HALO2

ZK-School session: Inseon Yu

Outline

1. Halo2 overview
2. Plonkish Arithmetization
3. IPA based Halo2 (ZCash)
4. Kate pairing based Halo2

1. Halo2 overview

- PLONKish Arithmetization
- Modified Inner Product Argument
- Proof aggregation

2. PLONKish Arithmetization

Vanilla PLONK

$\rightarrow (q_L) \cdot a + (q_R) \cdot b + (q_O) \cdot c + (q_M) \cdot ab + (q_C) = 0$

- Add: $(1) \cdot a + (1) \cdot b + (-1) \cdot c + (0) \cdot ab + (0) = a + b - c = 0$
- Mul: $(0) \cdot a + (0) \cdot b + (-1) \cdot c + (1) \cdot ab + (0) = -c + a \cdot b = 0$
- Copy constraint via Permutation check

2. PLONKish Arithmetization

Permutation check

a_0	a_1	a_2	a_3	a_4
$input_0$	$input_1$	$input_2$		$output$
va_1	vb_1	vc_1		vd_1
va_2	vb_2	vc_2		vd_2
va_3	vb_3	vc_3		vd_3
va_4	$\boxed{vb_4}$	vc_4	……	vd_4
va_5	vb_5	vc_5		vd_5
va_6	vb_6	$\boxed{vc_6}$		vd_6
va_7	vb_7	vc_7		vd_7
$\boxed{va_6}$	$\boxed{vb_6}$	vc_6		vd_6
va_7	vb_7	vc_7		vd_7

$$\rightarrow vb_4 = vc_6 = vb_6 = va_6$$

2. PLONKish Arithmetization

Custom gate

$$q_{add} \cdot (a_0 + a_1 - a_2) + y \cdot q_{mul} \cdot (a_0 \cdot a_1 - a_2) + y^2 \cdot q_{bool} \cdot (a_0 \cdot a_0 - a_0) = 0$$

- Defining bespoke gates (EC point additions, Hashes, …)
- Reduces # of gates
- Verifier challenge to keep gates linearly independent

2. PLONKish Arithmetization

a_0	a_1	a_2	a_3	a_4
$input_0$	$input_1$	$input_2$		$output$
va_1	vb_1	vc_1		vd_1
va_2	vb_2	vc_2		vd_2
va_3	vb_3	vc_3		vd_3
va_4	vb_4	vc_4	……	vd_4
va_5	vb_5	vc_5		vd_5
va_6	vb_6	vc_6		vd_6
va_7	vb_7	vc_7		vd_7
va_6	vb_6	vc_6		vd_6
va_7	vb_7	vc_7		vd_7

$$\rightarrow va_3 \cdot vb_3 \cdot vc_3 + vc_2 - vb_4 = 0$$

a_0	a_1	a_2	a_3	a_4
$input_0$	$input_1$	$input_2$		$output$
va_1	vb_1	vc_1		vd_1
va_2	vb_2	vc_2		vd_2
va_3	vb_3	vc_3		vd_3
va_4	vb_4	vc_4	……	vd_4
va_5	vb_5	vc_5		vd_5
va_6	vb_6	vc_6		vd_6
va_7	vb_7	vc_7		vd_7
va_6	vb_6	vc_6		vd_6
va_7	vb_7	vc_7		vd_7

$$\rightarrow vb_1 \cdot vc_1 + vc_2 - vc_3 = 0$$

High degree & More customized

2. PLONKish Arithmetization

a_0	a_1	a_2	a_3	a_4	T_0	T_1	T_2	T_3	T_4
$input_0$	$input_1$	$input_2$		$output$	0000	0000	0000		
va_1	vb_1	vc_1		vd_1	0000	0001	0001		
va_2	vb_2	vc_2		vd_2	0000	0010	0010		
va_3	vb_3	vc_3		vd_3	0000	0011	0011		
va_4	vb_4	vc_4	……	vd_4	0000	0100	0100		
va_5	vb_5	vc_5		vd_5	0000	0101	0101		
va_6	vb_6	vc_6		vd_6	……	……	……		
$\boxed{va_7}$	$\boxed{vb_7}$	$\boxed{vc_7}$	Lookup		1111	1101	0010		
va_6	vb_6	vc_6		vd_6	1111	1110	0001		
va_7	vb_7	vc_7		vd_7	1111	1111	0000		

witness (columns a_0–a_4) | Table 1 (columns T_0–T_2) | Table 2 (columns T_3–T_4)

$$\rightarrow va_7 \oplus vb_7 = vc_7$$

Lookup argument

- Calculate in advance a lookup table composed of valid input and output
- Prover proves that witness exist in the table

