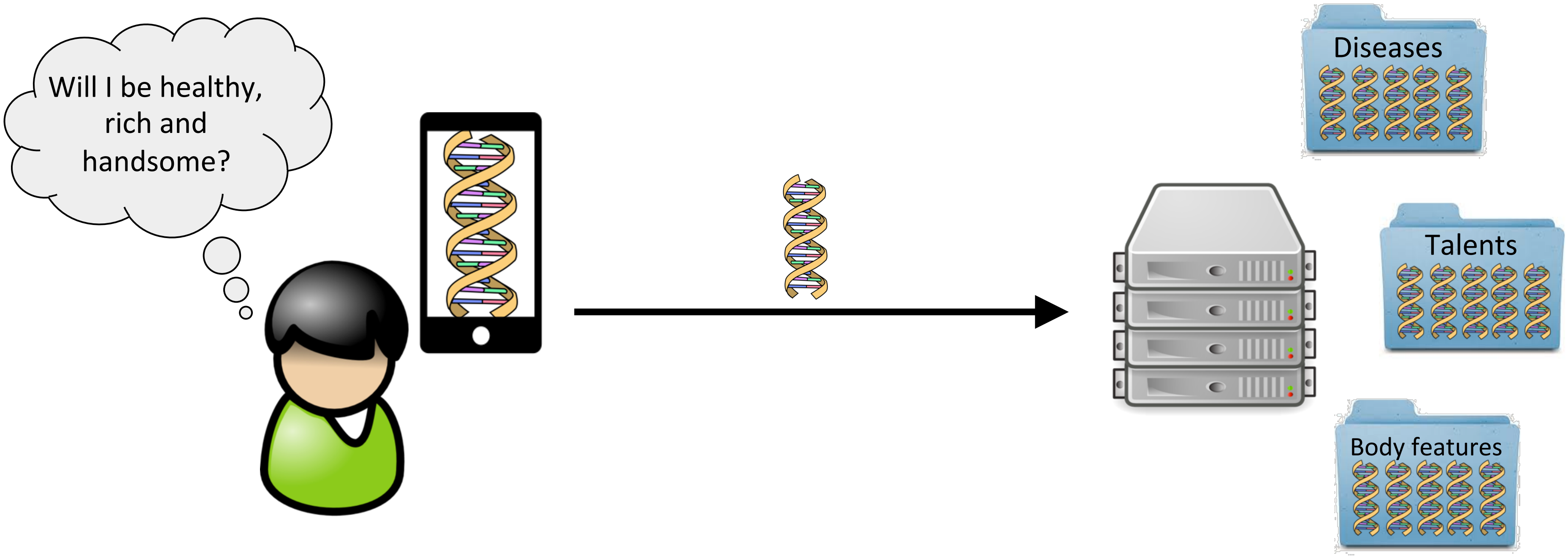
A Comparison of Secure Two-Party Computation Frameworks

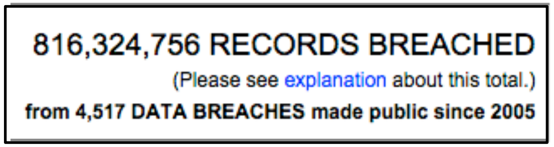
Jan Henrik Ziegeldorf, Jan Metzke, Martin Henze, Klaus Wehrle

Communication and Distributed Systems (COMSYS), RWTH Aachen, Germany

Motivating Scenario: Genetic Testing



Data leaks


816,324,756 RECORDS BREACHED

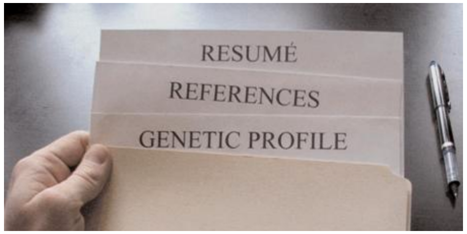
(Please see explanation about this total.)

from 4,517 DATA BREACHES made public since 2005

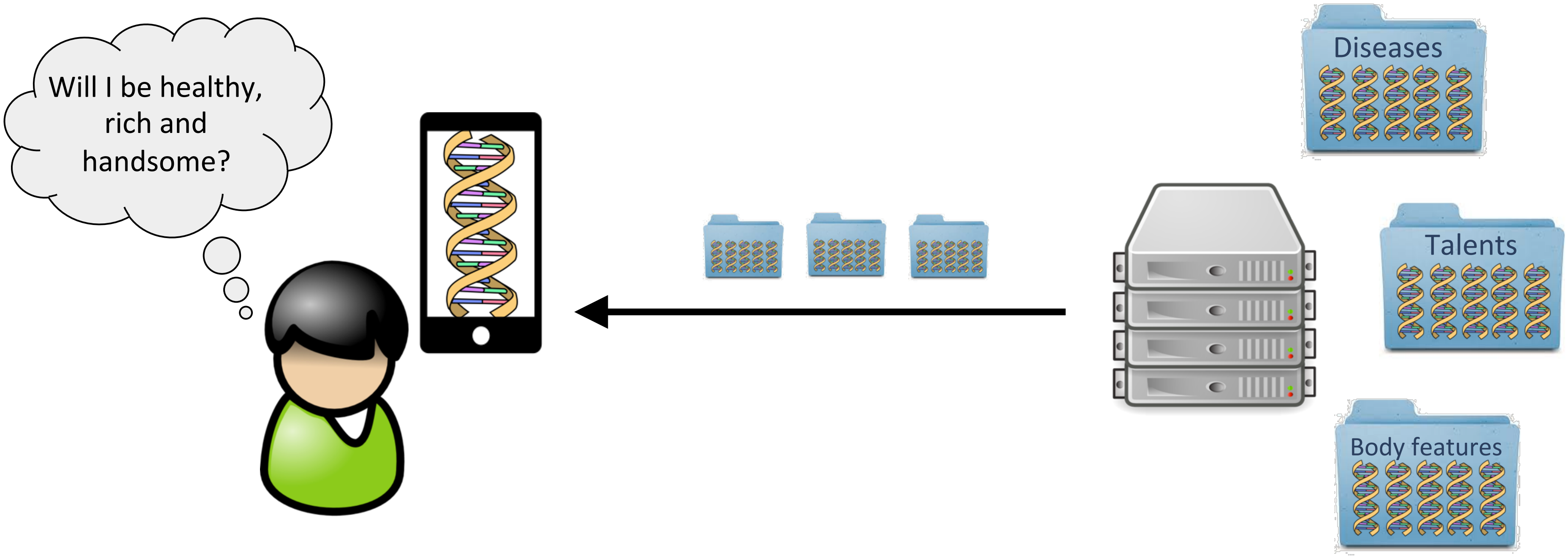
www.privacyrights.org/data-breach

Identification

Discrimination



Motivating Scenario: Genetic Testing



IPR & Business Secrets

GTLDNA GENETIC TESTING LABORATORIES INC.

