# MHz2k: MPC from HE over $\mathbb{Z}_{2^k}$

## with New Packing, Simpler Reshare, and Better ZKP


Jung Hee Cheon (Seoul National Univ. & Crypto Lab Inc.)

Dongwoo Kim (Western Digital Research)

**Keewoo Lee (Seoul National Univ.)**

# A Brief History (of actively secure dishonest majority MPC in preprocessing model)

SHE | LHE | OT

$\mathbb{Z}_p$

BeDOZa [BDOZ;Euro11]

SPDZ [DPSZ;Crypto12]

$\mathbb{Z}_{2^k}$

# Preprocessing Model

- Preprocessing Phase: Triple Generation
  - parties share random $[a]_i$, $[b]_i$, $[c]_i$ such that $a \times b = c$ in $\mathbb{Z}_p$.
- Online Phase: Secure Computation via Beaver's Trick
  - consumes Beaver's triple at each mult. gate
- Not enough for malicious setting
  - e.g. adversaries can deviate from described protocol in online phase

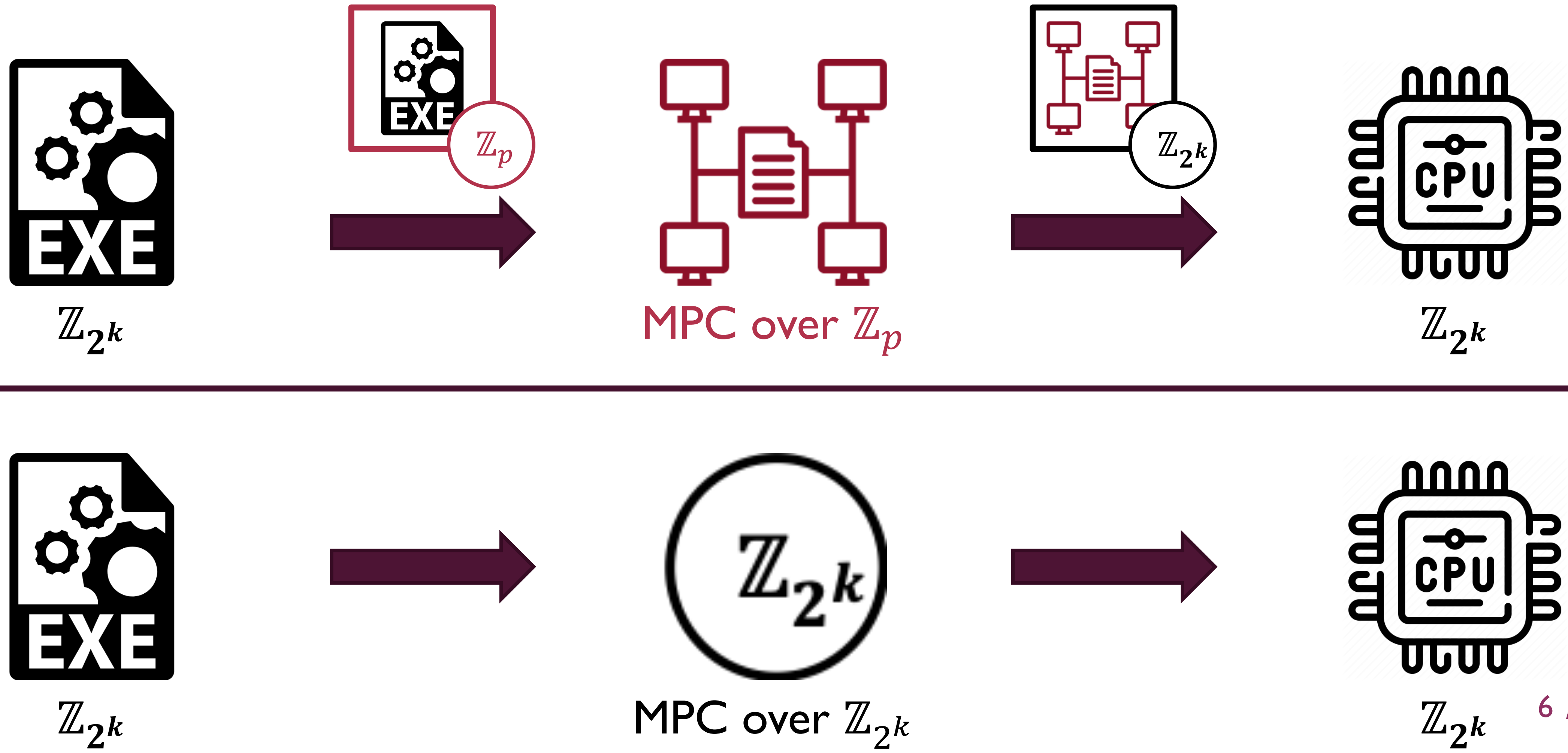
# Preprocessing Model (SPDZ)

