EUROCRYPT 2021 PRESENTS

THE RISE OF PAILLIER

Homomorphic Secret Sharing and Public-Key Silent OT

Starring:
CLAUDIO ORLANDI
PETER SCHOLL
SOPHIA YAKOUBOV

Aarhus University

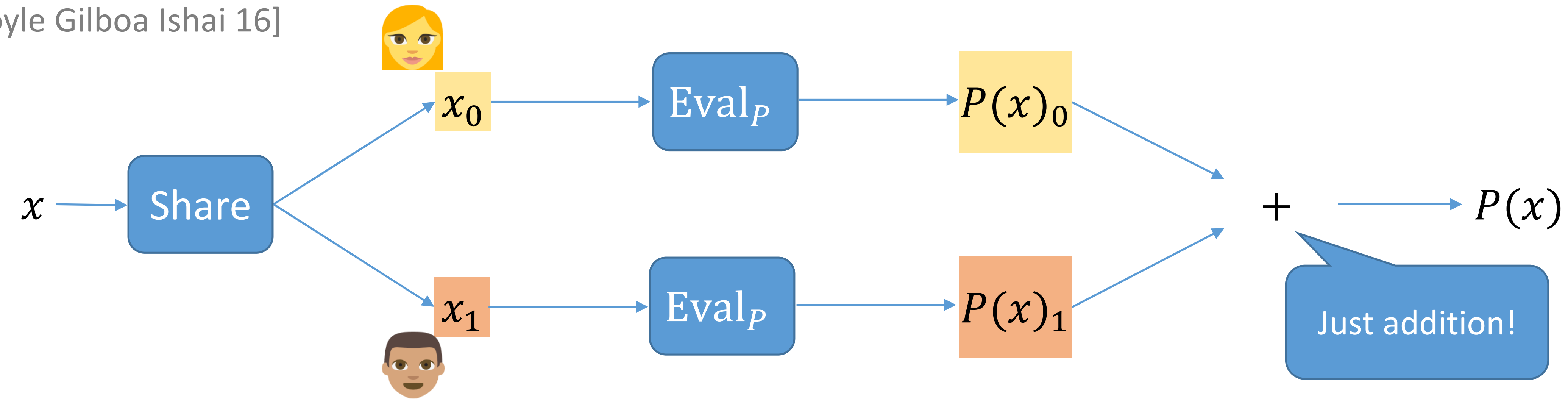
Outline

- Homomorphic Secret Sharing
  - Background
  - Share conversion and distributed multiplication for Paillier

- Pseudorandom correlation functions
  - Producing correlated randomness
  - Public-key setup for vector-OLE and oblivious transfer

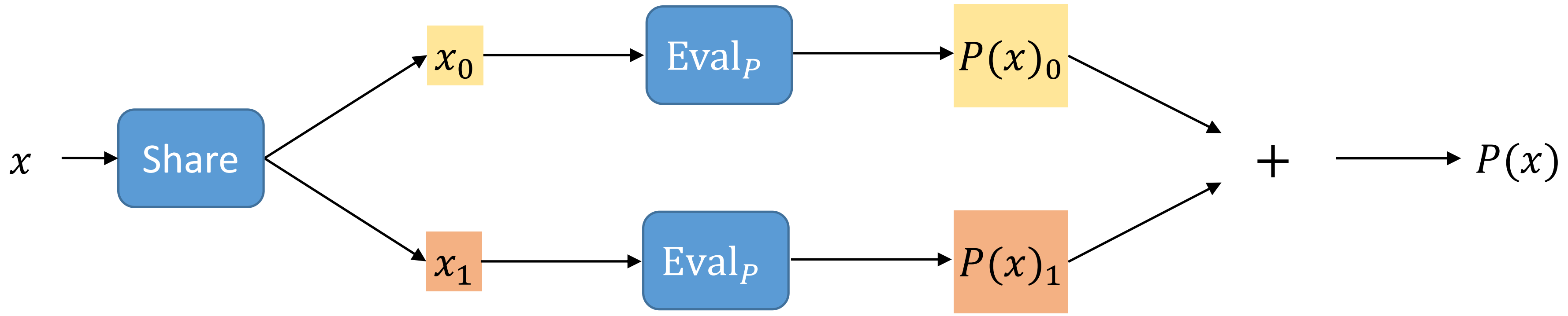
Homomorphic Secret Sharing

[Boyle Gilboa Ishai 16]



- Security: x_0 hides x, x_1 hides x
- Correctness: $\mathrm{Eval}_P(\ x_0\ ) + \mathrm{Eval}_P(\ x_1\ ) = P(x)$

