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

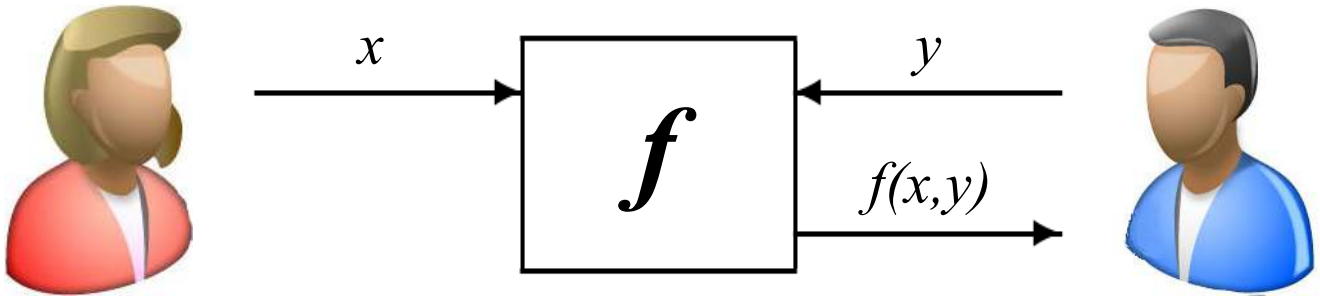
TECHNISCHE UNIVERSITÄT DARMSTADT

Michael Zohner (TU Darmstadt)

Joint work with
Daniel Demmler and Thomas Schneider

EC SPRIDE

Secure Two-Party Computation



This work: **semi-honest** adversaries

Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

Auctions [NPS99], ...

Private Set Intersection [PSZ14], ...

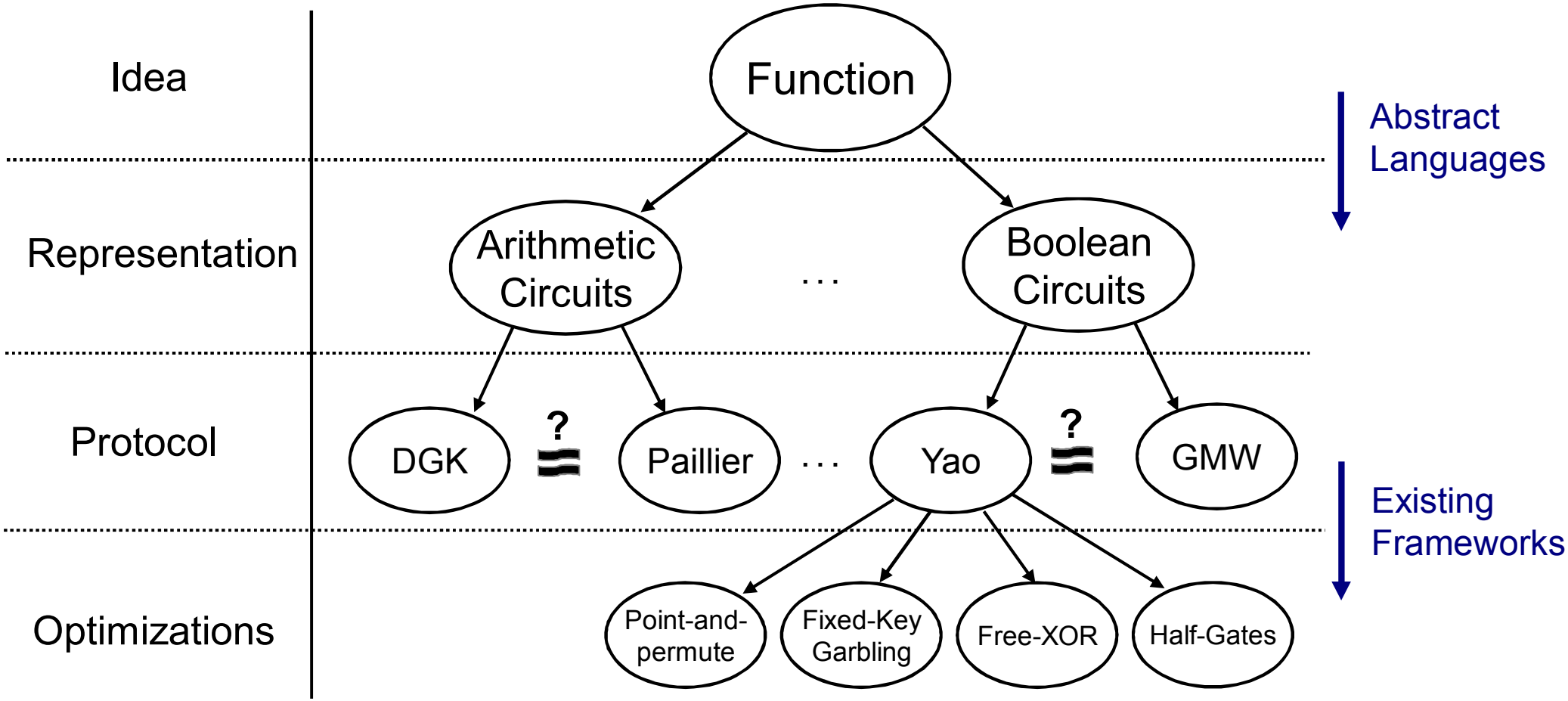
Machine Learning [BPTG15], ...