- Authentication via MAC
  - Linear MAC: $MAC_{\alpha}(x) = \alpha \cdot x$ in $\mathbb{Z}_p$ , where $\alpha$ is a global MAC key
- Preprocessing Phase: Authenticated Triple Generation
  - share random $[a]_i$, $[b]_i$, $[c]_i$ **&** $[\boldsymbol{\alpha a}]_{\boldsymbol{i}}$**,** $[\boldsymbol{\alpha b}]_{\boldsymbol{i}}$**,** $[\boldsymbol{\alpha c}]_{\boldsymbol{i}}$ such that $a \times b = c$ in $\mathbb{Z}_p$ ( **using Enc($\boldsymbol{\alpha}$)** )
  - No one knows the value of $\alpha$ ( Enc($\alpha$):=$\sum$ Enc($\alpha_i$) )
  - Online Phase is essentially the same by linearity of MAC

# A Brief History (of actively secure dishonest majority MPC in preprocessing model)

| | SHE | LHE | OT |
|---|---|---|---|
| $\mathbb{Z}_p$ | | BeDOZa [BDOZ;Euro11] | |
| | SPDZ [DPSZ;Crypto12] | | |
| | SPDZ2 [DKL+;ESORICS13] | | |
| | | | MASCOT [KOS;CCS16] |
| | | Overdrive [KPR;Euro18] | |
| | | TopGear [BCS;SAC19] | |
| $\mathbb{Z}_{2^k}$ | | | SPDZ2k [CDE+;Crypto18] |

# $\mathbb{Z}_p$ vs $\mathbb{Z}_{2^k}$



# MPC over $\mathbb{Z}_{2^k}$

- Pros: No overheads for emulations
- Cons: MPC over $\mathbb{Z}_{2^k}$ requires a less efficient SPD$\mathbb{Z}_{2^k}$ MAC
- [DEF+;S&P19] reports upto 5x improvements in online phases.
- Disclaimer: Not an absolute advantage (Preprocessing Phase)

# A Brief History (of actively secure dishonest majority MPC in preprocessing model)

SHE | LHE | OT

$\mathbb{Z}_p$

BeDOZa [BDOZ;Euro11]

SPDZ [DPSZ;Crypto12]

SPDZ2 [DKL+;ESORICS13]

MASCOT [KOS;CCS16]

Overdrive [KPR;Euro18]

TopGear [BCS;SAC19]

$\mathbb{Z}_{2^k}$

SPDZ2k [CDE+;Crypto18]

Overdrive2k [OSV;CT-RSA20]

Monza [CDFG;PKC20]

MHz2k [CKL;Crypto21]

# Authenticated Triple Generation (SHE-based)

$$
\begin{tikzcd}
{[a]_i} \arrow[r] & \text{Enc}([a]_i) \\ \text{ZKPoPK}([a]_i) \arrow[r] & \text{Enc}(a) \arrow[r, "\text{Enc}(\alpha)"] \arrow[d, phantom, "\times"] & \text{Enc}^0(\alpha a) \arrow[r] & {[\alpha a]_i} \\
{[b]_i} \arrow[r] & \text{Enc}([b]_i) \\ \text{ZKPoPK}([b]_i) \arrow[r] & \text{Enc}(b) \arrow[r] \arrow[d] & \text{Enc}^0(\alpha b) \arrow[r] & {[\alpha b]_i} \\
{[c]_i} & \text{Enc}^0(c) \arrow[l] \arrow[d] & & \\
& \text{Enc}(c) \arrow[r] & \text{Enc}^0(\alpha c) \arrow[r] & {[\alpha c]_i}
\end{tikzcd}
$$

