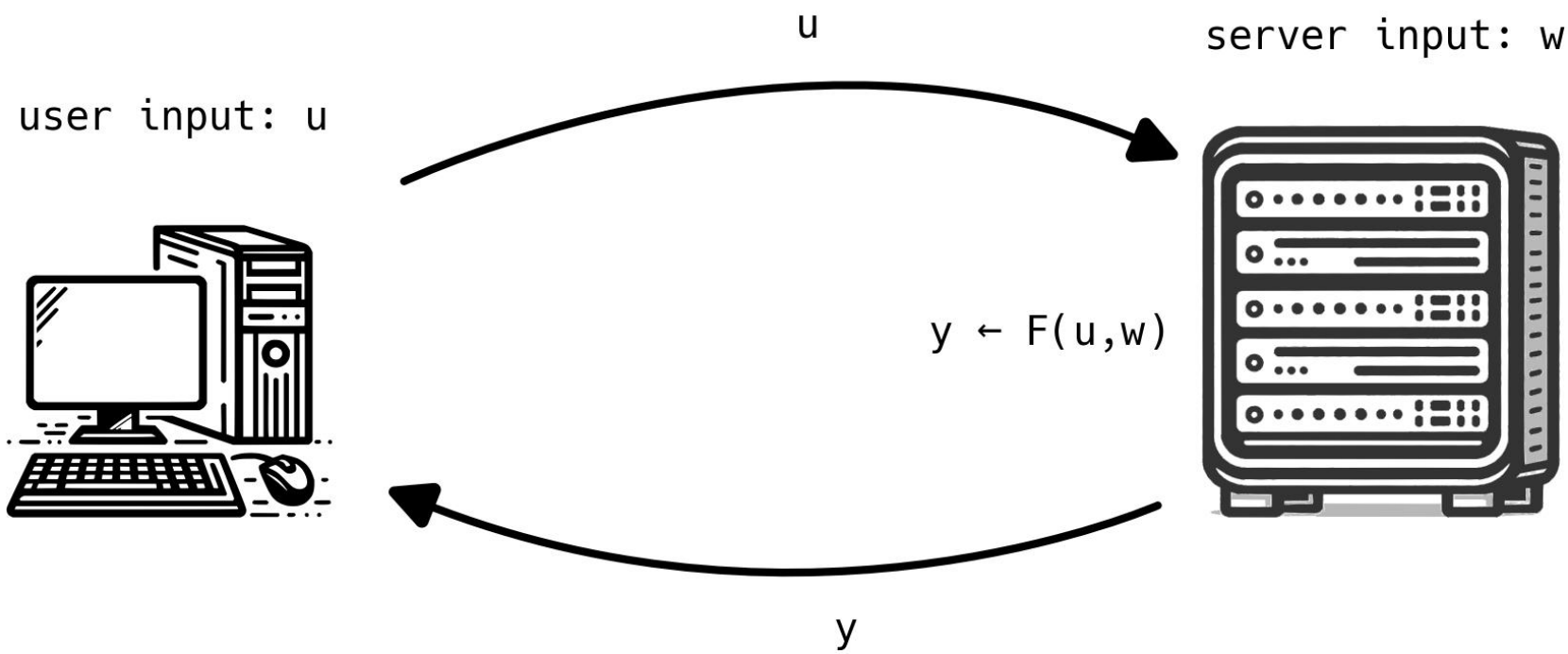
Blind zkSNARKs

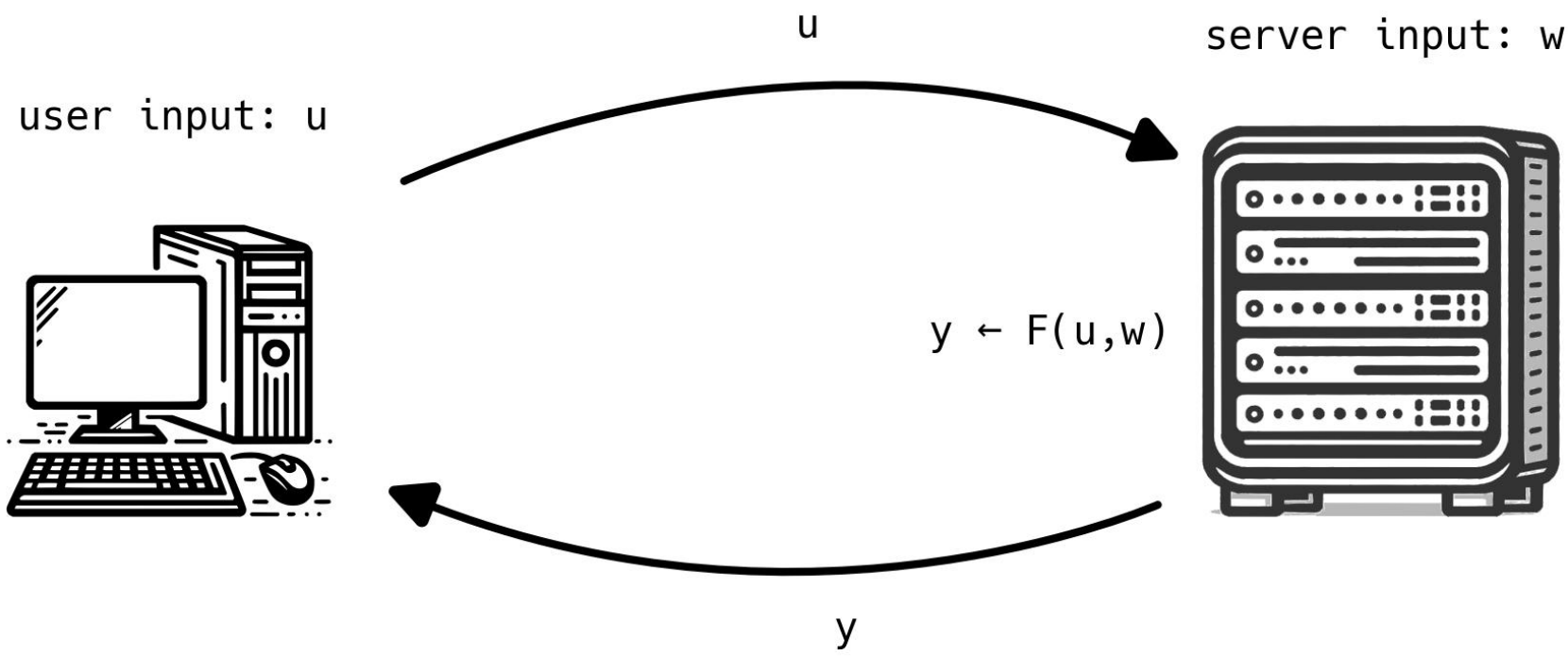
Private Proof Delegation and Verifiable Computation over Encrypted Data

Mariana Gama [1] Emad Heydari Beni [1,2] Jiayi Kang [1]
Jannik Spiessens [1] Frederik Vercauteren [1]

[1]COSIC, KU Leuven [2]Nokia Bell Labs

ia.cr/2024/1684


u
server input: w
user input: u
y ← F(u,w)
y

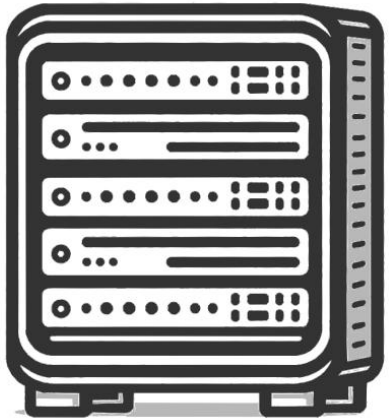


What if server is dishonest?

u

server input: w

user input: u



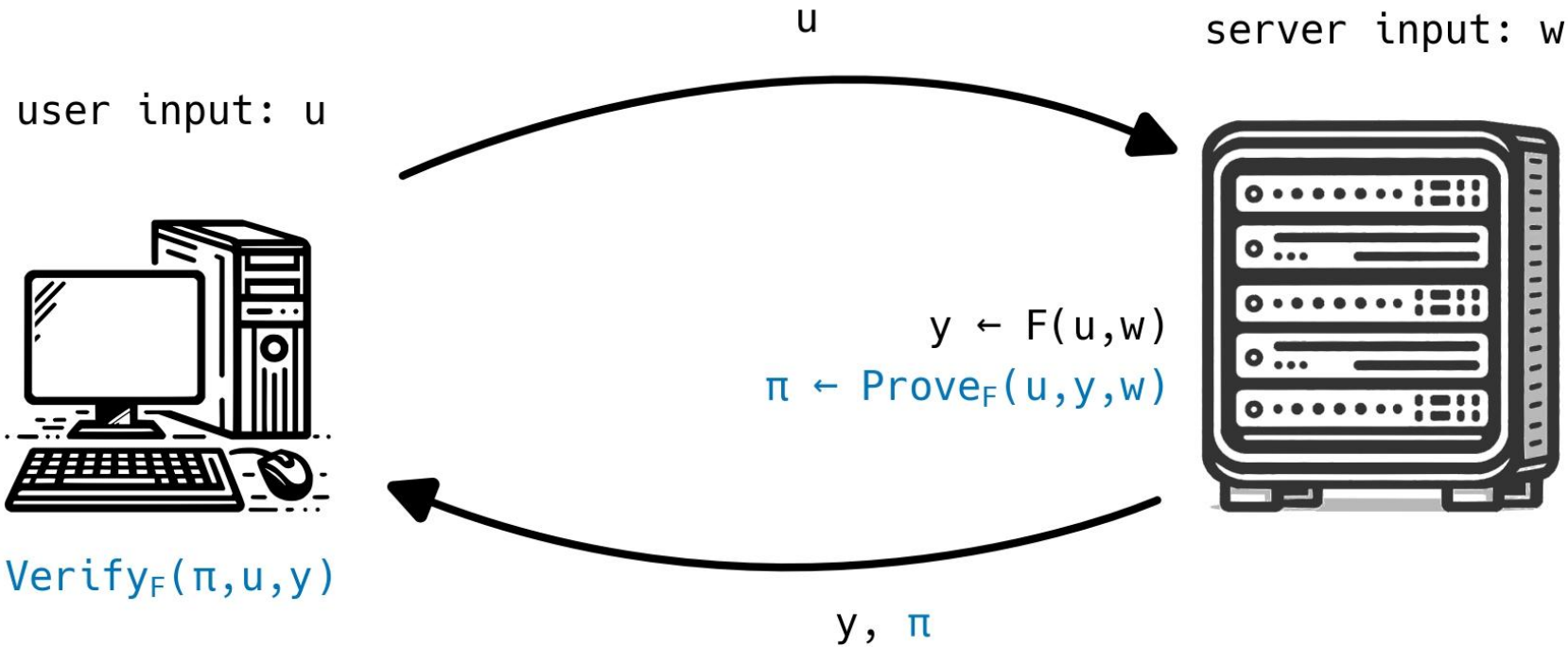
$y \leftarrow F(u,w)$

$\pi \leftarrow \text{Prove}_F(u,y,w)$

$\text{Verify}_F(\pi,u,y)$

y, π

Verifiable Computation (VC)