Biometric Identification [EFGKLT09], ...

- several cool applications from different fields

Protocol Development

TECHNISCHE UNIVERSITÄT DARMSTADT



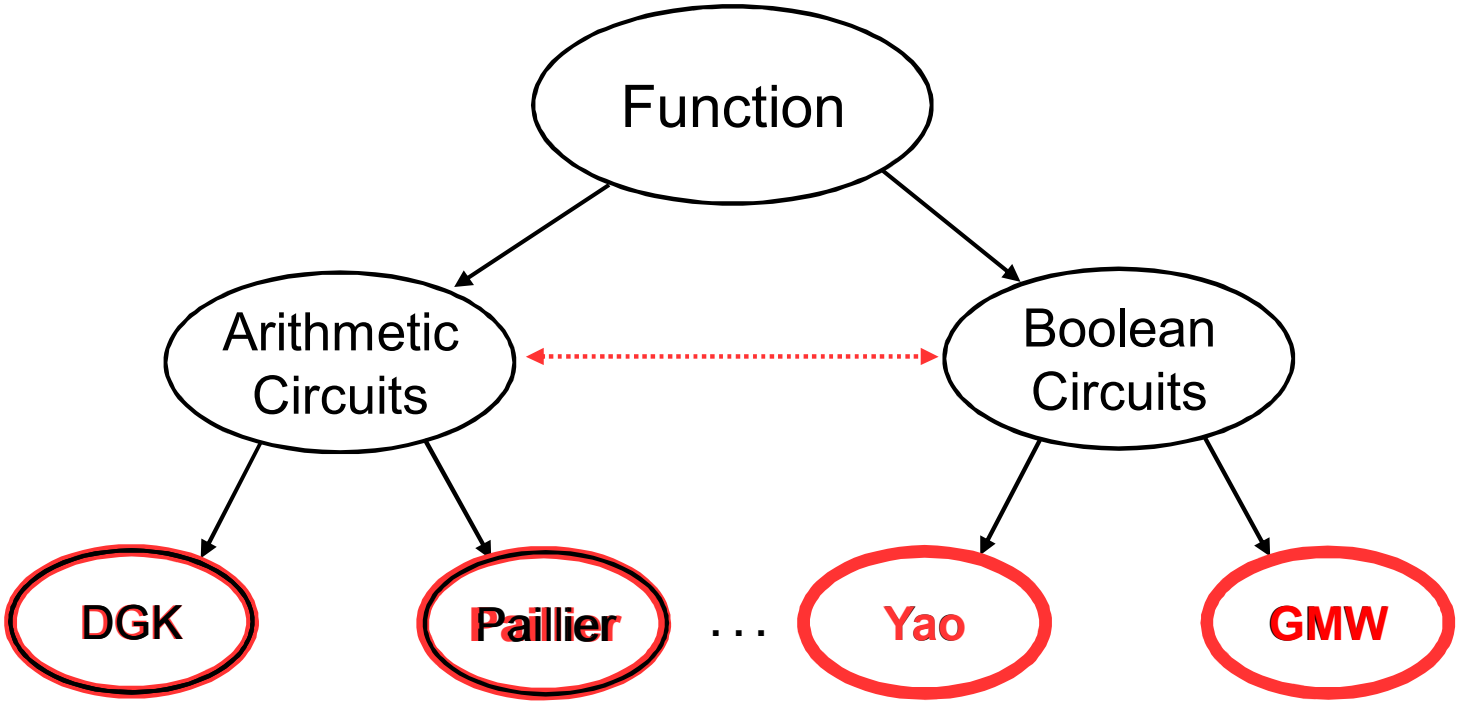
Secure computation is a vast area and protocol development is a tedious task