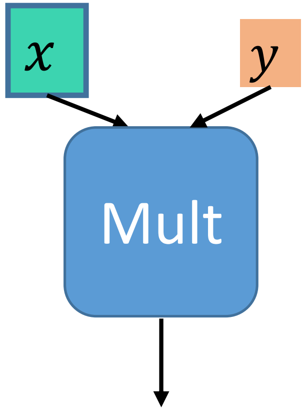
HSS Landscape



	Assumptions	Program type	Error pr.	Msg space
[Ben86]	None	linear	negl	exp
[DHRW16, BGI15, BGILT16]	LWE+	any	negl	exp
[GI14, BGI15]	OWF	simple (e.g. point)	negl	exp
[BCGIKS19]	LPN	low-deg polynomials	negl	exp
[BKS19]	LWE	Branching programs	negl	exp
[BGI16, FGJS17]	DDH, Paillier	Branching programs	1/poly	poly
This work	Paillier	Branching programs	negl	exp

HSS for Branching Programs: High-Level Template [Boyle Gilboa Ishai 16]

Branching program model: circuit, where every multiplication involves an input wire

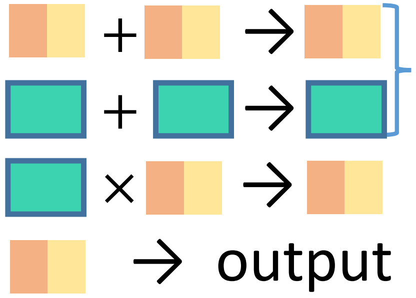


Value types:

Input : ciphertext $\mathrm{Enc}(x)$

Memory : linear shares y_0 y_1

Operations:

