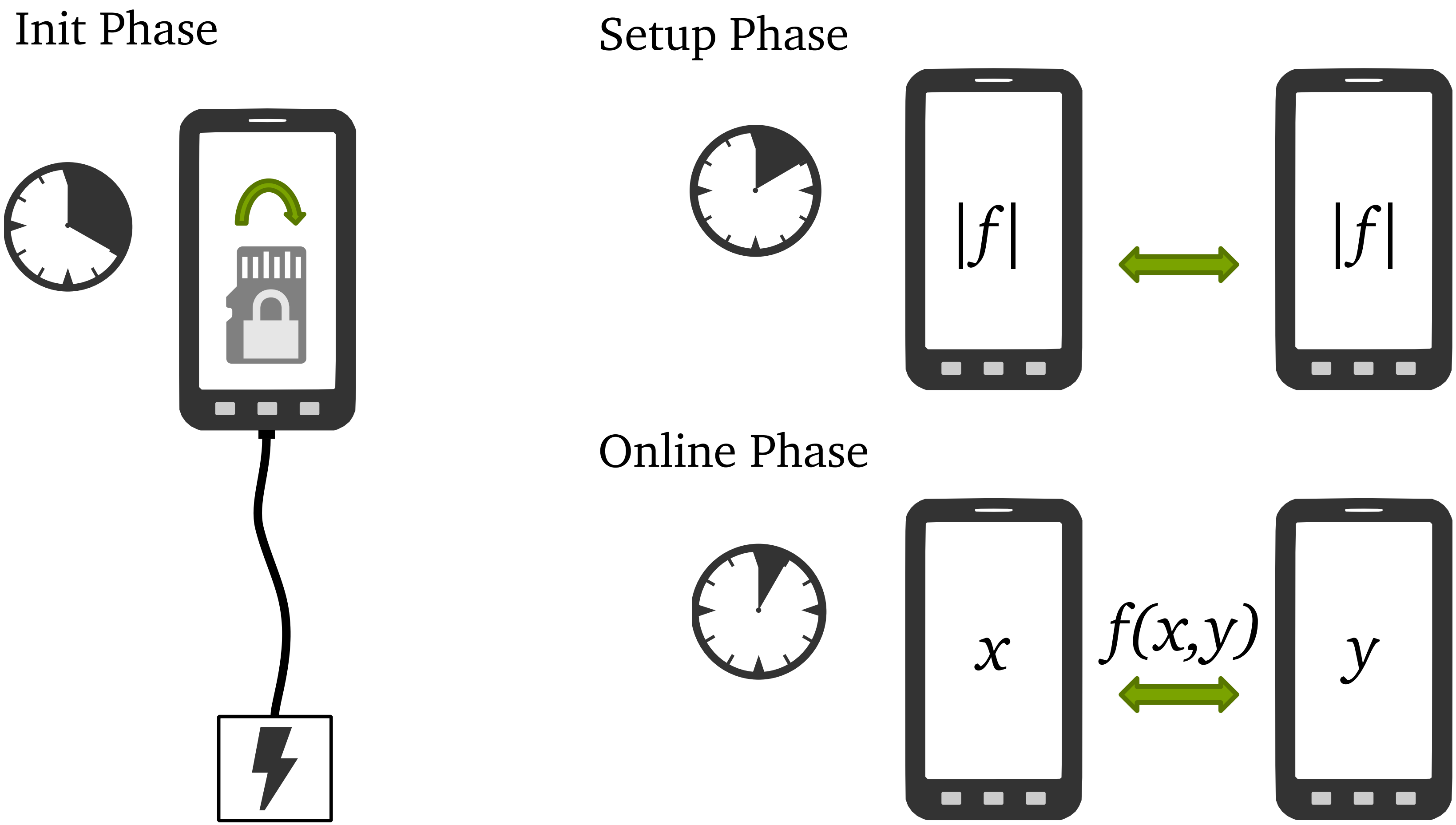
Online Phase



Secure Computation Phases



Secure Computation Phases



Multiplication Triple Generation

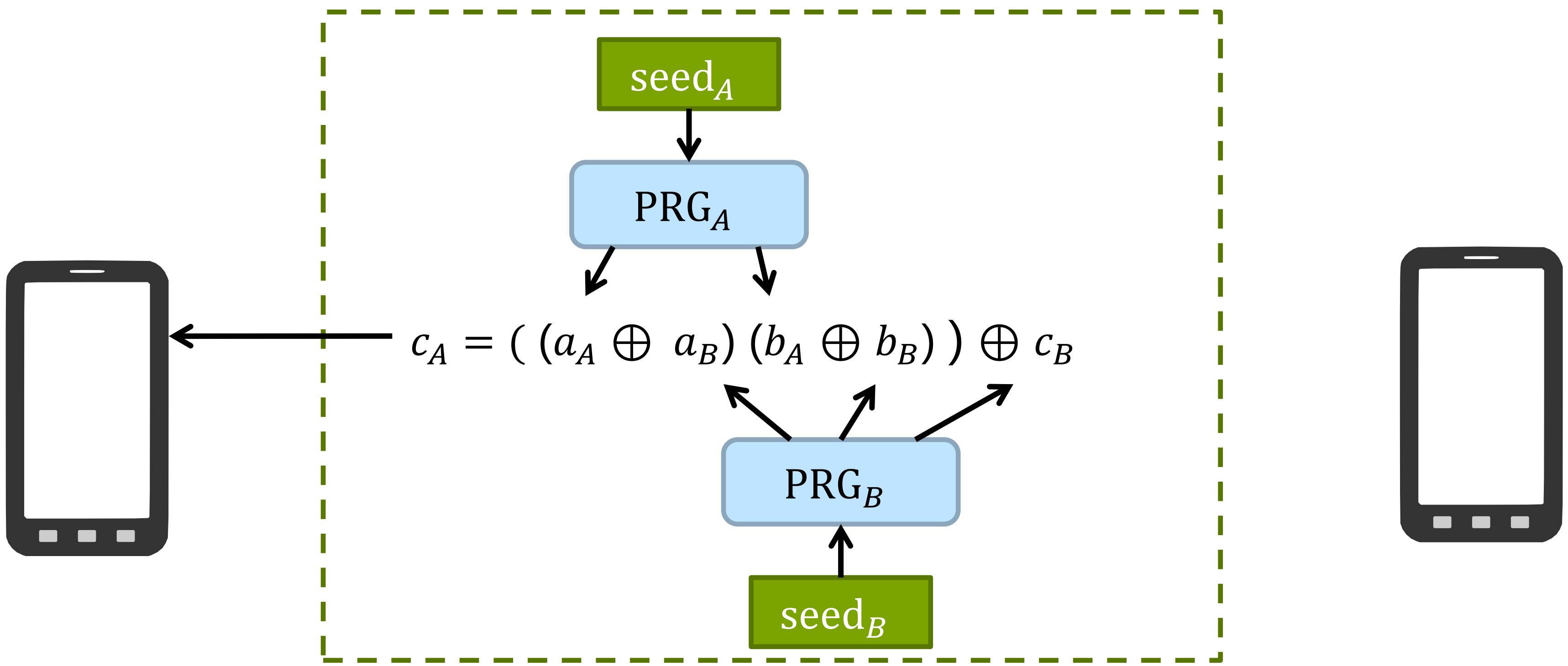
Multiplication Triple (MT): $(c_A \oplus c_B) = (a_A \oplus a_B)(b_A \oplus b_B)$

Shares intended for only one party: $a_A, b_A, c_A \Leftrightarrow a_B, b_B, c_B$