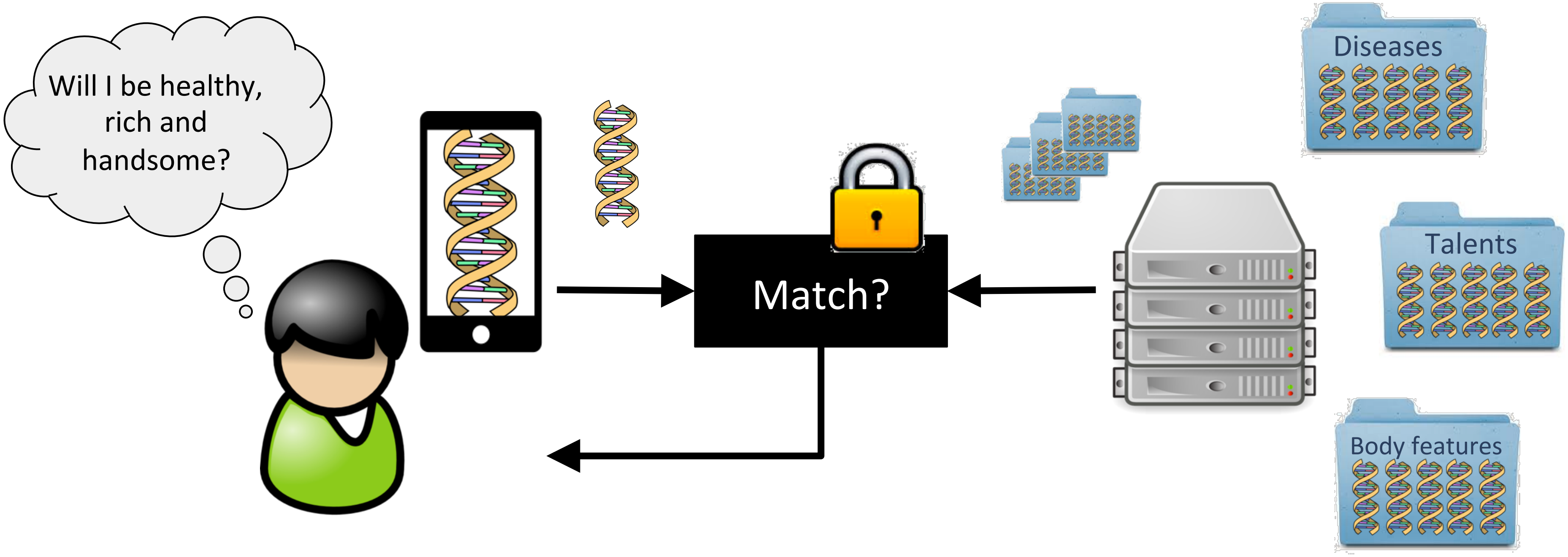
MDL Molecular Diagnostics Laboratories

genetica DNA Laboratories, inc.

23andMe

AFRICAN ANCESTRY TRACE YOUR DNA. FIND YOUR ROOTS.

Motivating Scenario: Genetic Testing

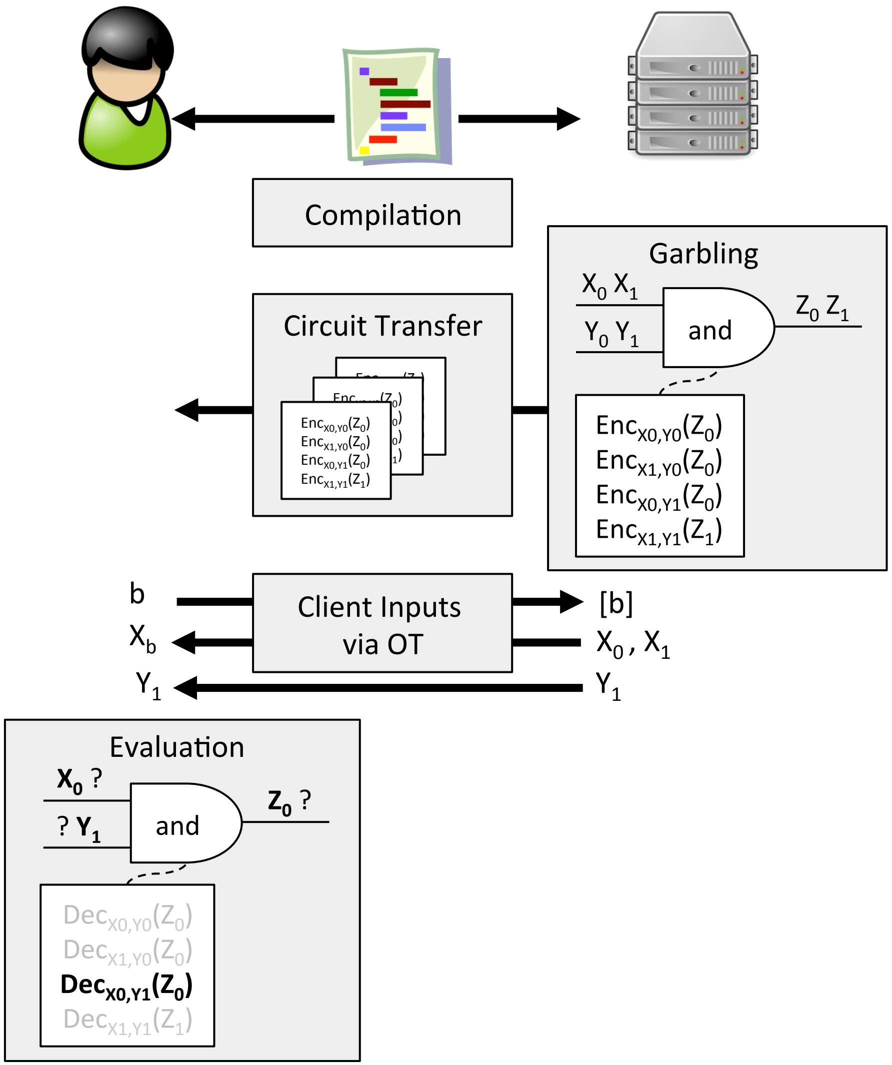


SECURE TWO-PARTY COMPUTATION (STC)

- Rigorous privacy protection
- Any efficiently computable functionality

Two flavors of STC

GARBLED CIRCUITS



Compilation

Garbling

X_0 X_1

Y_0 Y_1

and

Z_0 Z_1

$Enc_{X0,Y0}(Z_0)$
$Enc_{X1,Y0}(Z_0)$
$Enc_{X0,Y1}(Z_0)$
$Enc_{X1,Y1}(Z_1)$

Circuit Transfer

$Enc_{X0,Y0}(Z_0)$
$Enc_{X1,Y0}(Z_0)$
$Enc_{X0,Y1}(Z_0)$
$Enc_{X1,Y1}(Z_1)$

b

X_b

Y_1

Client Inputs via OT

[b]

X_0 , X_1

Y_1

Evaluation

$\mathbf{X_0}$?