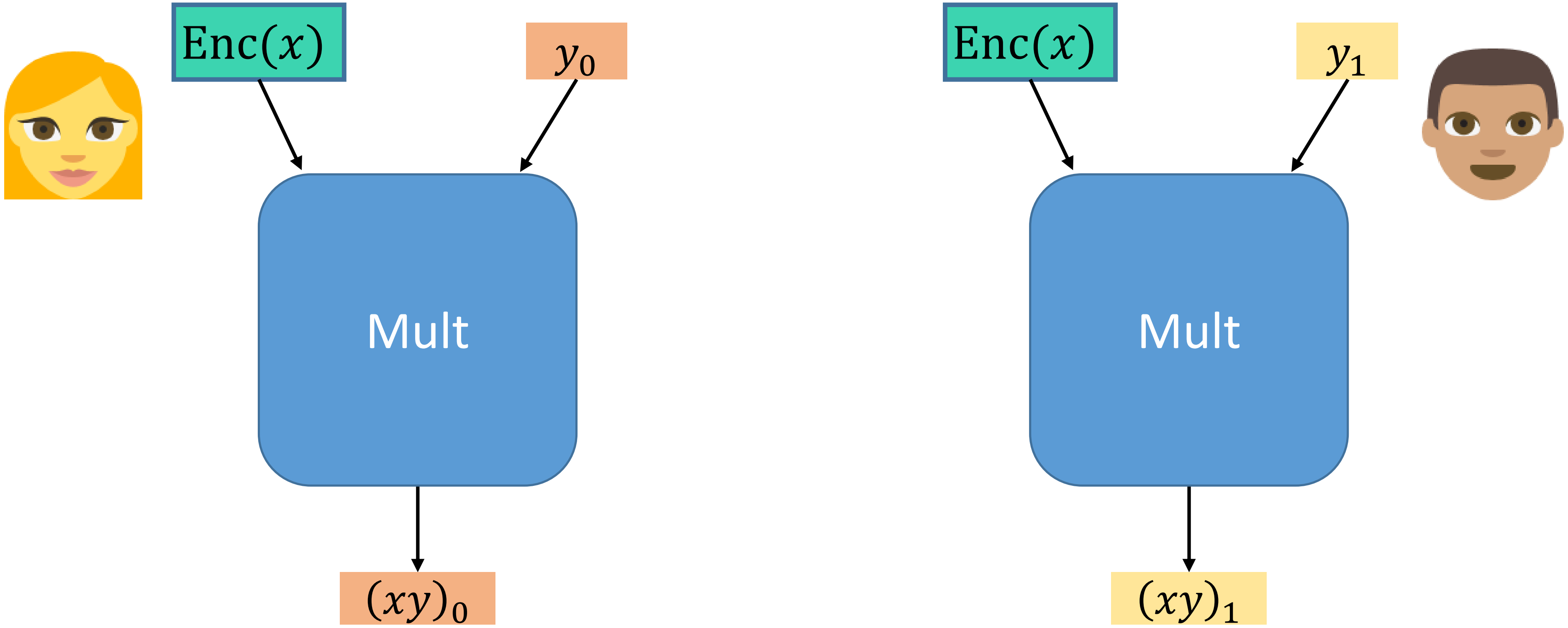


: via linearity

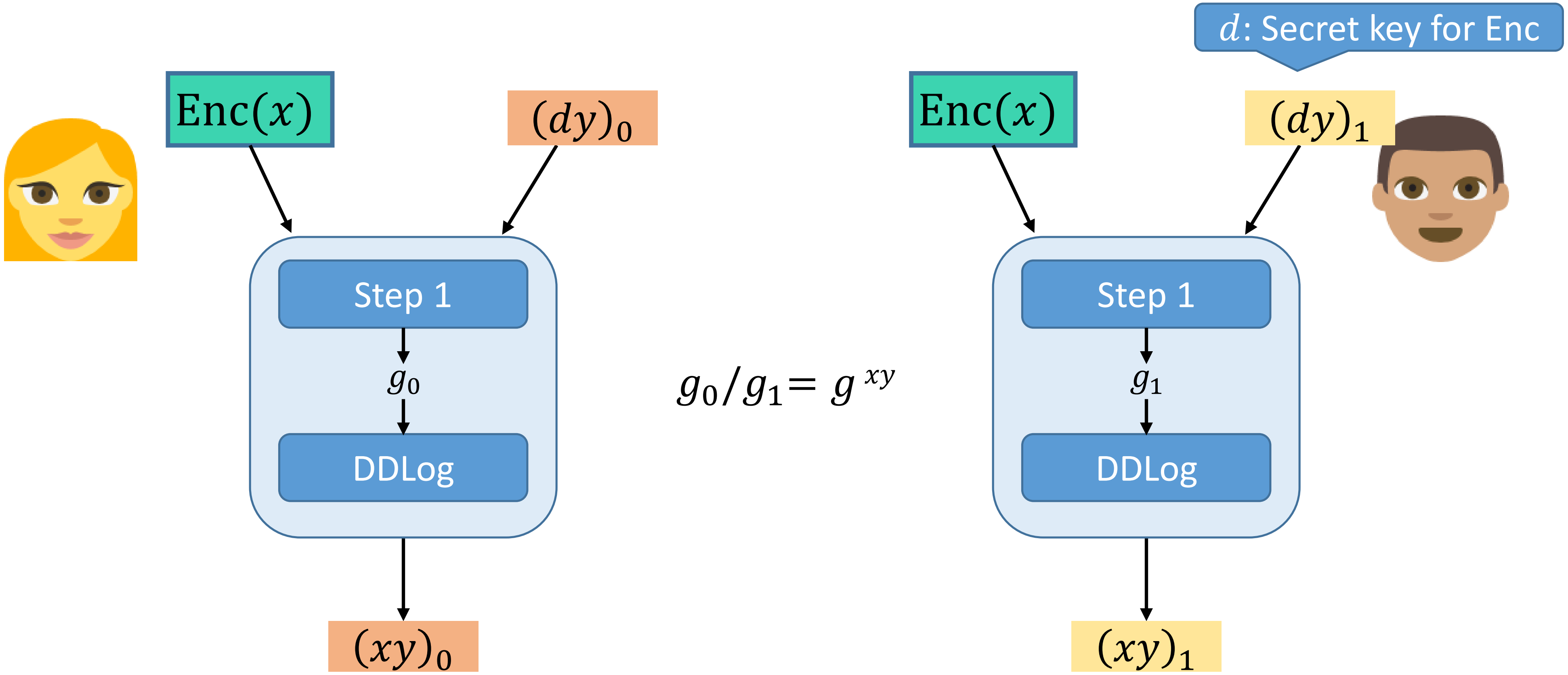
: ?

