PALISADE

HOMOMORPHIC ENCRYPTION FOR PALISADE USERS: TUTORIAL WITH APPLICATIONS

August 28, 2020

Yuriy Polyakov
David Bruce Cousins
contact@palisade-crypto.org

HOMOMORPHIC ENCRYPTION FOR PALISADE USERS

- Tutorial with applications consisting of 3 episodes (6 lectures)
- **Episode 1**
  - Introduction to Homomorphic Encryption
  - Boolean Arithmetic with Applications
- Episode 2
  - Integer Arithmetic
  - Applications of Homomorphic Encryption over Integers
- Episode 3
  - Approximate Number Arithmetic
  - Applications of Homomorphic Encryption over Approximate Numbers

HOMOMORPHIC ENCRYPTION FOR PALISADE USERS: TUTORIAL WITH APPLICATIONS

Introduction to Homomorphic Encryption

Yuriy Polyakov
ypolyakov@dualitytech.com

AGENDA

- Basics
  - What is homomorphic encryption?
  - Typical computations and examples of applications supported by HE
  - Main concepts
- Main approaches
  - Classes of homomorphic computations
    - Boolean circuit approach
    - Modular (exact) arithmetic approach
    - Approximate number approach
  - Setting security parameters

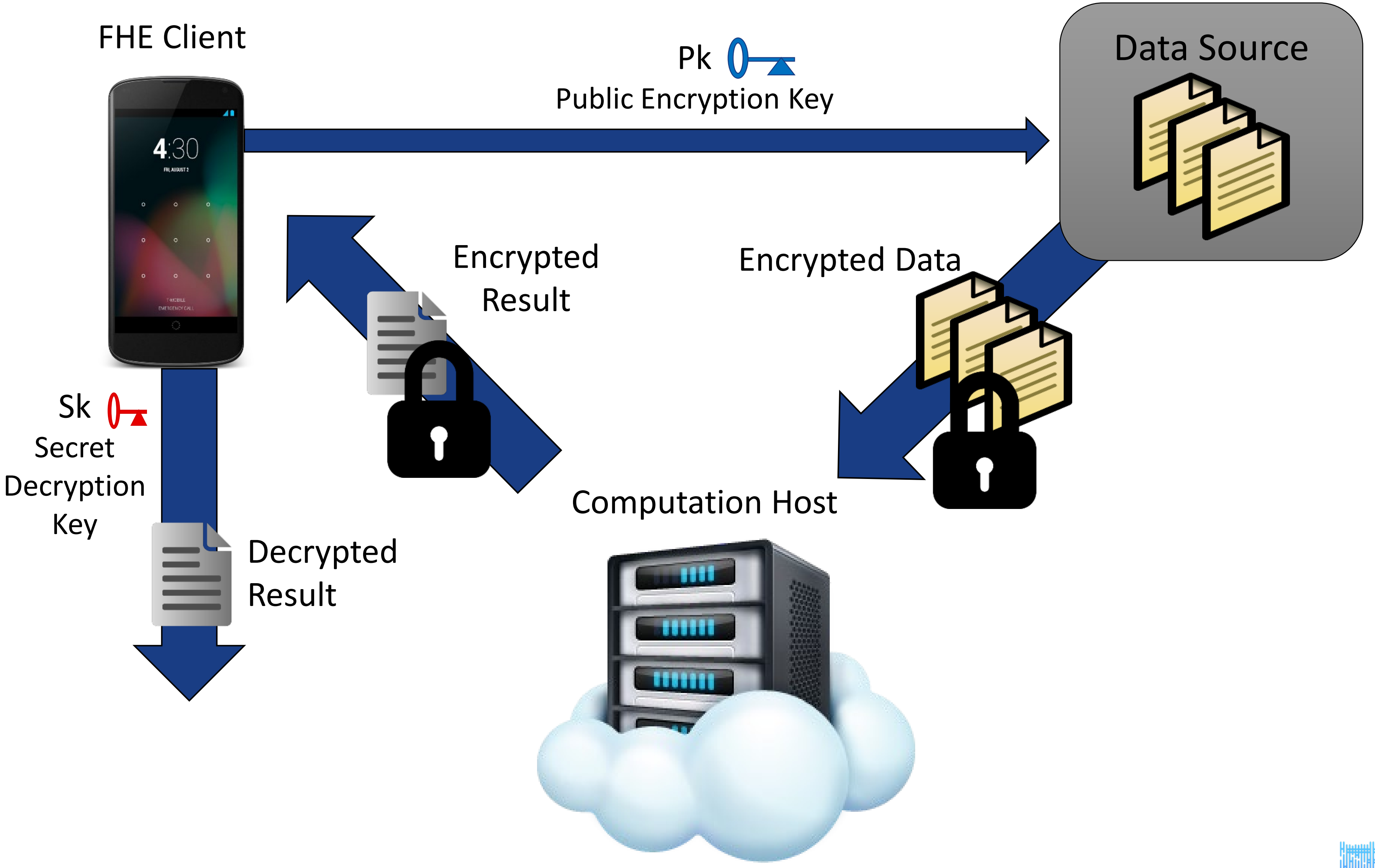
Basics

Introduces HE, typical computations, example applications, and main concepts

WHAT IS HOMOMORPHIC ENCRYPTION?

- Encryption protocol with one extra operation: Evaluation
  - Allows for computation on encrypted data
  - Enables outsourcing of data storage/processing
- How is HE related to symmetric and public key encryption?
  - HE schemes provide efficient instantiations of post-quantum public-key and symmetric-key encryption schemes
