Projective Arithmetic Functional Encryption
and
Indistinguishability Obfuscation (iO) from Degree-5 Multilinear maps

Prabhanjan Ananth

Amit Sahai

UCLA

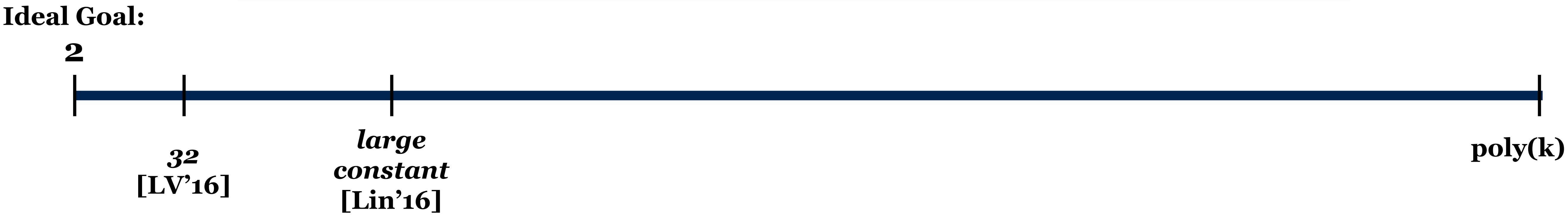
Center for Encrypted Functionalities

Constructions of iO

All current constructions of iO are based on multilinear maps
[GGHRSW13, BR14, BGKPS14, PST14, AGIS14, …, AB15, Zim15, GLSW15, GMMSZ16, Lin16a, LV16, Lin16b, …]

- Multilinear maps: *generalization of bilinear maps*
- Degree-D multilinear maps: *can compute degree-D polynomials in the exponents of the group*

What is the minimum degree of multilinear maps required to construct iO?



Ideal Goal:

2

32
[LV'16]

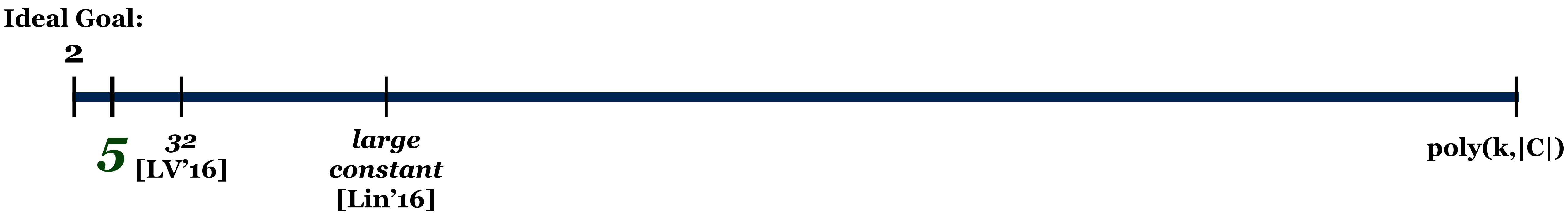
large constant
[Lin'16]

poly(k)

- Original works [GGHRSW'13, BGKPS'14, …]: degree = polynomial in security parameter
- Lin'16: degree = constant
- LV'16: degree = 32

This Work

iO from degree-**5** multinear maps



Ideal Goal:

2

5

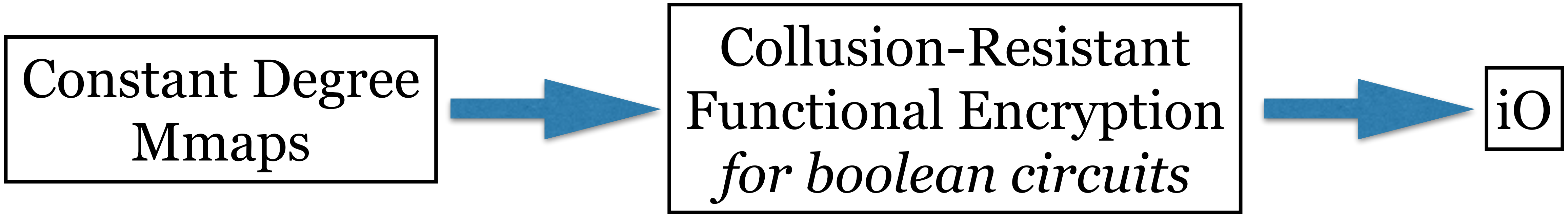
32
[LV’16]

large constant
[Lin’16]

poly(k,|C|)

A new template to construct iO from constant degree multilinear maps

Prior Works [Lin'16,LV'16]

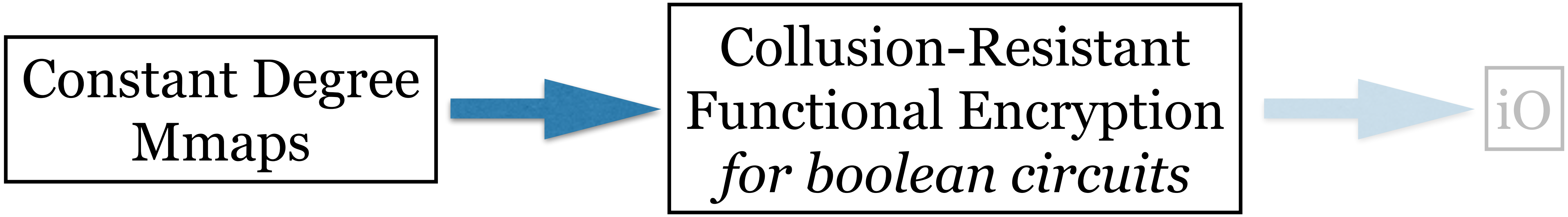


Constant Degree Mmaps

Collusion-Resistant Functional Encryption *for boolean circuits*

iO

Prior Works [Lin'16,LV'16]



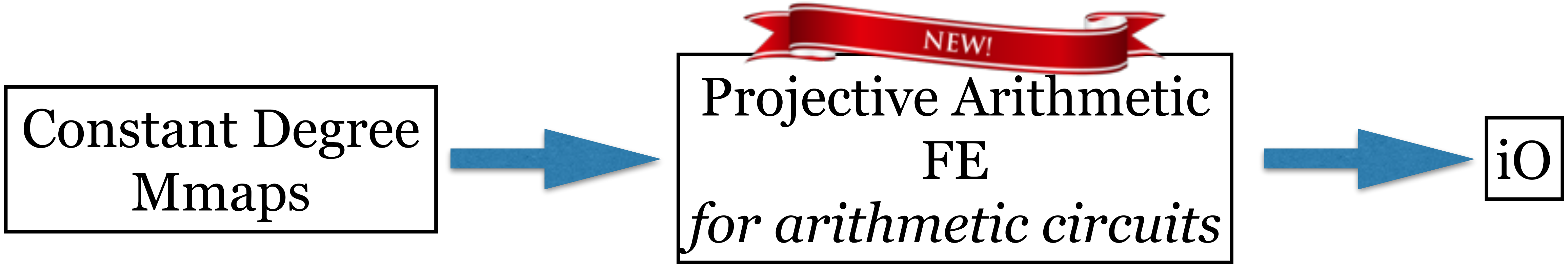
Constant Degree Mmaps

Collusion-Resistant Functional Encryption *for boolean circuits*

iO

- MMap computations performed over large fields
- To construct FE from mmaps: need to "arithmetize" the boolean circuits