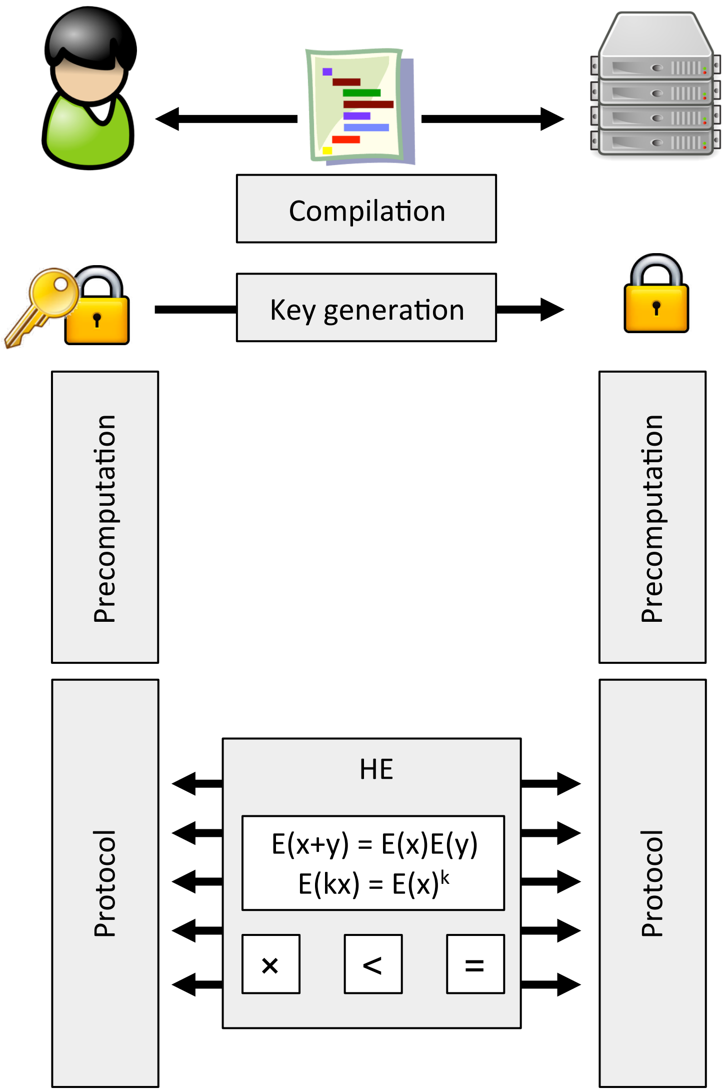
? $\mathbf{Y_1}$

and

$\mathbf{Z_0}$?

$Dec_{X0,Y0}(Z_0)$
$Dec_{X1,Y0}(Z_0)$
$\mathbf{Dec_{X0,Y1}(Z_0)}$
$Dec_{X1,Y1}(Z_1)$

HOMOMORPHIC ENC



Compilation

Key generation

Precomputation

Precomputation

Protocol

Protocol

HE

$E(x+y) = E(x)E(y)$
$E(kx) = E(x)^k$

× < =

Two flavors of STC

GARBLED CIRCUITS

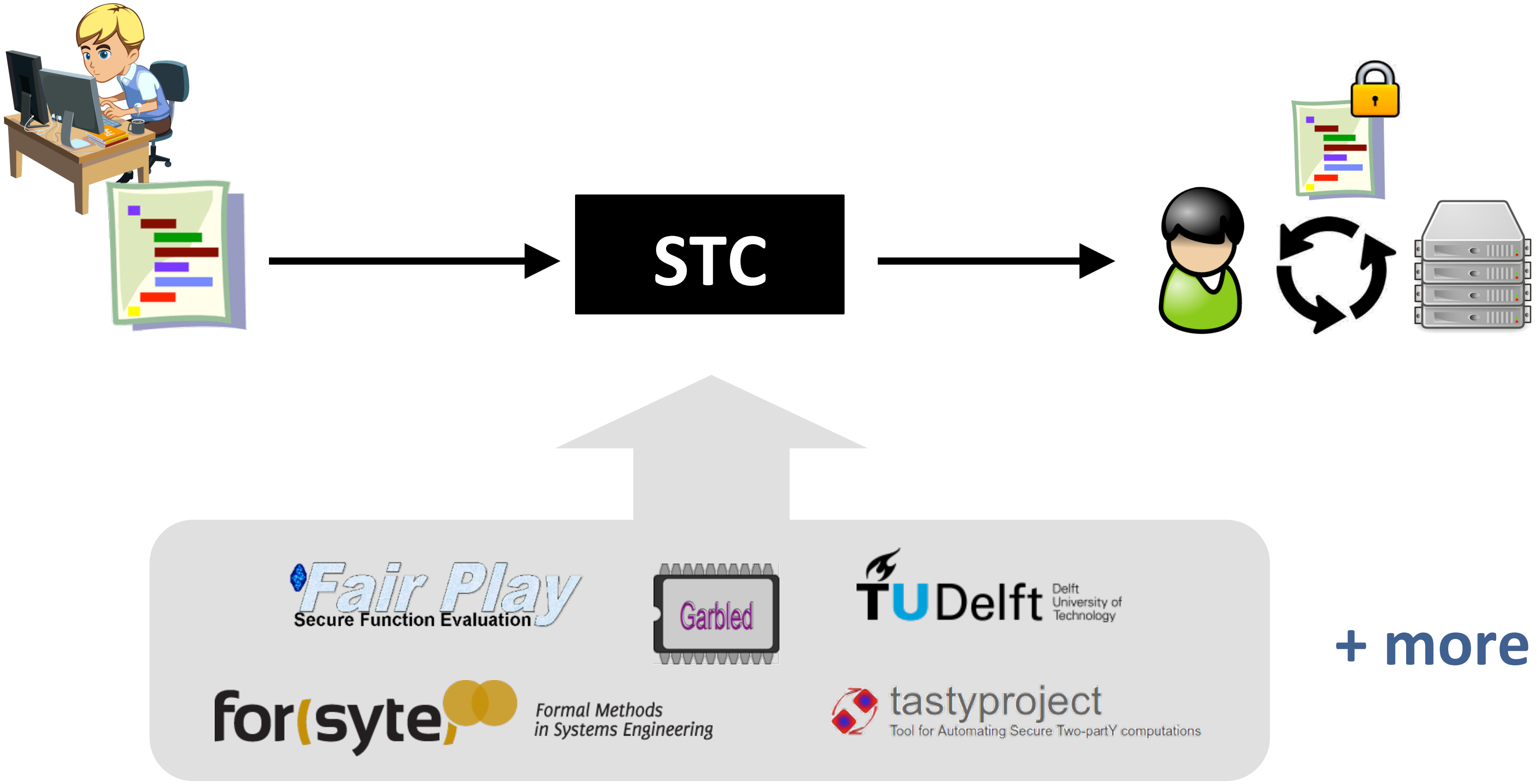
HOMOMORPHIC ENC

This presentation

IS STC A PRACTICAL TOOL FOR PRIVACY ENGINEERS?

