A Survey of Computational Assumptions on Bilinear and Multilinear Maps

Allison Bishop

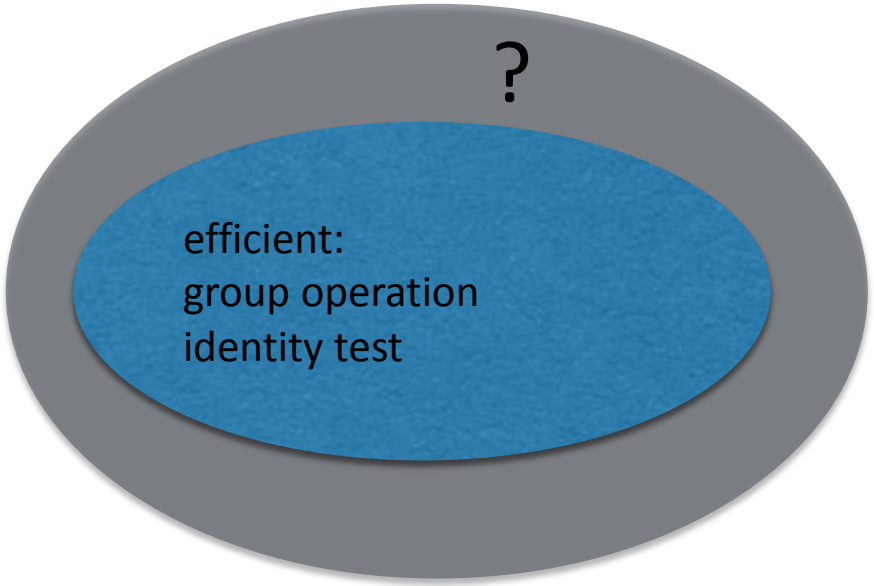
IEX and Columbia University

Group Basics

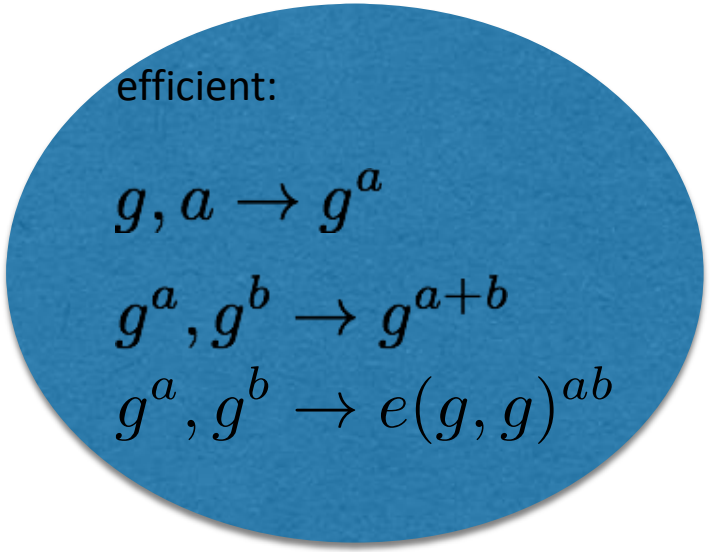
"There are two kinds of people in this world. Those who like additive group notation, and those who like multiplicative group notation."

$$g, g^a, g^b, g^{a+b}$$

inefficient:
discrete log



Bilinear Groups



When Faced with a New Group Assumption:

Kinds of Assumptions

- Generic group models
- q-type assumptions
- static assumptions

Variants:
Symmetric/Asymetric
Composite Order/ Prime Order
Linear/ Bilinear/ MultiLinear

Kinds of Proof Techniques

- Brute Force
- Cancelation
- Encoding
- Dual System
- Deja Q
- …

basic generic group arguments
BB IBE
G06, W'05 …
W09, LW10, LOSTW10,…
CM14, W16

Billinear Diffie-Hellman Assumption

Symmetric group:

$$g \in G, e : G \times G \rightarrow G_T$$

Given:

$$g, g^x, g^y, g^z$$

Distinguish:

$$e(g,g)^{xyz} \text{ from } e(g,g)^r$$

SXDH Assumption

Asymmetric group:

$$g \in G, h \in H, e : G \times H \rightarrow G_T$$

Given:

$$g, h, g^a, g^b$$

Distinguish:

$$g^{ab} \text{ from } g^r$$

A Basic q-type Assumption

Symmetric group:

$$g \in G, e : G \times G \rightarrow G_T$$

Given:

$$g, g^s, g^a, g^{a^2}, g^{a^3}, \ldots, g^{a^q}$$

Distinguish:

$$e(g,g)^{sa^{q+1}} \text{ from } e(g,g)^r$$

A Driving Example: IBE

$$PP : g, g^a, g^b, e(g,g)^\alpha$$

$$CT : Me(g,g)^{\alpha s} \qquad g^s \qquad g^{s(aID+b)}$$

$$SK : \qquad g^{\alpha + r(aID+b)} \qquad g^r$$

Decryption:

$$e(g,g)^{\alpha s + sr(aID+b) - sr(aID+b)}$$

$$= e(g,g)^{\alpha s}$$

Arguing Generic Security

Look at exponents you are given in G:

Look at the blinding factor:

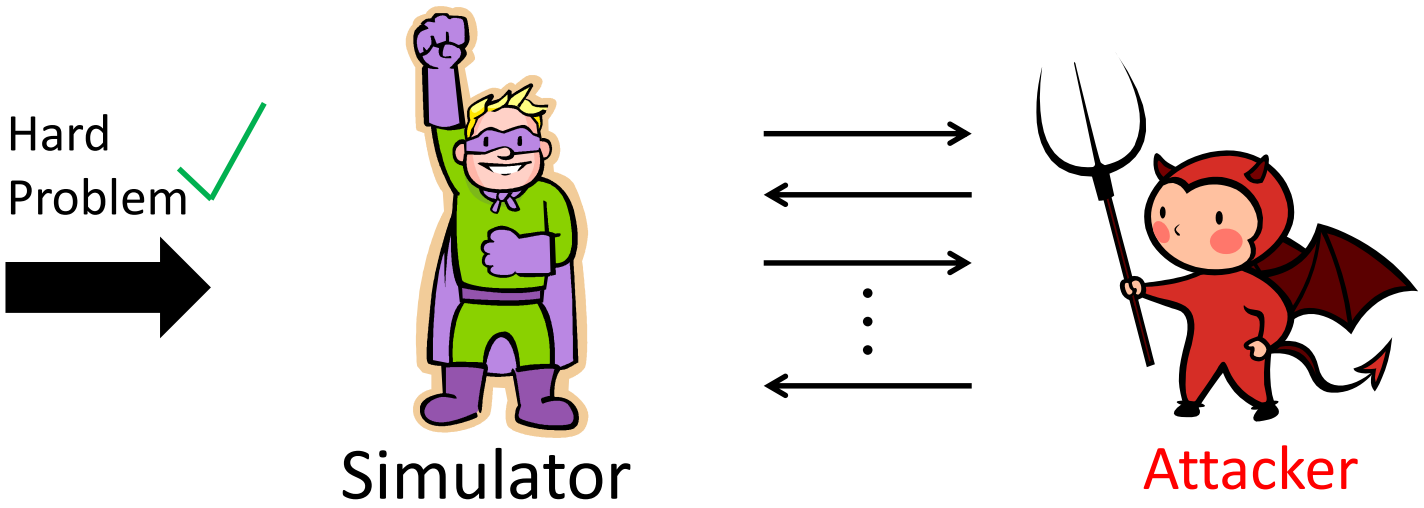
$$\alpha s$$

$$a,\ b,\ s,\ s(aID^* + b),$$
$$\alpha + r_1(aID_1 + b), r_1, \ldots, \alpha + r_q(aID_q + b), r_q$$

All you can do is take linear combinations of degree at most 2

$$\text{No way to get } \alpha s \text{ alone when } ID^* \neq ID_i \forall i$$

Proof Challenges Beyond Generic Security



Simulator must balance two competing goals:

answer
attacker
queries

leverage
attacker
success

Arguing Selective Security

- Embed the challenge as a function of known ID*

$$\alpha = xy,\ s = z$$

$$a = x,\ b = -xID^* + w$$

Given:

g, g^x, g^y, g^z

$$PP : g, g^a = g^x, g^b = (g^x)^{-ID^*} g^w, e(g,g)^\alpha = e(g^x, g^y)$$

$$CT : g^s = g^z, g^{s(aID^*+b)} = (g^z)^w$$

Distinguish:

$e(g,g)^{xyz}$ from $e(g,g)^r$

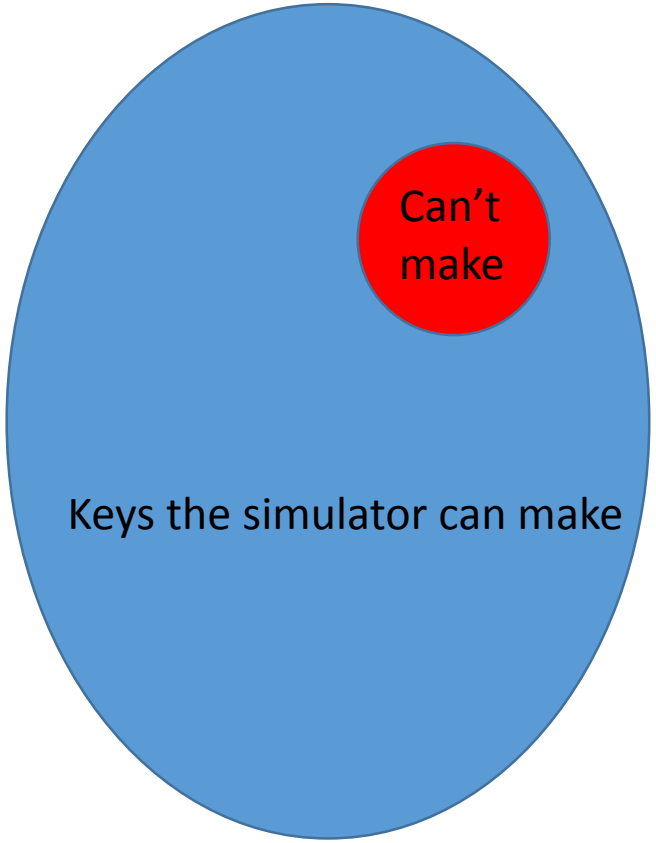
Choose $r = -\frac{y}{ID - ID^*} + r'$

then $\alpha + r(aID + b) = xy - xy + rw = rw$

Simulator can produce key for any ID not equal to ID!*

How to Leverage a q-Type Assumption [example from W05]

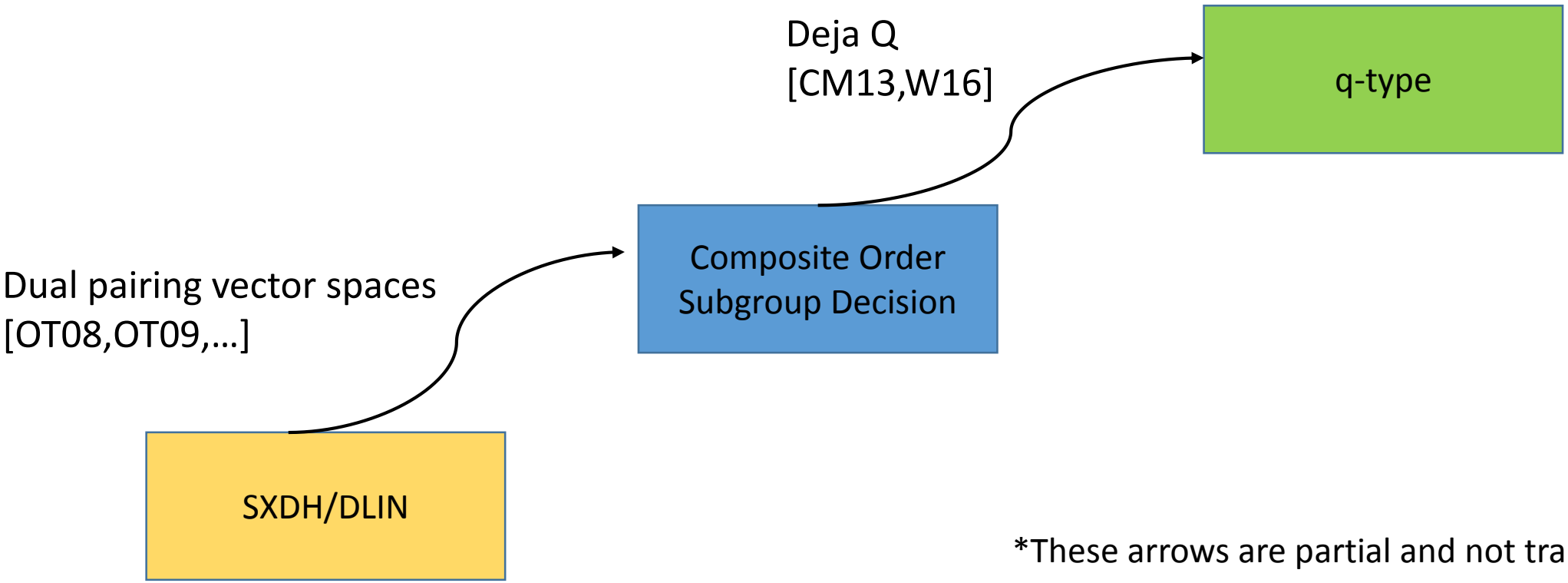
What if we don't want to fix ID* ahead of time?