2. PLONKish Arithmetization

Plookup VS Halo2 Lookup

- Both protocols can prove that $f \subset t$

- Plookup: needs f and t to build a new sequence s
  Then, proves $s \subset t$ by compairing the non-zero distance sets of elements in s are equal
  $f \subset s \subset t \rightarrow f \subset t$

- Halo2-lookup: proves $f \subset t$ directly (more concise)

- Both require sorting and completing the set
  Plookup: $|t| = |f| + 1$
  Halo2-lookup: $|t| = |f| = 2^k$

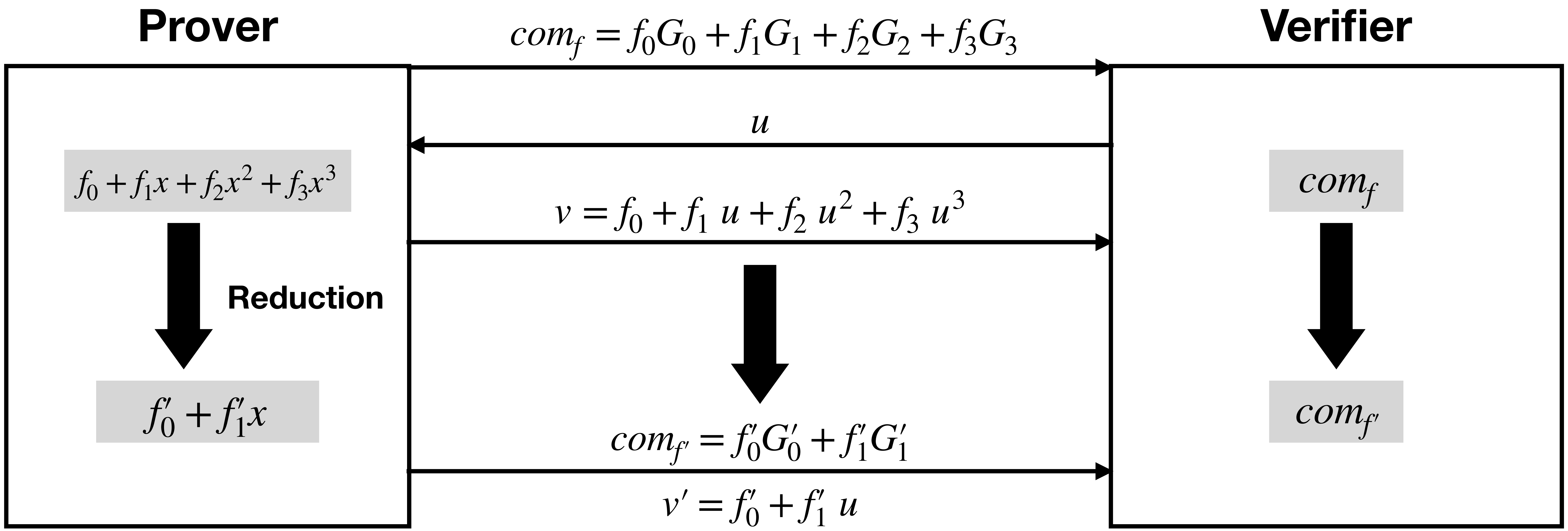
3. IPA based Halo2

Inner Product Argument

- Transparent setup: sample random $gp = (G_0, G_1, \ldots, G_d)$ in $\mathbb{G}$
- Commit: $f(x) = f_0 + f_1 x + f_2 x^2 + \ldots + f_d x^d$

