A Composable View of Verifiable Homomorphic Encryption in Multi-Party Settings

Ganyuan Cao

École Polytechnique Fédérale de Lausanne, Switzerland

March 31, 2025

Motivation

- On-the-Fly MPC [LTV12]

Motivation

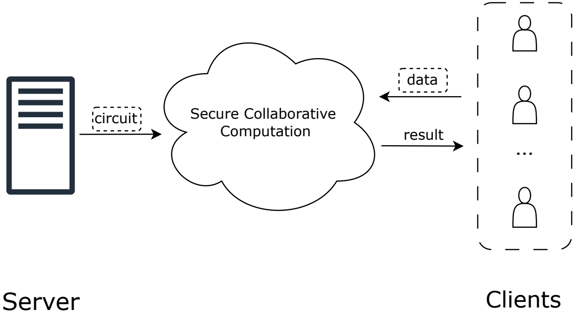
- On-the-Fly MPC [LTV12]
  - Dynamically joining parties.

Motivation

- On-the-Fly MPC [LTV12]
  - Dynamically joining parties.
  - Computation outsourced to untrusted but powerful server.

Motivation

- On-the-Fly MPC [LTV12]
  - Dynamically joining parties.
  - Computation outsourced to untrusted but powerful server.



Motivation

– Homomorphic Encryption (HE) is a good candidate...

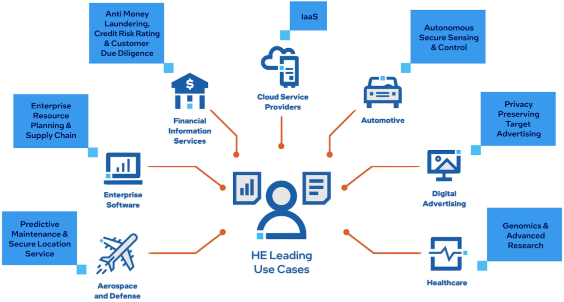
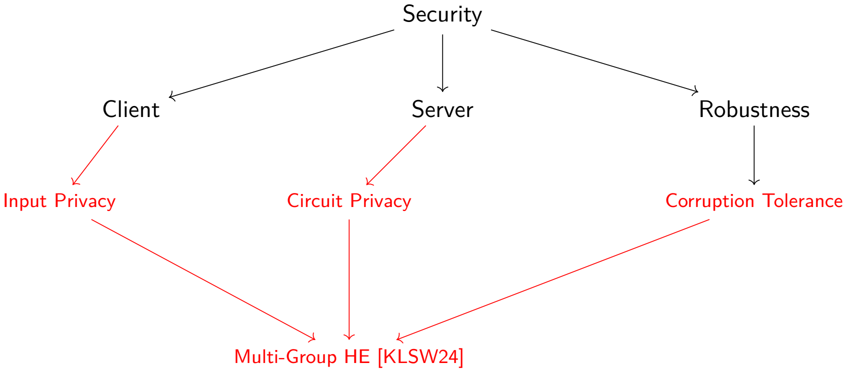
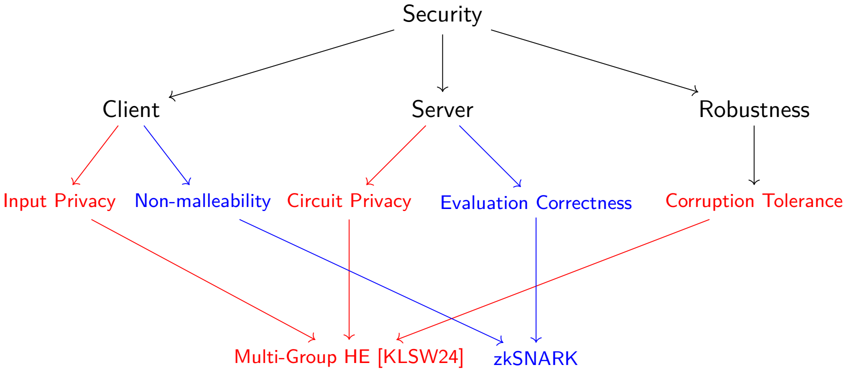


Figure: Use cases of HE [Int].

Goal


Security
Client
Server
Robustness
Input Privacy
Circuit Privacy
Corruption Tolerance
Multi-Group HE [KLSW24]

Goal


Security
Client
Server
Robustness
Input Privacy
Non-malleability
Circuit Privacy
Evaluation Correctness
Corruption Tolerance
Multi-Group HE [KLSW24]
zkSNARK

Contribution

– Formalism

Contribution

- Formalism
  - New game-based notions for (Multi-Key, Threshold, Multi-Group) HE in multi-party setting.

Contribution

- Formalism
  - New game-based notions for (Multi-Key, Threshold, Multi-Group) HE in multi-party setting.
  - UC functionality for HE in multi-party setting.

Contribution

- Formalism
  - New game-based notions for (Multi-Key, Threshold, Multi-Group) HE in multi-party setting.
  - UC functionality for HE in multi-party setting.
- Construction

Contribution

- Formalism
  - New game-based notions for (Multi-Key, Threshold, Multi-Group) HE in multi-party setting.
  - UC functionality for HE in multi-party setting.
- Construction
  - UC-secure MPC via verifiable multi-group HE.

Multi-Group HE (MGHE) [KLSW24]

- Hybrid approach between Threshold HE and Multi-Key HE

Multi-Group HE (MGHE)

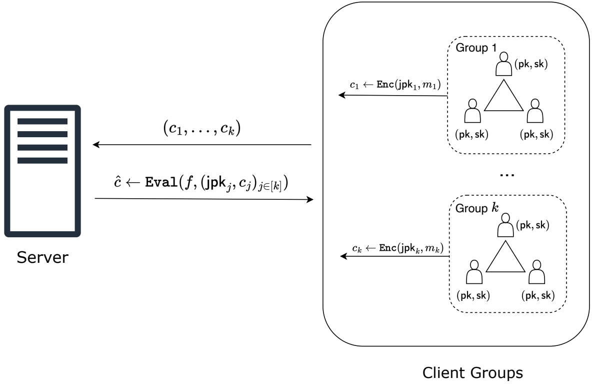
- Hybrid approach between Threshold HE and Multi-Key HE
    - Fewer public keys $\Rightarrow$ better scalability

Multi-Group HE (MGHE) [KLSW24]

- Hybrid approach between Threshold HE and Multi-Key HE
  - Fewer public keys $\Rightarrow$ better scalability
  - Allow for dynamically joining parties $\Rightarrow$ better flexibility

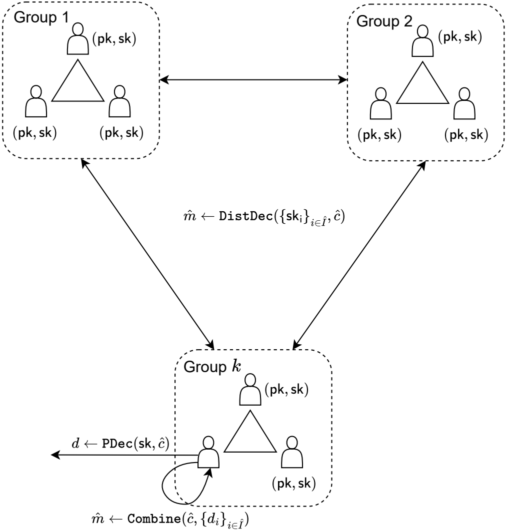
Multi-Group HE (MGHE) [KLSW24]

- Hybrid approach between Threshold HE and Multi-Key HE
  - Fewer public keys $\Rightarrow$ better scalability
  - Allow for dynamically joining parties $\Rightarrow$ better flexibility



Multi-Group HE

EPFL



Confidentiality with Multi-Group HE

$$\mathcal{G}_{\mathsf{KRK}} \xrightarrow{\Pi_{\mathsf{MGHE}}} \mathcal{F}_{\mathsf{MGHE}}$$