Example: Minimum Euclidean Distance

TECHNISCHE UNIVERSITÄT DARMSTADT

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

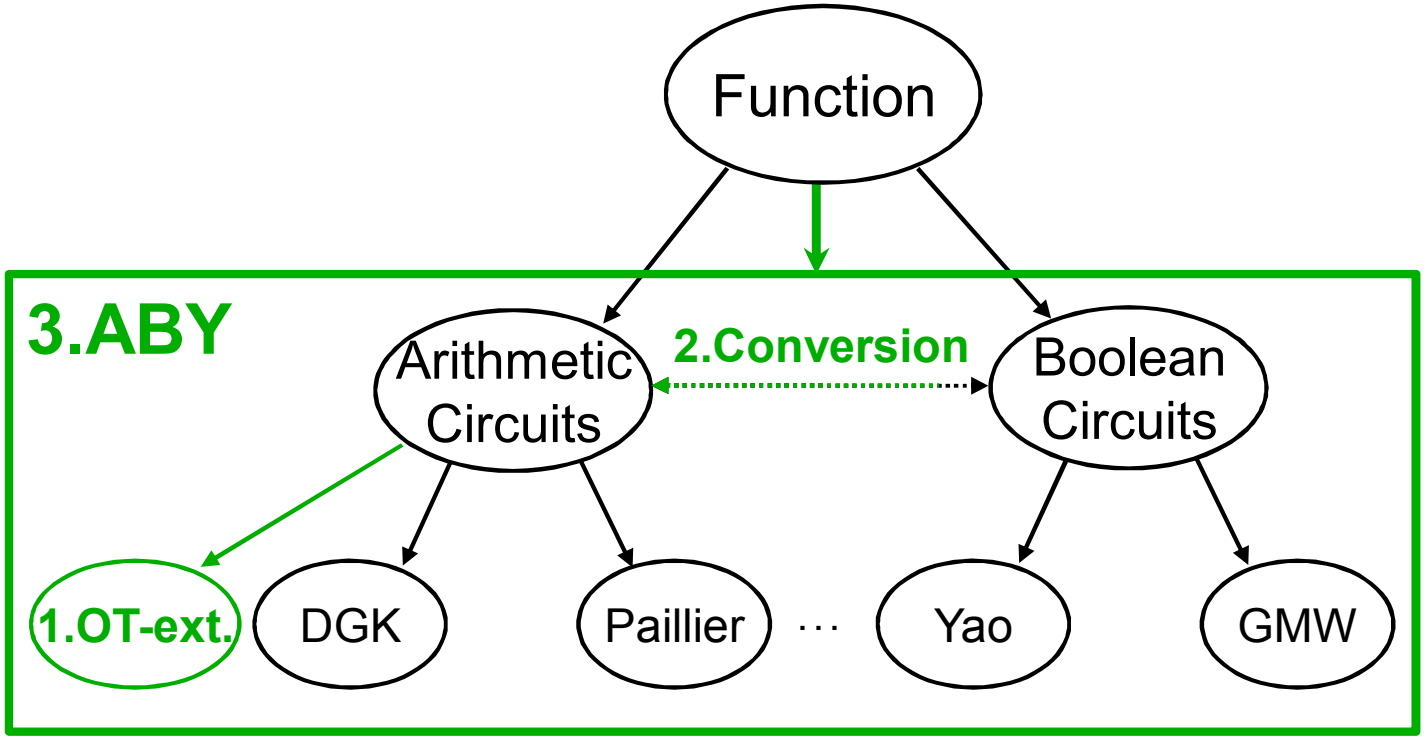
- Server holds database S, client holds query C
- Used in biometric matching (face-recognition, fingerprint, …)



EC SPRIDE

Our Contributions

TECHNISCHE UNIVERSITÄT DARMSTADT



1) Efficient multiplication using symmetric crypto

2) Efficient conversion

3) Mixed-protocol framework called ABY

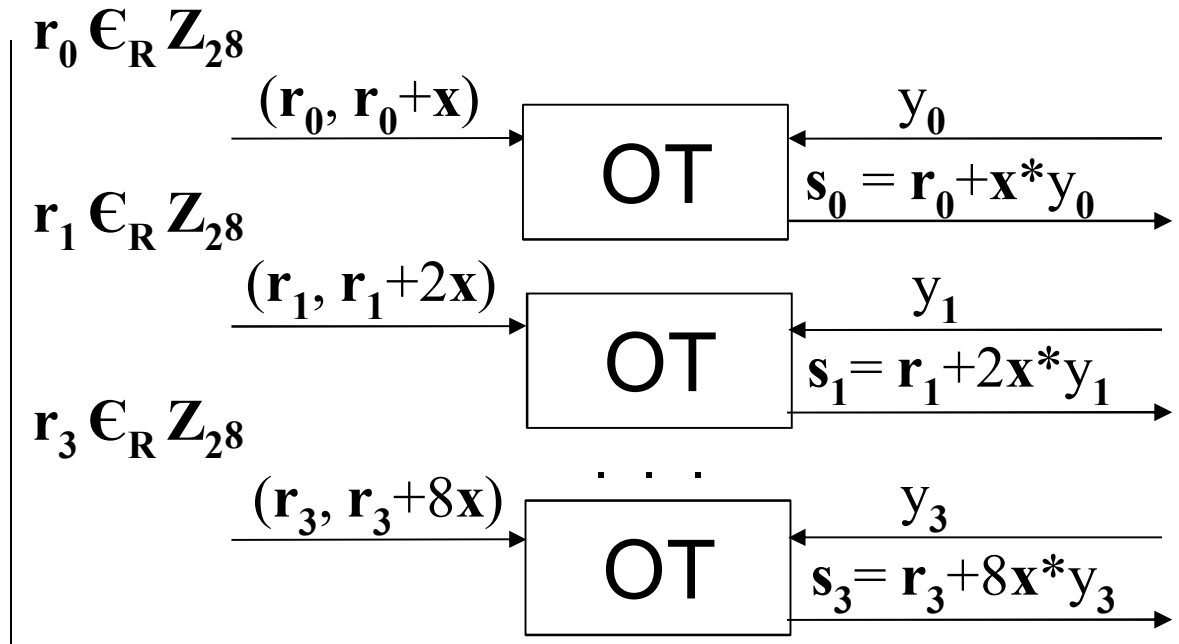
EC SPRIDE

Multiplication using OT [Gilboa99]