- Packing method
- Reshare protocol
- ZKP of Plaintext Knowledge

# New Packing

Tweaked Interpolation Packing for $\mathbb{Z}_{2^k}$-messages

# HE Packing

(Somewhat) Homomorphic Encryption

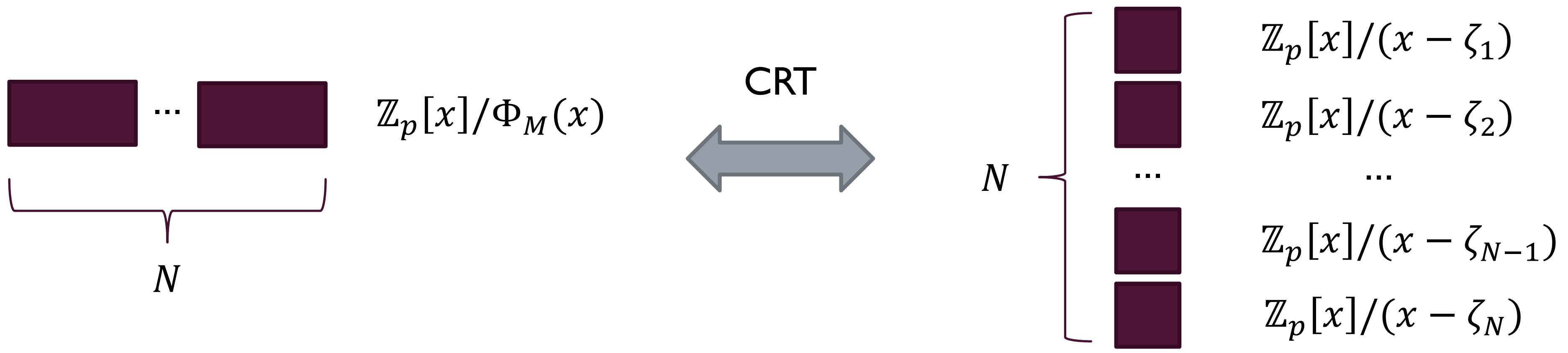
| Message Space | | Plaintext Space | | Ciphertext Space |
| --- | --- | --- | --- | --- |
| $\prod \mathbb{Z}_t$ | Packing $\rightarrow$ / $\leftarrow$ Unpacking | $\mathbb{Z}_p[x]/\Phi_M(x)$ | Encryption $\rightarrow$ / $\leftarrow$ Decryption | $(\mathbb{Z}_q[x]/\Phi_M(x))^2$ |

- enhances amortized performance (SIMD-like optimizations)
- In SPDZ-family, **packing density** directly affects the **throughput** of triple generation.

# Conventional Packing Method for $\mathbb{Z}_p$

$$\Phi_M(x) = \prod(x - \zeta_i) \pmod{\text{p}}$$

$\zeta_i$ : Mth root of unity mod p,
$p \equiv 1 \pmod{\text{M}}$

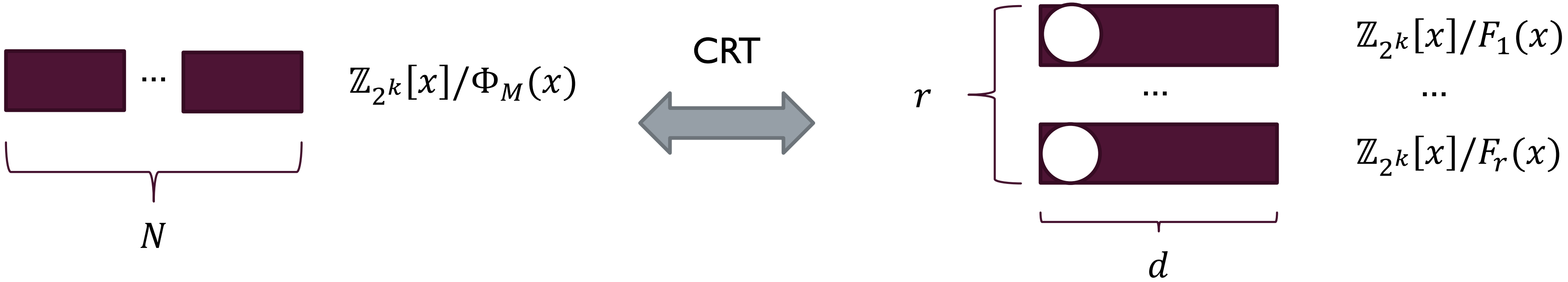



- Fully homomorphic correspondence & Level-consistent
- No redundancy (perfect packing density)