Desiderata

GOAL: USE STC AS BLACKBOX


STC
Fair Play
Secure Function Evaluation
Garbled
TU Delft
Delft University of Technology
for(syte)
Formal Methods in Systems Engineering
tastyproject
Tool for Automating Secure Two-partY computations

+ more

STC in the wild

SO, WHY IS STC RARELY USED PRACTICALLY?

Processing Overheads

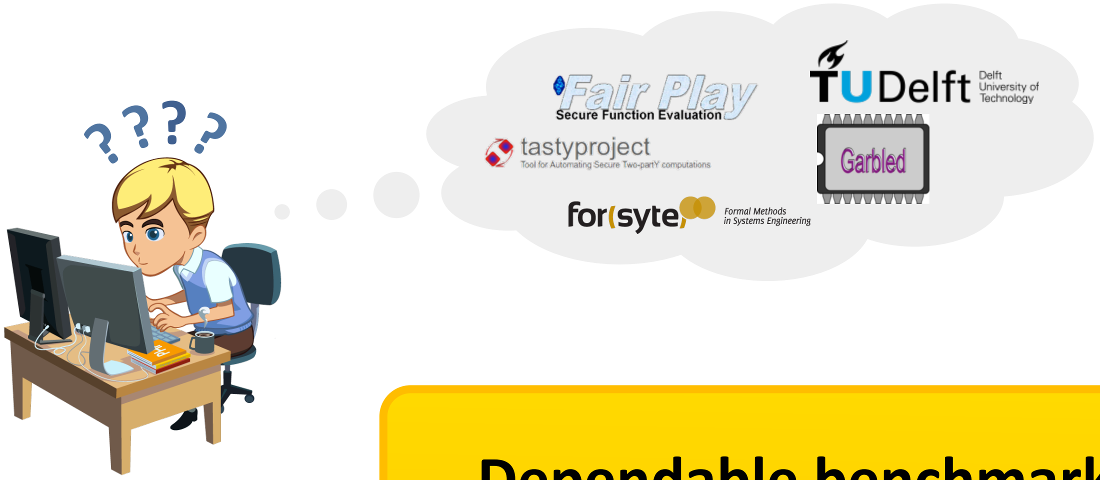
- Crypto ops
- Data blow-up
- Memory

Communication Overheads

- Interaction
- Data blow-up

Development & Usability

- Language support
- Abstractions
- (Documentation)

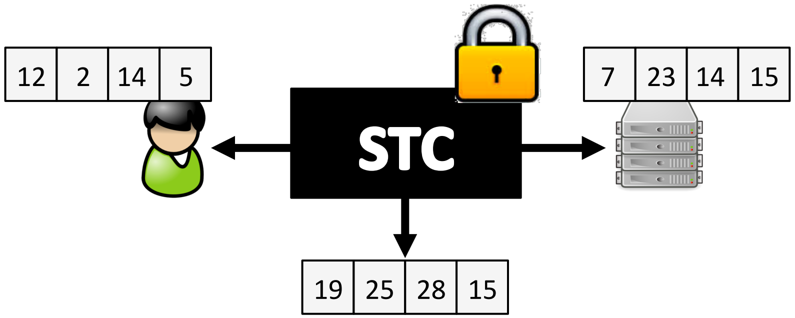

? ? ? ?
Fair Play
Secure Function Evaluation
tastyproject
Tool for Automating Secure Two-partY computations
for(syte
Formal Methods in Systems Engineering
TU Delft
Delft University of Technology
Garbled

Dependable benchmarks and comparison!

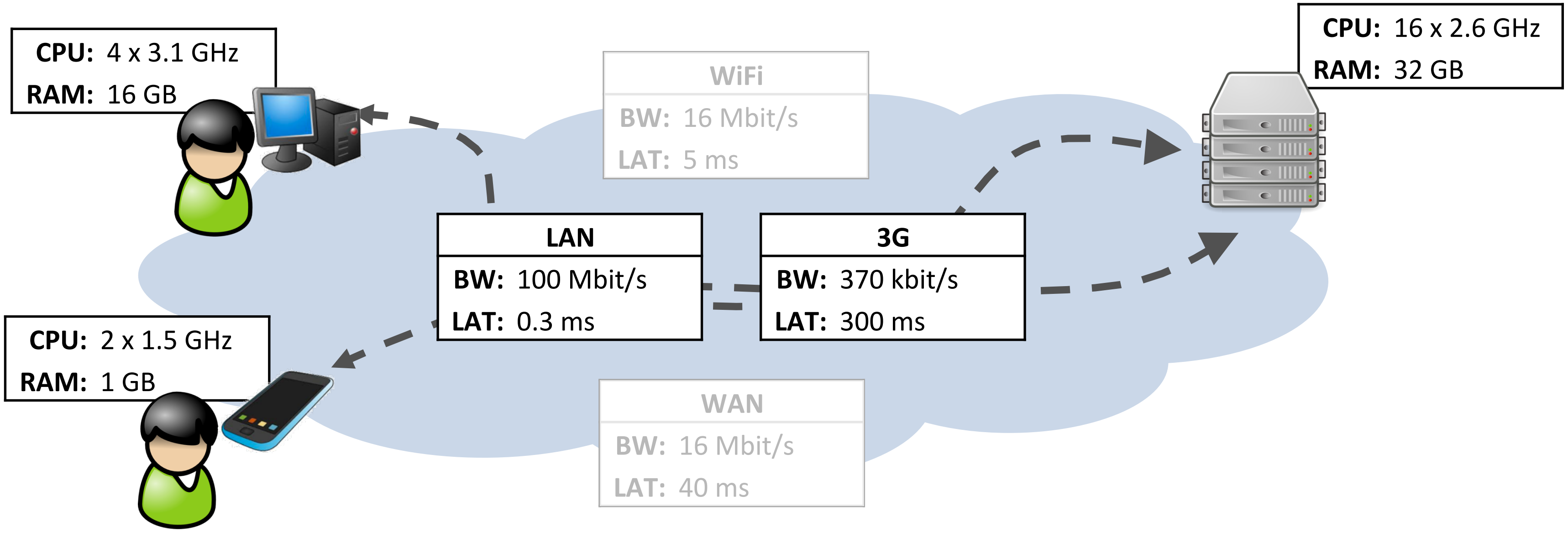
Methodology

Benchmarks

- Basic operations:
  - Arithmetic Operations: ADD and MULT
  - Logical Operations: MIN and ARGMIN
- Advanced operations:
  - MATRIX-MULT, SORT, more in work…