Intermezzo: Homomorphic Encryption (HE)

- Encryption scheme

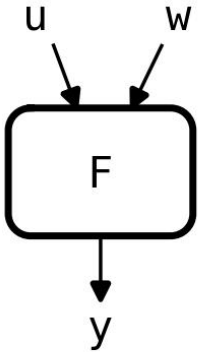
$$\texttt{ct[u]} \leftarrow \texttt{Enc(u)}$$
$$\texttt{u} \leftarrow \texttt{Dec(ct[u])}$$

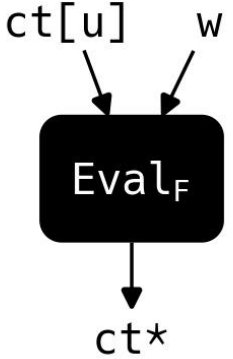
- Homomorphic

$$\texttt{u + w} = \texttt{Dec(ct[u] + ct[w])}$$
$$\texttt{u * w} = \texttt{Dec(ct[u] x ct[w])}$$

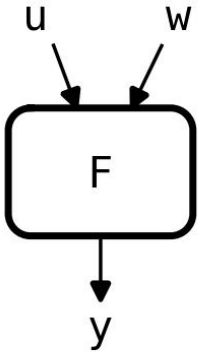
Intermezzo: Homomorphic Encryption (HE)

- Transforms arithmetic circuit F into homomorphic circuit $\mathtt{Eval}_F$



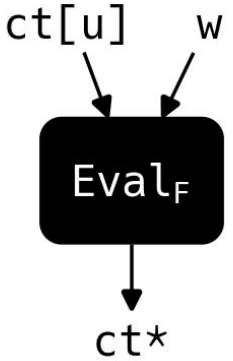


Intermezzo: Homomorphic Encryption (HE)

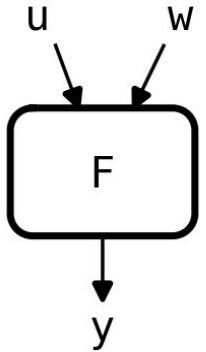


- Transforms arithmetic circuit F into homomorphic circuit $\mathtt{Eval}_F$
- Correctness:

$$\left.\begin{array}{rl} y & \leftarrow F(u,w) \\ ct[u] & \leftarrow Enc(u) \end{array}\right\} \Rightarrow y \leftarrow Dec(ct[y])$$



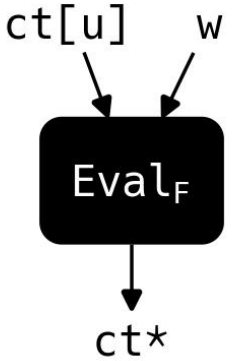
Intermezzo: Homomorphic Encryption (HE)

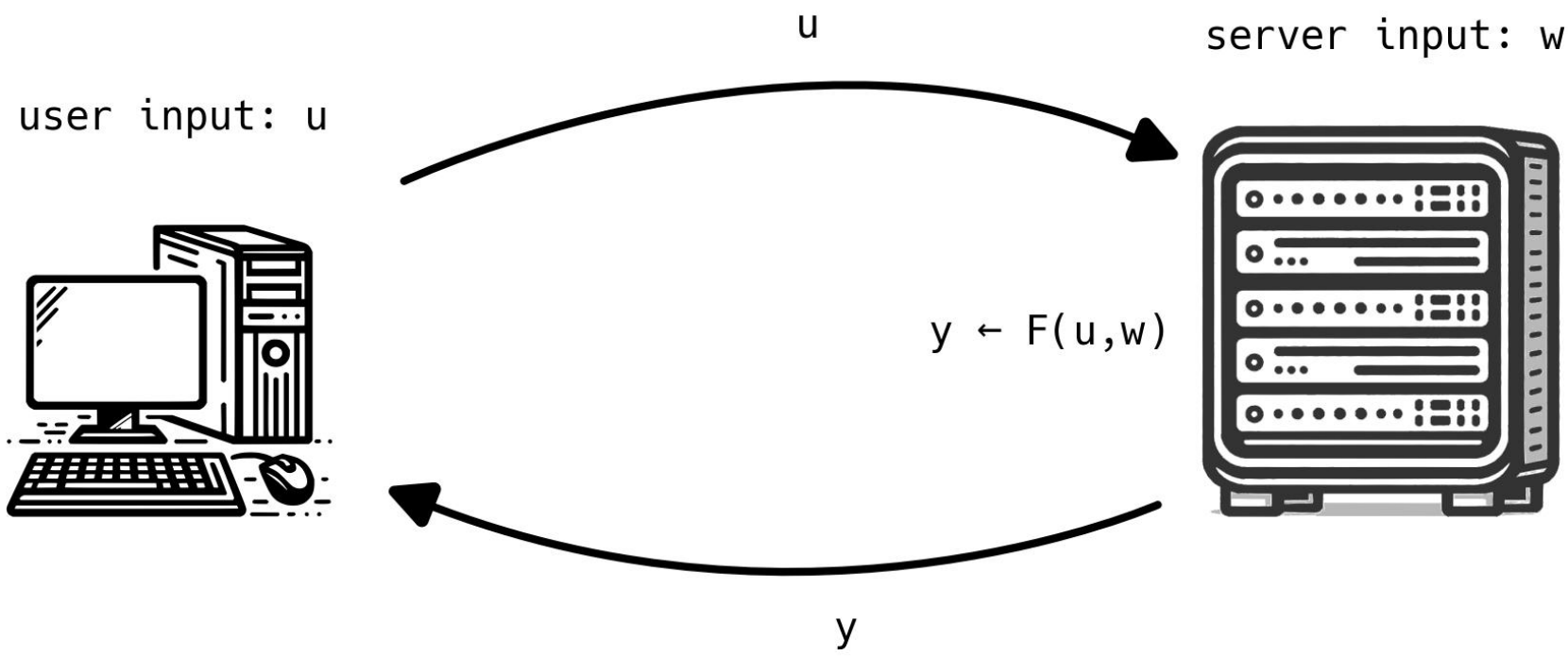


- Transforms arithmetic circuit F into homomorphic circuit $\mathtt{Eval}_F$
- Correctness:

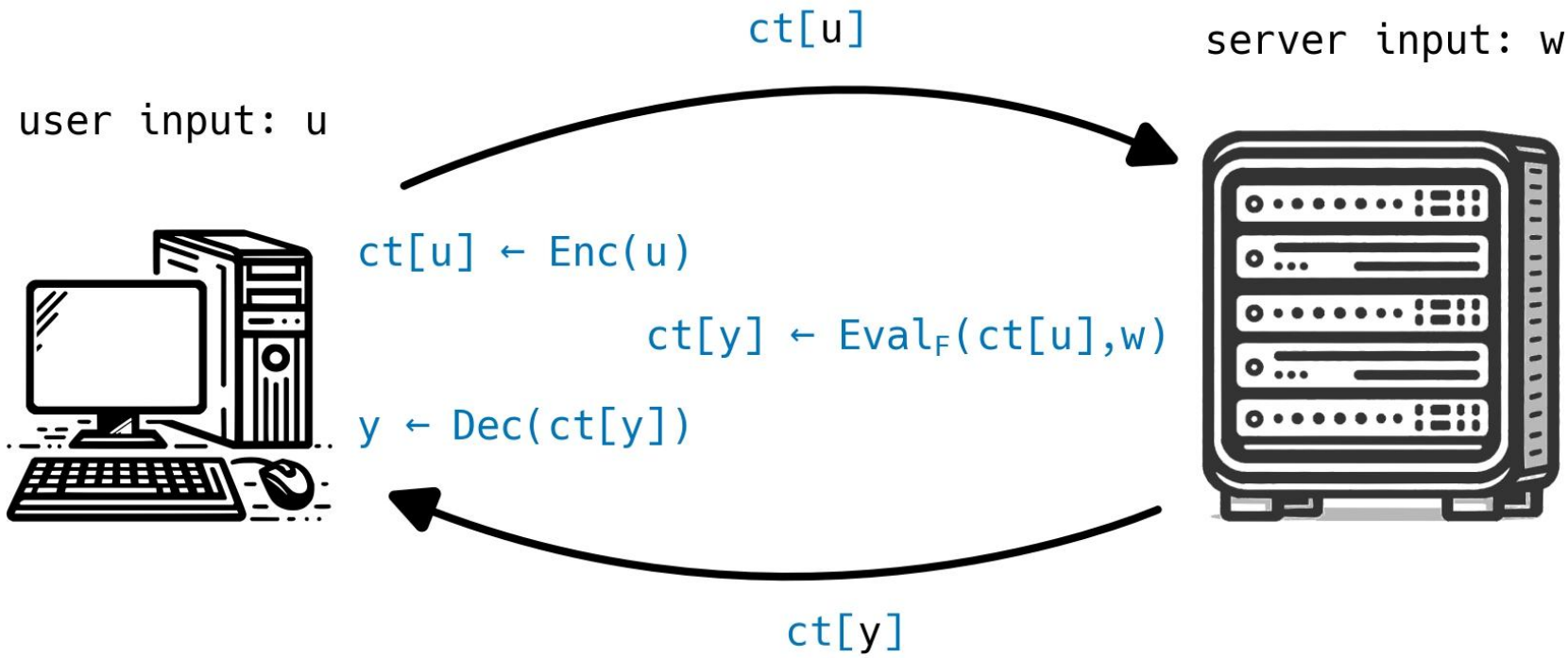
$$\left.\begin{array}{rl} y & \leftarrow F(u,w) \\ ct[u] & \leftarrow Enc(u) \end{array}\right\} \Rightarrow y \leftarrow Dec(ct[y])$$

