AtlantTIC
Research Center for
Information & Communication Technologies

Multiquadratic Rings and Oblivious Linear Function Evaluation

3rd BYMAT Conference
Bringing Young Mathematicians Together, *December 1-3, 2020*

Alberto Pedrouzo-Ulloa, *atlanTTic Research Center, University of Vigo*
apedrouzo@gts.uvigo.es

Outline

- Introduction
- Faster polynomial arithmetic
- OLE applications
- Conclusions

Introduction

Main motivations for this work

Homomorphic Encryption (HE)

- Modern HE (e.g., BFV, CKKS) relies heavily on polynomial arithmetic (**polynomial multiplication and addition**)

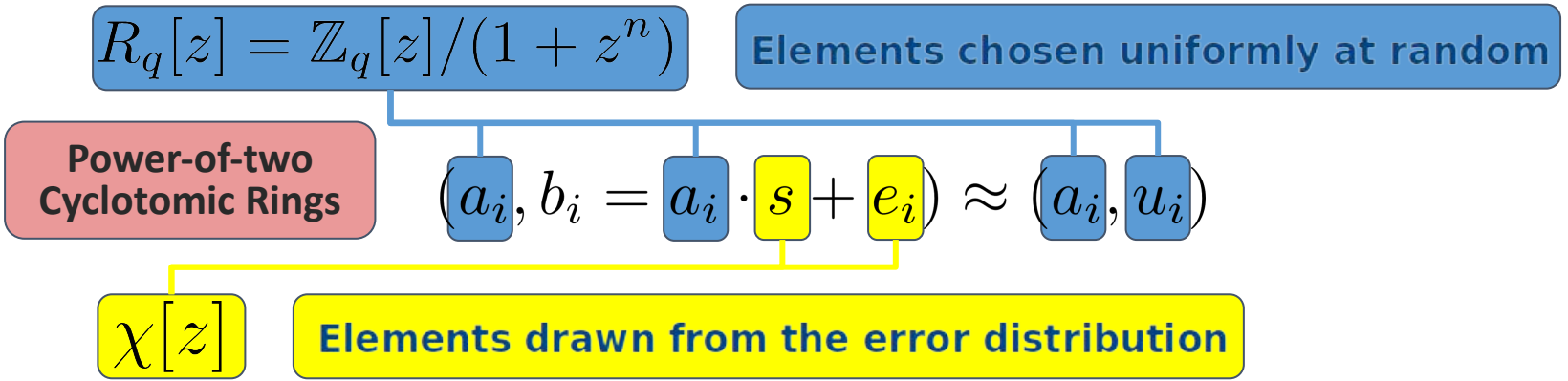
$$\left\{ \begin{array}{c} E(m_1) + E(m_2) = E(m_1 + m_2) \\ E(m_1) \cdot E(m_2) = E(m_1 \cdot m_2) \end{array} \right\}$$

Homomorphic Encryption (HE)

- Modern HE (e.g., BFV, CKKS) relies heavily on polynomial arithmetic (**polynomial multiplication and addition**)
- **The RLWE problem** relies upon the computational indistinguishability between the following pairs of samples:

Homomorphic Encryption (HE)

- Modern HE (e.g., BFV, CKKS) relies heavily on polynomial arithmetic (**polynomial multiplication and addition**)
- **The RLWE problem** relies upon the computational indistinguishability between the following pairs of samples:



Ciphertext representation

- Polynomial multiplication/addition **fits well in the CRT-NTT representation**

Ciphertext representation

- Polynomial multiplication/addition **fits well in the CRT-NTT representation**
  - **NTT** (Negacyclic Number Theoretic Transform):
    - The whole polynomial is transformed into the frequency domain (**convolution theorem**)
    - Used for **efficient FFT-based polynomial multiplication**
  - **CRT** (Chinese Remainder Theorem):
    - Each polynomial coefficient is decomposed in **several values fitting into the machine word**.

