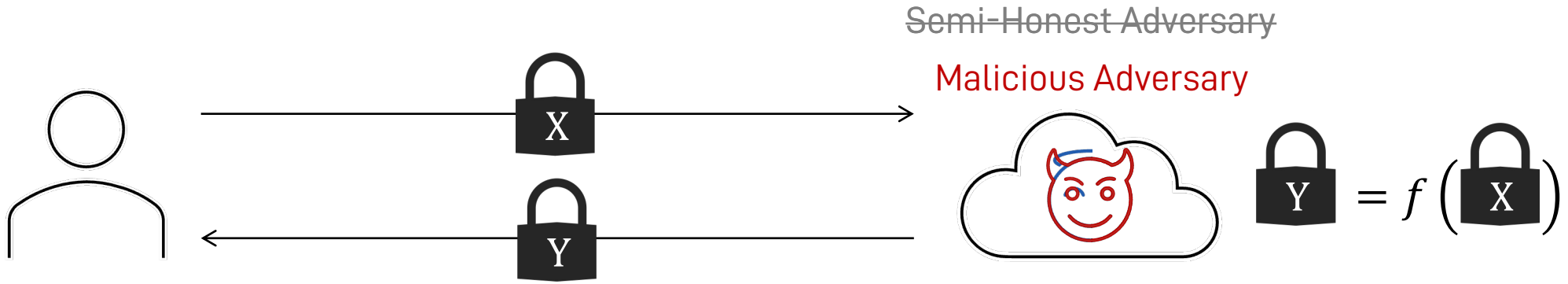
Verifiable Fully Homomorphic Encryption

Christian Knabenhans*, Alexander Viand*, Anwar Hithnawi

ETH zürich

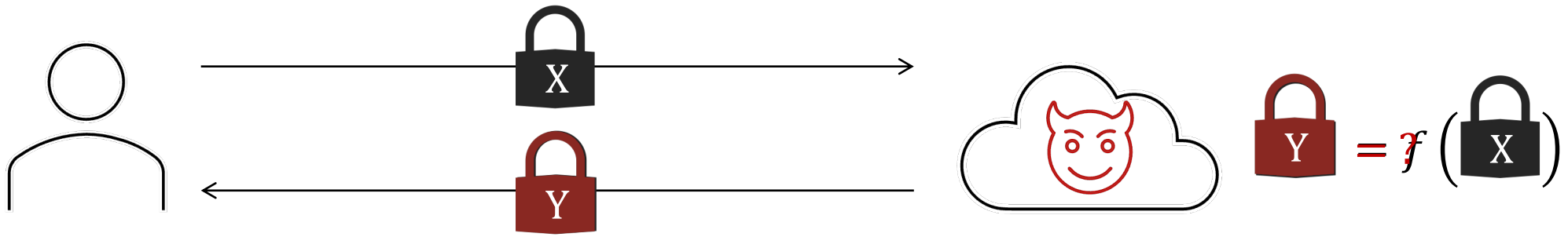
*equal contribution

A stronger adversarial model for FHE



No trusted server	Breaches	Programming errors	Monetary incentive

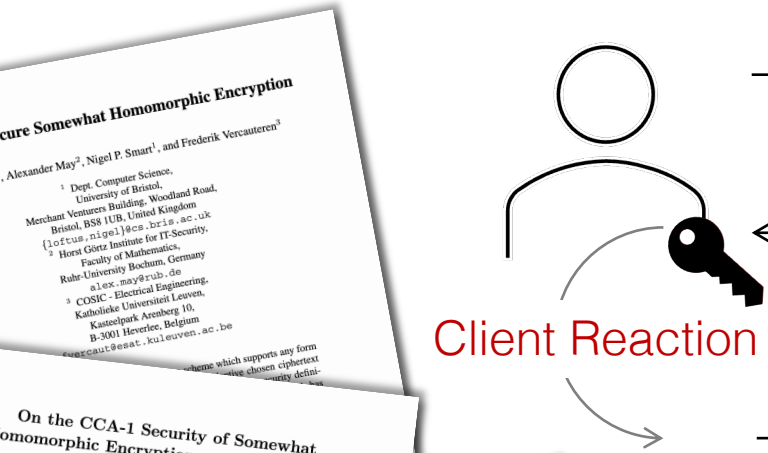
FHE correctness in malicious setting

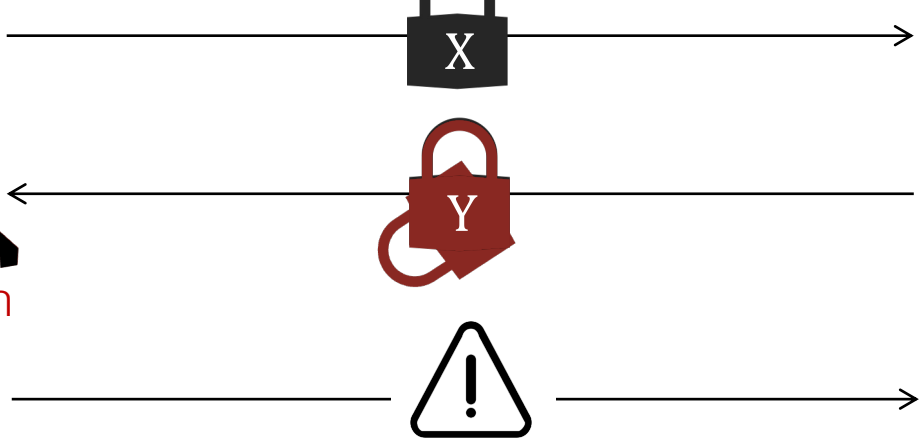


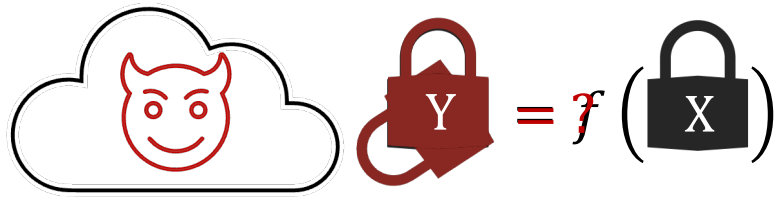
No robust correctness

FHE confidentiality in malicious setting

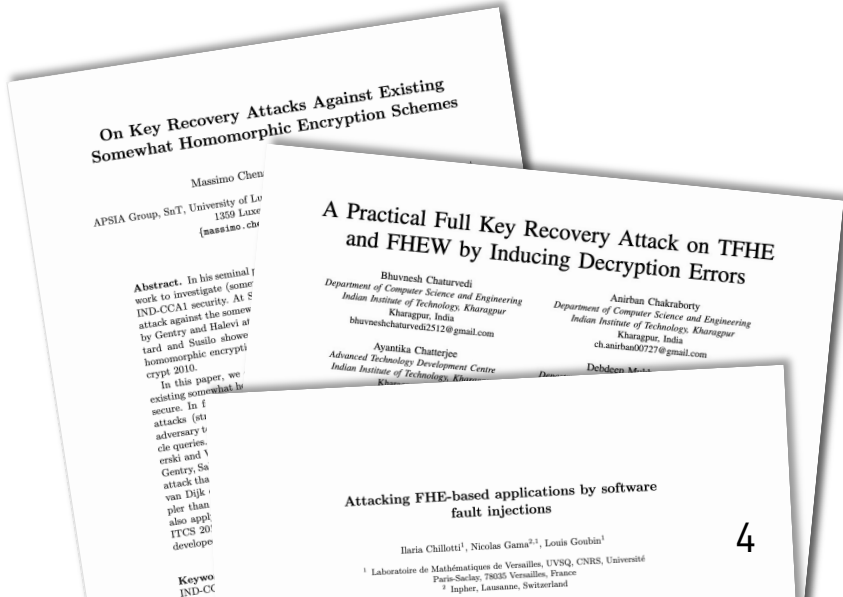
...







No robust confidentiality



Verifiable Fully Homomorphic Encryption

Alexander Viand*, Christian Knabenhans*, Anwar Hithnawi

ETH Zurich

***Abstract*—Fully Homomorphic Encryption (FHE) is seeing increasing real-world deployment to protect data in use by allowing computation over encrypted data. However, the same malleability that enables homomorphic computations also raises *integrity* issues, which have so far been mostly overlooked. While FHE's lack of integrity has obvious implications for correctness, it also has severe implications for confidentiality: a malicious server can leverage the lack of integrity to carry out interactive key-recovery attacks. As a result, virtually all FHE schemes and applications assume an honest-but-curious server who does not deviate from the protocol. In practice, however, this assumption is insufficient for a wide range of deployment scenarios. While there has been work that aims to address this gap, these have remained isolated efforts considering only aspects of the overall problem and fail to fully address the needs and characteristics of modern FHE schemes and applications. In this paper, we analyze existing FHE integrity approaches, present attacks that exploit gaps in prior work, and propose a new notion for maliciously-secure verifiable FHE. We then instantiate this new notion with a range of techniques, analyzing them and evaluating their performance in a range of different settings. We highlight their potential but also show where future work on tailored integrity solutions for FHE is still required.**

1. Introduction

Fully Homomorphic Encryption (FHE), which enables

Honest-but-Curious Assumption. Historically, the FHE research community has extensively made use of the assumption that the server running an FHE application would be honest-but-curious, rather than actively malicious [13]–[16]. This assumption may be reasonable in some deployment scenarios (e.g., when FHE is used only to ensure regulatory compliance or when dealing with trusted institutions cooperating on their own data). However, the necessity to trust the server to this extent is very limiting to the scope of application scenarios, since a violation of the assumption threatens not only correctness but also confidentiality. In addition, even otherwise trusted parties can be compromised by malicious third parties, exposing this attack surface. While FHE protects against passive attacks, a malicious or compromised server taking part in an FHE application can leverage this to undermine data confidentiality. A class of attacks known as *key-recovery attacks* exploits the interactive nature of real-world deployments to construct (partial) decryption oracles. These exploit the fact that a server can craft a ciphertext that fails to decrypt correctly for certain secret keys, using the client's reaction or lack thereof as an oracle. Practical key-recovery attacks have been developed for all major FHE schemes [8]–[12]. Therefore, there is an urgent need to strengthen FHE to maintain strong guarantees in the context of these attacks.

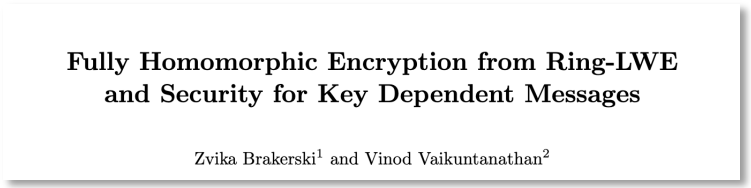
Existing FHE Integrity Approaches. In order to remediate these attacks, a line of research has emerged that constructs more robust FHE schemes [17]–[24] that achieve indistinguishability against chosen ciphertext attacks (IND-CCA1).

What do we mean with “FHE”?