# Packing Method for $\mathbb{Z}_{2^k}$: HELib

$$\Phi_M(x) = \prod F_i(x) \pmod{2^k}$$

M: odd
$\deg F_i = d = ord_M(2)$

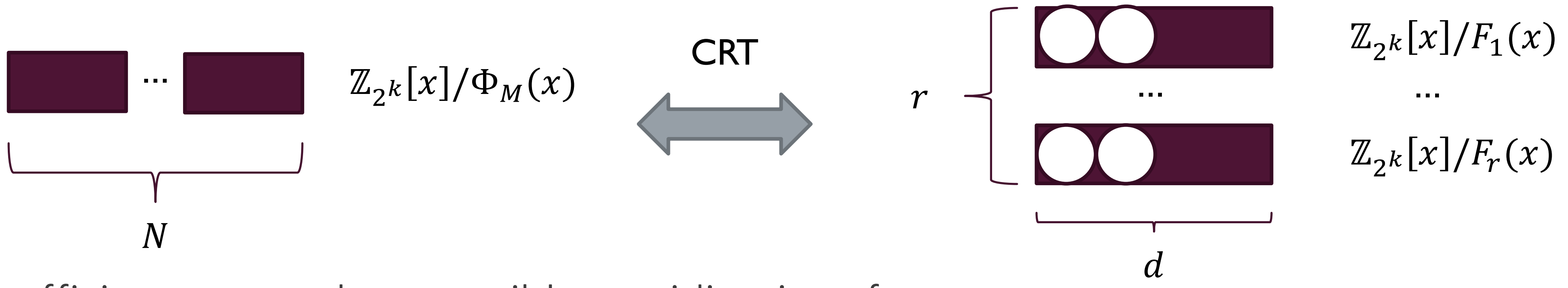



- Use constant coefficients only
- Fully homomorphic correspondence & Level-consistent
- Very low packing density (1/d)

