Recent Advances on HE for Multiple Parties

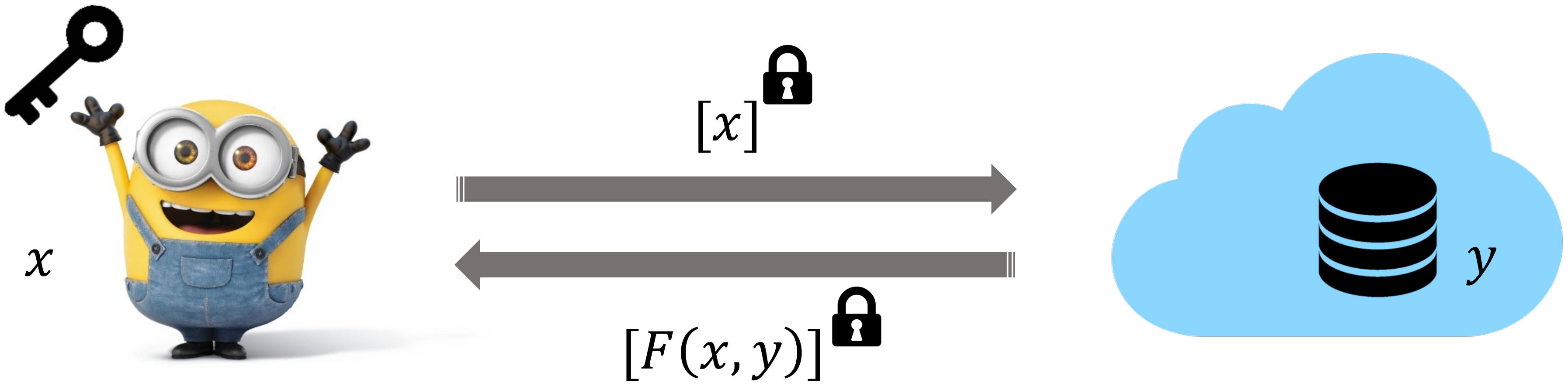
Yongsoo Song (Seoul National University)