- Encryption/Decryption using secret key sk

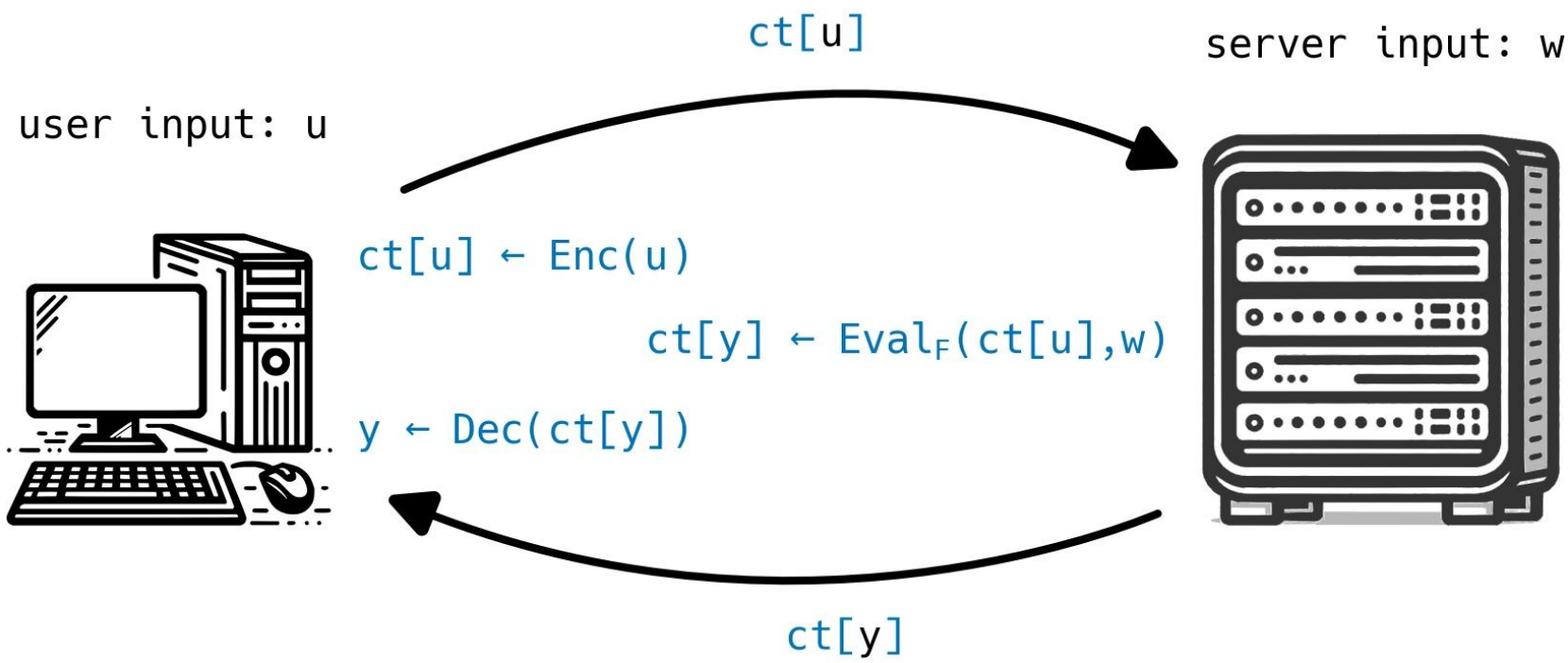


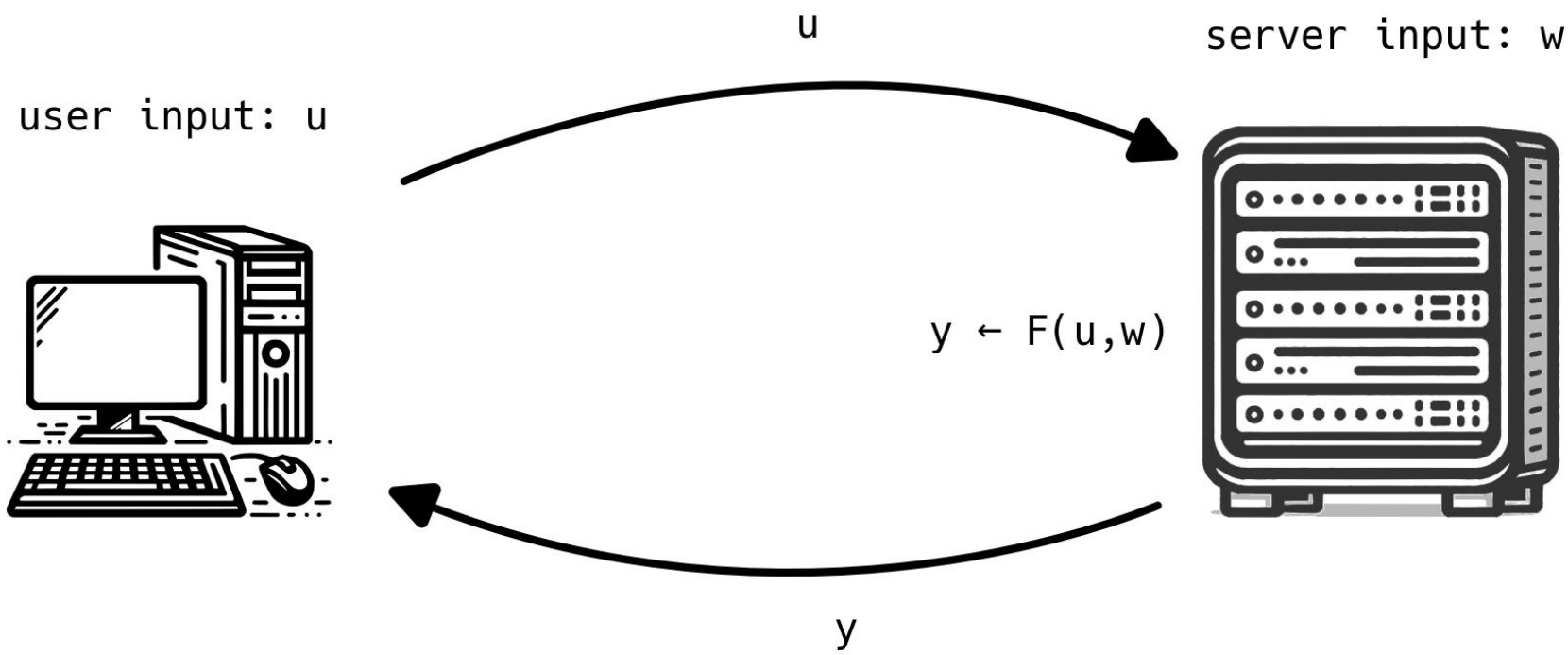


What if server is curious?

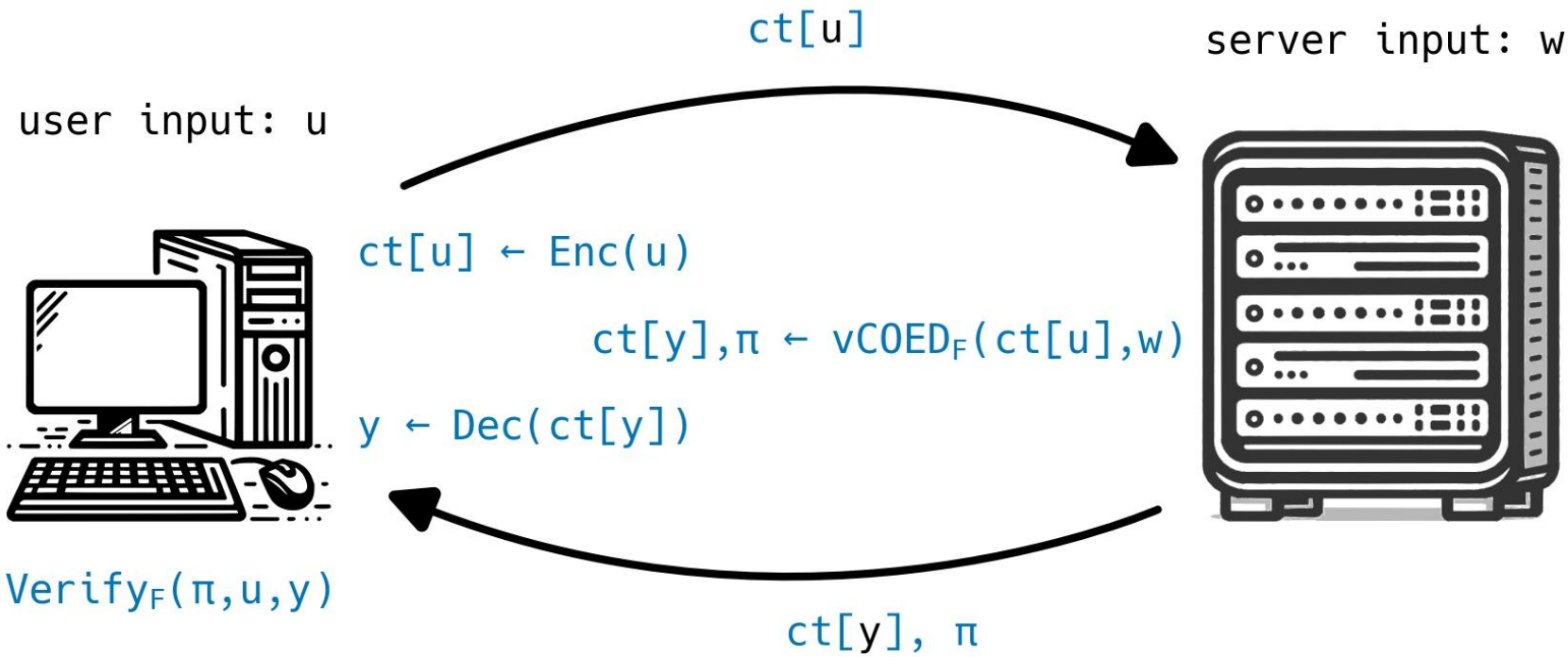

ct[u]
server input: w
user input: u
ct[u] ← Enc(u)
ct[y] ← Eval_F(ct[u],w)
y ← Dec(ct[y])
ct[y]

Computation Over Encrypted Data (COED)

