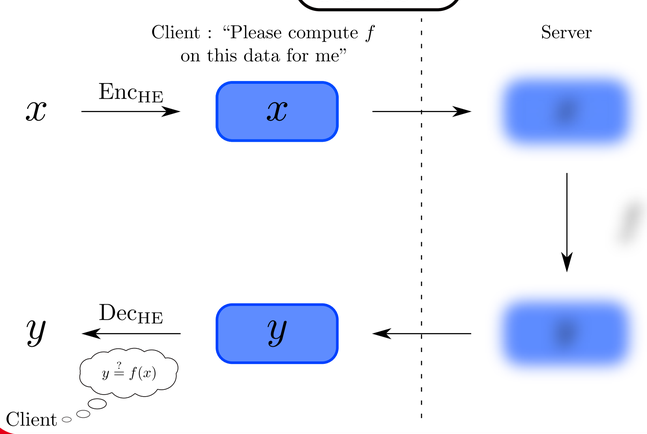
Towards Practical FHE Calculation Integrity by Means of Rinocchio

Marc Renard[1], Renaud Sirdey[1], Caroline Fontaine[2], Oana Stan[1]
[1]Université Paris-Saclay, CEA-List,Palaiseau, France
[2]Université Paris-Saclay, CNRS, ENS Paris-Saclay, Gif-sur-Yvette, France
[1]{name.surname}@cea.fr [2]{name.surname}@cnrs.fr

Context



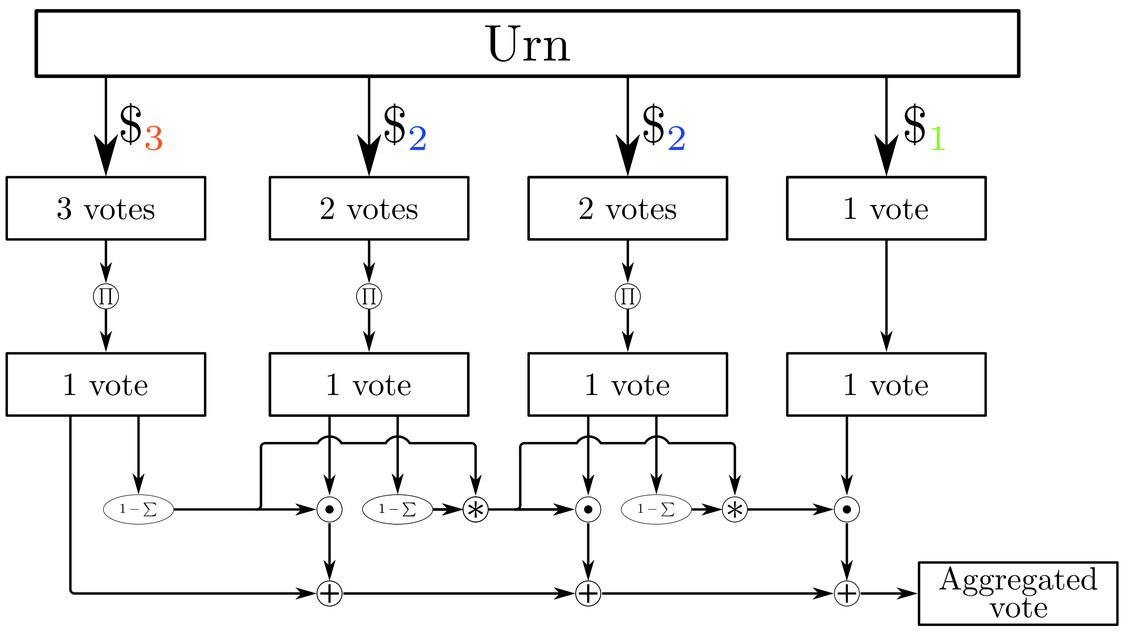
First motivations

FHE cryptosystems do not provide natively integrity guarantees. [1] and [2] proposed two approaches to solve this issue. In this work, we have focused on the solution proposed in [1]. We decided to use the Rinocchio zk-SNARK on the SHIELD algorithm [3], an approximate majority voting operator used in federated learning.

vCCA motivations

[4] recently introduced a new security notion achievable for FHE and using integrity checking. This is the highest security level known to date achievable for FHE. Consequently, this work can be viewed as a first attempt to instantiate one of the constructions proposed in [4] on a non trivial application.