TECHNISCHE UNIVERSITÄT DARMSTADT

School Multiplication $z = x * y$ with $x=x_3x_2x_1x_0$ and $y=y_3y_2y_1y_0$

$$
\begin{array}{rr}
 & x_3x_2x_1x_0 * y_3y_2y_1y_0 \\
\hline
 & (x_3x_2x_1x_0) * y_0 \\
+ & (x_3x_2x_1x_0\,0) * y_1 \\
+ & (x_3x_2x_1x_0\,0\,0) * y_2 \\
+ & (x_3x_2x_1x_0\,0\,0\,0) * y_3 \\
\hline
 & z_7z_6z_5z_4z_3z_2z_1z_0
\end{array}
$$



$[z]_A=\sum^3_{i=0}(-r_i)$

$[z]_B=\sum^3_{i=0} s_i = \sum^3_{i=0} r_i + \sum^3_{i=0} 2^i x*y_i$

$z= [z]_A+[z]_B$

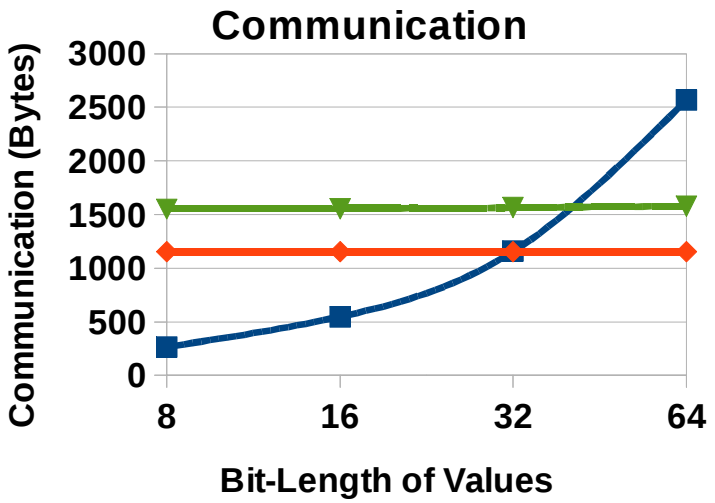
EC SPRIDE

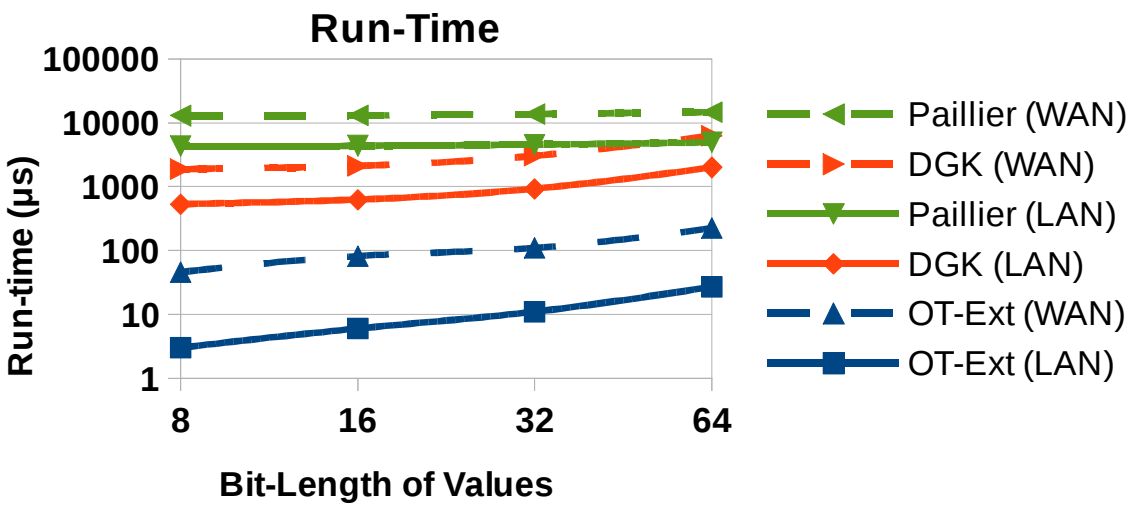
Multiplication using OT Results

TECHNISCHE UNIVERSITÄT DARMSTADT

Use a multiplication protocol that is based on **OT extension**

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

EC SPRIDE

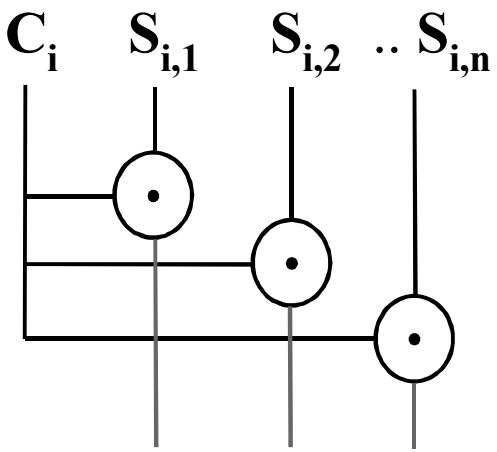
Scalar Multiplication (1)