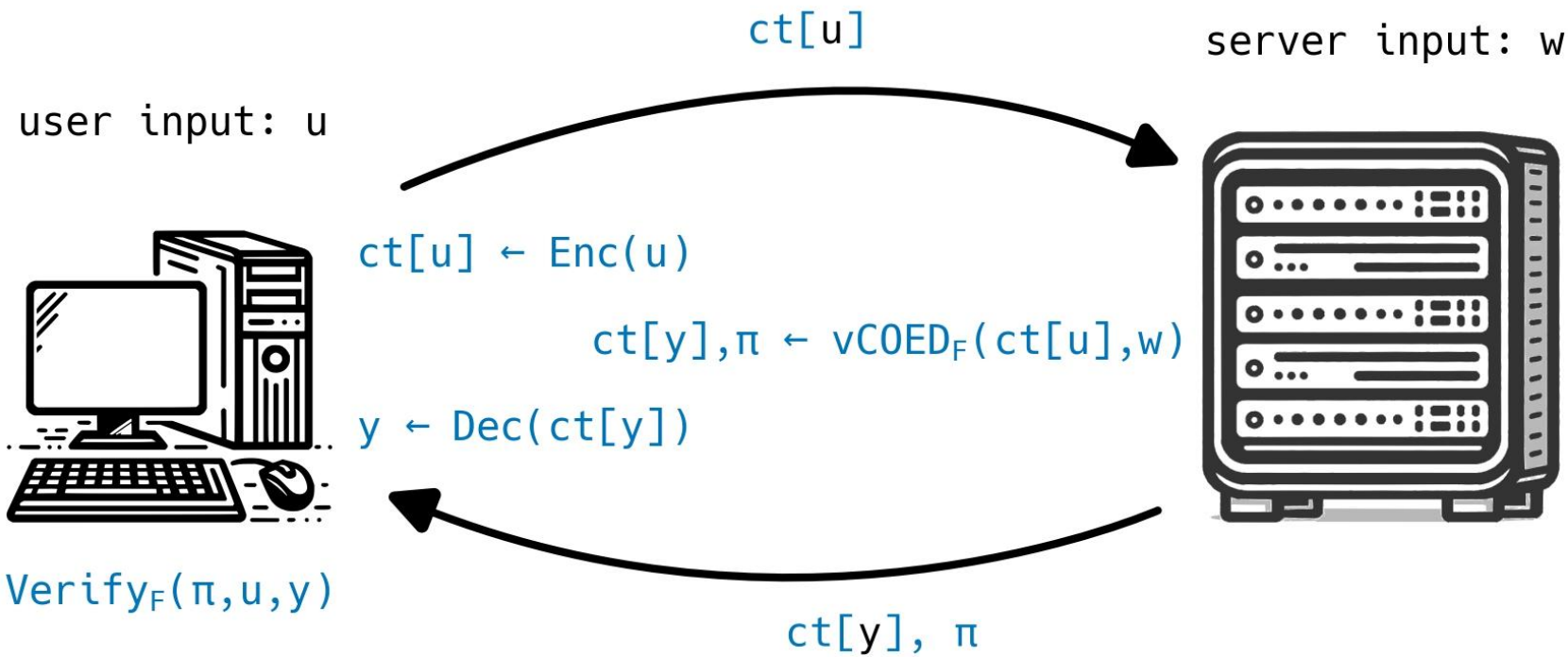




What if server is dishonest **and** curious?


ct[u]
server input: w
user input: u
ct[u] ← Enc(u)
ct[y],π ← vCOED_F(ct[u],w)
y ← Dec(ct[y])
Verify_F(π,u,y)
ct[y], π

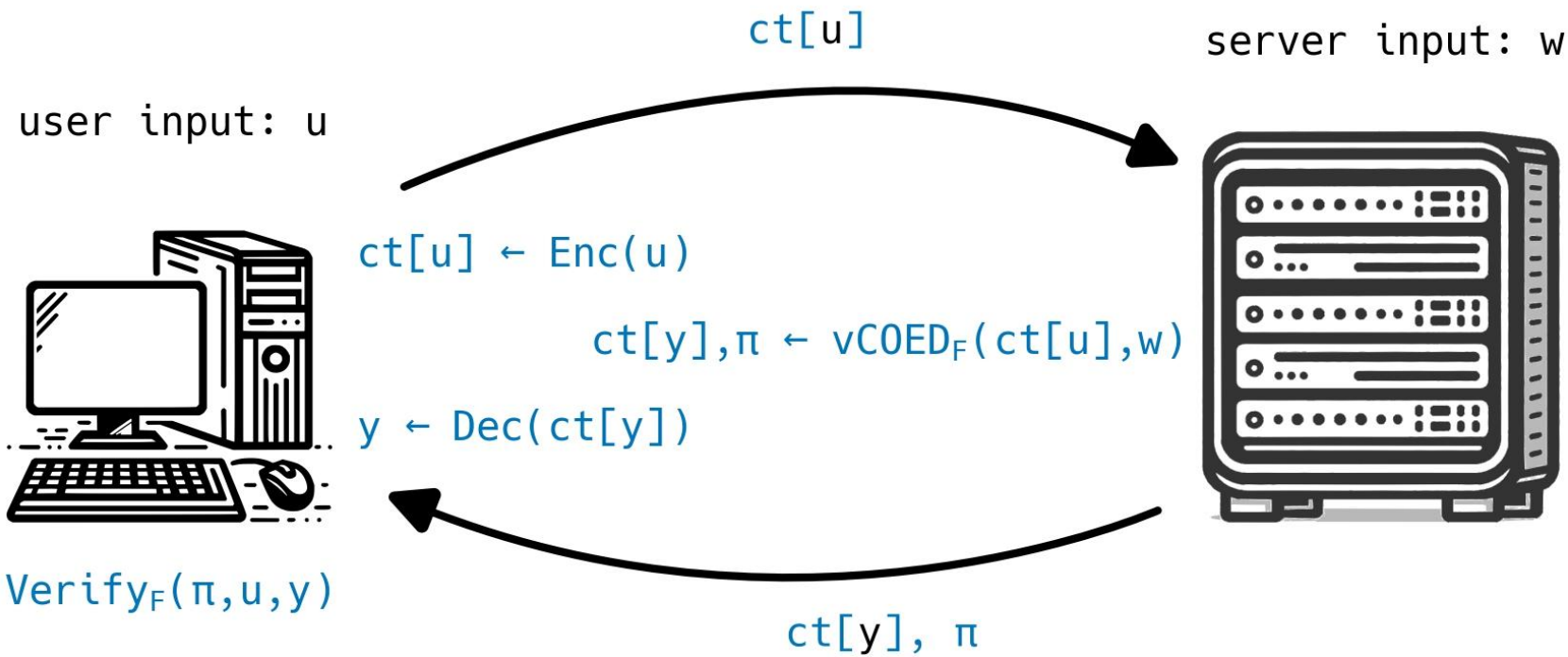
Verifiable COED (vCOED)



Verifiable COED (vCOED)

> introduced in [FGP14]

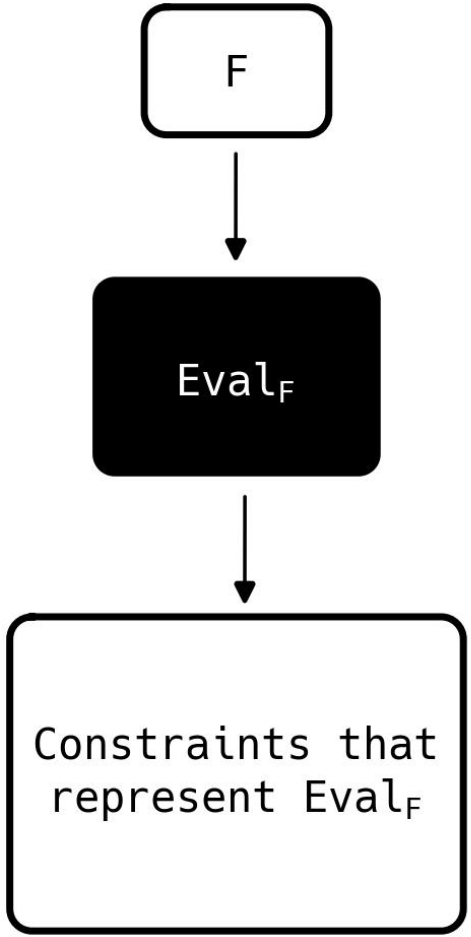
> Cryptographic TEE



vCOED from Verifiable FHE (vFHE)

> Prove homomorphic computations
>
> i.e. $\mathtt{vCOED}_F \leftarrow \mathtt{Prove}_{\mathtt{Eval}_F}$

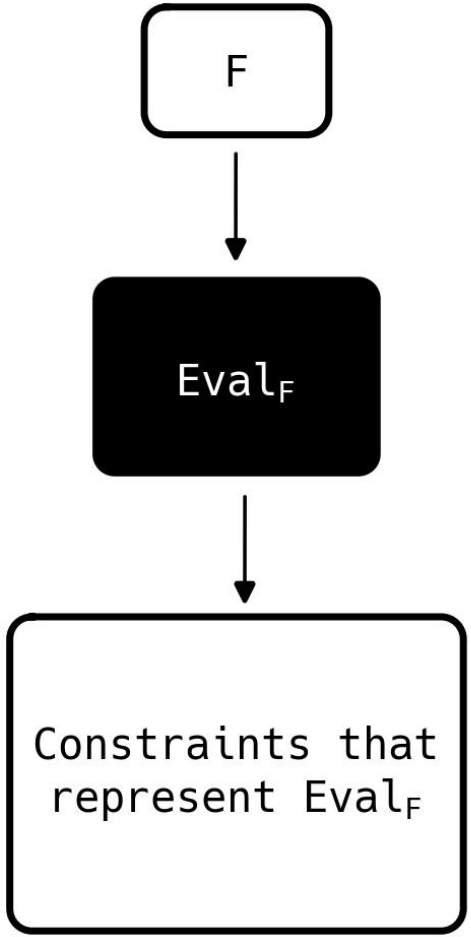
1. Construct homomorphic circuit $\mathtt{Eval}_F$
2. Represent $\mathtt{Eval}_F$ as a constraint system
3. Prove that constraint system using VC



Why vFHE is so hard

- Circuit size blowup in step 1
  - Operations in $\mathbb{F}_p$ become operations in $\mathbb{Z}_Q/(X^N+1)$
- Circuit size blowup in step 2
  - Maintenance operations require proving non-native arithmetic
- State-of-the-art:
  - F is a small circuit and/or has a small plaintext space

[VKH23][Ata+24][Wal24][Liu+25][Cas+25]