To partition small PP with parameter q:

Use a q-size assumption!

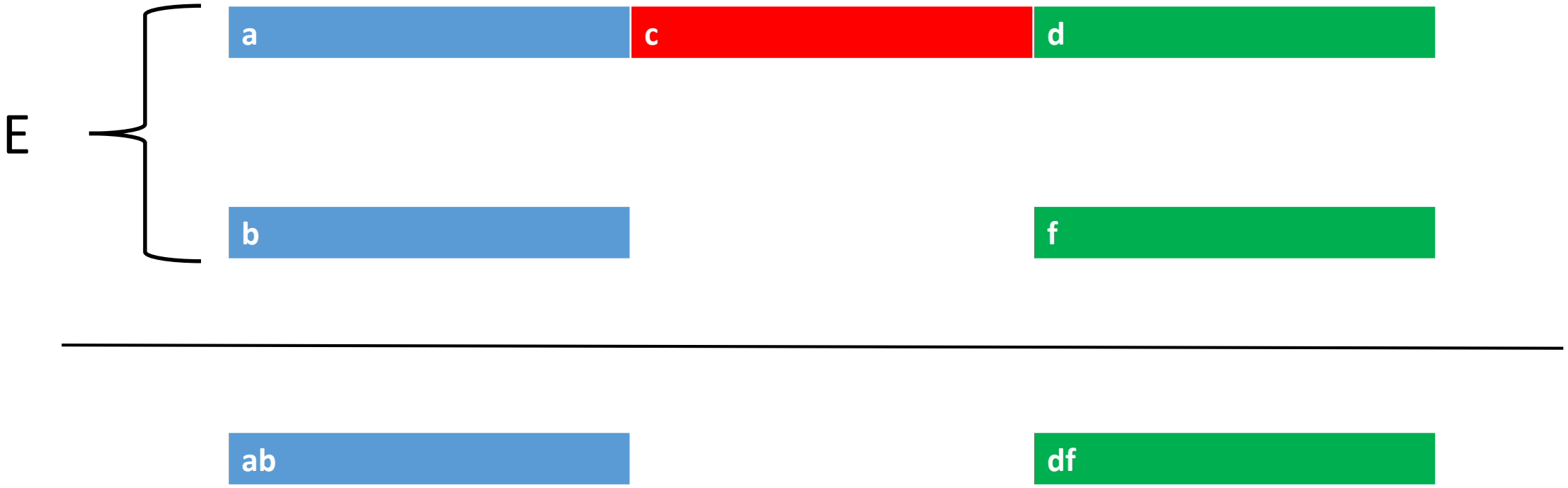
Simulation Techniques



*These arrows are partial and not transitive!

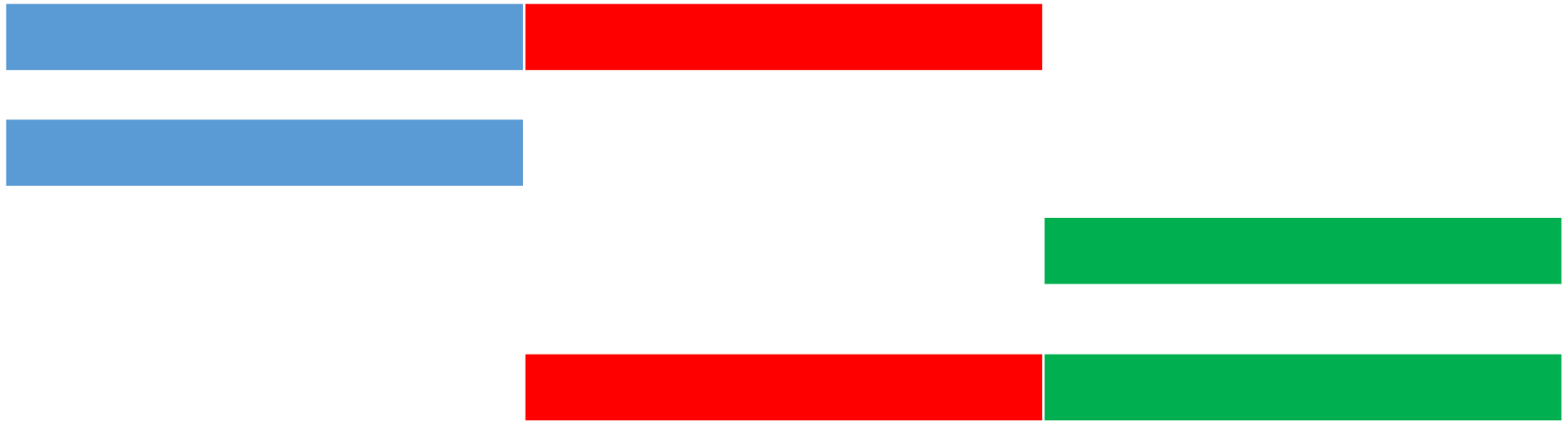
Composite Order Bilinear Groups

How the pairing operates:



Subgroup Decision Assumptions in Composite Order Bilinear Groups

Example: Given



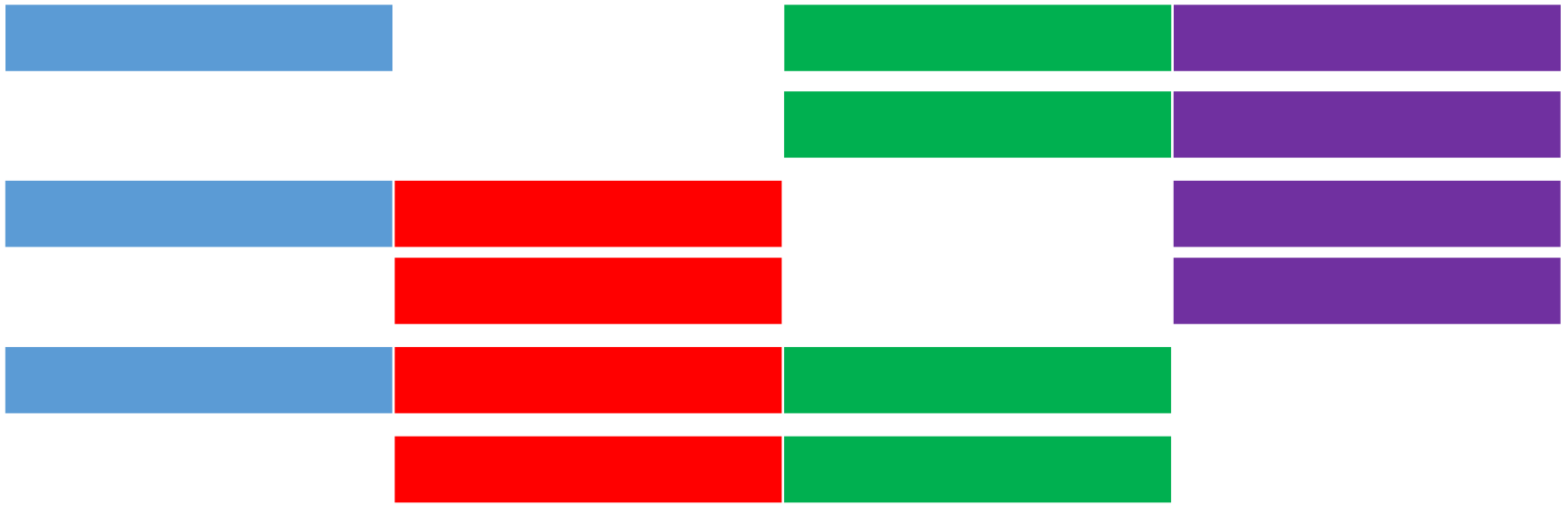
Distinguish

from

Subgroup Decision in a Multilinear Group?

Here's what it might look like in a 3-linear group:

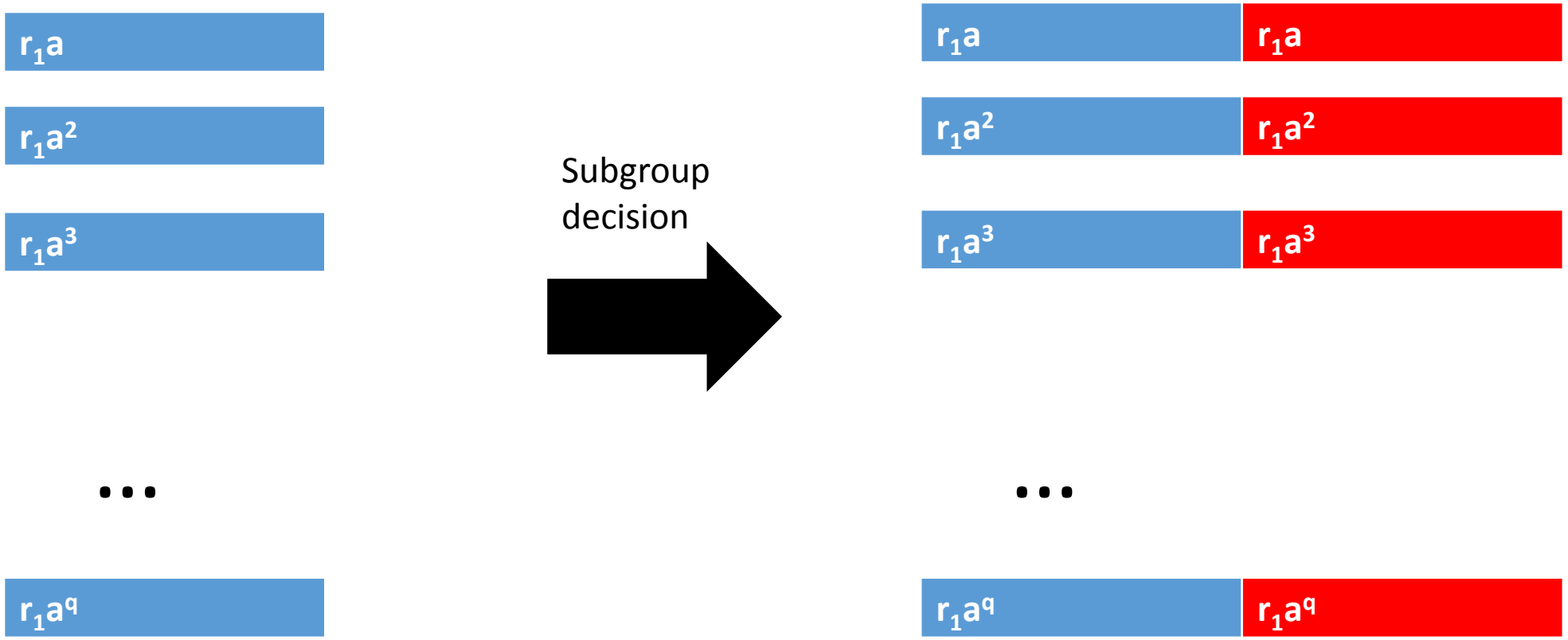
Given



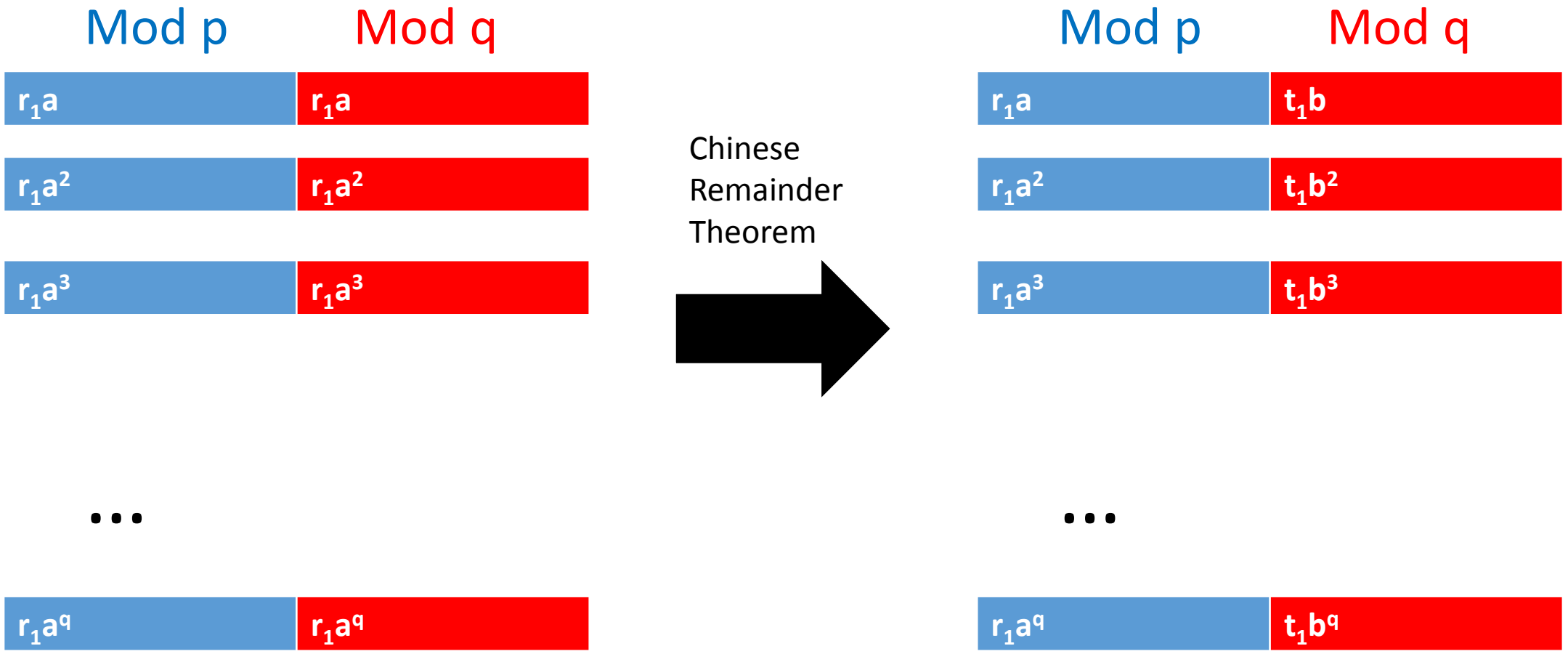
Distinguish

from

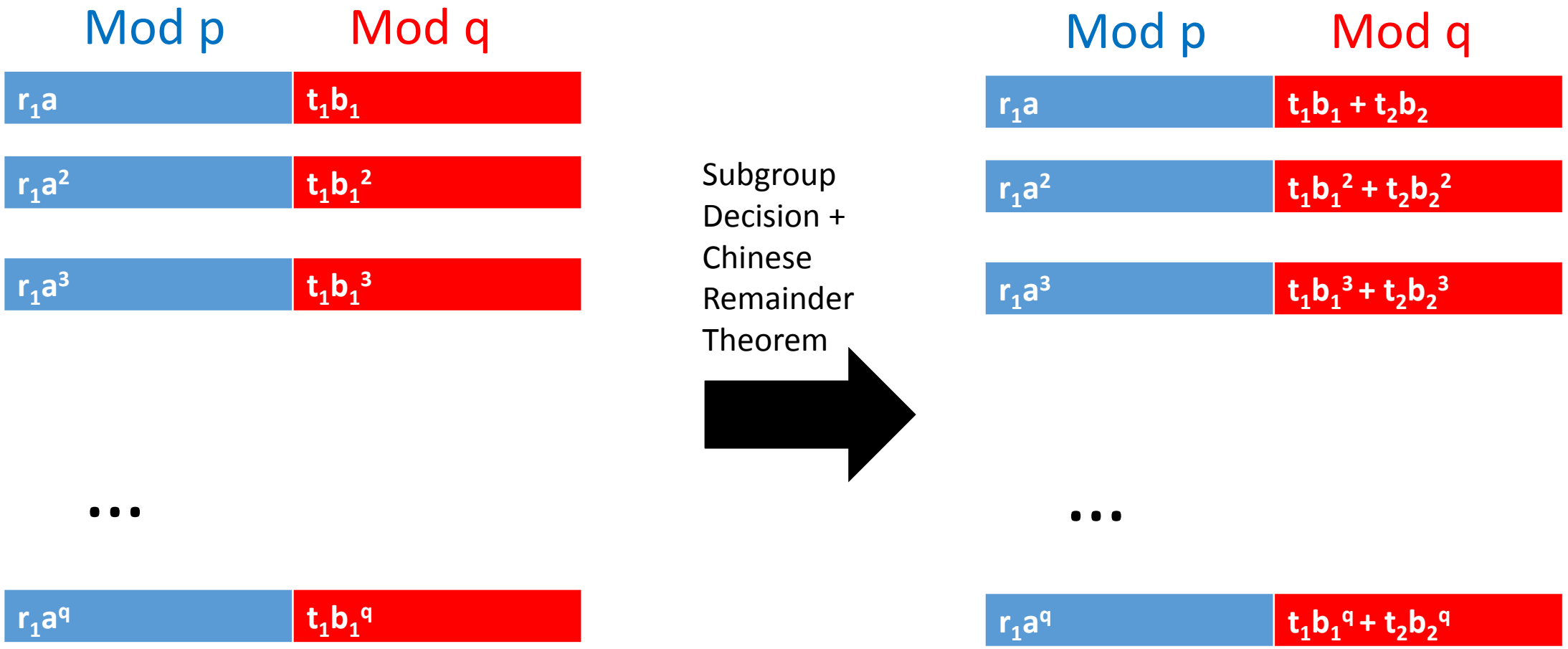
Deja Q – Basic Example



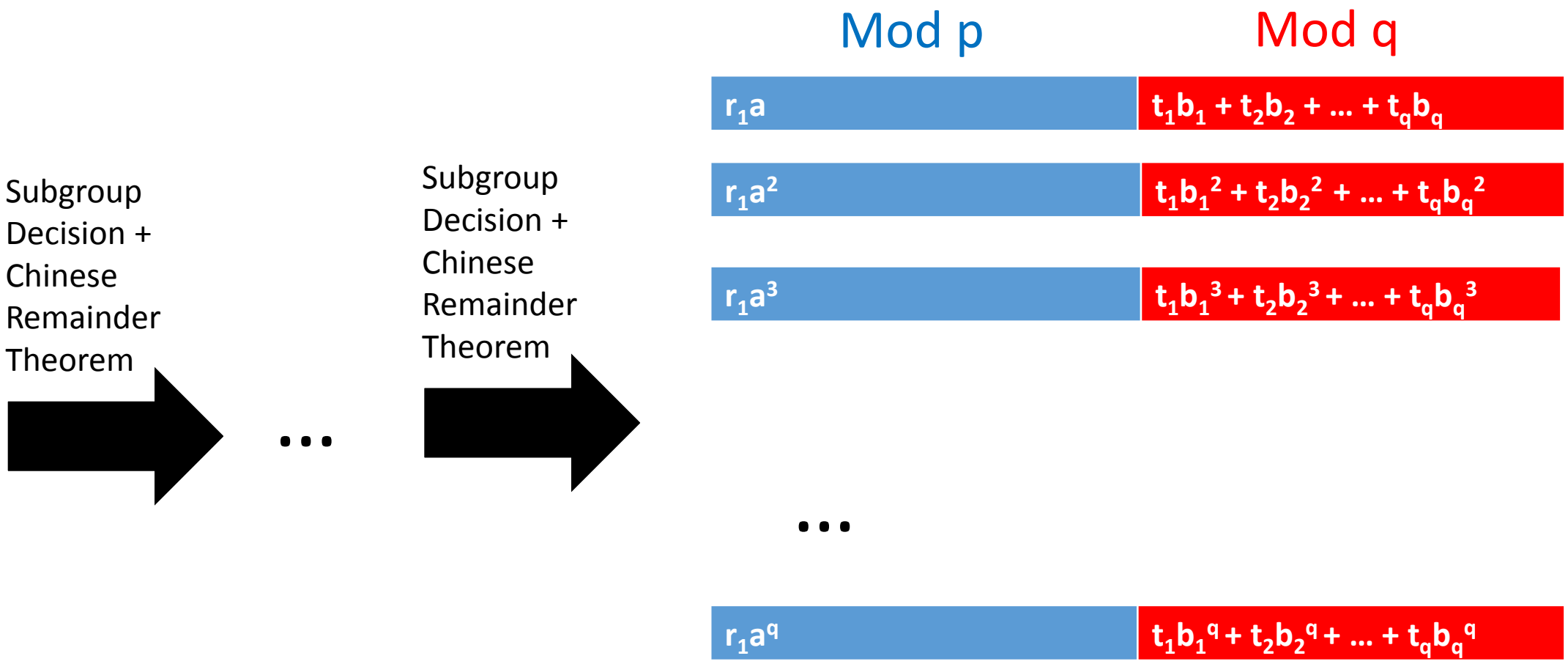
Deja Q – Basic Example



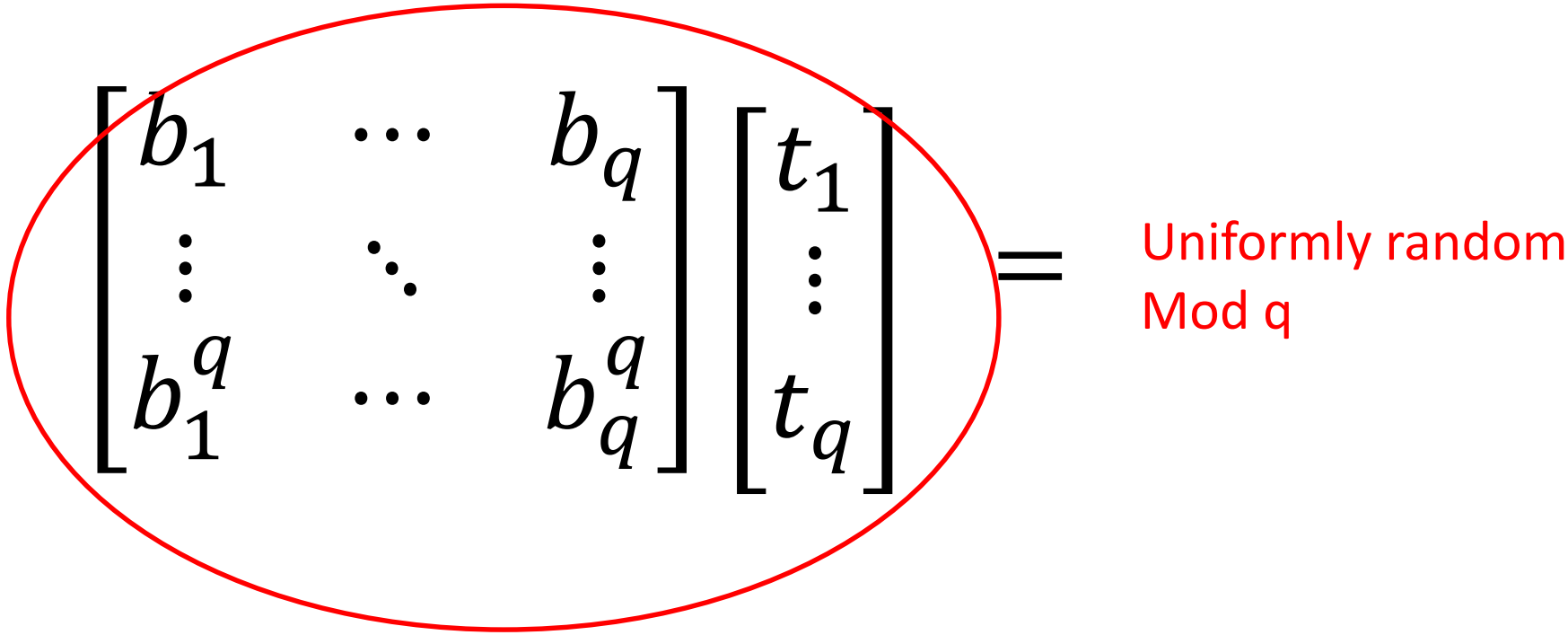
Deja Q – Basic Example



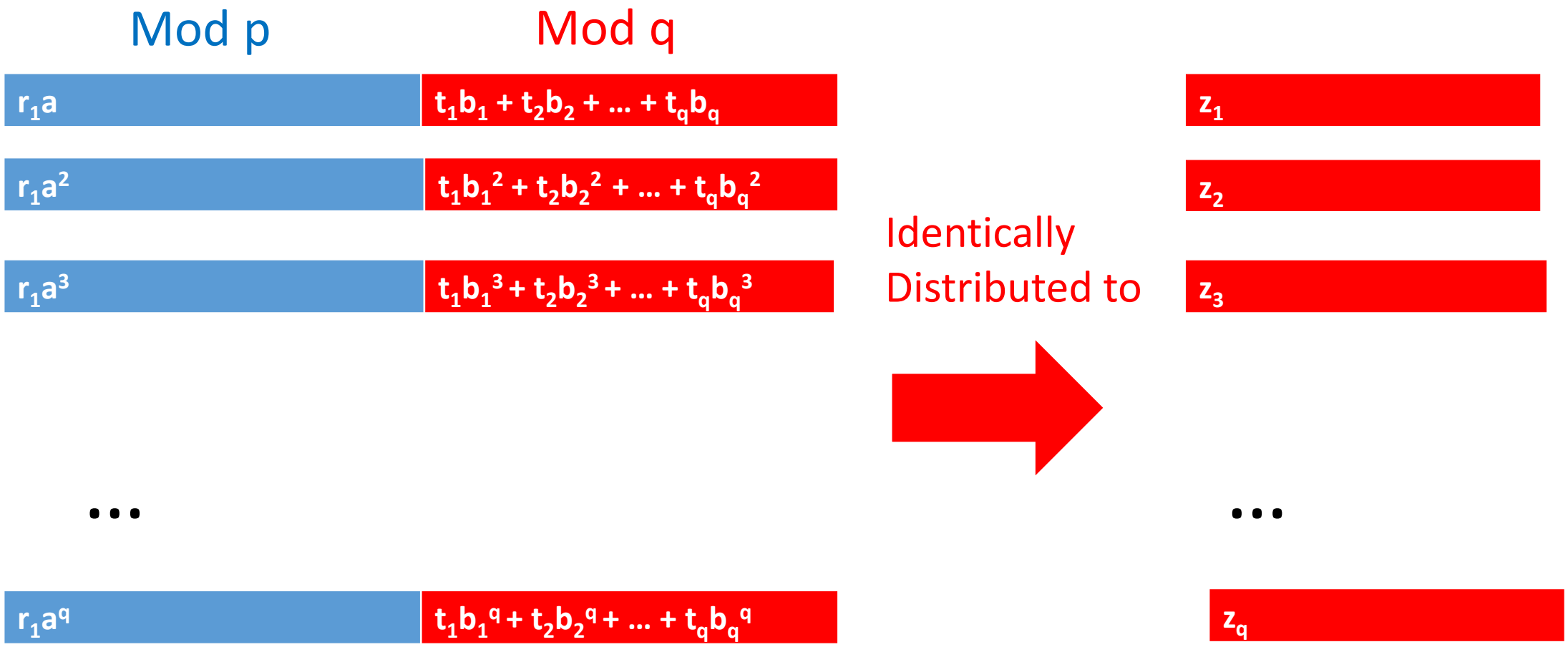
Deja Q – Basic Example



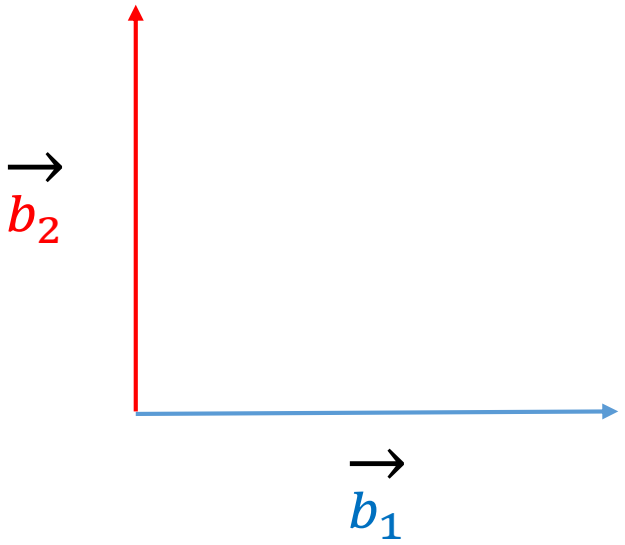
Deja Q – Basic Example



Deja Q – Basic Example



Dual Pairing Vector Spaces



$$b_1, b_2 \leftarrow \mathbb{Z}_p^d$$