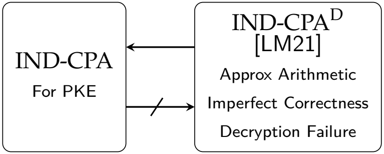
if MGHE satisfies

$$\begin{aligned} & \text{IND-CPA}^{\text{pD}} \\ \wedge\ & \{\text{IND}, \text{SIM}\}\text{-CIRC} \\ \wedge\ & \text{SIM-PDEC} \\ \wedge\ & \text{Decryption Consistency (DC)} \end{aligned}$$

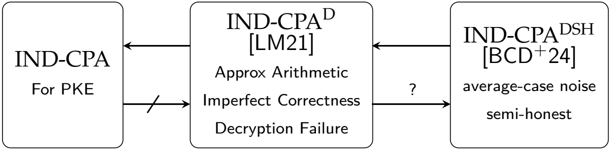
Client Side: IND-CPA$^{\mathrm{pD}}$ Security

IND-CPA
For PKE

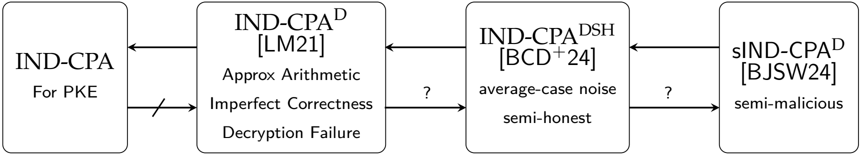
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



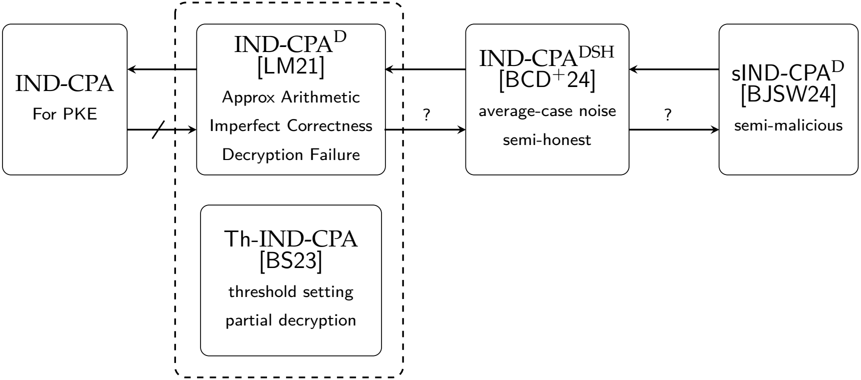
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



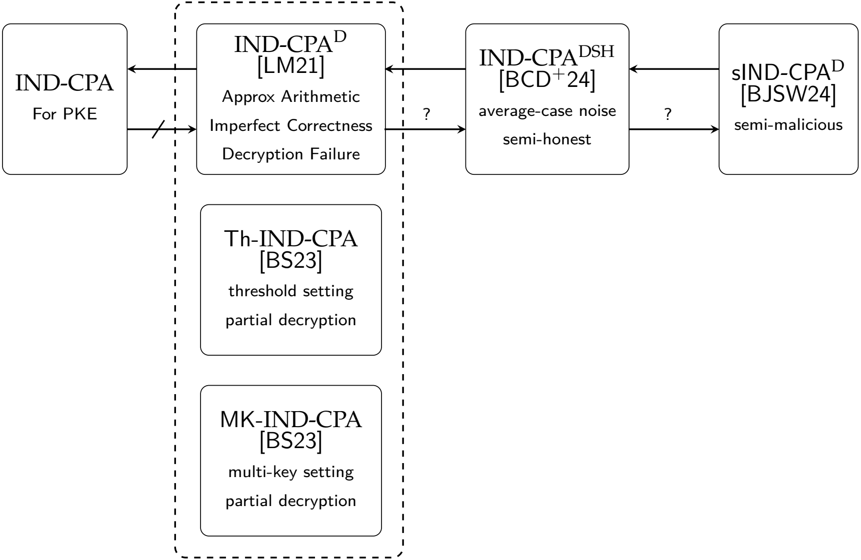
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



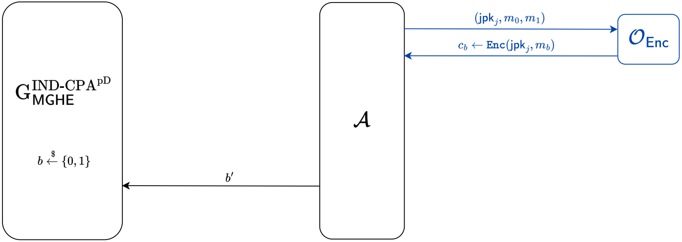
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



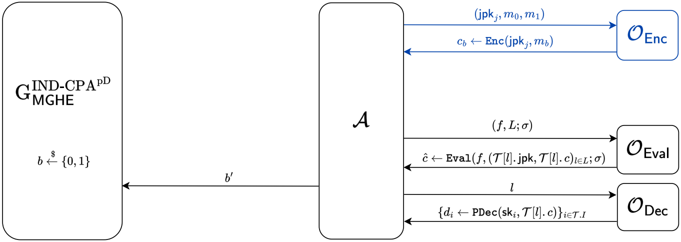
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



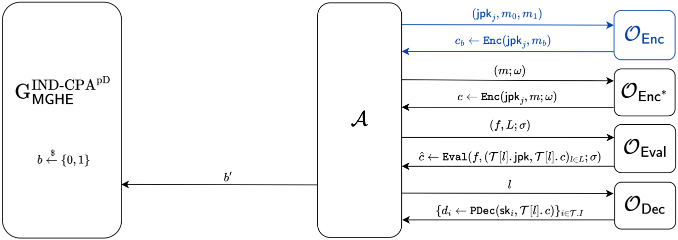
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



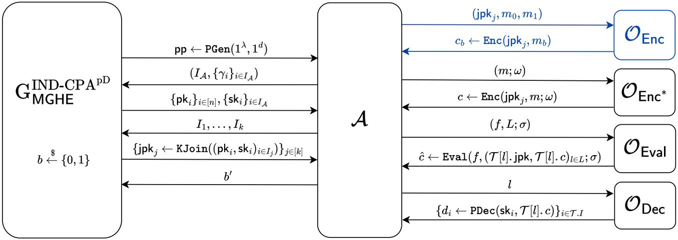
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



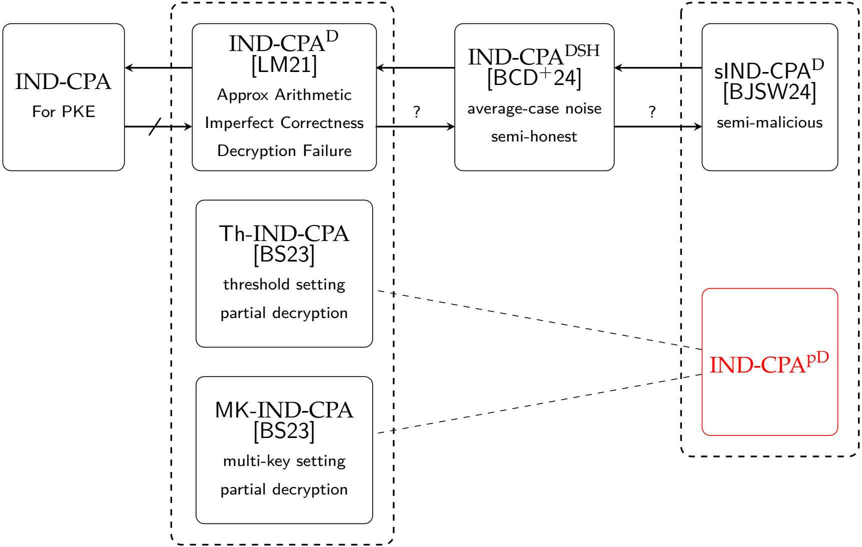
Client Side: IND-CPA$^{\mathrm{pD}}$ Security



Client Side: IND-CPA$^{\mathrm{pD}}$ Security



Client Side: IND-CPA$^{\mathrm{pD}}$ Security



Server Side: Circuit Privacy

- Usually formalized using *Simulation* [IP07, Gen09, BdPMW16].