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

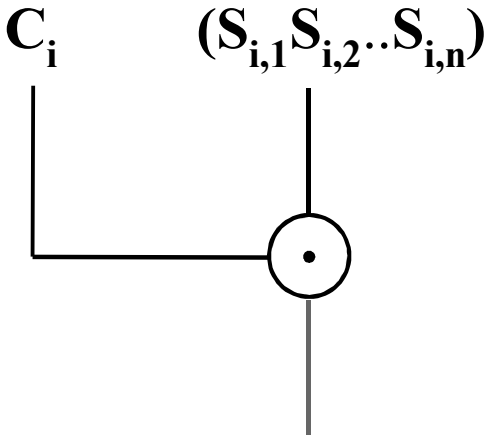
- Rewrite: $\sum^{d}_{i=1}(S_{i,1} - C_i)^2 = \sum^{d}_{i=1} S^2_{i,1} - \sum^{d}_{i=1} 2C_iS_{i,1} + \sum^{d}_{i=1} C^2_i$

Assume values of bit-length l

Naive: $2l*n*d$ OTs

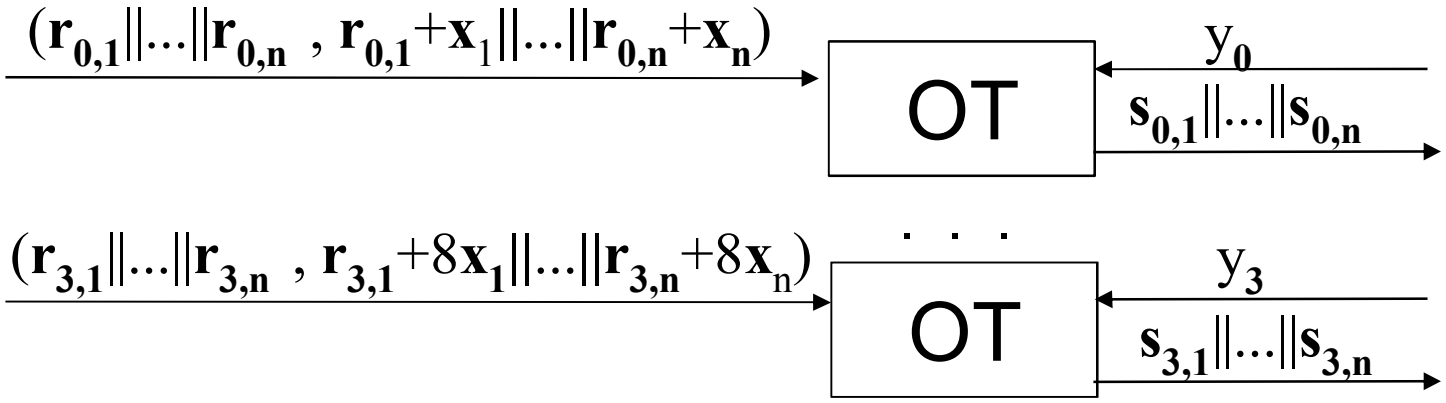


Scalar: $2l*d$ OTs



Scalar Multiplication (2)

Scalar Multiplication $(z_1,\ldots,z_n) = (x_1,..,x_n) * y$ with $y=y_3y_2y_1y_0$



$[z_1,\ldots,z_n]_A=\sum^3_{i=0}(-r_{i,1}),\ \ldots,\ \sum^3_{i=0}(-r_{i,n})$

$[z_1,\ldots,z_n]_B=\sum^3_{i=0}s_{i,1},\ \ldots,\ \sum^3_{i=0}s_{i,n}$

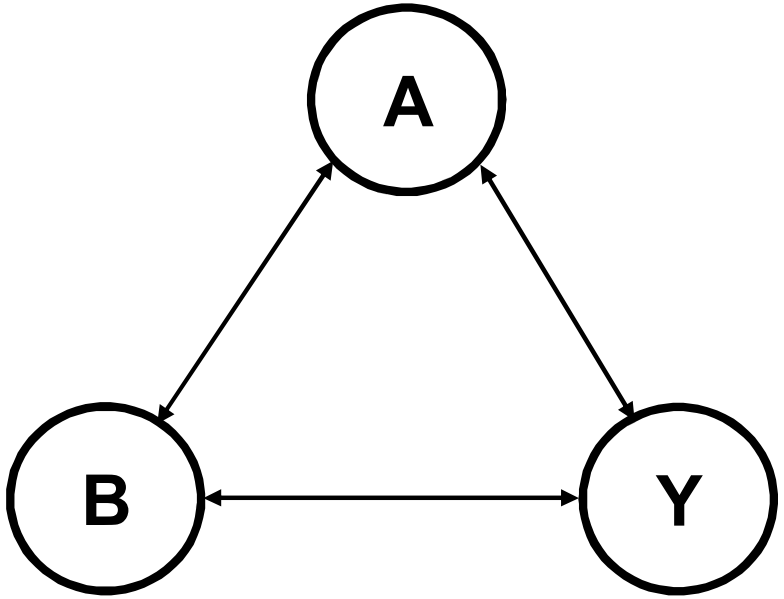
$$z_1,\ldots,z_n = [z_1,\ldots,z_n]_A+[z_1,\ldots,z_n]_B$$

The ABY framework

TECHNISCHE UNIVERSITÄT DARMSTADT

Combine:

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



Efficient conversions between schemes

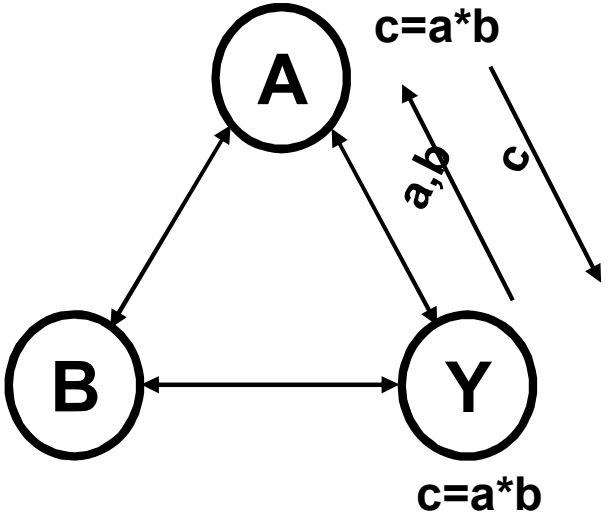
Use efficient techniques:

- batch pre-compute crypto
- use fixed-key AES where possible
- use sub-protocols with recent optimizations

Benchmarking Secure Computation Schemes

TECHNISCHE UNIVERSITÄT DARMSTADT

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



Arithmetic sharing:
- Free addition / cheap multiplication
- Good for multiplication

Boolean sharing:
- Free XOR / one interaction per AND
- Good for multiplexing

Yao's garbled circuits:
- Free XOR / no interaction per AND
- Good for comparison

Multiplication *(amort.)*	
Protocol	***Yao***
LAN [ms]	**1.1**
Comm. [KB]	**100**
Rounds	***0***

EC SPRIDE

Example: Minimum Euclidean Distance

TECHNISCHE UNIVERSITÄT DARMSTADT

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

```
01.  share* min_euclid_dist(share*** S, share** C, uint32_t dbsize, uint32_t dim,
     share** Ssqr, Circuit* dist, Circuit* min
02.      share **distance, *temp, *mindist;
03.  ...
04.      for (uint32_t i=0, j; i < dbsize; i++) {
05.          distance[i] = dist->PutMULGate(S[i][0], C[0]);
06.          for (j=1; j < dim; j++) {
07.              temp = dist->PutMULGate(S[
08.              distance[i] = dist->PutADD
09.          }
10.          temp = min->PutADDGate(Ssqr[i]
11.          distance[i] = min->PutSUBGate(
12.      }
13.  ...
14.      return min->PutMinGate(distance, dbsize);
15.  }
```

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

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

EC SPRIDE

Future Work

TECHNISCHE UNIVERSITÄT DARMSTADT

Generalize and implement **special purpose** operations

Automatically assign operations to protocols [KSS14]

Extension to **malicious adversaries**

EC SPRIDE

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

TECHNISCHE UNIVERSITÄT DARMSTADT

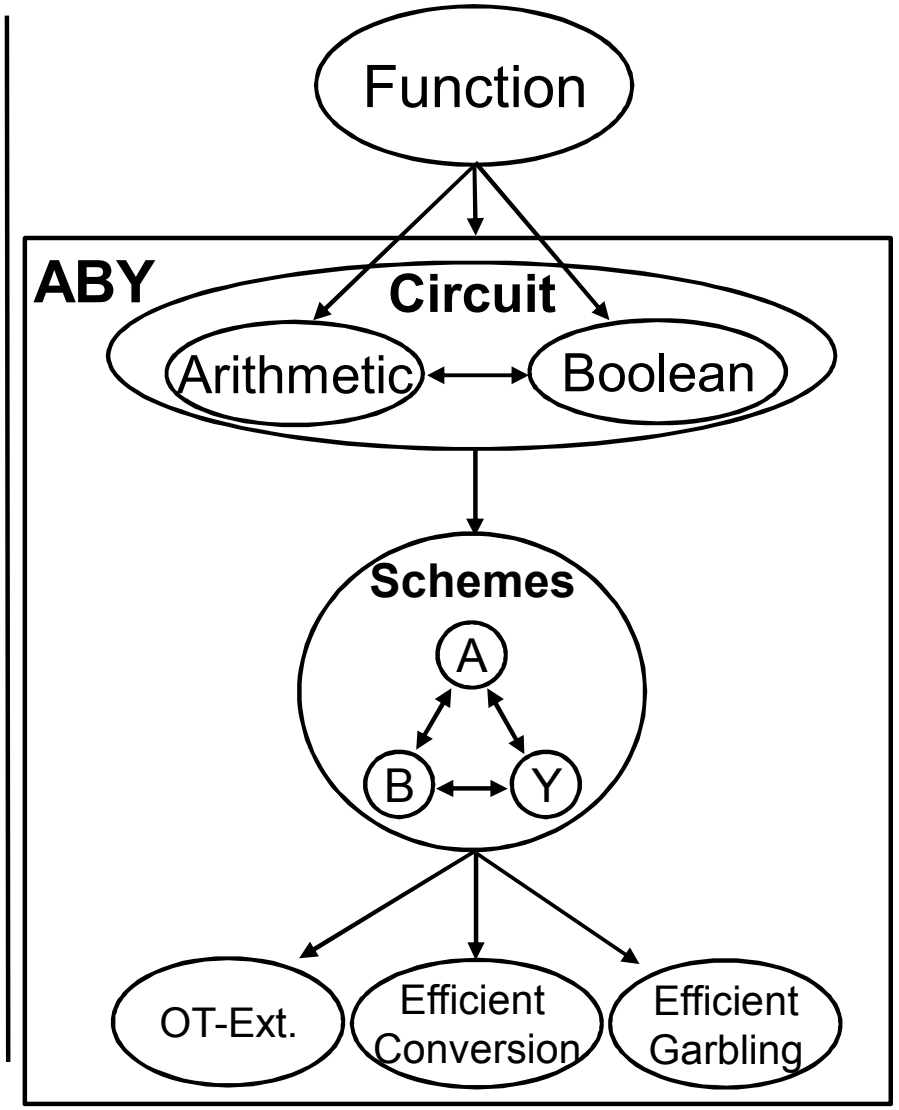
Questions?

