A Unified Framework of Homomorphic Encryption for Multiple Parties with Non-Interactive Setup

Hyesun Kwak, Dongwon Lee, and Yongsoo Song (Seoul National University)
Sameer Wagh (Devron)

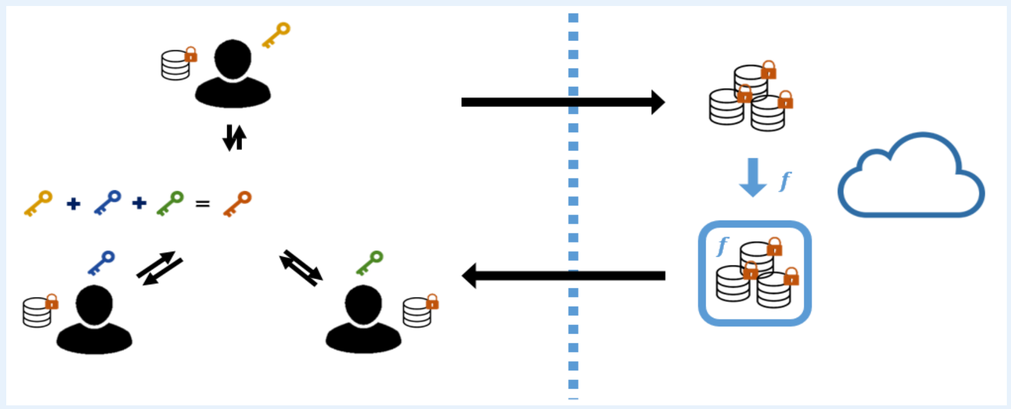
Cryptography&Privacy Lab
Dept. of Computer Science and Engineering, Seoul National University

Multi-Party Homomorphic Encryption & Multi-Key Homomorphic Encryption

- Multi-Party Homomorphic Encryption (MPHE)

Multiple parties work collaboratively to generate a joint public key and the joint secret key is (additively) shared among them.

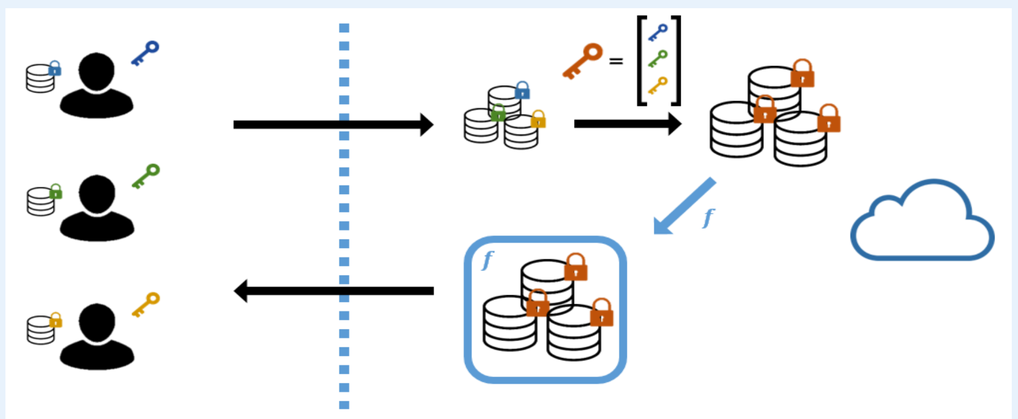
(+) Efficient computation.
(-) The set of parties should be determined in the setup phase & a multi-round protocol for a joint public key generation.



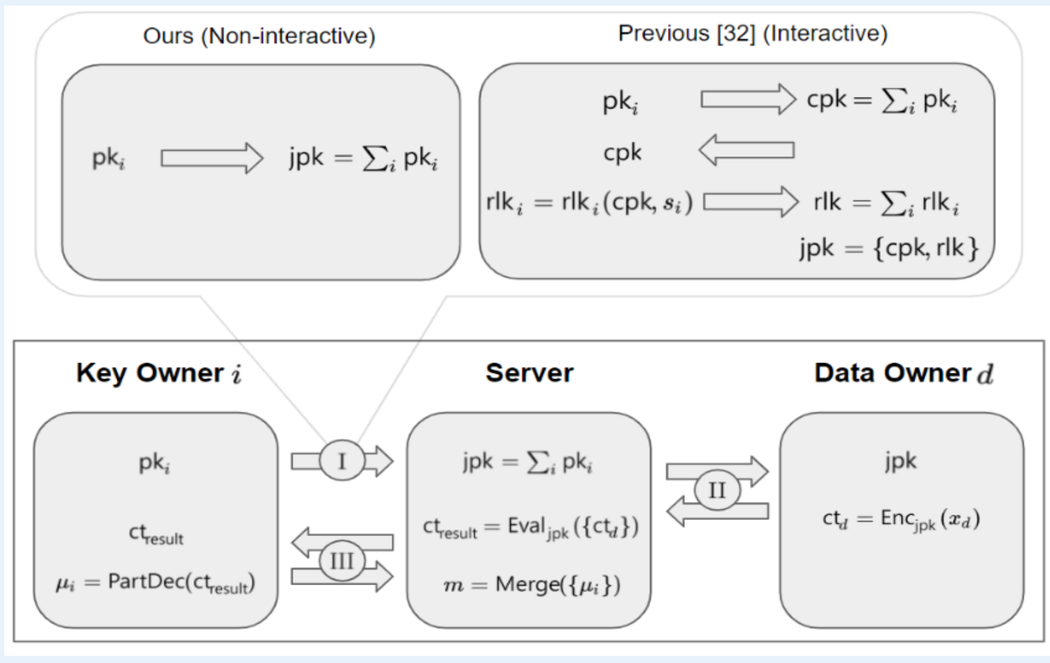
- Multi-Key Homomorphic Encryption (MKHE)

An MKHE scheme allows to generate secret and public keys without any knowledge of other parties. It supports homomorphic operations of ciphertexts encrypted under different keys.