Evaluation Setup


CPU: 4 x 3.1 GHz
RAM: 16 GB

CPU: 2 x 1.5 GHz
RAM: 1 GB

CPU: 16 x 2.6 GHz
RAM: 32 GB

WiFi
BW: 16 Mbit/s
LAT: 5 ms

LAN
BW: 100 Mbit/s
LAT: 0.3 ms

3G
BW: 370 kbit/s
LAT: 300 ms

WAN
BW: 16 Mbit/s
LAT: 40 ms

Framework comparison

How comprehensive are STC frameworks?

	Fairplay	SeComLib	TASTY	mightbeevil	CBMC-GC
Approach	GC	HE	GC/HE	GC	GC
Type	Compiler	Library	Interpreter	Framework	Compiler
Language	SFDL	C++	TASTYL	Java	ANSI-C
Network	✓	✗	✓	✓	✓
Addition	✓	✓	✓	✓	✓
Multiplication	✗	✓	(✓)	✗	✓
Comparison	✓	✓	✓	✓	✓
Minimum	✗	✓	(✓)	(✓)	✗
Argmin	✗	✗	✗	✗	✗

Standard implementation of advanced operations using basic ops!

Performance evaluation

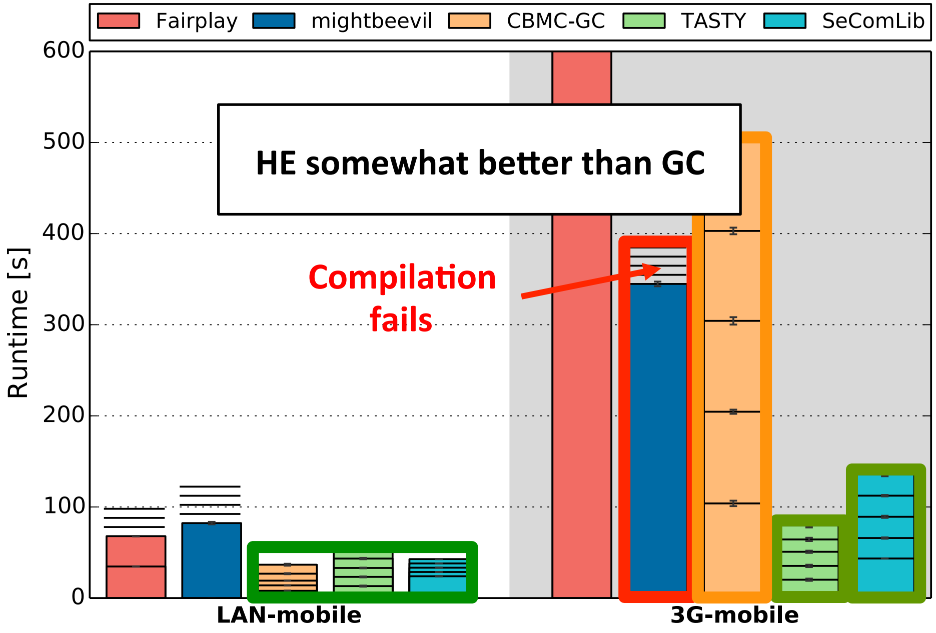
GC vs. HE – which approach to choose?

Arithmetic operations

- HE performs overall ok
- GC still manageable

Logical operations

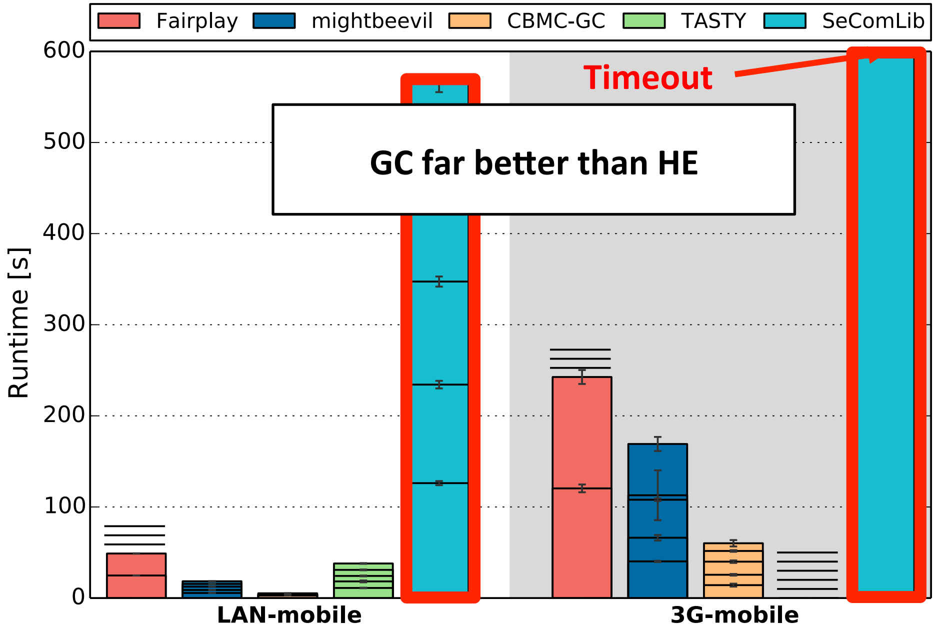
- GC very fast
- HE almost unusable



HE somewhat better than GC

Compilation fails

Multiplication



Timeout

GC far better than HE

Minimum

Performance evaluation

GC vs. HE – which approach to choose?