: reconstruction

Blueprint for Multiplication



Blueprint for Multiplication

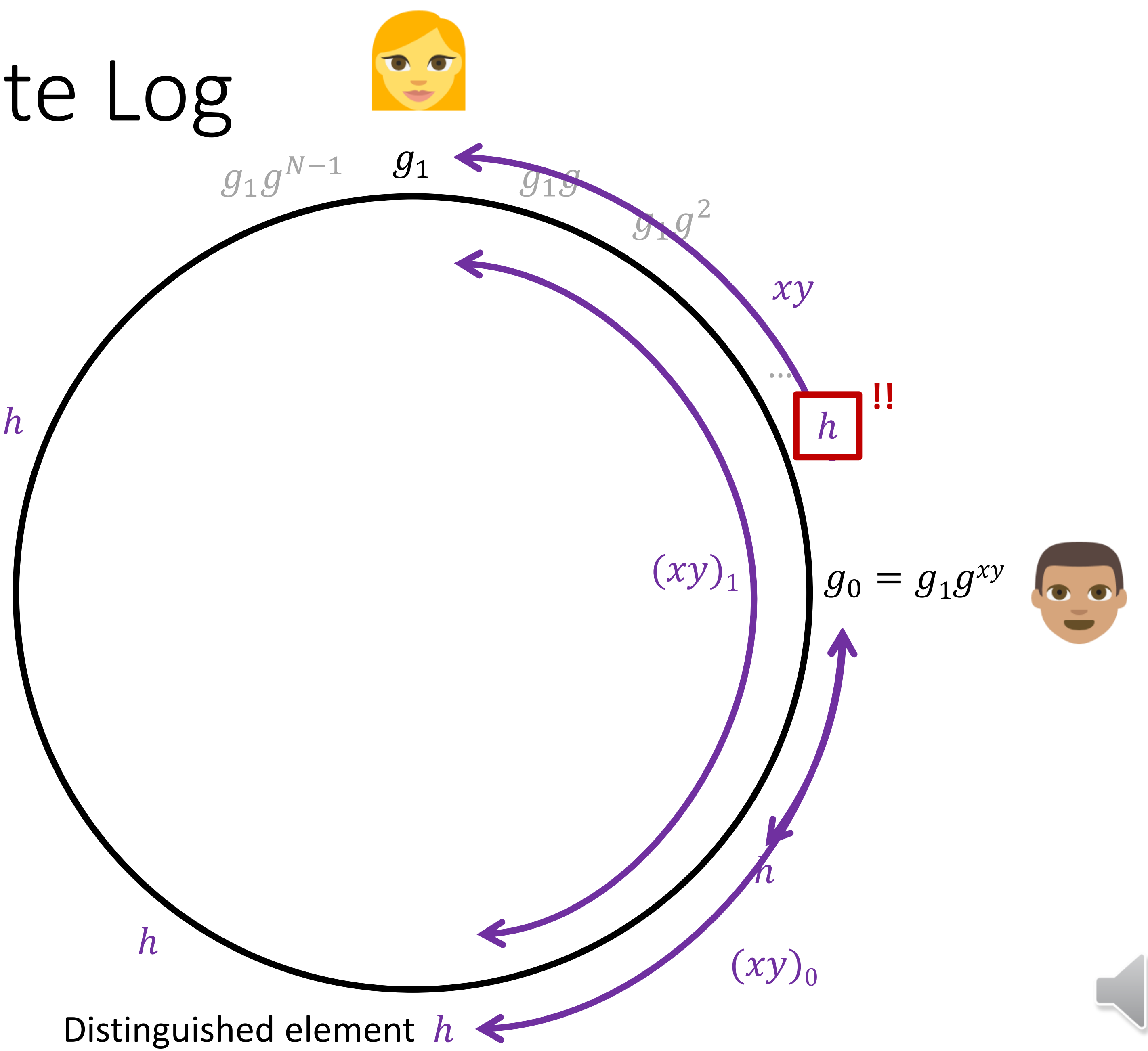


$$(xy)_0 + (xy)_1 = xy$$

Distributed Discrete Log

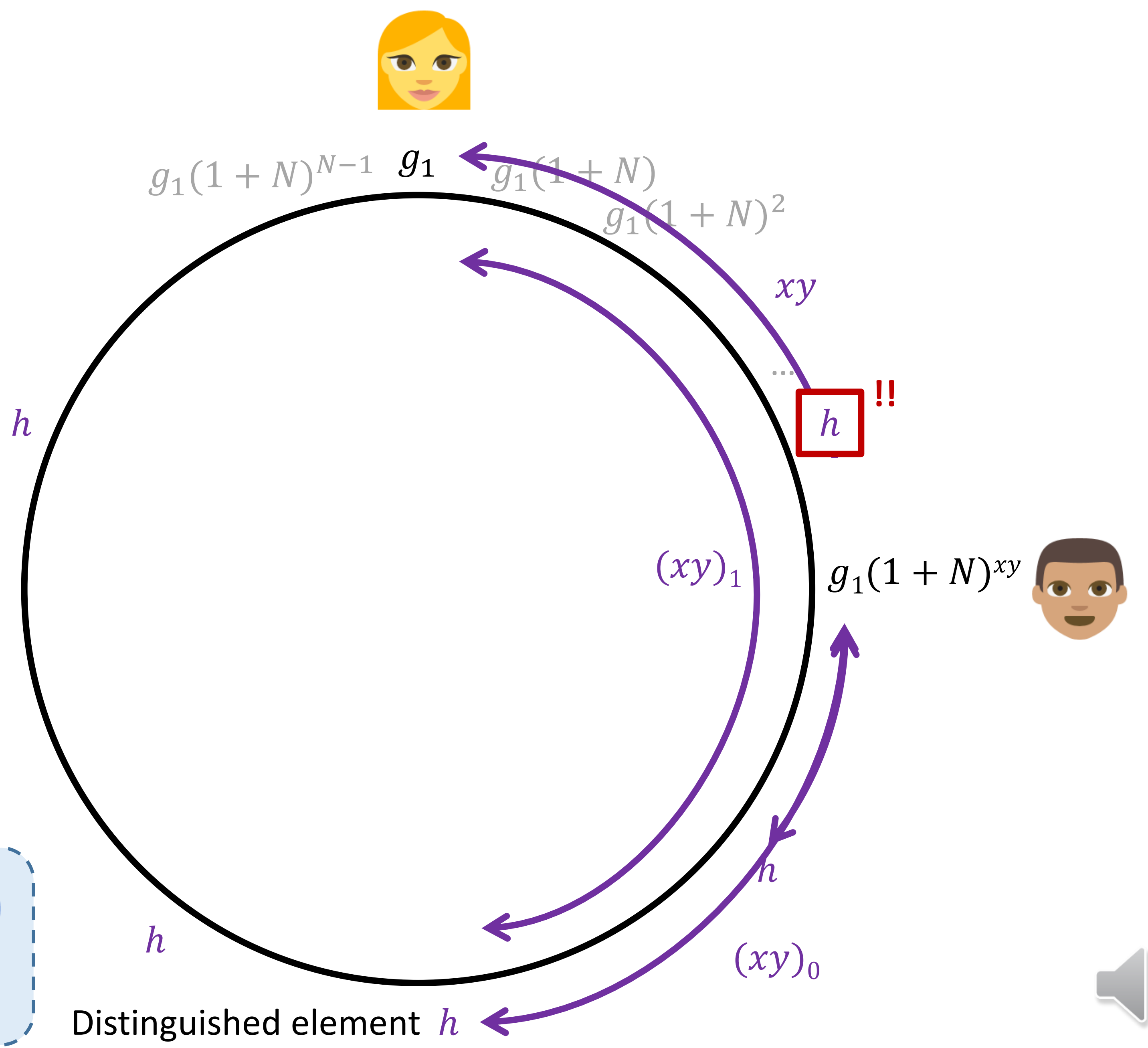
[Boyle Gilboa Ishai 16]