Our Template



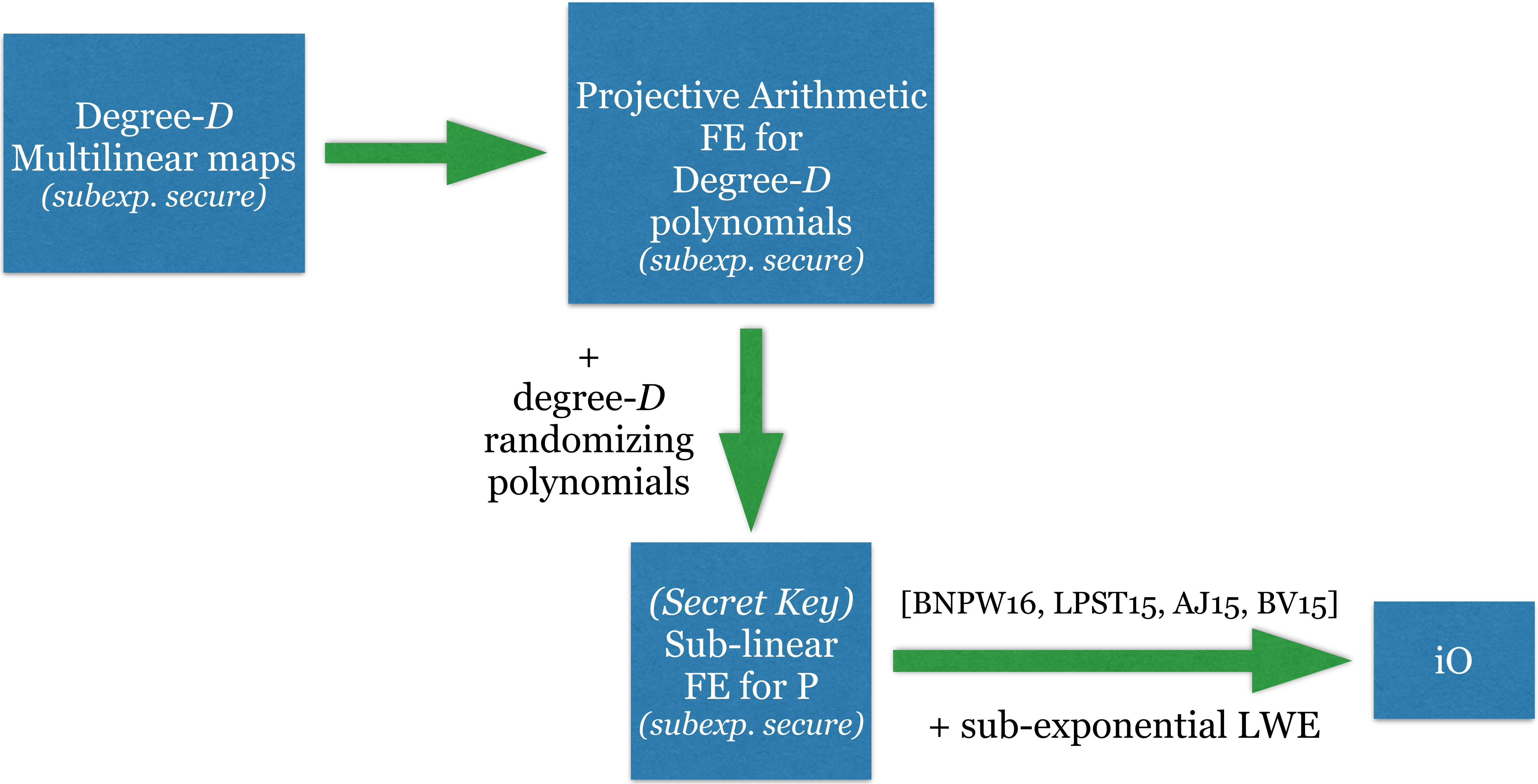
Constant Degree Mmaps

Projective Arithmetic FE *for arithmetic circuits*

iO

- PAFE is a version of functional encryption for arithmetic circuits

Our Template (in detail)



Degree-D Multilinear maps *(subexp. secure)*

Projective Arithmetic FE for Degree-D polynomials *(subexp. secure)*

+ degree-D randomizing polynomials

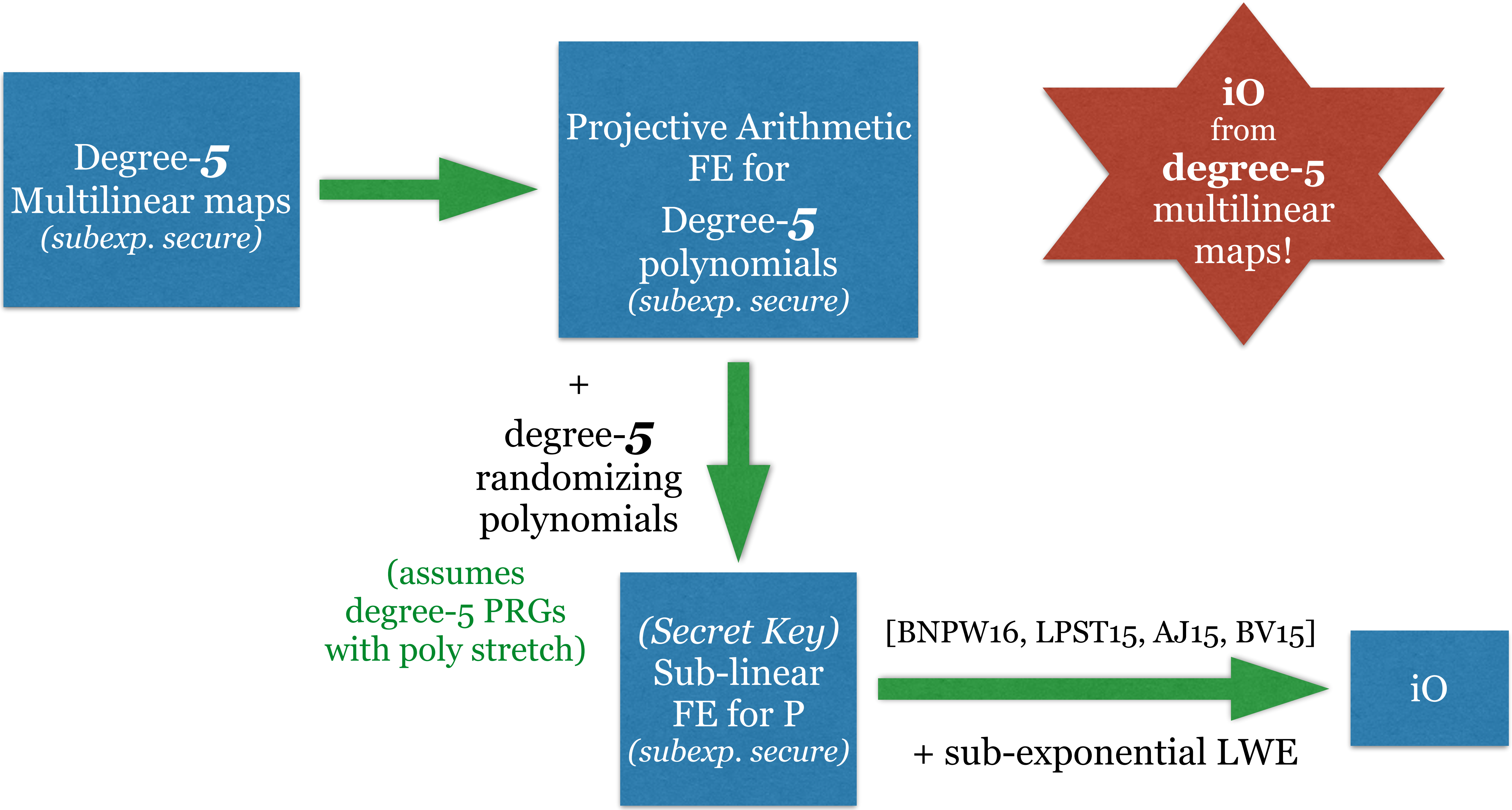
(Secret Key) Sub-linear FE for P *(subexp. secure)*

[BNPW16, LPST15, AJ15, BV15]

+ sub-exponential LWE

iO

Instantiation



Degree-***5***
Multilinear maps
(subexp. secure)

Projective Arithmetic FE for
Degree-***5***
polynomials
(subexp. secure)

iO
from
degree-5
multilinear
maps!