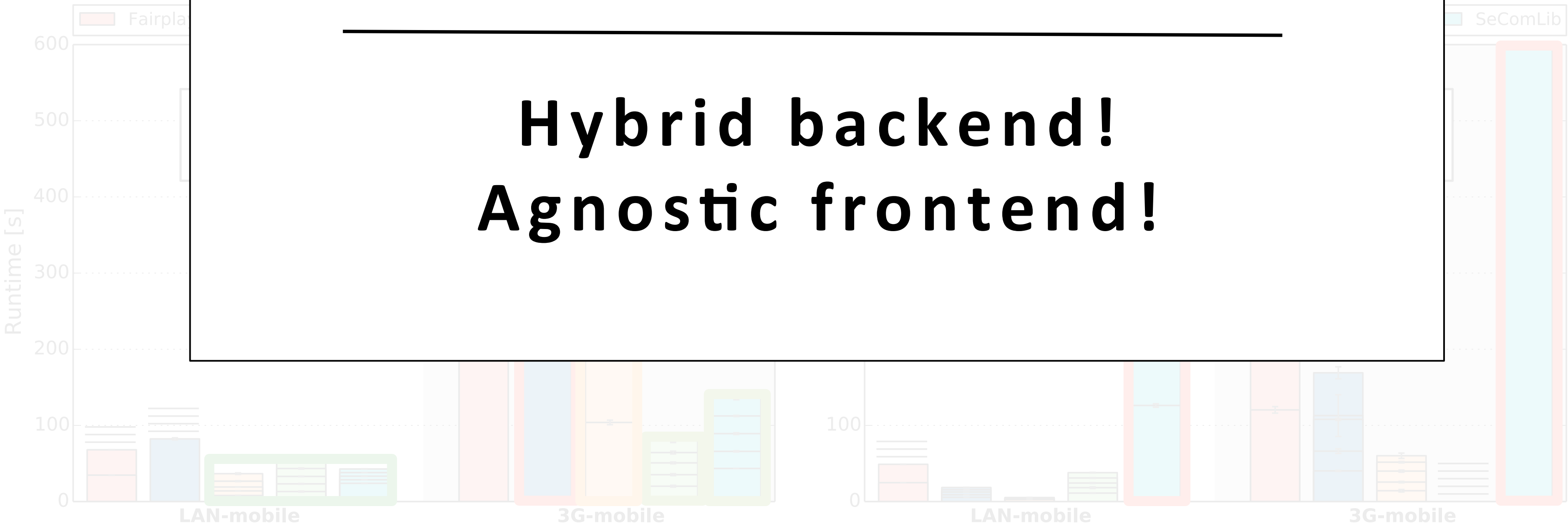
Arithmetic operations

- HE
- GC

Logical operations

GC or HE?

Hybrid backend!
Agnostic frontend!



Multiplication

Minimum

Performance evaluation

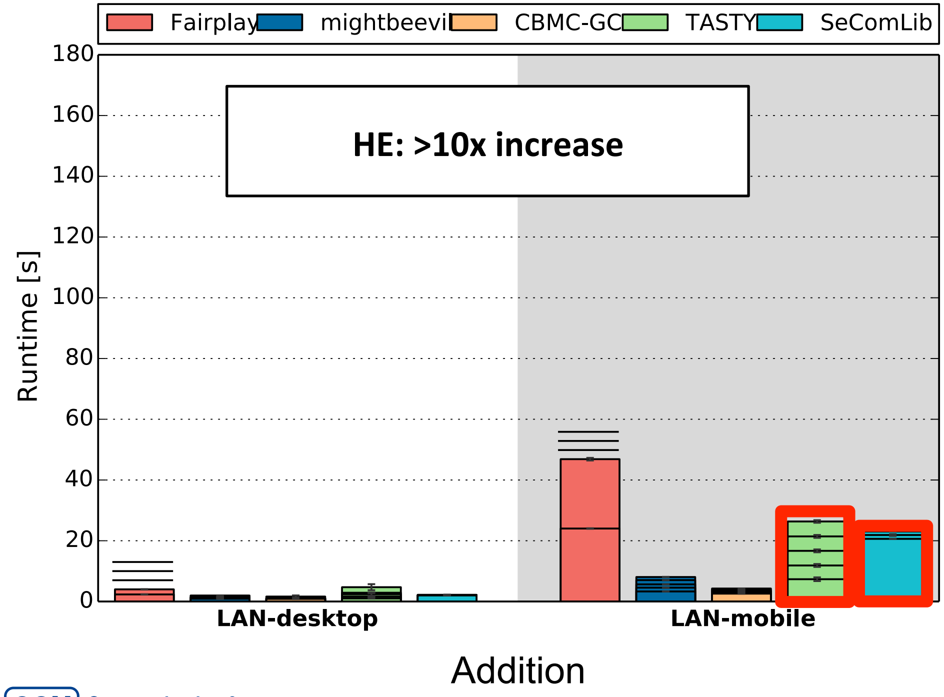
STC on mobile devices?

