TU Graz

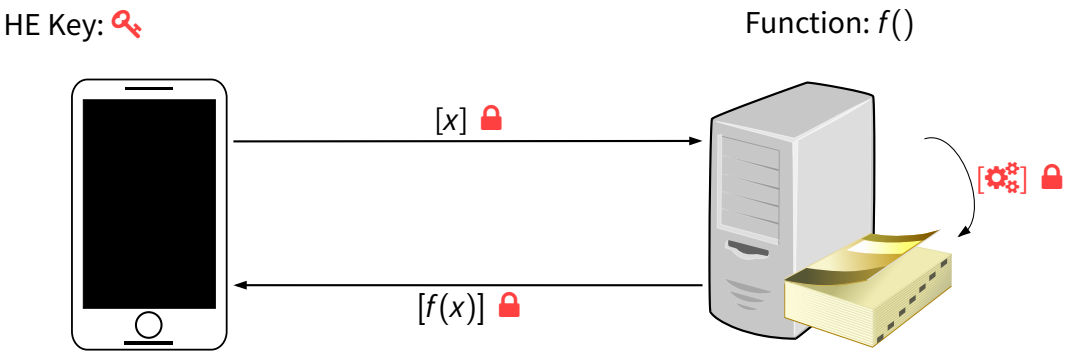
SCIENCE
PASSION
TECHNOLOGY

Pasta: A Case for Hybrid Homomorphic Encryption

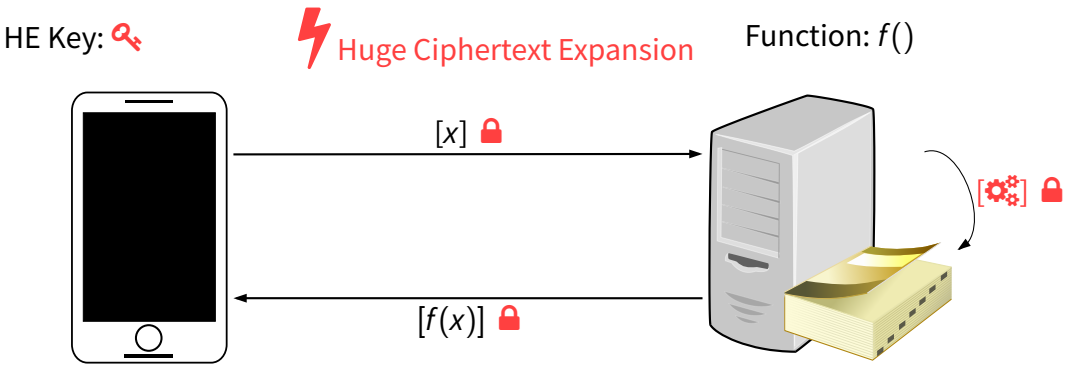
Christoph Dobraunig, Lorenzo Grassi, Lukas Helminger, Christian Rechberger, Markus Schofnegger, **Roman Walch**

29.05.2022

Homomorphic Use Cases



Homomorphic Use Cases



Hybrid Homomorphic Encryption

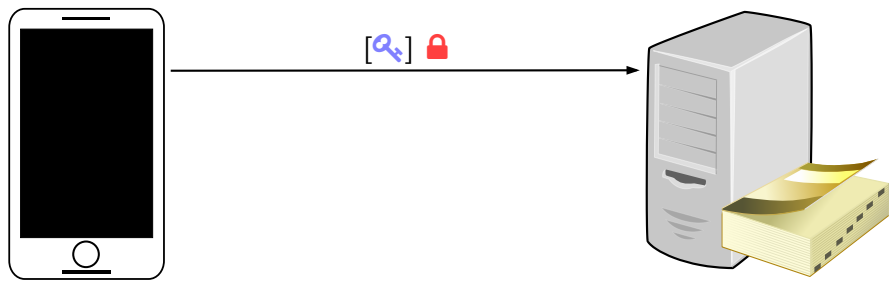
- Homomorphic ciphertext are orders of magnitude larger then plaintext
  - e.g., 7.4 MB for $\leq$ 250 kB
- Solution:
  - Send data encrypted using symmetric ciphers
    - Ciphertexts have same size as plaintexts
  - Homomorphically decrypt data before use case

