Weighted Cryptography with Weight-Independent Complexity

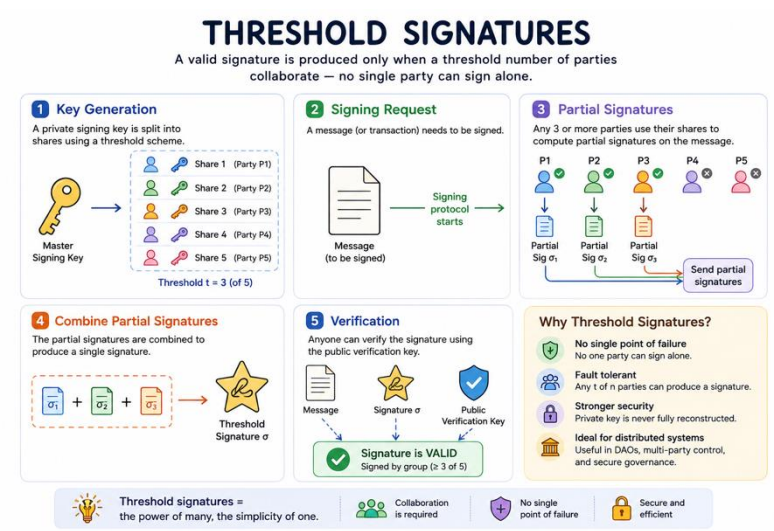
Aarushi Goel
Rutgers University

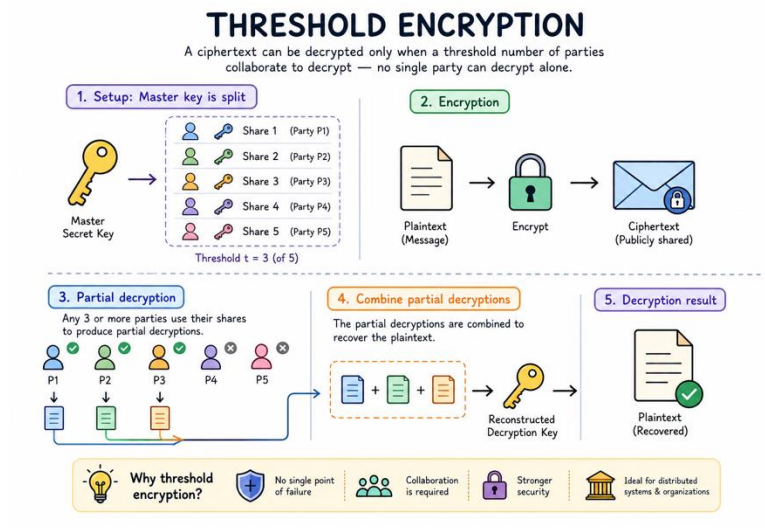
Swagata Sasmal
Indian Statistical Institute

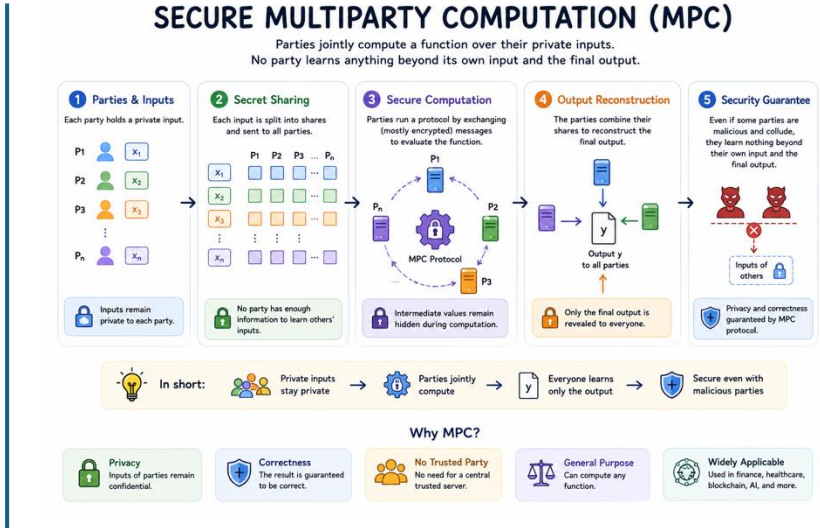
Mingyuan Wang
NYU Shanghai

Cryptography 10 Years Later: Reunion | July 16; 2026

Threshold Cryptosystems





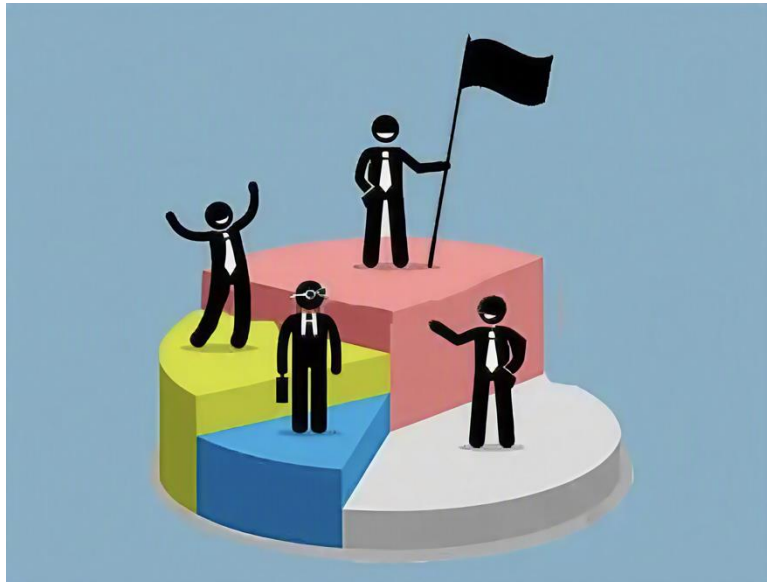


Trust Assumption: At most t-out-of-n corruptions

Images generated by ChatGPT

Weighted Threshold Cryptosystems

In many applications, parties are naturally asymmetric: stake, reputation, computational resources



Parties are assigned weights $w_1, \ldots, w_n$
Adversary can corrupt parties with cumulative weight $\leq t$

Could be as large as n^n [Mur71,Has94]

- Player Virtualization - Complexity grows with total weights.
- Better complexity?
  - [GJMSWZ23]: More efficient, but complexity still grows with weights.
  - [Yao89]: Non-black-box use of crypto.
  - [BW06]: Super-polynomial in the number of parties.
  - [MRVWZ21,...]: SNARK based approaches for weighted threshold signatures.