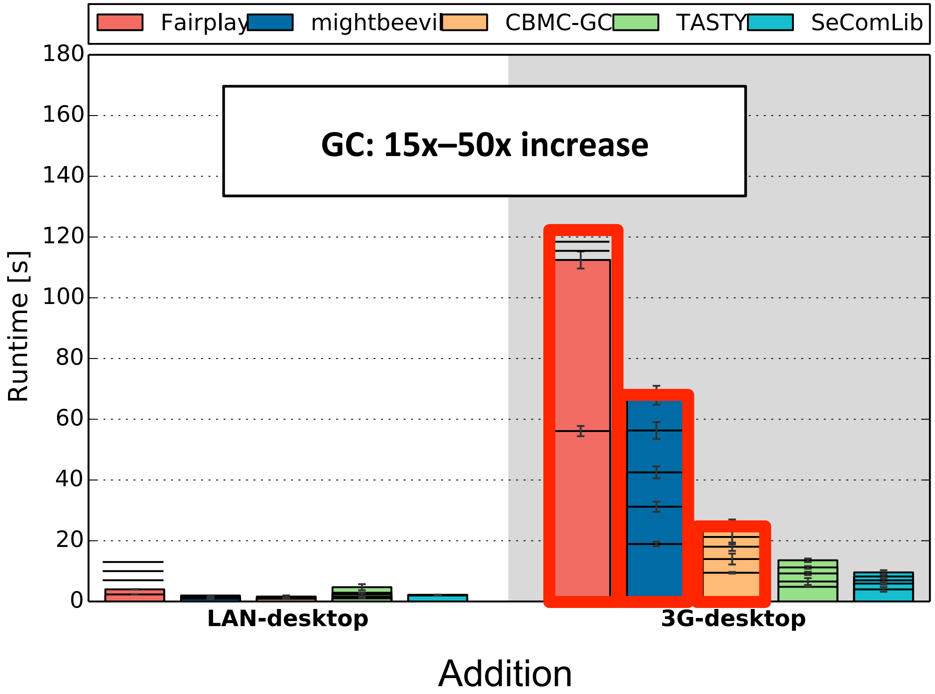
Processing

- Significant impact on HE
- Smaller but perceivable for GC

Bandwidth

- Tremendous impact on GC
- HE impacted mostly by latency





COM SYS Communication & Distributed Systems

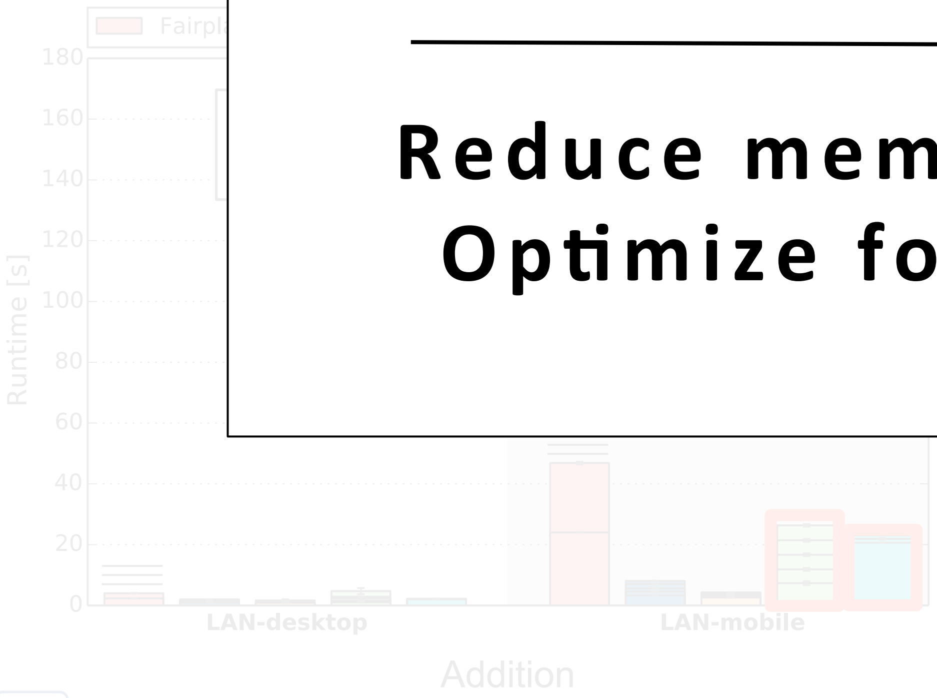
Performance evaluation

STC on mobile devices?

Processing

- Sig
- Sn

Bandwidth



STC on mobile devices?

Reduce memory footprint!

Optimize for bandwidth!

Performance evaluation

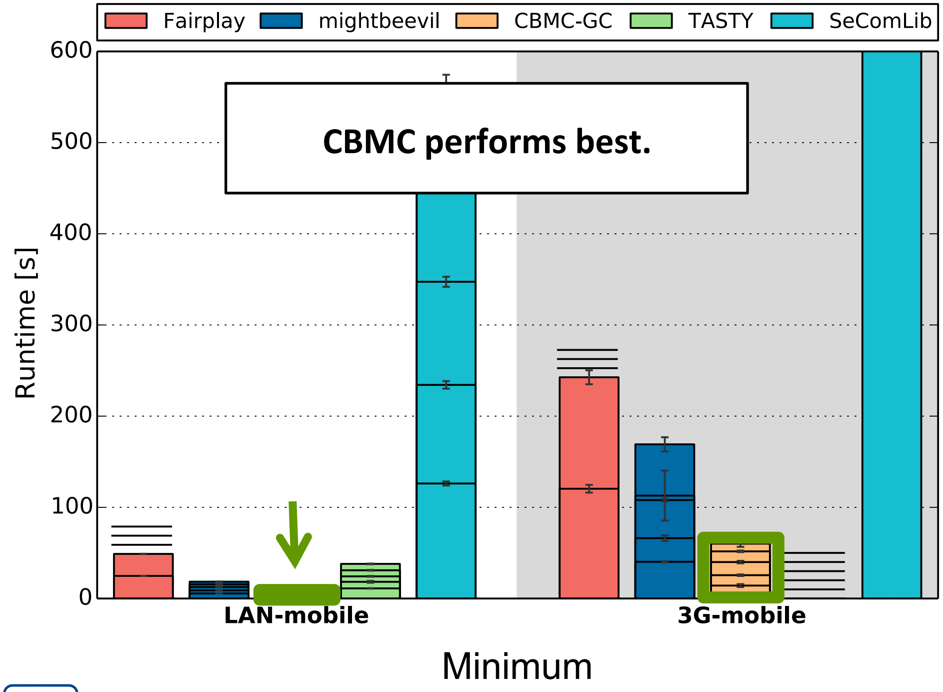
Is new functionality handled efficiently?

Yes!

- Example: Minimum in CBMC-GC

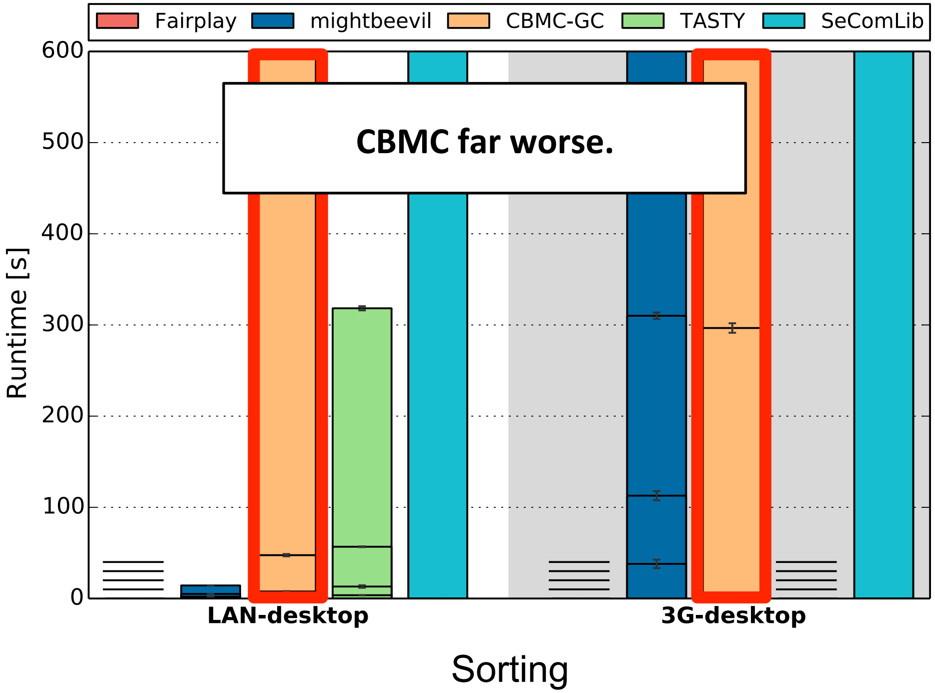
No!

- Example: Sorting in CBMC-GC



CBMC performs best.

Minimum



CBMC far worse.

Sorting

Performance evaluation

Is new functionality handled efficiently?

Yes!