+
degree-***5***
randomizing
polynomials

(assumes
degree-5 PRGs
with poly stretch)

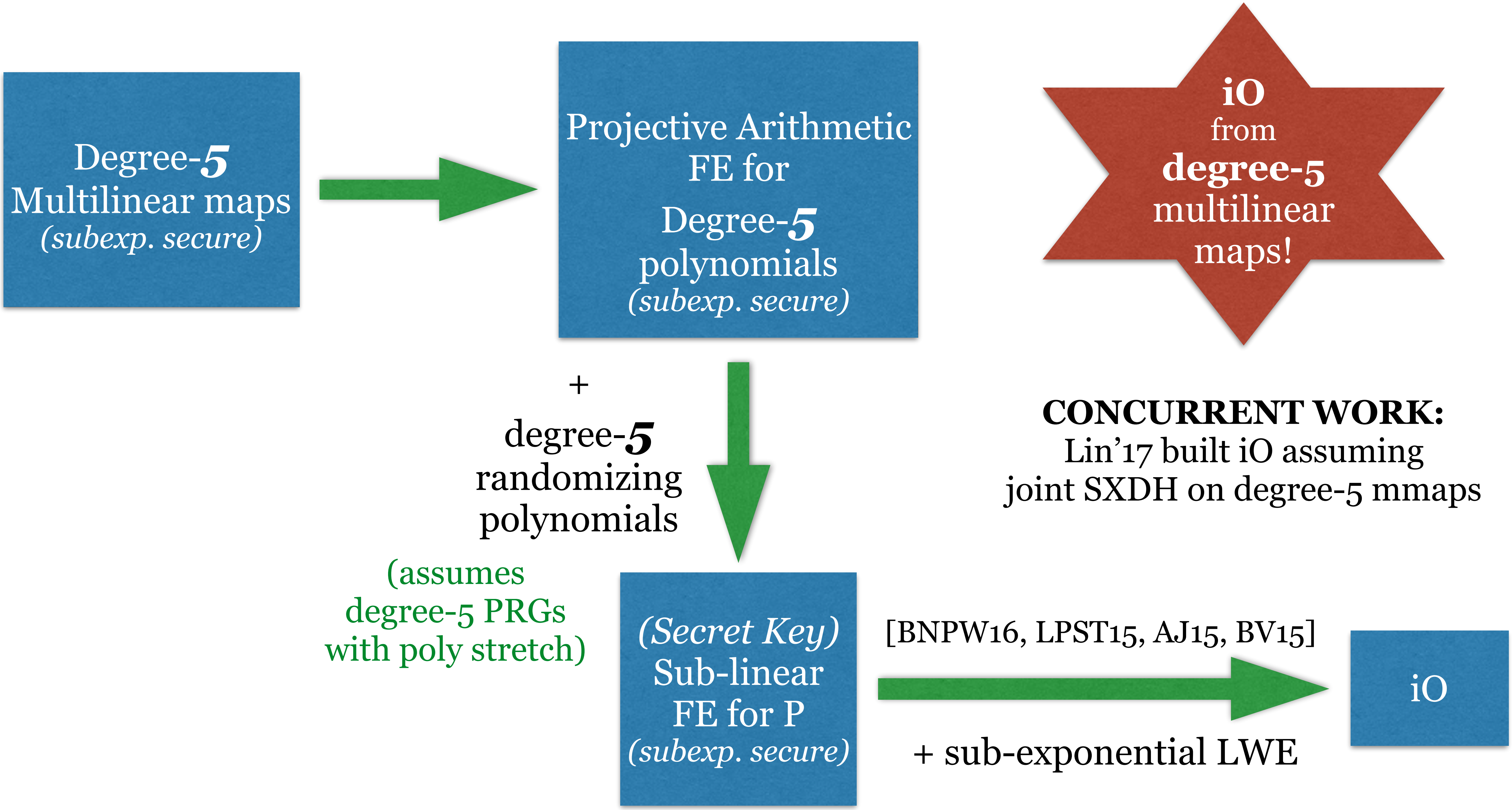
(Secret Key)
Sub-linear
FE for P
(subexp. secure)

[BNPW16, LPST15, AJ15, BV15]

+ sub-exponential LWE

iO

Instantiation



Degree-***5*** Multilinear maps *(subexp. secure)*

Projective Arithmetic FE for Degree-***5*** polynomials *(subexp. secure)*

+ degree-***5*** randomizing polynomials

(assumes degree-5 PRGs with poly stretch)

(Secret Key) Sub-linear FE for P *(subexp. secure)*

[BNPW16, LPST15, AJ15, BV15]

+ sub-exponential LWE

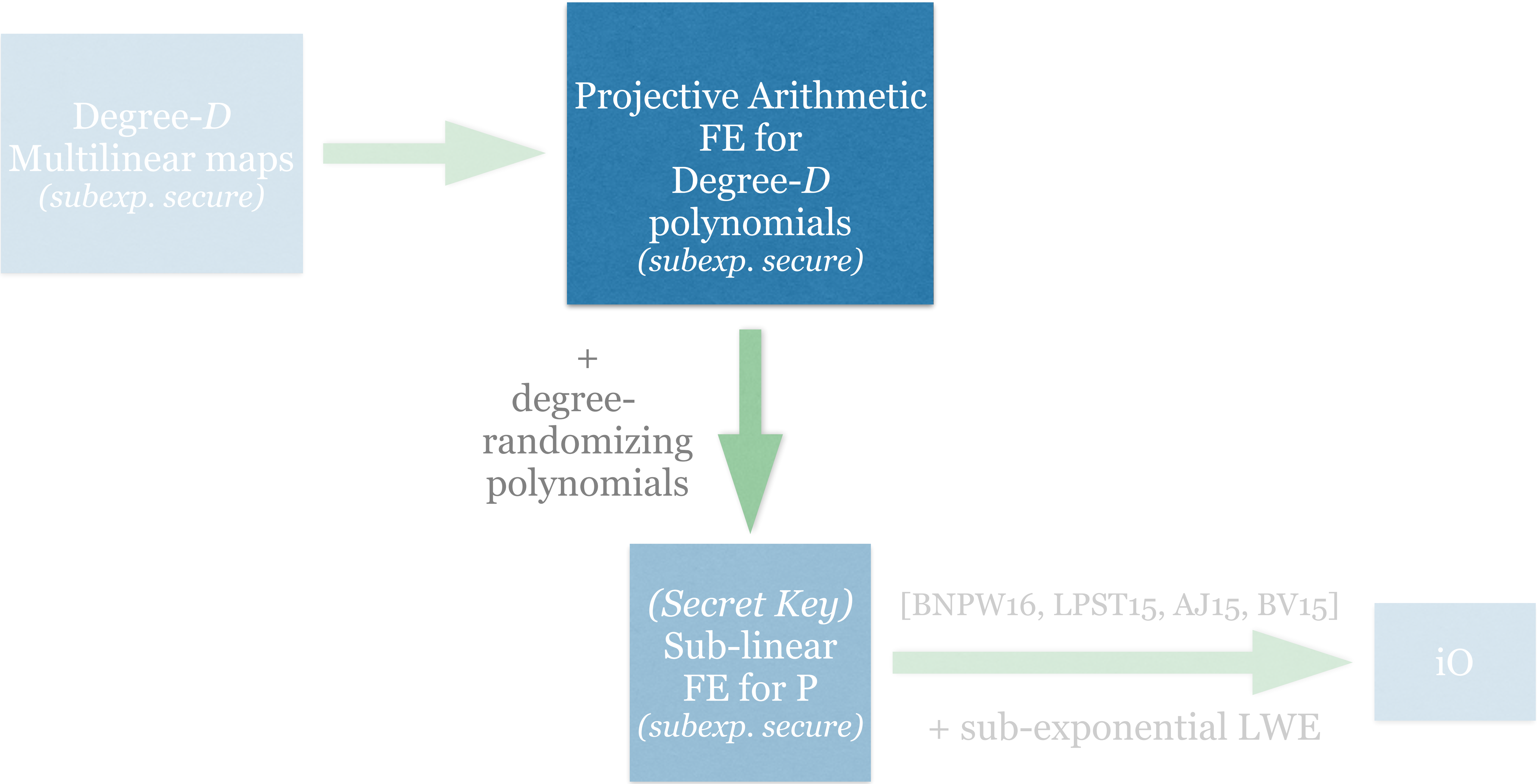
iO

iO from **degree-5** multilinear maps!