Ciphertext representation

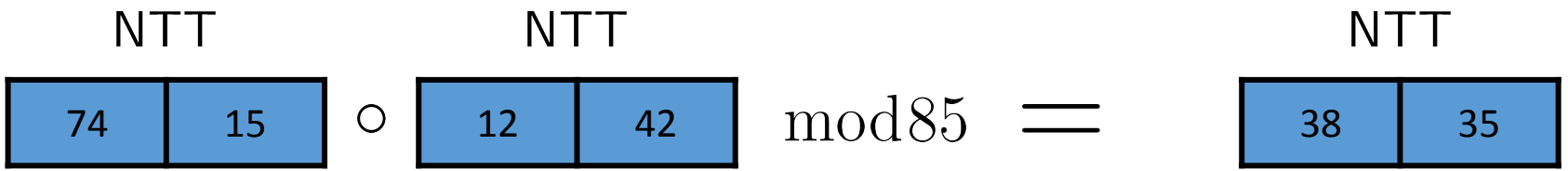
- Polynomial multiplication/addition **fits well in the CRT-NTT representation**

$$((2+1z)(27+60z) \bmod 85) \bmod 1+z^2 \quad == \quad 79+62z$$

Ciphertext representation

- Polynomial multiplication/addition **fits well in the CRT-NTT representation**

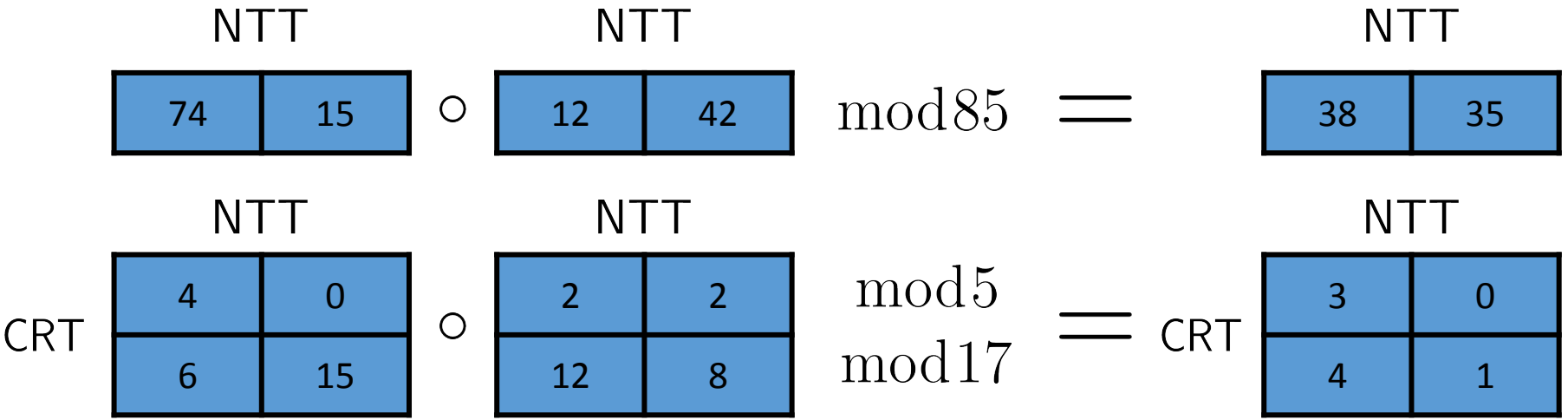
$$((2+1z)(27+60z) \bmod 85) \bmod 1+z^2 \quad = \quad 79+62z$$



Ciphertext representation

- Polynomial multiplication/addition **fits well in the CRT-NTT representation**

$$((2+1z)(27+60z) \bmod 85) \bmod 1+z^2 = 79+62z$$



Division and rounding over the CRT-NTT representation

- Rounding operations are also an **important step on current state-of-the-art HE schemes**.

- There exist **efficient methods** to perform rounding i**n the CRT representation**

- However, **we still have to reverse the NTT representation before rounding**.
  - This introduces an overhead of $\mathcal{O}(n \log n)$ multiplications to swap between representations

Division and rounding over the CRT-NTT representation

Main motivation for this work:

Can the cost of this overhead be improved?