Formal FHE → Early FHE → Modern FHE

An FHE scheme is a tuple of PPT algorithms $(KGen, Enc, Eval, Dec)$ *such that …*

BV scheme

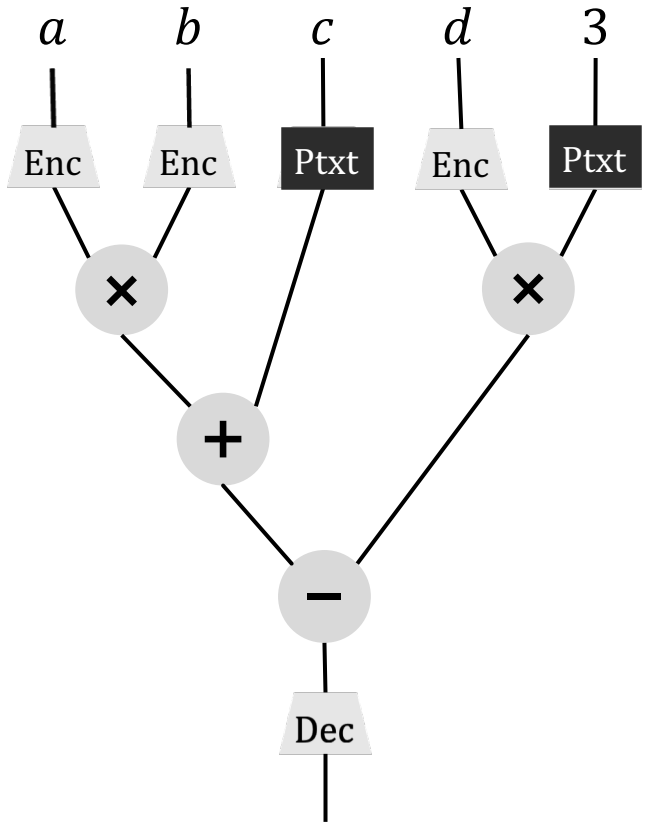


Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages

Zvika Brakerski[1] and Vinod Vaikuntanathan[2]