Question?

Can we design weighted cryptosystems with weight-independent complexity, while only relying on black-box use of cryptography?

Computational weighted secret sharing scheme with weight-independent complexity

(Special-form of linear homomorphism)

- Weighted MPC with Guaranteed output delivery
  (Trusted Setup Model)
- Semi-honest weighted threshold Schnorr signatures
- Semi-honest weighted threshold encryption
  (Two-round decryption)

Weighted Secret Sharing

(Prior Works on Weight-Independent Complexity)

Information theoretic Setting:

- [BW06] have share size super-polynomial in n, i.e., $n^{\Theta(\log n)}$

Unclear if we can do better than this in the information-theoretic setting?

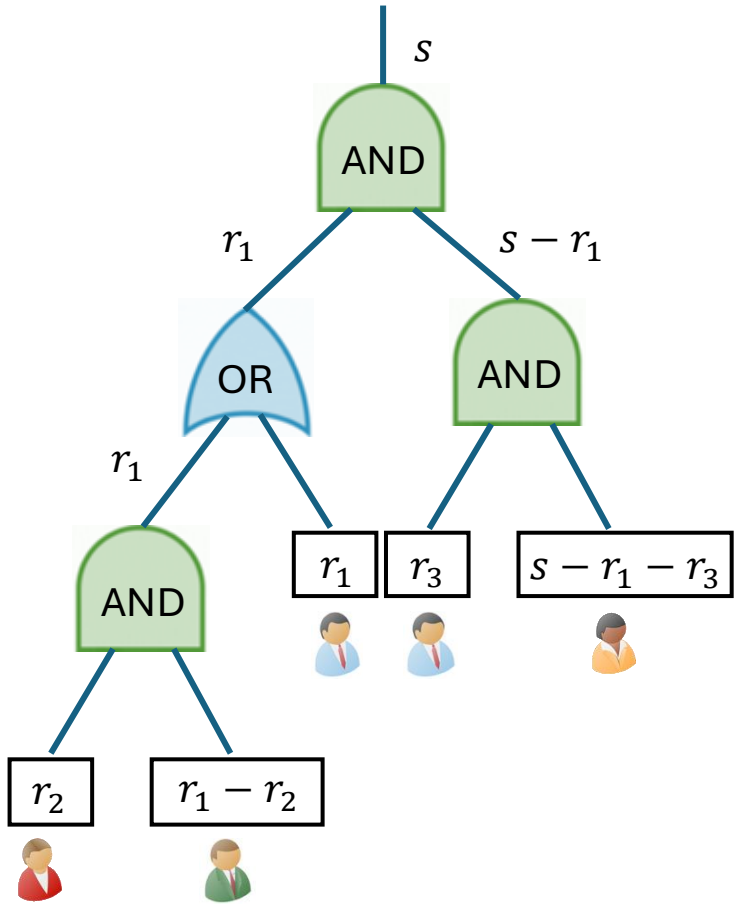
Computational Setting:

- [Yao89] is based on OWFs and has share size polynomial in n.
- No linear homomorphism.
- Using it to design weighted cryptosystems requires making non-black-box use of OWFs.

Making this scheme linearly homomorphic.