Mar 26, 2023

The 2nd FHE.org Conference

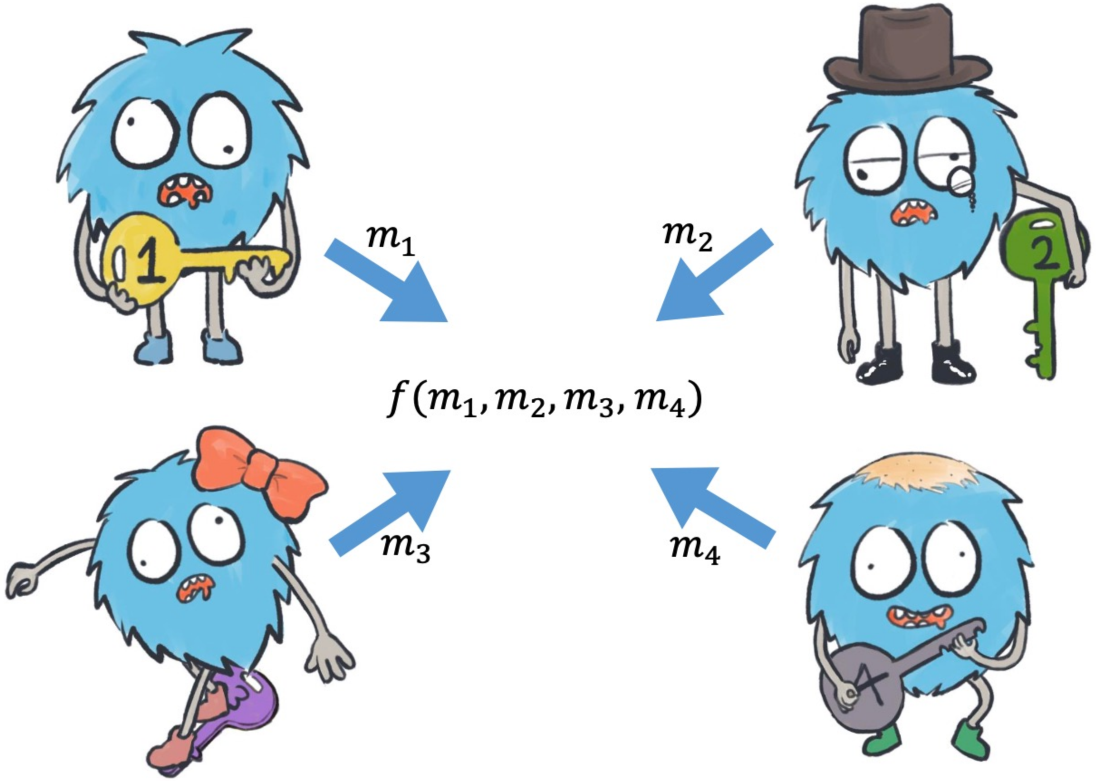
Roadmap

Use Cases of HE: Scenario 1



- Privacy-preserving personalized services
- Can be implemented with a standard (single-key) HE

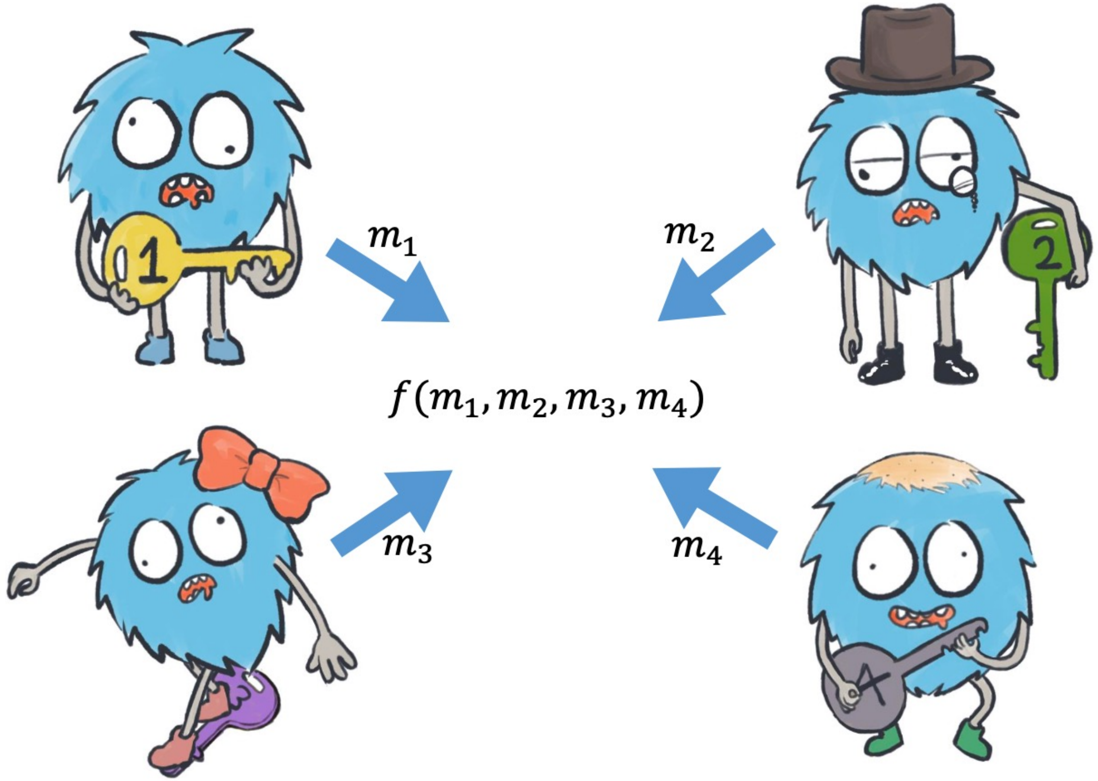
Use Cases of HE: Scenario 2



* Image courtesy of Seonhong Min

- Secure data aggregation and analysis
- The key management problem arises
- Need for HE variants with distributed authority