(+) Supports operations between ciphertexts under different keys & a new party can join the computation anytime.
(-) Space and time complexities increase as the number of parties increases.



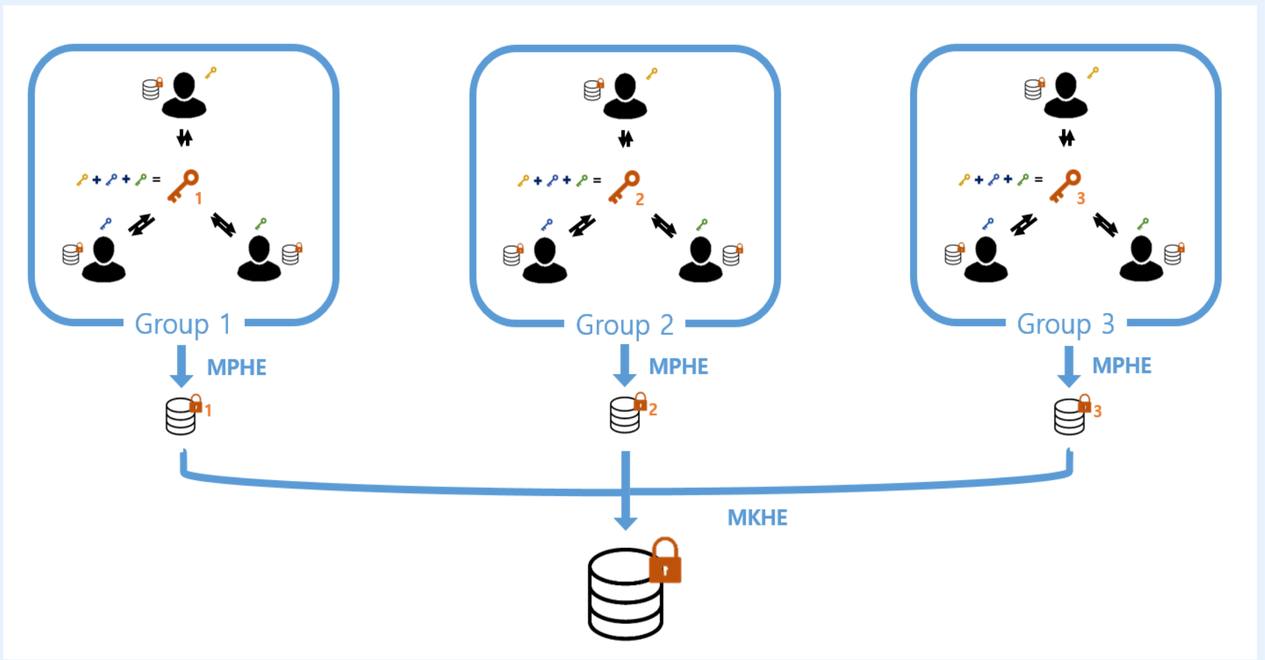
Contribution 1 Non-Interactive Setup MPHE



Previous MPHE constructions involve a multi-round key generation protocol to generate the relinearization key due to its non-linear structure with respect to the joint secret key. Modifying the public key to have a nearly linear structure, our MPHE scheme allows each party to independently generate and broadcast its public key only once, which adds up to the joint public key.

Contribution 2 Unified Framework for Multiple Parties

An MGHE scheme provides seamless integration between MPHE and MKHE, combining the best of both these primitives. In an MGHE scheme, a group of parties generates a public key jointly which results in compact ciphertexts and efficient homomorphic operations, similar to MPHE. However, unlike MPHE, it also supports computations on encrypted data under different keys (from different groups), a property enjoyed by MKHE schemes.



Experiments

The execution time depends on the dimension of the base ring and the number of groups participating in the evaluation. In addition, the number of parties in groups does not affect the execution time in both schemes.

We observe that some terms that appear in the process of relinearization are pre-computable and reusable. This idea consequently reduces the number of external products down to $2k^2 + 2k$ in total, compared to the former method which requires $4k^2$ external products.

n	k	Mult + Relin				Auto			
		BFV		CKKS		BFV		CKKS	
		Ours	[4]	Ours	[4]	Ours	[4]	Ours	[4]
2^{14}	1	100	110	51	59	21	22	25	24
	2	206	257	122	165	42	47	47	49
	4	514	717	331	521	81	88	92	95
	8	1,490	2,350	1,018	1,845	160	176	178	193
2^{15}	1	651	675	427	465	161	170	170	172
	2	1,443	1,715	1,035	1,364	317	333	333	359
	4	3,731	5,025	2,874	4,287	631	646	671	711
	8	11,425	17,450	9,040	15,159	1,303	1,332	1,338	1,413

Performance of MGHE and MKHE by Chen et al. (CDKS19): execution times to operate homomorphic multiplication (Mult + Relin) and automorphism (Auto), taken in milliseconds (ms). In the table, ‘n’ denotes the dimension of base ring and ‘k’ denotes the number of the associated groups (keys) to the ciphertext.

Applications

MGHE is to enable a secure workflow of machine learning models comprising the privacy of multiple data owners. If data owners can be determined before training, the MPHE scheme would be a reasonable solution for privacy-preserving training of the model because of the performance. In the case of inference, however, the client may not be determined aforehand and the model may deal with multiple independent clients. Thus, it is more reasonable to perform inference using an MKHE scheme which shows better flexibility.