- However, **we still have to reverse the NTT representation before rounding**.
  - This introduces an overhead of $\mathcal{O}(n \log n)$ multiplications to swap between representations

Faster polynomial arithmetic

over multiquadratic rings

Walsh-Hadamard Transform

- **The Walsh-Hadamard transform** over real numbers is usually defined by means of the recursion

$$\boldsymbol{H}_i = \frac{1}{\sqrt{2}} \begin{pmatrix} \boldsymbol{H}_{i-1} & \boldsymbol{H}_{i-1} \\ \boldsymbol{H}_{i-1} & -\boldsymbol{H}_{i-1} \end{pmatrix}$$

where $i \in \mathbb{N}^+$ and $\boldsymbol{H}_0 = 1$

Walsh-Hadamard Transform

- **The Hadamard transform** can be very efficiently computed with the **FWHT** (Fast Walsh Hadamard Transform)
- Cost of $\mathcal{O}(n \log n)$ additions and subtractions
- Security reasons prevent us from using $f(x_i) = x_i^2 - 1$ because it can be factored into $(x_i - 1)(x_i + 1)$ over $\mathbb{Z}$

Walsh-Hadamard Transform

- **The Hadamard transform** can be very efficiently computed with the **FWHT** (Fast Walsh Hadamard Transform)
- Cost of $\mathcal{O}(n \log n)$ additions and subtractions
- Security reasons prevent us from using $f(x_i) = x_i^2 - 1$ because it can be factored into $(x_i - 1)(x_i + 1)$ over $\mathbb{Z}$
- Alternative choice: **Multiquadratic Rings**

$$R[x_1, \ldots, x_l] = \mathbb{Z}[x_1, \ldots, x_l]/(d_1 + x_1^2, \ldots, d_l + x_l^2)$$

Walsh-Hadamard Transform

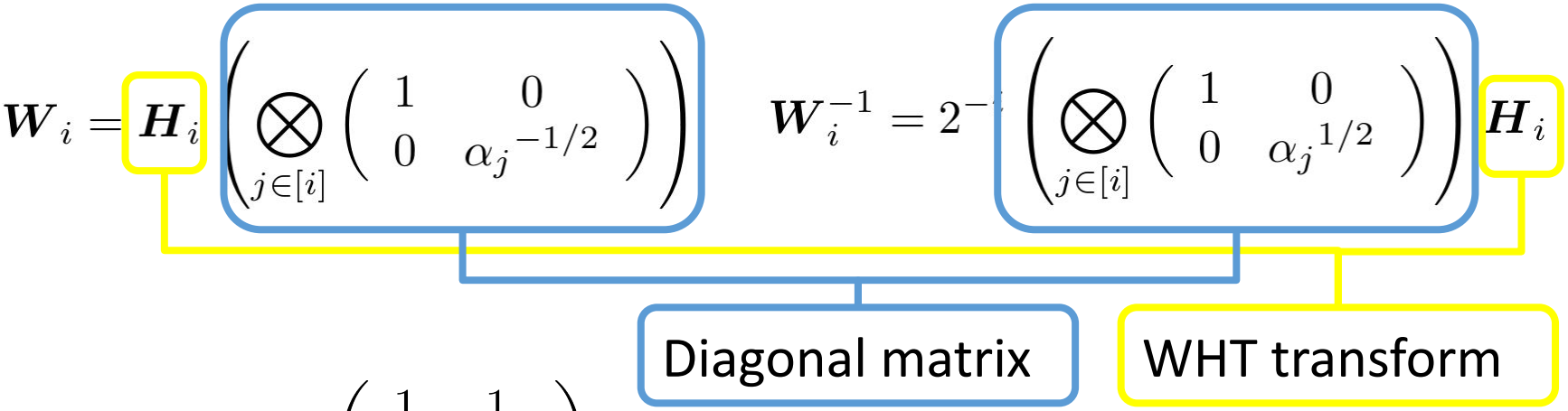
- **The Hadamard transform** can be very efficiently computed with the **FWHT** (Fast Walsh Hadamard Transform)
- Cost of $\mathcal{O}(n \log n)$ additions and subtractions
- Security reasons prevent us from using $f(x_i) = x_i^2 - 1$ because it can be factored into $(x_i - 1)(x_i + 1)$ over $\mathbb{Z}$
- Alternative choice: **Multiquadratic Rings**

$$P[x \ldots x_l] \mathbb{Z}[x \ldots x_l]/(d \ldots + x_1^2, \ldots, d_l + x_l^2)$$

Toy example: $\mathbb{Z}[x, y]/(x^2 + 3, y^2 - 5)$

$\propto$-Generalized Walsh-Hadamard Transform

- We define this transform by means of the following recursions



where $\boldsymbol{H}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

Faster polynomial arithmetic with multiquadratics

- Reduced cost of polynomial operations:
  - α-generalized FWHT enables polynomial multiplications with a computational cost of $\mathcal{O}(n \log n)$ additions and $\mathcal{O}(n)$ multiplications
  - The asymptotic cost of $\mathcal{O}(n \log n)$ coefficient products with cyclotomics is reduced by a $\log n$ factor