$$com_f = f_0 G_0 + f_1 G_1 + \ldots + f_d G_d$$

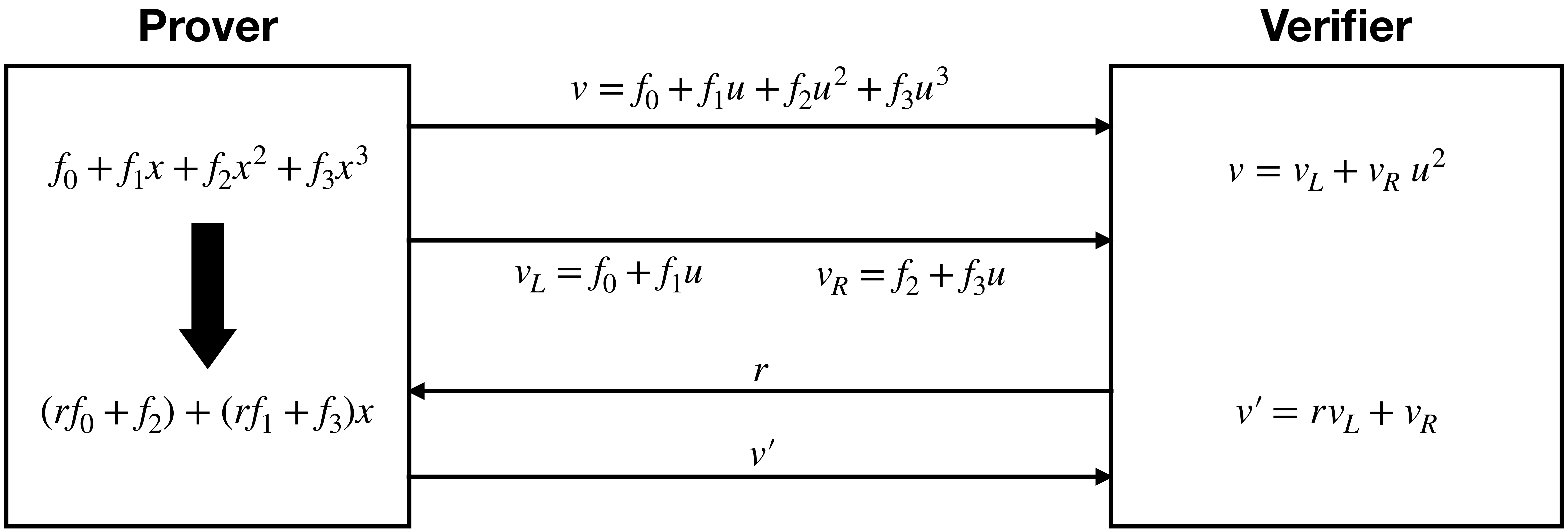
3. IPA based Halo2



3. IPA based Halo2

$$gp = (G_0, G_1, G_2, G_3)$$

$$com_f = f_0 G_0 + f_1 G_1 + f_2 G_2 + f_3 G_3$$