B/FV, BGV, TFHE, FHEW, CKKS

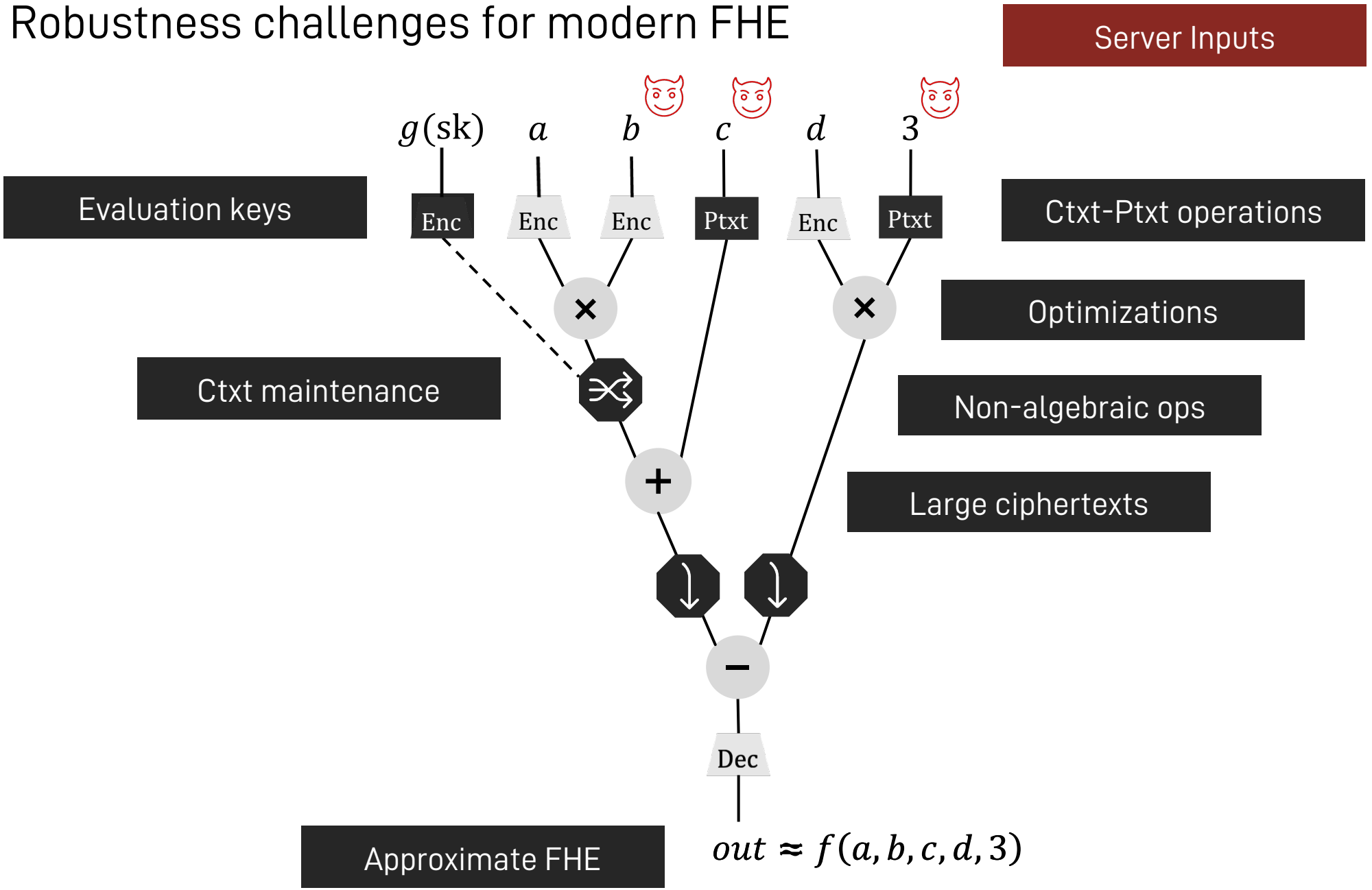
Robustness challenges for modern FHE

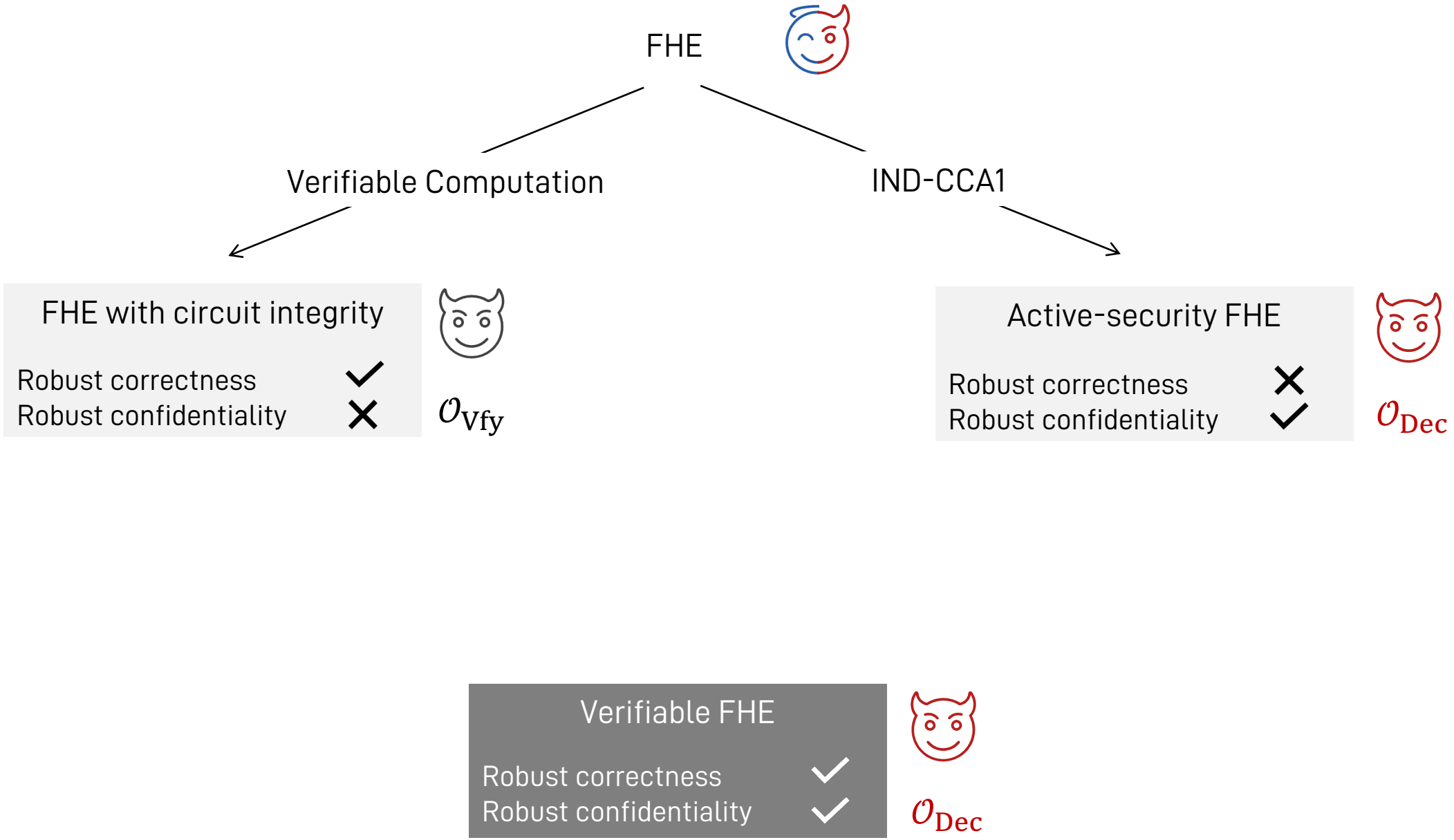


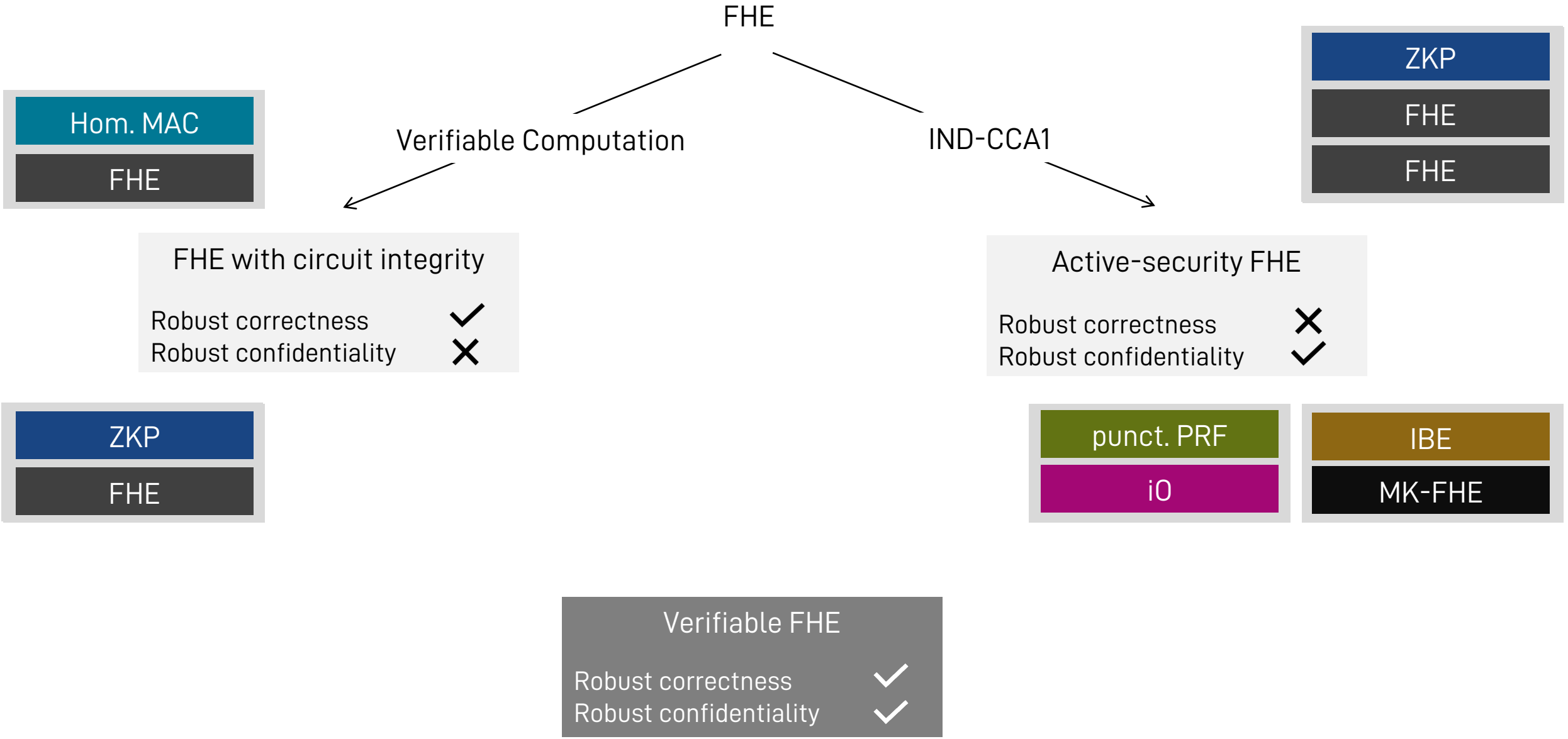
Ctxt-Ptxt operations

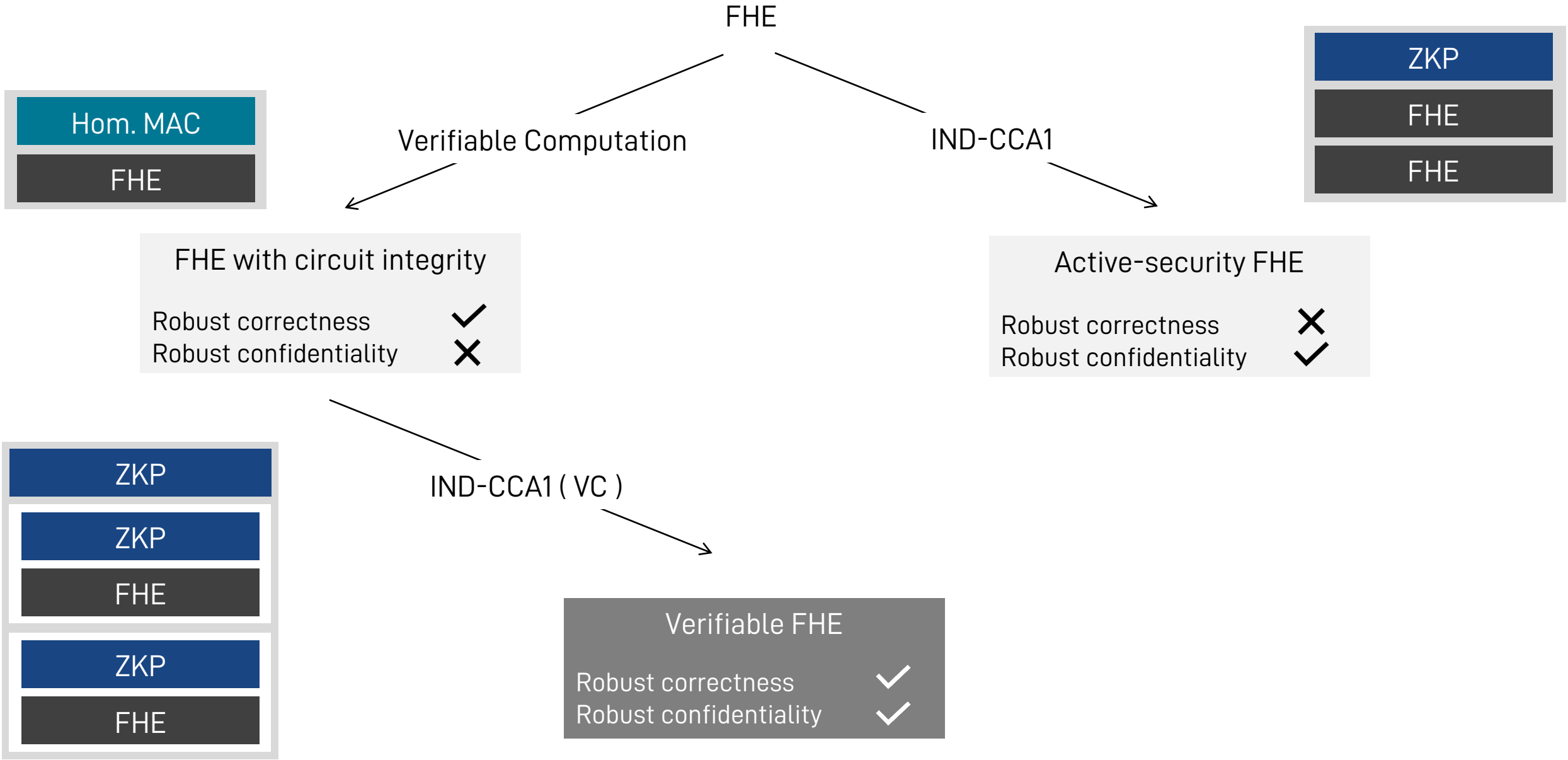
$$out = f(a, b, c, d, 3)$$

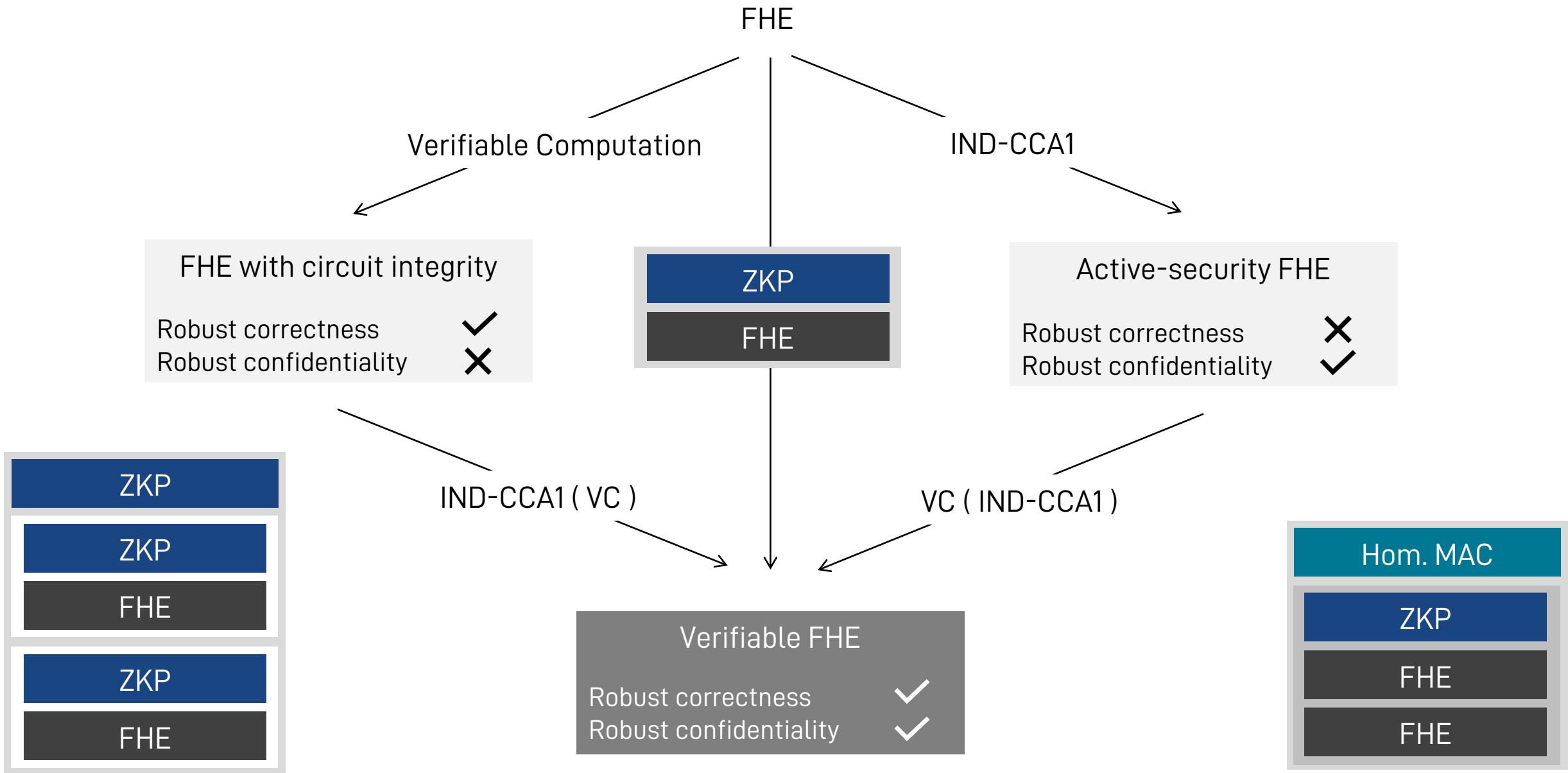
Robustness challenges for modern FHE




FHE
Verifiable Computation
IND-CCA1
FHE with circuit integrity
Robust correctness
Robust confidentiality
$\mathcal{O}_{\mathrm{Vfy}}$
Active-security FHE
Robust correctness
Robust confidentiality
$\mathcal{O}_{\mathrm{Dec}}$
Verifiable FHE
Robust correctness
Robust confidentiality
$\mathcal{O}_{\mathrm{Dec}}$


FHE
Verifiable Computation
IND-CCA1
Hom. MAC
FHE
ZKP
FHE
FHE
FHE with circuit integrity
Robust correctness
Robust confidentiality
Active-security FHE
Robust correctness
Robust confidentiality
ZKP
FHE
punct. PRF
iO
IBE
MK-FHE
Verifiable FHE
Robust correctness
Robust confidentiality


FHE
Verifiable Computation
IND-CCA1
Hom. MAC
FHE
ZKP
FHE
FHE
FHE with circuit integrity
Robust correctness
Robust confidentiality
Active-security FHE
Robust correctness
Robust confidentiality
IND-CCA1 (VC)
ZKP
ZKP
FHE
ZKP
FHE
Verifiable FHE
Robust correctness
Robust confidentiality


FHE
Verifiable Computation
IND-CCA1
FHE with circuit integrity
Robust correctness
Robust confidentiality
ZKP
FHE
Active-security FHE
Robust correctness
Robust confidentiality
ZKP
FHE
ZKP
FHE
IND-CCA1 (VC)
VC (IND-CCA1)
Verifiable FHE
Robust correctness
Robust confidentiality
Hom. MAC
ZKP
FHE
FHE

Starting from IND-CPA^D

IND-CPA FHE

↓

Exact FHE
or +Diff. Priv.

↓

IND-CPA^D FHE

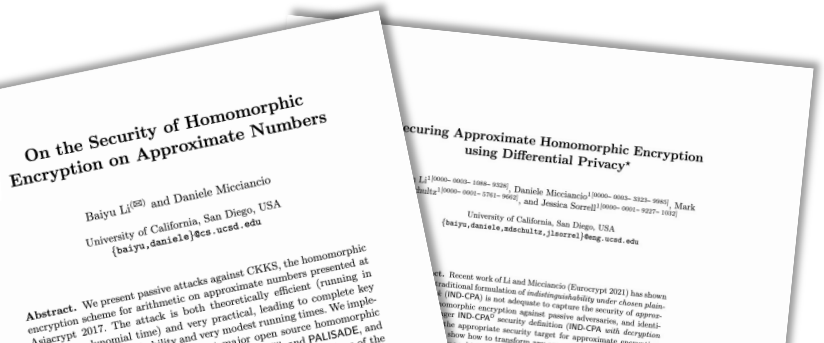
↓

ZKP

FHE

↓

Verifiable FHE



On the Security of Homomorphic Encryption on Approximate Numbers

Baiyu Li(✉) and Daniele Micciancio

University of California, San Diego, USA
{baiyu,daniele}@cs.ucsd.edu

Abstract. We present passive attacks against CKKS, the homomorphic encryption scheme for arithmetic on approximate numbers presented at Asiacrypt 2017. The attack is both theoretically efficient (running in polynomial time) and very practical, leading to complete key ...

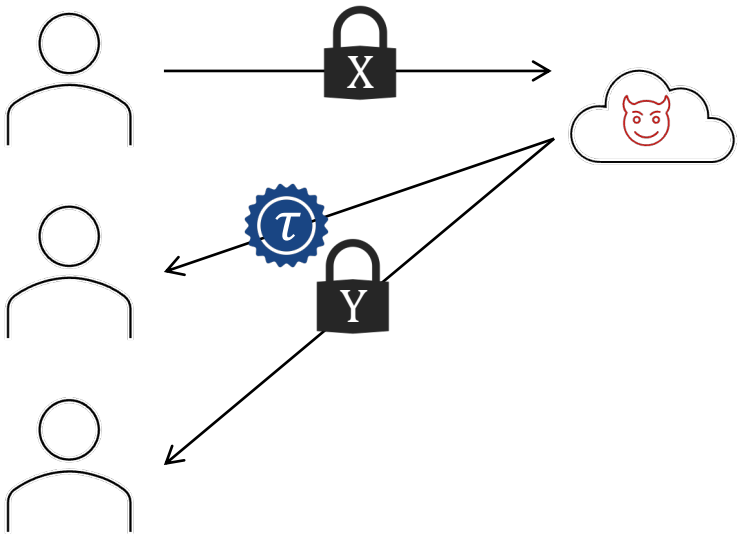
ecuring Approximate Homomorphic Encryption using Differential Privacy*

... Daniele Micciancio ... and Jessica Sorrell ... Mark

University of California, San Diego, USA
{baiyu,daniele,mdschultz,jlsorrel}@eng.ucsd.edu

Abstract. Recent work of Li and Micciancio (Eurocrypt 2021) has shown ... traditional formulation of *indistinguishability under chosen plaintext attack* (IND-CPA) is not adequate to capture the security of approximate homomorphic encryption against passive adversaries, and identify ... IND-CPA^D security definition (*IND-CPA with decryption errors*) ... the appropriate security target for approximate ... show how to transform ...