Implementation of the $\propto$-Generalized FWHT

- Implemented in C++
- Comparison with the radix-2 implementation of the NTT/INTT from the NFLlib library
- NFLlib makes use of SSE and AVX2 optimizations

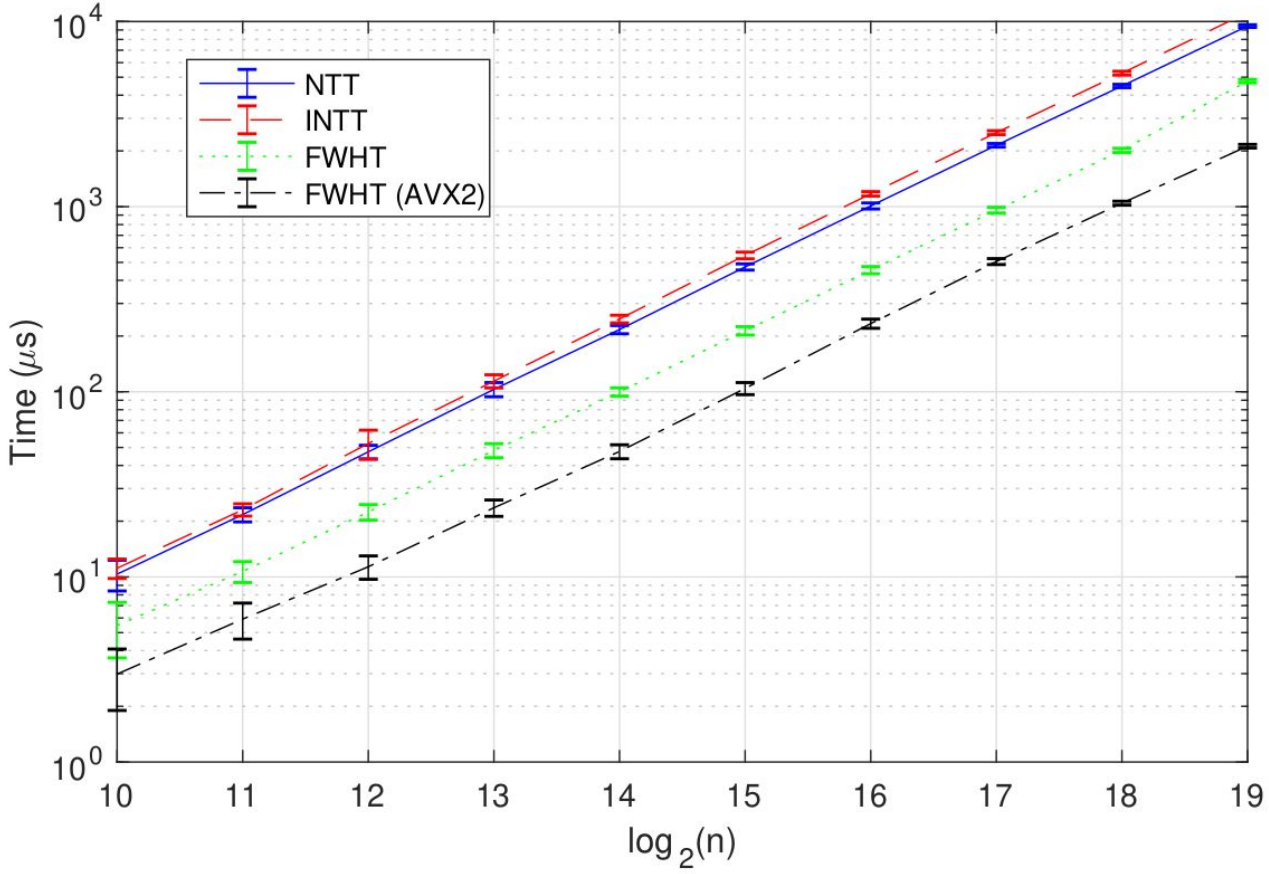
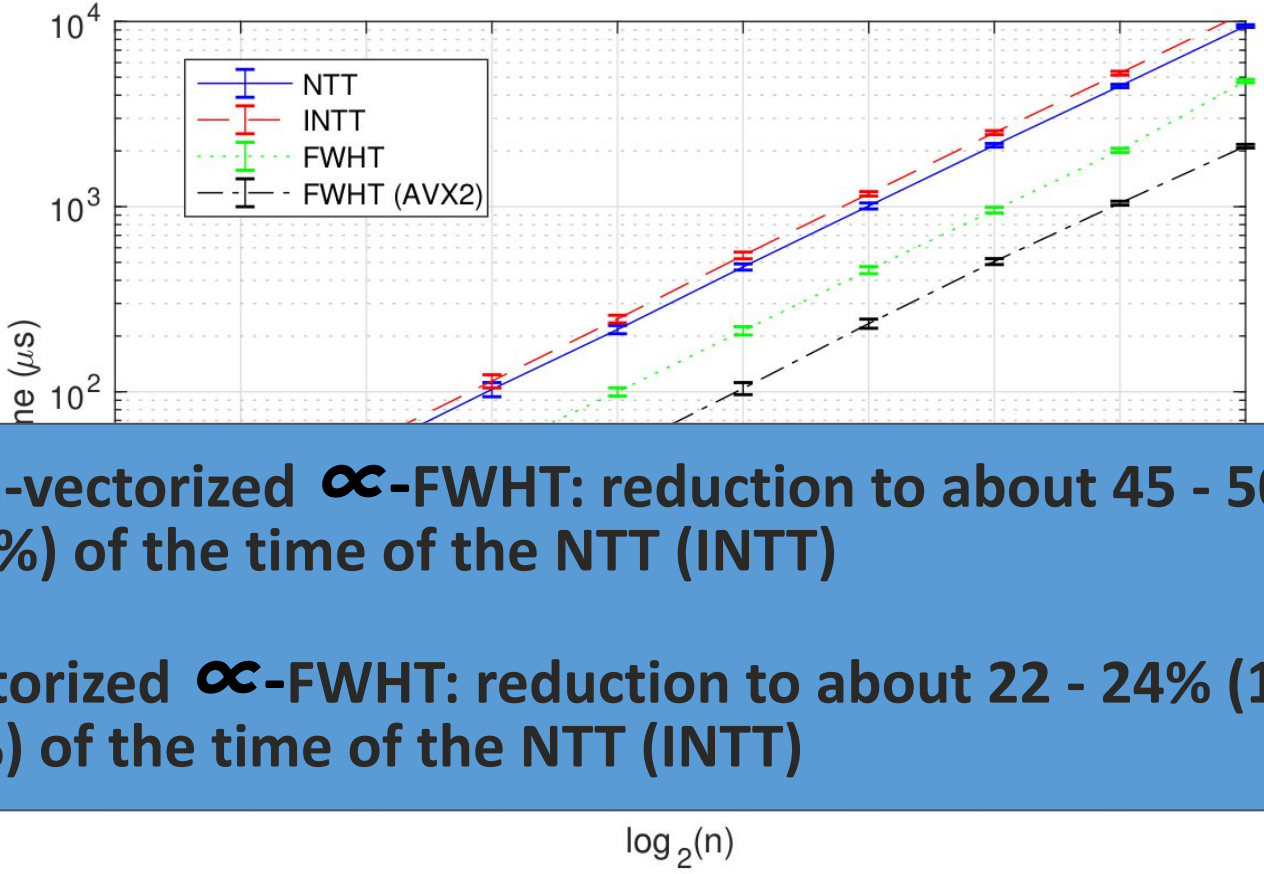