# Packing Method for $\mathbb{Z}_{2^k}$: Overdrive2k

$$\Phi_M(x) = \prod F_i(x) \pmod{2^k}$$

M: odd
$\deg F_i = d = ord_M(2)$

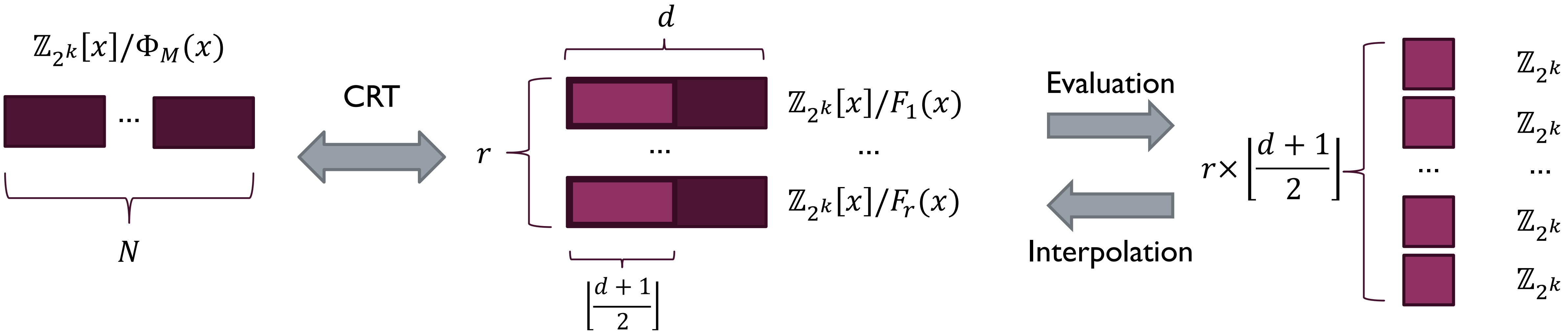



- Use coefficients as much as possible, avoiding interference.
- $(a_0 + a_1x + a_3x^3 + a_4x^4 + a_9x^9)(b_0 + b_1x + b_3x^3 + b_4x^4 + b_9x^9)$
  $= a_0b_0 + \cdots + a_1b_1x^2 + \cdots + a_3b_3x^6 + \cdots + a_4b_4x^8 + \cdots + a_9b_9x^{18}$
- **Somewhat** homomorphic correspondence (1 Mult) & Level-dependent
- Packing density $\approx 1/d^{0.4} \approx 1/5$

# Packing Method for $\mathbb{Z}_{2^k}$: Interpolation?

$$\Phi_M(x) = \prod F_i(x) \pmod{2^k}$$

M: odd
$\deg F_i = d = ord_M(2)$




- Avoid Degree Overflow: Somewhat homomorphic correspondence (1 Mult)
- Packing density $\approx$ 1/2
- **However, interpolation over $\mathbb{Z}_{2^k}$ is impossible in general: consider f(0) and f(2).**

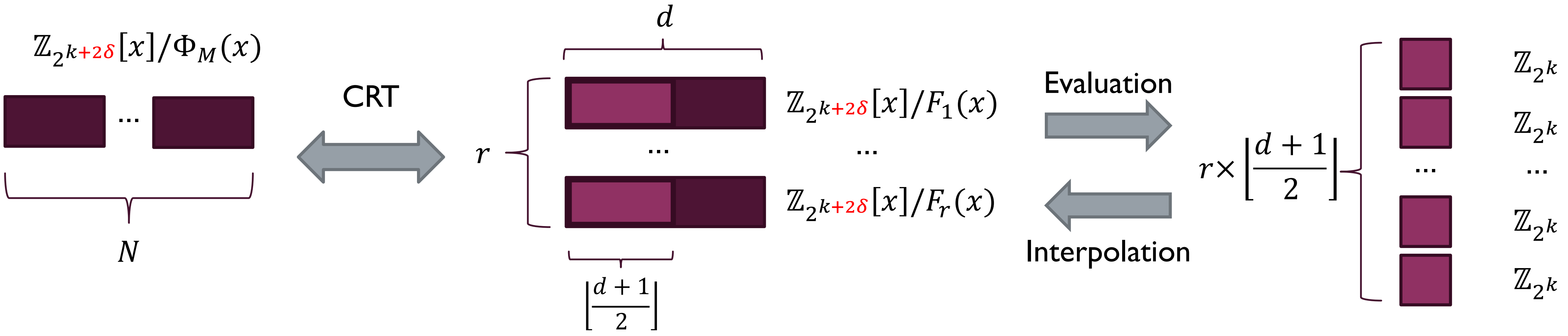
# Tweaked Interpolation over $\mathbb{Z}_{2^k}$

- We devised a method to perform pseudo-interpolation over $\mathbb{Z}_{2^k}$.
- **Lift the target points to a larger ring to cancel out effects of zero-divisors.**
- For any $(m_0, \cdots, m_{n-1}) \in \mathbb{Z}_{2^k}^n$, there exists $f(x) \in \mathbb{Z}_{2^{k+\delta}}[x]$ s.t.
  - $\deg(f) < n$
  - $f(i) = m_i \times 2^\delta$ for $i < n$

# Packing Method for $\mathbb{Z}_{2^k}$: Tweaked Interpolation!

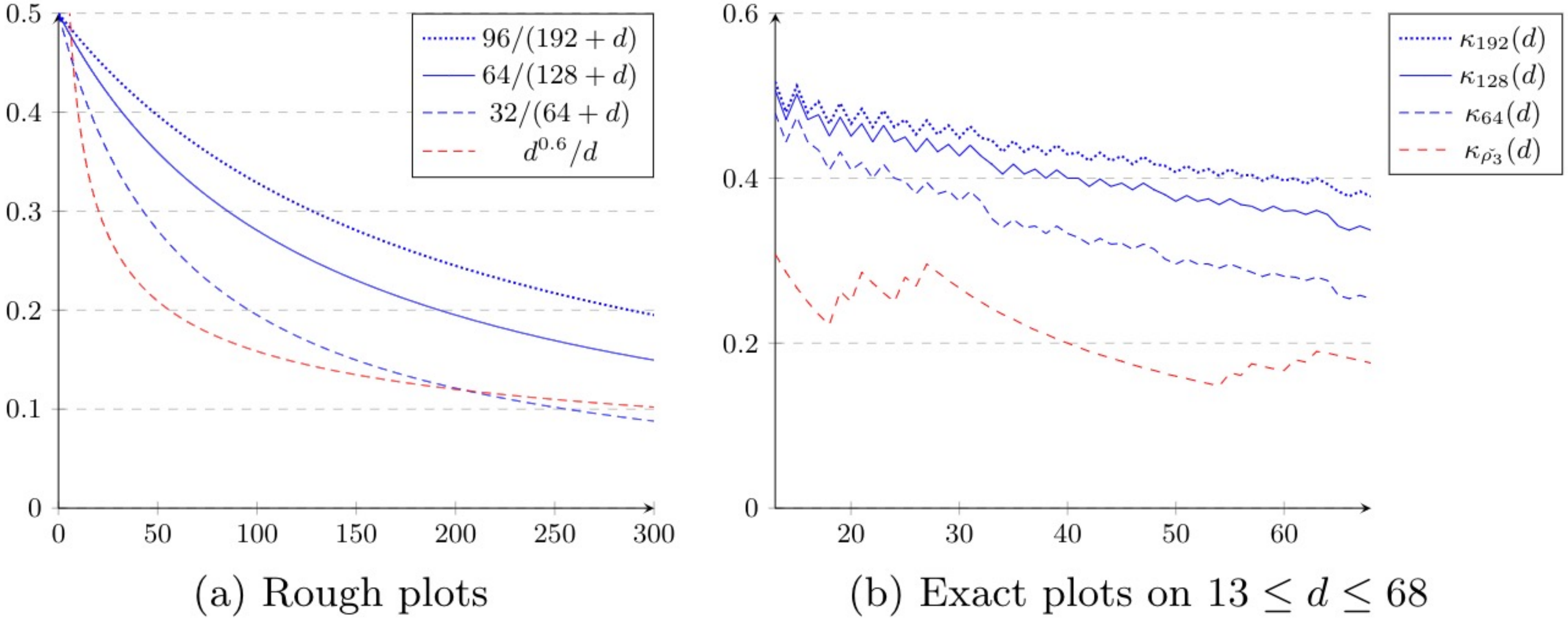
$$\Phi_M(x) = \prod F_i(x) \pmod{2^{k+2\delta}}$$

M: odd
$\deg F_i = d = ord_M(2)$




- Avoid Degree **& Modulus Overflow**
- Somewhat homomorphic correspondence (1 Mult) & Level-dependent
- **Packing density ≈ k/(2k+2d) ≈ 1/2** (This near optimal. See [CL21])

# Performance




- Upto 2.5x improvements in packing density versus Overdrive2k

# Simpler Reshare

## Reshare Protocol for Level-dependent Packings

# Level-dependent Packings

- Packings for $\mathbb{Z}_{2^k}$ (Overdrive2k & MHz2k) are "**Level-dependent**"
  - i.e. different packing structure after mult.
  - This is inevitable! (See [CL21])
  - in contrast to the conventional level-consistent packing for $\mathbb{Z}_p$
- Issue: no homomorphic computation between different packing levels

# Reshare Protocol for Level-dependent Packings

- Reshare: "re-encrypt" a level-0 HE ctxt into a fresh HE ctxt

$$\mathsf{Enc}^0(c) \xrightarrow{+\mathsf{Enc}^0(r)} \mathsf{Enc}^0(r+c) \xrightarrow{\mathsf{Dec}(\cdot)} r+c \xrightarrow{\mathsf{Enc}(\cdot)} \mathsf{Enc}^1(r+c) \xrightarrow{-\mathsf{Enc}^1(r)} \mathsf{Enc}^1(c)$$

  - masking ctxt is used **twice** at **different levels**
  - ctxt level can be adjusted by "modulus switching"
  - packing level…? (not a problem in level-consistent packing for $\mathbb{Z}_p$)
  - Overdrive2k provides **two** masking ctxt with **same message** and **different packing levels**

# Reshare Protocol for Level-dependent Packings

- We resolve this issue by a technical trick
  - Reshare is only used to support an additional mult. with a **constant** (MAC key $\alpha$)
  - We use different packing structure for MAC key (constant packing)
  - closes gap between $\mathbb{Z}_p$ and $\mathbb{Z}_{2^k}$ caused by level-consistency
  - offers 1.4x reduction in comm. cost compared to the solution of Overdrive2k

# Better ZKP

TopGear2k: Better ZKP for Lattice Enc. on $\mathbb{Z}[X]/\Phi_p(X)$

# ZKPoPK on HE

- ZKP of Plaintext Knowledge
  - guarantees that a ciphertext is validly generated from a plaintext
  - restricts adversaries from submitting maliciously generated ciphertexts

$Enc(x,r), x, r$

**Prover**

$Enc(y,s)$ $\longrightarrow$

$\longleftarrow$ Challenge $c \in \{0,1\}$

$cx+y, cr+s$ $\longrightarrow$

$Enc(x,r)$

**Verifier**

Check

$$Enc(cx+y, cr+s) = c \cdot Enc(x,r) + Enc(y,s)$$

and Size Bounds

# ZKPoPK on HE (TopGear)

- TopGear [BCS;SAC19]: Efficient ZKPoPK with larger challenge space
  - leverages the structure of HE plaintext space $\mathbb{Z}[X]/\Phi_M(X)$ with $M = 2^m$
  - favorable in comm. cost, latency, and memory consumption (in batched version)

$Enc(x,r), x, r$ **Prover** $\xrightarrow{Enc(y,s)}$ **Verifier** $Enc(x,r)$

Challenge $c \in \{0\} \cup \{X^i\}_i$ $\longleftarrow$

$\xrightarrow{cx+y,\, cr+s}$

Check

$$Enc(cx+y, cr+s) = c \cdot Enc(x,r) + Enc(y,s)$$

and Size Bounds

# Math behind the Scene

1) Multiplying $X^i$ in $\mathbb{Z}[X]/\Phi_M(X)$ does not increase coefficients (too much).

2) There is a "small" "pseudo-inverse" of $(X^i - X^j)$ in $\mathbb{Z}[X]/\Phi_M(X)$.

   - first observed in [BCK+;Asia14] and is being widely employed
   - affects soundness of ZKPoPK

- The lemmas were known only for $M = 2^m$ case
  - $\Phi_{2^m}(X)$ is irreducible modulo $2^k$ : cannot leverage parallelism via CRT
  - Overdrive2k could not employ TopGear optimization

# TopGear2k: ZKPoPK on HE over $\mathbb{Z}_{2^k}$

- We extended the lemmas to $M = p$ case
  - and even to $M = p^s q^t$ case (See [CKKL21])

- TopGear2k: Efficient ZKPoPK over $\mathbb{Z}[X]/\Phi_M(X)$ with $M = p^s q^t$
  - allows to use larger challenge space for $\mathbb{Z}_{2^k}$ as in TopGear for $\mathbb{Z}_p$
  - favorable in comm. cost, latency, and memory consumption (in batched version)

# ZKPoMK

ZKP of Message Knowledge

# Non-surjective Packings

- Packings for $\mathbb{Z}_{2^k}$ (Overdrive2k & MHz2k) are "**Non-surjective**"
  - i.e. there exist "**invalid**" packings in plaintext space
  - This is inevitable! (See [CL21])
  - in contrast to the conventional surjective packing for $\mathbb{Z}_p$
- Issue: Malicious adversaries may make use of invalid packings

# ZKPoMK

- We conceptualize ZKP of Message Knowledge.
  - which guarantees that a **ctxt** encrypts a **valid packing**.
  - ZKPoMK was neglected in Overdrive2k.
  - ZKPoMK can be easily integrated into ZKPo**P**K, if the small challenge space {0,1} is used.
- We design an efficient ZKPoMK for our new packing method.