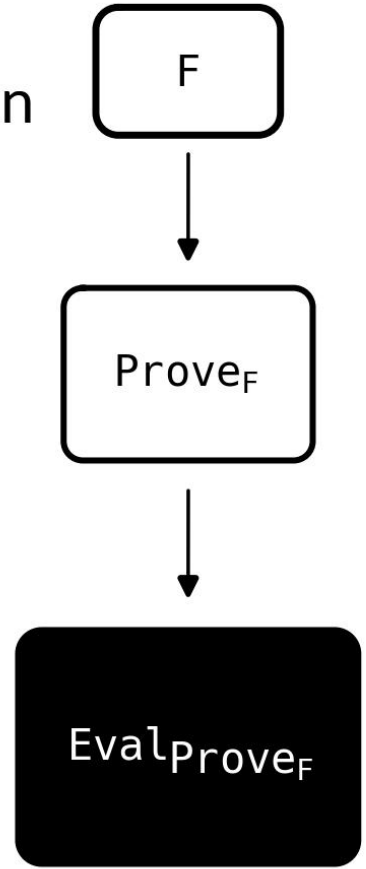
The opposite approach

- Homomorphically compute proving computation
  i.e. $\mathtt{vCOED}_{\mathsf{F}} \leftarrow \mathtt{Eval}_{\mathtt{Prove}_{\mathsf{F}}}$
- *Only proves plaintext operations*

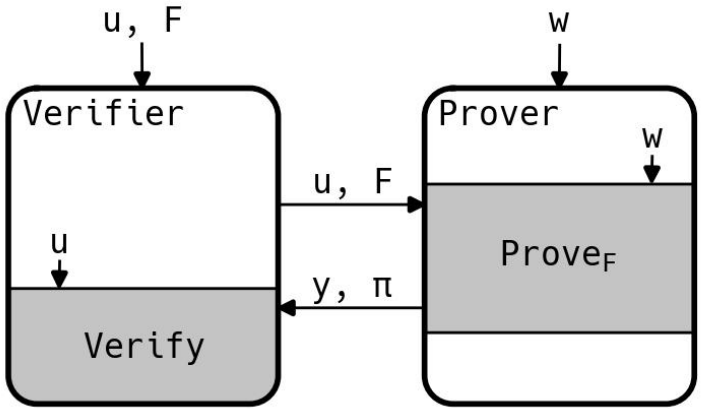
  e.g. $\mathsf{F} : \mathsf{y} = \mathsf{u} + \mathsf{w}$

  $$\mathtt{ct}[\mathsf{y}] = \begin{cases} \mathtt{ct}[\mathsf{u}] + \mathtt{ct}[\mathsf{w}] \\ \mathtt{ct}[\mathsf{u}] + \mathtt{ct}[\mathsf{w}] + \mathtt{ct}[0] \end{cases}$$ OR

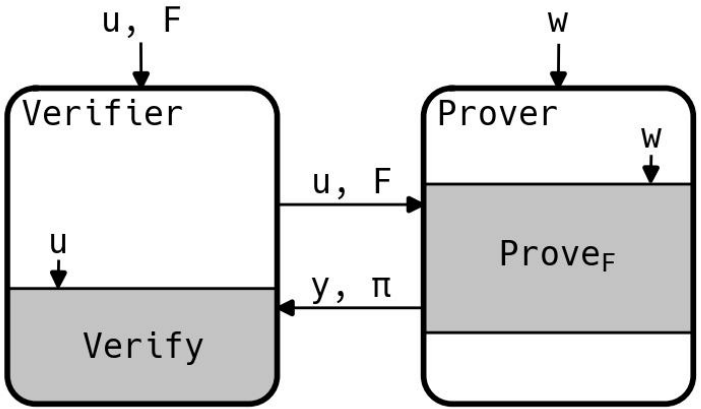
  can both construct valid proof!
- Related work: [GGW24][Ara+24][Zha+25]

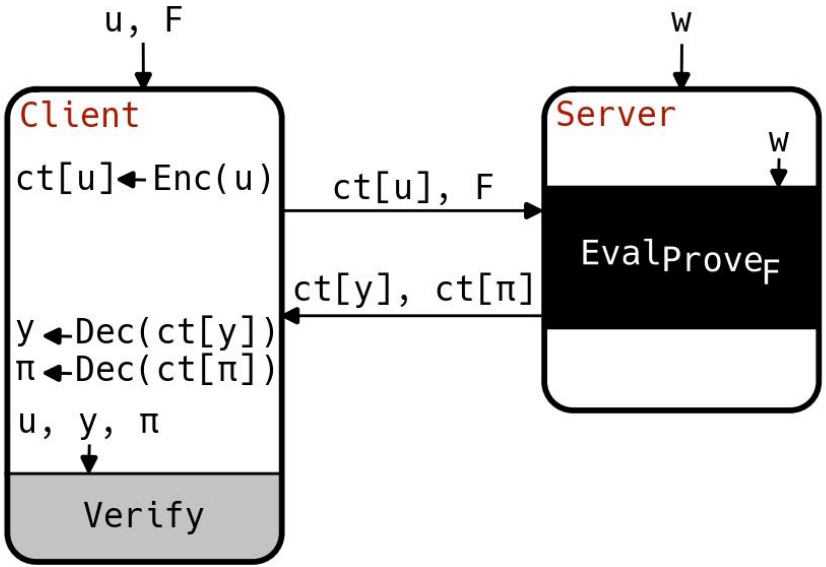


Blind zkSNARK for vCOED

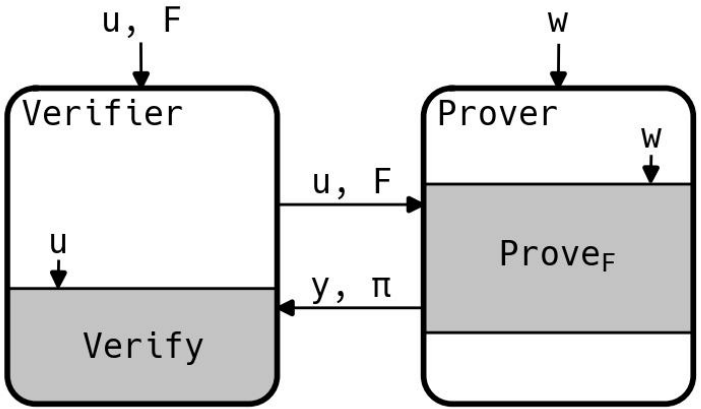


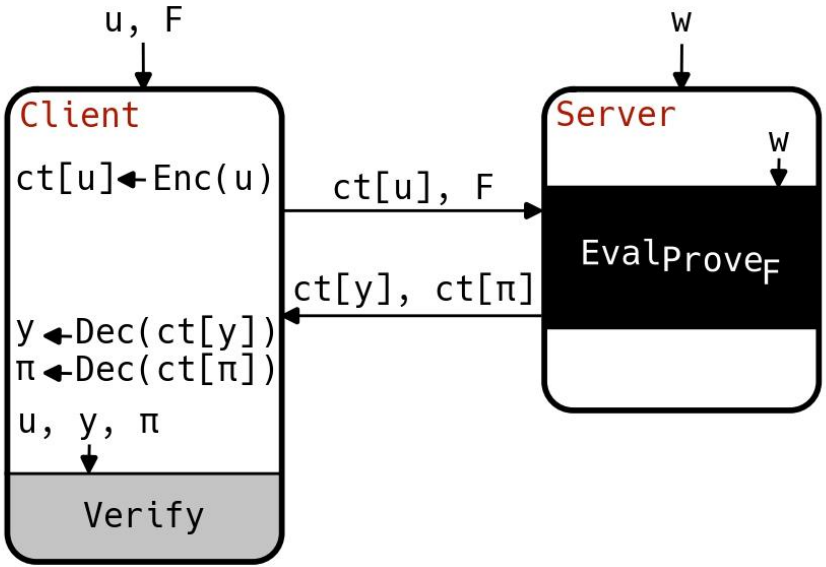
Blind zkSNARK for vCOED





Blind zkSNARK for vCOED

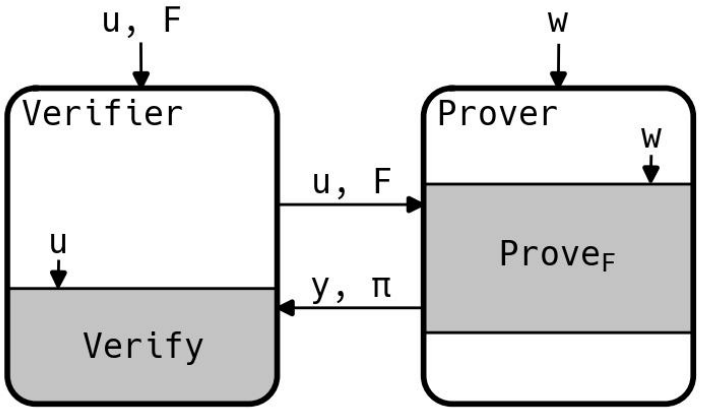




- Server returns partly encrypted proof ct[π]
- Designated-verifier

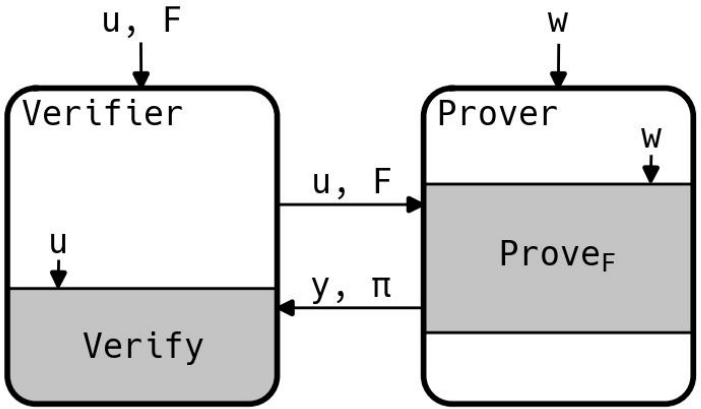
Blind zkSNARK for zkDel

zkDel =
Private Proof Delegation
to a single party



Blind zkSNARK for zkDel

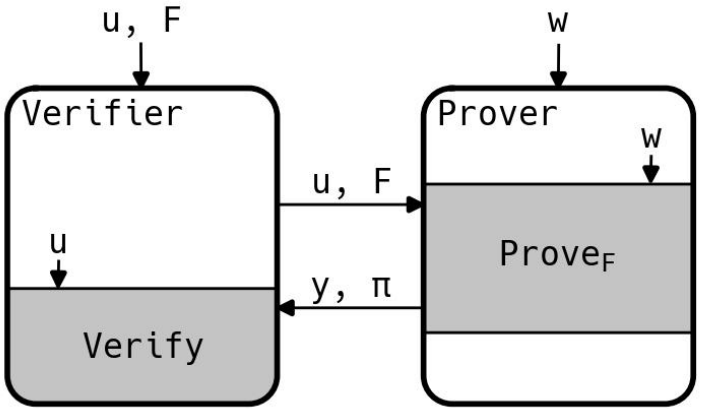
zkDel =
Private Proof Delegation
to a single party