Fig. 2. Runtimes of the proposed FWHT compared to the NTT/INTT [18].



- **Non-vectorized $\propto$-FWHT: reduction to about 45 - 50% (38 - 43%) of the time of the NTT (INTT)**
- **Vectorized $\propto$-FWHT: reduction to about 22 - 24% (19 - 22%) of the time of the NTT (INTT)**

Fig. 2. Runtimes of the proposed FWHT compared to the NTT/INTT [18].

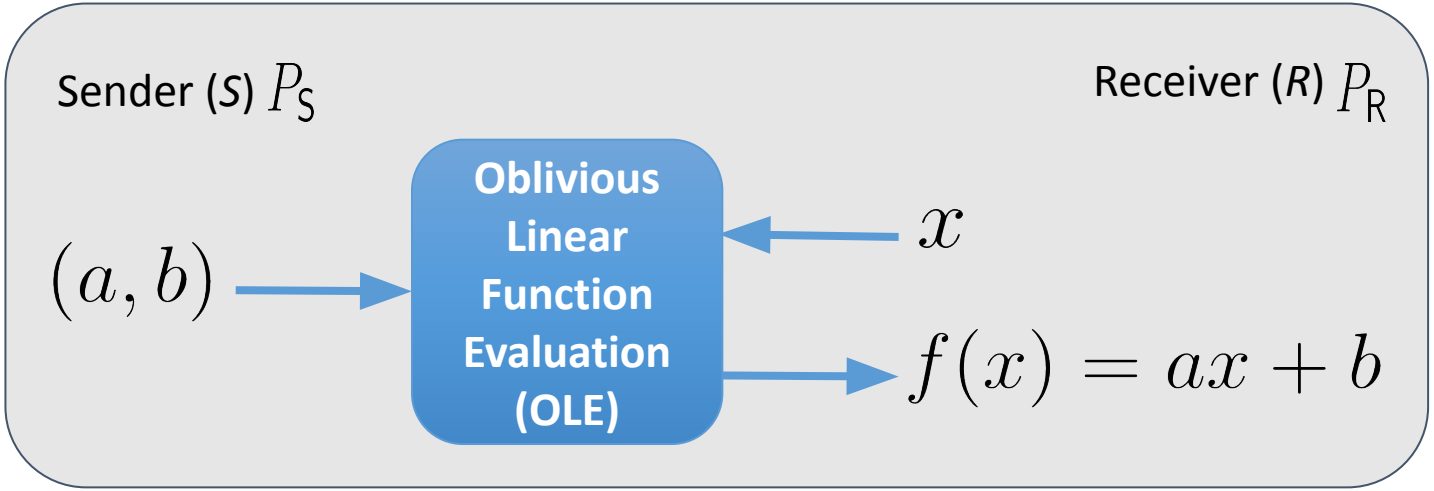
OLE applications

for matrix products and convolutions

OLE (Oblivious Linear Function Evaluation) applications

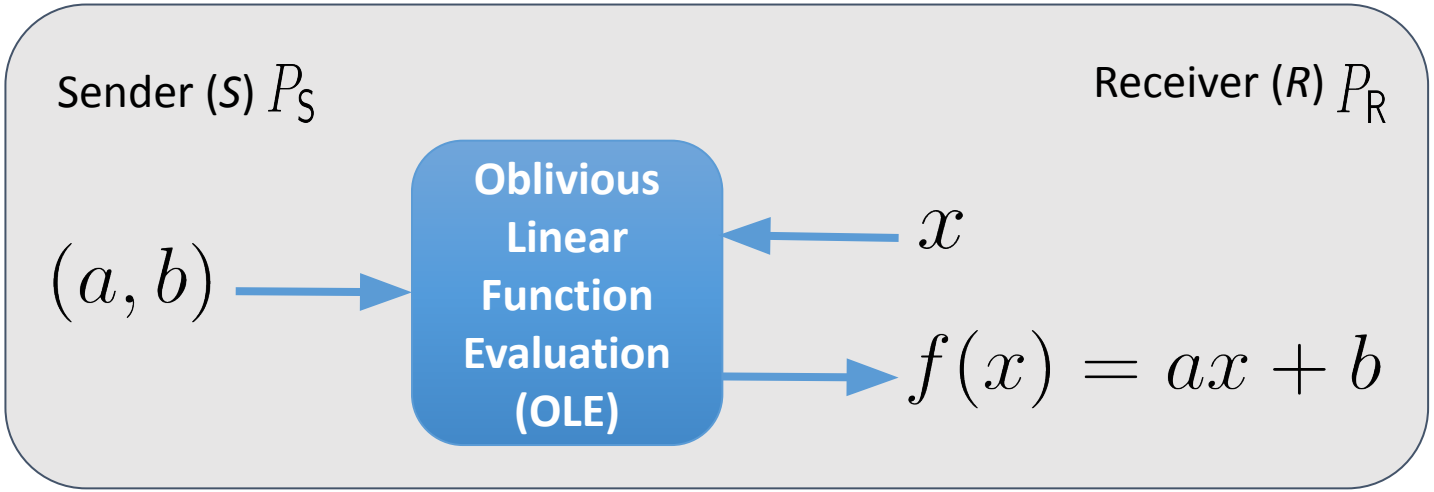
- **OLE is one of the main building blocks for MPC**:
  - It can **implement OPE** (Oblivious Polynomial Evaluation)
    - Function evaluation by means of their polynomial approximation
  - **Secure set intersection**
  - **Generation of correlated randomness** in the offline phase of SPDZ-type protocols
- More practical use cases:
  - Secure computation of **matrix-vector products**
  - Secure **computation of the convolution layers** of a neural network