# Conclusion

# Authenticated Triple Generation (SHE-based)

$$
\begin{tikzcd}
{[a]_i} \arrow[r] & \begin{array}{c}\mathrm{Enc}([a]_i)\\ \mathrm{ZKPoPK}([a]_i)\end{array} \arrow[r] & \mathrm{Enc}(a) \arrow[r, "\mathrm{Enc}(\alpha)"] \arrow[d, phantom, "\times"] & \mathrm{Enc}^0(\alpha a) \arrow[r] & {[\alpha a]_i} \\
{[b]_i} \arrow[r] & \begin{array}{c}\mathrm{Enc}([b]_i)\\ \mathrm{ZKPoPK}([b]_i)\end{array} \arrow[r] & \mathrm{Enc}(b) \arrow[r] \arrow[d] & \mathrm{Enc}^0(\alpha b) \arrow[r] & {[\alpha b]_i} \\
& {[c]_i} & \mathrm{Enc}^0(c) \arrow[l] \arrow[d] & & \\
& & \mathrm{Enc}(c) \arrow[r] & \mathrm{Enc}^0(\alpha c) \arrow[r] & {[\alpha c]_i}
\end{tikzcd}
$$

- Packing method
- Reshare protocol
- ZKPoPK and ZKPoMK

# Conclusion

- MHz2k: MPC over $\mathbb{Z}_{2^k}$ secure against actively corrupted majority
  - 2.2x ~ 4.8x improvements in amortized comm. cost (vs. previous best schemes)
  - 3.7x ~ 6.4x improvements in memory consumption (vs. Overdrive2k)
- New Techniques and Concepts
  - tweaked interpolation & tweaked interpolation packing
  - level-dependency & surjectivity for packings
  - constant packing trick for simpler Reshare
  - generalization of [BCK+;Asia] lemma
  - new notion of ZKPoMK

# Thank You!

* Live Session: Aug 20th 15:00-15:50 UTC
* E-mail: activecondor@snu.ac.kr

# Abbreviations

- Ctxt : Ciphertext
- Comm. Cost : Communication Cost
- Dec. : Decryption
- Enc. : Encryption
- HE : Homomorphic Encryption
- LHE : Linearly Homomorphic Encryption
- MAC : Message Authentication Code
- MPC : Secure Multi-party Computation
- Mult. : Multiplication
- OT : Oblivious Transfer
- SHE : Somewhat Homomorphic Encryption
- SIMD : Single Instruction Multiple Data
- ZKP : Zero-knowledge Proof
- ZKPoPK : ZKP of Plaintext Knowledge
- ZKPoMK : ZKP of Message Knowledge

# References


- [BCK+;Asia14] Better Zero-knowledge Proofs for Lattice Encryption and Their Application to Group Signatures
- [BCS;SAC19] Using TopGear in Overdrive: A more efficient ZKPoK for SPDZ
- [BDOZ;Euro11] Semi-homomorphic Encryption and Multiparty Computation
- [CDE+;Crypto18] SPD$\mathbb{Z}_{2^k}$: Efficient MPC mod $2^k$ for Dishonest Majority
- [CDFG;PKC20] Mon$\mathbb{Z}_{2^k}$a: Fast Maliciously Secure Two Party Computation on $\mathbb{Z}_{2^k}$
- [CKKL;arXiv21] On the Scaled Inverse of $(x^i - x^j)$ modulo Cyclotomic Polynomial of the form $\Phi_{p^s}(x)$ or $\Phi_{p^s q^t}(x)$
- [CL;ePrint21] Limits of Polynomial Packings for $\mathbb{Z}_{p^k}$ and $\mathbb{F}_{p^k}$
- [DEF+;S&P19] New Primitives for Actively-secure MPC over Rings with Applications to Private Machine Learning
- [DKL+;ESORICS13] Practical Covertly Secure MPC for Dishonest Majority--or: Breaking the SPDZ Limits
- [DPSZ;Crypto12] Multiparty Computation from Somewhat Homomorphic Encryption
- [KOS;CCS16] MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer
- [KPR;Euro18] Overdrive: Making SPDZ Great Again
- [OSV;CT-RSA20] Overdrive2k: Efficient Secure MPC over $\mathbb{Z}_{2^k}$ from Somewhat Homomorphic Encryption