

ABY - A Framework for Efficient Mixed-Protocol Secure Two-Party Computation



TECHNISCHE
UNIVERSITÄT
DARMSTADT

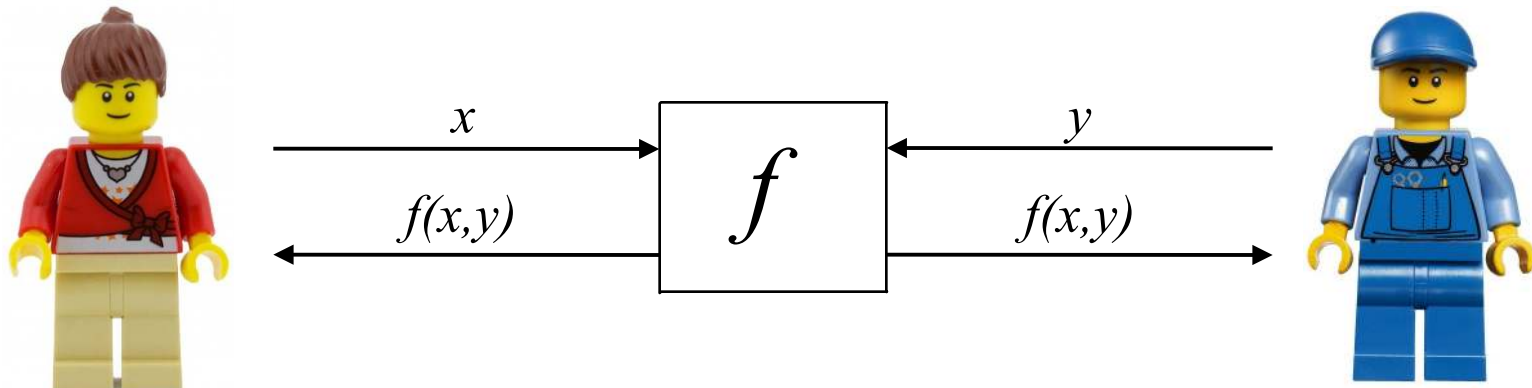
Thomas Schneider

joint work with
Daniel Demmler and Michael Zohner

published at NDSS 2015



Secure Two-Party Computation



Here we consider only **semi-honest** (passive) adversaries.

Privacy-Preserving Applications



Private Set Intersection [Meadows86], ...



Auctions [Naor-Pinkas-Sumner99], ...