$\Rightarrow$ No ciphertext expansion

Hybrid HE Use Cases

HE Key: , Sym Key:

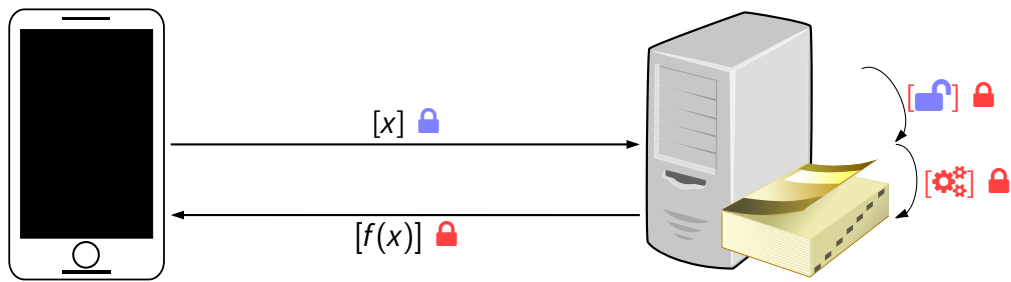
Function: $f()$



Hybrid HE Use Cases

HE Key: , Sym Key:

Function: $f()$, [ ]



Hybrid Homomorphic Encryption (cont.)

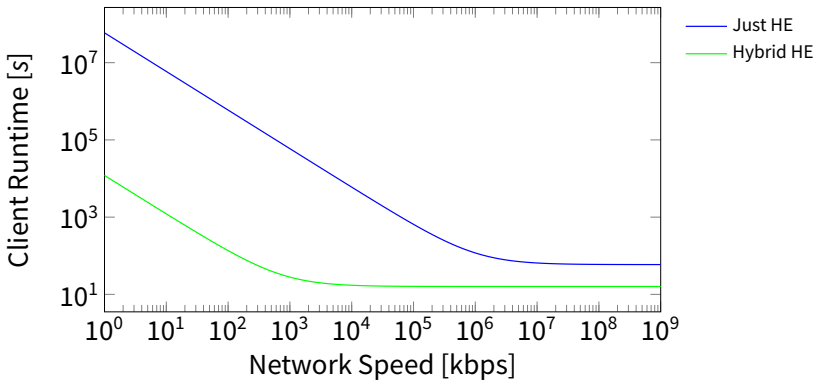
- Server evaluates symmetric decryption circuit before use case
  - Contributes to multiplicative depth of homomorphic computation
  - Without bootstrapping:
    - Shallow decryption circuit required
    - Requires more noise budget
    - $\Rightarrow$ Slower runtime
  - $\Rightarrow$ traditional ciphers (e.g., AES) not well suited

$\Rightarrow$ Tradeoff: data transmission vs. server runtime

- Usable for constrained clients in slow networks!

Client Runtime Comparison

- Encryption + upload time depending on network speed:



High-Level Take-Aways

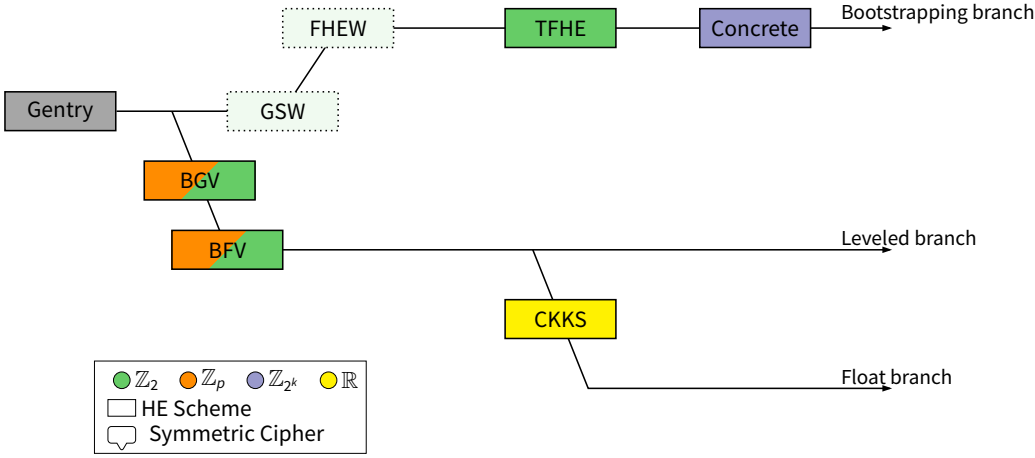
- Client Savings:
  - ✚ Bandwidth savings:
    - Only send actual data size
  - ✚ Runtime savings:
    - Symmetric encryption significantly faster as homomorphic encryption
    - No randomness required
- Server overhead:
  - ━ Addition to multiplicative depth
    - Depends on cipher
  - ━ Runtime overhead:
    - Depends on use case and total multiplicative depth
    - Depends on HE library