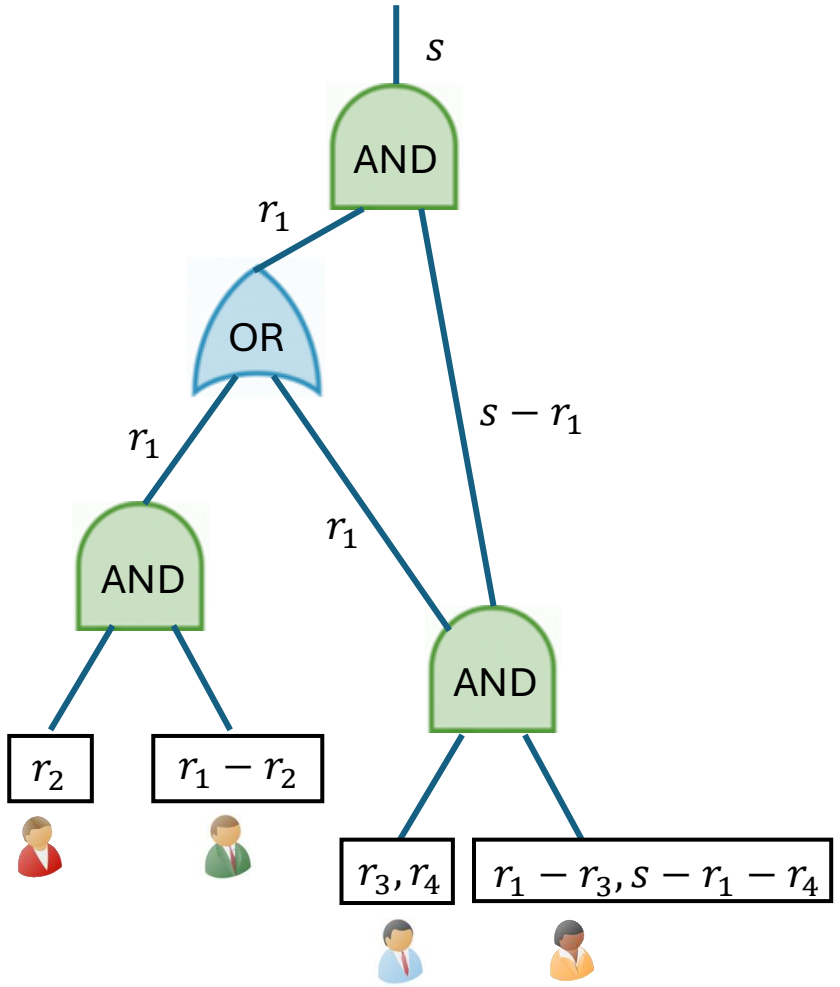
Our Techniques

[BL90] Secret Sharing Scheme



An information-theoretic secret sharing for access structures that can be efficiently represented as monotone Boolean formulas.

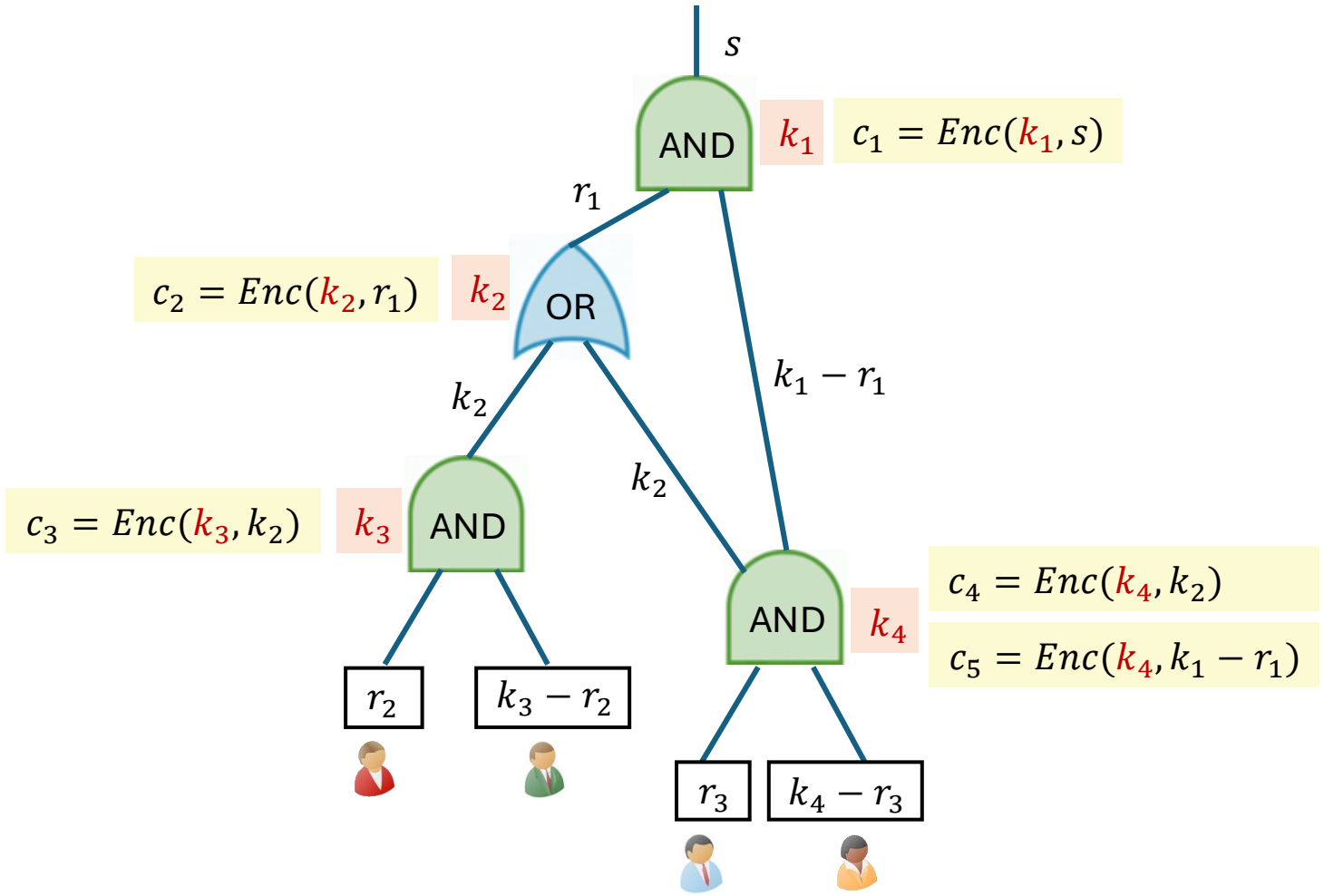
[BL90] Secret Sharing Scheme



When used for monotone Boolean circuits, share sizes grow exponentially with the depth of circuit.

