Post Quantum Secure OT Extension

Xiaoye Wu
Tim Fischer

Supervised by Gowri R Chandran

TECHNISCHE UNIVERSITÄT DARMSTADT

CS

CYSEC Cybersecurity TU Darmstadt

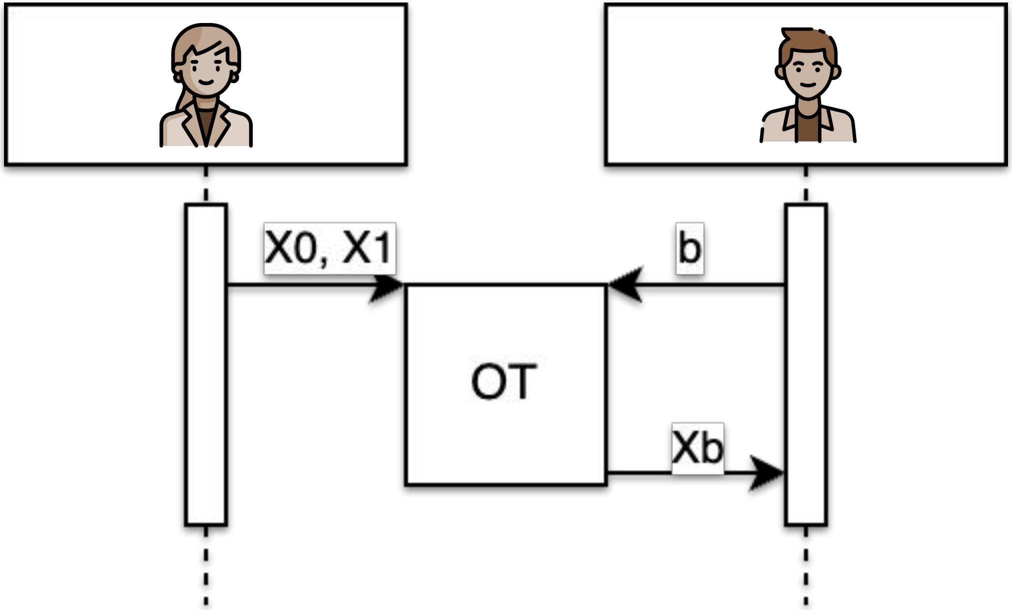
ENCRYPTO CRYPTOGRAPHY AND PRIVACY ENGINEERING

Table of Contents

What is an OT? → Related Work → The Goal

1. Introduction

What is an Oblivious Transfer ?



What is an OT? → Related Work → The Goal

Related Work

TECHNISCHE UNIVERSITÄT DARMSTADT

OT

- Base OT
- Extended OT

Post Quantum Secure OT

- Base OT
- Extended OT

1-out-of-2 OT
{MI81} [5]

OT Extension
{ALSZ17} [1]

Post Quantum Secure OT
{BDK+20} [2]