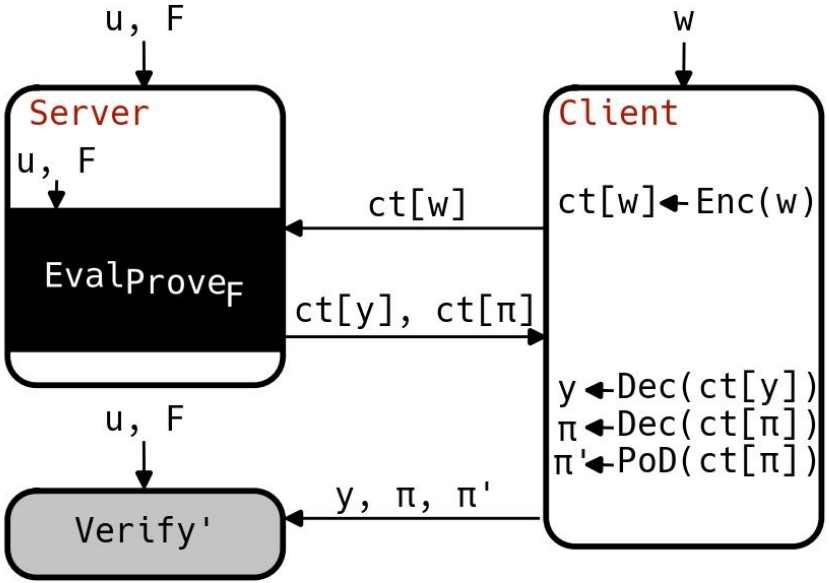


- What if the client is the prover?
- Privacy Enhancing Technologies (PETs)
  - anonymous credentials
  - private cryptocurrencies
  - …
- Computing $\mathtt{Prove}_F$ can be expensive

Blind zkSNARK for zkDel

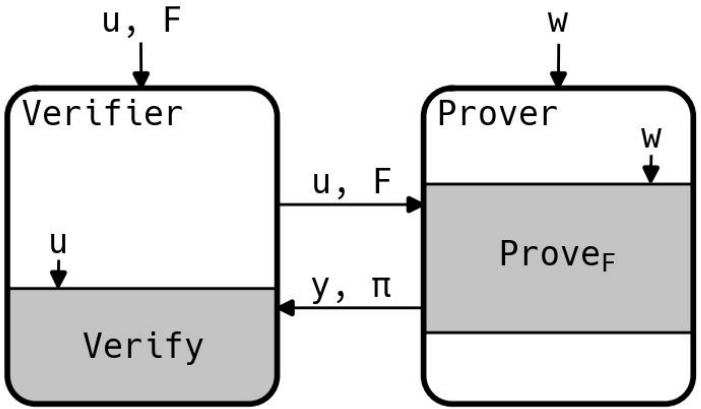
zkDel =
Private Proof Delegation
to a single party

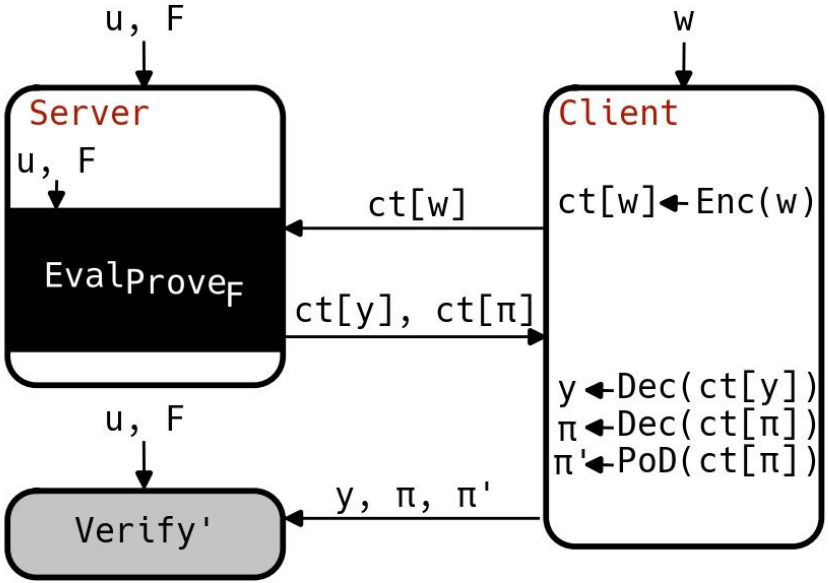




Blind zkSNARK for zkDel

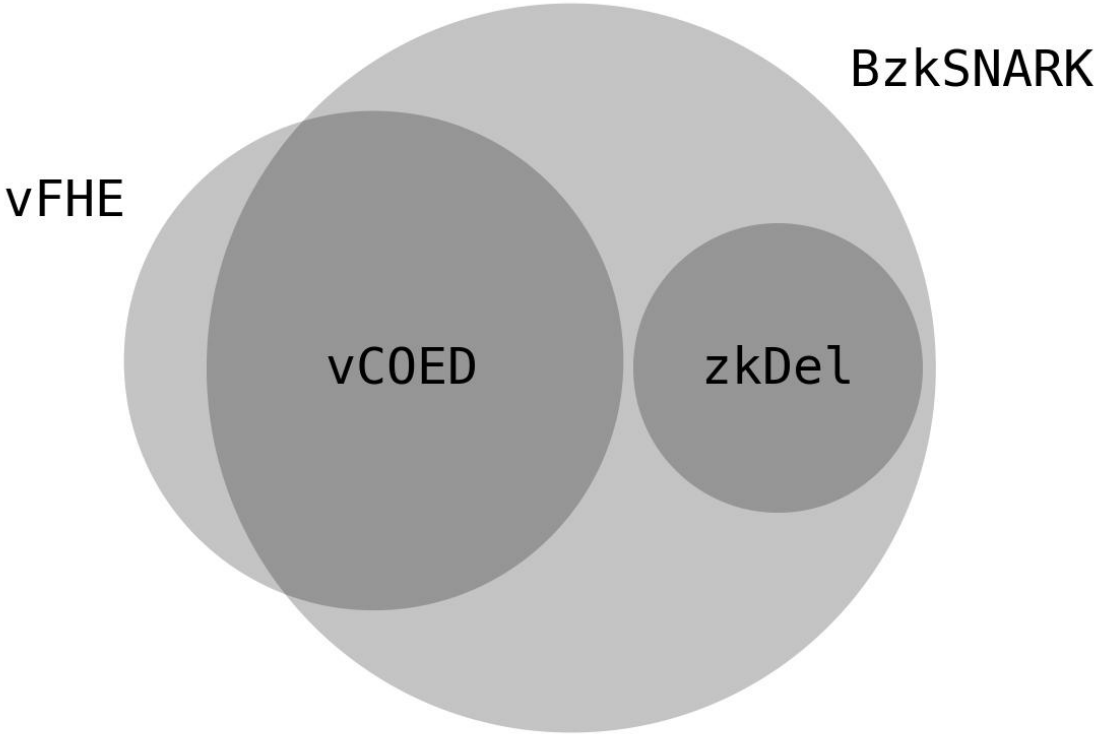
zkDel =
Private Proof Delegation
to a single party





- Server returns partly encrypted proof `ct[π]`
- Client transforms into publicly verifiable proof
  - by appending a **Proof of Decryption (PoD)**

A summary of BzkSNARK applications



Making proof systems blind

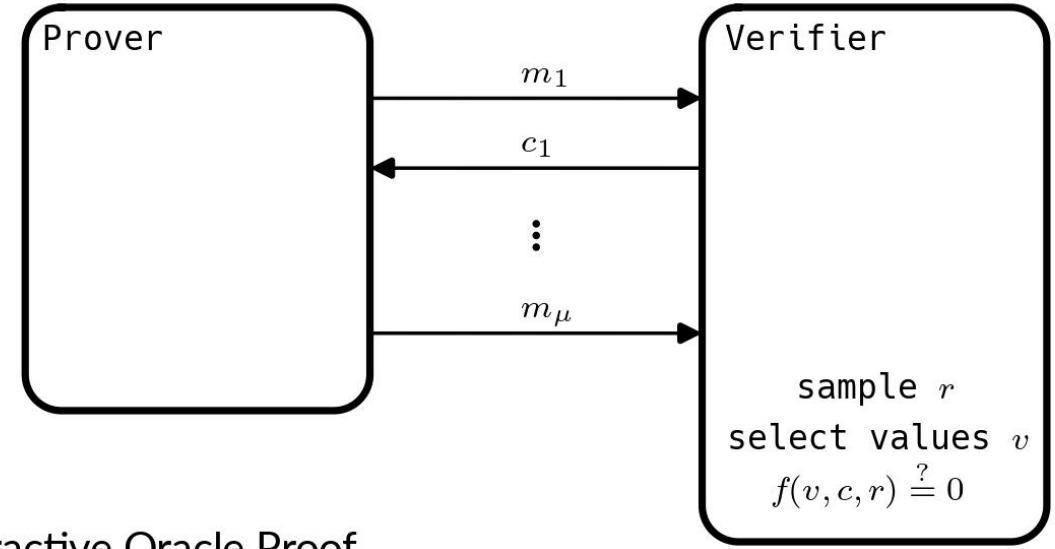
Proof

- proves that $(x, w) \in R$
- implies knowledge
- native server-privacy

Blind proof

- proves $(\mathsf{C}_{\mathsf{sk}}, \mathtt{ct}[x], \mathtt{ct}[w]) \in \mathcal{E}[R]$ $\Rightarrow (\mathtt{Dec}_{\mathsf{sk}}(x), \mathtt{Dec}_{\mathsf{sk}}(w)) \in R$
- implies *plaintext* knowledge
- server-privacy requires circuit-private HE scheme