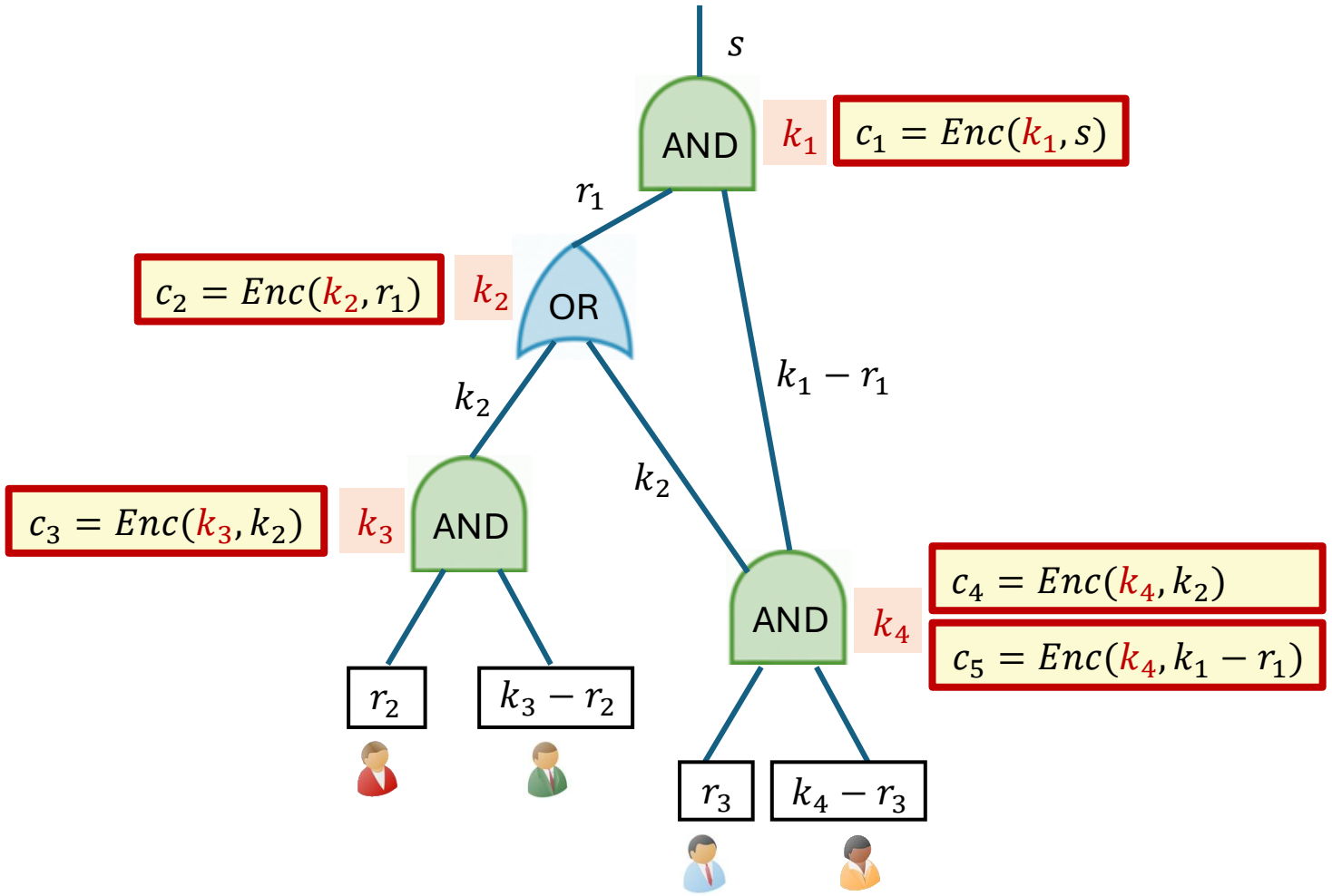
[BW06] showed how general weighted threshold access structures can be represented using a monotone Boolean circuit of depth $O(\log n)$. Hence, share sizes are super-polynomial in n.

[Yao89] Secret Sharing Scheme



Computational version of [BL90] for any access structure that can be represented as a monotone Boolean circuit.