Extended Post Quantum Secure OT

The Goal

TECHNISCHE UNIVERSITÄT DARMSTADT

Implement

- Only the extension
- Semi-honest adversaries

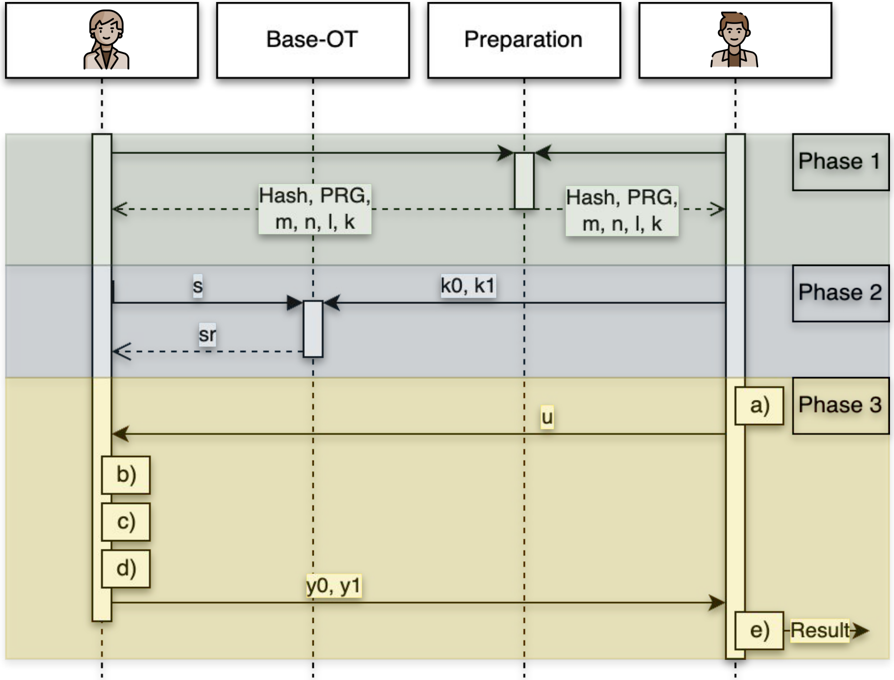
Benchmark

- Compare to base OT

Overview → Phase 1 → Phase 2 → Phase 3

2. The Protocol in a Nutshell

Overview



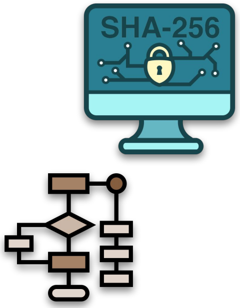
Base-OT
Preparation
Phase 1
Hash, PRG,
m, n, l, k
Hash, PRG,
m, n, l, k
Phase 2
s
k0, k1
sr
Phase 3
a)
u
b)
c)
d)
y0, y1
e)
Result

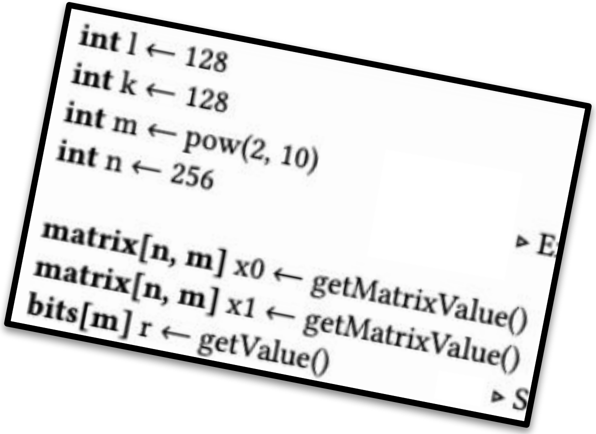
Phase 1 - Preparation

Connection channel

Exchange

- Parameters
- Algorithms





int l ← 128
int k ← 128
int m ← pow(2, 10)
int n ← 256

matrix[n, m] x0 ← getMatrixValue() ▷ E
matrix[n, m] x1 ← getMatrixValue()
bits[m] r ← getValue() ▷ S