High-Level Take-Aways

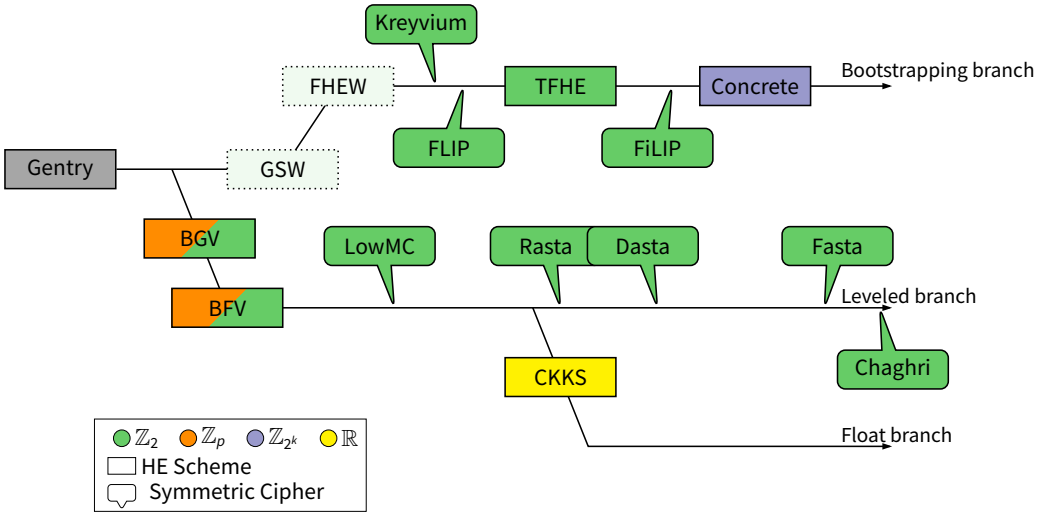
- Client Savings:
  - ✚ Bandwidth savings:
    - Only send actual data size
  - ✚ Runtime savings:
    - Symmetric encryption significantly faster as homomorphic encryption
    - No randomness required
- Server overhead:
  - ━ Addition to multiplicative depth
    - Depends on cipher
  - ━ Runtime overhead:
    - Depends on use case and total multiplicative depth
    - Depends on HE library

Ciphers for HHE

A Zoo of Ciphers for HHE



A Zoo of Ciphers for HHE



Inefficiency of $\mathbb{Z}_2$ ciphers

- So far all ciphers over $\mathbb{Z}_2$
  - Good for binary use cases
- Not ideal otherwise:
  - Less throughput compared to $\mathbb{Z}_p$
  - No homomorphic conversion from $\mathbb{Z}_2$ to $\mathbb{Z}_p$ in BGV/BFV
    - Use case afterwards in $\mathbb{Z}_2$
    - Binary circuits for integer arithmetic