$$b_1^*, b_2^* \text{ s.t. } b_i \cdot b_j^* = 0 \text{ if } i \neq j,$$

$$b_i \cdot b_i^* = 1$$

Emulates some features of composite order, asymmetric group:

E

$$\frac{g^{rb_1+tb_2} \quad g^{sb_1^*+zb_2^*}}{e(g,g)^{rs+tz}}$$

Emulating Subgroup Decision using SXDH

Asymmetric group: $g \in G, h \in H, e : G \times H \to G_T$

Given:

g, h, g^a, g^b

Distinguish:

g^{ab} from g^r

$$b_1 := ax_1 + x_2 \quad b_2 := x_2$$

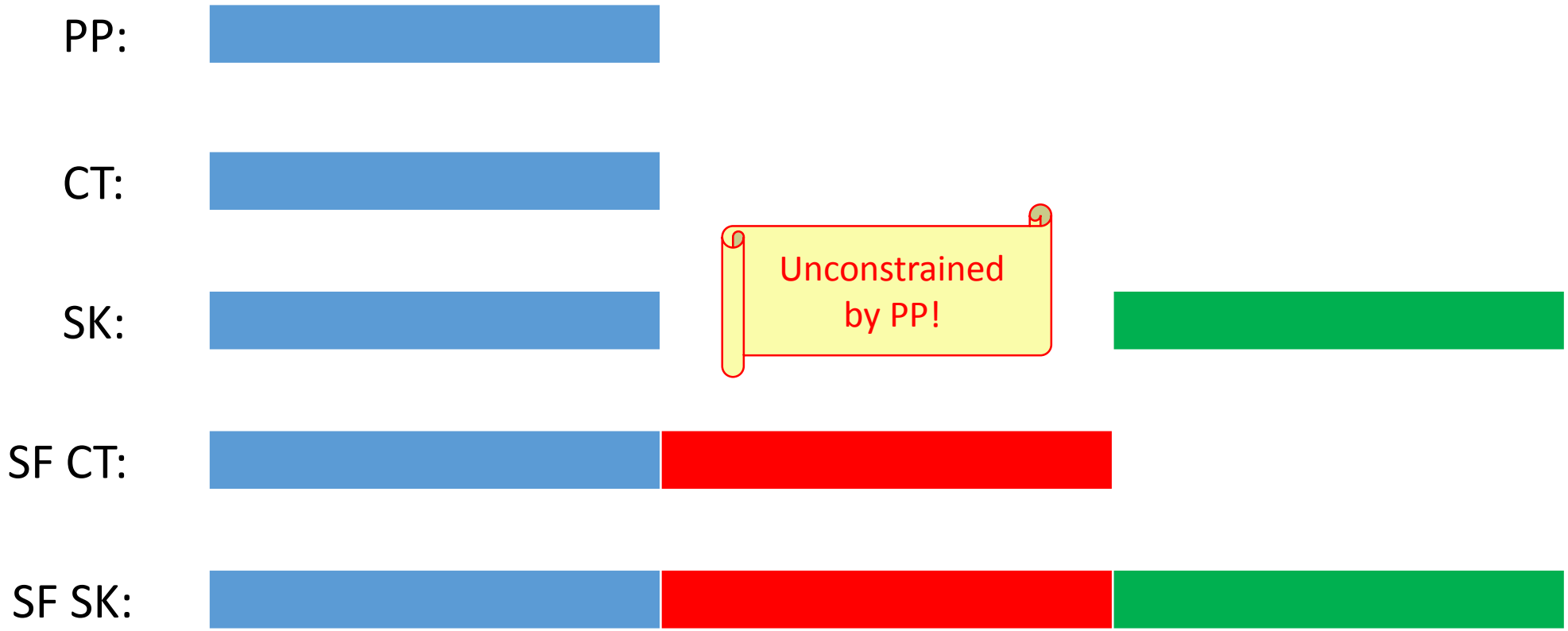
$$b_1^* = a^{-1}x_1^* \quad b_2^* = x_2^* - ax_1^*$$

$$\text{can make } g^{b_1}, g^{b_2}$$

$$\text{can make } h^{rb_1^*}, \text{ not } h^{rb_2^*}$$

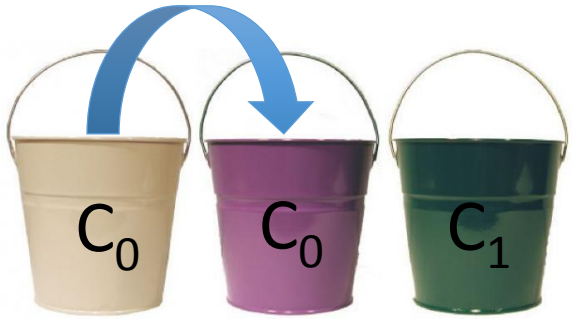
Dual System – Using Subgroup Assumptions for Functional Encryption [W09 + too many to cite*]

Most Basic Template:



Using Subgroup Assumptions for Obfuscation [GBSW 15]

- Reduction will ***isolate*** each input.
- Main idea:
  - Have poly many "parallel" obfuscations, each responsible for a bucket of inputs
  - Hybrid Type 1: Allocate/Transfer inputs among different buckets, ***but*** programs do not change at all. **Assumption used here.**



Ok. So what are these buckets really like?