Define values

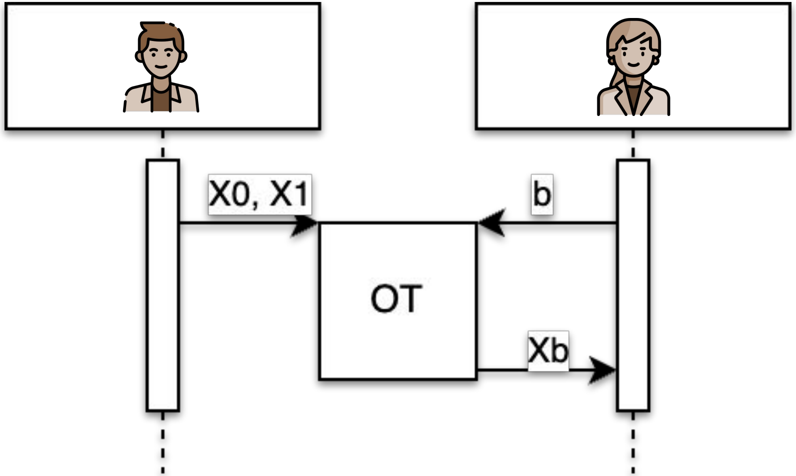
Phase 2 - Base OT

Role Switch

- Alice = Receiver
- Bob = Sender

Execution

- Random values
- Exchange Parameters

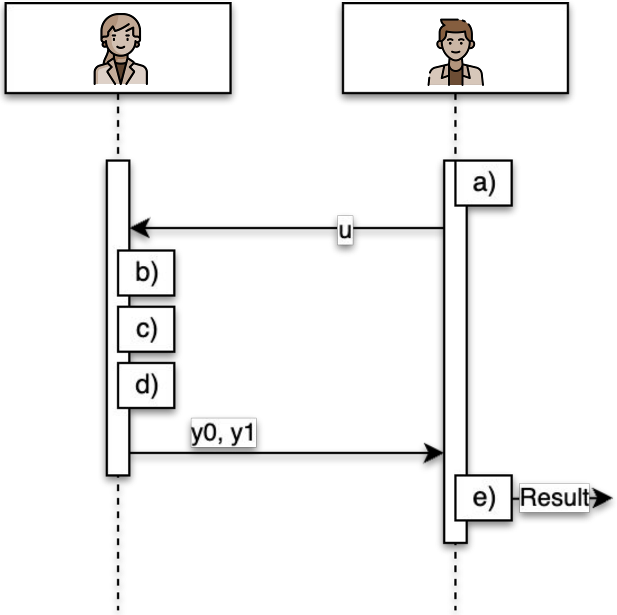


Phase 3 - Extended OT

TECHNISCHE UNIVERSITÄT DARMSTADT

Steps

- <u>B</u> - Calculate U
- <u>A</u> - Calculate Q
- <u>A</u> - Transposes Q
- <u>A</u> - Obfuscate origin to y0, y1
- <u>B</u> - Select yr
- <u>B</u> - Unveil yr



Environment → Data Type → Transposition → Algorithms

3. Implementation Details

Environment

Language

- C++

Base

- PQ-MPC [4]

Libraries

- GNU Multiple Precision Arithmetic Library (GMP)
- OpenSSL cryptographic library

Data Type

mpz_t

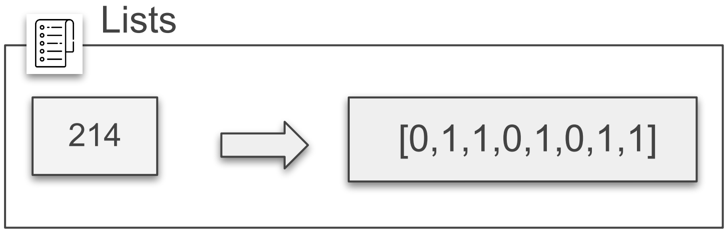
- Arbitrary precision
- Performant
- Flexible