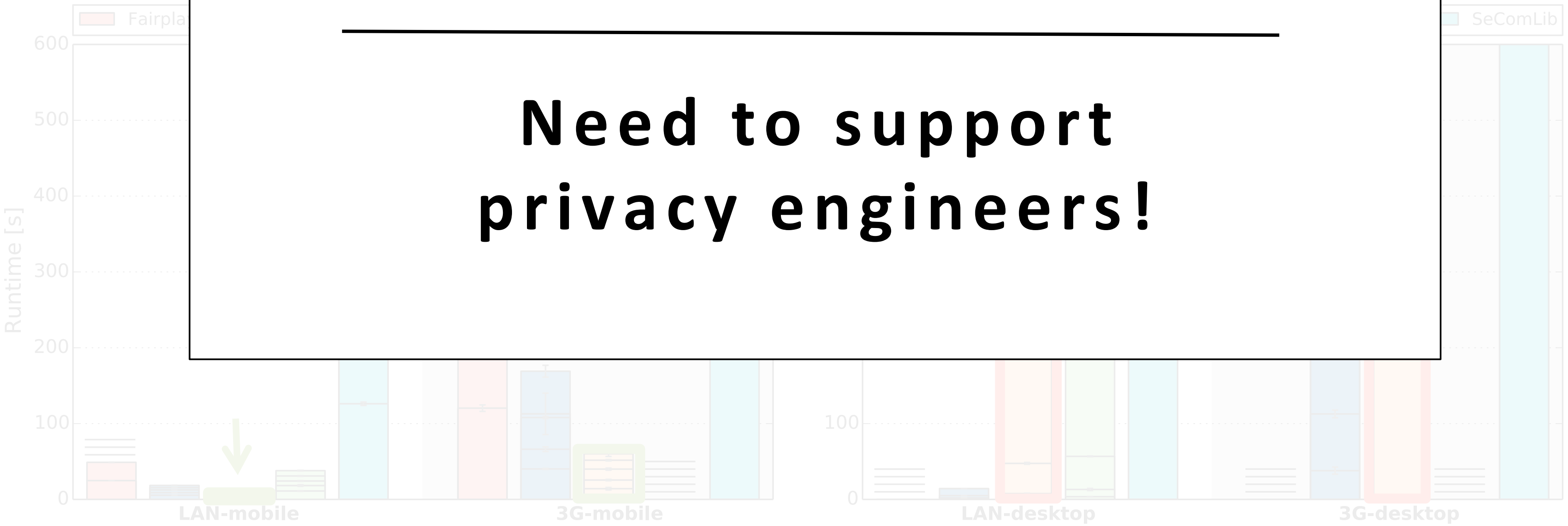
- Ex

No!

beevil

Handling new functionality?

Need to support privacy engineers!



Minimum

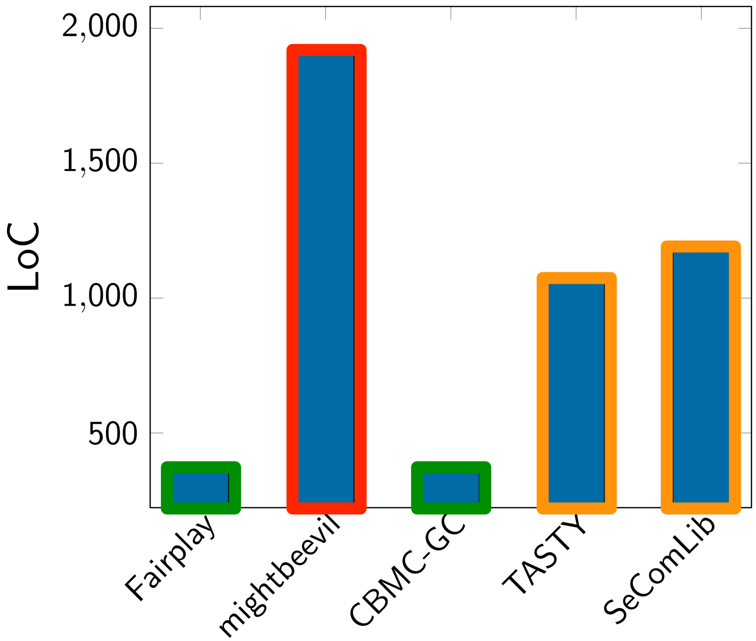
Sorting

Qualitative comparison

Which framework is the most usable?

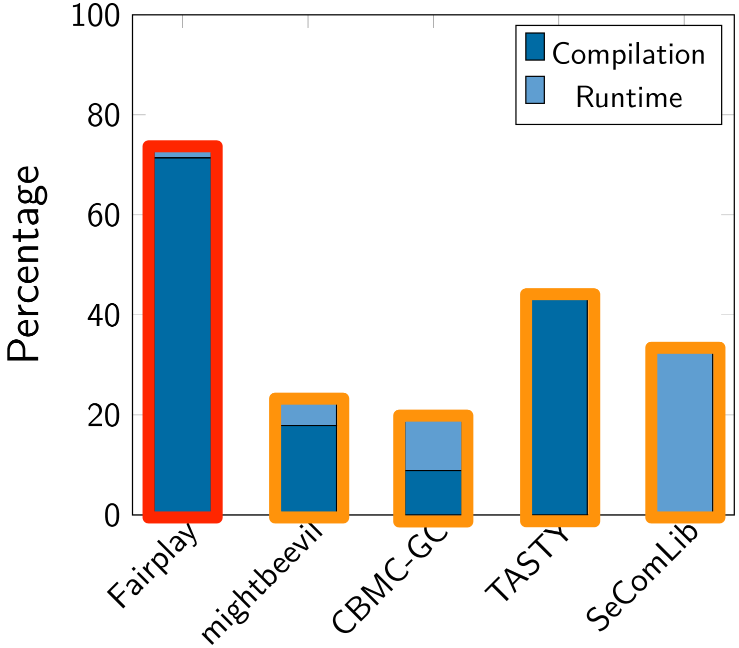
Lines of Code

- Compiler approach wins
- Library approach usable
- *mightbeevil* too low-level



Failures

- GC approaches limited by RAM
- HE limited by time-out



Conclusion and Directions

GCs more promising than HE

- Lower bounds on circuit sizes? (e.g., *Half-Gates, Eurocrypt’15)*
- Hybrid Approaches? (e.g., *ABY, NDSS’15)*
- Reducing memory of GC? (e.g., *Tiny-Garble, S&P’15)*

Mobile and interactive STCs

- Bandwidth-optimized STC?

Implementing / extending functionality

- How to guide the inexperienced STC developer?

Many open engineering issues

- Flexible STCs with inputs of unknown lengths?
- Language support for STC?

Further results, code and documentation

http://www.comsys.rwth-aachen.de/short/iwpe15/

✉ ziegeldorf@comsys.rwth-aachen.de

🌐 http://www.comsys.rwth-aachen.de/team/henrik-ziegeldorf/