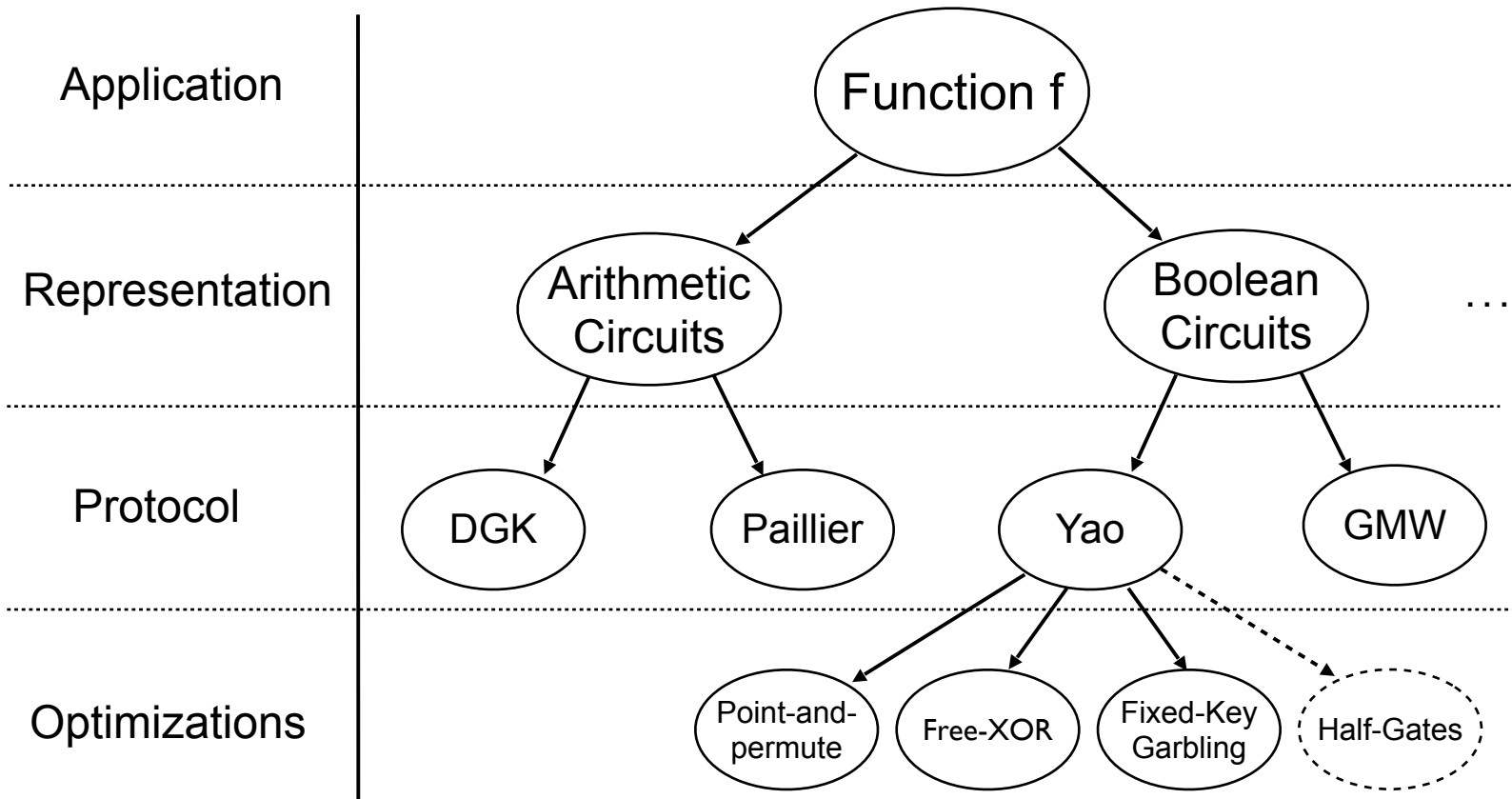
Biometric Identification [Erkin-Franz-Guajardo-Katzenbeisser-Langendijk-Toft09], ...



Machine Learning
[Bost-Popa-Tu-Goldwasser15], ...

etc.