CONCURRENT WORK:
Lin'17 built iO assuming joint SXDH on degree-5 mmaps

Technical Overview

Our Template



Degree-D Multilinear maps *(subexp. secure)*

Projective Arithmetic FE for Degree-D polynomials *(subexp. secure)*

+ degree-randomizing polynomials

(Secret Key) Sub-linear FE for P *(subexp. secure)*

[BNPW16, LPST15, AJ15, BV15]

+ sub-exponential LWE

iO

Projective Arithmetic FE (PAFE)

- **FIRST ATTEMPT:**

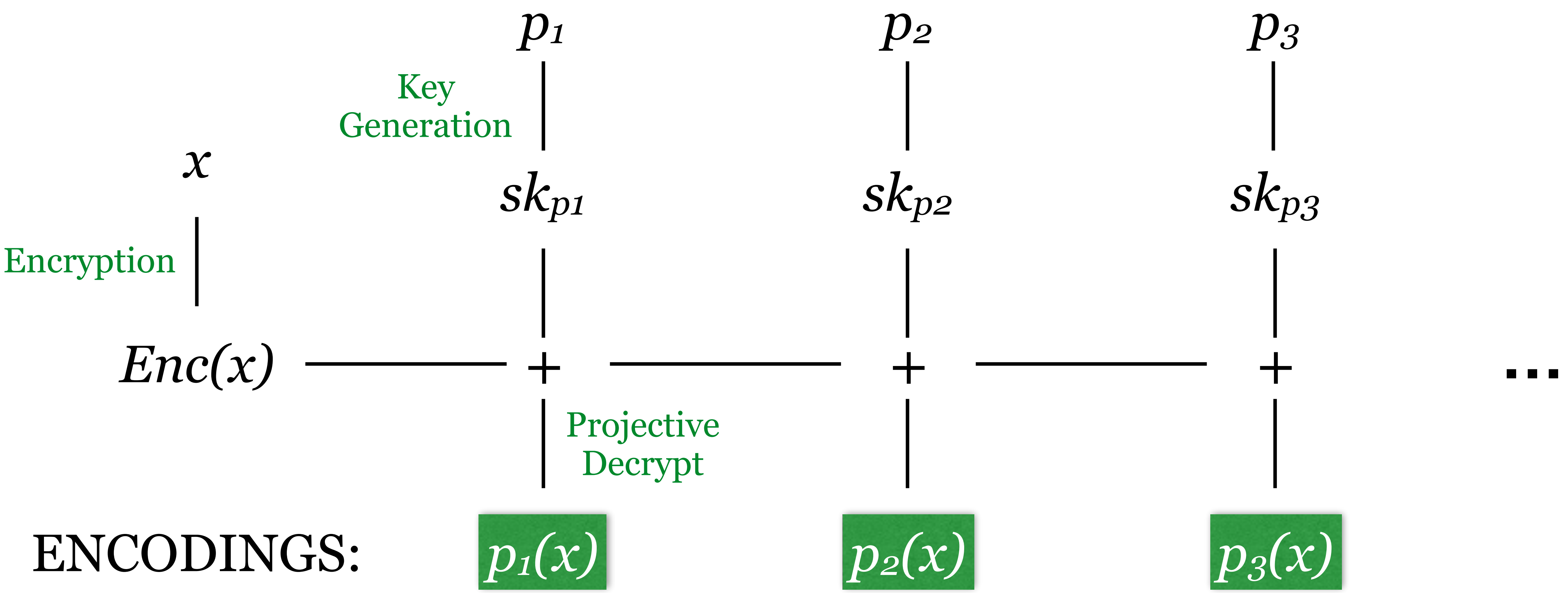
Same syntax as FE for boolean circuits except that functional keys issued for polynomials (over large fields)

$$\boxed{\text{Encryption of } x + \text{Key of polynomial } p := p(x)}$$

ISSUE: Current techniques are a limiting factor!

- If p(x) is large, we don’t know how to construct this notion
- **Reason:** Decryption in existing FE schemes yields *Encoding(p(x))* and can decode only if p(x) is small

Projective Arithmetic FE (PAFE)



p_1 p_2 p_3

Key Generation

x

sk_{p1} sk_{p2} sk_{p3}

Encryption

$Enc(x)$ + + + ...

Projective Decrypt

ENCODINGS: $p_1(x)$ $p_2(x)$ $p_3(x)$

Can recover *linear function of* $(p_1(x), p_2(x), p_3(x), \ldots)$

if output of linear function is "small"

Efficiency

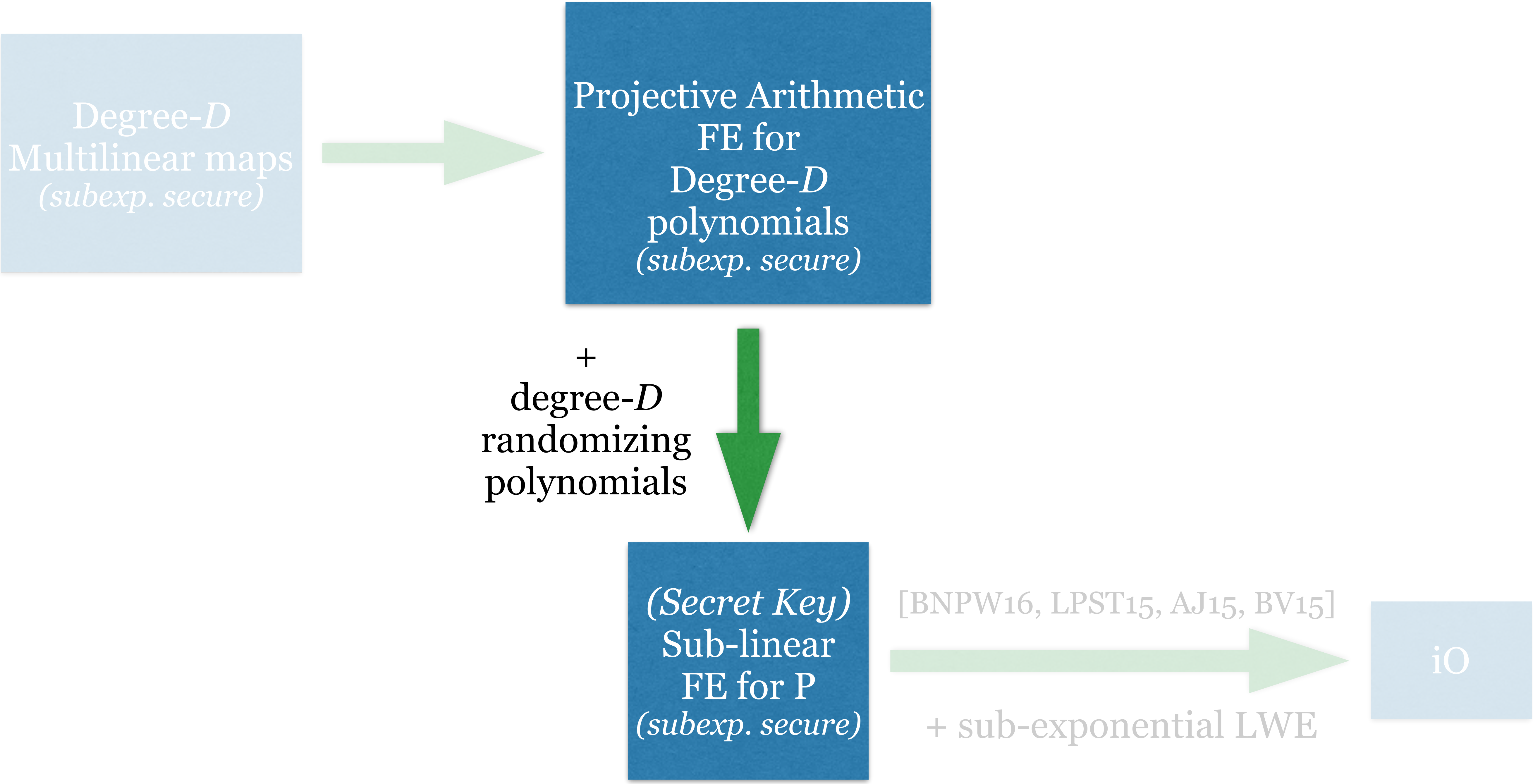
- **Linear Overhead:**
  - Size of encryption of y := |y| poly(k,D)