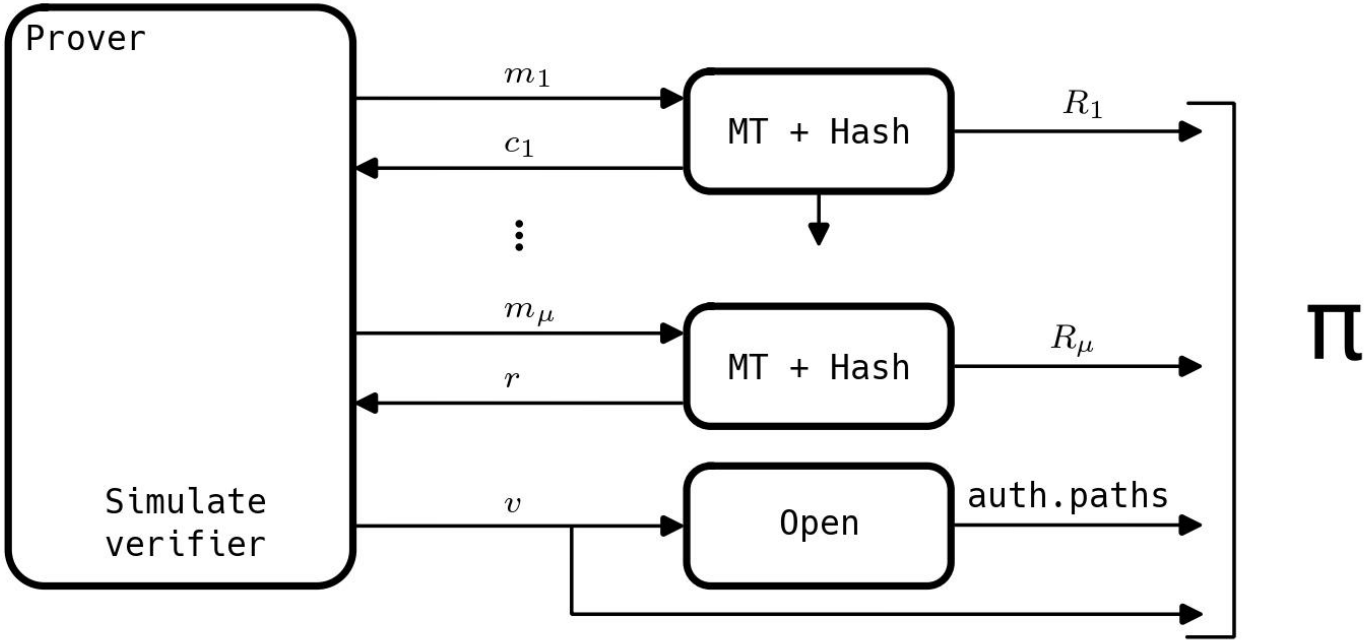
From hIOP to zkSNARK



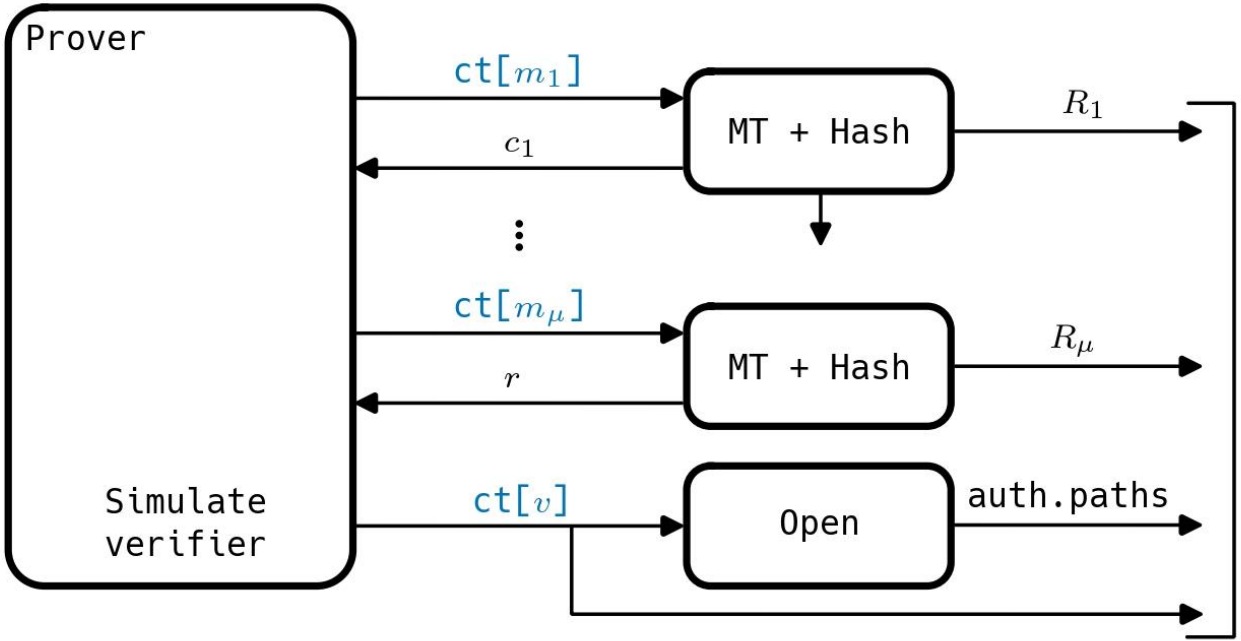
- (holographic) Interactive Oracle Proof
- public-coin
- μ rounds

From hIOP to zkSNARK

BCS compilation [BCS16]

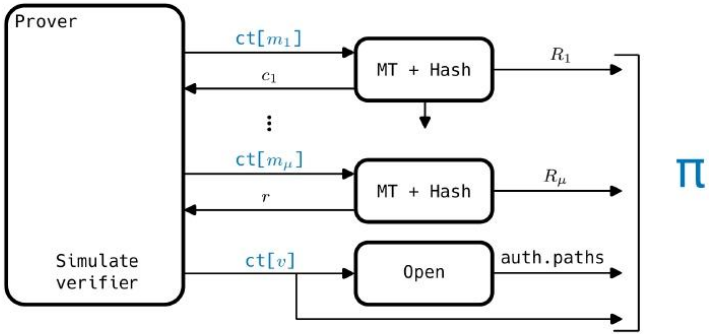


From BhIOP to BzkSNARK



From BhIOP to BzkSNARK

- BhIOP inherits round-by-round (knowledge) soundness
- BCS compilation applies if hIOP is non-adaptive
- Designated-verifier:
  1. Decrypt $\mathtt{ct}[v]$
  2. Check identity $f(v, c, r) \stackrel{?}{=} 0$
  3. Check openings and Fiat-Shamir
- Make proof publicly verifiable:
  1. append plaintexts v to proof
  2. append **Proof of Decryption (PoD)**

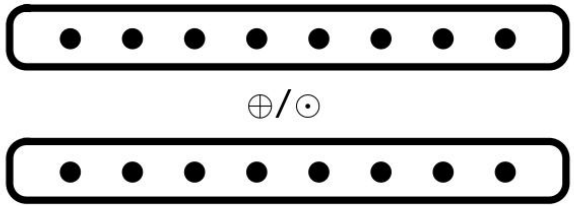


Constructing an efficient Blind zkSNARK

- Select a linear hIOP and high-precision HE scheme
  - as hIOP: Fractal [COS20] + FRI [Ben+18]
- Optimize the hIOP proving algorithm for homomorphic computation in the HE scheme
- Construct an efficient PoD for the selected HE scheme

Computational interface: HE scheme

- Ciphertext space $\mathbb{Z}_Q[X]/\Phi(X)$ homomorphic to plaintext space $\mathcal{P}$: vectors on finite field $\mathbb{F}$
- Operations:
  - elt-wise add `pt` or `ct`
  - elt-wise multiply by `pt` or `ct`
  - permutation in vector



- Noise growth
- Generalized BFV (GBFV) [GV24]
  - supports SIMD slots (cf. BGV/BFV [FV12])
  - supports high-precision arithmetic (cf. CLPX [Che+18])
  - we select $\mathcal{P} = \mathbb{F}_{p^2}^{256}$ for $p = 2^{64} - 2^{32} + 1$

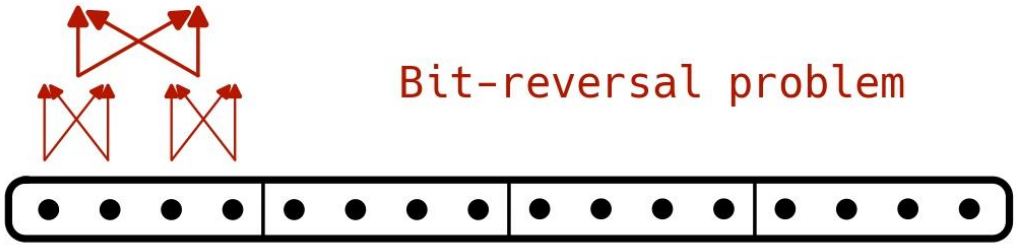
Modifying the hIOP

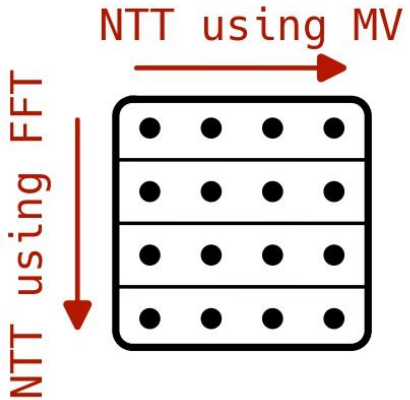
- Generally a trade-off between number of operations and noise depth
  - e.g. domain extensions: compute $f|_L$ from $f|_H$

    Min. number of operations: $f|_L = \mathsf{NTT}(\mathtt{iNTT}(f|_H))$ using FFT

    Min. noise growth: $f|_L = V_L V_H^{-1} f|_H$ using matrix-vector product

    Solution: 2D NTT





Modifying the hIOP

> Example estimate: R1CS with 2^{20} constraints

Computation	Noise (bits)	$C_{\mathtt{add}}$	$C_{\mathtt{ptct}}$	$C_{\mathtt{aut}}$	$C_{\mathtt{ctct}}$
Unpacking	9	0	4096	4096	0
Computing $\mathtt{ct}[Mz]$	14	9421459	9433747	2978354	0
Computing $\mathtt{ct}[\vec{f_z}]$/$\mathtt{ct}[\vec{f_{Mz}}]$	64	4636672	4653056	491520	0
Computing $\mathtt{ct}[\vec{g}]$	138	6762496	6782976	552960	0
Computing $\mathtt{ct}[\vec{f_{\mathsf{FRI}}}]$	0	65536	65536	0	8192
Computing FRI	16	98305	106496	0	0
Ringswitching	9	0	196604	589812	0

Operation count and noise estimates for computing blind Fractal.

Proof of Decryption

- Proves that $\|c_0 + c_1 \cdot \mathsf{sk} - \lfloor \Delta \cdot m \rceil\|_\infty \leq B$ w.r.t. committed sk
- Based on [LNP22] Approximate Range Proofs
  - work over $\mathbb{Z}_{q'}[X]/(X^d + 1)$ with $q' - 1 \equiv 4 \mod 8$ such that

$$X^d + 1 = (X^{d/2} - r)(X^{d/2} + r)$$

  - large set of invertible challenges $c \in \mathcal{C}$ s.t. $\sigma_{-1}(c) = c$
  - use identity $\widetilde{\sigma_{-1}(r)} \cdot s = \langle \vec{r}, \vec{s} \rangle$ to prove norms
  - For $\vec{y} \in [-Q/2, Q/2)^l$ and $B \leq Q/41l$, random $R \leftarrow \mathrm{Bin}_2^{256 \times l}$ and $\vec{v} = R\vec{y} \bmod Q$

$$\|\vec{v}\| < \sqrt{26}B \Rightarrow \|\vec{y}\| < B$$

    except with probability 2^{-256}.

Proof of Decryption

- Proves that $\|c_0 + c_1 \cdot \mathsf{sk} - \lfloor \Delta \cdot m \rceil\|_\infty \leq B$ w.r.t. committed sk
- translate into $\|\vec{v}_{inh}\|_\infty < B_{q'}$ where

$$\vec{v}_{inh} = \mathsf{Rot}_m(c_1) \cdot \vec{\mathsf{sk}} + \vec{c_0} - \overrightarrow{\lfloor \Delta \cdot m \rceil}$$

  - First modswitch from $\mathbb{Z}_q[X]/\Phi_m(X)$ to $\mathbb{Z}_{q'}[X]/\Phi_m(X)$
  - Then represent $\mathbb{Z}_{q'}[X]/\Phi_m(X)$ as $\mathbb{Z}_{q'}^n$ operations
- Approx. norm proof only complete for $\|\vec{v}_{inh}\|_\infty < B_{\mathsf{PoD}}$
  - Relaxation factor $\Phi_r := B_{q'}/B_{\mathsf{PoD}} \approx 31$ bits

Proof of Decryption

- Batching naively scales as $\mathcal{O}(rn^2)$

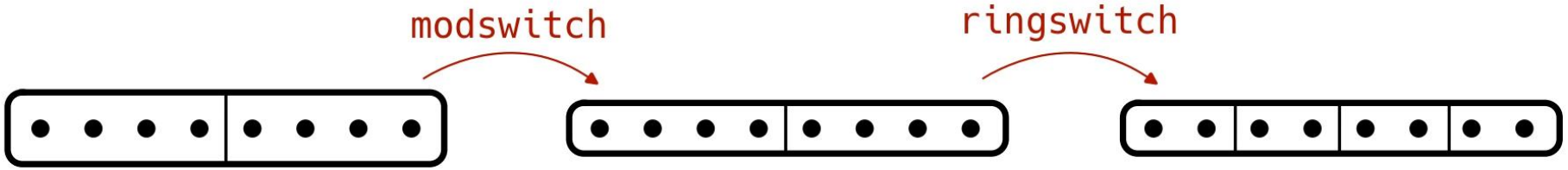
r	$\Pi_{\text{vec-ANP}}$ w/out $\Pi^{(2)}_{eval}$	$\Pi^{(2)}_{eval}$		Total runtime	
		single thread	8 threads	single thread	8 threads
1	0.04	1.15	0.45	1.19	0.49
8	0.14	6.92	1.26	7.06	1.40
64	0.93	53.01	8.09	53.94	9.02
512	7.28	424.17	64.30	431.45	71.58
1024	14.68	846.59	126.89	861.27	141.57
2048	29.40	1688.15	253.55	1717.55	282.95
4096	58.81	3407.10	516.12	3465.91	574.93

Proof of Decryption

- Batching naively scales as $\mathcal{O}(rn^2)$
- Introduce new protocol for batching r PoDs
  - for some random linear combination $f : \vec{m} \mapsto \sum_{i \in [r]} \alpha_i m_i$ prove that $f(\vec{m}) = \mathsf{Dec}\left(\mathsf{Eval}\left(f, \overrightarrow{\mathtt{ct}[m]}\right)\right)$
  - based on homomorphic property and Schwartz-Zippel lemma
- reduces proves cost $\mathcal{O}(rn^2) \to \mathcal{O}(n^2 + rn \log n)$
- at cost of $\approx 6 + \log r$ bits of noise

Proof of Decryption

- Optimized using HE operations
  - Modswitch
    From “FHE-friendly” modulus q to “LNP22-friendly” q'
    i.e. from 398 bits to 97 bits
  - Ringswitch
    From “efficient” $\mathbb{Z}_{q'}/\Phi_{2^{11}\cdot 3\cdot 7}(X)$ to “small” $\mathbb{Z}_{q'}/\Phi_{2^8\cdot 3\cdot 7}(X)$
    i.e. from 96 slots to 24 slots
- MS and RS performed again inside batching protocol



Proof of Decryption

- Implemented in C
- Built upon the LaZer library [LSS24]
- Our parameters: blind zkSNARK for 2^{20} R1CS gates
  - Proof size: 12 kB
  - Prover runtime: 1.7s (1 thread) or 0.65s (8 threads)

[Ara+24] Diego F. Aranha et al. “HELIOPOLIS: Verifiable Computation over Homomorphically Encrypted Data from Interactive Oracle Proofs is Practical”. In: Springer-Verlag, 2024.

[Ata+24] Shahla Atapoor et al. “Verifiable FHE via Lattice-based SNARKs”. In: *IACR Communications in Cryptology (CiC)* 1.1 (2024), p. 24. DOI: 10.62056/a6ksdkp10.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. “Interactive Oracle Proofs”. In: *TCC 2016-B: 14th Theory of Cryptography Conference, Part II*. Ed. by Martin Hirt and Adam D. Smith. Vol. 9986. Lecture Notes in Computer Science. Beijing, China: Springer, Berlin, Heidelberg, Germany, Oct. 2016, pp. 31–60. DOI: 10.1007/978-3-662-53644-5_2.

[Ben+18] Eli Ben-Sasson et al. “Fast Reed-Solomon Interactive Oracle Proofs of Proximity”. In: *ICALP 2018: 45th International Colloquium on Automata, Languages and Programming*. Ed. by Ioannis Chatzigiannakis et al. Vol. 107. LIPIcs. Prague, Czech Republic: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, July 2018, 14:1–14:17. DOI: 10.4230/LIPIcs.ICALP.2018.14.

[Cas+25] Ignacio Cascudo et al. *Verifiable Computation for Approximate Homomorphic Encryption Schemes*. Cryptology ePrint Archive, Paper 2025/286. 2025. URL: https://eprint.iacr.org/2025/286.

[Che+18] Hao Chen et al. “High-Precision Arithmetic in Homomorphic Encryption”. In: *Topics in Cryptology – CT-RSA 2018*. Ed. by Nigel P. Smart. Vol. 10808. Lecture Notes in Computer Science. San Francisco, CA, USA: Springer, Cham, Switzerland, Apr. 2018, pp. 116–136. DOI: 10.1007/978-3-319-76953-0_7.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. “Fractal: Post-quantum and Transparent Recursive Proofs from Holography”. In: *Advances in Cryptology – EUROCRYPT 2020, Part I*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12105. Lecture Notes in Computer Science. Zagreb, Croatia: Springer, Cham, Switzerland, May 2020, pp. 769–793. DOI: 10.1007/978-3-030-45721-1_27.

[FGP14] Dario Fiore, Rosario Gennaro, and Valerio Pastro. “Efficiently Verifiable Computation on Encrypted Data”. In: *ACM CCS 2014: 21st Conference on Computer and Communications Security*. Ed. by Gail-Joon Ahn, Moti Yung, and Ninghui Li. Scottsdale, AZ, USA: ACM Press, Nov. 2014, pp. 844–855. DOI: 10.1145/2660267.2660366.

[FV12] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2012/144. 2012. URL: https://eprint.iacr.org/2012/144.

[GGW24] Sanjam Garg, Aarushi Goel, and Mingyuan Wang. "How to Prove Statements Obliviously?" In: *Advances in Cryptology – CRYPTO 2024, Part X*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14929. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 2024, pp. 449–487. DOI: 10.1007/978-3-031-68403-6_14.

[GV24] Robin Geelen and Frederik Vercauteren. *Fully Homomorphic Encryption for Cyclotomic Prime Moduli*. Cryptology ePrint Archive, Paper 2024/1587. 2024. URL: https://eprint.iacr.org/2024/1587.

[Liu+25] Fengrun Liu et al. *HasteBoots: Proving FHE Bootstrapping in Seconds*. Cryptology ePrint Archive, Paper 2025/261. 2025. URL: https://eprint.iacr.org/2025/261.

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. "Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General". In: *Advances in Cryptology – CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 2022, pp. 71–101. DOI: 10.1007/978-3-031-15979-4_3.

[LSS24] Vadim Lyubashevsky, Gregor Seiler, and Patrick Steuer. *The LaZer Library: Lattice-Based Zero Knowledge and Succinct Proofs for Quantum-Safe Privacy*. Cryptology ePrint Archive, Paper 2024/1846. 2024. URL: https://eprint.iacr.org/2024/1846.

[VKH23] Alexander Viand, Christian Knabenhans, and Anwar Hithnawi. "Verifiable Fully Homomorphic Encryption". In: *CoRR* abs/2301.07041 (2023). DOI: 10.48550/arXiv.2301.07041. arXiv: 2301.07041. URL: https://doi.org/10.48550/arXiv.2301.07041.

[Wal24] Michael Walter. *What Have SNARGs Ever Done for FHE?* Cryptology ePrint Archive, Paper 2024/1207. 2024. URL: https://eprint.iacr.org/2024/1207.

[Zha+25] Xinxuan Zhang et al. *FHE-SNARK vs. SNARK-FHE: From Analysis to Practical Verifiable Computation*. Cryptology ePrint Archive, Paper 2025/302. 2025. URL: https://eprint.iacr.org/2025/302.

Q&A