Building Multiparty Protocols from HE

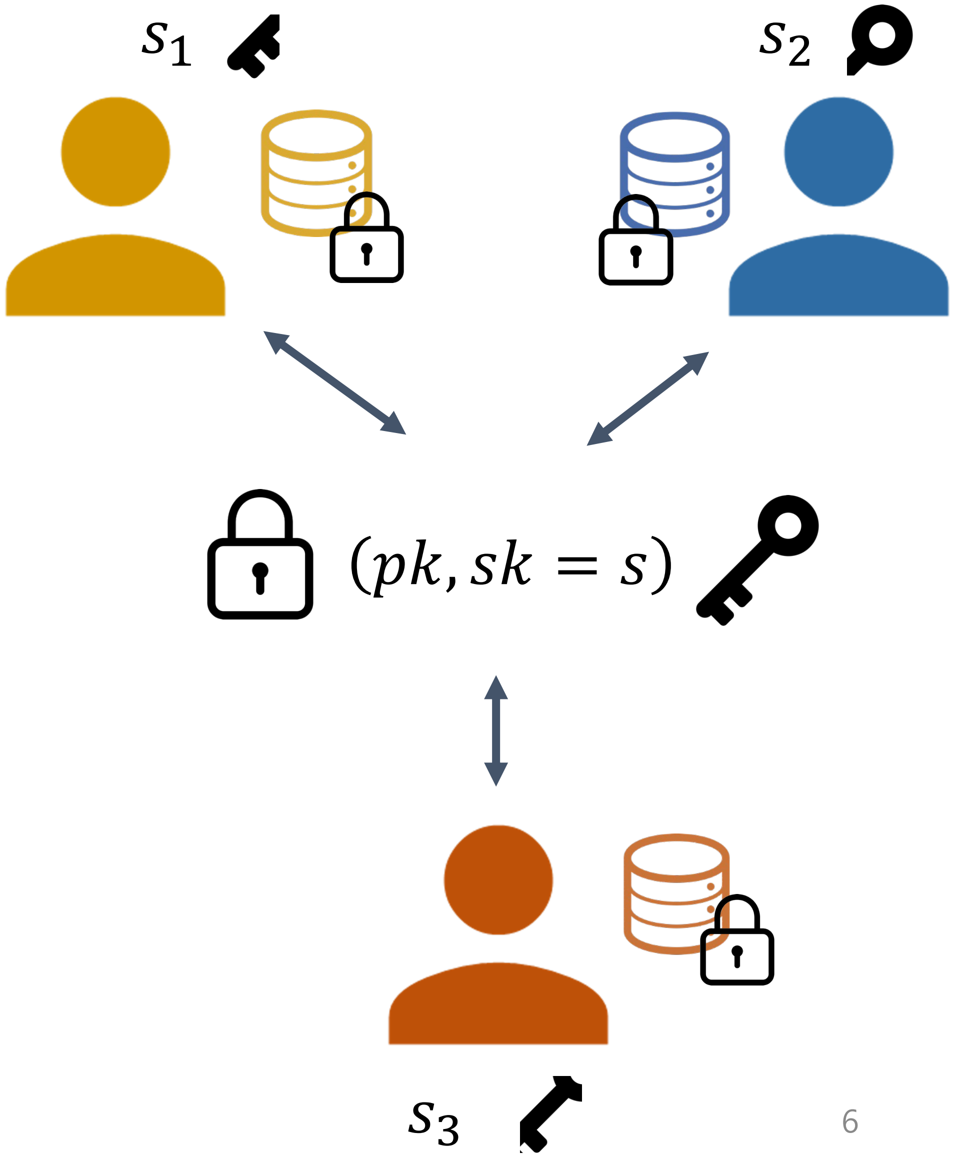


* Image courtesy of Seonhong Min

- Key Generation – Encryption – Evaluation – (Distributed) Decryption
  - (+) Low communication cost, user-friendly
  - (–) High computational complexity (cloud)