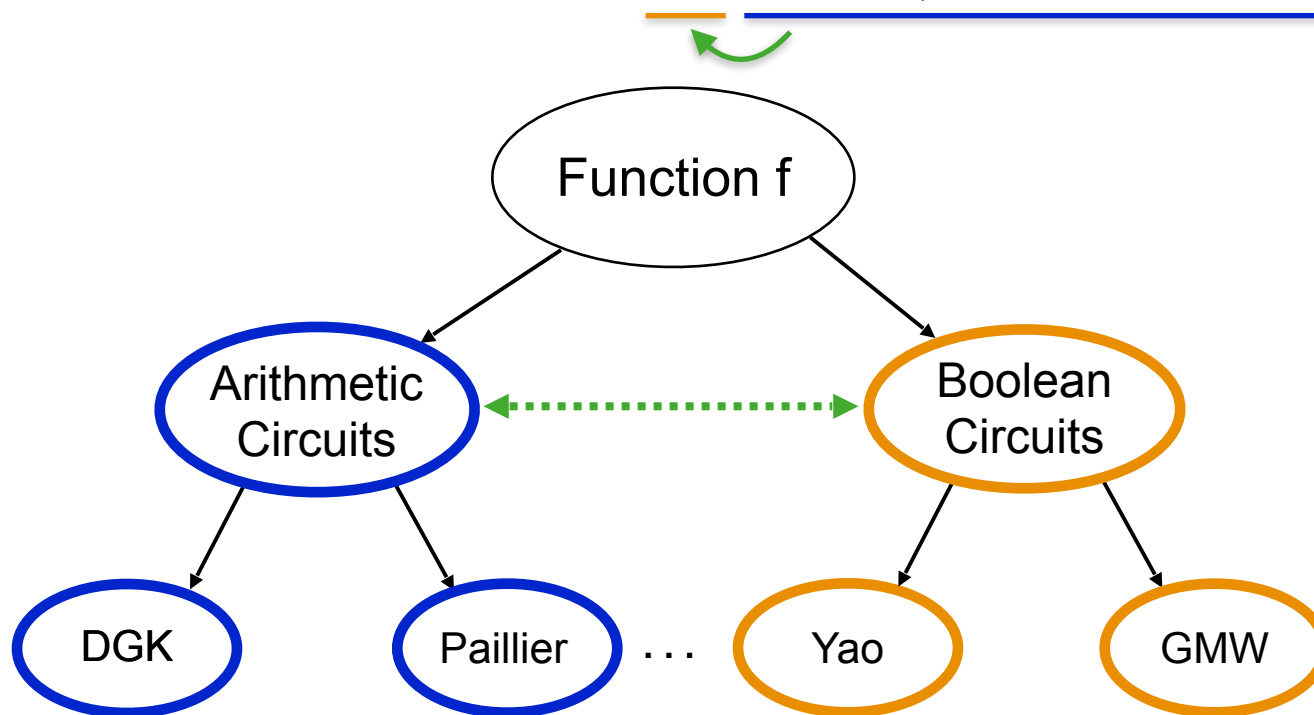
An Application Developer's Perspective



DGK: Damgård-Geisler-Krøigaard, GMW: Goldreich-Micali-Wigderson

Motivating Example for Mixed Protocols: Minimum Euclidean Distance

- Application: biometric matching (face-recognition, fingerprint, ...)
- Server holds database S_1, \dots, S_n , client holds query C
- Minimum Euclidean Distance: $f = \min(\sum_{i=1}^d (S_{1,i} - C_i)^2, \dots, \sum_{i=1}^d (S_{n,i} - C_i)^2)$



Mixed-Protocol Secure Computation

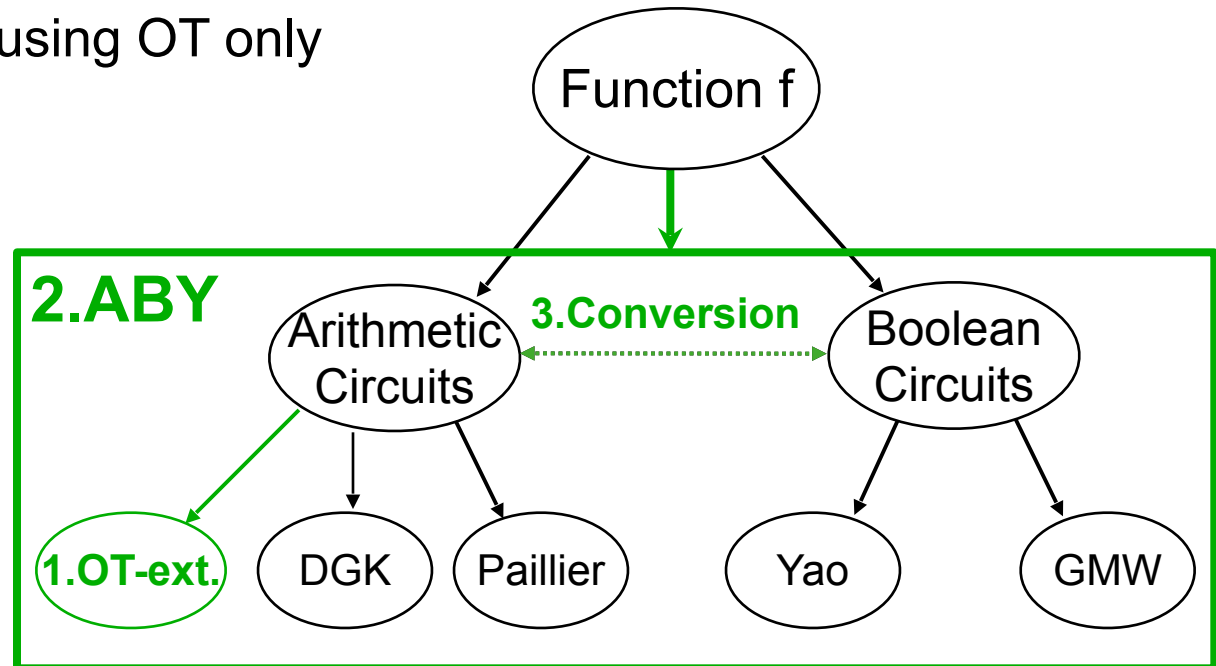
- Some functionalities are particularly expensive in one representation
 - Addition: Boolean circuit: $O(\ell)$ gates vs. Arithmetic circuit: 1 gate
 - Multiplication: Boolean circuit: $O(\ell^2)$ gates vs. Arithmetic circuit: 1 gate
- TASTY [Henecka-Kögl-Sadeghi-S-Wehrenberg10] combines Paillier (Arithmetic) and Yao (Boolean)



- **Multiplication** and **conversion** previously used expensive PK operations
 - Yao is often more efficient than Paillier [Kerschbaum-S-Schröpfer14]
 - **Our goal:** completely avoid PK operations & use Beaver multiplication triples to precompute symmetric crypto!