Simple example:

Want to implement:

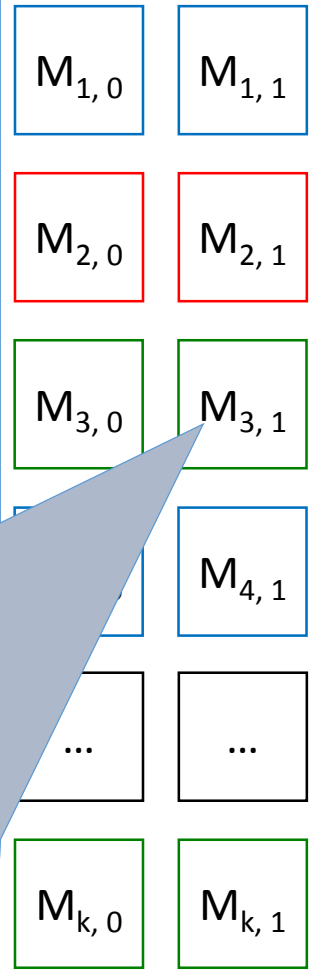
$F(x_1\ x_2) = XOR(\ x_1, x_2\ )$

$$M_{1,0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad M_{1,1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$M_{2,0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad M_{2,1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

[Barrington]: All log-depth (NC^1) circuits have poly-size Matrix Branching Programs

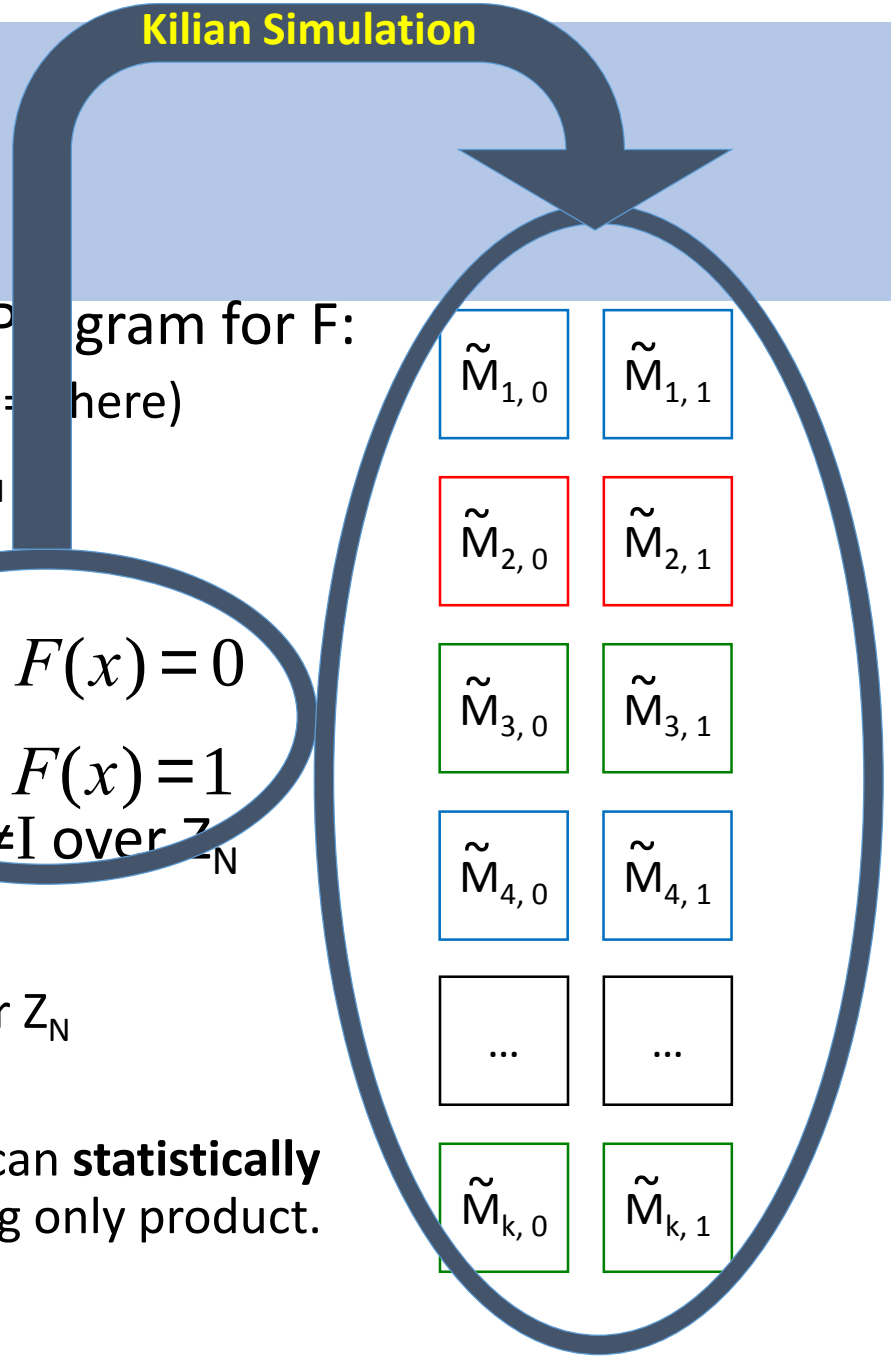


Towards Obfuscation

- Oblivious Matrix Branching P gram for F:
  - n-bit input $x=x_1x_2\ldots x_n$ (e.g. n= here)
  - 2k invertible matrices over Z_N
  - Evaluation on x:

$$\tilde{O}\ M_{i,x_{(i \bmod n)}} = \begin{cases} I & if\ F(x)=0 \\ B & if\ F(x)=1 \end{cases}$$

  - Where B is fixed matrix $\neq I$ over Z_N
- Kilian Randomization:
  - Chose $R_1, \ldots, R_{k-1}$ random over Z_N
  - $\tilde{M}_{i,b} = R_{i-1} \cdot M_{i,b} \cdot R_i^{-1}$
  - Kilian shows that for each x, can **statistically** simulate M_x matrices knowing only product.



Hybrids Intuition