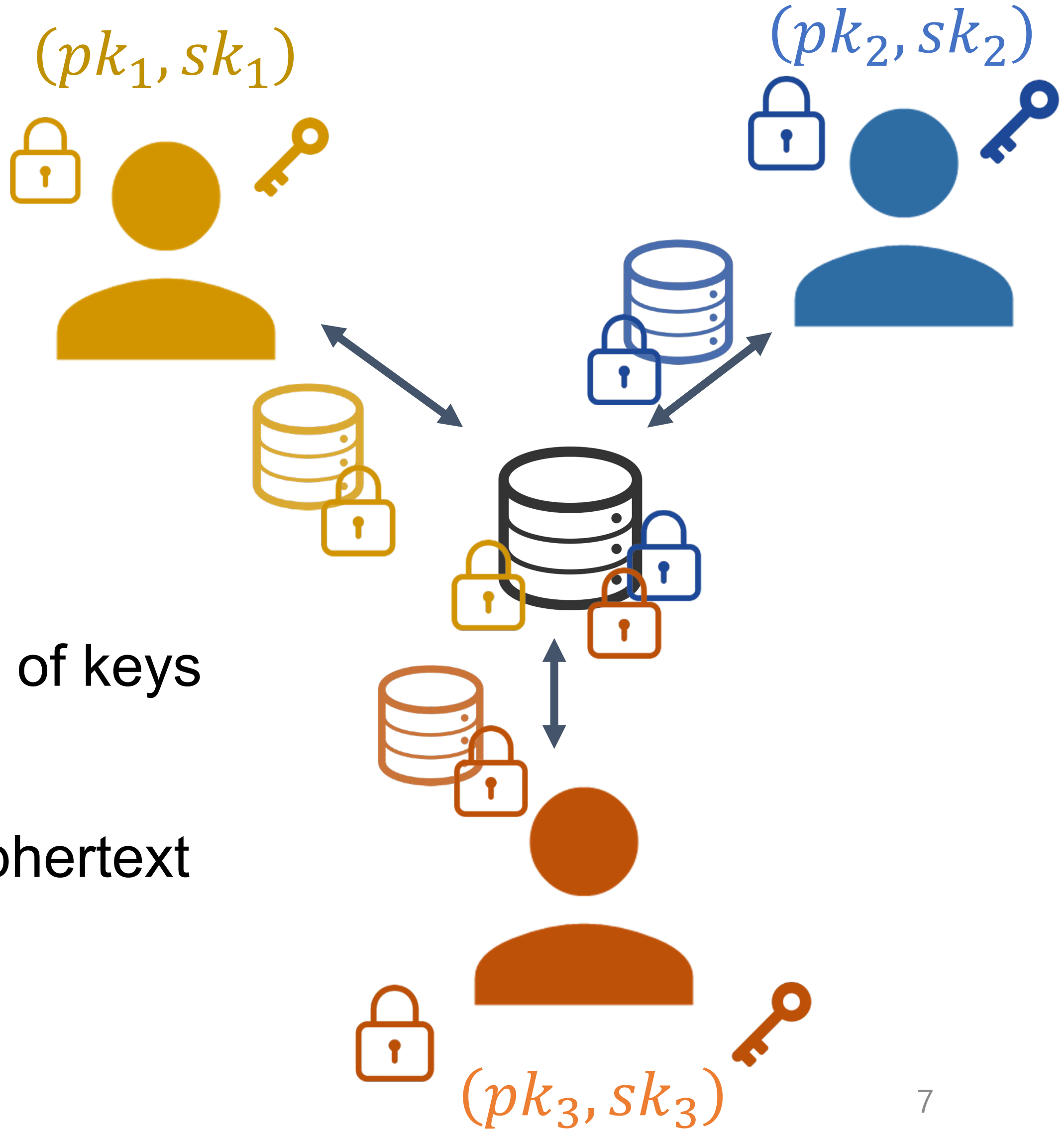
Direction 1 – Threshold HE (ThHE)