    D - *degree of polynomials*

Security

- **Semi-functional security:**
  - Inspired by ABE literature [Wat09,LOS+10,...,GGHZ14]
  - Captures a weak form of function hiding

Our Template



Degree-D Multilinear maps *(subexp. secure)*

Projective Arithmetic FE for Degree-D polynomials *(subexp. secure)*

+ degree-D randomizing polynomials

(Secret Key) Sub-linear FE for P *(subexp. secure)*

[BNPW16, LPST15, AJ15, BV15]

+ sub-exponential LWE

iO

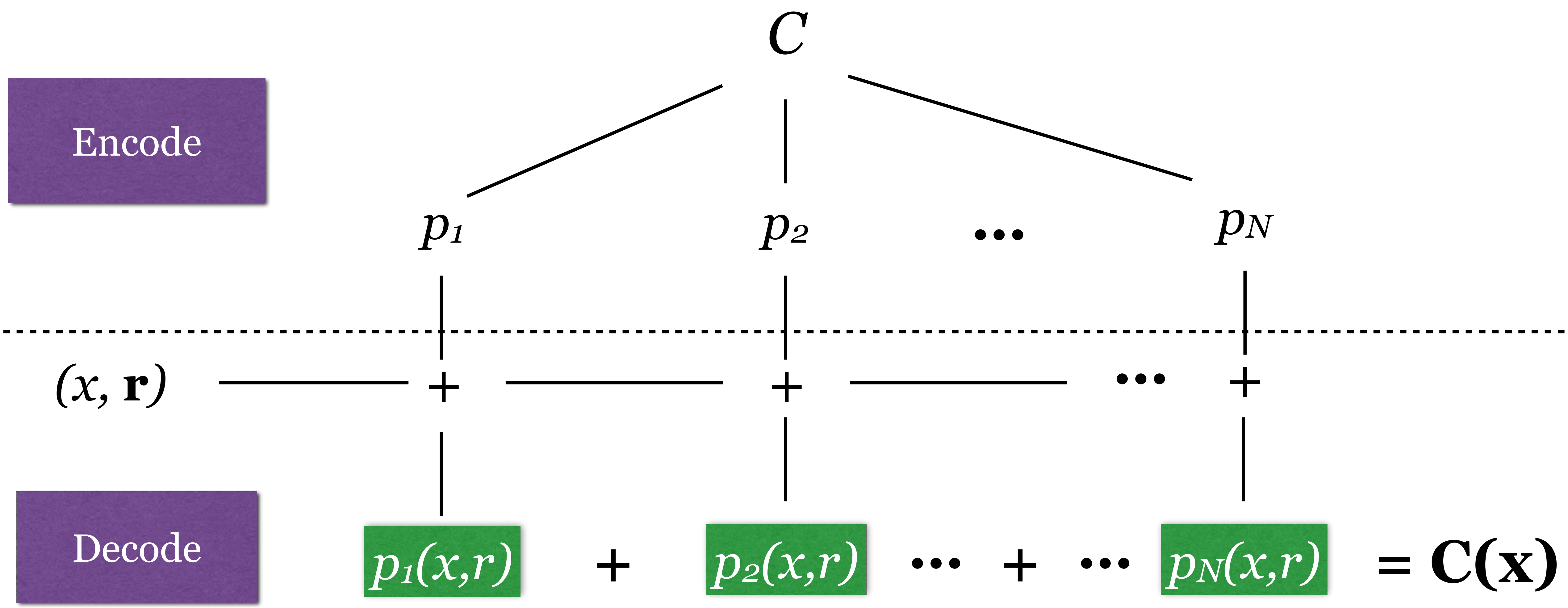
Sub-linear (Secret Key) FE

for Boolean circuits

SUB-LINEARITY

$|Enc(x)| = |C|^{e}\, poly(k,|x|) \ ; \ e < 1$

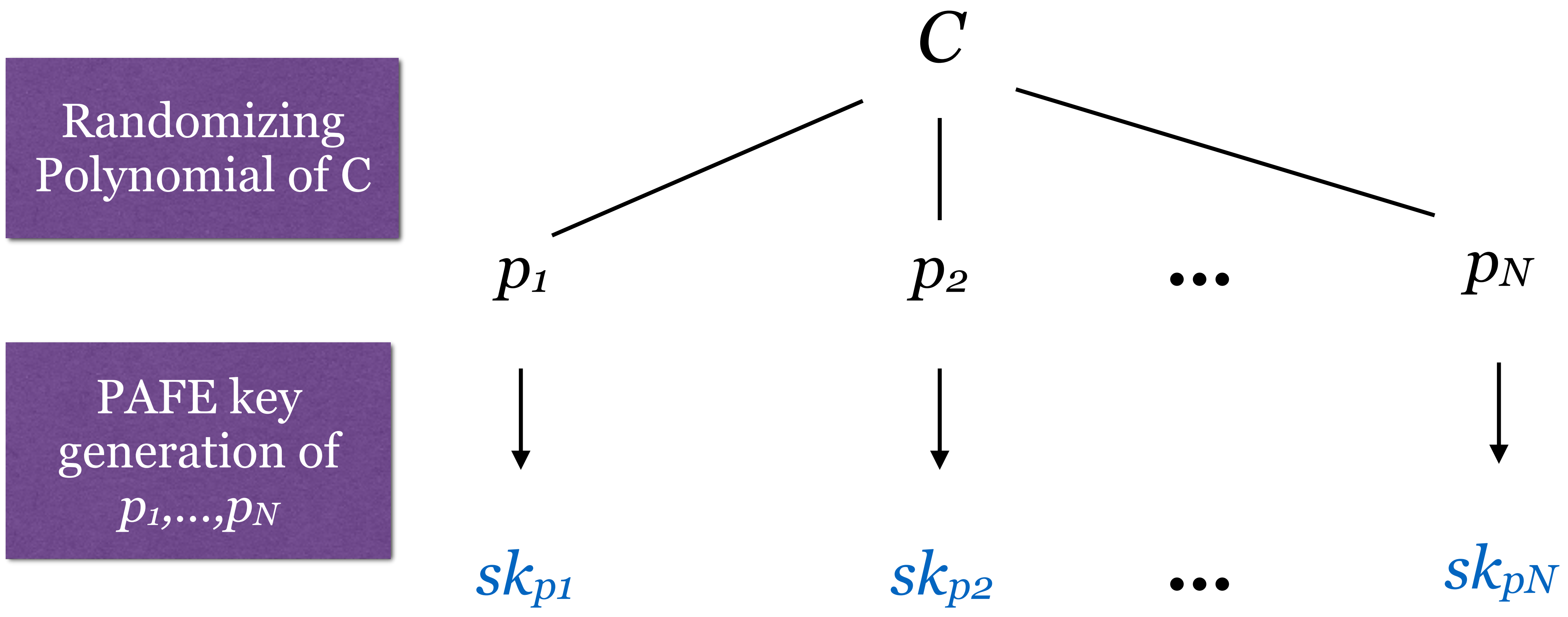
Randomizing Polynomials



If all $\mathbf{p_i}$ **is of degree D** then
it is a degree-D randomizing polynomial