Multiplication Triple Generation

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

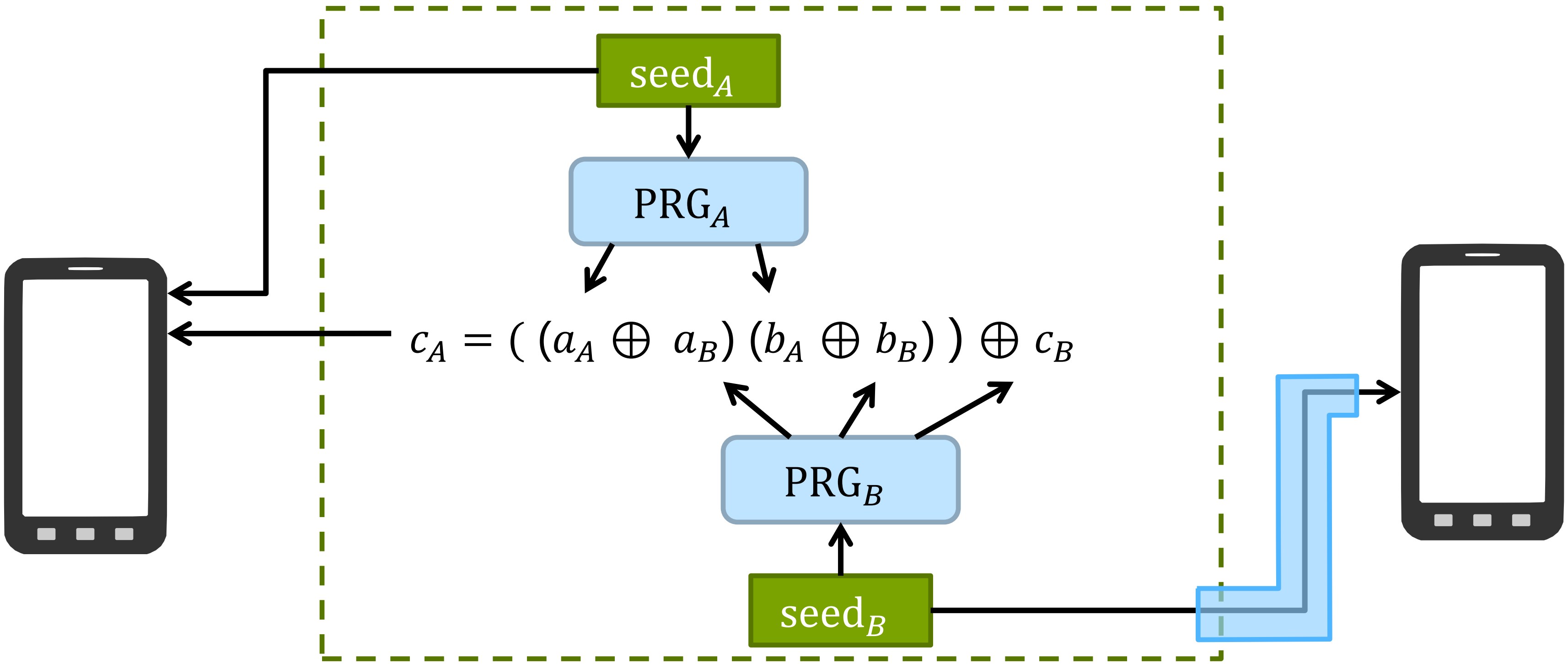
Shares intended for only one party: $a_A, b_A, c_A \Leftrightarrow a_B, b_B, c_B$



Multiplication Triple 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: $a_A, b_A, c_A \Leftrightarrow a_B, b_B, c_B$