Simplified setup



Starting from IND-CPA^D

IND-CPA FHE

↓

Exact FHE
or +Diff. Priv.

↓

IND-CPA^D FHE

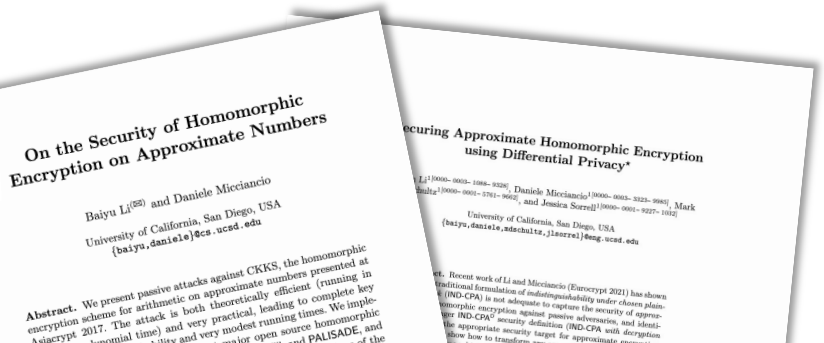
↓

ZKP

FHE

↓

Verifiable FHE



On the Security of Homomorphic Encryption on Approximate Numbers

Baiyu Li(✉) and Daniele Micciancio

University of California, San Diego, USA
{baiyu,daniele}@cs.ucsd.edu

Abstract. We present passive attacks against CKKS, the homomorphic encryption scheme for arithmetic on approximate numbers presented at Asiacrypt 2017. The attack is both theoretically efficient (running in polynomial time) and very practical, leading to complete key ... very and very modest running times. We imple- ... major open source homomorphic ... and PALISADE, and the ...

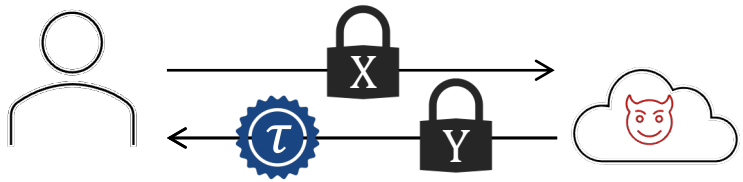
ecuring Approximate Homomorphic Encryption using Differential Privacy*

... Daniele Micciancio[0000-0003-3323-9985], Mark ... and Jessica Sorrell[0000-0001-9227-1012]

University of California, San Diego, USA
{baiyu,daniele,mdschultz,jlsorrel}@eng.ucsd.edu

ct. Recent work of Li and Micciancio (Eurocrypt 2021) has shown ... traditional formulation of *indistinguishability under chosen plain-* ... (IND-CPA) is not adequate to capture the security of approx- ... omorphic encryption against passive adversaries, and identi- ... IND-CPA^D security definition (*IND-CPA with decryption* ... the appropriate security target for approximate ... show how to transform ...

Simplified setup



Plaintext server inputs only

$$\tau, \boxed{Y} = f(\boxed{X}, \boxed{W})$$

Generic vFHE construction

KGen

Generate FHE keys (pk, sk)
Generate keys for ZK proof system

Enc

x → $c = \mathrm{Enc}_{pk}(x)$

Eval

c_1
w

→ $c = \mathrm{Eval}_{ek}(F, c_1, \dots, c_k, w)$

$\tau = \mathrm{Prove}(c{=}\mathrm{Eval}_{ek}(F, c_1, \dots, c_k, w)$
$\wedge\ w$ is valid plaintext)

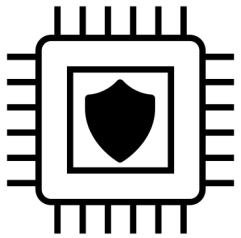
Verify
Decrypt

c
τ

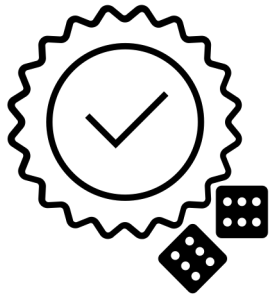
$\mathrm{Verify}(\tau, c) = \mathrm{true}$ →

$y = \mathrm{Dec}_{sk}(c)$

Proof System → Homomorphic MACs

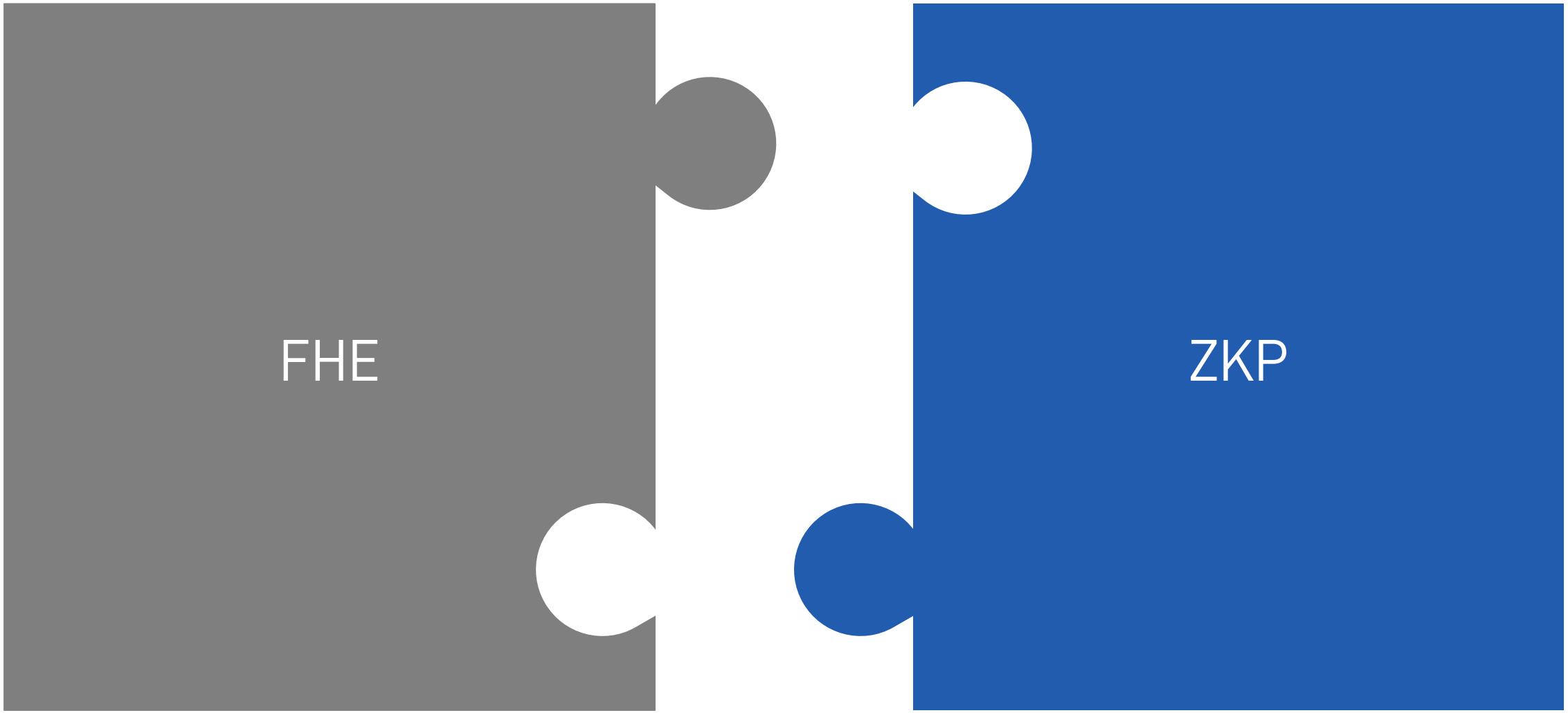


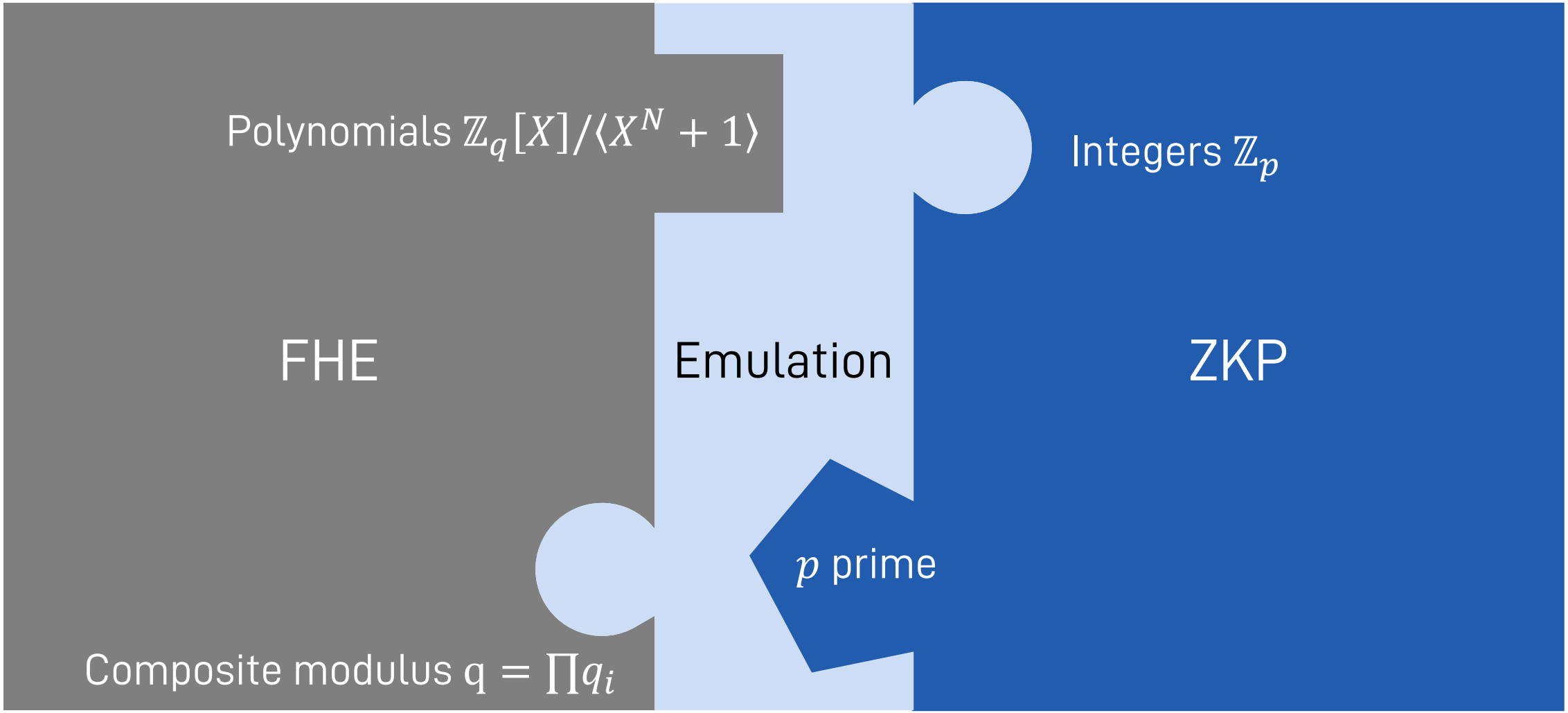
Trusted Execution Environment (TEE)