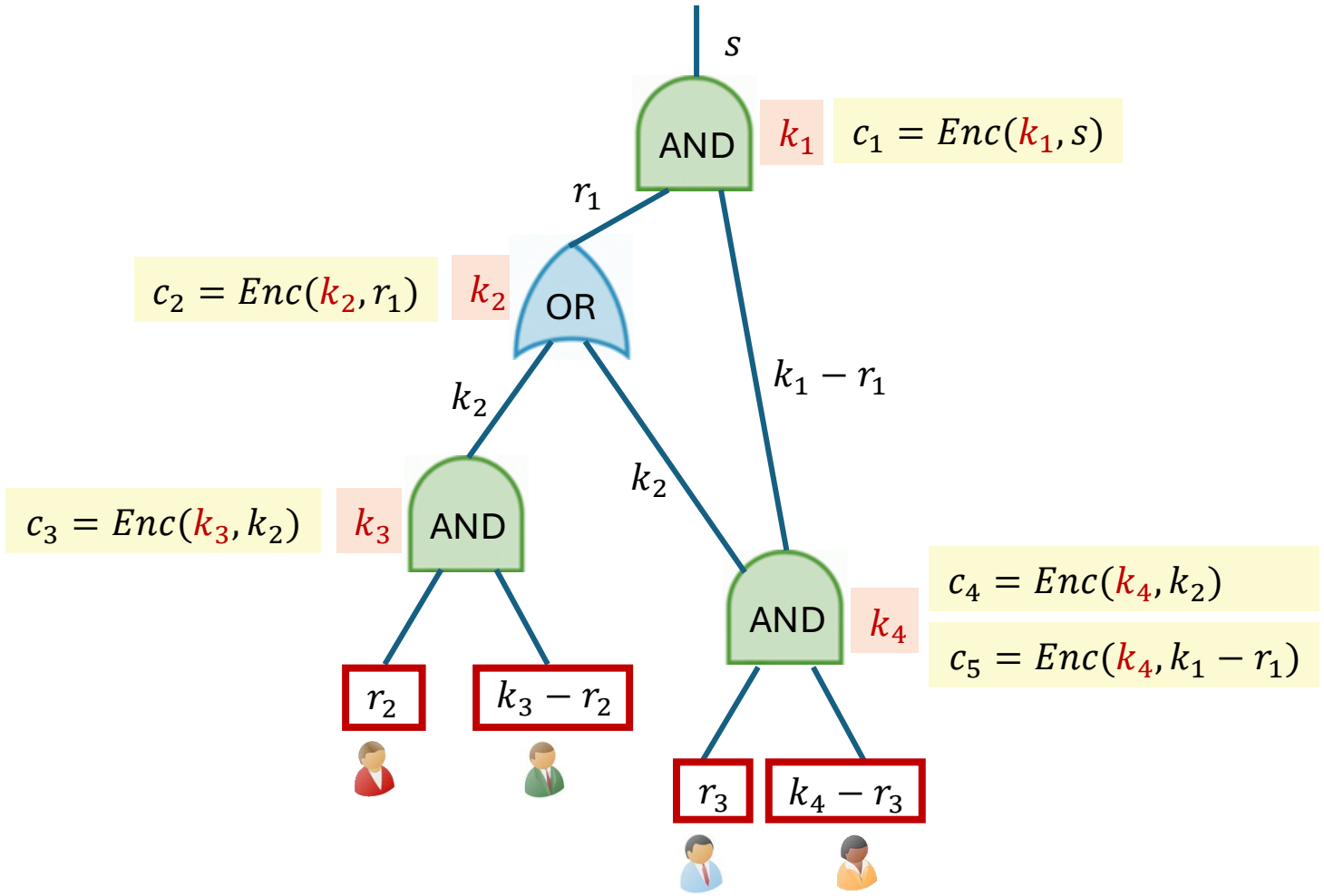
[Yao89] Secret Sharing Scheme



Computational version of [BL90] for any access structure that can be represented as a monotone Boolean circuit.

These ciphertexts form the public component of the shares.

[Yao89] Secret Sharing Scheme



Computational version of [BL90] for any access structure that can be represented as a monotone Boolean circuit.

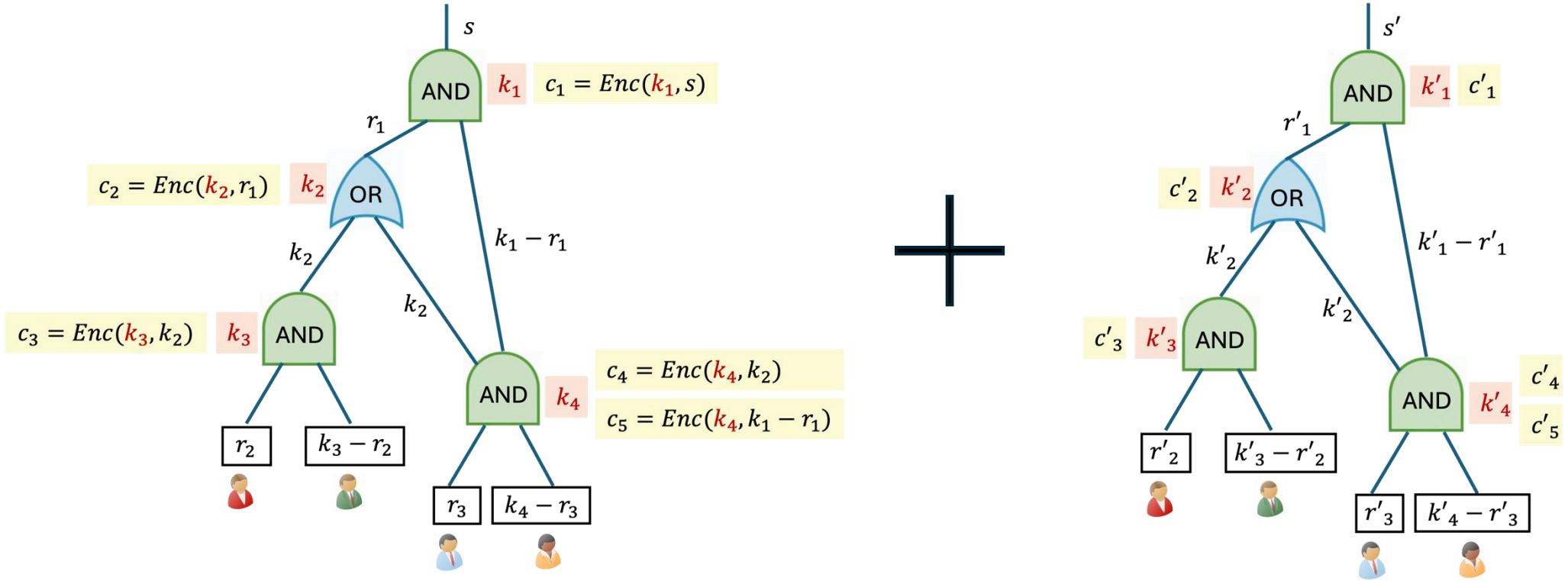
These ciphertexts form the public component of the shares.

These values assigned to input wires correspond to the associated party's private component of the share.

Share sizes grow with the number of wires in the circuit.