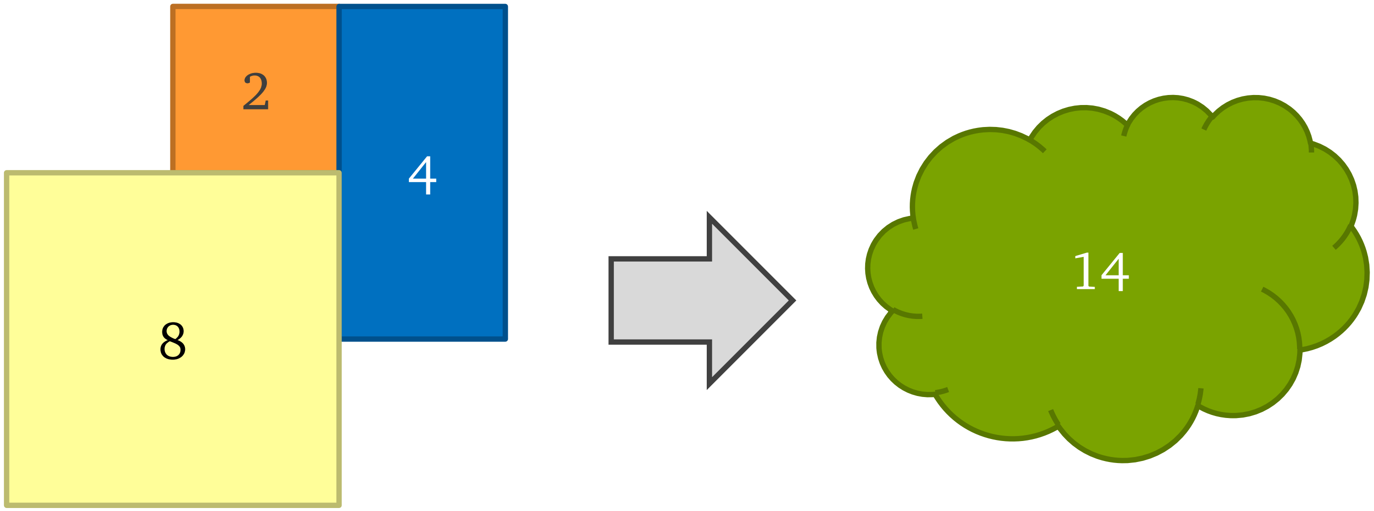


Multiplication Triple Sets

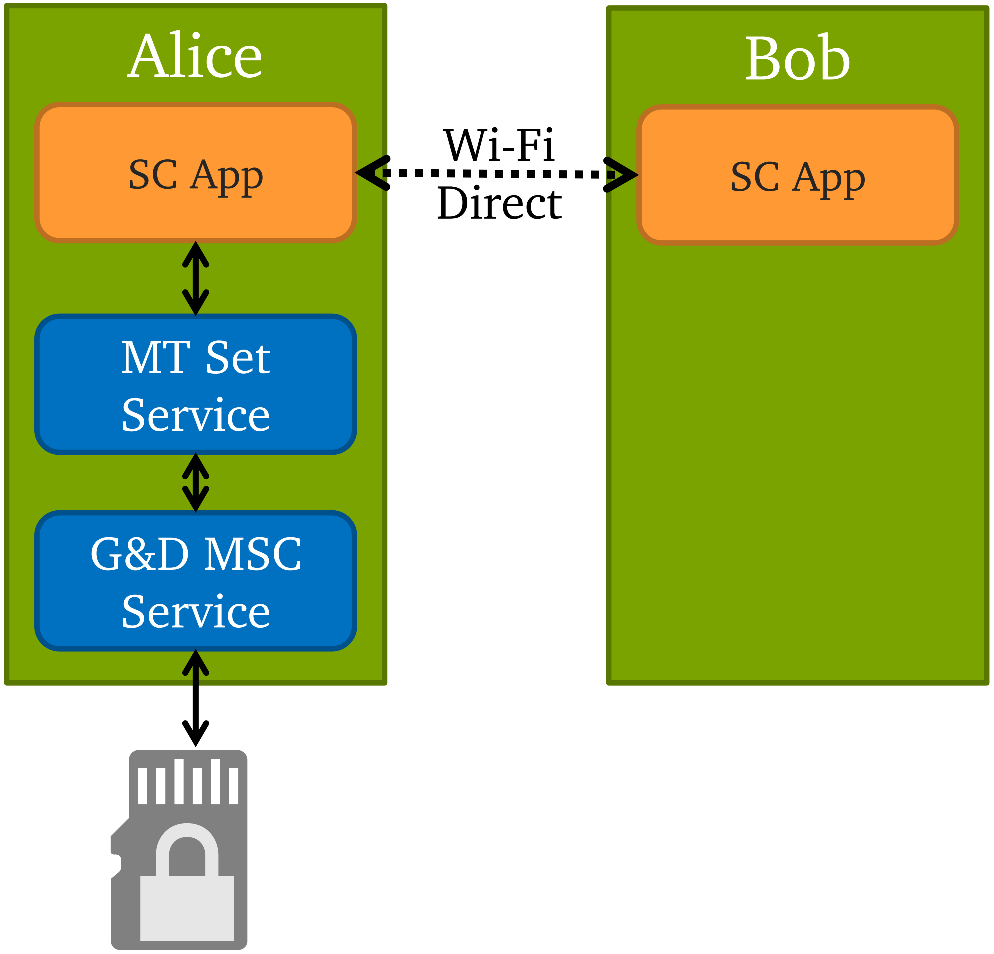
What if $|f|$ is unknown in the init phase?

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



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

	Scheduling
$\|f\|$ / d(f)	392 / 1
Init	*0.37 s*
Setup	1.3 s
Online	0.003 s
Ad-Hoc	**1.3 s**
[HCE11]	3.82 s
[HEK12]	---

Benchmarks – Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

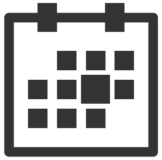
Scheduling for a week with 392 time slots (16 bit coordinates)

	Scheduling	Location Scheduling
$\|f\|$ / d(f)	392 / 1	280 605 / 87
Init	*0.37 s*	*48.5 s*
Setup	1.3 s	1.8 s
Online	0.003 s	0.82 s
Ad-Hoc	**1.3 s**	**2.6 s**
[HCE11]	3.82 s	---
[HEK12]	---	---

Benchmarks – Applications

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

512 contacts with 32 bit each



	Scheduling	Location Scheduling	Common Contacts
$\|f\|$ / $d(f)$	392 / 1	280 605 / 87	799 232 / 79
Init	*0.37 s*	*48.5 s*	*137.9 s*
Setup	1.3 s	1.8 s	2.5 s
Online	0.003 s	0.82 s	1.9 s
Ad-Hoc	**1.3 s**	**2.6 s**	**4.4 s**
[HCE11]	3.82 s	---	1 468.0 s
[HEK12]	---	---	4.95 s

Comparison with Related Work

	f unknown in init phase	low ad-hoc communication	low ad-hoc computation
Yao's Garbled Circuits	✓	✗	✗
Token-Based Yao	✗	✓	✗
GMW	✓	✗	✗
Our Approach	✓	✓	✓

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://encrypto.de**