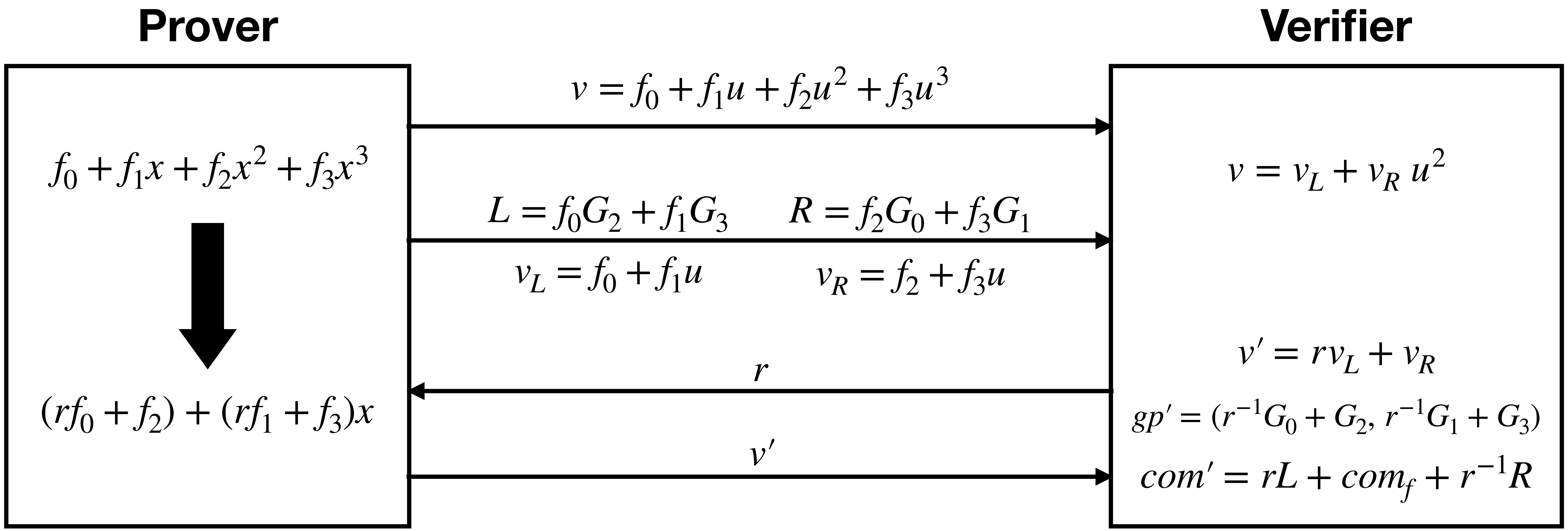


Combine polynomials via random linear combination

3. IPA based Halo2

$$gp = (G_0, G_1, G_2, G_3)$$

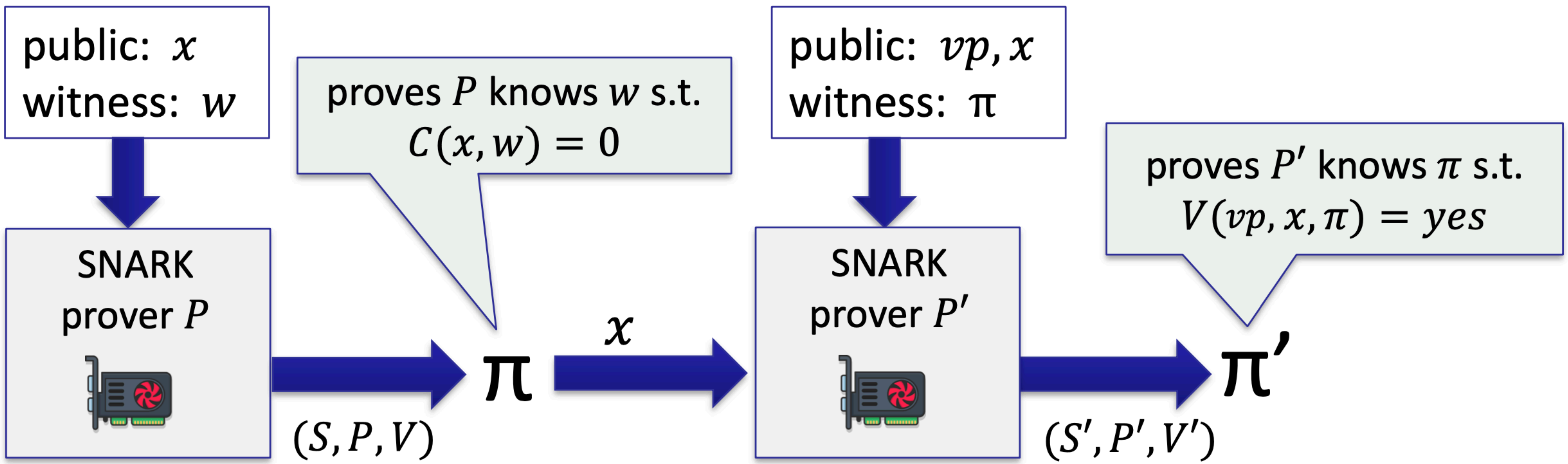
$$com_f = f_0G_0 + f_1G_1 + f_2G_2 + f_3G_3$$