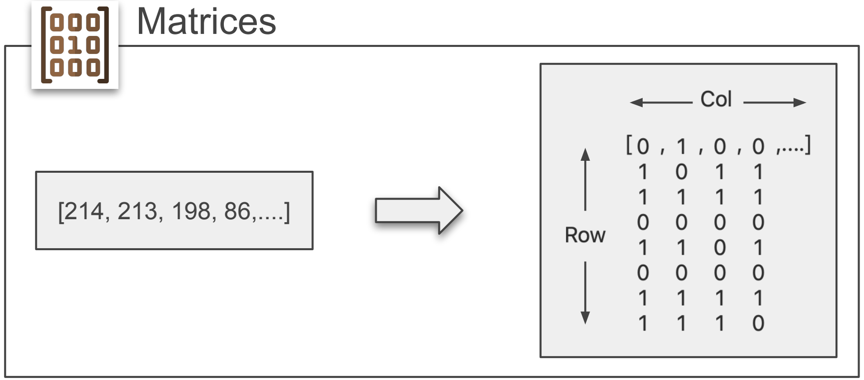
Matrices

- 2d Integer Array

Lists

- Integer





Transposition

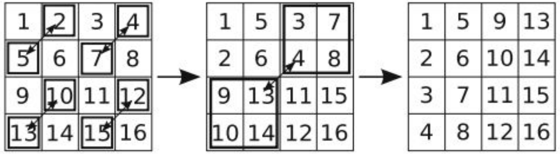
Transposition

- Simple

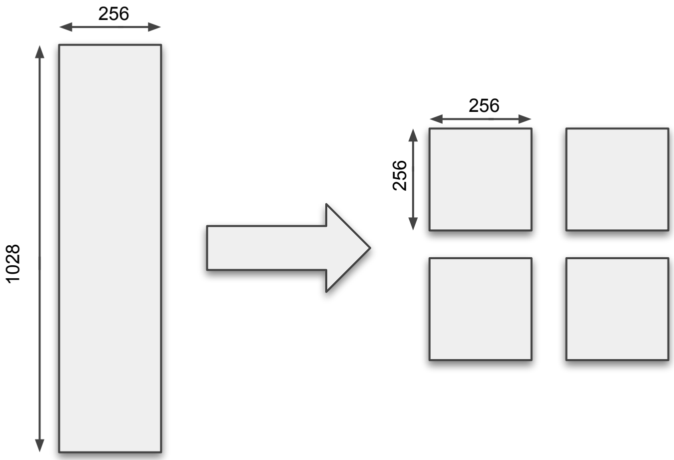
Eklundh's Transposition [3]

- based on ALSZ17 and Ekl72
  - To-swap indices computation
  - Square Matrices
  - Non-Square Matrices

Step 1 & 2



Step 3



Environment → Data Type → **Transposition** → Algorithms

Algorithms

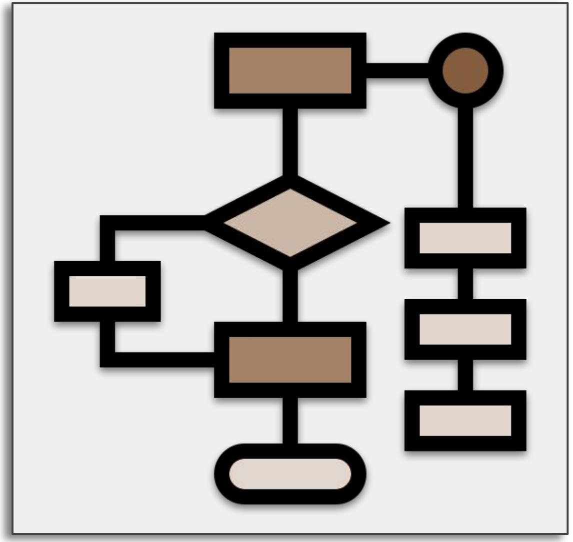
TECHNISCHE UNIVERSITÄT DARMSTADT

Correlation Robust Function

- SHA-256

PRG

- AES-CTR



Environment → Performance

4. Benchmark

Environment

TECHNISCHE UNIVERSITÄT DARMSTADT

Specs

- CPU: 2x Intel Xeon Gold 6144 @ 3.5 GHz (8 physical cores, 24.75 MB L3 Cache each)
- RAM: 16 x 32GB DDR4 ECC @ 2666 MHz (= 512GB)
- SSD: 6 x 1TB Intel NVMe U.2
- Single threaded

Fundamentals

- OT Count: $2^{10} \sim 2^{23}$
- Base OT Count = PRG security parameter = 128
- Data length = SHA-256 output = 256

Performance

# OT(2^n)	Base OT		OT Extension			
			Naive Matrix Transposition		Eklundh's Matrix Transposition	
	Time(s)	Comm.(KiB)	Time(s)	Comm.(KiB)	Time(s)	Comm.(KiB)
10	2.97	640.3	0.02	209.7	0.07	209.7
11	4.11	640.3	0.03	418.3	0.14	418.3
12	4.21	640.3	0.05	835.5	0.28	835.5
13	4.02	640.3	0.10	1670.0	0.56	1670.0
14	6.24	640.3	0.20	3338.9	1.15	3338.9
15	3.08	640.3	0.41	6676.7	2.34	6676.7
16	3.92	640.3	0.80	11352.3	4.85	11352.3
17	3.42	640.3	1.64	26703.7	9.76	26703.7
18	5.83	640.3	3.41	53406.1	19.5	53406.1
19	3.49	640.3	6.93	106810.6	39.76	106810.6
20	3.34	640.3	14.42	213621.1	79.76	213621.1
21	3.99	640.3	29.94	427240.1	161.19	427240.1
22	4.74	640.3	62.71	854479	322.48	854479
23	3.13	640.3	133.18	1708959	-	-