- $g_0 / g_1 = g^{xy}$
- $(xy)_1 - (xy)_0 = xy$
- Problem: what if $(xy)_0$, $(xy)_1$ are large?
  - Have many h's
  - Poly-size message space
- Problem: error if parties hit different h
  - Gives 1/poly error!
- Various optimizations: still 1/poly error, poly message space [BGI16, BGI17, BCGIO17, DKK18]
- Variant in Paillier groups: same limitations [FGJS 17]



DDLog: Paillier

- $g_0/g_1 = (1+N)^{xy} \mod N^2$
- Use just one h:
  - $h/g_i = (1+N)^{(xy)_i} \quad \rightarrow \quad (xy)_i$
  - Use $h := g\_1 \bmod N = g\_0 \mod N$ (in Z_{N^2})
- h is in the same coset!
  - Intuition: g_1 mod N uniquely represents the coset defined by g_1
  - $(h \mod N) = h \Rightarrow h$ is in the coset
- Large message space, negl error!



Paillier 101:
- Paillier group: $Z^*_{N^2}$, $N = pq$
- $1 + N$ generates an easy DLog subgroup

HSS from Paillier: summary

- Basic construction:
  - Negligible correctness error, exponential message space
  - To repeatedly multiply, need circular security of Paillier
  - **Concurrent work:** [Roy-Singh 21] using Damgård-Jurik
- Public-key variant:
  - Share input without knowing private key
  - Use ElGamal over $Z_{N^2}^*$
- Circular secure variant:
  - Based on [Brakerski-Goldwasser 10] encryption

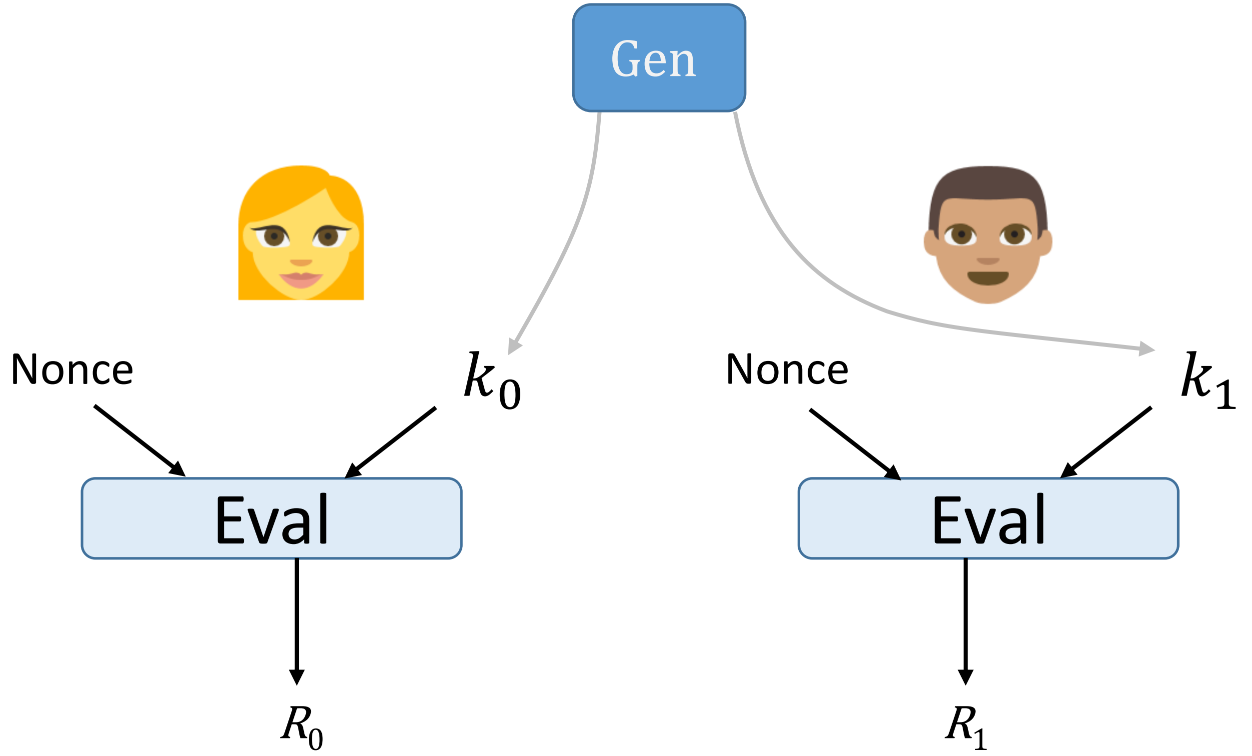
Share size:
$O(1)$ group elements

$O(\lambda)$ group elements

II: Pseudorandom Correlation Functions

Pseudorandom Correlation Function

[BCGIKS20]

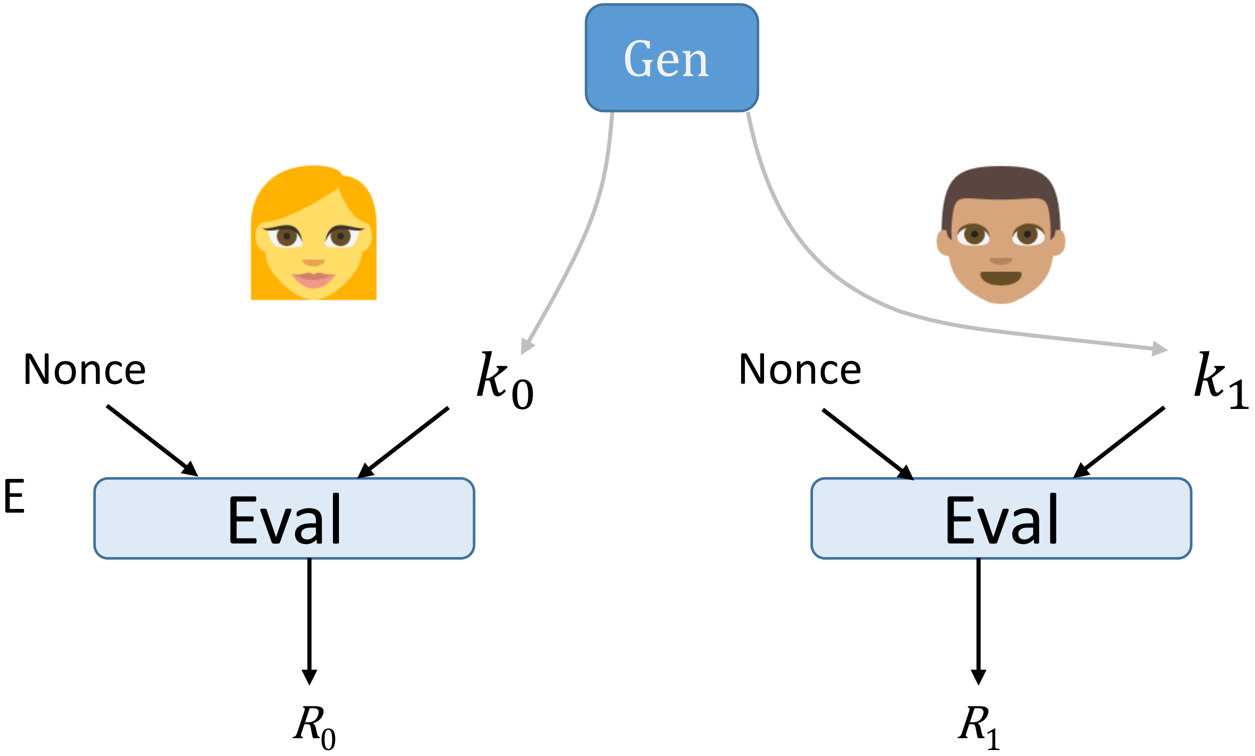


Correlation	R_0	R_1
Random Oblivious Transfer (OT)	b, s_b	s_0, s_1
Oblivious Linear Evaluation (OLE)	$x, (xy)_0$	$y, (xy)_1$
Vector OLE (VOLE)	$x_i, (x_i y)_0$	$(x_i y)_1$

y

Pseudorandom Correlation Function

[BCGIKS20]



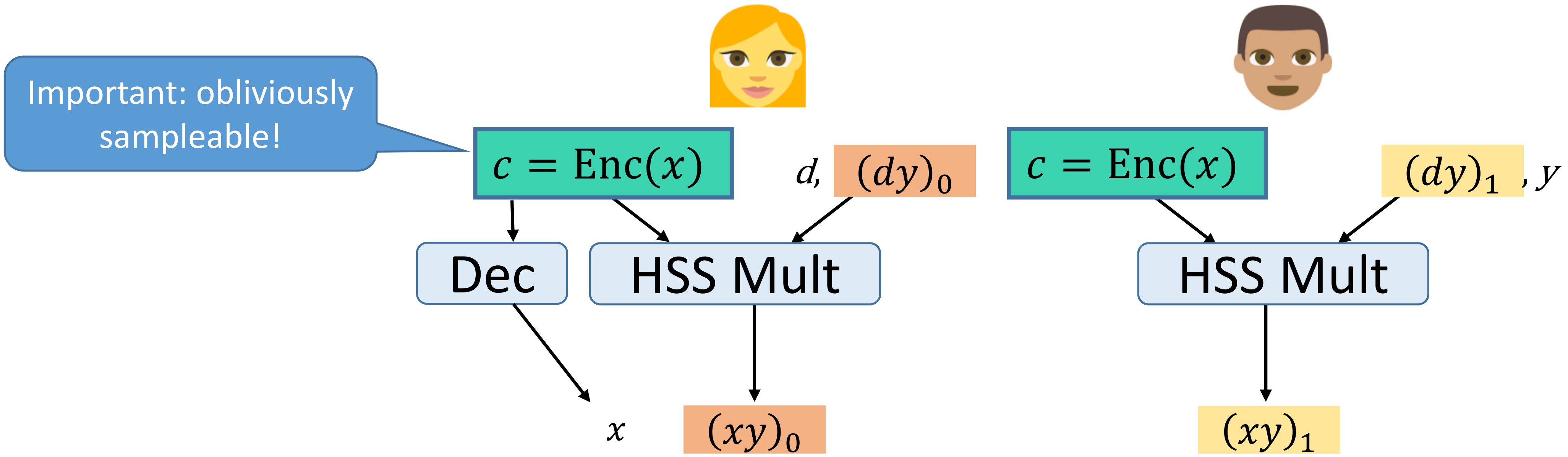
Previous constructions:

- additive correlations from LWE (expensive)
- OT, deg-2 correlations from Variable-density LPN (new assumption)

This work: VOLE, OT from Paillier, QR

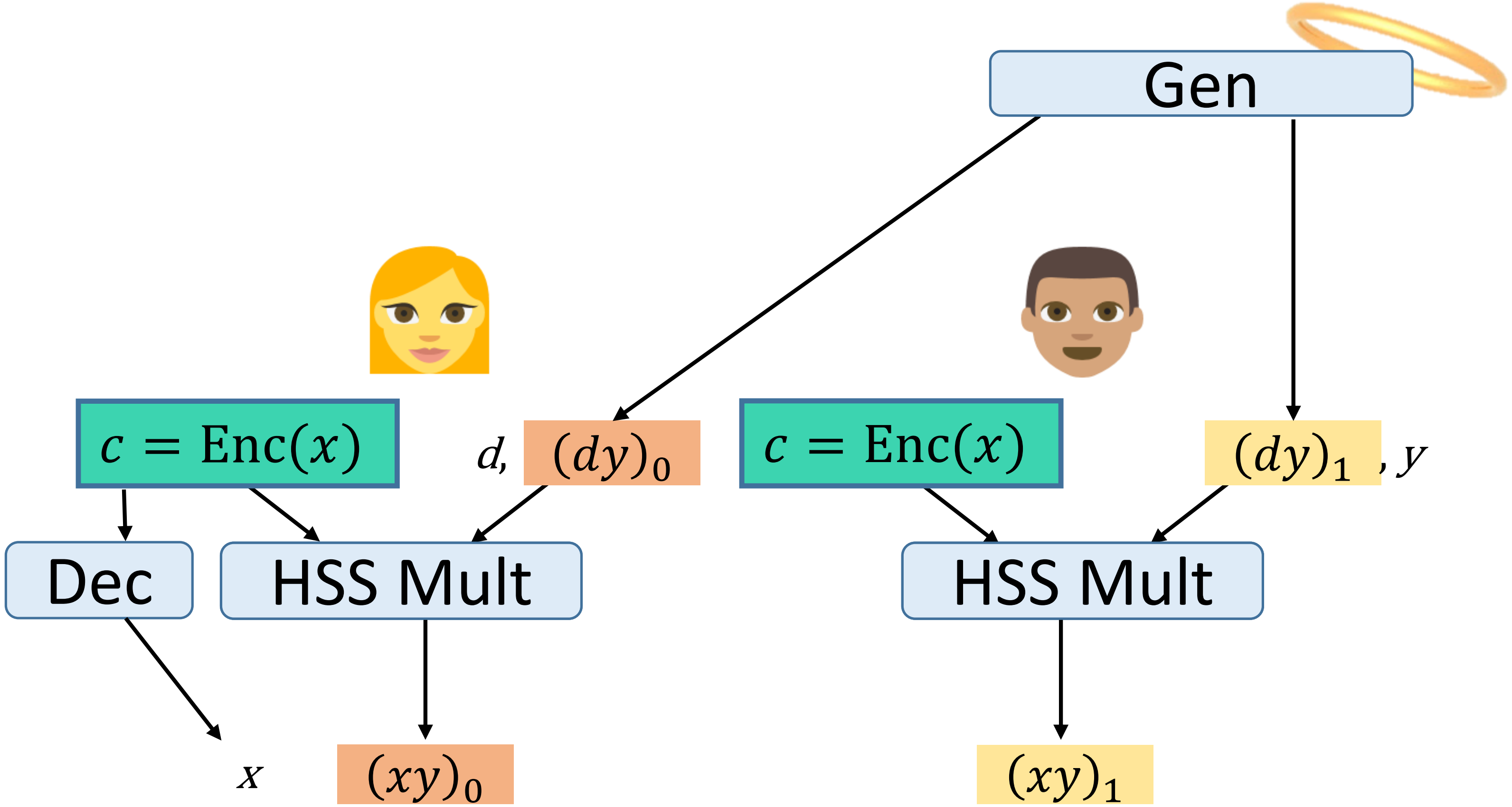
- Note on efficiency
  - All require exponentiations!
    ⇒ slower than LPN-based alternatives
  - Advantages: smaller keys, simpler constructions, standard assumptions

PCF for VOLE



Vector OLE (VOLE)	$x_i, (x_i y)_0$	$(x_i y)_1$	y

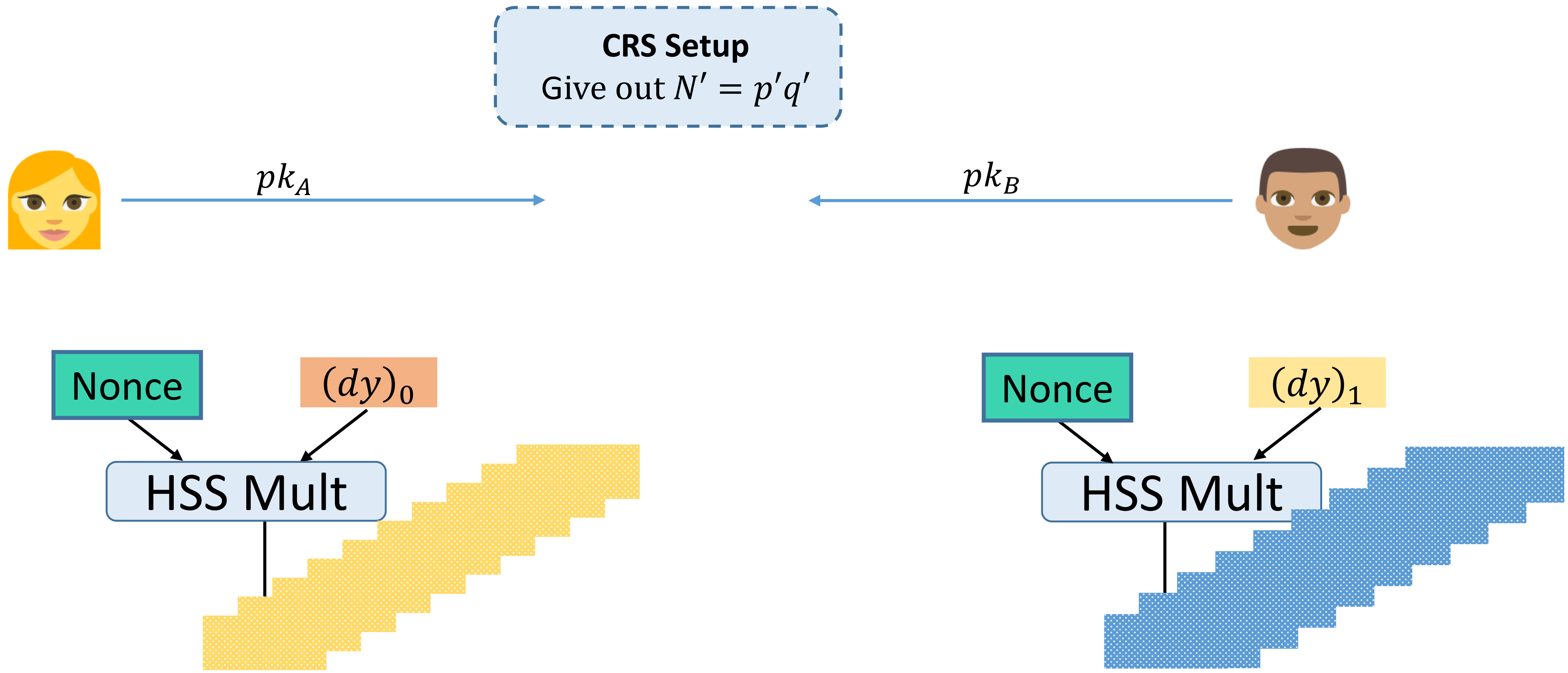
Setup?



Public-key setup:

- One message from Alice/Bob
- With non-interactive (vector)-OLE based on DDLog

Public Key PCF: Protocol Flow



Conclusion

- Share conversion for Paillier:

  Locally convert multiplicative shares of $(1+N)^x$ into additive shares of x

- Homomorphic secret sharing for branching programs
  - Negligible error, large plaintexts
- Pseudorandom correlation functions
  - Produce arbitrary quantity of VOLE or OT
  - Based on oblivious ciphertext sampling
  - Public-key setup

Open problems

- Improve OT efficiency
  - $O(\lambda)$ exponentiations
- Remove CRS N' from public-key setup
- More correlations: OLE from Paillier?
- Public-key PCFs from other assumptions (LPN?)
- Beyond two parties?