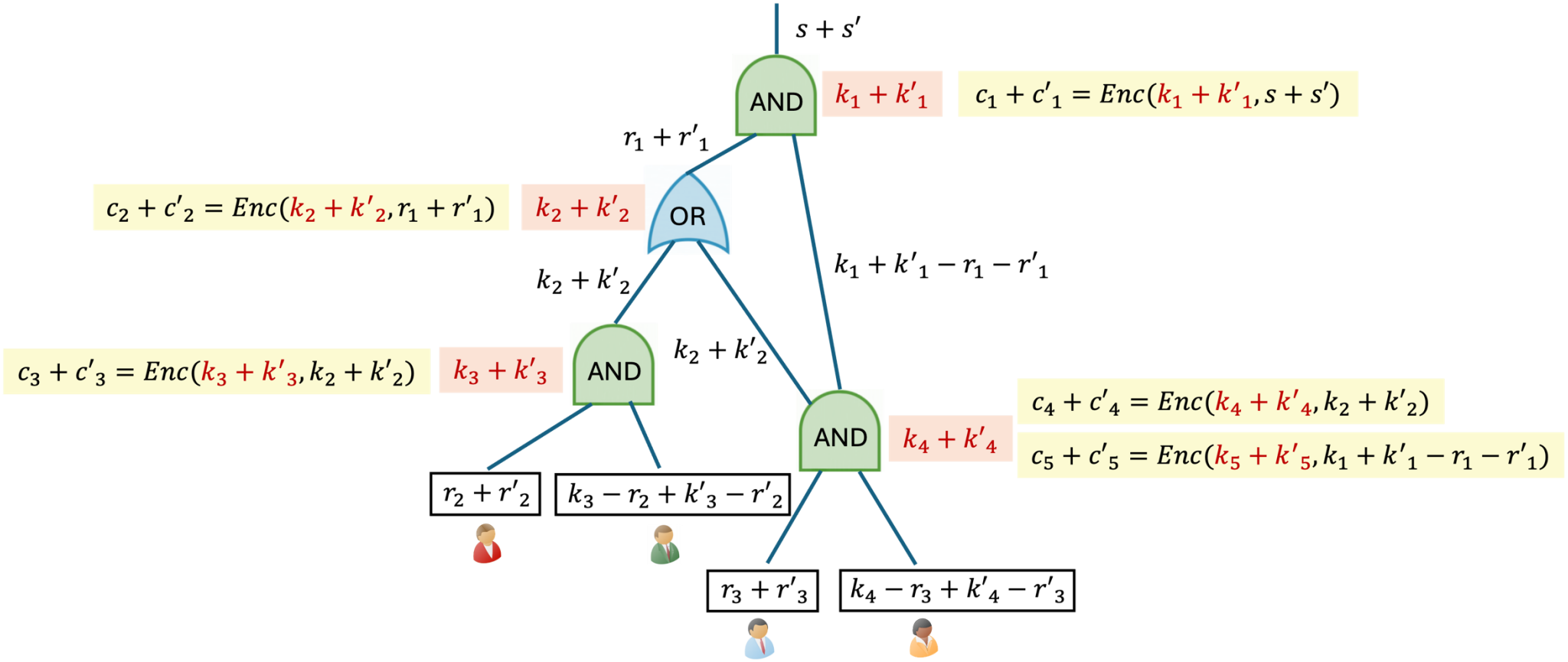
Making [Yao89] Linearly Homomorphic

If Enc was linearly homomorphic over both the message space and the key space



Making [Yao89] Linearly Homomorphic

If Enc was linearly homomorphic over both the message space and the key space



Are we done?

Security Issue: *

- Given $c = Enc(sk, m)$, $c' = Enc(sk', m')$ and $(sk + sk')$
- We want the parties to only learn $(m + m')$.
- Generic LHE schemes do not guarantee privacy of m and m' in this setting.

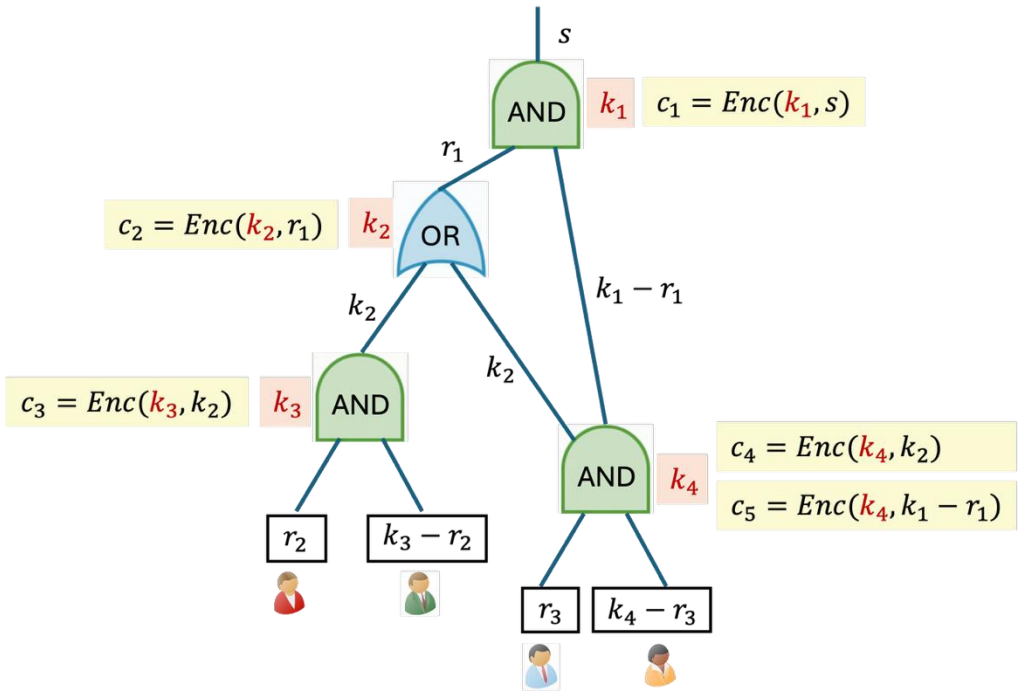
Correctness Issue:

- Key and message space should be the same

- Reminiscent of issues encountered in recent works on Arithmetic Garbling.
- [BLLL23] propose LHE schemes based on LWE and DCR with the above security guarantee, where the key space is $\mathbb{Z}$.

*We want something stronger, more on that later

Addressing the Correctness Issue



- In [BLLL23], keys are bounded integers drawn from $\mathbb{Z}$.
- Additive secret sharing over $\mathbb{Z}$ is not defined.

<u>Our Idea:</u>

- Let message space be $\mathbb{Z}_p$. Compute additive shares of the keys over $\mathbb{Z}_p$.
- The bound on keys should be independent of message space.
- Let max be the bound on the keys and B be the bound on the sum of coefficients used for linearly combining any set of ciphertexts. We need to ensure $p > B.max$.

Addressing the Security Issue

Security Requirement in [BLLL23]:

- Given $a, c = Enc(sk, m),\ c' = Enc(sk', m')\ and\ (a.sk + sk')$
- The parties should only learn $(a.m + m')$.
- Captured by requiring existence of a simulator, that can simulate $c, c', (a.sk + sk')$ given $(a.m + m')$.

How do they achieve this?

Adding some noise to the key and ciphertexts

- Encryption with two-modes: normal mode, smudged mode
- Cannot guarantee the above when combining two ciphertexts encrypted in the normal mode.
- For their application, it suffices to have c in normal mode and c' in smudged mode.
- The coefficient corresponding to c' can only be 1.

Addressing the Security Issue

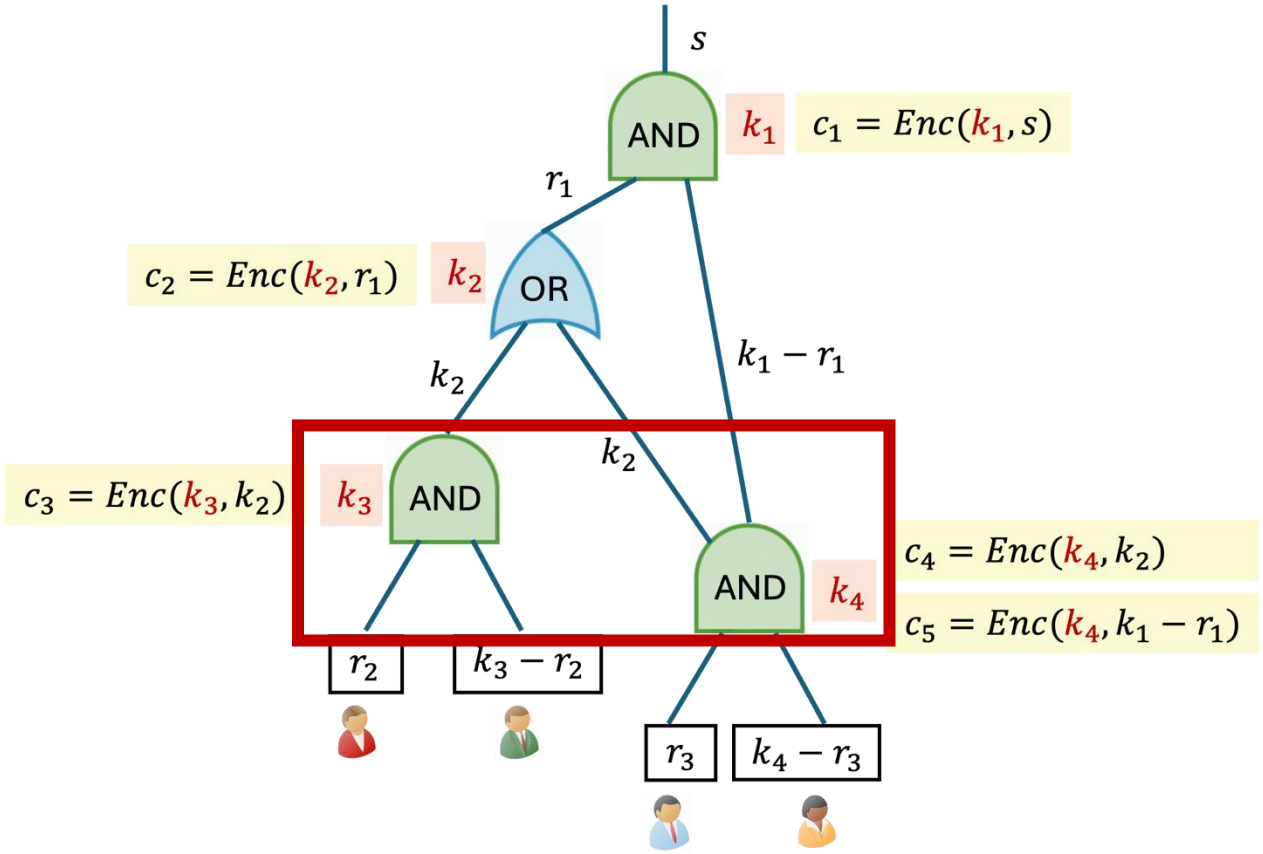
Our Security Requirement:

- Batch encryption.
- Adaptive simulation. For a fixed $c = Enc(sk, m)$ encrypted in the normal mode, the simulator can simulate smudged c', $(a.sk + sk')$, given $(a'.m + m')$.
- Ciphertexts encrypted in the normal mode can be used in any number of linear combinations, as long as each linear combination has one "fresh" smudged ciphertext.

We show that both LWE and DCR based constructions of LHE in [BLLL23] satisfy these stronger requirements.