Construction of Sub-linear FE

Key Generation of C:



Randomizing Polynomial of C

PAFE key generation of $p_1,\ldots,p_N$

C

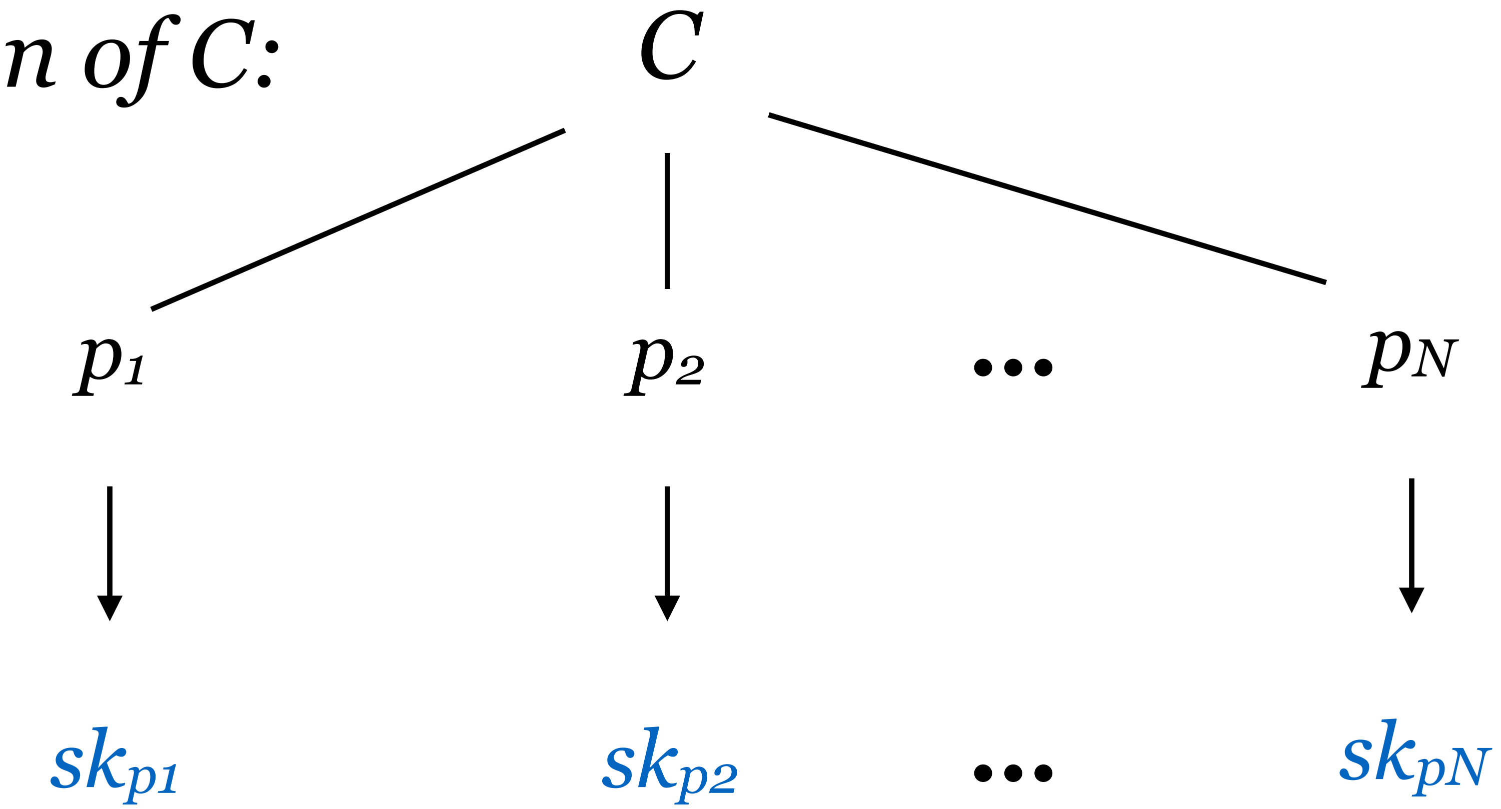
p_1 p_2 … p_N

sk_{p1} sk_{p2} … sk_{pN}

Functional key of C = $(sk_{p1}, \ldots, sk_{pN})$

Construction of Sub-linear FE

Key Generation of C:



C

p_1 p_2 … p_N

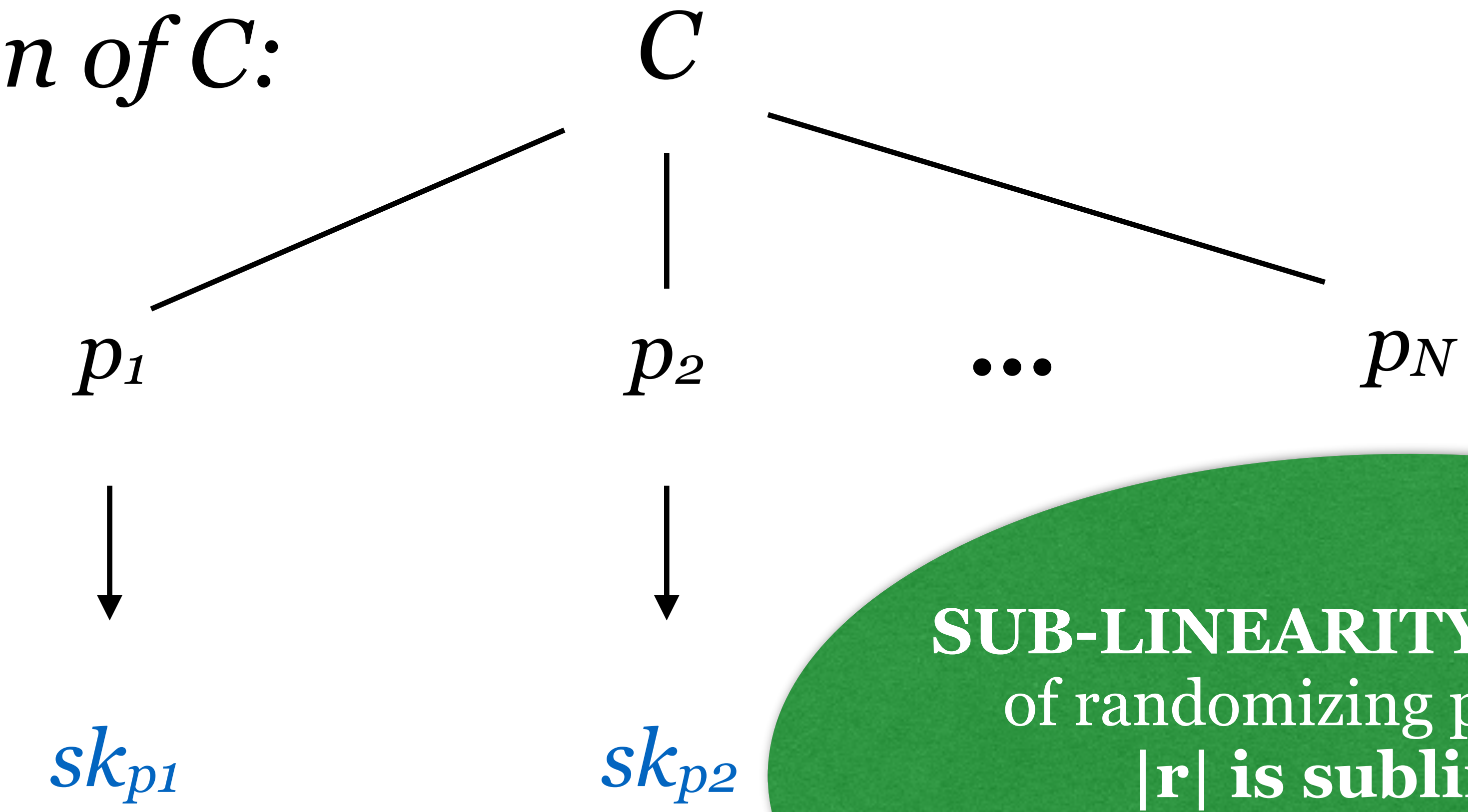
sk_{p1} sk_{p2} … sk_{pN}

Encryption of x:

$$x \xrightarrow[\mathbf{r}]{} (x, \mathbf{r})$$

Construction of Sub-linear FE

Key Generation of C:



C

p_1 p_2 ••• p_N

sk_{p1} sk_{p2}

SUB-LINEARITY PROPERTY
of randomizing polynomials:
|r| is sublinear in
the length of circuit description

Encryption of x:

$$x \xrightarrow[\mathbf{r}]{} (x, \mathbf{r})$$

Construction of Sub-linear FE

Decryption (INTUITION):

- Execute PAFE **ProjectiveDecrypt**
- Execute **Recover** to obtain encoding of (C,x)
- Execute the decoding procedure

Instantiation of degree-5 randomizing polynomials

(with sub-linearity property)

WARMUP:

- Consider degree-3 randomizing polynomials [AIK'06] *(without sub-linearity property)*

- Compress randomness using PRGs!
  - Use degree 5 PRGs
    (maps seed of length n to $n^{1.49}$)

TOTAL DEGREE = 5 * 3 = 15

Instantiation of degree-5 randomizing polynomials

(with sub-linearity property)

WARMUP:

- Consider degre
  [AIK'06] *(without su*

- Compress randomness using PRGs!
  - Use degree 5 PRGs
    (maps seed of length n to $n^{1.49}$)

Goldreich PRG candidate:
Analysed by O'Donnell and Witmer'14

TOTAL DEGREE = 5 * 3 = 15

Instantiation of degree-5 randomizing polynomials

(with sub-linearity property)

WARMUP:

- Co ials

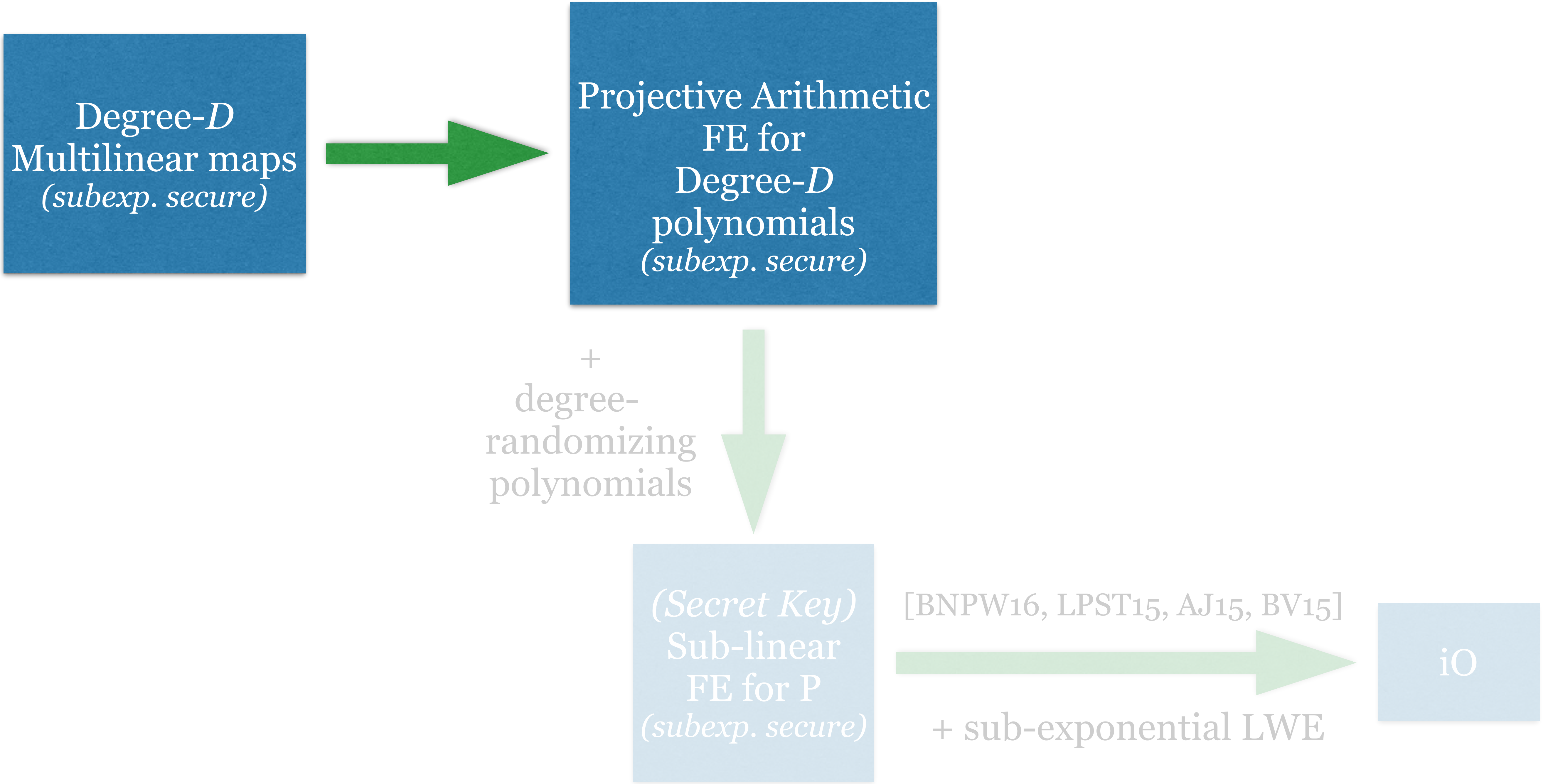
Degree-5 randomizing polynomials:

We use pre-processing trick!