- **Setup:** parameters, a set of parties $P_1, \ldots, P_n$
- **Key Generation Protocol**
  - Build a joint public key pk
  - Each party P_i obtains a secret share s_i
- **Encryption & Evaluation**
  - The public key pk is commonly used
- **Decryption**
  - t out of n shares $s_1, \ldots, s_n$ can be used to recover the secret s
  - Distributed decryption by t parties



Direction 2 – Multi-key HE (MKHE)

- **Setup:** parameters
- **Key Generation Algorithm**
  - Each party P_i generates its own key pair
- **Encryption**
  - Output a single-key ciphertext
- **Evaluation**
  - On ciphertexts under possibly different sets of keys
- **Decryption**
  - Need all secret keys associated with the ciphertext
  - Distributed decryption is possible



ThHE vs MKHE

- **Threshold HE**

(+) Efficiency

Comparable to single-key HE

(–) Static & Interactive

A set of parties should be determined at the beginning and cannot be changed later.

The joint key generation requires interaction.

- **Multi-key HE**

(+) Flexibility & Dynamism

Independent key generation & encryption

Anyone can join the computation at any time

(–) Inefficient

Large ciphertext & expensive operation

Depending on the number of parties

Research Landscape (ThHE)

ThHE
(t out of n access structure)

MKHE

$t < n$

Very limited results

[BGG+18] Impractical

[MBH22] Stronger assumption in decryption

Multiparty HE (MPHE)
$t = n$

[AJL+12] Theoretic

[MTBH20,Park21] Interactive key generation (relin. key)

Multi-group HE (MGHE)

[KLSW21] The best of two worlds: MP+MKHE

Non-interactive keygen

Check out our **poster 1** :)

Research Landscape (MKHE)

MKHE

Old papers (~2017)
[LATV12,CM15,MW16,
PS16,BP16,CZW17]
Theoretic studies,
Mostly based on GSW,
No implementation

MK-TFHE

[CCS19] Hybrid product between MK-RLWE & SK-GSW, quadratic complexity w/ n, first MKHE implementation

[KMS21] Quasi-linear complexity. Visit our **poster 2.**

MK-CKKS/BFV

[CDKS19] MK relinearization w/ quadratic complexity

[KKLSS22] Linear complexity. **More in this talk.**

Overview of [CDKS19]

- Encryption is the same as single-key CKKS
- A fresh ciphertext is a pair $\boldsymbol{c} = (c_0, c_1) \in R_Q^2$ such that $c_0 + c_1 s \approx m \pmod{Q}$.
- Let $\boldsymbol{c} = (c_0, c_1)$, $\boldsymbol{c}' = (c_0', c_1')$ be fresh ciphertexts under secrets s, s'.
  - Then we define $\boldsymbol{c} + \boldsymbol{c}' = (c_0 + c_0', c_1, c_1') \pmod{Q}$
  - Decryptable by two keys as $(c_0 + c_0') + c_1 s + c_1' s' \approx m + m' \pmod{Q}$.
- In general, an MK ciphertext is of the form $\boldsymbol{c} = (c_0, c_1, \ldots, c_n)$
  - n is the number of parties associated with the ciphertext.
  - $c_0 + c_1 s_1 + \cdots + c_n s_n \approx m \pmod{Q}$.

MK Homomorphic Mult [CDKS19]

- Input: $\boldsymbol{c} = (c_0, c_1, \dots, c_n)$, $\boldsymbol{c}' = (c'_0, c'_1, \dots, c'_n)$
- Step 1: Simple product
  - Compute $\boldsymbol{c} \otimes \boldsymbol{c}' = \left(c_{i,j}\right)_{0 \le i,j \le n}$ where $c_{i,j} = c_i \cdot c'_j$.
  - Encryption of mm', under secret $\left(s_i \cdot s_j\right)_{0 \le i,j \le n}$.
- Step 2: Relinearization
  - Need a key-switching key for $s_i \cdot s_j$
  - Combine public keys of P_i and P_j to relinearize $c_{i,j}$:

$$\left(\left(c_{i,j} \boxdot \boldsymbol{b}_j\right) \boxdot \boldsymbol{v}_i,\ \left(c_{i,j} \boxdot \boldsymbol{b}_j\right) \boxdot \boldsymbol{u}_i,\ c_{i,j} \boxdot \boldsymbol{d}_i\right) \text{ under } \left(1, s_i, s_j\right).$$

  - Require a quadratic complexity with n.