Compute new commitment via L, R

3. IPA based Halo2

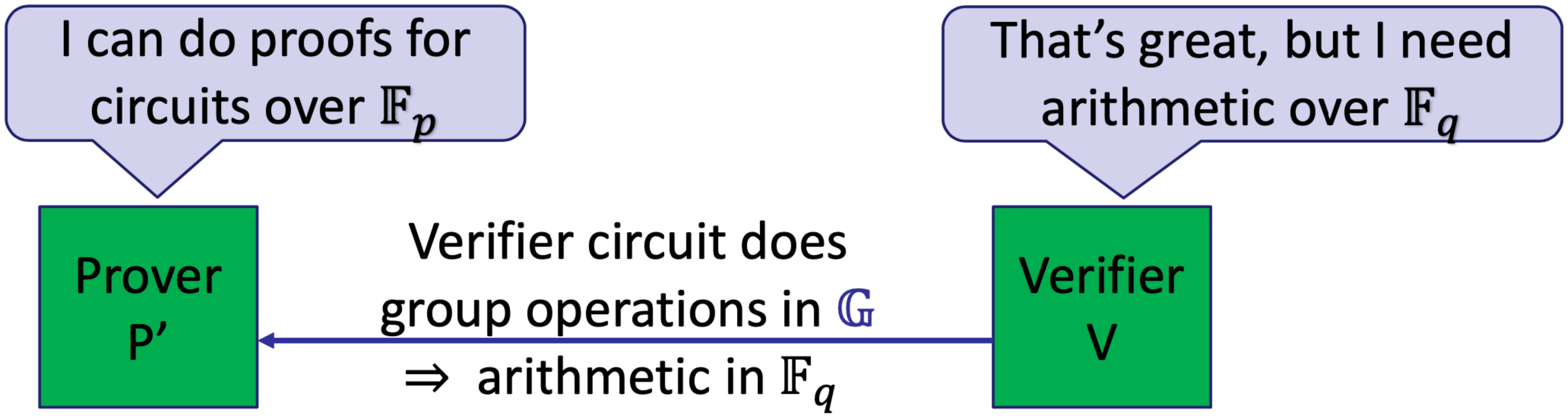
Proof recursion



3. IPA based Halo2

- Let $\mathbb{G}$ be an group of order p, defined over $\mathbb{F}_q$

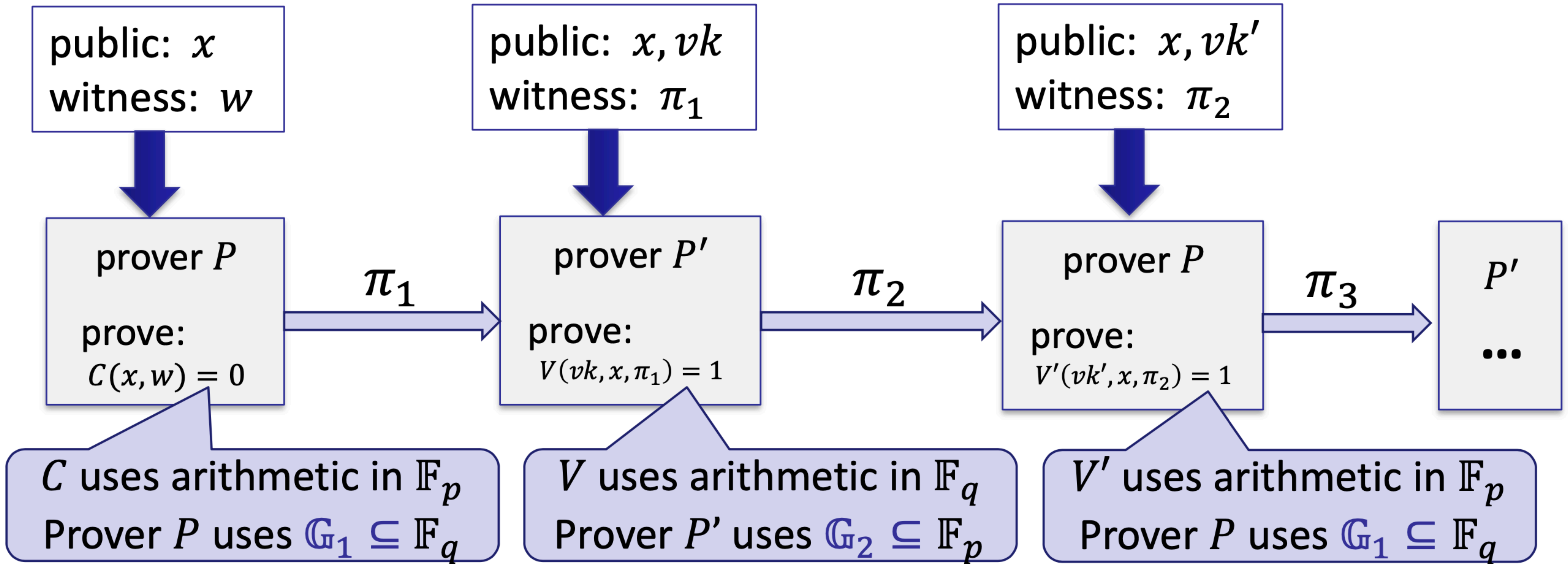
  The prover supports circuits over $\mathbb{F}_p$, but verifier needs $\mathbb{F}_q$ for group ops.