$\Rightarrow$ Focus on ciphers over $\mathbb{Z}_p$

- Our approach: Pasta
  - First public release: June 2021

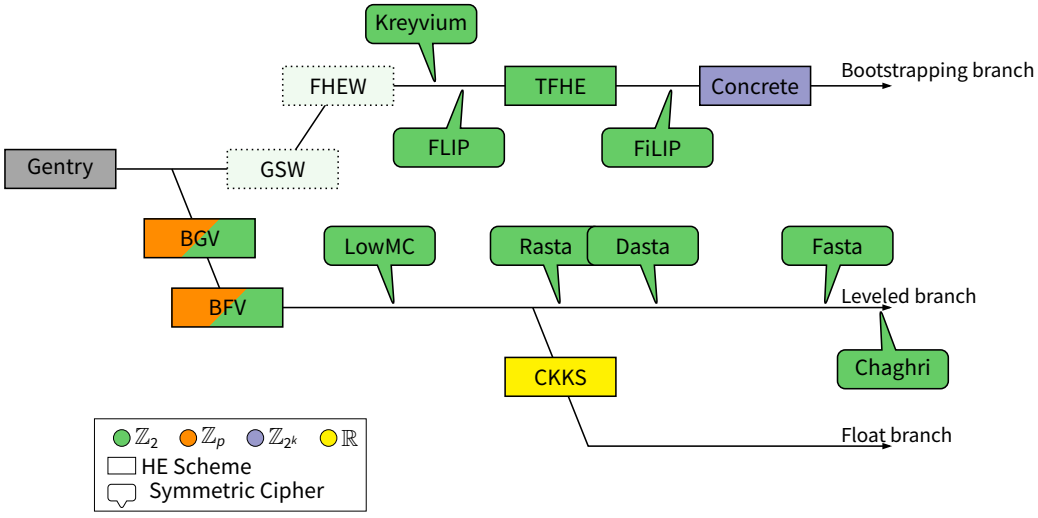
Inefficiency of $\mathbb{Z}_2$ ciphers

- So far all ciphers over $\mathbb{Z}_2$
  - Good for binary use cases
- Not ideal otherwise:
  - Less throughput compared to $\mathbb{Z}_p$
  - No homomorphic conversion from $\mathbb{Z}_2$ to $\mathbb{Z}_p$ in BGV/BFV
    - Use case afterwards in $\mathbb{Z}_2$
    - Binary circuits for integer arithmetic

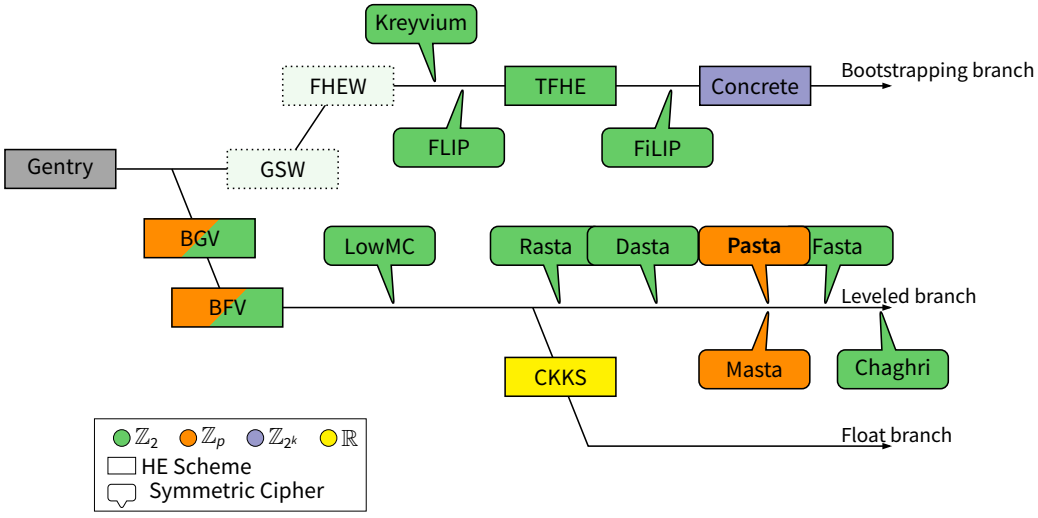
$\Rightarrow$ Focus on ciphers over $\mathbb{Z}_p$