MK Homomorphic Mult [CDKS19]

- Input: $\boldsymbol{c} = (c_0, c_1, \ldots, c_n)$, $\boldsymbol{c}' = (c_0', c_1', \ldots, c_n')$
- Step 1: Simple product
  - Compute $\boldsymbol{c} \otimes \boldsymbol{c}' = (c_{i,j})_{0 \le i,j \le n}$ where $c_{i,j} = c_i \cdot c_j'$.
  - Encryption of mm', under secret $(s_i \cdot s_j)_{0 \le i,j \le n}$.
- Step 2: Relinearization
  - Need a key-switching key for $s_i \cdot s_j$
  - Combine public keys of P_i and P_j to relinearize $c_{i,j}$:

$$\left((c_{i,j} \boxdot \boldsymbol{b}_j) \boxdot \boldsymbol{v}_i,\ (c_{i,j} \boxdot \boldsymbol{b}_j) \boxdot \boldsymbol{u}_i,\ c_{i,j} \boxdot \boldsymbol{d}_i \right) \text{ under } (1, s_i, s_j).$$

  - Require a quadratic complexity with n.

- $\boldsymbol{a}$: a common random
- r_i : a second secret key
- $\boldsymbol{b}_i + s_i \cdot \boldsymbol{a} \approx \boldsymbol{0} \pmod{Q}$
- $\boldsymbol{d}_i + r_i \cdot \boldsymbol{a} \approx s_i \cdot \boldsymbol{g} \pmod{Q}$
- $\boldsymbol{v}_i + s_i \cdot \boldsymbol{u}_i \approx -r_i \cdot \boldsymbol{g} \pmod{Q}$

Motivation

- Eventually we aim to compute $(c_0^*, c_1^*, \dots, c_n^*)$ where

$$c_0 = \sum_{i,j}\left(c_{i,j} \boxdot \boldsymbol{b}_j\right) \boxdot \boldsymbol{v}_i,$$
$$c_k = \sum_j\left(c_{k,j} \boxdot \boldsymbol{b}_j\right) \boxdot \boldsymbol{u}_k + \sum_i c_{i,k} \boxdot \boldsymbol{d}_i \quad \text{for } k \neq 0.$$

- Quadratic complexity is inevitable if we compute all $c_{i,j} = c_i \cdot c_j'$.

- Can we relinearize this term directly from c_i and c_j' without computing $c_{i,j}$?

- It seems infeasible since it involves a gadget decomposition $h(c_{i,j})$ but h is not a homomorphism.

Homomorphic Gadget Decomp. [KKLSS22]

- **Main Idea:** the primary goal of gadget decomposition is to find a short vector in the inverse image $g^{-1}(\cdot)$.

- **Definition:** a gadget decomposition $h: R_Q \to R^k$ is called homomorphic if
$$h(a) + h(b) \in g^{-1}(a+b),\ \ h(a) \circ h(b) \in g^{-1}(ab) \text{ for all } a, b.$$

- It is a fascinating fact that the RNS-based decomposition is homomorphic!
    - From the property that $[a]_{q_i} \cdot [b]_{q_i} = ab \pmod{q_i}$.