  - Homomorphic encryption can be viewed as a generalization of public key encryption
- Key milestones in the history of homomorphic encryption
  - Rivest, Adleman, Dertouzos (1978) -- “On Data Banks and Privacy Homomorphisms”
  - Gentry (2009) -- “A Fully Homomorphic Encryption Scheme”
  - Multiple HE schemes developed after 2009

EXAMPLE OF HE WORKFLOW



PALISADE

HE vs OTHER SECURE COMPUTING APPROACHES

	HE	MPC	SGX
Performance	Compute-bound	Network-bound	
Privacy	Encryption	Encryption / Non-collusion	Trusted Hardware
Non-interactive	✓	✗	✓
Cryptographic security	✓	✓	✗ (known attacks)

Hybrid approaches are also possible, e.g., MPC + HE

PALISADE

TYPICAL HE OPERATIONS

- Encrypt bits and perform logical AND, OR, XOR operations on the ciphertexts.
  - 0 AND 1 → 0, 0 OR 1 → 1, 1 XOR 1 → 0
- Encrypt small integers and perform addition and multiplication, as long as the result does not exceed some fixed bound, for instance, if the bound is 10000
  - 123 + 456 → 579, 12 * 432 → 5184, 35 * 537 → overflow
- Encrypt 8-bit unsigned integers (between 0 and 255) and perform addition and multiplication modulo 256
  - 128 + 128 → 0, 2 * 129 → 2
- Encrypt fixed-point numbers and perform addition and multiplication with the result rounded to a fixed precision, for instance, two digits after the decimal point
  - 12.42 + 1.34 → 13.76, 2.23 + 5.19 → 11.57
- Different homomorphic encryption schemes support different plaintext types and different operations on them.

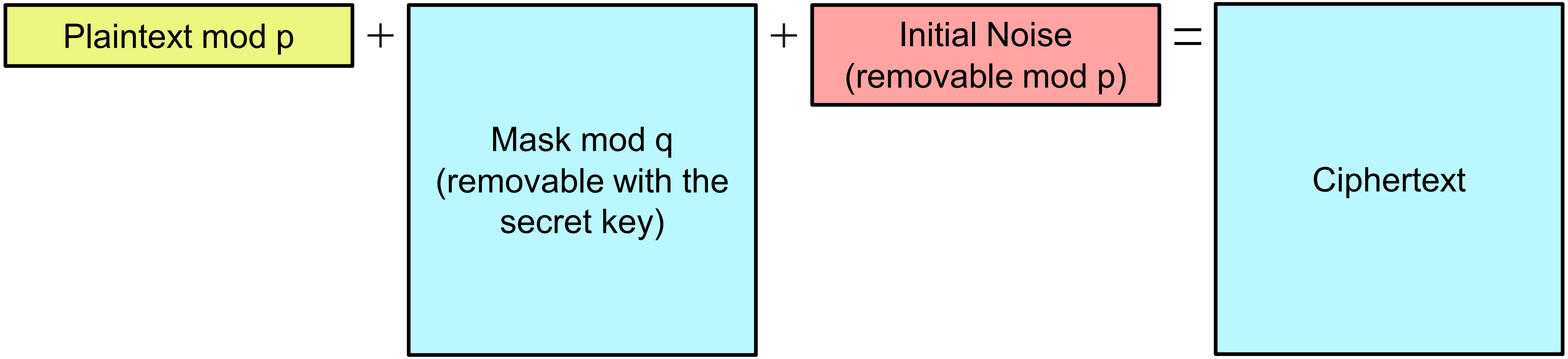
SOME EXAMPLES OF REAL-SCALE HE APPLICATIONS

- Private information retrieval
  - https://eprint.iacr.org/2017/1142, IEEE S&P 2018
- Private set intersection
  - https://eprint.iacr.org/2017/299, ACM CCS 2017
- Genome-wide association studies based on chi-square test and logistic regression training
  - https://eprint.iacr.org/2020/563, PNAS 2020
- Logistic regression training
  - https://eprint.iacr.org/2018/662, AAAI Conference on AI 2019

MAIN CONCEPTS

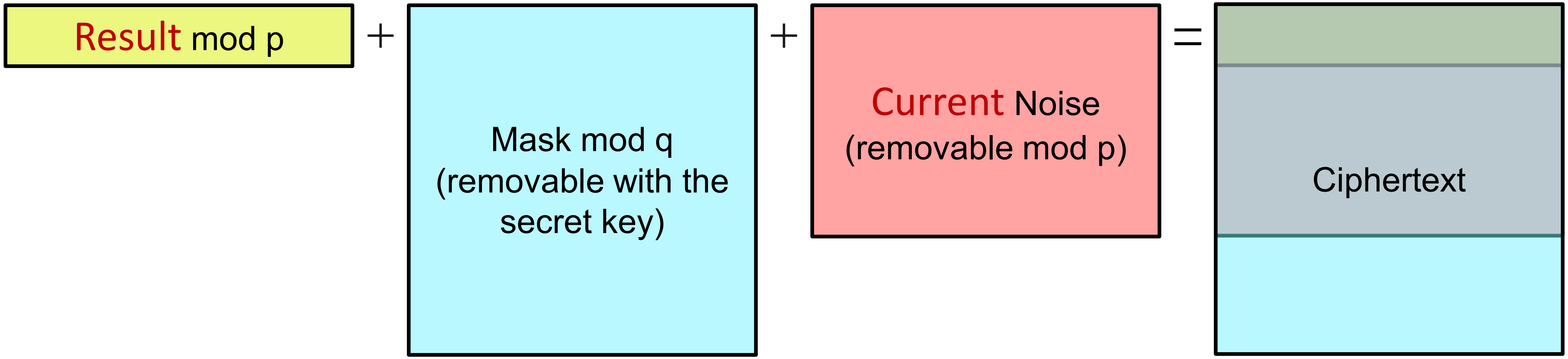
- *Homomorphic*: a (secret) mapping from plaintext space to ciphertext space that preserves arithmetic operations.
- *Mathematical Hardness: (Ring) Learning with Errors Assumption*
  - Every image (ciphertext) of this mapping looks uniformly random in range (ciphertext space).
- *Security level*: the hardness of inverting this mapping without the secret key
  - Often estimated as a work factor.
    - Example: 128 bits → 2^{128} operations to break using best known lattice attack
- *Plaintext*: Elements and operations of a polynomial ring (mod $x^n + 1$, mod p).
  - Example: $3x^5 + x^4 + 2x^3 + \ldots$
  - For all practical purposes, you can think of it as a vector of (small) finite integers
- *Ciphertext*: elements and operations of a polynomial ring (mod $x^n + 1$, mod q).
  - Example: $7862x^5 + 5652x^4 + \ldots$
  - For all practical purposes, you can think of it as a vector of (larger) finite integers
- *Noise*: random integers with Gaussian distribution, which are "added" to the plaintext to achieve the desired security level based on Ring Learning With Errors

FRESH ENCRYPTION



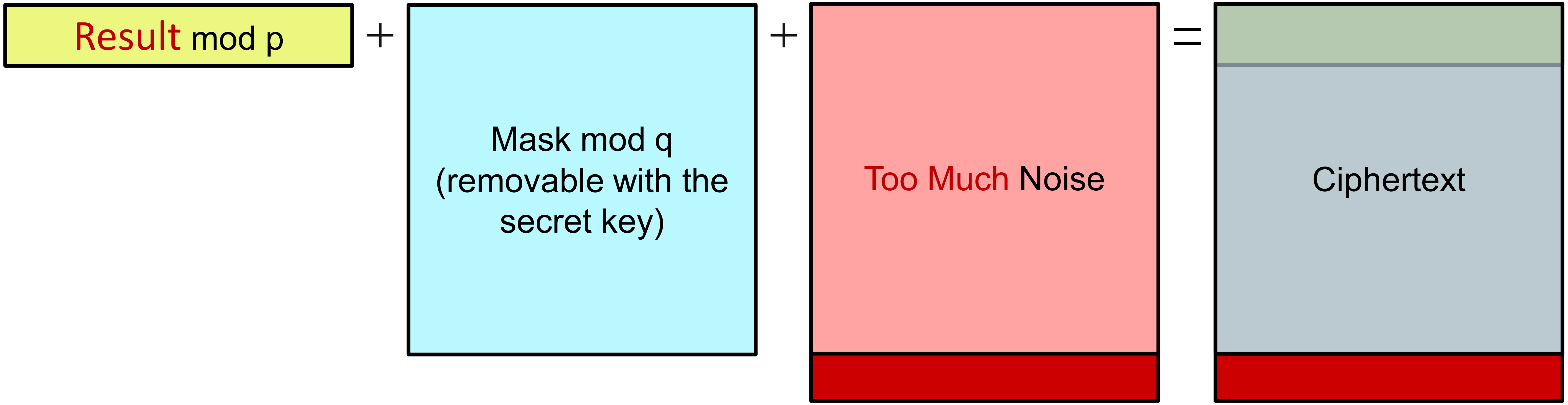
- Horizontal: each coefficient in a polynomial or in a vector.
- Vertical: size of coefficients.
- Initial noise is small in terms of coefficients' size.

AFTER SOME COMPUTATIONS



- Horizontal: each coefficient in a polynomial or in a vector.
- Vertical: size of coefficients.
- Initial noise is small in terms of coefficients' size.

NOISE OVERFLOW (RESULTS IN DECRYPTION FALURE)

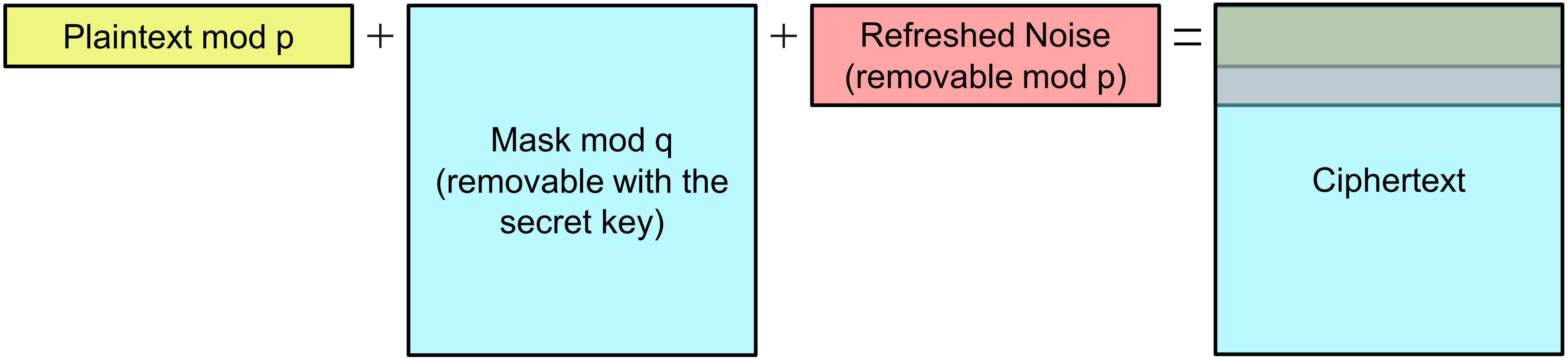


- Horizontal: each coefficient in a polynomial or in a vector.
- Vertical: size of coefficients.
- Initial noise is small in terms of coefficients' size.

PALISADE

BOOTSTRAPPING (NOISE REFRESHING PROCEDURE)

Evaluates the decryption circuit homomorphically and resets the noise.



- Horizontal: each coefficient in a polynomial or in a vector.
- Vertical: size of coefficients.
- Initial noise is small in terms of coefficients' size.

TYPES OF HOMOMORPHIC ENCRYPTION

- Partially homomorphic encryption (weakest notion)
  - supports only one type of operation, e.g. addition or multiplication.
- Somewhat homomorphic encryption schemes
  - can evaluate two types of gates/operations, but only for a subset of circuits.
- **Leveled fully homomorphic encryption**
  - supports more than one operation but only computations of a predetermined size (typically multiplicative depth); supports much deeper circuits than somewhat homomorphic encryption
- **Fully homomorphic encryption**
  - supports arbitrary computation on encrypted data, and is the strongest notion of homomorphic encryption.

Main approaches

Introduces classes of homomorphic computations and security parameters

CLASSES OF HOMOMORPHIC COMPUTATIONS

It is important to choose the right approach for your HE computation:

1. Boolean Circuits
   - Plaintext data represented as **bits**
   - Computations expressed as **Boolean circuits**
2. Modular (Exact) Arithmetic
   - Plaintext data represented as **integers modulo a plaintext modulus "*p*"** (or their vectors)
   - Computations expressed as **integer arithmetic mod *p***
3. Approximate Number Arithmetic
   - Plaintext data represented as **real numbers** (or complex numbers)
   - Compute model similar to **floating-point arithmetic** but dealing with fixed-point numbers

BOOLEAN CIRCUITS APPROACH

- Features:
  - Fast number comparison
  - Supports arbitrary Boolean circuits
  - Fast bootstrapping (noise refreshing procedure)

- Selected schemes:
  - Gentry-Sahai-Waters (GSW) [GSW13] - foundation for other schemes
  - Fastest Homomorphic Encryption in the West (FHEW) [DM15]
  - Fast Fully Homomorphic Encryption over the Torus (TFHE) [CGGI16,CGGI17]

MODULAR (EXACT) ARITHMETIC APPROACH

- Features:
  - Efficient SIMD computations over vectors of integers (using batching)
  - Fast high-precision integer arithmetic
  - Fast private information retrieval/private set intersection
  - Leveled design (often used without bootstrapping)

- Selected schemes:
  - Brakerski-Vaikuntanathan (BV) [BV11] - foundation for other schemes
  - Brakerski-Gentry-Vaikuntanathan (BGV) [BGV12, GHS12]
  - Brakerski/Fan-Vercauteren (BFV) [Brakerski12, FV12, BEHZ16, HPS18]

APPROXIMATE NUMBER ARITHMETIC APPROACH

- Features:
  - Efficient SIMD computations over vectors of real numbers (using batching)
  - Fast polynomial approximation
  - Relatively fast multiplicative inverse and Discrete Fourier Transform
  - Deep approximate computations, such as logistic regression learning
  - Leveled design (often used without bootstrapping)

- Selected schemes:
  - Cheon-Kim-Kim-Song (CKKS) [CKKS17]

SCHEMES SUPPORTED BY PALISADE

Library/ Scheme or Extension	BGV	BFV	CKKS	FHEW	TFHE	Threshold FHE (MP)	Proxy Re-Encryption (MP)
FHEW				✓			
HEAAN/HEAAN-RNS			✓				
HELib	✓		✓				
Lattigo		✓	✓			✓	
PALISADE	✓	✓	✓	✓	✓	✓	✓
SEAL		✓	✓				
TFHE					✓		

PALISADE

SELECTING SECURITY PARAMETERS

The ciphertext dimension (degree of polynomial) should be chosen according to the security tables published at HomomorphicEncryption.org (PALISADE selects it automatically).

distribution	n	security level	logq	uSVP	dec	dual
(-1, 1)	1024	128	27	131.6	160.2	138.7
		192	19	193.0	259.5	207.7
		256	14	265.6	406.4	293.8
	2048	128	54	129.7	144.4	134.2
		192	37	197.5	233.0	207.8
		256	29	259.1	321.7	273.5
	4096	128	109	128.1	134.9	129.9
		192	75	194.7	212.2	198.5
		256	58	260.4	292.6	270.1
	8192	128	218	128.5	131.5	129.2
		192	152	192.2	200.4	194.6
		256	118	256.7	273.0	260.6

THANK YOU

ypolyakov@dualitytech.com