Server Side: Circuit Privacy

– Usually formalized using *Simulation* [IP07, Gen09, BdPMW16].

$$\texttt{Sim}_{circ}((\mathsf{jpk}_1, \ldots, \mathsf{jpk}_\ell), f(m_1, \ldots, m_\ell))$$
$$\overset{s}{\approx}$$
$$\hat{c} \leftarrow \mathsf{MGHE}.\texttt{Eval}(f, (\mathsf{jpk}_j, c_j)_{j \in [\ell]})$$

Server Side: Circuit Privacy

– Usually formalized using *Simulation* [IP07, Gen09, BdPMW16].

$$\mathtt{Sim}_{circ}((\mathsf{jpk}_1, \ldots, \mathsf{jpk}_\ell), f(m_1, \ldots, m_\ell))$$
$$\overset{s}{\approx}$$
$$\hat{c} \leftarrow \mathsf{MGHE}.\mathtt{Eval}(f, (\mathsf{jpk}_j, c_j)_{j \in [\ell]})$$

– Stronger security with *statistical indistinguishability*.

Server Side: Circuit Privacy

– Not suitable for schemes with approximate evaluation like [CKKS17].

Server Side: Circuit Privacy

– Not suitable for schemes with approximate evaluation like [CKKS17].

$$\hat{m} = f(m_1, \ldots, m_\ell)$$

$$\hat{m} + \varepsilon \leftarrow \mathsf{MGHE}.\mathtt{Dec}(\hat{c})$$

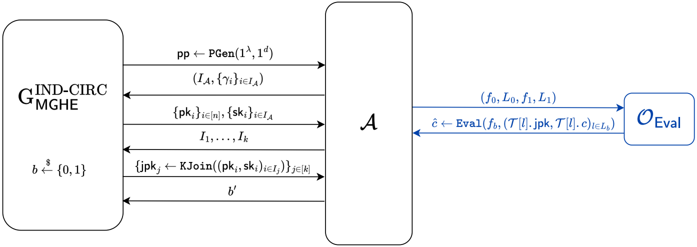
Server Side: Circuit Privacy

- Variant of IND-CIRC security [KS23] in multi-group setting.

Server Side: Circuit Privacy

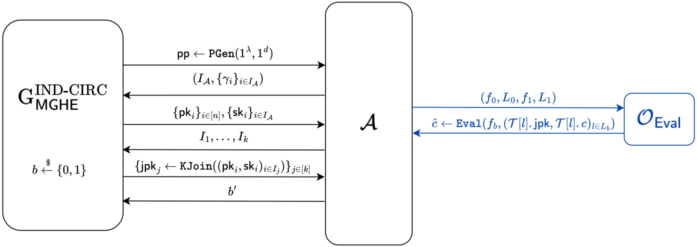
EPFL

- Variant of IND-CIRC security [KS23] in multi-group setting.



Server Side: Circuit Privacy

- Variant of IND-CIRC security [KS23] in multi-group setting.

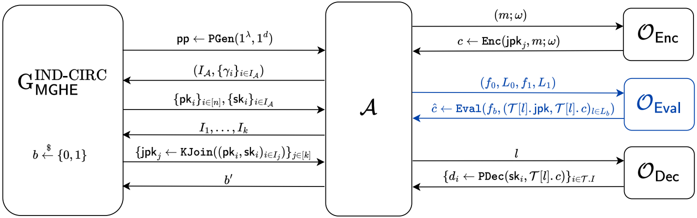


- Challenge with (f_0, L_0, f_1, L_1) instead s.t.

$$f_0(\{m_j\}_{j \in L_0}) = f_1(\{m_j\}_{j \in L_1})$$

Server Side: Circuit Privacy

– Variant of IND-CIRC security [KS23] in multi-group setting.



– Challenge with (f_0, L_0, f_1, L_1) instead s.t.

$$f_0(\{m_j\}_{j\in L_0}) = f_1(\{m_j\}_{j\in L_1})$$

Threshold Security: SIM-PDEC Security

- Simulatability of partial decryption

Threshold Security: SIM-PDEC Security

- Simulatability of partial decryption

$$\mathtt{Sim}_{th}(c, m, \{\mathsf{sk}_i\}_{i \in I_{\mathcal{A}}}) \overset{s}{\approx} d \leftarrow \mathsf{MGHE}.\mathtt{PDec}(\mathsf{sk}_j, c), j \notin I_{\mathcal{A}}$$

Threshold Security: SIM-PDEC Security

- Simulatability of partial decryption

$$\mathtt{Sim}_{th}(c, m, \{\mathsf{sk}_i\}_{i \in I_{\mathcal{A}}}) \overset{s}{\approx} d \leftarrow \mathsf{MGHE}.\mathtt{PDec}(\mathsf{sk}_j, c), j \notin I_{\mathcal{A}}$$

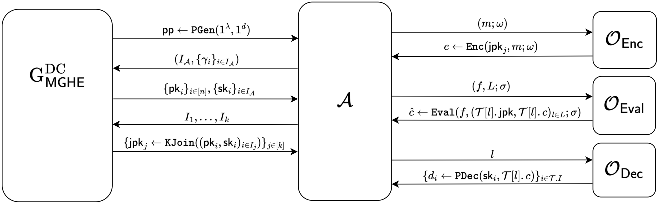
- Security of secret key sk_j of honest client i.e., $j \notin I_{\mathcal{A}}$

Threshold Security: Decryption Consistency

- In a (t, n)-threshold structure, message is reconstructed correctly as long as sufficient partial decryptions have been obtained.

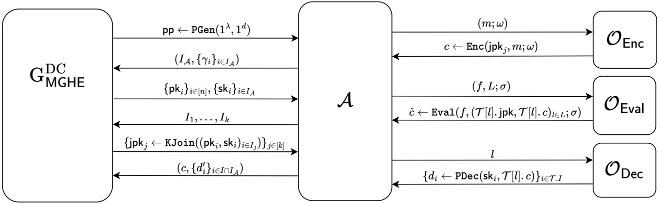
Threshold Security: Decryption Consistency

– In a (t, n)-threshold structure, message is reconstructed correctly as long as sufficient partial decryptions have been obtained.



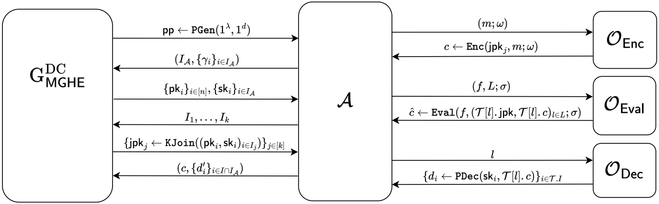
Threshold Security: Decryption Consistency

- In a (t, n)-threshold structure, message is reconstructed correctly as long as sufficient partial decryptions have been obtained.



Threshold Security: Decryption Consistency

– In a (t, n)-threshold structure, message is reconstructed correctly as long as sufficient partial decryptions have been obtained.

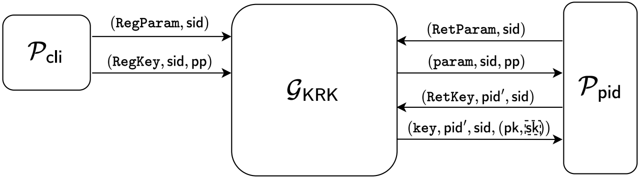


– $\mathcal{A}$ wins if

$$m \neq \mathtt{Combine}(c, \{d_i\}_{i \in I \setminus I_\mathcal{A}} \cup \{d'_i\}_{i \in I \cap I_\mathcal{A}})$$

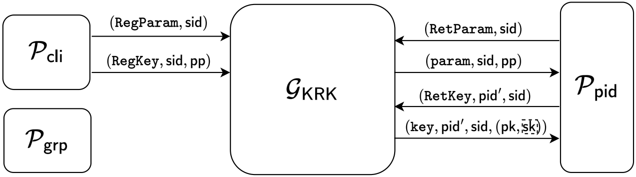
UC: Global Key Registry $\mathcal{G}_{\mathsf{KRK}}$

– Global subroutine for key management taken from [BCNP04].



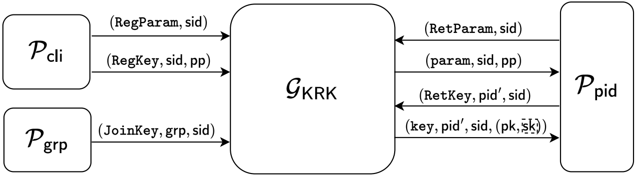
UC: Global Key Registry $\mathcal{G}_{\mathsf{KRK}}$

- Global subroutine for key management taken from [BCNP04].
- "Virtual entity" $\mathcal{P}_{\mathsf{grp}}$ for a group $\mathsf{grp} = \{\mathsf{cli}_1, \mathsf{cli}_2, \ldots, \mathsf{cli}_n\}$.

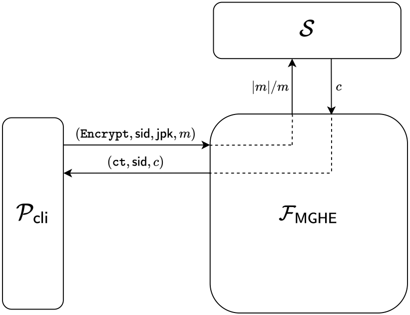


UC: Global Key Registry $\mathcal{G}_{\mathsf{KRK}}$

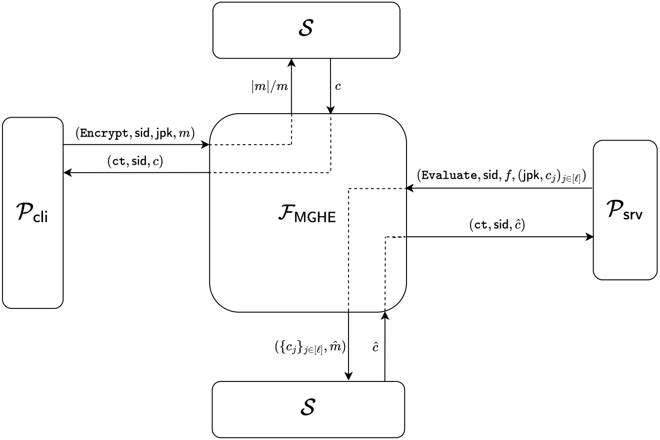
- Global subroutine for key management taken from [BCNP04].
- "Virtual entity" $\mathcal{P}_{\mathsf{grp}}$ for a group $\mathsf{grp} = \{\mathsf{cli}_1, \mathsf{cli}_2, \ldots, \mathsf{cli}_n\}$.
- Key aggregation for groups (equivalent to $\mathcal{F}_{\mathsf{MPC}}$).



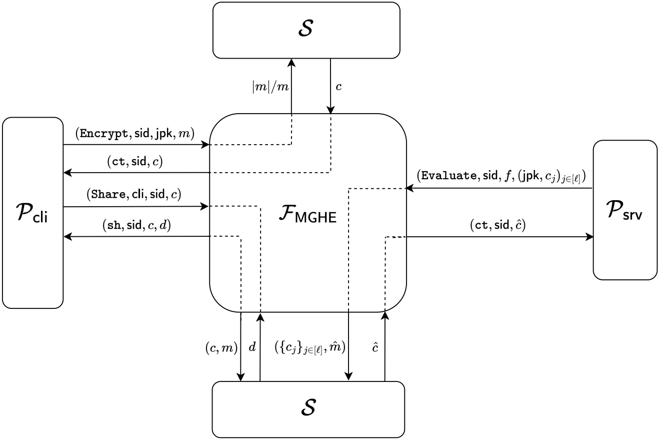
UC: Ideal Functionality $\mathcal{F}_{\mathsf{MGHE}}$



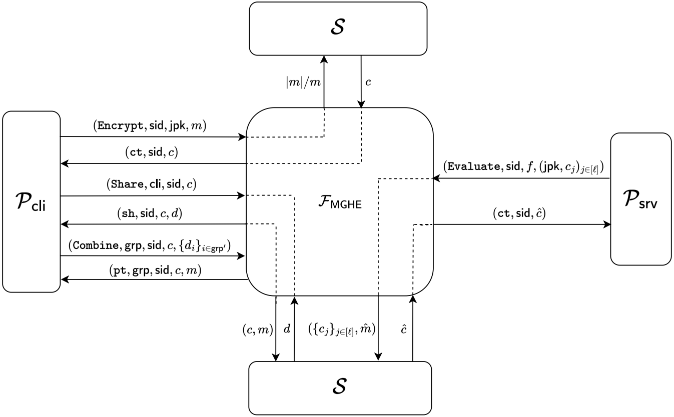
UC: Ideal Functionality $\mathcal{F}_{\mathsf{MGHE}}$



UC: Ideal Functionality $\mathcal{F}_{\mathsf{MGHE}}$



UC: Ideal Functionality $\mathcal{F}_{\mathsf{MGHE}}$



Realization of $\mathcal{F}_{\mathsf{MGHE}}$

EPFL

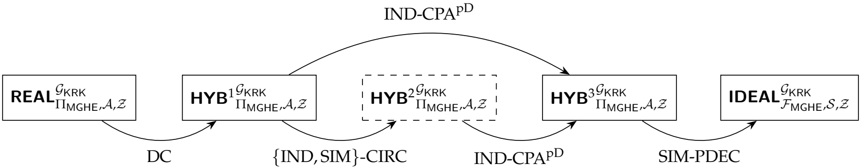
Theorem 1

Π_{MGHE} UC-realizes $\mathcal{F}_{\mathsf{MGHE}}$ against a semi-malicious adversary in presence of $\mathcal{G}_{\mathsf{KRK}}$ if MGHE *is* $\mathrm{IND\text{-}CPA}^{\mathrm{pD}}$, $\mathrm{IND\text{-}CIRC}$ ($\mathrm{SIM\text{-}CIRC}$)*, and* $\mathrm{SIM\text{-}PDEC}$ *secure under the static corruption of clients in a group up to the threshold and possibly the server.*

Realization of $\mathcal{F}_{\mathsf{MGHE}}$

$$
\begin{tikzcd}
\boxed{\mathbf{REAL}^{\mathcal{G}_{\mathsf{KRK}}}_{\Pi_{\mathsf{MGHE}},\mathcal{A},\mathcal{Z}}} \arrow[r, bend right, "\text{DC}"'] & \boxed{\mathbf{HYB}^{1\,\mathcal{G}_{\mathsf{KRK}}}_{\Pi_{\mathsf{MGHE}},\mathcal{A},\mathcal{Z}}} \arrow[rr, bend left, "\text{IND-CPA}^{\text{pD}}"] & & \boxed{\mathbf{HYB}^{3\,\mathcal{G}_{\mathsf{KRK}}}_{\Pi_{\mathsf{MGHE}},\mathcal{A},\mathcal{Z}}} \arrow[r, bend right, "\text{SIM-PDEC}"'] & \boxed{\mathbf{IDEAL}^{\mathcal{G}_{\mathsf{KRK}}}_{\mathcal{F}_{\mathsf{MGHE}},\mathcal{S},\mathcal{Z}}}
\end{tikzcd}
$$

Realization of $\mathcal{F}_{\mathsf{MGHE}}$



Integrity via Verifiability

MGHE $\Rightarrow$ Security against *semi-malicious* adversary

Integrity via Verifiability

MGHE $\Rightarrow$ Security against *semi-malicious* adversary

MGHE + zkSNARK $\Rightarrow$ Security against *(full) malicious* adversary

zkSNARK in ROM

- UC-secure zkSNARK

zkSNARK in ROM

– UC-secure zkSNARK

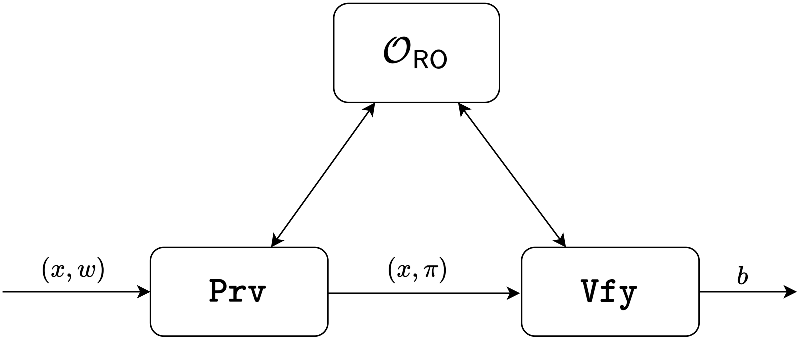
zkSNARK		[CF24]	[BFKT24]	[GKO^{+}23]
NIZK	[BS21]	[LR22]		[LR22]
	CRS	ROM	ROM-AGM	CRS-ROM

zkSNARK in ROM

– UC-secure zkSNARK

zkSNARK		[CF24]	[BFKT24]	[GKO+23]
NIZK	[BS21]	[LR22]		[LR22]
	CRS	ROM	ROM-AGM	CRS-ROM

zkSNARK in ROM



Properties of zkSNARK

- *Completeness*: Valid arguments must be accepted.

$$\forall (x,w) \in R,\ \Pr\left[\mathtt{Vfy}^{\mathcal{O}_{\mathsf{RO}}}(x,\pi) = 1 \;\middle|\; \begin{array}{l} \mathcal{O}_{\mathsf{RO}} \leftarrow \mathcal{U}(\lambda) \\ \pi \leftarrow \mathtt{Prv}^{\mathcal{O}_{\mathsf{RO}}}(x,w) \end{array}\right] = 1.$$

Properties of zkSNARK

- *Zero-Knowledge*: Arguments do not disclose information about witness.

$$\left\{ \mathsf{out} \;\middle|\; \begin{array}{l} \mathcal{O}_{\mathsf{RO}} \leftarrow \mathcal{U}(\lambda) \\ (x, w, \mathsf{aux}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{RO}}} \\ \pi \leftarrow \mathtt{Prv}^{\mathcal{O}_{\mathsf{RO}}}(x, w) \\ \mathsf{out} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{RO}}}(\mathsf{aux}, \pi) \end{array} \right\} \approx \left\{ \mathsf{out} \;\middle|\; \begin{array}{l} \mathcal{O}_{\mathsf{RO}} \leftarrow \mathcal{U}(\lambda) \\ (x, w, \mathsf{aux}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{RO}}} \\ (\pi, \mathsf{pg}) \leftarrow \mathtt{Sim}^{\mathcal{O}_{\mathsf{RO}}}(x) \\ \mathsf{out} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{RO}}[\mathsf{pg}]}(\mathsf{aux}, \pi) \end{array} \right\}$$

Properties of zkSNARK

- *Simulation Soundness*: Non-malleability of arguments.

$$\Pr\left[\begin{array}{l}|x| \leq n \\ \wedge\, x \notin \mathcal{L}(R) \\ \wedge\, \texttt{Vfy}^{\mathcal{O}_{\mathsf{RO}}}(x, \pi) = 1\end{array} \;\middle|\; \begin{array}{l}\mathcal{O}_{\mathsf{RO}} \leftarrow \mathcal{U}(\lambda) \\ (x, \pi) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{RO}}}(\texttt{Sim})\end{array}\right] \leq \text{negl.}$$

Properties of zkSNARK

- *Succinctness*: Argument is efficient.

$$|\pi| \ll |w|$$

On-the-Fly MPC

- Three-phase protocol

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading
  - Circuit Evaluation

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading
  - Circuit Evaluation
  - Result Retrieval

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading
  - Circuit Evaluation
  - Result Retrieval
- Security

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading
  - Circuit Evaluation
  - Result Retrieval
- Security
  - against *malicious* adversary.

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading
  - Circuit Evaluation
  - Result Retrieval
- Security
  - against *malicious* adversary.
  - under *non-adaptive* corruption of

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading
  - Circuit Evaluation
  - Result Retrieval
- Security
  - against *malicious* adversary.
  - under *non-adaptive* corruption of
    - the server

On-the-Fly MPC

- Three-phase protocol
  - Data Uploading
  - Circuit Evaluation
  - Result Retrieval
- Security
  - against *malicious* adversary.
  - under *non-adaptive* corruption of
    - the server
    - the clients in a group up to the threshold

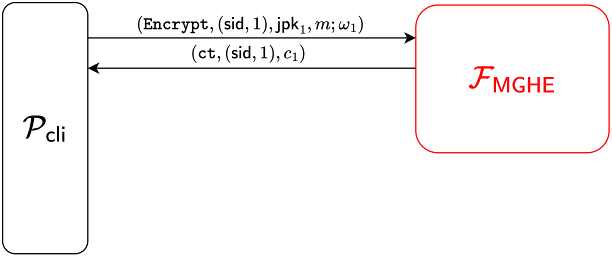
Phase 1: Data Uploading

- Naor-Yung Double Encryption Paradigm.

Phase 1: Data Uploading

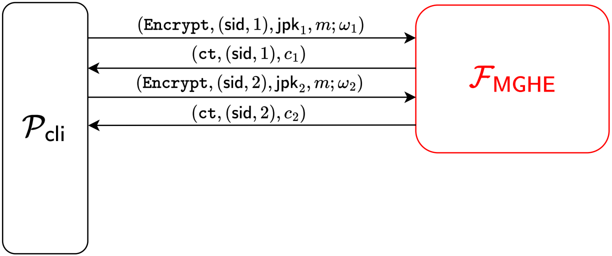
EPFL

- Naor-Yung Double Encryption Paradigm.



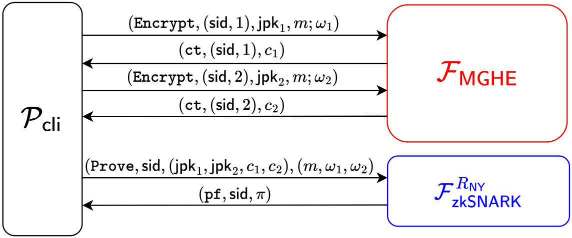
Phase 1: Data Uploading

- Naor-Yung Double Encryption Paradigm.

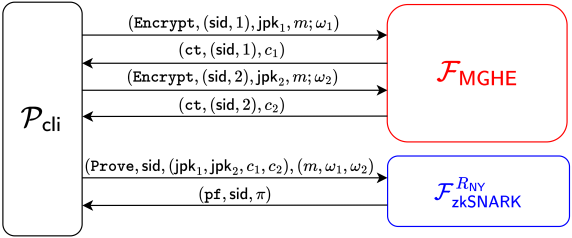


Phase 1: Data Uploading

- Naor-Yung Double Encryption Paradigm.



– Naor-Yung Double Encryption Paradigm.



$$R_{\mathsf{NY}} = \left\{ \left(\begin{pmatrix} \mathsf{jpk}_1, c_1, \\ \mathsf{jpk}_2, c_2 \end{pmatrix}, (m, \omega_1, \omega_2) \right) \middle| \begin{array}{l} c_1 = \mathsf{MGHE}.\texttt{Enc}(\mathsf{jpk}_1, m; \omega_1) \\ \wedge \\ c_2 = \mathsf{MGHE}.\texttt{Enc}(\mathsf{jpk}_2, m; \omega_2) \end{array} \right\}$$

Phase 1: Data Uploading

– CCA1-secure HE as in [LMSV12, BSW12, CRRV17].

Phase 1: Data Uploading

- CCA1-secure HE as in [LMSV12, BSW12, CRRV17].
  - Tampering with c_1 or c_1 $\Rightarrow$ Verification fails.

Phase 1: Data Uploading

- CCA1-secure HE as in [LMSV12, BSW12, CRRV17].
  - Tampering c_1 or c_1 $\Rightarrow$ Verification fails.
  - Must know m to generate valid ciphertext tuple.