Contact:
Code: http://encrypto.de/code/ABY

EC SPRIDE

ABY Development

TECHNISCHE UNIVERSITÄT DARMSTADT



Idea

Representation

Protocol

Optimizations

Extensibility

Special purpose circuits

Optimize existing / implement new schemes

Implement further optimizations

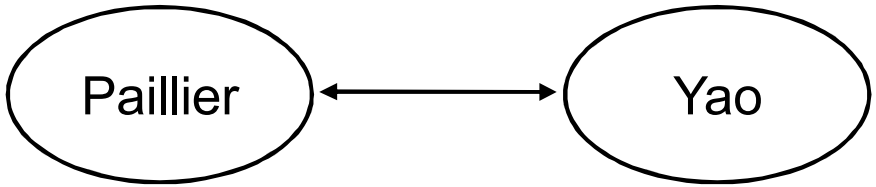
EC SPRIDE

Mixed-Protocols

TECHNISCHE UNIVERSITÄT DARMSTADT

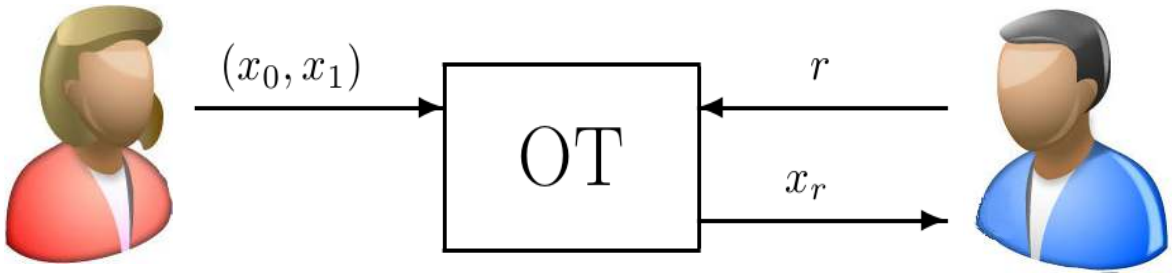
- Some functionalities have a more efficient circuit representation
  - Multiplication in Boolean circuits: $O(n^2)$
  - Comparison in Arithmetic circuits: $O(n)$ multiplications of q-bit values

- TASTY [HKSSW10] combines Paillier (Arithmetic) and Yao (Boolean)



- **Multiplication** and **conversion** requires public-key operation
  - For long-term security, Yao-only is often most efficient [KSS14]

OT Extension



Input: Alice holds two strings (x_0, x_1), Bob holds a choice bit r

Output: Alice learns nothing, Bob only learns x_r

Traditionally, OT requires public-key crypto

OT extension allows extending few “real” OTs to arbitrary many OTs using symmetric key cryptography only

References

[NPS99]: Moni Naor, Benny Pinkas, Reuban Sumner: Privacy preserving auctions and mechanism design. EC 1999: 129-139.

[BPTG15] Raphael Bost, Raluca Ada Popa, Stephen Tu, Shafi Goldwasser: Machine Learning Classification over Encrypted Data. NDSS 2015.

[EFGKLT09]: Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, Tomas Toft: Privacy-Preserving Face Recognition. Privacy Enhancing Technologies 2009: 235-253.

[KSS14]: Florian Kerschbaum, Thomas Schneider, Axel Schröpfer: Automatic Protocol Selection in Secure Two-Party Computations. ACNS 2014: 566-584.

DGK: Ivan Damgård, Martin Geisler, Mikkel Krøigaard: A correction to 'efficient and secure comparison for on-line auctions'. IJACT 1(4): 323-324 (2009).

Paillier: Pascal Paillier: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. EUROCRYPT 1999: 223-238,

GMW: Oded Goldreich, Silvio Micali, Avi Wigderson: How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. STOC 1987: 218-229.

Yao: Andrew Chi-Chih Yao: Protocols for Secure Computations (Extended Abstract). FOCS 1982: 160-164.