Weighted secret sharing with two modes

Adding Robust to our Secret Sharing Scheme

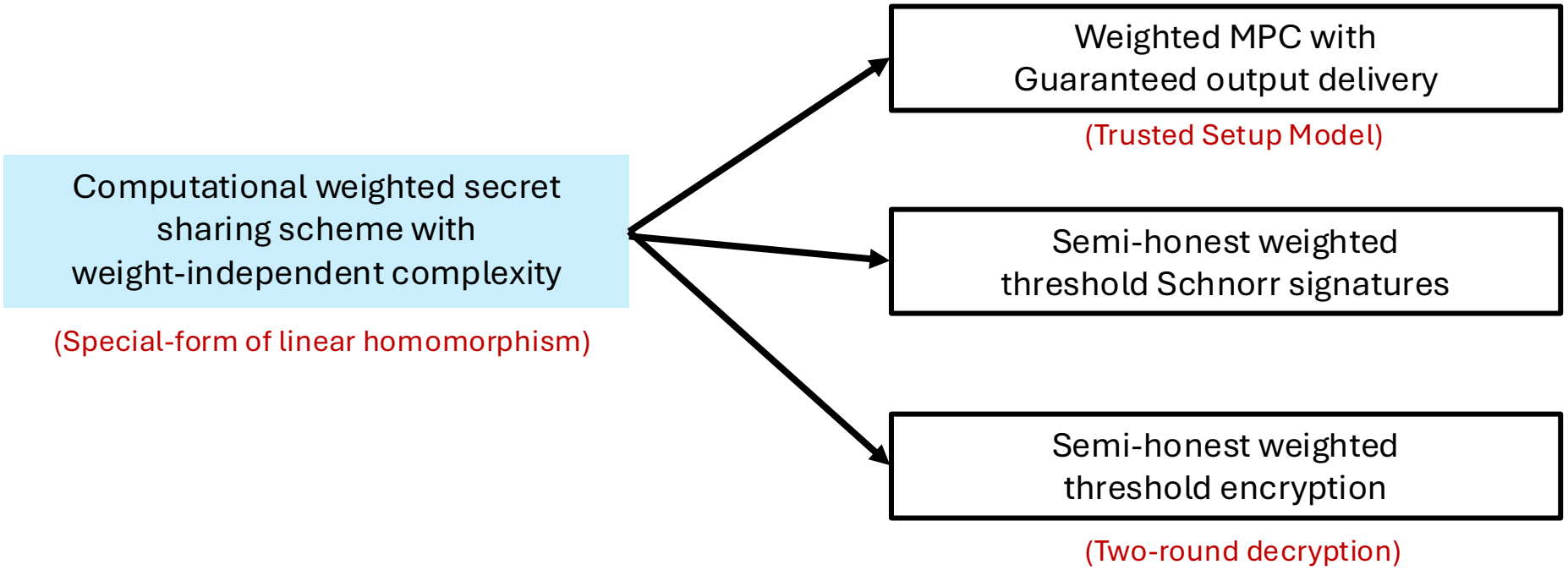


For our application to MPC with guaranteed output delivery, we need the secret shares to be robust.

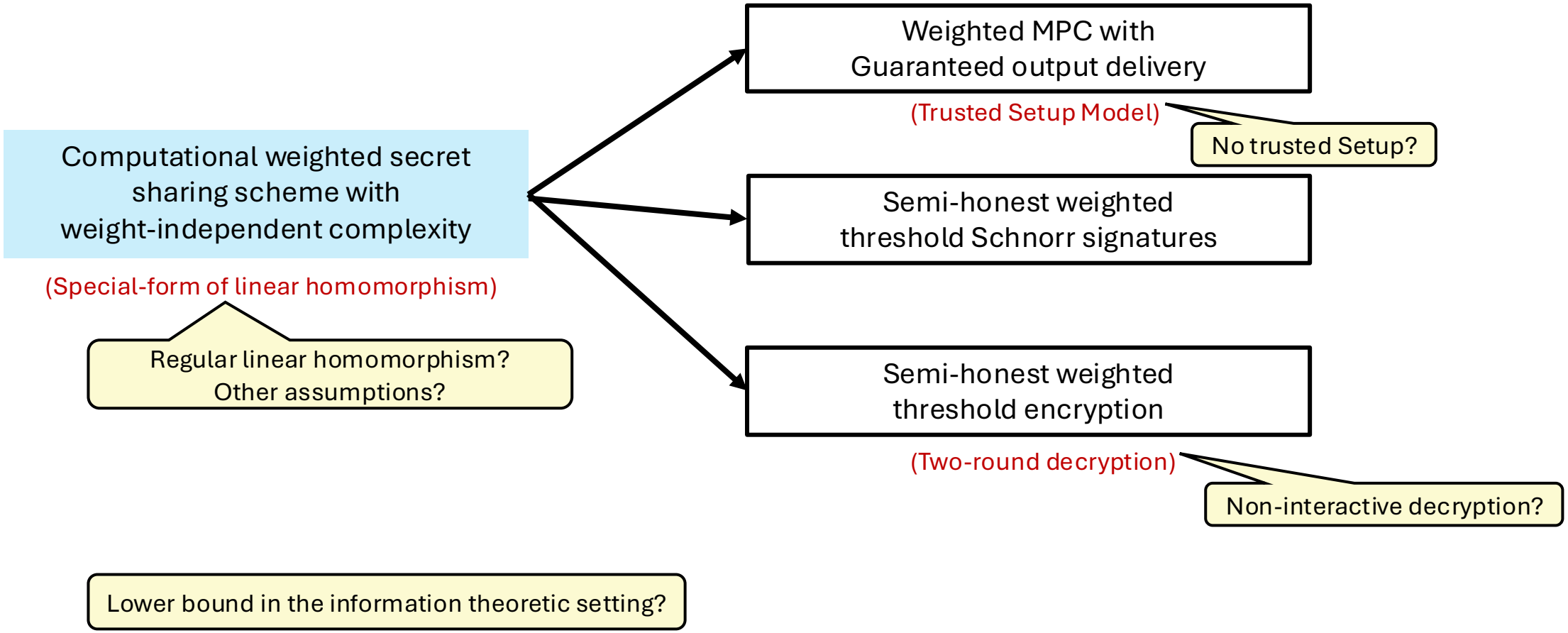
Our Idea: Also compute public commitments to the keys associated with the input gates.

Summary

- Robust weighted secret sharing scheme with weight-independent complexity and a special form of linear homomorphism.
- Instantiations based on LWE and DCR.
- Interesting connection with techniques used in arithmetic garbling.
- Simple modification of Yao's scheme.
- Also works for general access structures that can be efficiently represented as monotone Boolean functions.



Open Problems



Thanks!