- Our approach: Pasta
  - First public release: June 2021

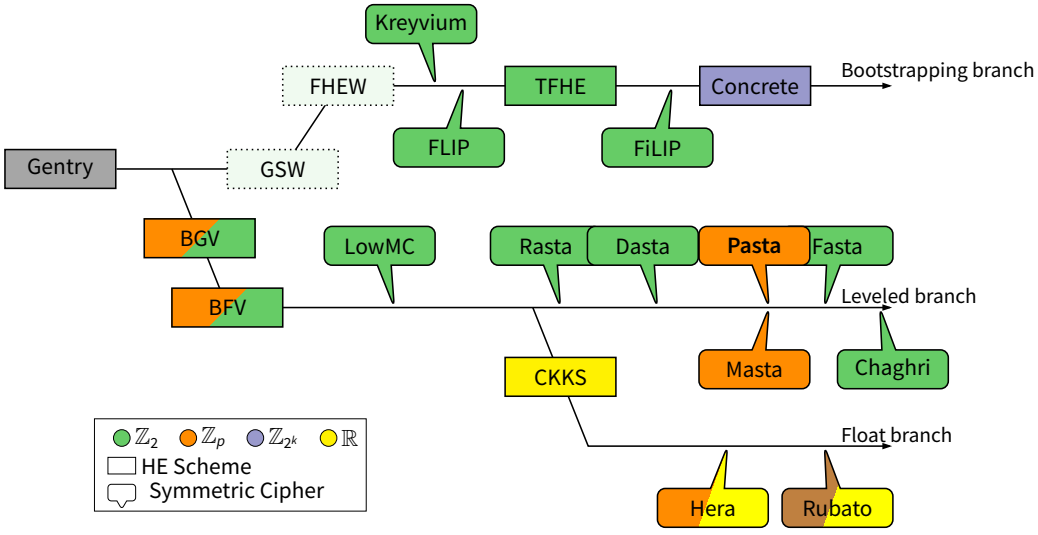
A Zoo of Ciphers for HHE (cont.)



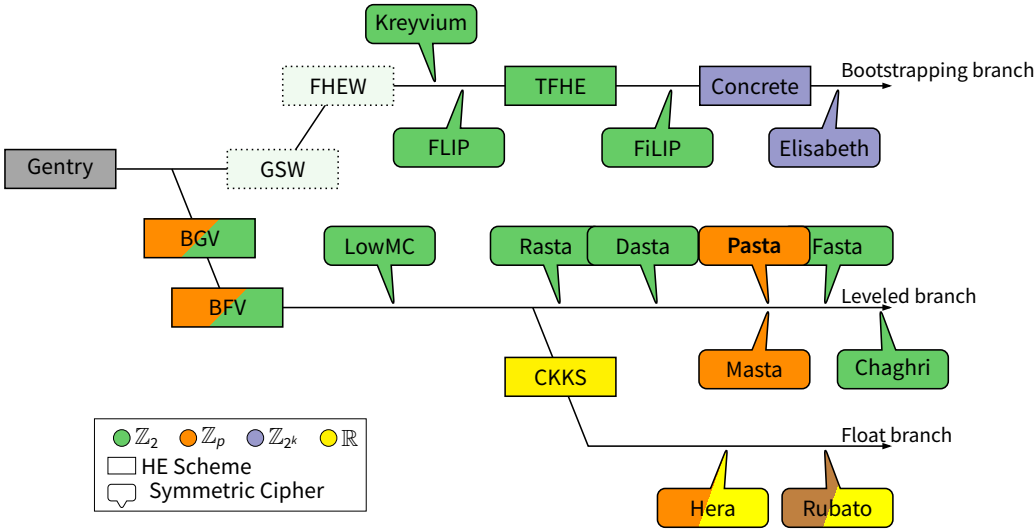
A Zoo of Ciphers for HHE (cont.)



A Zoo of Ciphers for HHE (cont.)



A Zoo of Ciphers for HHE (cont.)



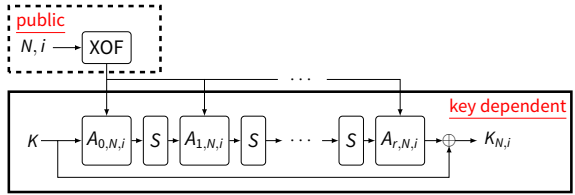
Pasta

Design Goals

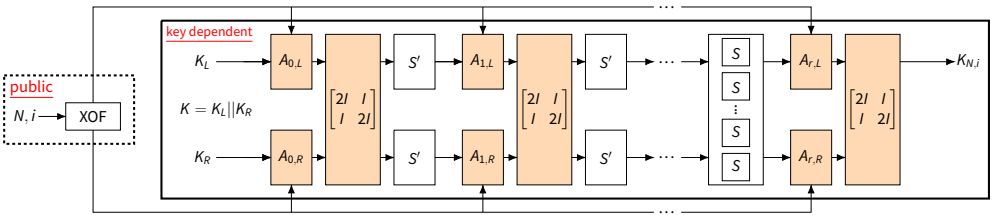
- Cipher over $\mathbb{Z}_p$
- Low multiplicative depth
  - Minimize number of rounds
  - Minimize depth per round
  - $\Rightarrow$ Often large statesize
- Optimize for packing in BFV/BGV
  - Fast homomorphic evaluation time despite large statesize

Background: The RASTA Design Strategy

- Stream-Cipher
- Affine transformation:
  - XOF seeded with N, i
  - Random invertible matrices
  - Random round constants
- MASTA: Direct translation to $\mathbb{Z}_p$



The PASTA Design Strategy – Linear Layer



- Linear layer:

$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \times \begin{bmatrix} A_{i,L}(\vec{x}_L) \\ A_{i,R}(\vec{x}_R) \end{bmatrix}$$

  - ... with random A_j

Reminder Packing

- Packing:
  - Encode a vector of integers into polynomials
  - Homomorphic additions/multiplication $\Rightarrow$ slotwise vector operations
  - Vector rotations via galois automorphisms
- BSGS optimized diagonal method for matrix-vector multiplication
  - $M \cdot \vec{x}$ on packed $\vec{x}$ using rotations
- Rotations in power-of-two cyclotomic rings

$$\begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} \overset{\text{encode}}{\rightarrow} x \in R_p : \qquad \tau_{3^i}(x) \overset{\text{decode}}{\rightarrow} \begin{bmatrix} \texttt{rot}_i(\vec{x}_L) \\ \texttt{rot}_i(\vec{x}_R) \end{bmatrix}, \qquad \tau_{N-1}(x) \overset{\text{decode}}{\rightarrow} \begin{bmatrix} \vec{x}_R \\ \vec{x}_L \end{bmatrix},$$

Reminder Packing

- Packing:
  - Encode a vector of integers into polynomials
  - Homomorphic additions/multiplication $\Rightarrow$ slotwise vector operations
  - Vector rotations via galois automorphisms
- BSGS optimized diagonal method for matrix-vector multiplication
  - $M \cdot \vec{x}$ on packed $\vec{x}$ using rotations
- Rotations in power-of-two cyclotomic rings

$$\begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} \overset{\text{encode}}{\rightarrow} x \in R_p : \qquad \tau_{3^i}(x) \overset{\text{decode}}{\rightarrow} \begin{bmatrix} \texttt{rot}_i(\vec{x}_L) \\ \texttt{rot}_i(\vec{x}_R) \end{bmatrix}, \qquad \tau_{N-1}(x) \overset{\text{decode}}{\rightarrow} \begin{bmatrix} \vec{x}_R \\ \vec{x}_L \end{bmatrix},$$

Efficient Homomorphic PASTA Implementation – Linear Layer

- PASTA in power-of-two cyclotomic rings:
  - Parallelize evaluation of $A_{j,L}$ and $A_{j,R}$

$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} A_{j,L}(\vec{x}_L) \\ A_{j,R}(\vec{x}_R) \end{bmatrix}$$

  - Mixing:

$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \cdot \begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} = \begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} + \begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} + \begin{bmatrix} \vec{x}_R \\ \vec{x}_L \end{bmatrix},$$

$\Rightarrow$ Cost reduced from $(2t \times 2t)$ to $(t \times t)$ matrix-vector multiplication

Efficient Homomorphic PASTA Implementation – Linear Layer

- PASTA in power-of-two cyclotomic rings:
  - Parallelize evaluation of $A_{j,L}$ and $A_{j,R}$

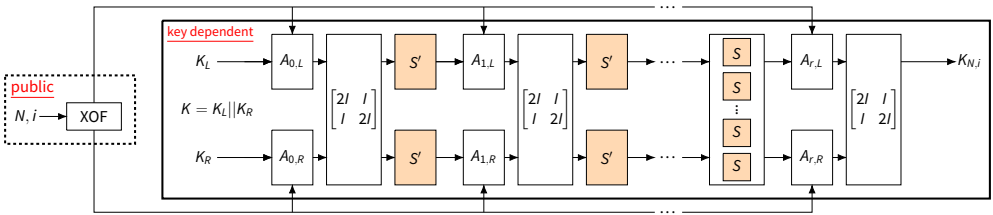
$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} A_{j,L}(\vec{x}_L) \\ A_{j,R}(\vec{x}_R) \end{bmatrix}$$

  - Mixing:

$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \cdot \begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} = \begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} + \begin{bmatrix} \vec{x}_L \\ \vec{x}_R \end{bmatrix} + \begin{bmatrix} \vec{x}_R \\ \vec{x}_L \end{bmatrix},$$

$\Rightarrow$ Cost reduced from $(2t \times 2t)$ to $(t \times t)$ matrix-vector multiplication

The PASTA Design Strategy – S-box



- Feistel-like S-box:
  - Low-degree $\Rightarrow$ low depth

$$[S'(\vec{x})]_i = \begin{cases} x_0 & \text{if } i = 0 \\ x_i + (x_{i-1})^2 & \text{else} \end{cases}$$

- Cube S-box:
  - Higher degree
  - Only last round

$$S(x) = x^3$$

Pasta vs. Masta

- PASTA has less rounds for same keystream words
  - …also scales better with rounds

Instance	Rounds	# Key Words	# Plain/Cipher Words
PASTA-3	3	256	128
PASTA-4	4	64	32
MASTA-4	4	128	128
MASTA-5	5	64	64

Table: 128 bit security instances of PASTA and MASTA.

Benchmarks

Benchmarking Framework

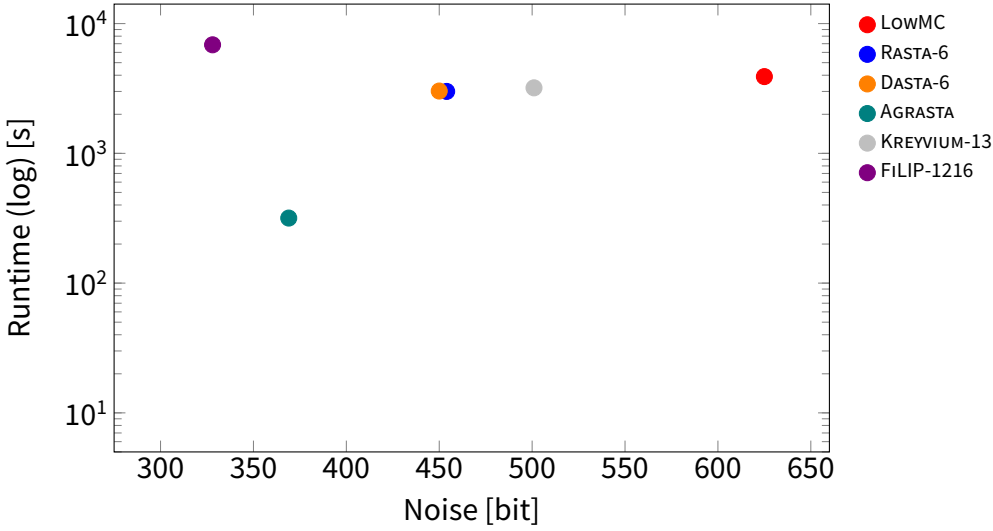
- Hybrid Homomorphic Encryption Framework
  - Extensive benchmarks in different HE libraries
  - SEAL, HElib for $\mathbb{Z}_2$ and $\mathbb{F}_p$
  - TFHE for $\mathbb{Z}_2$
  - https://github.com/IAIK/hybrid-HE-framework
- Benchmarks
  - Homomorphic evaluation of decryption circuit
  - HHE with use case evaluation
    - More meaningful benchmarks!