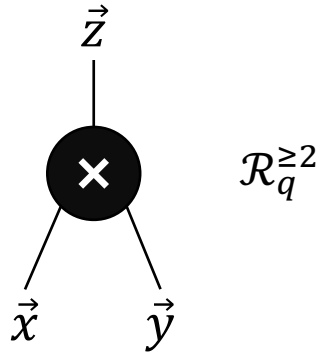


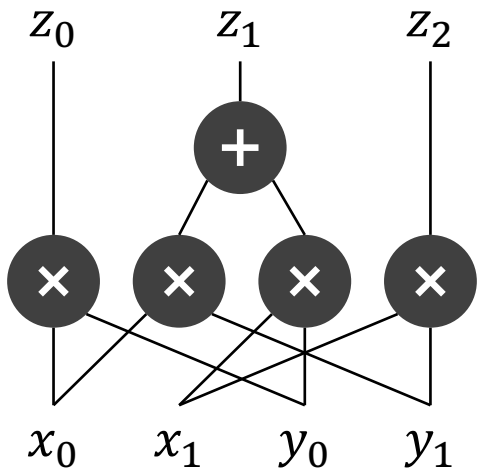
Cryptographic Proof

- Field-based
- Ring-based


FHE
ZKP


Polynomials $\mathbb{Z}_q[X]/\langle X^N + 1\rangle$
Integers $\mathbb{Z}_p$
FHE
Emulation
ZKP
p prime
Composite modulus $q = \prod q_i$

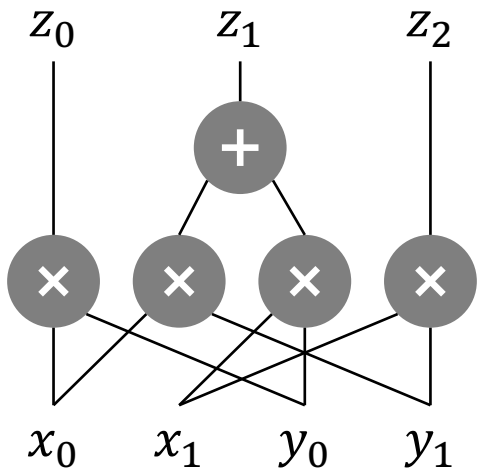

$\vec{z}$
$\vec{x}$
$\vec{y}$
$\mathcal{R}_q^{\geq 2}$



$\mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$

Overhead ×4

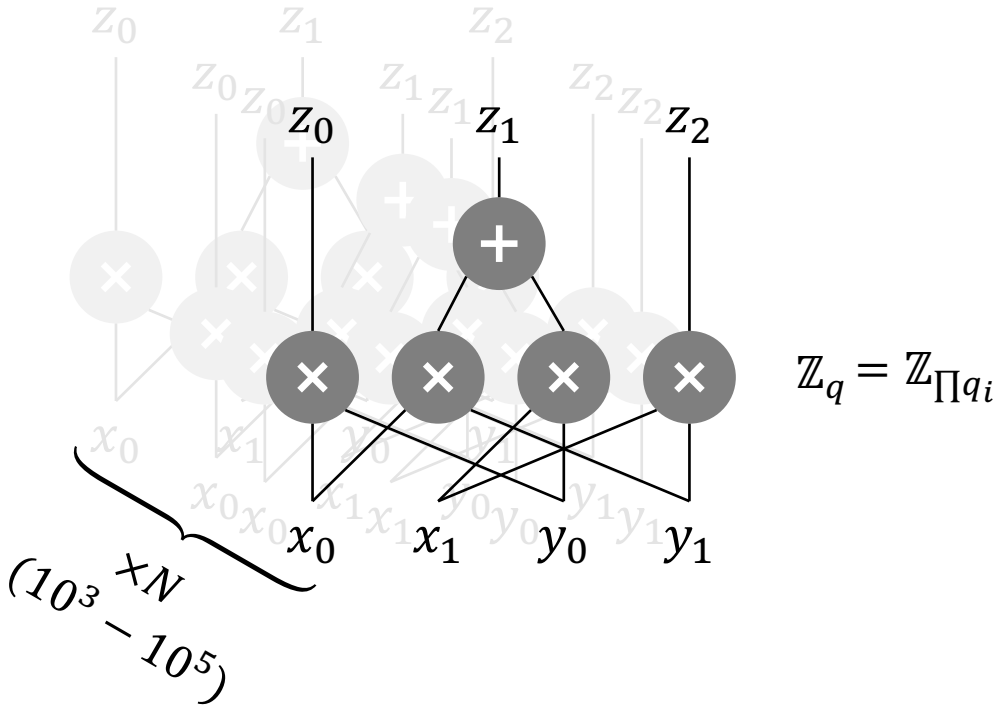
$\mathcal{R}_q^{\geq 2}$ → $\mathcal{R}_q$ → $\mathbb{Z}_q(\times N)$ → $\mathbb{F}$



$\mathbb{Z}_q^N \cong \mathbb{Z}_q[X]/\langle X^N + 1\rangle$

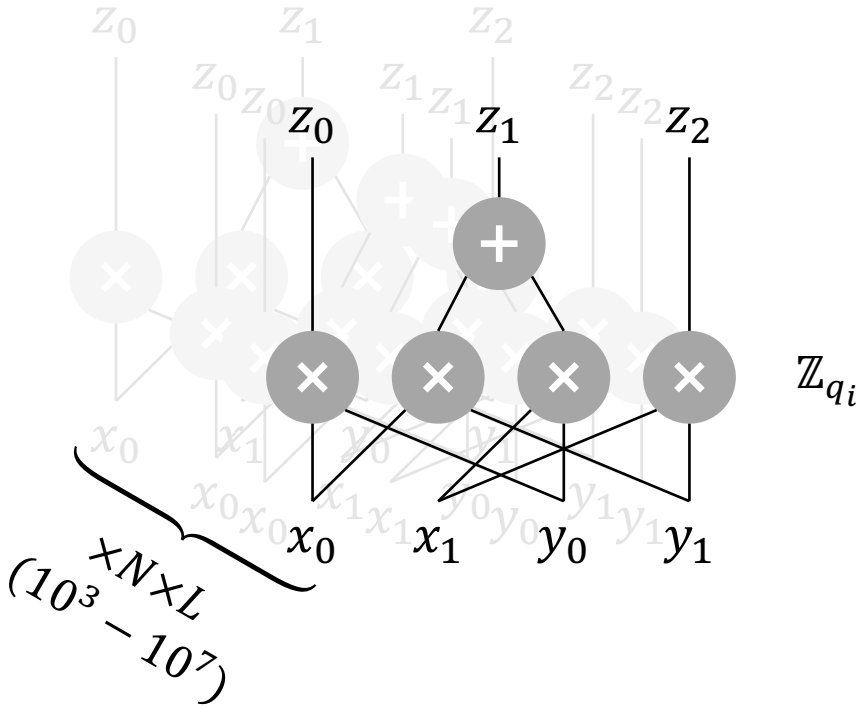
Overhead ×4

$\mathcal{R}_q^{\geq 2}$ → $\mathcal{R}_q$ → $\mathbb{Z}_q(\times N)$ → $\mathbb{F}$



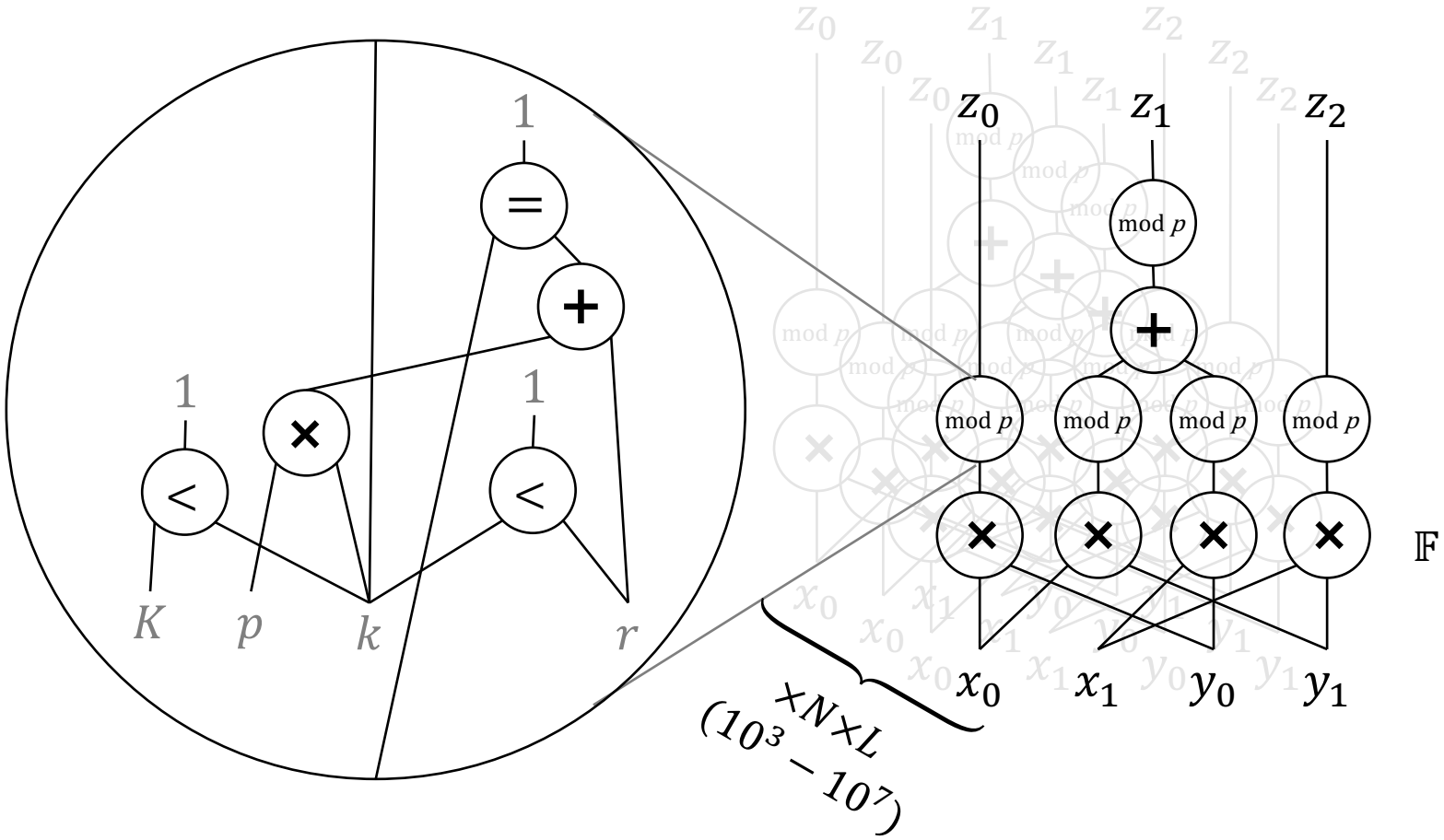
Overhead ×32'000

$\mathcal{R}_q^{\geq 2}$ → $\mathcal{R}_q$ → $\mathbb{Z}_q(\times N)$ → $\mathbb{Z}_{q_i}(\times N \times L)$ → $\mathbb{F}$