Results → Issues

5. Conclusion

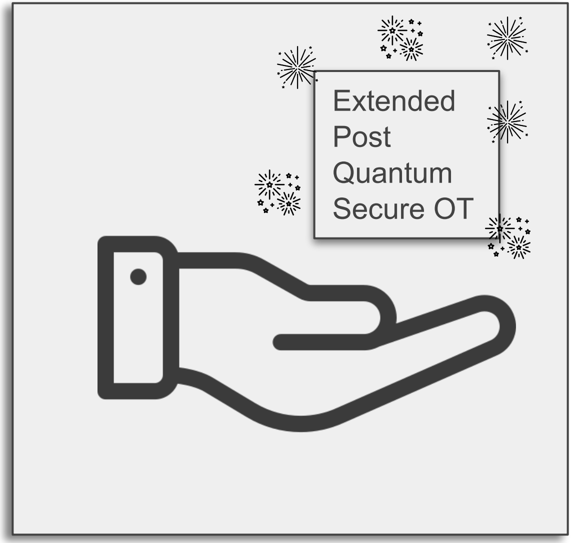
Results

Time Consumption

- ≈ Base OT (< 2^{19})

Executable OTs

- Local machine: 2^{20}
- Virtualization server: 2^{23}



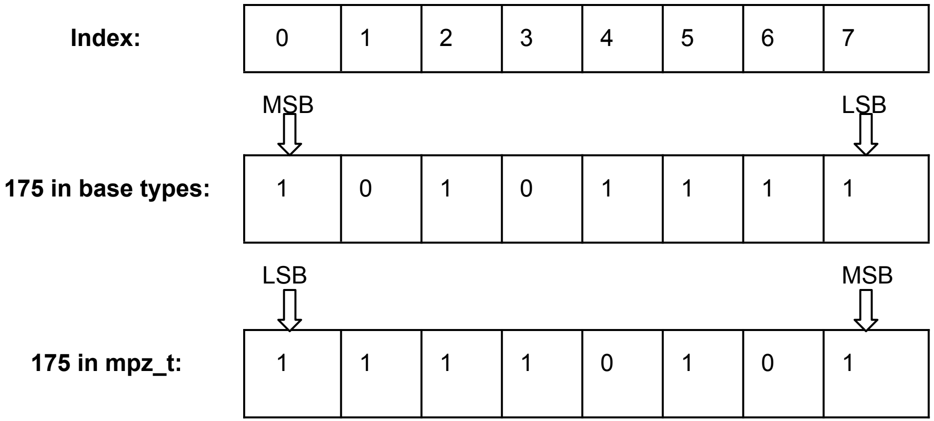
Issues

Eklundh's Transposition

- mpz_t performance

Hard to Debug

- mpz_t bit order



6. Demo

7. Q & A

References - Sources

1. [ALSZ17] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions, 2017.
2. [BDK+20] Niklas Büscher, Daniel Demmler, Nikolaos P. Karvelas, Stefan Katzenbeisser, Juliane Krämer, Deevashwer Rathee, T. Schneider, and Patrick Struck. Secure two-party computation in a quantum world, 2020.
3. [Ekl72] J.O. Eklundh. A fast computer method for matrix transposing. IEEE Transactions on Computers, C-21(7):801–803, 1972.
4. https://github.com/encryptogroup/PQ-MPC
5. [MI81] Rabin, Michael. (1981). How To Exchange Secrets with Oblivious Transfer

References - Images

TECHNISCHE UNIVERSITÄT DARMSTADT

1. https://www.flaticon.com/free-icons/lock
2. https://www.flaticon.com/free-icons/man
3. https://www.flaticon.com/free-icons/woman
4. https://www.flaticon.com/free-icons/machine-learning
5. https://www.flaticon.com/free-icons/algorithm
6. https://www.flaticon.com/free-icons/goal
7. https://www.flaticon.com/free-icons/hand
8. https://www.flaticon.com/free-icons/firework

THE END

Special thanks to Gowri and Jens for the support

Thanks for your Attention