Phase 1: Data Uploading

Naor-Yung + Simulation Soundness

Phase 1: Data Uploading

Naor-Yung + Simulation Soundness

or

One-Pass + Simulation Extractability

Phase 1: Data Uploading

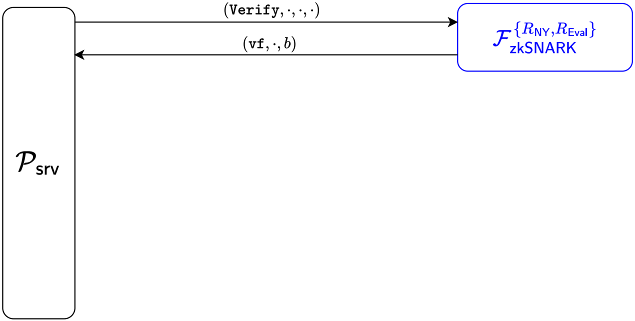
Naor-Yung + Simulation Soundness

or

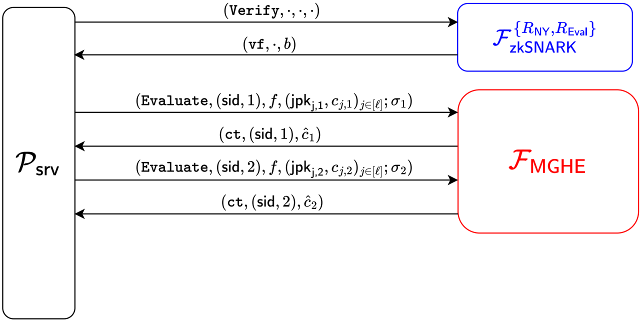
One-Pass + Simulation Extractability

$$R_{\mathsf{NY}} = \{(\mathsf{jpk}, c), (m, \omega)) \quad c = \mathsf{MGHE}.\mathtt{Enc}(\mathsf{jpk}, m; \omega)\}$$

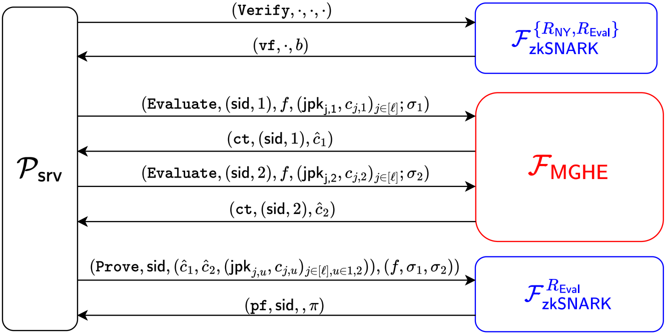
Phase 2: Circuit Evaluation



Phase 2: Circuit Evaluation



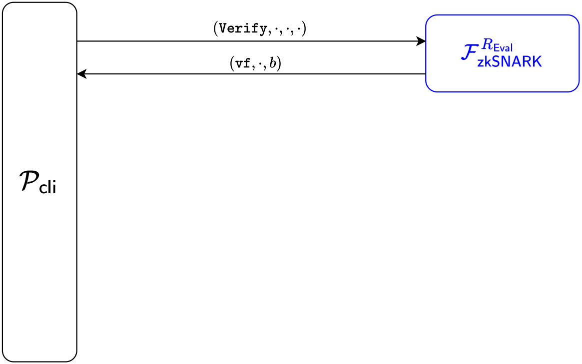
Phase 2: Circuit Evaluation



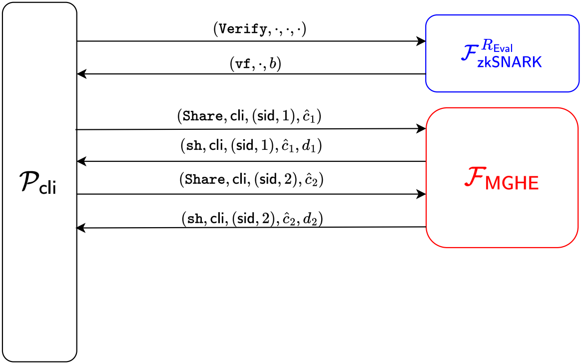
Phase 2: Circuit Evaluation

$$R_{\mathsf{Eval}} = \left\{ \begin{array}{c|c} \left(\begin{pmatrix} \hat{c}_1, (\mathsf{jpk}_{j,1}, c_{j,1})_{j \in [\ell]}, \\ \hat{c}_2, (\mathsf{jpk}_{j,2}, c_{j,2})_{j \in [\ell]} \end{pmatrix}, (f, \sigma_1, \sigma_2) \right) & \\ & \begin{array}{l} \hat{c}_1 = \mathsf{MGHE}.\mathtt{Eval}(f, (\mathsf{jpk}_{j,1}, c_{j,1})_{j \in [\ell]}; \sigma_1) \\ \wedge \\ \hat{c}_2 = \mathsf{MGHE}.\mathtt{Eval}(f, (\mathsf{jpk}_{j,2}, c_{j,2})_{j \in [\ell]}; \sigma_2) \end{array} \end{array} \right\}.$$

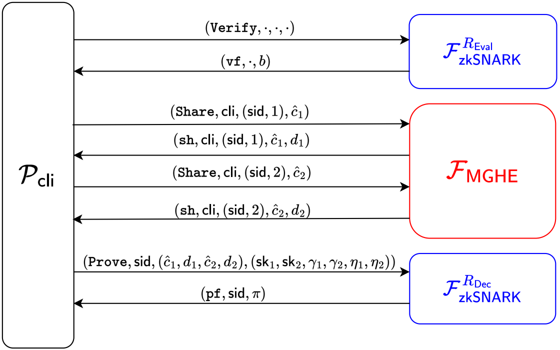
Phase 3: Result Retrieval - Partial Decryption



Phase 3: Result Retrieval - Partial Decryption

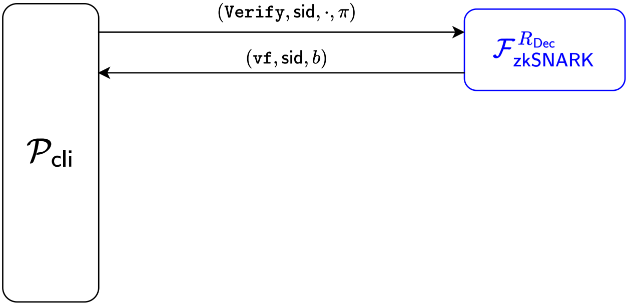


Phase 3: Result Retrieval - Partial Decryption

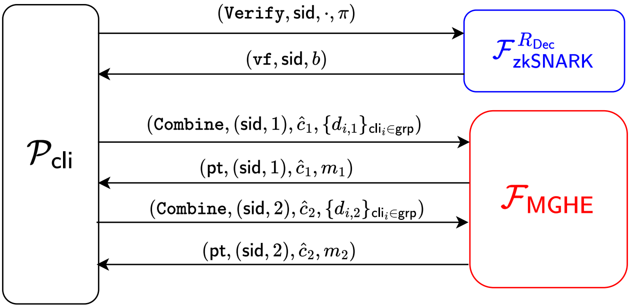


$$R_{\mathsf{Dec}} = \left\{ \begin{pmatrix} \mathsf{pp}_1, \mathsf{pk}_1, c_1, d_1, \\ \mathsf{pp}_2, \mathsf{pk}_2, c_2, d_2 \end{pmatrix}, \begin{pmatrix} \mathsf{sk}_1, \gamma_1, \eta_1 \\ \mathsf{sk}_2, \gamma_2, \eta_2 \end{pmatrix} \middle| \begin{array}{l} \forall u \in \{1, 2\} : \\ \quad d_u = \mathsf{MGHE.PDec}(\mathsf{sk}_u, c_u; \eta_u) \\ \quad \wedge\, \mathsf{pk}_u = \mathsf{PKGen}(\mathsf{pp}_u, \mathsf{sk}_u; \gamma_u) \end{array} \right\}.$$

Phase 3: Result Retrieval - Reconstruction

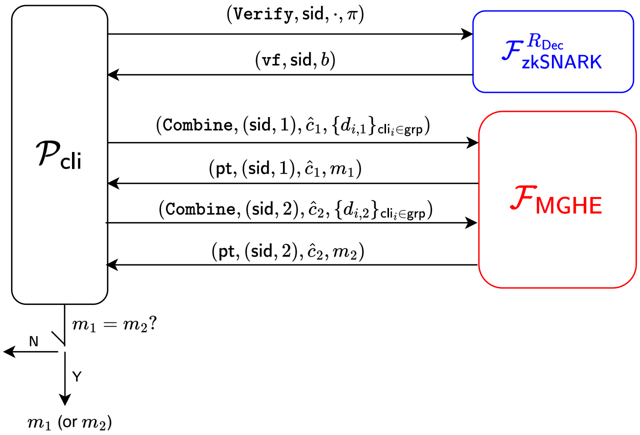


Phase 3: Result Retrieval - Reconstruction



Phase 3: Result Retrieval - Reconstruction

EPFL



Realization of On-the-Fly MPC

Theorem 2

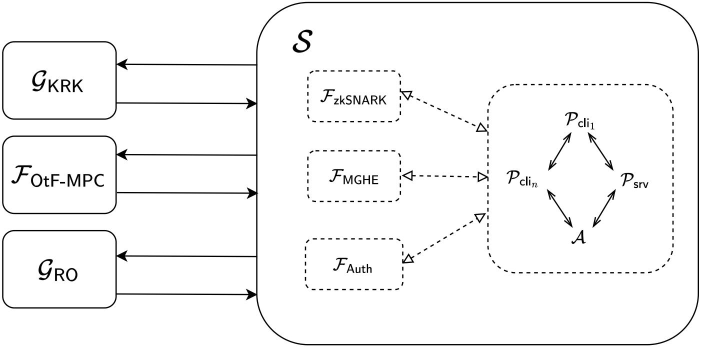
$\Pi_{\mathsf{OtF\text{-}MPC}}$ *UC-realizes* $\mathcal{F}_{\mathsf{OtF\text{-}MPC}}$ *in* $[\mathcal{F}_{\mathsf{MGHE}}, \mathcal{F}_{\mathsf{zkSNARK}}, \mathcal{F}_{\mathsf{Aut}}]$*-hybrid model in presence of* $\mathcal{G}_{\mathsf{KRK}}$ *and* $\mathcal{G}_{\mathsf{RO}}$.

Realization of On-the-Fly MPC

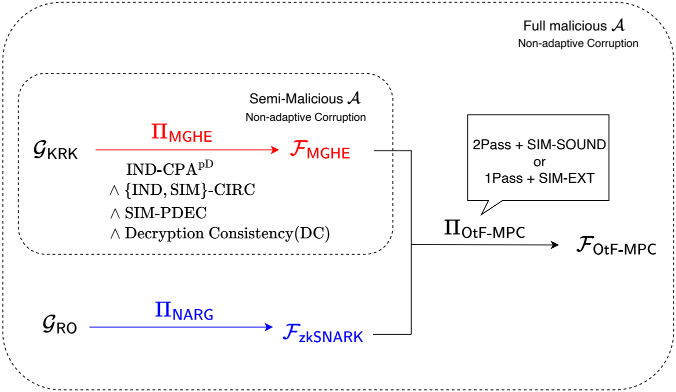
EPFL

Theorem 3

$\Pi_{\mathsf{OtF\text{-}MPC}}$ UC-realizes $\mathcal{F}_{\mathsf{OtF\text{-}MPC}}$ in $[\mathcal{F}_{\mathsf{MGHE}}, \mathcal{F}_{\mathsf{zkSNARK}}, \mathcal{F}_{\mathsf{Aut}}]$-hybrid model in presence of $\mathcal{G}_{\mathsf{KRK}}$ and $\mathcal{G}_{\mathsf{RO}}$.

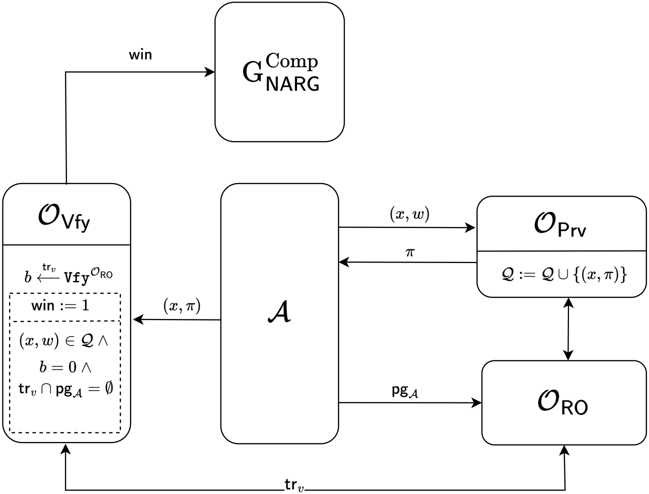


Conclusion


Full malicious $\mathcal{A}$
Non-adaptive Corruption
Semi-Malicious $\mathcal{A}$
Non-adaptive Corruption
$\mathcal{G}_{\mathsf{KRK}}$
Π_{MGHE}
$\mathcal{F}_{\mathsf{MGHE}}$
$\text{IND-CPA}^{\text{pD}}$
$\wedge$ {IND, SIM}-CIRC
$\wedge$ SIM-PDEC
$\wedge$ Decryption Consistency(DC)
2Pass + SIM-SOUND
or
1Pass + SIM-EXT
$\Pi_{\mathsf{OtF\text{-}MPC}}$
$\mathcal{F}_{\mathsf{OtF\text{-}MPC}}$
$\mathcal{G}_{\mathsf{RO}}$
Π_{NARG}
$\mathcal{F}_{\mathsf{zkSNARK}}$

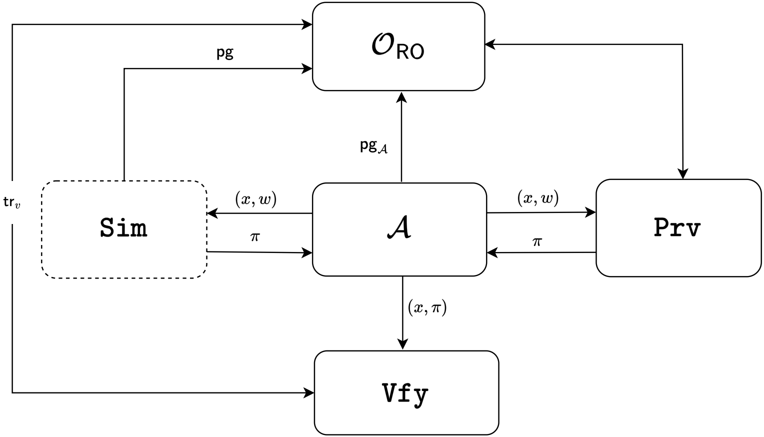
Appendix: Completeness

– Valid arguments must be accepted.



Appendix: Zero-Knowledge

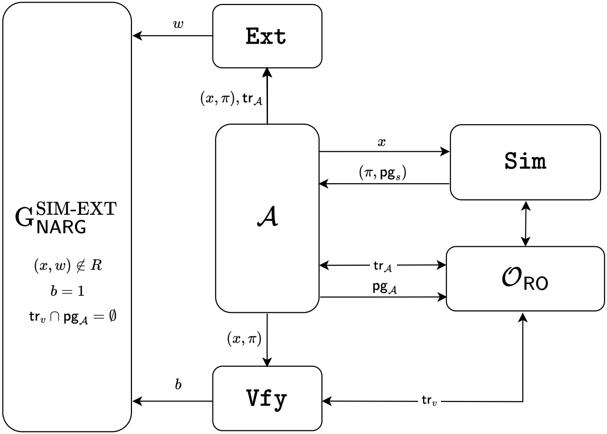
- Arguments does not disclose information about witness.



Appendix: sSIM-EXT

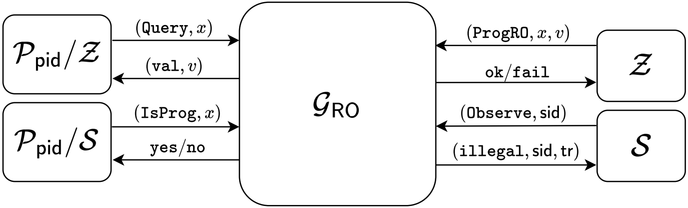
EPFL

- Stronger version for *Knowledge Soundness*.
- *Non-malleability* for UC-security.

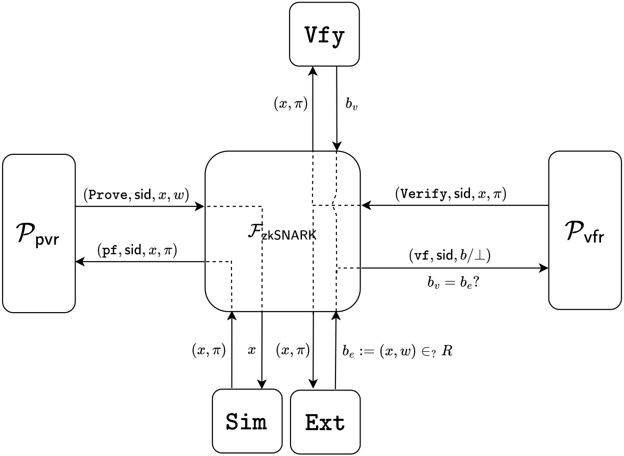


Appendix: Global Random Oracle $\mathcal{G}_{\mathsf{RO}}$

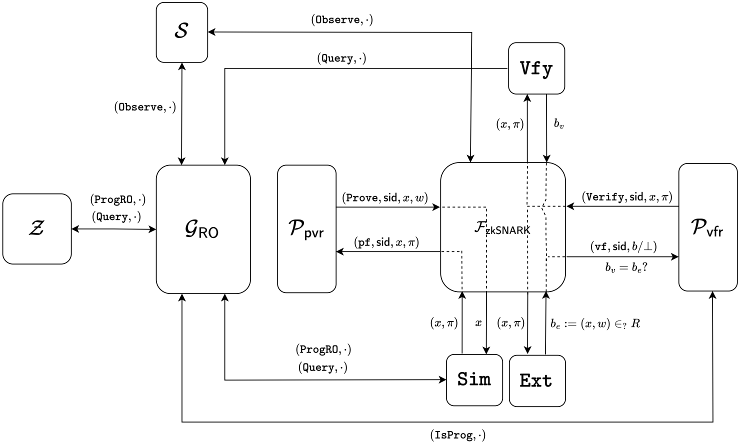
- Global random oracle with restricted *programming* and *observability* [CDG+18].



Appendix: Functionality $\mathcal{F}_{\mathsf{zkSNARK}}$ [CF24]



Appendix: $\mathcal{F}_{\mathsf{zkSNARK}} + \mathcal{G}_{\mathsf{RO}}$



Appendix: Realization of $\mathcal{F}_{\mathsf{zkSNARK}}$

Theorem 4[?, TCC:ChiFen24

]Π_{NARG} securely realizes $\mathcal{F}_{\mathsf{zkSNARK}}$ in $\mathcal{G}_{\mathsf{RO}}$-hybrid model if NARG *satisfies completeness,* $s\mathrm{SIM\text{-}EXT}$*, and zero-knowledge.*

$$
\begin{tikzcd}
\boxed{\mathbf{REAL}^{\mathcal{G}}_{\Pi,\mathcal{A},\mathcal{Z}}} \arrow[r, bend right, "\text{Completeness}"'] & \boxed{\mathbf{HYB}^{1\,\mathcal{G}}_{\Pi,\mathcal{A},\mathcal{Z}}} \arrow[r, bend right, "\text{ZK}"'] & \boxed{\mathbf{HYB}^{2\,\mathcal{G}}_{\Pi,\mathcal{A},\mathcal{Z}}} \arrow[r, bend right, "s\mathrm{SIM\text{-}EXT}"'] & \boxed{\mathbf{IDEAL}^{\mathcal{G}}_{\mathcal{F},\mathcal{S},\mathcal{Z}}}
\end{tikzcd}
$$

Flavio Bergamaschi, Anamaria Costache, Dana Dachman-Soled, Hunter Kippen, Lucas LaBuff, and Rui Tang.
Revisiting the security of approximate FHE with noise-flooding countermeasures.
Cryptology ePrint Archive, Report 2024/424, 2024.
URL: https://eprint.iacr.org/2024/424.

Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass.
Universally composable protocols with relaxed set-up assumptions.
pages 186–195, 2004.
doi:10.1109/FOCS.2004.71.

Florian Bourse, Rafaël del Pino, Michele Minelli, and Hoeteck Wee.
FHE circuit privacy almost for free.
pages 62–89, 2016.
doi:10.1007/978-3-662-53008-5_3.

Jan Bobolz, Pooya Farshim, Markulf Kohlweiss, and Akira Takahashi.

The brave new world of global generic groups and UC-secure zero-overhead SNARKs.
pages 90–124, 2024.
doi:10.1007/978-3-031-78011-0_4.

Olivier Bernard, Marc Joye, Nigel P. Smart, and Michael Walter.
Drifting towards better error probabilities in fully homomorphic encryption schemes.
Cryptology ePrint Archive, Paper 2024/1718, 2024.
URL: https://eprint.iacr.org/2024/1718.

Karim Baghery and Mahdi Sedaghat.
Tiramisu: Black-box simulation extractable NIZKs in the updatable CRS model.
pages 531–551, 2021.
doi:10.1007/978-3-030-92548-2_28.

Katharina Boudgoust and Peter Scholl.
Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus.

pages 371–404, 2023.
doi:10.1007/978-981-99-8721-4_12.

Dan Boneh, Gil Segev, and Brent Waters.
Targeted malleability: homomorphic encryption for restricted computations.
pages 350–366, 2012.
doi:10.1145/2090236.2090264.

Jan Camenisch, Manu Drijvers, Tommaso Gagliardoni, Anja Lehmann, and Gregory Neven.
The wonderful world of global random oracles.
pages 280–312, 2018.
doi:10.1007/978-3-319-78381-9_11.

Alessandro Chiesa and Giacomo Fenzi.
zkSNARKs in the ROM with unconditional UC-security.
pages 67–89, 2024.
doi:10.1007/978-3-031-78011-0_3.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song.
Homomorphic encryption for arithmetic of approximate numbers.
pages 409–437, 2017.
doi:10.1007/978-3-319-70694-8_15.

Ran Canetti, Srinivasan Raghuraman, Silas Richelson, and Vinod Vaikuntanathan.
Chosen-ciphertext secure fully homomorphic encryption.
pages 213–240, 2017.
doi:10.1007/978-3-662-54388-7_8.

Craig Gentry.
A fully homomorphic encryption scheme.
PhD thesis, Stanford University, 2009.
crypto.stanford.edu/craig.

Chaya Ganesh, Yashvanth Kondi, Claudio Orlandi, Mahak Pancholi, Akira Takahashi, and Daniel Tschudi.
Witness-succinct universally-composable SNARKs.

pages 315–346, 2023.
doi:10.1007/978-3-031-30617-4_11.

Intel.
Intel® homomorphic encryption toolkit.
https://www.intel.com/content/www/us/en/developer/tools/homomorphic-encryption/overview.html.

Yuval Ishai and Anat Paskin.
Evaluating branching programs on encrypted data.
pages 575–594, 2007.
doi:10.1007/978-3-540-70936-7_31.

Hyesun Kwak, Dongwon Lee, Yongsoo Song, and Sameer Wagh.
A general framework of homomorphic encryption for multiple parties with non-interactive key-aggregation.
pages 403–430, 2024.
doi:10.1007/978-3-031-54773-7_16.

Kamil Kluczniak and Giacomo Santato.

On circuit private, multikey and threshold approximate homomorphic encryption.
Cryptology ePrint Archive, Report 2023/301, 2023.
URL: https://eprint.iacr.org/2023/301.

Baiyu Li and Daniele Micciancio.
On the security of homomorphic encryption on approximate numbers.
pages 648–677, 2021.
doi:10.1007/978-3-030-77870-5_23.

Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren.
On CCA-secure somewhat homomorphic encryption.
pages 55–72, 2012.
doi:10.1007/978-3-642-28496-0_4.

Anna Lysyanskaya and Leah Namisa Rosenbloom.
Universally composable Σ-protocols in the global random-oracle model.
pages 203–233, 2022.
doi:10.1007/978-3-031-22318-1_8.

Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan.
On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption.
pages 1219–1234, 2012.
doi:10.1145/2213977.2214086.