Roadmap / Our Contributions

- 1) OT-based multiplication is substantially faster than using PK crypto
- 2) Mixed-protocol framework ABY
- 3) Efficient conversions using OT only



1) Multiplication using OT [Gilboa99]

Schoolbook Multiplication $z = \mathbf{x} * \mathbf{y}$ with $\mathbf{x} = x_2x_1x_0$ and $\mathbf{y} = y_2y_1y_0$:

$$\mathbf{z} = \mathbf{x} * \mathbf{y}_0 + 2\mathbf{x} * \mathbf{y}_1 + 4\mathbf{x} * \mathbf{y}_2$$

$$\mathbf{r}_0 \in_{\mathbf{R}} \mathbf{Z}_{2^6}$$



$$\mathbf{r}_1 \in_{\mathbf{R}} \mathbf{Z}_{2^6}$$



$$\mathbf{r}_2 \in_{\mathbf{R}} \mathbf{Z}_{2^6}$$



$$[\mathbf{z}]_A = -\sum_{i=0}^2 \mathbf{r}_i$$

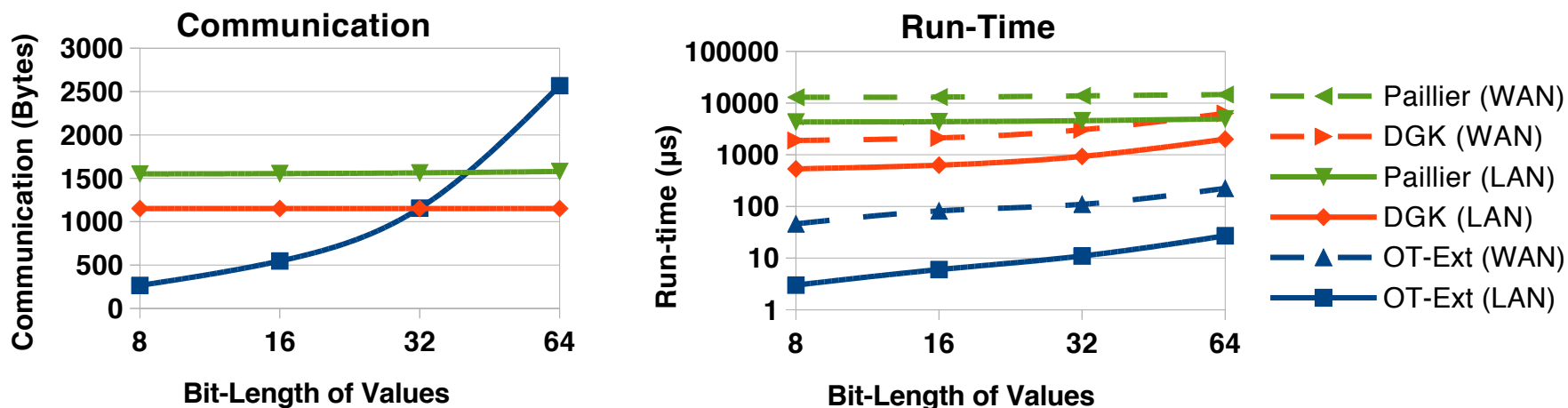
$$[\mathbf{z}]_B = \sum_{i=0}^2 \mathbf{s}_i = \sum_{i=0}^2 \mathbf{r}_i + \sum_{i=0}^2 2^i \mathbf{x} * \mathbf{y}_i$$

$$\mathbf{z} = [\mathbf{z}]_A + [\mathbf{z}]_B = \mathbf{x} * \mathbf{y}$$

1) Multiplication using OT Benchmarks

Instantiate OT efficiently with OT extension [Ishai-Kilian-Nissim-Petrank03, Asharov-Lindell-S-Zohner13]