3. IPA based Halo2

A cycles of groups [BCTV'14]

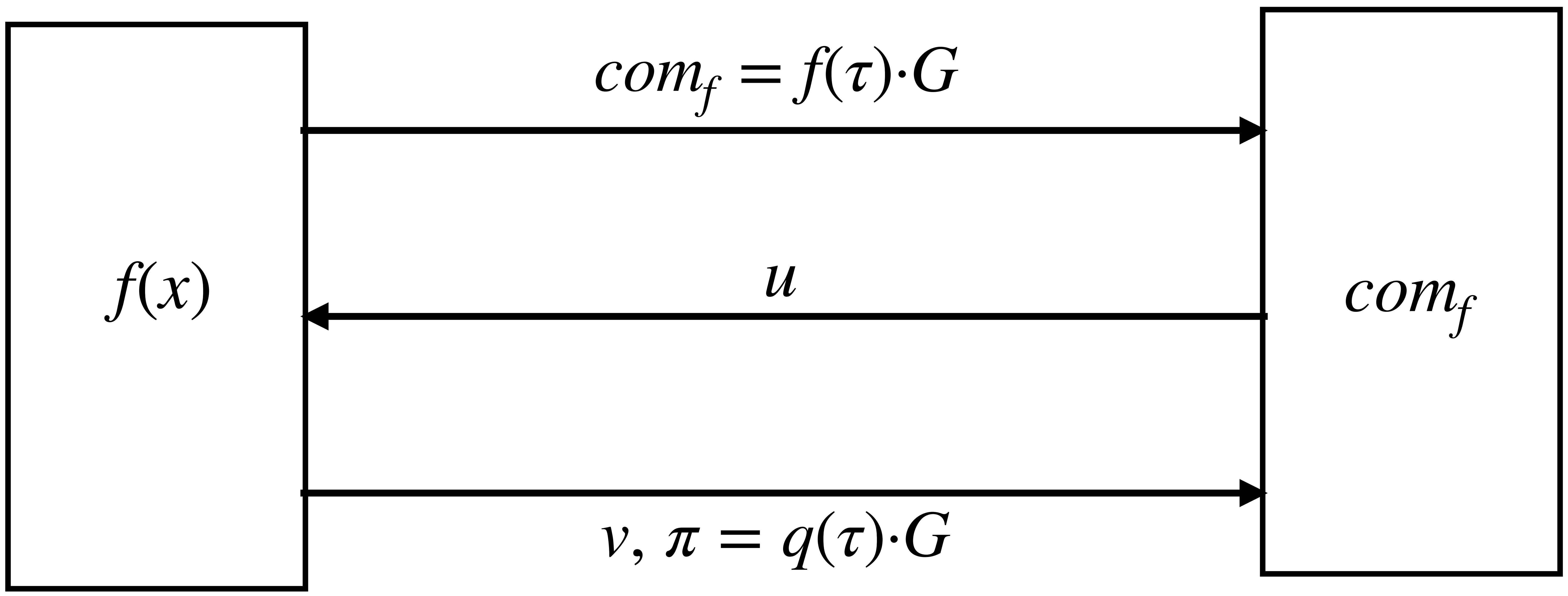
- $\mathbb{G}_1$ has order p, defined over $\mathbb{F}_q$
- $\mathbb{G}_2$ has order q, defined over $\mathbb{F}_p$



4. KZG based Halo2

$$gp = (G, \tau G, \tau^2 G, \ldots, \tau^d G)$$

Prover **Verifier**



$$f(x) - f(u) = (x - u)\, q(x)$$

$$e(G, G)^{f(\tau) - f(u)} = e(G, G)^{(\tau - u)\, q(\tau)}$$

4. KZG based Halo2

	Halo2 (ZCash)	Halo2-KZG
Curves	Pasta curves	BN256
Proving scheme	IPA	KZG
Zero-knowledge	Uncompromisable	Optianal

Thank you!!