$\Rightarrow$ Used by many followup designs

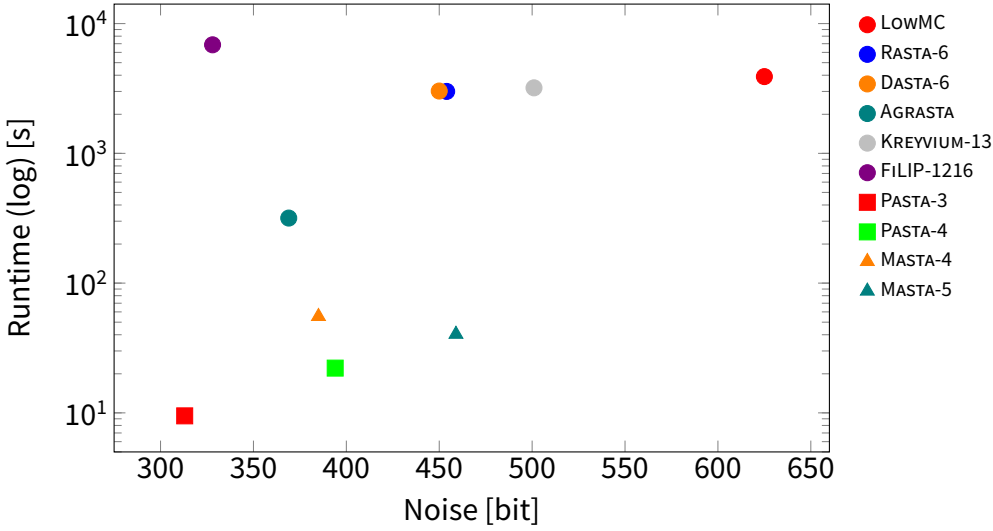
Use Cases

- $\mathbb{Z}_2$ ciphers:
  - Binary circuits for integer use cases
  - Large multiplicative depth!
- Example: Small linear regression
  - 5×5 Matrix-vector multiplication with 16-bit integers
  - Already shows inefficiency of $\mathbb{Z}_2$ ciphers

HHE with Small Linear Regression Use Case



HHE with Small Linear Regression Use Case



$\mathbb{F}_p$ ciphers

- PASTA (and MASTA) make larger integer use cases possible!
- Larger use case in paper:
  - 3 Affine layers (200 integers mod p)
  - Interleaved by elementwise squaring
  - Benchmarked in 3 primes fields (17-bit, 33-bit, 60-bit)
  - $\Rightarrow$ Can be interpreted as small 3-layer neural network

$\Rightarrow$ Benchmarks show: PASTA outperforms MASTA!

- PASTA runtime: 120 *s* for 17 and 33-bit prime
- More details in paper

Benchmark results

- Our benchmarks:
  - Binary use cases:
    - TFHE: Kreyvium
    - BGV/BFV: Agrasta or Dasta
  - BFV/BGV:
    - Pasta
- General results:
  - CKKS:
    - Rubato
  - Concrete:
    - Elisabeth

Benchmark results

- Our benchmarks:
  - Binary use cases:
    - TFHE: Kreyvium
    - BGV/BFV: Agrasta or Dasta
  - BFV/BGV:
    - Pasta
- General results:
  - CKKS:
    - Rubato
  - Concrete:
    - Elisabeth

Conclusion

- Hybrid Homomorphic Encryption:
  - Tradeoff: Client upload size vs server runtime
- Our paper:
  - First time investigation of tradeoffs on client side
  - Extensive benchmarks (inclusive use cases)
  - Pasta: Optimized HHE cipher over $\mathbb{F}_p$
  - https://eprint.iacr.org/2021/731.pdf
- Extensive benchmarks in different HE libraries including use cases
  - Framework: https://github.com/IAIK/hybrid-HE-framework

Questions

TU Graz

SCIENCE
PASSION
TECHNOLOGY

Pasta: A Case for Hybrid Homomorphic Encryption

Christoph Dobraunig, Lorenzo Grassi, Lukas Helminger, Christian Rechberger, Markus Schofnegger, **Roman Walch**

29.05.2022