Compare one amortized multiplication using Paillier, DGK, and OT extension



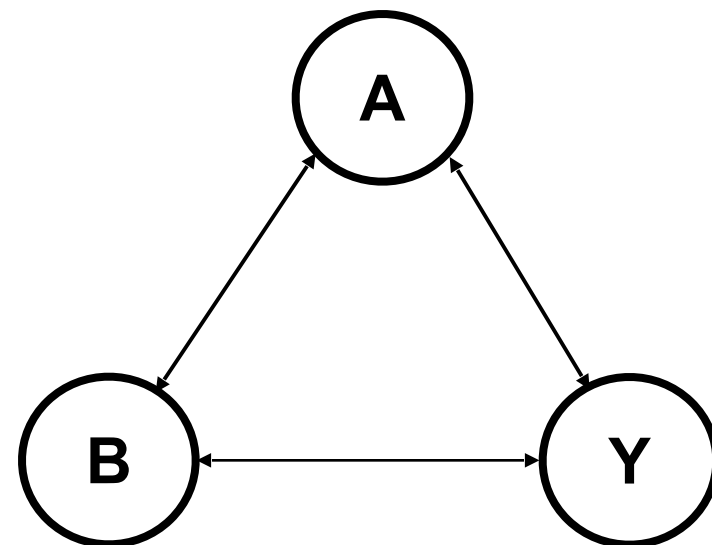
Communication and run-time for 1 multiplication in LAN and WAN for long-term security

2) The ABY framework

Combine:

- **A**rithmetic sharing
- **B**oolean sharing (GMW)
- **Y**ao's garbled circuits

Efficient conversions between schemes



Implement using state-of-the-art optimizations:

- batch pre-compute crypto operations
- use strong assumptions for maximum efficiency
- use fixed-key AES where possible (with AES-NI instruction set)

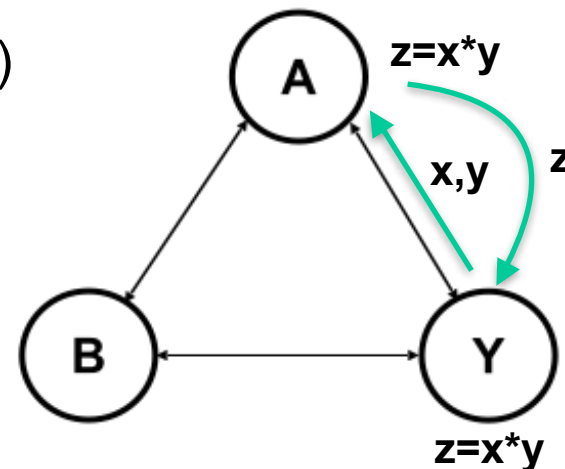
2) The ABY framework

- A** rithmetic sharing: $v = a + b \bmod 2^\ell$
- Free addition / cheap multiplication (1 msg)
 - Good for multiplication

- B** oolean sharing: $v = a \oplus b$
- Free XOR / 1 online msg per AND
 - Good for multiplexing (using 2 OTs)

- Y** ao's garbled circuits: A: k_0, k_1 ; B: k_v
- Free XOR / no interaction per AND
 - Good for comparisons

Benchmark primitive operations (+, *, >, =, ...)



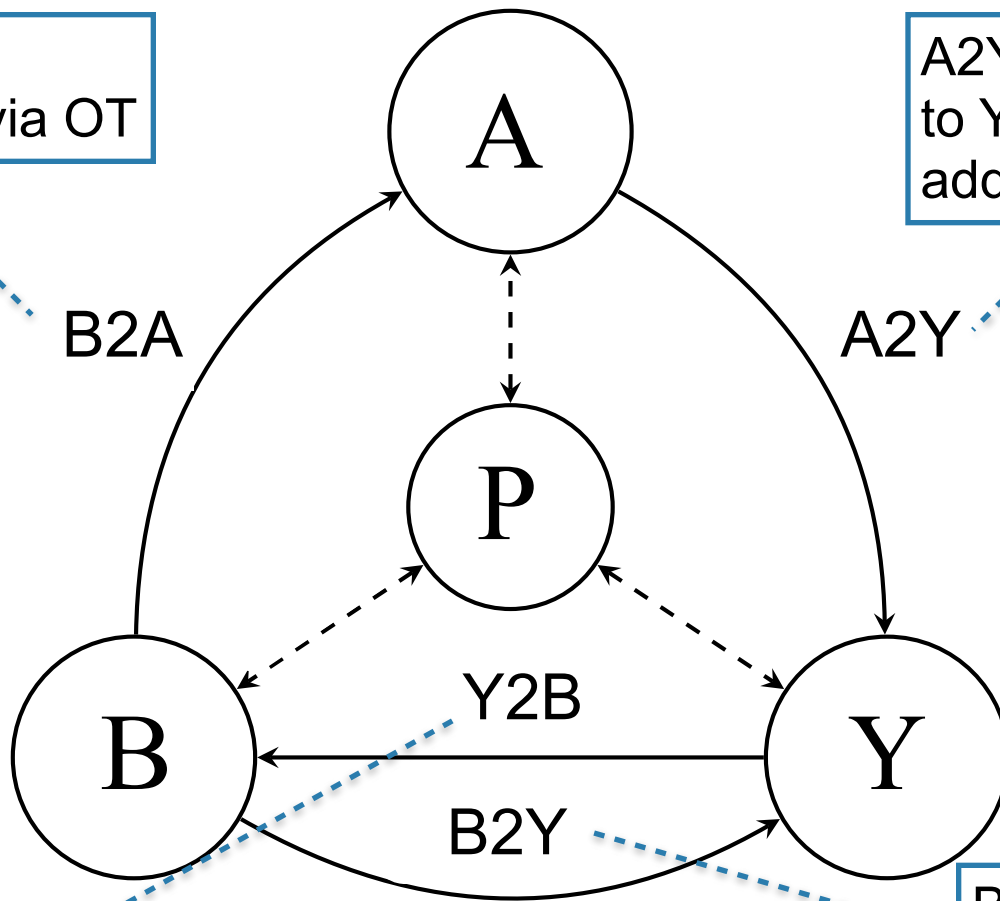
32-bit Multiplication (amortized)