SHIELD circuit with parametrization $X^3 + 2X^2 + X$



Votes : Vectors of ciphertexts which are all encrypting 0 except one which encrypts 1

$\$_k$: Draw k random votes from the urn

⓫ : Component-by-component vector multiplication

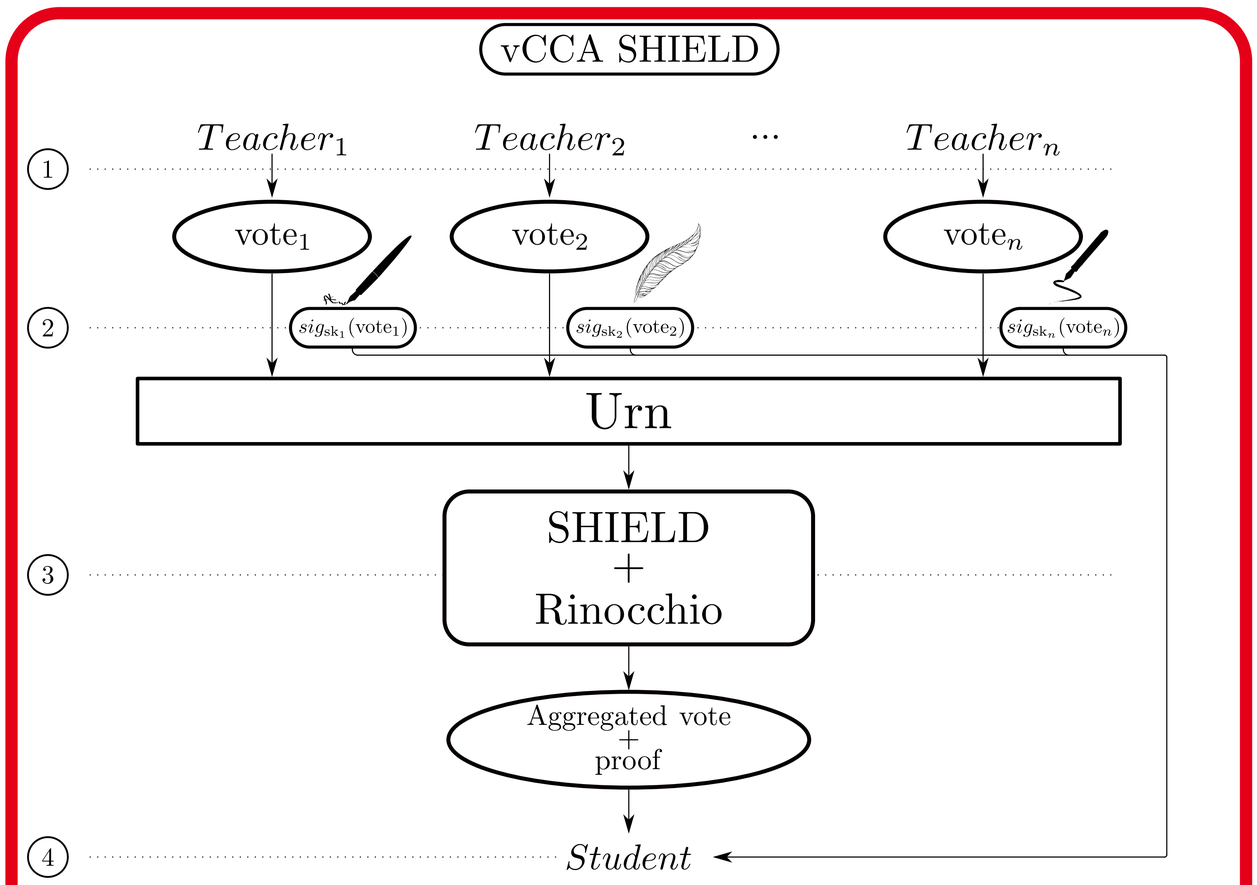
$\oplus$: Addition of vectors

$\circledast$: Product of two scalars

$1-\Sigma$: 1 minus the sum of the components of a vector

$\odot$: Product of a vector by a scalar

vCCA SHIELD



(1) Teachers encrypt their votes with a $k \geq 2$ out of n threshold homomorphic cryptosystem.

(2) They each sign their vote with their own signature key sk_i and send theses signatures to the Student.

(3) SHIELD and Rinocchio are applied.

(4) The Student verifies the proof and signatures, and with the help of at least $k-1$ Teachers he decrypt the aggregated vote.

Preliminary results

We used the implementation of Rinocchio given by [5] (based on SEAL [6]) as a starting point. We then implemented a tool to work directly over ciphertexts and then over votes.

We have run SHIELD on an urn containing votes of size 10 ciphertexts. The different results we obtained are shown in the following table:

SHIELD setting	BV params		Nb constraints	Nb polynomials	Ciphertext final size	SHIELD evaluation time	Correct decryption	Encoding parameters found	CRS		Proof		Verif
	N	$log_2(q)$							time	size	time	size	time
$X^2 + X$	2048	54 (1 prime)	173	675	4	56ms	True	True	6, 8s	771Mo	19s	14, 3Mo	2, 5s
$2X^2 + X$	2048	54 (1 prime)	438	940	6	82ms	False	True	20s	2006Mo	71s	14, 3Mo	14s
$2X^2 + X$	8192	218 (5 primes)	438	940	6	1, 3s	True	False	—	—	—	—	—
$X^3 + X$	4096	109 (3 primes)	294	796	5	265ms	True	False	—	—	—	—	—