References

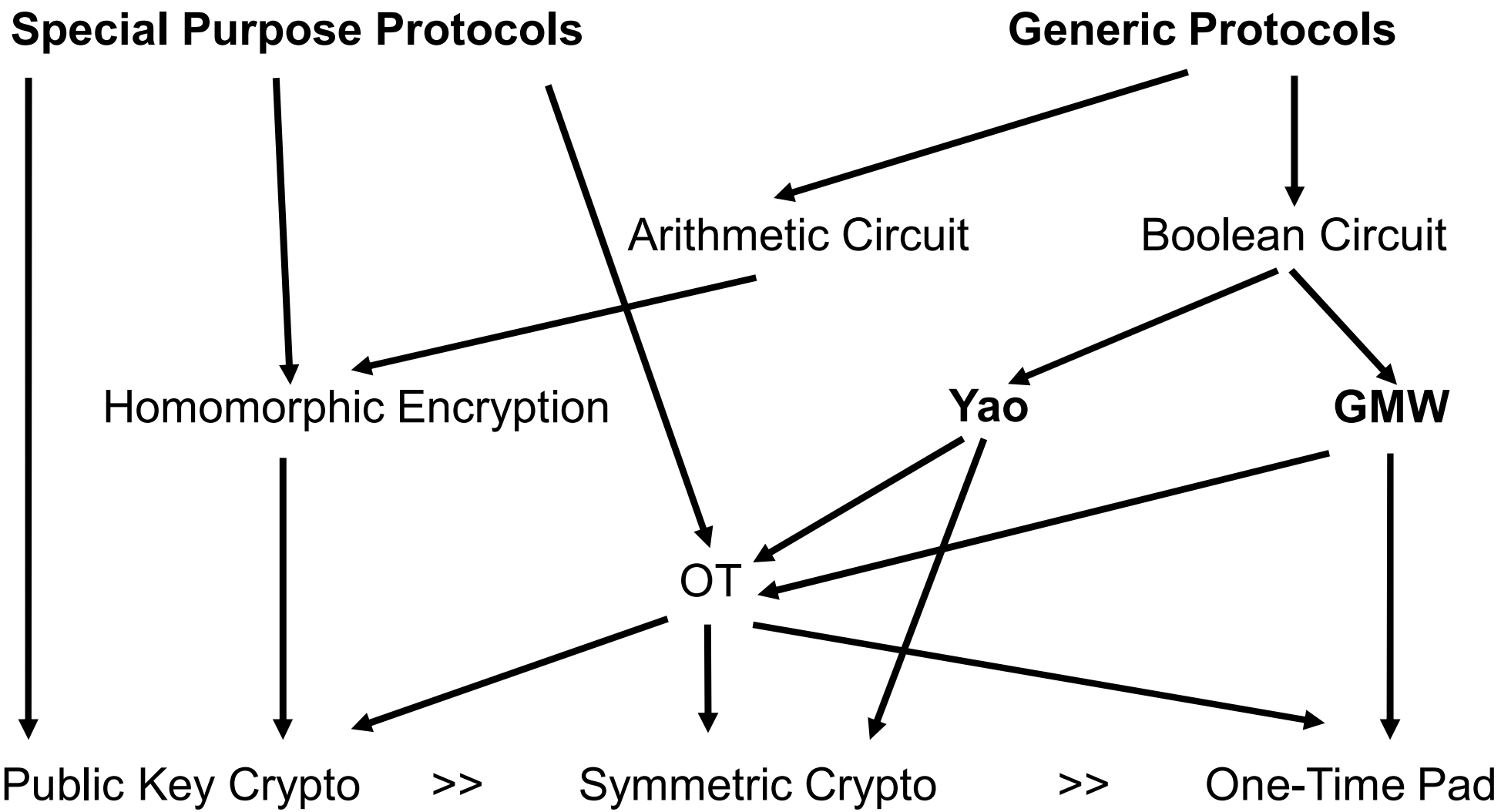
[BG11]: Marina Blanton, Paolo Gasti: Secure and Efficient Protocols for Iris and Fingerprint Identification. ESORICS 2011: 190-209.

[HKSSW10]: Wilko Henecka, Stefan Kögl, Ahmad-Reza Sadeghi, Thomas Schneider, Immo Wehrenberg: TASTY: tool for automating secure two-party computations. ACM Conference on Computer and Communications Security 2010: 451-462.

[Gilboa99]: Niv Gilboa: Two Party RSA Key Generation. CRYPTO 1999: 116-129.

Protocol Overview

TECHNISCHE UNIVERSITÄT DARMSTADT



EC SPRIDE

Example: Minimum Euclidean Distance

TECHNISCHE UNIVERSITÄT DARMSTADT

Minimum Euclidean Distance: $\min(\sum^{d}_{i=1}(\mathbf{S}_{i,1} - \mathbf{C}_{\mathrm{i}})^2, \ldots, \sum^{d}_{i=1}(\mathbf{S}_{i,n} - \mathbf{C}_{i})^2)$

- Server holds database $\mathbf{S}$, client holds query $\mathbf{C}$
- Used in biometric matching (face-recognition, fingerprint, …)

1) Evaluate in Arithmetic circuits using Paillier [EFGJKT09] or DGK [BG11]
   - Comparison is costly

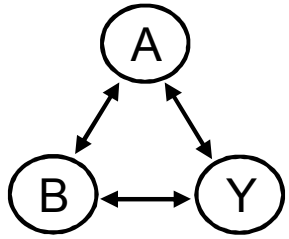
2) Multiplication in Arithmetic; Comparison in Boolean circuits [HKSSW10]
   - Costly conversion/multiplication: expensive public-key crypto

3) Evaluate everyting in Boolean circuits using Yao [KSS14]

Take Away Message

TECHNISCHE UNIVERSITÄT DARMSTADT

Developed a **mixed-protocol** secure computation framework



Abstract from underlying secure computation protocol

Use only **fast symmetric key crypto**

Code is available at **GitHub**: