Succinct Functional Commitment for a Large Class of Arithmetic Circuits

Asiacrypt 2020 Long Presentation

Helger Lipmaa, Kateryna Pavlyk

Simula UiB, Norway

December 1, 2020

Commitment Schemes

$\mathsf{CS} = (\mathbf{KC}, \mathbf{com}, \mathbf{open}, \mathbf{verify})$

Prover/Committer		Verifier
$(\mathsf{ck}, \mathsf{tk}) \leftarrow \mathsf{KC}(1^\lambda),\ M$		
$(C, D) \leftarrow \mathbf{com}(\mathsf{ck}, M; r)$	$\mathsf{ck}, \mathrm{C} \xrightarrow{\quad\quad}$	
$\mathsf{op} \leftarrow \mathbf{open}(\mathsf{ck}, C, D)$	$\mathsf{op} \xrightarrow{\quad\quad}$	$\{0,1\} \leftarrow \mathbf{verify}(\mathsf{ck}, C, \mathsf{op})$ accepts or rejects

Security Properties

- **Hiding:** C should not reveal information about the message M.

  (*distributions of* $\mathbf{com}(M_1)$ *and* $\mathbf{com}(M_2)$ *are (computationally) indistinguishable for* $M_1 \neq M_2$)

- **Binding:** it is impossible to change C after commitment.

  (*its (computationally) hard to find such* $C, M_0, M_1, \mathsf{op}_0, \mathsf{op}_1$ *that* $M_0 \neq M_1$ *but* $\mathbf{verify}(\mathsf{ck}, C, \mathsf{op}_0) = \mathbf{verify}(\mathsf{ck}, C, \mathsf{op}_1) = 1$)

Evolution of Commitment Schemes

- **VC**: commit to a vector $\boldsymbol{\alpha}$ of values, open at specific positions (i.e., prove that α_i is the i-th committed message) [Catalano, Fiore, 2013]
- **PC**: commit to a polynomial $\alpha(X) \in \mathbb{Z}_p[X]$, open evaluation $\alpha(\beta)$ on certain input β [Kate, Zaverucha, Golbderg, 2010]
- **FC** (*for linear functions only*): commit to a vector $\boldsymbol{\alpha}$, open to $y = \langle \boldsymbol{\beta}, \boldsymbol{\alpha} \rangle = \sum \beta_i \alpha_i$ for a public $\boldsymbol{\beta}$ (based on [Izabachéne, Libert, Vergnaud, 2011], [Libert, Ramanna, Yung, 2016])
- Recently very popular due to SNARKs, cryptocurrencies.
- a large gap ...

Our Main Result

Succinct SNARK-based FC for a large class of circuits based on falsifiable assumptions: commit to a α, open to $\xi = \mathsf{Crkt}(\alpha, \beta)$ (verifier chooses β in the time of opening) [this work]

- *non-falsifiable* assumptions $\Rightarrow$ SFC for all poly-size circuits (using SNARKs as a black-box: opening consists of a SNARK argument that $\mathsf{Crkt}(\alpha, \beta)$ is equal to the claimed value)
- non-succinct NIZK $\Rightarrow$ non-succinct FC from *falsifiable* assumptions (Bitansky)
- Constructing SFC is a much **harder task** than constructing nSFC since one cannot rely on the SNARK as a black-box.

Problem

Design

- a succinct
  functional commitment scheme
- for wider classes of functions
- under nice (falsifiable) assumptions
- in groups with a bilinear map

Functional Commitment Scheme: Syntax

Let $\mathcal{D}$ be a domain, CC be a class of circuits, Crkt $\in$ CC

$$\mathsf{Crkt} : \mathcal{D}^{\mu_\alpha} \times \mathcal{D}^{\mu_\beta} \to \mathcal{D}^{\kappa} : (\boldsymbol{\alpha}, \boldsymbol{\beta}) \mapsto \boldsymbol{\xi} = \boldsymbol{\mathcal{F}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) := (\mathcal{F}_i(\boldsymbol{\alpha}, \boldsymbol{\beta})_{i=1}^{\kappa})$$

$(\mathsf{ck}, \mathsf{tk}) \leftarrow \mathsf{KC}(1^\lambda)$

Prover/Committer		**Verifier**
$(C, D) \leftarrow \mathbf{com}(\mathsf{ck}, \boldsymbol{\alpha}; r)$	$\xrightarrow{C}$	
	$\xleftarrow{\beta}$	
$(\xi, \mathsf{op}_\xi) \leftarrow \mathbf{open}(\ldots, \boldsymbol{\beta})$	$\xrightarrow{(\xi, \mathsf{op}_\xi)}$	$\{0,1\} \leftarrow \mathbf{verify}(\ldots)$
		Accepts that $\mathsf{Crkt}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \xi$

Example: $\langle \cdot, \cdot \rangle : \mathcal{D}^n \times \mathcal{D}^n \to \mathcal{D}$, $\langle \cdot, \cdot \rangle = \sum_{i=1}^{n} \alpha_i \beta_i$, $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{D}^n$

Security and Efficiency Requirements

Stronger hiding:

- perfect com-hiding: *commitments* do not reveal any information about $\boldsymbol{\alpha}$
- perfect open-hiding: *commitment and the opening together* do not reveal more information on $\boldsymbol{\alpha}$ than the values $\mathsf{Crkt}(\boldsymbol{\alpha}, \boldsymbol{\beta}_i)$ on queried values $\boldsymbol{\beta}_i$
- perfect zero-knowledge: hiding in the sense of *simulatability*

Evaluation-binding:

- given a commitment, it is intractable to open it to $\boldsymbol{\xi} = \mathsf{Crkt}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $\boldsymbol{\xi}' = \mathsf{Crkt}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ for $\boldsymbol{\xi} \neq \boldsymbol{\xi}'$

Succinctness:

- both the commitment and the opening have length $\mathsf{polylog}(|\boldsymbol{\alpha}|, |\boldsymbol{\beta}|)$

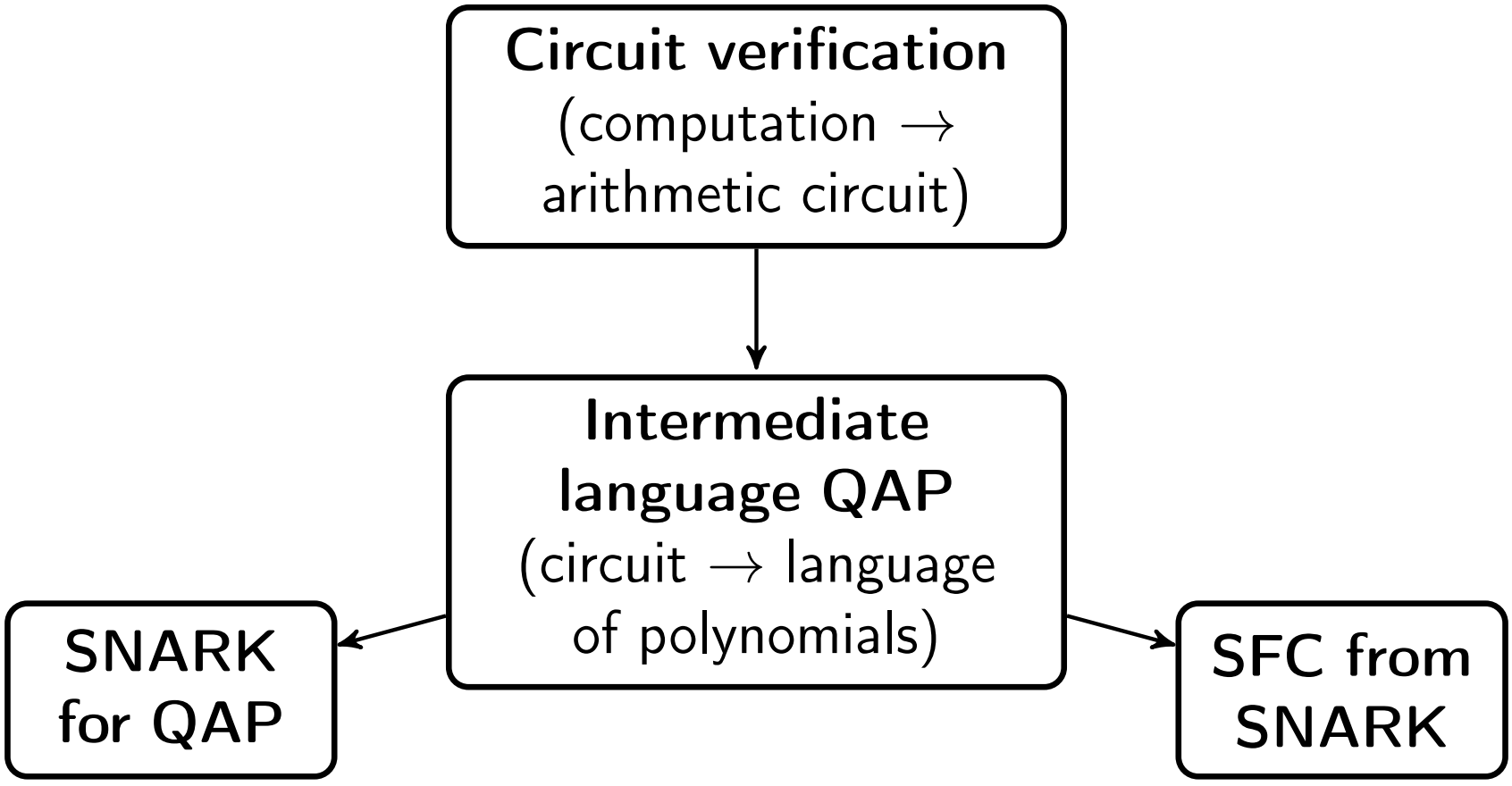
Inspiration: SNARK

Succint **N**on-interactive **AR**gument of **K**nowledge: a computationally sound proof, where we require that a polynomial-time prover cannot convince the verifier of a false statement

[Groth, 2016]: most efficient SNARK for *arithmetic circuits satisfiability*

[Lipmaa, 2019]: less trapdoors, HAK-assumption

Roadmap


Circuit verification
(computation → arithmetic circuit)
Intermediate language QAP
(circuit → language of polynomials)
SNARK for QAP
SFC from SNARK

Techniques: SNARK vs FC

	SNARK for Crkt	**FC** for Crkt
objectives	**knowledge-soundness:** impossible to give a wrong proof for a *single* false statement	**binding:** impossible to give openings for *two* contradicting statements
access to input	P has *full* access to both inputs	*gradually*: P gets more from V later
argument	*single* bit string	*division* into 2 parts: commitment and opening
assumpt.	non-falsifiable	**falsifiable**

Large Class of Circuits?

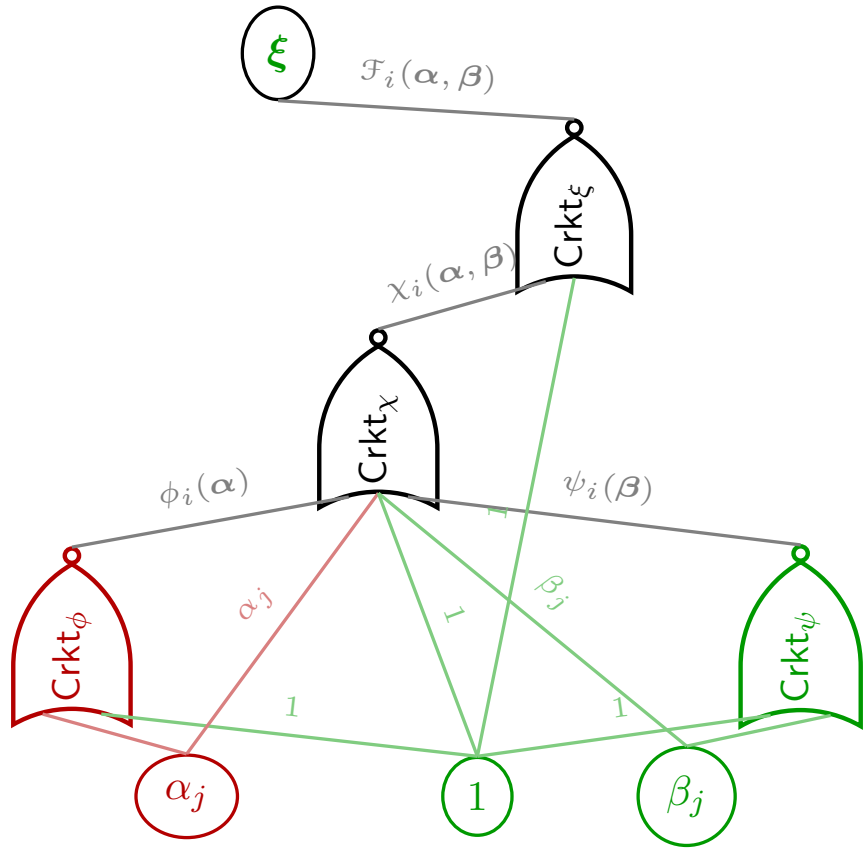
We compile $\mathsf{Crkt} \mapsto \mathsf{Crkt}^* = (\mathsf{Crkt}_\phi, \mathsf{Crkt}_\psi, \mathsf{Crkt}_\chi, \mathsf{Crkt}_\xi)$

Compilation restricts CC since Crkt_χ **must have multiplicative depth** 1

- No other restrictions

Define
$\mathsf{CC} := \{\mathsf{Crkt} : \mathsf{Crkt}_\chi \text{ has m.d. } 1\}$

Polynomial $f \in \mathsf{CC}$ if it can be computed by $\mathsf{Crkt} \in \mathsf{CC}$



- Any **sparse polynomial** $f \in \mathsf{CC}$.
- Exists a **non-sparse polynomial** $f \in \mathsf{CC}$.
- Exists a polynomial $f \in \mathbf{VP}$ such that $f \notin \mathsf{CC}$.

QAP representation of Crkt*

- Construct $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$, such that

$$\boldsymbol{U}\boldsymbol{a} \circ \boldsymbol{V}\boldsymbol{a} = \boldsymbol{W}\boldsymbol{a}$$

iff Crkt* is correctly computed. Here

$$\boldsymbol{a} = 1||\boldsymbol{\alpha}||\phi(\boldsymbol{\alpha})||\boldsymbol{\beta}||\psi(\boldsymbol{\beta})||\boldsymbol{\chi}(\boldsymbol{\alpha},\boldsymbol{\beta})||\boldsymbol{\mathcal{F}}(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathbb{Z}_p^{\mu}$$

is the value of all wires of Crkt*.

- Construct matrices for checking that the various subcircuits of the Crkt* are correctly computed:

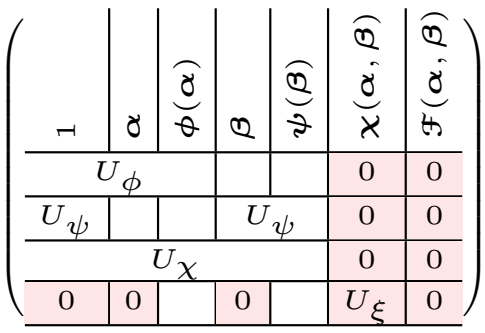
$$U_{\phi}\boldsymbol{\alpha}^* \circ V_{\phi}\boldsymbol{\alpha}^* = \boldsymbol{\phi}(\boldsymbol{\alpha}) \qquad U_{\psi}\boldsymbol{\beta}^* \circ V_{\psi}\boldsymbol{\beta}^* = \boldsymbol{\psi}(\boldsymbol{\beta})$$

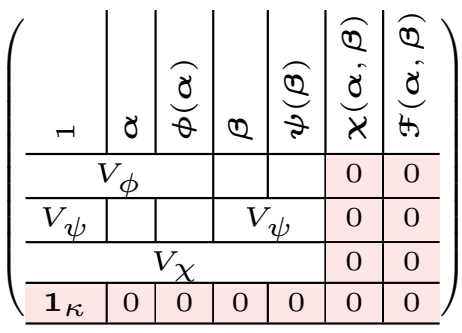
$$U_{\chi}\begin{pmatrix}\boldsymbol{\alpha}^*\\ \boldsymbol{\beta}\\ \boldsymbol{\psi}(\boldsymbol{\beta})\end{pmatrix} \circ V_{\chi}\begin{pmatrix}\boldsymbol{\alpha}^*\\ \boldsymbol{\beta}\\ \boldsymbol{\psi}(\boldsymbol{\beta})\end{pmatrix} = \boldsymbol{\chi}(\boldsymbol{\alpha},\boldsymbol{\beta}) \qquad U_{\xi}\begin{pmatrix}1\\ \boldsymbol{\chi}(\boldsymbol{\alpha},\boldsymbol{\beta})\end{pmatrix} \circ 1 = \boldsymbol{\mathcal{F}}(\boldsymbol{\alpha},\boldsymbol{\beta})$$

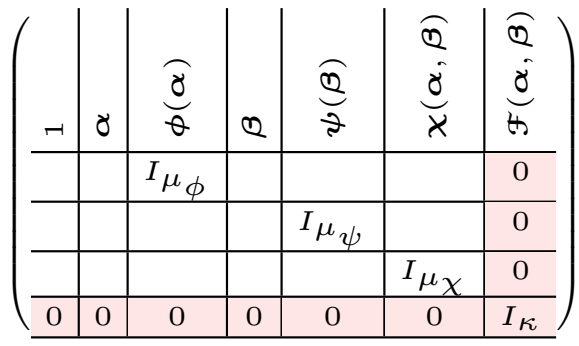
for $\boldsymbol{\alpha}^* = (1||\boldsymbol{\alpha}||\boldsymbol{\phi}(\boldsymbol{\alpha}))$ and $\boldsymbol{\beta}^* = (1||\boldsymbol{\beta}||\boldsymbol{\psi}(\boldsymbol{\beta}))$.

Requirement on Matrices

- Need to fix pink entries in matrices $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$:







- Some entries are fixed for the evaluation-binding proof
- Others are fixed for the assumption to be secure
- Compiled circuit Crkt* satisfies given conditions

Groth's SNARK

$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Kcrs}(1^\lambda)$

Prover/Committer		Verifier
$\pi \leftarrow \mathbf{P}(\ldots \alpha, \beta, \ldots)$		
// $\pi = ([\mathsf{A}, \mathsf{C_{sp}}]_1, [\mathsf{B}]_2)$	$\xrightarrow{\pi}$	Recompute $[\mathsf{C_p}]_1$
		$[\mathsf{C}]_1 \leftarrow [\mathsf{C_{sp}}]_1 + [\mathsf{C_p}]_1$
		$\pi' \leftarrow ([\mathsf{A}, \mathsf{C}]_1, [\mathsf{B}]_2)$
		$\{0,1\} \leftarrow \mathbf{verify}(\ldots \beta, \pi')$

New Succinct Functional Commitment

$(\mathsf{ck}, \mathsf{tk}) \leftarrow \mathsf{KC}(1^\lambda)$

Prover/Committer		Verifier
$com \leftarrow \mathbf{com}(\ldots \alpha, \ldots)$		
// $com = ([\mathsf{A}_s, \mathsf{B}_i^{aux}]_1, [\mathsf{B}_s]_2)$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\beta}$	
$[\mathsf{C_{sp}}]_1 \leftarrow \mathbf{open}(\ldots \alpha, \beta, \ldots)$	$\xrightarrow{[\mathsf{C_{sp}}]_1}$	Recompute $[\mathsf{A}_p, \mathsf{C_p}]_1, [\mathsf{B}_p]_2$
		$[\mathsf{A}]_1 \leftarrow [\mathsf{A}_s]_1 + [\mathsf{A}_p]_1$;
		$[\mathsf{B}]_2 \leftarrow [\mathsf{B}_s]_2 + [\mathsf{B}_p]_2$
		$[\mathsf{C}]_1 \leftarrow [\mathsf{C_{sp}}]_1 + [\mathsf{C_p}]_1$
		$\pi' \leftarrow ([\mathsf{A}, \mathsf{C}]_1, [\mathsf{B}]_2)$
		$\{0, 1\} \leftarrow \mathbf{verify}(\ldots \beta, \pi')$

Such division possible due to circuit compilation

Main Result: SFC for $\mathsf{Crkt} \in \mathsf{CC}$

Let $\mathsf{Crkt} \in \mathsf{CC}$. We construct succinct functional commitment scheme $\mathbf{FC}^{\mathsf{Crkt}}$ for Crkt.

Theorem.

- **$\mathbf{FC}^{\mathsf{Crkt}}$ is complete and zero-knowledge.**
- **If the $(\mathcal{R}, \mathcal{S}, \{f_i\})$-computational span-uber-assumption holds in $\mathbb{G}_1$, then $\mathbf{FC}^{\mathsf{Crkt}}$ is computationally evaluation-binding.**

Intuition: Evaluation-Binding Proof

- We are given collusion, s.t. verifier accepts:

$$[\mathsf{A}_s + \mathsf{A}_p + y^{\delta}]_1 \bullet [\mathsf{B}_s + \mathsf{B}_p + y^{\eta}]_2 = [\mathsf{C}_{\mathsf{sp}}]_1 \bullet [1]_2 + [\mathsf{C}_{\mathsf{p}}]_1 \bullet [y^{\gamma}]_2 + [y^{\delta+\eta}]_T$$
$$[\mathsf{A}_s + \tilde{\mathsf{A}}_p + y^{\delta}]_1 \bullet [\mathsf{B}_s + \tilde{\mathsf{B}}_p + y^{\eta}]_2 = [\tilde{\mathsf{C}}_{\mathsf{sp}}]_1 \bullet [1]_2 + [\tilde{\mathsf{C}}_p]_1 \bullet [y^{\gamma}]_2 + [y^{\delta+\eta}]_T$$

- Circuit compilation guarantees in particular that $\mathsf{B}_p = \tilde{\mathsf{B}}_p$
- Thus we get

$$[\mathsf{A}_s + (\mathsf{A}_p - \tilde{\mathsf{A}}_p) + y^{\delta}]_1 \bullet [\mathsf{B}_s + \mathsf{B}_p + y^{\eta}]_2$$
$$= [\mathsf{C}_{\mathsf{sp}} - \tilde{\mathsf{C}}_{\mathsf{sp}}]_1 \bullet [1]_2 + [\mathsf{C}_{\mathsf{p}} - \tilde{\mathsf{C}}_{\mathsf{p}}]_1 \bullet [y^{\gamma}]_2$$

- We break a **span-uber-assumption**, by using
  - The fact that many elements of the matrices are zeros
  - Elements $[\mathsf{B}_i^{aux}]_1$ (one per circuit output)

$(\mathcal{R}, \mathcal{S}, \{f_i\})$-Span-Uber-Assumption

$(\mathcal{R}, \mathcal{S}, \mathcal{T}, f)$-Uber Assumption in $\mathbb{G}_\iota$, $\iota \in \{1, 2, T\}$:

$\mathcal{R} = \{r_i(\boldsymbol{x})\}$, $\mathcal{S} = \{s_j(\boldsymbol{x})\}$, $\mathcal{T} = \{t_k(\boldsymbol{x})\}$, f

$$[\mathcal{R}(\boldsymbol{x})]_1, [\mathcal{S}(\boldsymbol{x})]_2, [\mathcal{T}(\boldsymbol{x})]_T \quad \not\Rightarrow \quad [f(\boldsymbol{x})]_\iota$$

// Famous assumption

$(\mathcal{R}, \mathcal{S}, \{f_i\})$-Span-Uber Assumption: $f_i \notin span(\mathcal{R})$

$$[\mathcal{R}(\boldsymbol{x})]_1, [\mathcal{S}(\boldsymbol{x})]_2 \quad \not\Rightarrow \quad (\boldsymbol{\Delta}, \textstyle\sum_{i=1}^{\kappa} \Delta_i [f_i(\boldsymbol{x})]_1)$$

// New assumption

- $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_\kappa) \neq \boldsymbol{0}$, where $\Delta_i = \sigma_i - \sigma'_i$ – component-wise different between two claimed values of $\mathcal{F}_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$.
- κ is the number of outputs of Crkt (if $\kappa = 1$, it's the uber-assumption in $\mathbb{G}_1$)

Analysis of the Span-Uber Assumption

1. *Weaker* that a related uber-assumption in $\mathbb{G}_T$

> $(\mathcal{R}, \mathcal{S}, f'_I)$ uber-assumption holds in $\mathbb{G}_T$ for each $I \in [1..\kappa]$ $\Rightarrow$ $(\mathcal{R}, \mathcal{S}, \{f_i\}_{i=1}^{\kappa})$ span-uber-assumption holds in $\mathbb{G}_1$.

2. Holds under a hash-algebraic knowledge assumption

> If $f_i(X, Y) \notin \mathrm{span}(\mathcal{R})$, then the $(\mathcal{R}, \mathcal{S}, \{f_i\})$-span-uber-assumption holds under a PDL and a HAK.

3. Implied by *subgoup hiding* in **composite order groups** (Dèjá Q)

> Subgroup hiding in $\mathbb{G}_1$, $\mathbb{G}_2$ $\Rightarrow$ $(\mathcal{R}, \mathcal{S}, \{f_i\})$-span-uber-assumption in $\mathbb{G}_1$.

Applications of $\mathbf{FC}^{\mathsf{Crkt}}$

- *Succinct* aggregated inner product,
  $\mathcal{F}_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{n} \alpha_j \beta_{ij}$
- *Succinct* aggregated polynomial commitment,
  $\boldsymbol{\mathcal{F}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = (\alpha_i(\beta_i))_{i=1}^{\kappa} = (\sum_{j=0}^{n} \alpha_{ij} \beta_i^j)_{i=1}^{\kappa}$
- *Succinct* subvector commitment scheme,
  $\boldsymbol{\mathcal{F}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = (\alpha_{S_i})_{i=1}^{\kappa}$, $S_i \in [1..n]$
- *Succinct* evaluation-point commitment,
  $\boldsymbol{\mathcal{F}}(\alpha, \boldsymbol{\beta}) = (\beta_i(\alpha))_{i=1}^{\kappa} = (\sum_{j=0}^{n} \beta_{ij} \alpha^j)_{i=1}^{\kappa}$
- *Succinct* multivariate polynomial commitment,
  $\boldsymbol{\mathcal{F}}(\alpha, \boldsymbol{\beta}) = \alpha(\boldsymbol{\beta}) = \sum_j \alpha_j \prod_{k=1}^{c} \beta_k^{j_k}$, $j = (j_1, \ldots, j_c)$,
  $\sum j_c \leq d$

Aggregation: $\mathbf{FC}^{\mathsf{Crkt}}$ achieves easy aggregation

Conclusions

- New functional commitment scheme
  - for a large class of arithmetic circuits
  - based on SNARK techniques
  - succinct
  - evaluation-binding under a new falsifiable span-uber-assumption
- Extended justification for the assumption

Open Questions

- For which class of circuits CC, the compiled circuit Crkt^* has poly-size?
- Can we construct a succinct FC scheme for all poly-size arithmetic circuits?
- Is it possible to construct a succinct FC scheme based on a static security assumption in prime-order group?

Thank You!