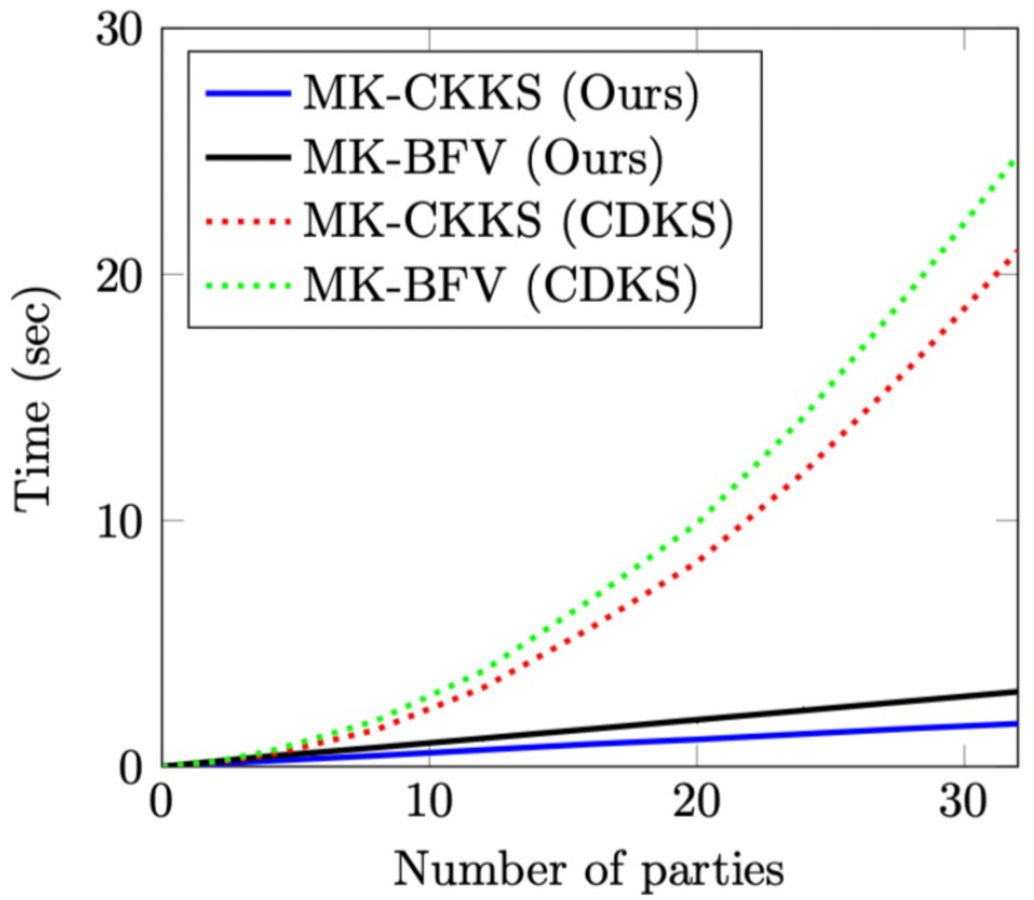
Implication

- Suppose that h is homomorphic

- In the relinearization, we substitute $h(c_{i,j})$ with $h(c_i) \circ h(c'_j)$. Then:
  - $\sum_i c_{i,k} \boxdot \boldsymbol{d}_i = \sum_i h(c_{i,k}) \cdot \boldsymbol{d}_i$ becomes $\sum_i (h(c_i) \circ h(c'_k)) \cdot \boldsymbol{d}_i = h(c'_k) \cdot (\sum_i h(c_i) \circ \boldsymbol{d}_i)$
  - Here $(\sum_i h(c_i) \circ \boldsymbol{d}_i)$ is independent from k, so is pre-computable).
  - A similar can be done for $\sum_j c_{k,j} \boxdot \boldsymbol{b}_j = \sum_j h(c_{k,j}) \cdot \boldsymbol{b}_j$.

Results & Other Issues

- We achieve a linear complexity (asymptotically optimal)
- Applying it to BFV is not straightforward (due to the unnatural tensor product), but still possible.
- The new multiplication introduces a larger error, but there is an easy fix.



Conclusion

- ThHE / MPHE / MKHE / MGHE techniques have developed significantly.
- The need is acute & fast enough to be useful.
- It is time to put these tools into practice!