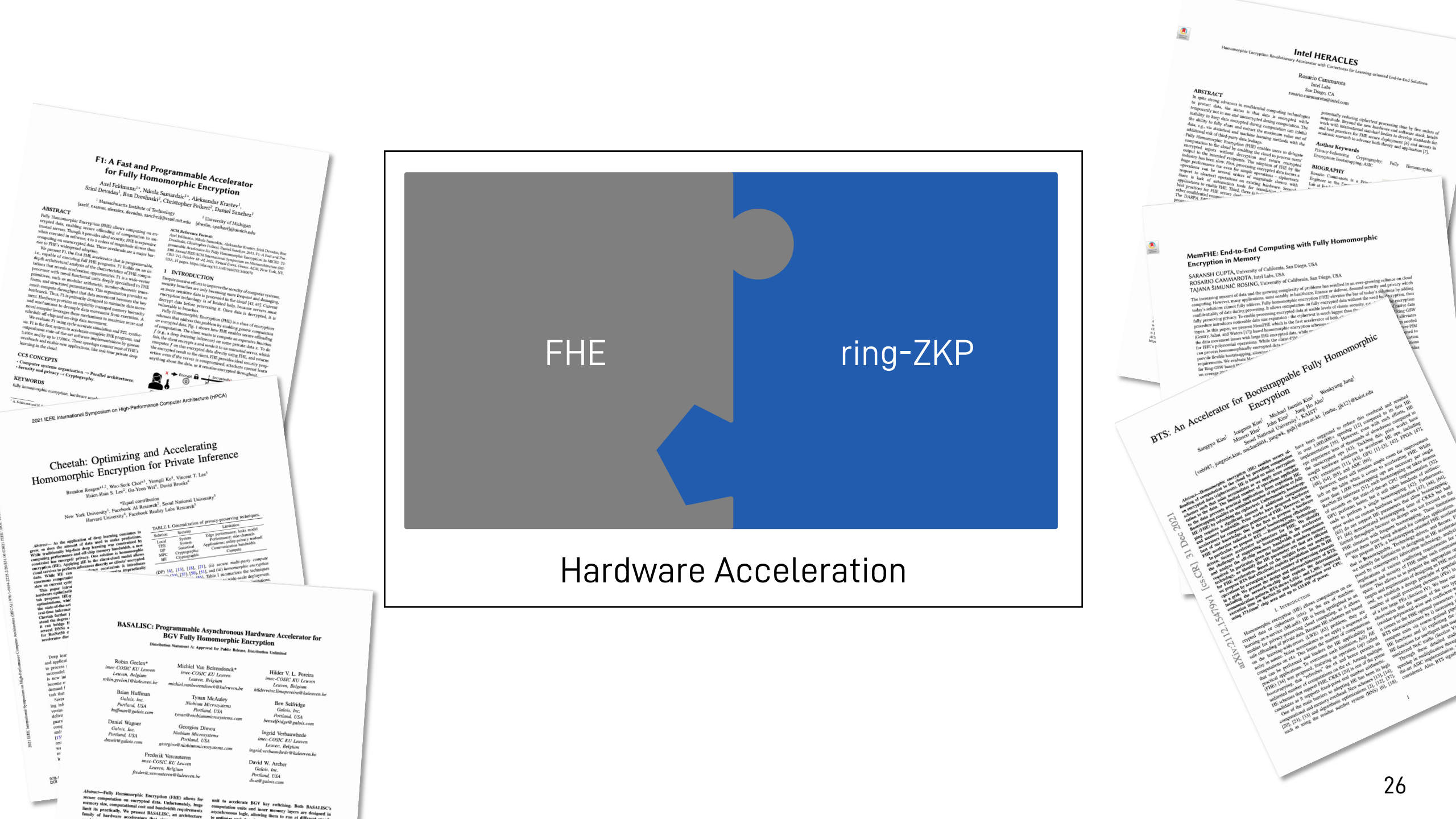
Overhead ×130'000

$\mathcal{R}_q^{\geq 2}$ → $\mathcal{R}_q$ → $\mathbb{Z}_q(\times N)$ → $\mathbb{Z}_{q_i}(\times N \times L)$ → $\mathbb{F}$

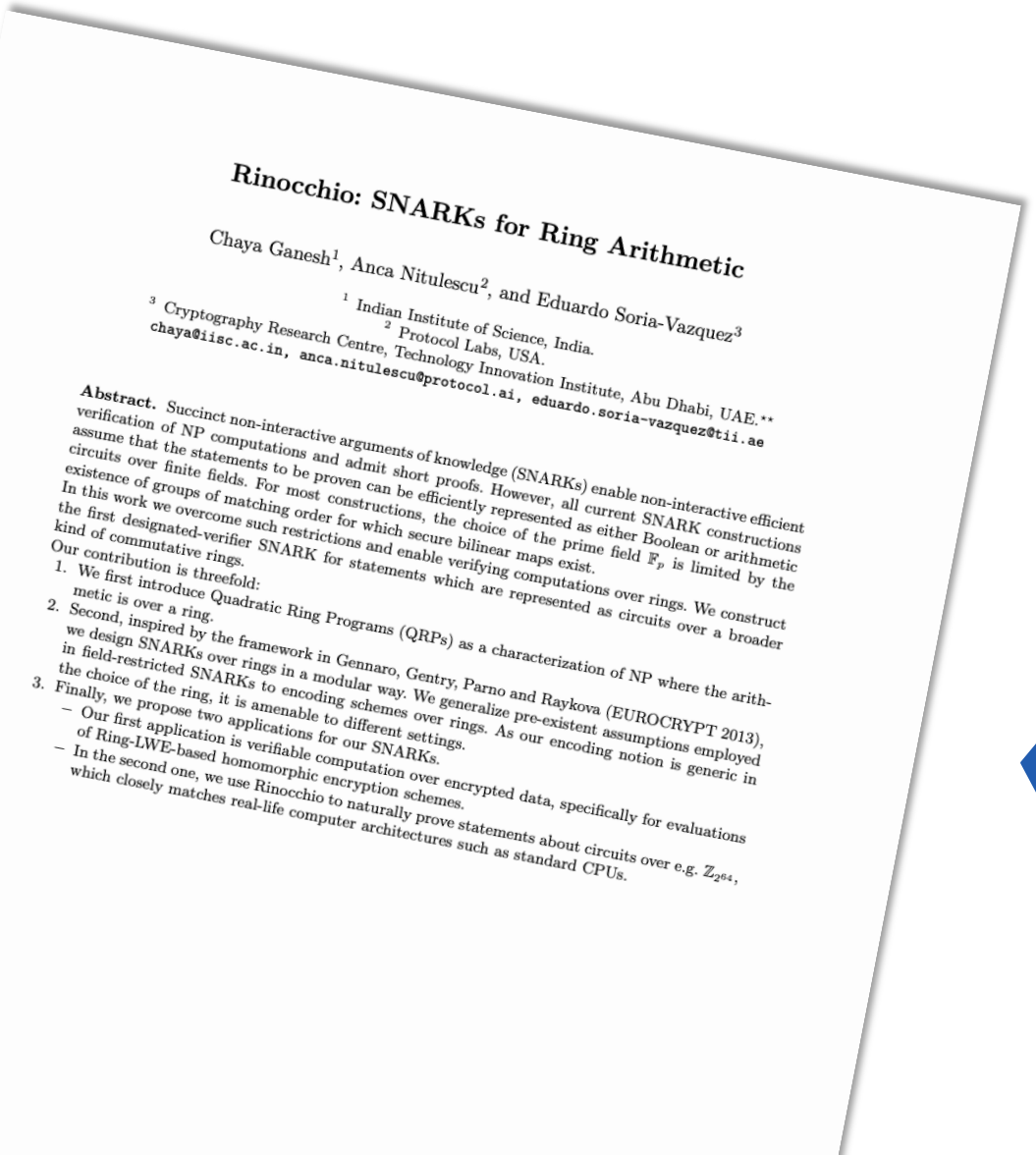


Overhead ×16'000'000

A better fit: ring-based ZKPs


F1: A Fast and Programmable Accelerator for Fully Homomorphic Encryption
Cheetah: Optimizing and Accelerating Homomorphic Encryption for Private Inference
BASALISC: Programmable Asynchronous Hardware Accelerator for BGV Fully Homomorphic Encryption
Intel HERACLES
MemFHE: End-to-End Computing with Fully Homomorphic Encryption in Memory
BTS: An Accelerator for Bootstrappable Fully Homomorphic Encryption
FHE
ring-ZKP
Hardware Acceleration

State-of-the-art ring-ZKP: Rinocchio



Rinocchio: SNARKs for Ring Arithmetic

Chaya Ganesh[1], Anca Nitulescu[2], and Eduardo Soria-Vazquez[3]

[1] Indian Institute of Science, India.
[2] Protocol Labs, USA.
[3] Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE.**
chaya@iisc.ac.in, anca.nitulescu@protocol.ai, eduardo.soria-vazquez@tii.ae

Abstract. Succinct non-interactive arguments of knowledge (SNARKs) enable non-interactive efficient verification of NP computations and admit short proofs. However, all current SNARK constructions assume that the statements to be proven can be efficiently represented as either Boolean or arithmetic circuits over finite fields. For most constructions, the choice of the prime field $\mathbb{F}_p$ is limited by the existence of groups of matching order for which secure bilinear maps exist.
In this work we overcome such restrictions and enable verifying computations over rings. We construct the first designated-verifier SNARK for statements which are represented as circuits over a broader kind of commutative rings.
Our contribution is threefold:

1. We first introduce Quadratic Ring Programs (QRPs) as a characterization of NP where the arithmetic is over a ring.
2. Second, inspired by the framework in Gennaro, Gentry, Parno and Raykova (EUROCRYPT 2013), we design SNARKs over rings in a modular way. We generalize pre-existent assumptions employed in field-restricted SNARKs to encoding schemes over rings. As our encoding notion is generic in the choice of the ring, it is amenable to different settings.
3. Finally, we propose two applications for our SNARKs.
   - Our first application is verifiable computation over encrypted data, specifically for evaluations of Ring-LWE-based homomorphic encryption schemes.
   - In the second one, we use Rinocchio to naturally prove statements about circuits over e.g. $\mathbb{Z}_{2^{64}}$, which closely matches real-life computer architectures such as standard CPUs.

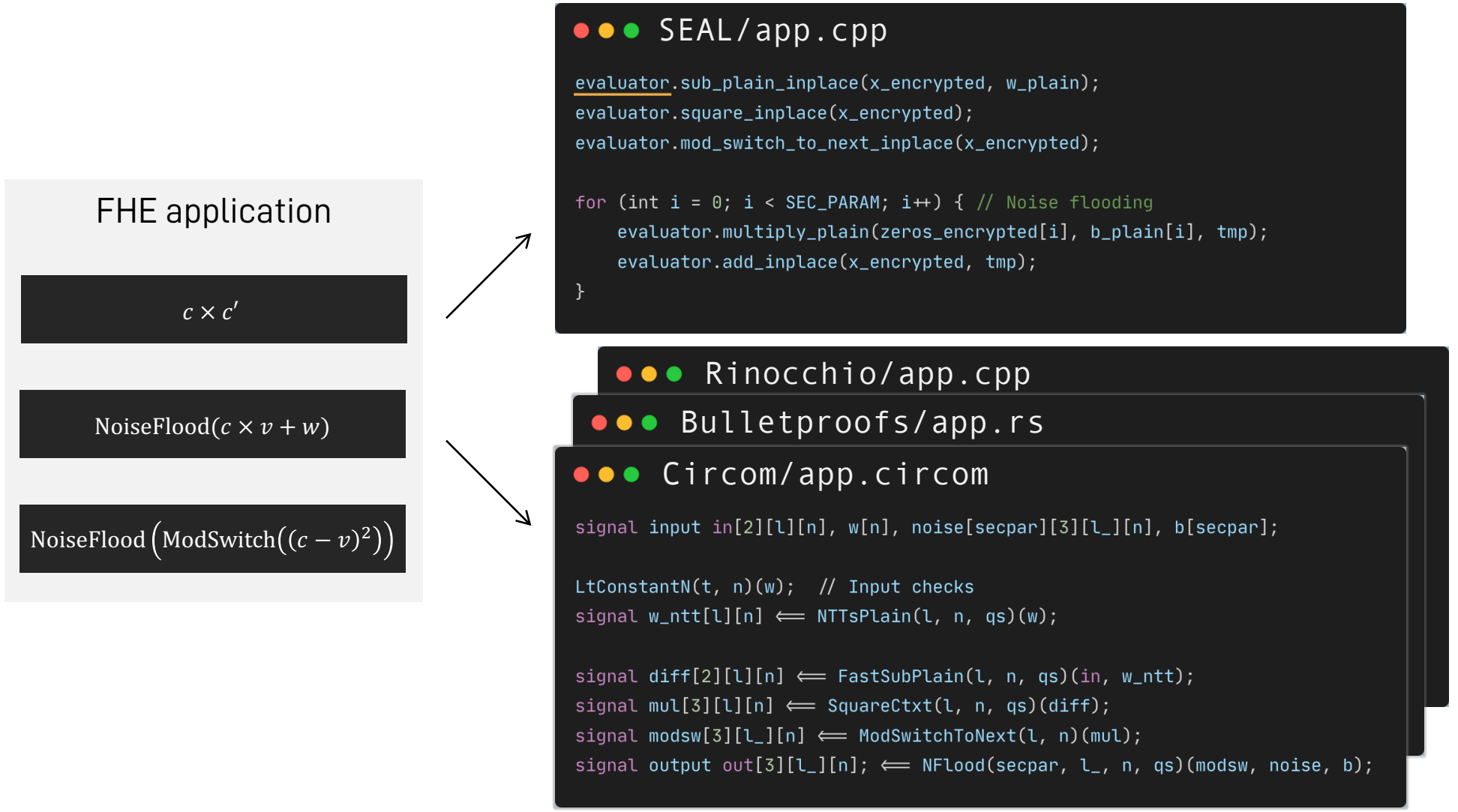
Optimized encodings
(≈ 1000× faster)