(pre-compute some partial terms ahead of time)

TOTAL DEGREE = 5 * 3 = 15

Our Template



Degree-D Multilinear maps *(subexp. secure)*

Projective Arithmetic FE for Degree-D polynomials *(subexp. secure)*

+ degree-randomizing polynomials

(Secret Key) Sub-linear FE for P *(subexp. secure)*

[BNPW16, LPST15, AJ15, BV15]

+ sub-exponential LWE

iO

Slotted Encodings

An abstraction of composite order multi-linear maps

Encoding of (a,b,c) w.r.t color: $\boxed{a \mid b \mid c}$

Addition w.r.t same color: $\boxed{a_1 \mid b_1 \mid c_1} + \boxed{a_2 \mid b_2 \mid c_2} = \boxed{a_1+a_2 \mid b_1+b_2 \mid c_1+c_2}$

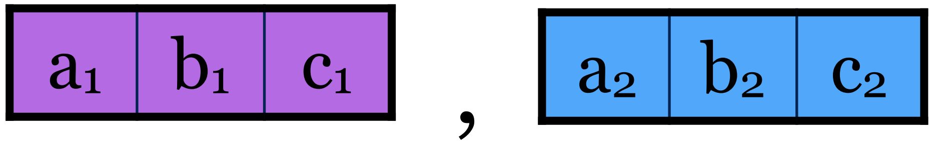
Multiplication w.r.t *"compatible"* colors: $\boxed{a_1 \mid b_1 \mid c_1} * \boxed{a_2 \mid b_2 \mid c_2} = \boxed{a_1*a_2 \mid b_1*b_2 \mid c_1*c_2}$

Zero Test w.r.t color red: $\boxed{a \mid b \mid c}$ is **ZERO** if and only if $\boldsymbol{a+b+c=0}$

Degree-D Slotted Encodings
from
Degree-D Prime order mmap

Degree-D slotted encodings: if it allows for evaluating polynomials of degree at most D

SIMPLE CASE: Degree=2



a_1	b_1	c_1

,

a_2	b_2	c_2

Degree-D Slotted Encodings
from
Degree-D Prime order mmap

Degree-D slotted encodings: if it allows for evaluating polynomials of degree at most D

SIMPLE CASE: Degree=2

Pick vectors $\boldsymbol{u_1}$, $\boldsymbol{u_2}$, $\boldsymbol{u_3}$, $\boldsymbol{v_1}$, $\boldsymbol{v_2}$, $\boldsymbol{v_3}$

$$\boxed{a_1\boldsymbol{u_1} + b_1\boldsymbol{u_2} + c_1\boldsymbol{u_3}} \;,\; \boxed{a_2\boldsymbol{v_1} + b_2\boldsymbol{v_2} + c_2\boldsymbol{v_3}}$$

such that $\langle \boldsymbol{u_i}, \boldsymbol{v_j} \rangle = 1$, *if i=j*

$= 0$, otherwise

Degree-D Slotted Encodings from Degree-D Prime order mmap

Degree-D slotted encodings: if it allows for evaluating polynomials of degree at most D

SIMPLE CASE: Degree=2

Pick vectors $\boldsymbol{u_1}, \boldsymbol{u_2}, \boldsymbol{u_3}, \boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}$

Dual vector spaces! [OT08,OT09,BJK15]

$a_1\boldsymbol{u_1} + b_1\boldsymbol{u_2} + c_1\boldsymbol{u_3}$, $a_2\boldsymbol{v_1} + b_2\boldsymbol{v_2} + c_2\boldsymbol{v_3}$

such that $<\boldsymbol{u_i},\boldsymbol{v_j}> = 1$, *if i=j*

$= 0$, otherwise

Degree-D Slotted Encodings
from
Degree-D Prime order mmap

Degree-D slotted encodings: if it allows for evaluating polynomials of degree at most D

SIMPLE CASE: Degree=2

$$\left\langle \boxed{a_1\boldsymbol{u_1} + b_1\boldsymbol{u_2} + c_1\boldsymbol{u_3}} \;,\; \boxed{a_2\boldsymbol{v_1} + b_2\boldsymbol{v_2} + c_2\boldsymbol{v_3}} \right\rangle$$

$$= \boxed{a_1a_2 + b_1b_2 + c_1c_2}$$

Degree-D Slotted Encodings
from
Degree-D Prime order mmap

Higher (constant) degrees: tensoring of dual vector spaces

Example: Degree=3

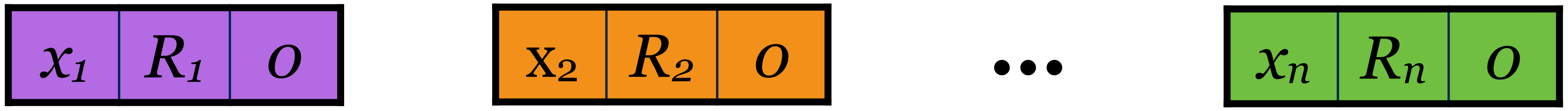
$$< \boxed{a_1\boldsymbol{w_1u_1} + b_1\boldsymbol{w_2u_2} + c_1\boldsymbol{w_3u_3}} \; , \; \boxed{a_2\boldsymbol{v_1} + b_2\boldsymbol{v_2} + c_2\boldsymbol{v_3}} >$$

$$= \boxed{a_1a_2\boldsymbol{w_1} + b_1b_2\boldsymbol{w_2} + c_1c_2\boldsymbol{w_3}} \; , \quad \cdots$$

Construction of PAFE (Intuition)

Setup: Pick $R_1,\ldots,R_n$

Encryption of x:



x_1 | R_1 | 0

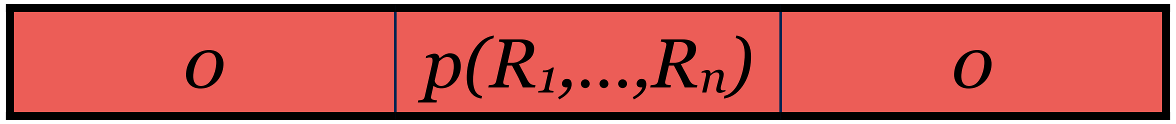
x_2 | R_2 | 0

…

x_n | R_n | 0

Key Generation of polynomial p:

p ,



0 | $p(R_1,\ldots,R_n)$ | 0

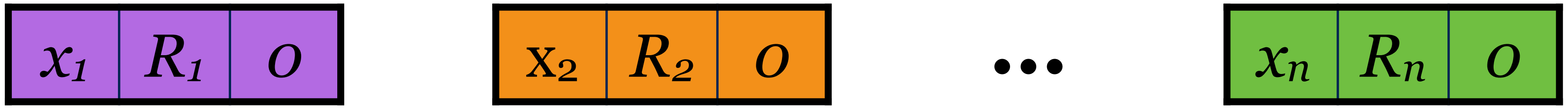
WHY IS IT SECURE?

$p(R_1,\ldots,R_n)$ in second slot "forces" homomorphic evaluation of p on ciphertext encodings

Construction of PAFE (Intuition)

Setup: Pick $R_1,\dots,R_n$

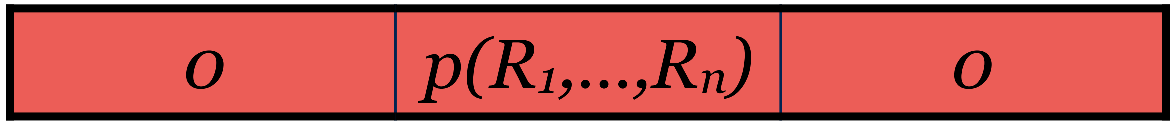
Encryption of x:



x_1 | R_1 | 0 ... x_2 | R_2 | 0 ... x_n | R_n | 0

Key Generation of polynomial p:

p ,



0 | $p(R_1,\dots,R_n)$ | 0

MAIN ISSUE: Mix-and-match attacks

encodings from different ciphertexts can be mixed

Construction of PAFE (Intuition)

Setup: Pick $R_1,\dots,R_n$

Encryption of x:

x_1	R_1	0	x_2	R_2	0	…	x_n	R_n	0

Key Generation of polynomial p:

p ,

0	$p(R_1,\dots,R_n)$	

Prevented by having *"ciphertext-specific"* checks!

MAIN ISSUE: Mix-and-match attacks

encodings from different ciphertexts can be mixed

Conclusions

- A new template for iO from degree-5 multilinear maps.
    - [Lin-Tessaro'17]: iO from **degree-3** multilinear maps
    - [Lin-Tessaro'17]: Show degree-D block-wise local PRGs + degree-D mmaps imply iO

Future Directions

- Explore notions of degree-2 PRGs that suffice to construct iO

- This would yield iO from bilinear maps

    - Negative Results on degree-2 PRGs [BBKK'17, LV'17] merci!