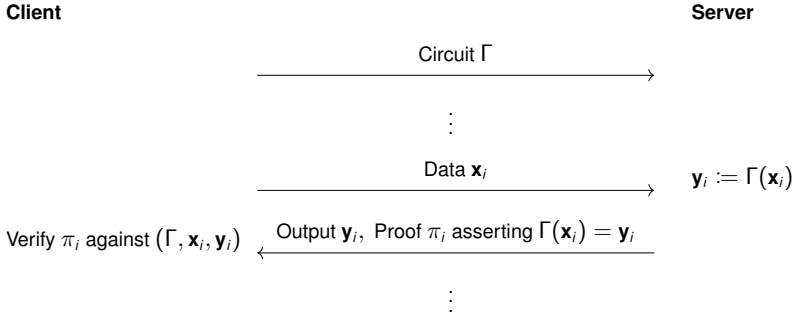
Chainable Functional Commitments for Unbounded-Depth Circuits

David Balbás[1,2], Dario Catalano[3], Dario Fiore[1], **Russell W. F. Lai**[4]

[1] IMDEA Software Institute, Spain
[2] Universidad Politecnica de Madrid, Spain
[3] University of Catania, Italy
[4] Aalto University, Finland

@TCC 2023

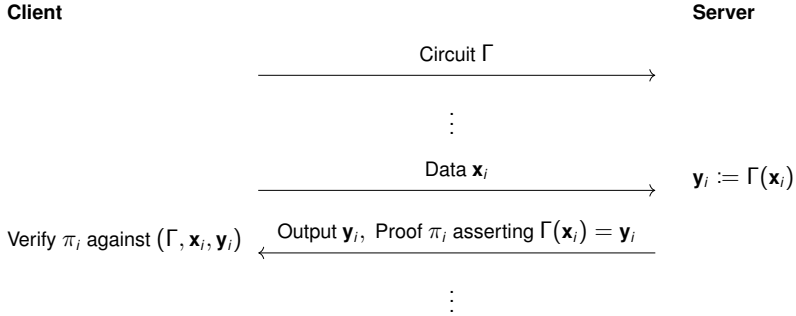
Motivating Application I: Verifiable Computation (VC)



Desiderata:

- Completeness: If indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$, then π_i passes verification.
- Soundness: If π_i passes verification, then indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$.
- (Amortised) Efficiency: Verifying π_i is faster than computing $\Gamma(\mathbf{x}_i)$, assuming preprocessing of Γ

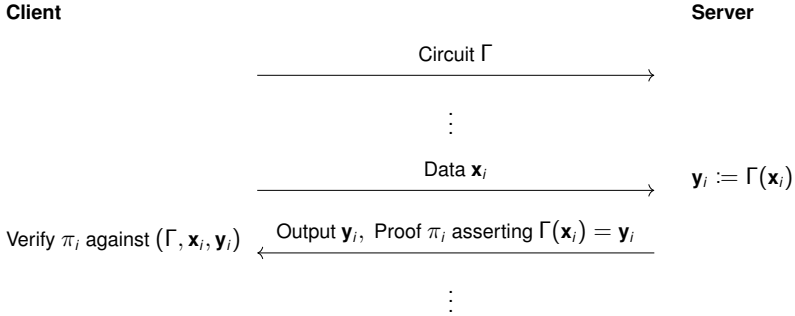
Motivating Application I: Verifiable Computation (VC)



Desiderata:

- Completeness: If indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$, then π_i passes verification.
- Soundness: If π_i passes verification, then indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$.
- (Amortised) Efficiency: Verifying π_i is faster than computing $\Gamma(\mathbf{x}_i)$, assuming preprocessing of Γ

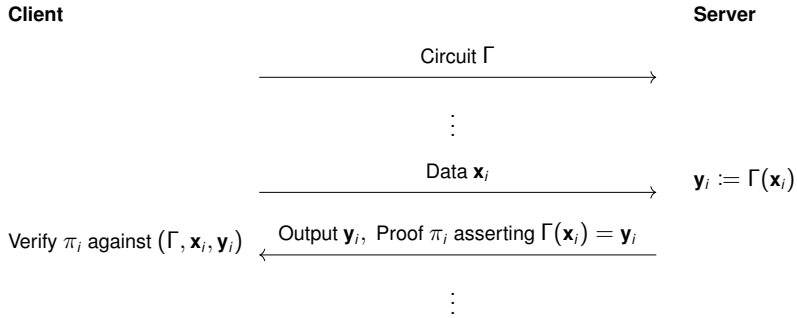
Motivating Application I: Verifiable Computation (VC)



Desiderata:

- † Completeness: If indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$, then π_i passes verification.
- † Soundness: If π_i passes verification, then indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$.
- † (Amortised) Efficiency: Verifying π_i is faster than computing $\Gamma(\mathbf{x}_i)$, assuming preprocessing of Γ

Motivating Application I: Verifiable Computation (VC)

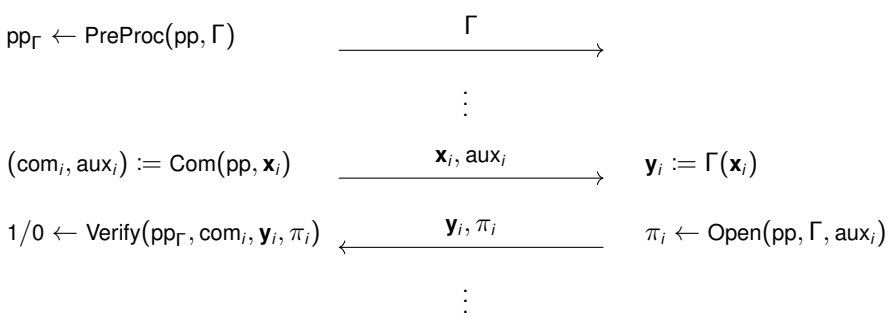


Desiderata:

- † Completeness: If indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$, then π_i passes verification.
- † Soundness: If π_i passes verification, then indeed $\Gamma(\mathbf{x}_i) = \mathbf{y}_i$.
- † (Amortised) Efficiency: Verifying π_i is faster than computing $\Gamma(\mathbf{x}_i)$, assuming preprocessing of Γ

(Preprocessing) Functional Commitments (FC) for VC

$\mathsf{FC} = (\mathsf{Setup}, \mathsf{PreProc}, \mathsf{Com}, \mathsf{Open}, \mathsf{Verify})$. Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ public.

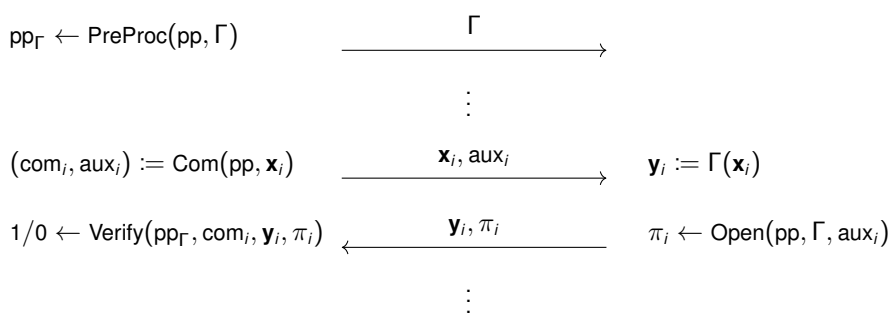


† Evaluation Binding: Infeasible to open (Γ, com_i) to $\mathbf{y}_i$ and $\mathbf{y}'_i$ for $\mathbf{y}_i \neq \mathbf{y}'_i$.
$\implies$ VC soundness: If $\mathbf{y}_i \neq \Gamma(\mathbf{x}_i)$, client opens (Γ, com_i) to $\mathbf{y}'_i = \Gamma(\mathbf{x}_i) \neq \mathbf{y}_i$, breaks eval. binding.

† Succinctness: $\mathsf{Verify}(\mathsf{pp}_\Gamma, \mathsf{com}_i, \mathbf{y}_i, \pi_i)$ takes time $o(|\Gamma|) \cdot \mathsf{poly}(\lambda)$.

(Preprocessing) Functional Commitments (FC) for VC

$\mathsf{FC} = (\mathsf{Setup}, \mathsf{PreProc}, \mathsf{Com}, \mathsf{Open}, \mathsf{Verify})$. Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ public.



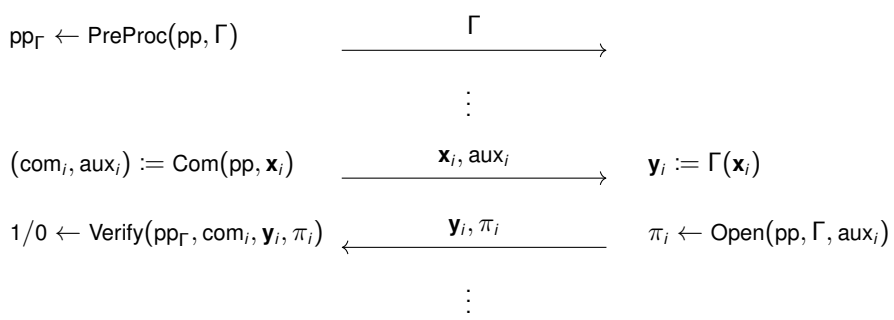
† Evaluation Binding: Infeasible to open (Γ, com_i) to $\mathbf{y}_i$ and $\mathbf{y}'_i$ for $\mathbf{y}_i \neq \mathbf{y}'_i$.

$\implies$ VC soundness: If $\mathbf{y}_i \neq \Gamma(\mathbf{x}_i)$, client opens (Γ, com_i) to $\mathbf{y}'_i = \Gamma(\mathbf{x}_i) \neq \mathbf{y}_i$, breaks eval. binding.

† Succinctness: $\mathsf{Verify}(\mathsf{pp}_\Gamma, \mathsf{com}_i, \mathbf{y}_i, \pi_i)$ takes time $o(|\Gamma|) \cdot \mathsf{poly}(\lambda)$.

(Preprocessing) Functional Commitments (FC) for VC

$\mathsf{FC} = (\mathsf{Setup}, \mathsf{PreProc}, \mathsf{Com}, \mathsf{Open}, \mathsf{Verify})$. Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ public.

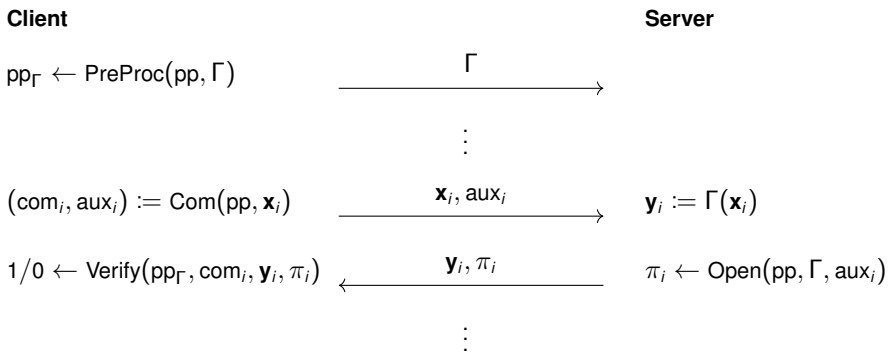


† Evaluation Binding: Infeasible to open (Γ, com_i) to $\mathbf{y}_i$ and $\mathbf{y}'_i$ for $\mathbf{y}_i \neq \mathbf{y}'_i$.
$\implies$ VC soundness: If $\mathbf{y}_i \neq \Gamma(\mathbf{x}_i)$, client opens (Γ, com_i) to $\mathbf{y}'_i = \Gamma(\mathbf{x}_i) \neq \mathbf{y}_i$, breaks eval. binding.

† Succinctness: $\mathsf{Verify}(\mathsf{pp}_\Gamma, \mathsf{com}_i, \mathbf{y}_i, \pi_i)$ takes time $o(|\Gamma|) \cdot \mathsf{poly}(\lambda)$.

(Preprocessing) Functional Commitments (FC) for VC

$\mathsf{FC} = (\mathsf{Setup}, \mathsf{PreProc}, \mathsf{Com}, \mathsf{Open}, \mathsf{Verify})$. Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ public.



† Evaluation Binding: Infeasible to open (Γ, com_i) to $\mathbf{y}_i$ and $\mathbf{y}'_i$ for $\mathbf{y}_i \neq \mathbf{y}'_i$.
$\implies$ VC soundness: If $\mathbf{y}_i \neq \Gamma(\mathbf{x}_i)$, client opens (Γ, com_i) to $\mathbf{y}'_i = \Gamma(\mathbf{x}_i) \neq \mathbf{y}_i$, breaks eval. binding.

† Succinctness: $\mathsf{Verify}(\mathsf{pp}_\Gamma, \mathsf{com}_i, \mathbf{y}_i, \pi_i)$ takes time $o(|\Gamma|) \cdot \mathsf{poly}(\lambda)$.

Motivating Application II: Bootstrapping SNARKs

$$\begin{aligned}\text{SNARKs} &= \text{Succinct Non-interactive ARguments of Knowledge} \\ &= \text{Very short proofs for NP language of interest}\end{aligned}$$

SNARKs for well-formedness of com $\xrightarrow{\text{FC for quadratic maps}}$ SNARKs for NP

[GW11]: Adaptively sound SNARKs require non-falsifiable assumptions or non-black-box reductions.

Rationale:

- † Minimise the component which requires non-falsifiable assumptions
- † Could be efficient – constructing special-purpose SNARKs is easier

Motivating Application II: Bootstrapping SNARKs

$$\begin{aligned}\text{SNARKs} &= \text{Succinct Non-interactive ARguments of Knowledge}\\ &= \text{Very short proofs for NP language of interest}\end{aligned}$$

SNARKs for well-formedness of com $\xrightarrow{\text{FC for quadratic maps}}$ SNARKs for NP

[GW11]: Adaptively sound SNARKs require non-falsifiable assumptions or non-black-box reductions.

Rationale:

- † Minimise the component which requires non-falsifiable assumptions
- † Could be efficient – constructing special-purpose SNARKs is easier

Motivating Application II: Bootstrapping SNARKs

$$\begin{aligned} \text{SNARKs} &= \text{Succinct Non-interactive ARguments of Knowledge} \\ &= \text{Very short proofs for NP language of interest} \end{aligned}$$

$$\text{SNARKs for well-formedness of com} \xrightarrow{\text{FC for quadratic maps}} \text{SNARKs for NP}$$

[GW11]: Adaptively sound SNARKs require non-falsifiable assumptions or non-black-box reductions.

Rationale:

- † Minimise the component which requires non-falsifiable assumptions
- † Could be efficient – constructing special-purpose SNARKs is easier

FC Landscape before Sept 2022

	Linear	Semi-Sparse Poly	Const.-Deg. Poly	NC1	Circuits
Non-Falsifiable					SNARK for NP
Falsifiable, Pairings	[LRY16] [LM19]	[LP20]	[CFT22]	[CFT22]	[This work] (bounded-width)
Falsifiable, Lattices			[ACLMT22]		[CP23] [WW23] (bounded-depth) [This work] (bounded-width)

FC Landscape Now

	Linear	Semi-Sparse Poly	Const.-Deg. Poly	NC1	Circuits
Non-Falsifiable					SNARK for NP
Falsifiable, Pairings	[LRY16] [LM19]	[LP20]	[CFT22]	[CFT22]	[This work] (bounded-width)
Falsifiable, Lattices			[ACLMT22]		[CP23] [WW23] (bounded-depth) [This work] (bounded-width)

Our Results: Chainable FC (CFC) for Unbounded-Depth Circuits

† New notion of Chainable Functional Commitments (CFC)

† Chaining CFC for quadratic polynomials $\Longrightarrow$ (C)FC for unbounded-depth* circuits

† CFC for quadratic polynomial maps from pairing- or lattice-based falsifiable assumptions

‡ Pairings: New assumption called "HiKer" (Hinted-Kernel), proven in generic group model (GGM)

‡ Lattices: Twin-version of the kRISIS assumption [ACLMT22] (equivalent to kRISIS [AFLN23])

† Corollary [CFT22]: Homomorphic signatures for unbounded-depth** circuits

*assuming bounded-width, opening size linear in depth

**assuming bounded-width, signature size linear in depth, multi-hop evaluation sequential only

Our Results: Chainable FC (CFC) for Unbounded-Depth Circuits

- † New notion of Chainable Functional Commitments (CFC)
- † Chaining CFC for quadratic polynomials $\Longrightarrow$ (C)FC for unbounded-depth* circuits
- † CFC for quadratic polynomial maps from pairing- or lattice-based falsifiable assumptions
  - ‡ Pairings: New assumption called "HiKer" (Hinted-Kernel), proven in generic group model (GGM)
  - ‡ Lattices: Twin-version of the kRISIS assumption [ACLMT22] (equivalent to kRISIS [AFLN23])
- † Corollary [CFT22]: Homomorphic signatures for unbounded-depth** circuits

*assuming bounded-width, opening size linear in depth

**assuming bounded-width, signature size linear in depth, multi-hop evaluation sequential only

Our Results: Chainable FC (CFC) for Unbounded-Depth Circuits

- † New notion of Chainable Functional Commitments (CFC)
- † Chaining CFC for quadratic polynomials $\implies$ (C)FC for unbounded-depth* circuits
- † CFC for quadratic polynomial maps from pairing- or lattice-based falsifiable assumptions
  - ‡ Pairings: New assumption called "HiKer" (Hinted-Kernel), proven in generic group model (GGM)
  - ‡ Lattices: Twin-version of the kRISIS assumption [ACLMT22] (equivalent to kRISIS [AFLN23])
- † Corollary [CFT22]: Homomorphic signatures for unbounded-depth** circuits

*assuming bounded-width, opening size linear in depth

**assuming bounded-width, signature size linear in depth, multi-hop evaluation sequential only

Our Results: Chainable FC (CFC) for Unbounded-Depth Circuits

- † New notion of Chainable Functional Commitments (CFC)
- † Chaining CFC for quadratic polynomials $\implies$ (C)FC for unbounded-depth* circuits
- † CFC for quadratic polynomial maps from pairing- or lattice-based falsifiable assumptions
  - ‡ Pairings: New assumption called "HiKer" (Hinted-Kernel), proven in generic group model (GGM)
  - ‡ Lattices: Twin-version of the kRISIS assumption [ACLMT22] (equivalent to kRISIS [AFLN23])
- † Corollary [CFT22]: Homomorphic signatures for unbounded-depth** circuits

*assuming bounded-width, opening size linear in depth
**assuming bounded-width, signature size linear in depth, multi-hop evaluation sequential only

Chainable Functional Commitments (CFC)

Recall FC $= (\mathsf{Setup}, \mathsf{PreProc}, \mathsf{Com}, \mathsf{Open}, \mathsf{Verify})$:

$$
\begin{aligned}
\mathsf{pp} &\leftarrow \mathsf{Setup}(1^\lambda) \\
\mathsf{pp}_\Gamma &\leftarrow \mathsf{PreProc}(\mathsf{pp}, \Gamma) \\
(\mathsf{com}, \mathsf{aux}) &\leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}) \\
\pi &\leftarrow \mathsf{Open}(\mathsf{pp}, \Gamma, \mathsf{aux}) \\
b &\leftarrow \mathsf{Verify}(\mathsf{pp}_\Gamma, \mathsf{com}, \mathbf{y}, \pi) \qquad \text{// } 1 \text{ if } \mathbf{y} = \Gamma(\mathbf{x})
\end{aligned}
$$

Evaluation Binding: Infeasible to open (Γ, com) to distinct $\mathbf{y}$ and $\mathbf{y}'$.

Chainable Functional Commitments (CFC)

$\mathsf{CFC} = (\mathsf{Setup}, \mathsf{PreProc}, \mathsf{Com}, \mathsf{Open}, \mathsf{Verify})$:

$$
\begin{aligned}
\mathsf{pp} &\leftarrow \mathsf{Setup}(1^\lambda) \\
\mathsf{pp}_\Gamma &\leftarrow \mathsf{PreProc}(\mathsf{pp}, \Gamma) \\
(\mathsf{com}, \mathsf{aux}) &\leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}) \\
\pi &\leftarrow \mathsf{Open}(\mathsf{pp}, \Gamma, (\mathsf{aux}_{\mathtt{in},i})_{i=1}^m) \\
b &\leftarrow \mathsf{Verify}(\mathsf{pp}_\Gamma, (\mathsf{com}_{\mathtt{in},i})_{i=1}^m, \mathsf{com}_{\mathtt{out}}, \pi) \qquad \text{// 1 if } \mathbf{y} = \Gamma(\mathbf{x}_1, \ldots, \mathbf{x}_m)
\end{aligned}
$$

Evaluation Binding: Infeasible to open $(\Gamma, \mathsf{com}_{\mathtt{in},1}, \ldots, \mathsf{com}_{\mathtt{in},m})$ to distinct $\mathsf{com}_{\mathtt{out}}$ and $\mathsf{com}'_{\mathtt{out}}$.

FC	CFC
1 committed input $\mathbf{x}$	m committed inputs $\mathbf{x}_1, \ldots, \mathbf{x}_m$
plaintext output $\mathbf{y}$	output $\mathbf{y}$ committed in $\mathsf{com}_{\mathtt{out}}$

Output commitment can be used as input commitment $\Longrightarrow$ Chainable

From CFC for Quadratic Polynomial Maps to (C)FC Circuits

Compiler in a nutshell:

- Given width-w depth-d circuit Γ and input $\mathbf{x} \in \{0,1\}^w$
- Partition Γ into multiplicative layers $\Gamma_1, \ldots, \Gamma_d$ where each Γ_i is quadratic, and

$$\begin{aligned} \mathbf{x}_0 &:= \mathbf{x}, \\ \mathbf{x}_1 &:= \Gamma_1(\mathbf{x}_0), \\ \mathbf{x}_2 &:= \Gamma_2(\mathbf{x}_0, \mathbf{x}_1), \\ &\vdots \\ \mathbf{y} := \mathbf{x}_d &:= \Gamma_d(\mathbf{x}_0, \ldots, \mathbf{x}_{d-1}). \end{aligned}$$

- To (C)FC-open $(\Gamma, \mathsf{com}_{\mathbf{x}})$ to $\mathbf{y} = \Gamma(\mathbf{x})$ (or $\mathsf{com}_{\mathbf{y}}$):
  - Commit to each intermediate layer $\mathsf{com}_{\mathbf{x}_1}, \ldots \mathsf{com}_{\mathbf{x}_d}$.
  - For each layer $i \in [d]$, CFC-open $(\Gamma_i, \mathsf{com}_{\mathbf{x}_0}, \ldots, \mathsf{com}_{\mathbf{x}_{i-1}})$ to $\mathsf{com}_{\mathbf{x}_i}$.

From CFC for Quadratic Polynomial Maps to (C)FC Circuits

Compiler in a nutshell:

- Given width-w depth-d circuit Γ and input $\mathbf{x} \in \{0,1\}^w$
- Partition Γ into multiplicative layers $\Gamma_1, \ldots, \Gamma_d$ where each Γ_i is quadratic, and

$$\begin{aligned} \mathbf{x}_0 &:= \mathbf{x}, \\ \mathbf{x}_1 &:= \Gamma_1(\mathbf{x}_0), \\ \mathbf{x}_2 &:= \Gamma_2(\mathbf{x}_0, \mathbf{x}_1), \\ &\vdots \\ \mathbf{y} := \mathbf{x}_d &:= \Gamma_d(\mathbf{x}_0, \ldots, \mathbf{x}_{d-1}). \end{aligned}$$

- To (C)FC-open $(\Gamma, \mathsf{com}_{\mathbf{x}})$ to $\mathbf{y} = \Gamma(\mathbf{x})$ (or $\mathsf{com}_{\mathbf{y}}$):
  - Commit to each intermediate layer $\mathsf{com}_{\mathbf{x}_1}, \ldots \mathsf{com}_{\mathbf{x}_d}$.
  - For each layer $i \in [d]$, CFC-open $(\Gamma_i, \mathsf{com}_{\mathbf{x}_0}, \ldots, \mathsf{com}_{\mathbf{x}_{i-1}})$ to $\mathsf{com}_{\mathbf{x}_i}$.

From CFC for Quadratic Polynomial Maps to (C)FC Circuits

Compiler in a nutshell:

- Given width-w depth-d circuit Γ and input $\mathbf{x} \in \{0,1\}^w$
- Partition Γ into multiplicative layers $\Gamma_1, \ldots, \Gamma_d$ where each Γ_i is quadratic, and

$$\begin{aligned} \mathbf{x}_0 &:= \mathbf{x}, \\ \mathbf{x}_1 &:= \Gamma_1(\mathbf{x}_0), \\ \mathbf{x}_2 &:= \Gamma_2(\mathbf{x}_0, \mathbf{x}_1), \\ &\vdots \\ \mathbf{y} := \mathbf{x}_d &:= \Gamma_d(\mathbf{x}_0, \ldots, \mathbf{x}_{d-1}). \end{aligned}$$

- To (C)FC-open $(\Gamma, \mathsf{com}_{\mathbf{x}})$ to $\mathbf{y} = \Gamma(\mathbf{x})$ (or $\mathsf{com}_{\mathbf{y}}$):
  - Commit to each intermediate layer $\mathsf{com}_{\mathbf{x}_1}, \ldots \mathsf{com}_{\mathbf{x}_d}$.
  - For each layer $i \in [d]$, CFC-open $(\Gamma_i, \mathsf{com}_{\mathbf{x}_0}, \ldots, \mathsf{com}_{\mathbf{x}_{i-1}})$ to $\mathsf{com}_{\mathbf{x}_i}$.

CFC from Pairings or Lattices

Technical highlights of constructions:

- † Commitment = Inner product between commitment key and message, Pedersen's style
- † Opening proof = "Preimage" of quadratic function of commitments
- † Main difficulty: Manage cross-terms when multiplying commitments to avoid unwanted collision
- † 3 types of commitment keys: Type-α, Type-β, Type-γ
- † Main type = Type-α
- † To prove quadratic relation $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, where $\mathbf{F}$, $\mathsf{Com}_\alpha(\mathbf{x})$, $\mathsf{Com}_\alpha(\mathbf{y})$, $\mathsf{Com}_\alpha(\mathbf{z})$ are public:
  - ‡ Compute $\mathsf{Com}_\beta(\mathbf{y})$ and $\mathsf{Com}_\gamma(\mathbf{z})$
  - ‡ Open $(\mathbf{F}, \mathsf{Com}_\alpha(\mathbf{x}), \mathsf{Com}_\beta(\mathbf{y}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$
  - ‡ Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{y}))$ to $\mathsf{Com}_\beta(\mathbf{y})$, where $\mathbf{I}$ is the identity matrix
  - ‡ Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{z}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$, where $\mathbf{I}$ is the identity matrix

CFC from Pairings or Lattices

Technical highlights of constructions:

- † Commitment = Inner product between commitment key and message, Pedersen's style
- † Opening proof = "Preimage" of quadratic function of commitments
- † Main difficulty: Manage cross-terms when multiplying commitments to avoid unwanted collision
- † 3 types of commitment keys: Type-α, Type-β, Type-γ
- † Main type = Type-α
- † To prove quadratic relation $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, where $\mathbf{F}$, $\mathsf{Com}_\alpha(\mathbf{x})$, $\mathsf{Com}_\alpha(\mathbf{y})$, $\mathsf{Com}_\alpha(\mathbf{z})$ are public:
  - ‡ Compute $\mathsf{Com}_\beta(\mathbf{y})$ and $\mathsf{Com}_\gamma(\mathbf{z})$
  - ‡ Open $(\mathbf{F}, \mathsf{Com}_\alpha(\mathbf{x}), \mathsf{Com}_\beta(\mathbf{y}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$
  - ‡ Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{y}))$ to $\mathsf{Com}_\beta(\mathbf{y})$, where $\mathbf{I}$ is the identity matrix
  - ‡ Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{z}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$, where $\mathbf{I}$ is the identity matrix

CFC from Pairings or Lattices

Technical highlights of constructions:

- Commitment = Inner product between commitment key and message, Pedersen's style
- Opening proof = "Preimage" of quadratic function of commitments
- Main difficulty: Manage cross-terms when multiplying commitments to avoid unwanted collision
- 3 types of commitment keys: Type-α, Type-β, Type-γ
- Main type = Type-α
- To prove quadratic relation $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, where $\mathbf{F}$, $\mathsf{Com}_\alpha(\mathbf{x})$, $\mathsf{Com}_\alpha(\mathbf{y})$, $\mathsf{Com}_\alpha(\mathbf{z})$ are public:
  - Compute $\mathsf{Com}_\beta(\mathbf{y})$ and $\mathsf{Com}_\gamma(\mathbf{z})$
  - Open $(\mathbf{F}, \mathsf{Com}_\alpha(\mathbf{x}), \mathsf{Com}_\beta(\mathbf{y}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$
  - Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{y}))$ to $\mathsf{Com}_\beta(\mathbf{y})$, where $\mathbf{I}$ is the identity matrix
  - Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{z}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$, where $\mathbf{I}$ is the identity matrix

CFC from Pairings or Lattices

Technical highlights of constructions:

- Commitment = Inner product between commitment key and message, Pedersen's style
- Opening proof = "Preimage" of quadratic function of commitments
- Main difficulty: Manage cross-terms when multiplying commitments to avoid unwanted collision
- 3 types of commitment keys: Type-α, Type-β, Type-γ
- Main type = Type-α
- To prove quadratic relation $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, where $\mathbf{F}$, $\mathsf{Com}_\alpha(\mathbf{x})$, $\mathsf{Com}_\alpha(\mathbf{y})$, $\mathsf{Com}_\alpha(\mathbf{z})$ are public:
  - Compute $\mathsf{Com}_\beta(\mathbf{y})$ and $\mathsf{Com}_\gamma(\mathbf{z})$
  - Open $(\mathbf{F}, \mathsf{Com}_\alpha(\mathbf{x}), \mathsf{Com}_\beta(\mathbf{y}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$
  - Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{y}))$ to $\mathsf{Com}_\beta(\mathbf{y})$, where $\mathbf{I}$ is the identity matrix
  - Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{z}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$, where $\mathbf{I}$ is the identity matrix

CFC from Pairings or Lattices

Technical highlights of constructions:

- Commitment = Inner product between commitment key and message, Pedersen's style
- Opening proof = "Preimage" of quadratic function of commitments
- Main difficulty: Manage cross-terms when multiplying commitments to avoid unwanted collision
- 3 types of commitment keys: Type-α, Type-β, Type-γ
- Main type = Type-α
- To prove quadratic relation $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, where $\mathbf{F}$, $\mathsf{Com}_\alpha(\mathbf{x})$, $\mathsf{Com}_\alpha(\mathbf{y})$, $\mathsf{Com}_\alpha(\mathbf{z})$ are public:
  - Compute $\mathsf{Com}_\beta(\mathbf{y})$ and $\mathsf{Com}_\gamma(\mathbf{z})$
  - Open $(\mathbf{F}, \mathsf{Com}_\alpha(\mathbf{x}), \mathsf{Com}_\beta(\mathbf{y}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$
  - Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{y}))$ to $\mathsf{Com}_\beta(\mathbf{y})$, where $\mathbf{I}$ is the identity matrix
  - Open $(\mathbf{I}, \mathsf{Com}_\alpha(\mathbf{z}))$ to $\mathsf{Com}_\gamma(\mathbf{z})$, where $\mathbf{I}$ is the identity matrix

CFC from Pairings or Lattices

- Commitments: $\mathsf{Com}_\alpha(\mathbf{x}) = \sum_i \alpha_i \cdot x_i \quad \mathsf{Com}_\beta(\mathbf{y}) = \sum_j \beta_j \cdot y_j \quad \mathsf{Com}_\gamma(\mathbf{z}) = \sum_k \gamma_k \cdot z_k$
- Want to prove: $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, i.e. $\forall k,\ z_k = \sum_{i,j} f_{i,j,k} x_i y_j$
- Verifier "commits" to $\mathbf{F} = (f_{i,j,k})_{i,j,k}$ by: $\mathsf{Com}_{\alpha^{-1},\beta^{-1},\gamma}(\mathbf{F}) = \sum_{i,j,k} \alpha_i^{-1} \cdot \beta_j^{-1} \cdot \gamma_k \cdot f_{i,j,k}$
- Observe:

$$\mathsf{Com}_{\alpha^{-1},\beta^{-1},\gamma}(\mathbf{F}) \cdot \mathsf{Com}_\alpha(\mathbf{x}) \cdot \mathsf{Com}_\beta(\mathbf{y}) - \mathsf{Com}_\gamma(\mathbf{z}) = \sum_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}} \frac{\alpha_{i'}}{\alpha_i} \cdot \frac{\beta_{j'}}{\beta_j} \cdot \gamma_k \cdot f_{i,j,k} x_{i'} y_{j'}$$

  (Pairing-based construction is more complicated: Cannot simply multiply commitments.)
- In CRS, give away "preimages" of $\left(\frac{\alpha_{i'}}{\alpha_i} \cdot \frac{\beta_{j'}}{\beta_j} \cdot \gamma_k\right)_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}}$ as hints.
- Prover computes "preimage" by combining hints linearly with coefficients $\left(f_{i,j,k} x_{i'} y_{j'}\right)_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}}$.

CFC from Pairings or Lattices

- Commitments: $\mathsf{Com}_\alpha(\mathbf{x}) = \sum_i \alpha_i \cdot x_i \quad \mathsf{Com}_\beta(\mathbf{y}) = \sum_j \beta_j \cdot y_j \quad \mathsf{Com}_\gamma(\mathbf{z}) = \sum_k \gamma_k \cdot z_k$
- Want to prove: $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, i.e. $\forall k,\ z_k = \sum_{i,j} f_{i,j,k} x_i y_j$
- Verifier "commits" to $\mathbf{F} = (f_{i,j,k})_{i,j,k}$ by: $\mathsf{Com}_{\alpha^{-1},\beta^{-1},\gamma}(\mathbf{F}) = \sum_{i,j,k} \alpha_i^{-1} \cdot \beta_j^{-1} \cdot \gamma_k \cdot f_{i,j,k}$
- Observe:

$$\mathsf{Com}_{\alpha^{-1},\beta^{-1},\gamma}(\mathbf{F}) \cdot \mathsf{Com}_\alpha(\mathbf{x}) \cdot \mathsf{Com}_\beta(\mathbf{y}) - \mathsf{Com}_\gamma(\mathbf{z}) = \sum_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}} \frac{\alpha_{i'}}{\alpha_i} \cdot \frac{\beta_{j'}}{\beta_j} \cdot \gamma_k \cdot f_{i,j,k} x_{i'} y_{j'}$$

(Pairing-based construction is more complicated: Cannot simply multiply commitments.)

- In CRS, give away "preimages" of $\left(\frac{\alpha_{i'}}{\alpha_i} \cdot \frac{\beta_{j'}}{\beta_j} \cdot \gamma_k\right)_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}}$ as hints.
- Prover computes "preimage" by combining hints linearly with coefficients $\left(f_{i,j,k} x_{i'} y_{j'}\right)_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}}$.

CFC from Pairings or Lattices

- Commitments: $\mathsf{Com}_\alpha(\mathbf{x}) = \sum_i \alpha_i \cdot x_i \quad \mathsf{Com}_\beta(\mathbf{y}) = \sum_j \beta_j \cdot y_j \quad \mathsf{Com}_\gamma(\mathbf{z}) = \sum_k \gamma_k \cdot z_k$
- Want to prove: $\mathbf{z} = \mathbf{F}(\mathbf{x} \otimes \mathbf{y})$, i.e. $\forall k,\ z_k = \sum_{i,j} f_{i,j,k} x_i y_j$
- Verifier "commits" to $\mathbf{F} = (f_{i,j,k})_{i,j,k}$ by: $\mathsf{Com}_{\alpha^{-1},\beta^{-1},\gamma}(\mathbf{F}) = \sum_{i,j,k} \alpha_i^{-1} \cdot \beta_j^{-1} \cdot \gamma_k \cdot f_{i,j,k}$
- Observe:

$$\mathsf{Com}_{\alpha^{-1},\beta^{-1},\gamma}(\mathbf{F}) \cdot \mathsf{Com}_\alpha(\mathbf{x}) \cdot \mathsf{Com}_\beta(\mathbf{y}) - \mathsf{Com}_\gamma(\mathbf{z}) = \sum_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}} \frac{\alpha_{i'}}{\alpha_i} \cdot \frac{\beta_{j'}}{\beta_j} \cdot \gamma_k \cdot f_{i,j,k} x_{i'} y_{j'}$$

(Pairing-based construction is more complicated: Cannot simply multiply commitments.)

- In CRS, give away "preimages" of $\left(\frac{\alpha_{i'}}{\alpha_i} \cdot \frac{\beta_{j'}}{\beta_j} \cdot \gamma_k \right)_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}}$ as hints.
- Prover computes "preimage" by combining hints linearly with coefficients $\left(f_{i,j,k} x_{i'} y_{j'} \right)_{\substack{i,i',j,j',k \\ i \neq i', j \neq j'}}$.

Conclusion

- † New notion of Chainable Functional Commitments (CFC)
- † Chaining CFC for quadratic polynomials $\Longrightarrow$ (C)FC for unbounded-depth* circuits
- † CFC for quadratic polynomial maps from pairing- or lattice-based falsifiable assumptions
- † Corollary [CFT22]: Homomorphic signatures for unbounded-depth** circuits

*assuming bounded-width, opening size linear in depth

**assuming bounded-width, signature size linear in depth, multi-hop evaluation sequential only

Russell W. F. Lai
Aalto University, Finland
russell.lai@aalto.fi
russell-lai.hk

ia.cr/2022/1365

We are hiring 2 PhD students in lattice-based cryptography.
Job ad will appear in IACR job board soon.

Conclusion

- † New notion of Chainable Functional Commitments (CFC)
- † Chaining CFC for quadratic polynomials $\Longrightarrow$ (C)FC for unbounded-depth* circuits
- † CFC for quadratic polynomial maps from pairing- or lattice-based falsifiable assumptions
- † Corollary [CFT22]: Homomorphic signatures for unbounded-depth** circuits

*assuming bounded-width, opening size linear in depth

**assuming bounded-width, signature size linear in depth, multi-hop evaluation sequential only

Russell W. F. Lai
Aalto University, Finland
russell.lai@aalto.fi
russell-lai.hk

ia.cr/2022/1365

We are hiring 2 PhD students in lattice-based cryptography.
Job ad will appear in IACR job board soon.

References I

[ACLMT22] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. “Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable - (Extended Abstract)”. In: *CRYPTO 2022, Part II*. Vol. 13508. 2022, pp. 102–132. DOI: 10.1007/978-3-031-15979-4_4 (pages 13–18).

[AFLN23] Martin R. Albrecht, Giacomo Fenzi, Oleksandra Lapiha, and Ngoc Khanh Nguyen. *SLAP: Succinct Lattice-Based Polynomial Commitments from Standard Assumptions*. Cryptology ePrint Archive, Paper 2023/1469. https://eprint.iacr.org/2023/1469. 2023 (pages 15–18).

[CFT22] Dario Catalano, Dario Fiore, and Ida Tucker. “Additive-Homomorphic Functional Commitments and Applications to Homomorphic Signatures”. In: *ASIACRYPT 2022, Part IV*. Vol. 13794. 2022, pp. 159–188. DOI: 10.1007/978-3-031-22972-5_6 (pages 13–18, 32, 33).

References II

[CP23] Leo de Castro and Chris Peikert. “Functional Commitments for All Functions, with Transparent Setup and from SIS”. In: *EUROCRYPT 2023, Part III*. Vol. 14006. 2023, pp. 287–320. DOI: 10.1007/978-3-031-30620-4_10 (pages 13, 14).

[GW11] Craig Gentry and Daniel Wichs. “Separating succinct non-interactive arguments from all falsifiable assumptions”. In: *43rd ACM STOC*. 2011, pp. 99–108. DOI: 10.1145/1993636.1993651 (pages 10–12).

[LM19] Russell W. F. Lai and Giulio Malavolta. “Subvector Commitments with Application to Succinct Arguments”. In: *CRYPTO 2019, Part I*. Vol. 11692. 2019, pp. 530–560. DOI: 10.1007/978-3-030-26948-7_19 (pages 13, 14).

[LP20] Helger Lipmaa and Kateryna Pavlyk. “Succinct Functional Commitment for a Large Class of Arithmetic Circuits”. In: *ASIACRYPT 2020, Part III*. Vol. 12493. 2020, pp. 686–716. DOI: 10.1007/978-3-030-64840-4_23 (pages 13, 14).

References III

[LRY16] Benoît Libert, Somindu C. Ramanna, and Moti Yung. “Functional Commitment Schemes: From Polynomial Commitments to Pairing-Based Accumulators from Simple Assumptions”. In: *ICALP 2016*. Vol. 55. 2016, 30:1–30:14. DOI: 10.4230/LIPIcs.ICALP.2016.30 (pages 13, 14).

[WW23] Hoeteck Wee and David J. Wu. “Succinct Vector, Polynomial, and Functional Commitments from Lattices”. In: *EUROCRYPT 2023, Part III*. Vol. 14006. 2023, pp. 385–416. DOI: 10.1007/978-3-031-30620-4_13 (pages 13, 14).