Non-standard assumption

Limited soundness (60 bits)

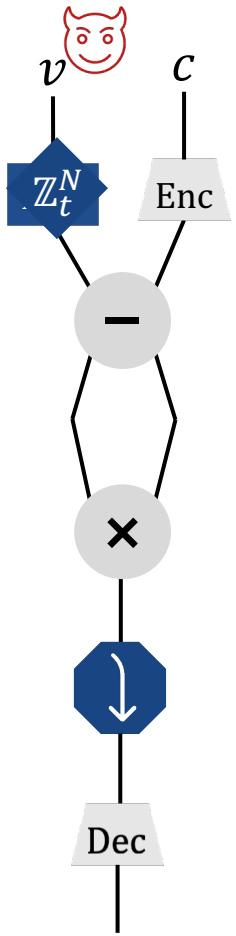
MarbleHE/ringSNARK

Evaluation


FHE application
$c \times c'$
$\text{NoiseFlood}(c \times v + w)$
$\text{NoiseFlood}\left(\text{ModSwitch}\left((c - v)^2\right)\right)$
SEAL/app.cpp
evaluator.sub_plain_inplace(x_encrypted, w_plain);
evaluator.square_inplace(x_encrypted);
evaluator.mod_switch_to_next_inplace(x_encrypted);
for (int i = 0; i < SEC_PARAM; i++) { // Noise flooding
evaluator.multiply_plain(zeros_encrypted[i], b_plain[i], tmp);
evaluator.add_inplace(x_encrypted, tmp);
}
Rinocchio/app.cpp
Bulletproofs/app.rs
Circom/app.circom
signal input in[2][l][n], w[n], noise[secpar][3][l_][n], b[secpar];
LtConstantN(t, n)(w); // Input checks
signal w_ntt[l][n] ⟸ NTTsPlain(l, n, qs)(w);
signal diff[2][l][n] ⟸ FastSubPlain(l, n, qs)(in, w_ntt);
signal mul[3][l][n] ⟸ SquareCtxt(l, n, qs)(diff);
signal modsw[3][l_][n] ⟸ ModSwitchToNext(l, n)(mul);
signal output out[3][l_][n]; ⟸ NFlood(secpar, l_, n, qs)(modsw, noise, b);

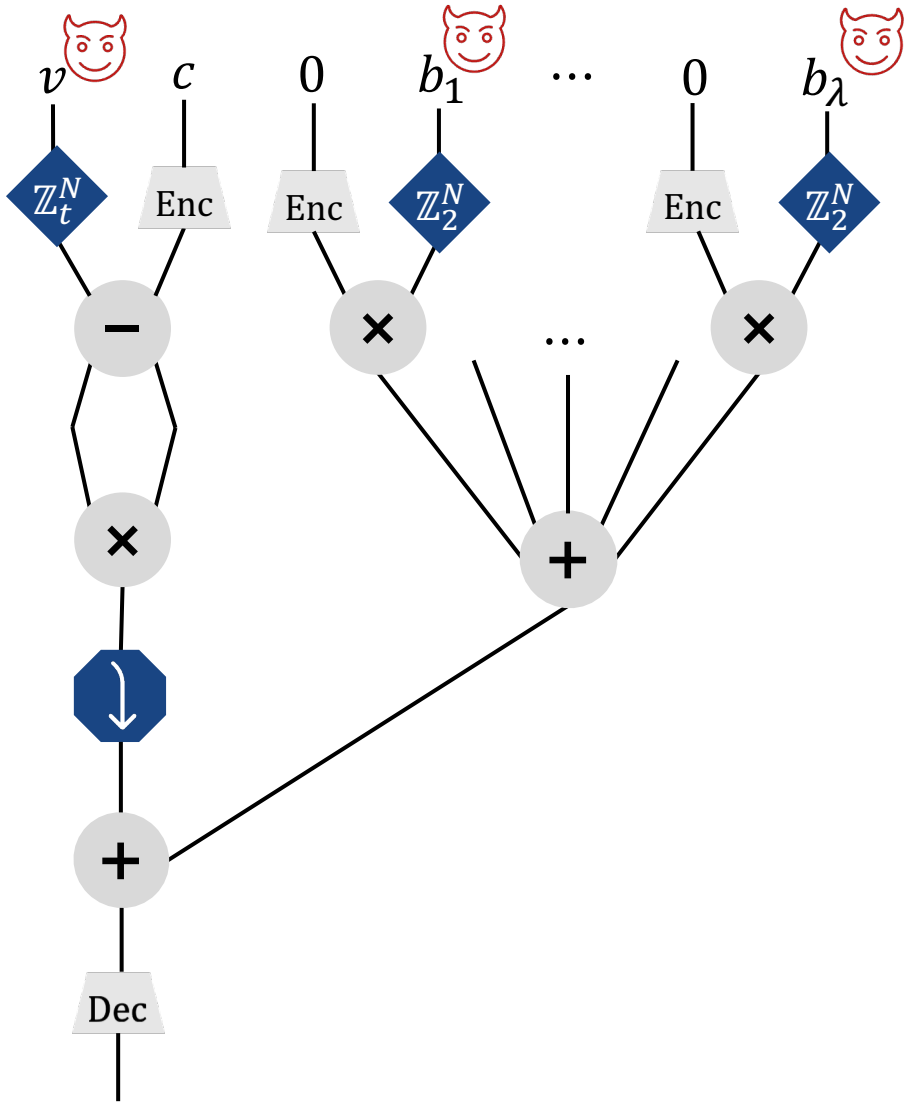
$$f = \text{ModSwitch}\big((c - v)^2\big)$$

Validity range check : $v \in \mathbb{Z}_t^N$



$$f = \text{NoiseFlood}\left(\text{ModSwitch}\left((c - v)^2\right)\right)$$

Validity range check : $v \in \mathbb{Z}_t$



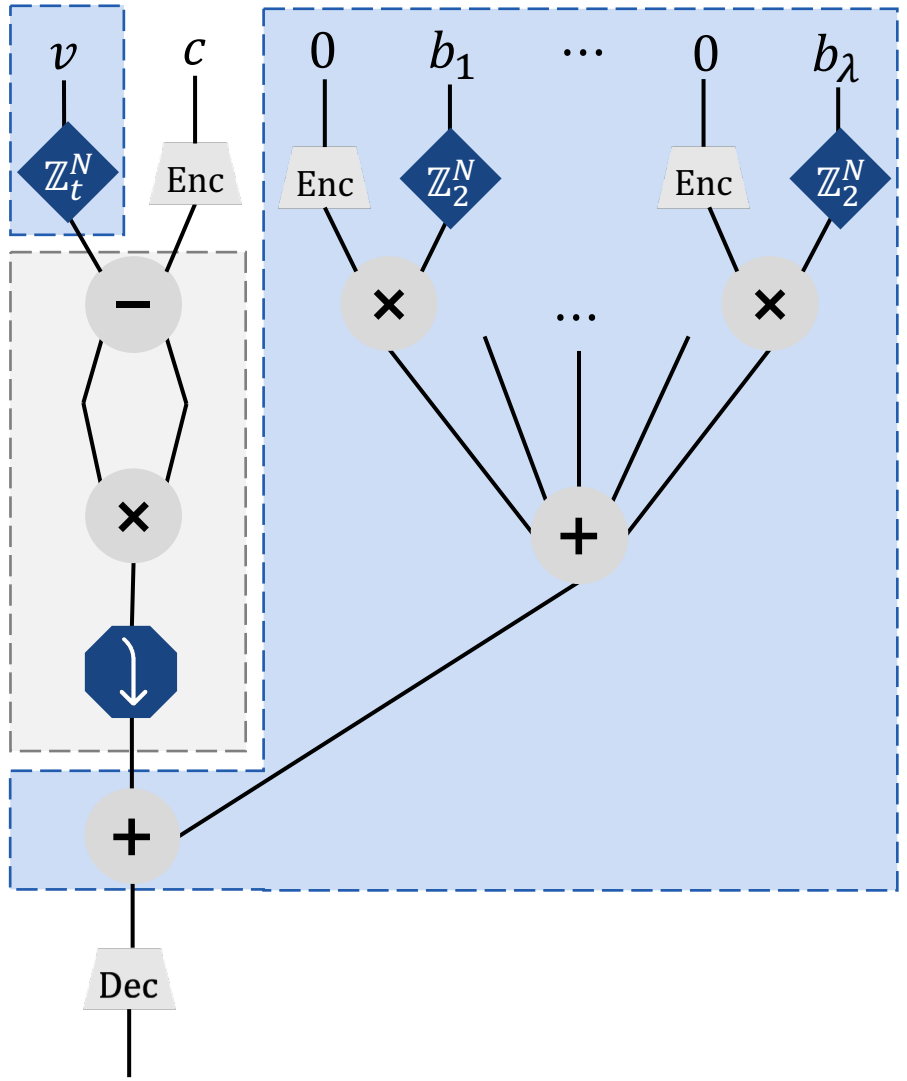
Binary check : $b_i \in \mathbb{Z}_2$

$$f = \text{NoiseFlood}\left(\text{ModSwitch}\left((c - v)^2\right)\right)$$

# R1CS constraints	
ZKP over $\mathbb{F}$	1.6×10^7
ZKP over $\mathcal{R}$	546

Ptxt check + NTT		
$\mathbb{F}$	1.4×10^6	8 %
$\mathcal{R}$	30	5.5 %

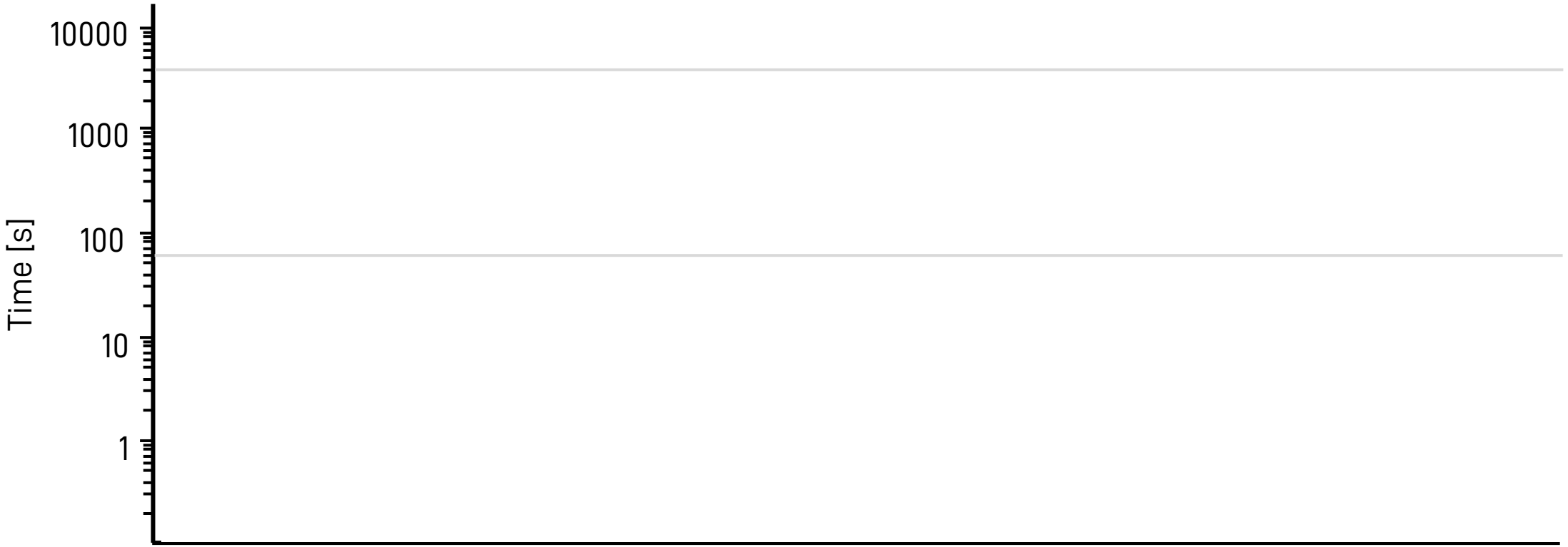
FHE circuit		
$\mathbb{F}$	6.1×10^6	37 %
$\mathcal{R}$	3	0.5 %



Noise flooding		
$\mathbb{F}$	9.1×10^6	55 %
$\mathcal{R}$	512	94 %

BGV, N = 8192, log q = 183, log t = 30

Is robust FHE practical?



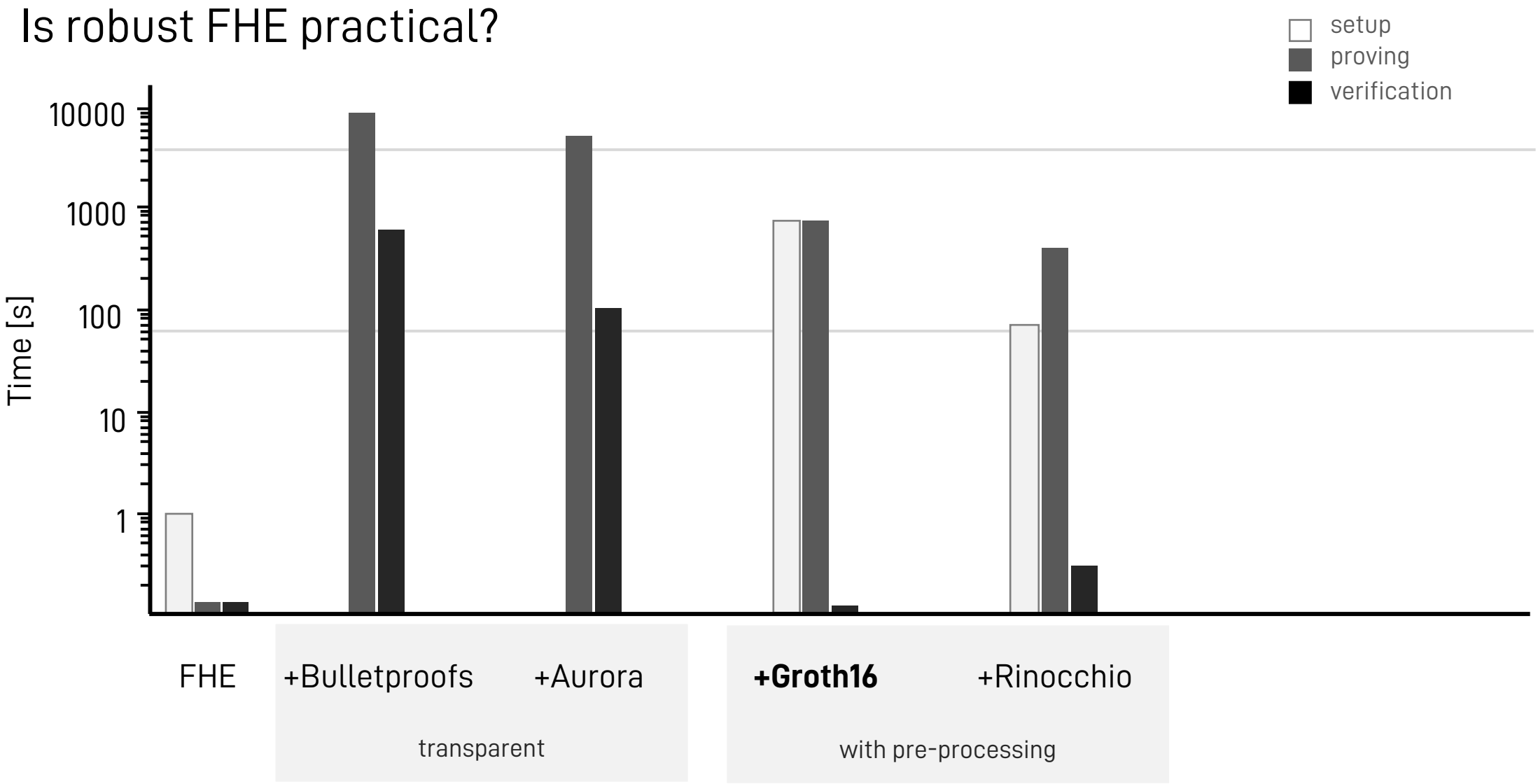
BGV, N = 8192, log q = 183, log t = 30

Is robust FHE practical?

Impractical

Acceptable

Practical

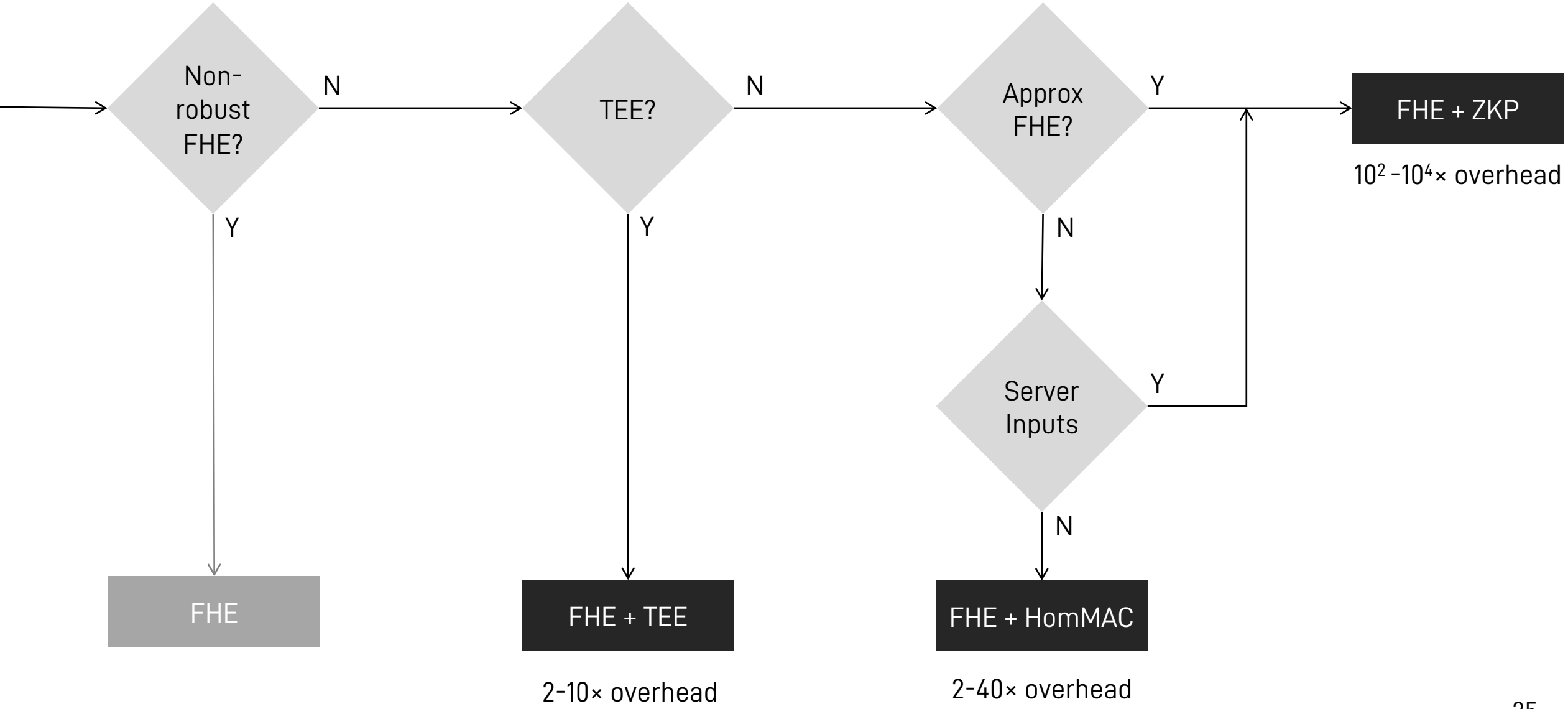


BGV, N = 8192, log q = 183, log t = 30

Is robust FHE practical?

	TEE	Field-ZKP			Ring-ZKP
		Transparent		Pre-processing	
	SGX	Bulletproofs	Aurora	Groth16	Rinocchio
Small circuits	Practical	Impractical	Impractical	Acceptable	Acceptable
Big circuits	Acceptable	Impractical	Impractical	Impractical	Acceptable

Take-aways for practitioners



ETH zürich

This work:

- Unified definitions for robust FHE
- Theoretic & concrete instantiation
- Improvements to FHE-in-TEE & ringSNARK

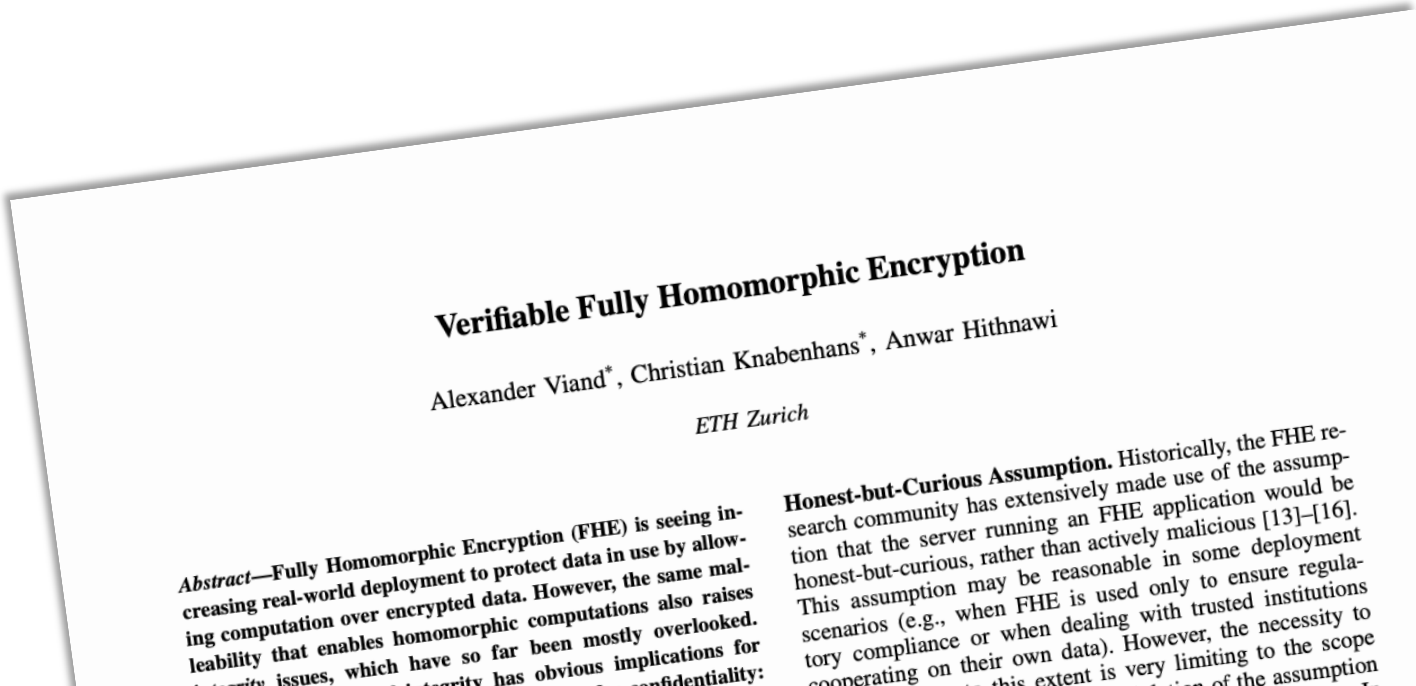
Future work:

- Stronger & application-specific notions
- More FHE-friendly proof systems
- Tooling & integration with compilers

MarbleHE/FHE-in-TEE

MarbleHE/ringSNARK

arXiv.org/abs/2301.07041



Verifiable Fully Homomorphic Encryption

Alexander Viand*, Christian Knabenhans*, Anwar Hithnawi

ETH Zurich

***Abstract*—Fully Homomorphic Encryption (FHE) is seeing increasing real-world deployment to protect data in use by allowing computation over encrypted data. However, the same malleability that enables homomorphic computations also raises integrity issues, which have so far been mostly overlooked. Integrity has obvious implications for confidentiality:**

Honest-but-Curious Assumption. Historically, the FHE research community has extensively made use of the assumption that the server running an FHE application would be honest-but-curious, rather than actively malicious [13]–[16]. This assumption may be reasonable in some deployment scenarios (e.g., when FHE is used only to ensure regulatory compliance or when dealing with trusted institutions cooperating on their own data). However, the necessity to this extent is very limiting to the scope of the assumption