Oblivious Linear Function Evaluation



Oblivious Linear Function Evaluation

- The **OLE primitive** can be computed by means of homomorphic encryption schemes (**AHE-based OLE**)



Performance comparison for OLE

- Multiquadratic rings cause an increase in the number of limbs

TABLE II

EXAMPLE PARAMETER SET FOR A TOTAL OF $32768 \cdot l$ OLES WITH A PLAINTEXT'S MODULUS p OF 120 BITS (2 LIMBS) AND $\sigma = 3.2$, $\lambda = 80$.

Parameter	Par. set 1	Par. set 2	Par. set 3
$\{n, \#\text{batches}, B\}$	$\{32768, l, 32 \cdot D\}$	$\{32768, l, 32\}$	$\{16384, 2l, 32\}$
q	720 bits (12 limbs)	420 bits (7 limbs)	420 bits (7 limbs)
bit security	> 128	> 256	≈ 128

Performance comparison for OLE

- Multiquadratic rings cause an increase in the number of limbs
- A higher ring expansion factor δ_R

$$5 \log_2 D \approx 337 \text{ bits}$$

TABLE II

EXAMPLE PARAMETER SET FOR A TOTAL OF $32768 \cdot l$ OLES WITH A PLAINTEXT'S MODULUS p OF 120 BITS (2 LIMBS) AND $\sigma = 3.2$, $\lambda = 80$.

Parameter	Par. set 1	Par. set 2	Par. set 3
$\{n, \#\text{batches}, B\}$	$\{32768, l, 32 \cdot D\}$	$\{32768, l, 32\}$	$\{16384, 2l, 32\}$
q	720 bits (12 limbs)	420 bits (7 limbs)	420 bits (7 limbs)
bit security	> 128	> 256	≈ 128

Performance comparison for OLE

- Multiquadratic rings cause an increase in the number of limbs

Multiquadratic Ring

Power-of-two cyclotomics

$$5 \log_2 D \approx 337 \text{ bits}$$

TABLE II
EXAMPLE PARAMETER SET FOR A TOTAL OF $32768 \cdot l$ OLES WITH A PLAINTEXT'S MODULUS p OF 120 BITS (2 LIMBS) AND $\sigma = 3.2$, $\lambda = 80$.

Parameter	Par. set 1	Par. set 2	Par. set 3
$\{n, \#\text{batches}, B\}$	$\{32768, l, 32 \cdot D\}$	$\{32768, l, 32\}$	$\{16384, 2l, 32\}$
q	720 bits (12 limbs)	420 bits (7 limbs)	420 bits (7 limbs)
bit security	> 128	> 256	≈ 128

Performance comparison for OLE

- Multiquadratic rings cause an increase in the number of limbs
- Estimated runtimes:
  - Par. set 1 is at least **2 times faster** than Par. set 3
  - Par. set 1 is at least **2.5 times faster** than Par. set 2

Multiquadratic Ring

Power-of-two cyclotomics

TABLE II

RAM[illegible] $2768 \cdot l$ OLEs WITH A PLAINTEXT'S MODULUS p OF 120 BITS (2 LIMBS) AND $\sigma = 3.2$, $\lambda = 80$.

Parameter	Par. set 1	Par. set 2	Par. set 3
$\{n, \#\text{batches}, B\}$	$\{32768, \ , 32 \cdot D\}$	$\{32768 \ l, 32\}$	$\{16384, 2l, 32\}$
q	720 bits (12 limbs)	420 bits (7 limbs)	420 bits (7 limbs)
bit security	> 128	> 256	≈ 128

Conclusions

A very brief summary of our results

Conclusions

- This work studies the use of **multiquadratic rings in the RLWE problem**
- We introduce a **generalization of the Walsh-Hadamard Transform** for finite rings
- The multiplicative cost of the polynomial multiplication is **reduced by a logarithmic factor**
- We **exemplify the use of multiquadratic rings for the implementation of OLE**, where it can bring about efficiency improvements comparing to power-of-two cyclotomics

Thanks for your attention!