Protocol	Yao
LAN [ms]	1.1
Comm. [KB]	100
#Msg	0

3) Efficient Conversions

B2A: similar to multiplication via OT

A2Y: convert shares to Yao and evaluate addition circuit with Yao



Y2B: for free - permutation bits of Yao sharing are Boolean sharing

B2Y: obviously send Yao key via OT

3) Efficient Conversions

Conversion of ℓ -bit values for symmetric security parameter κ

Conversion	Computation [#symm]	Communication [bits]	#Msg
Y2B	0	0	0
P2A, P2B, *2P	0	ℓ	1
P2Y _A	ℓ	$\ell\kappa$	1
B2A	6ℓ	$\ell\kappa + (\ell^2 + \ell)/2$	2
B2Y, P2Y _B	6ℓ	$2\ell\kappa$	2
A2Y	12ℓ	$6\ell\kappa$	2

} OT

Application 1: Minimum Euclidean Distance

Minimum Euclidean Distance: $\min(\sum_{i=1}^d (S_{1,i} - C_i)^2, \dots, \sum_{i=1}^d (S_{n,i} - C_i)^2)$

dist	min	LAN [s]	WAN [s]	Comm [MB]	#Msg
Y	Y	2.55	24.62	147.7	2
B	B	2.43	39.41	99.9	129
A	Y	0.19	3.42	5.0	8
A	B	0.21	26.41	4.6	101

Minimum Euclidean distance for $n = 512$ values of 32-bit length and $d = 4$.

LAN: Two standard PCs connected via Gigabit Ethernet.

WAN: Two Amazon EC2 c3.large instances - one located at US east cost and the other one in Japan.

Application 2: Private Set Intersection

PSI using Sort-Compare-Shuffle Circuit of [Huang-Evans-Katz12]



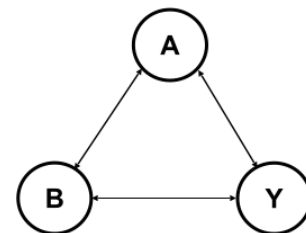
contains many multiplexers \Rightarrow benefits from Boolean sharing

Sort + Compare	Shuffle	LAN [s]	WAN [s]	Comm [MB]	#Msg
Y	Y	4.3	34.0	247	2
B	B	2.6	34.1	163	123
Y	B	3.3	30.0	182	27

PSI on 4096 elements of length 32 bit

Summary

ABY = framework for **mixed-protocol** secure computation



Abstract from details of underlying secure computation protocol



Use **only fast symmetric key crypto**



Code is available on **GitHub**: <http://encrypto.de/code/ABY>



Future Work

Use ABY framework for further **applications**



Automatically assign operations to sharing types



Extend mixed protocols to **stronger adversaries**



ABY - A Framework for Efficient Mixed-Protocol Secure Two-Party Computation



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Thanks!

Questions?

Contact: <http://encrypto.de>

Code: <http://encrypto.de/code/ABY>