$X^3 + X^2 + X$	8192	218 (5 primes)	603	1105	7	1, 6s	True	False	—	—	—	—	—

Conclusions and prospects

We still face some limitations. The first one is to find suitable parameters allowing to evaluate SHIELD correctly and that don't force the parameters used in Rinocchio (in the encoding scheme) to be out of scale. The second one is to deal with the relinearization which is a non arithmetic operation and therefore cannot be directly supported by Rinocchio. One solution can be the following: instead of proving the decomposition of a polynomial in a base (non arithmetic operation) before the multiplication with the relinearization keys, we can use the decomposition as a private server input and prove that: its coefficient are in the right range and the "recomposition" is correct.

[1] Ganesh, C., Nitulescu, A., Soria-Vazquez, E.: Rinocchio: SNARKs for ring arithmetic. *Journal of Cryptology* 36(4),41 (2023)
[2] Atapoor, S., Baghery,K., Pereira, H.V., Spiessens, J.: Verifiable FHE via Lattice-based SNARKs. Cryptology ePrint Archive, Paper 2024/032
[3] Grivet Sébert, A., Zuber, M., Stan, O., Sirdey, R., Gouy-Pailler, C.: When approximate design for fast homomorphic computation provides differential privacy guarantees. arXiv preprint arXiv:2304.02959 (2023)
[4] Manulis, M., Nguyen, J.: Fully Homomorphic Encryption beyond IND-CCA1 Security: Integrity through Verifiability. Cryptology ePrint Archive, Paper 2024/202 (Eurocrypt 2024)
[5] Viand, A., Knabenhans, C., Hithnawi, A.: Verifiable fully homomorphic encryption. arXiv preprint arXive:2301.07041 (2023)
[6] Microsoft SEAL (release 4.0). https://github.com/Microsoft/SEAL (2022)