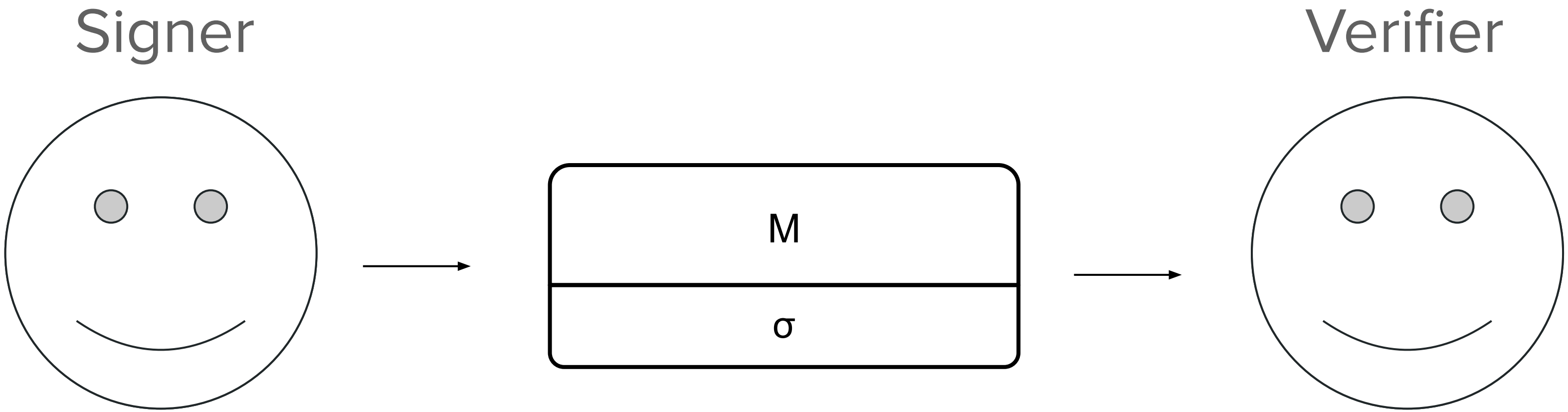
Scalable Distributed Key Generation

Robert Chen
Mentored by Alin Tomescu

2019 PRIMES Conference
5/19/19

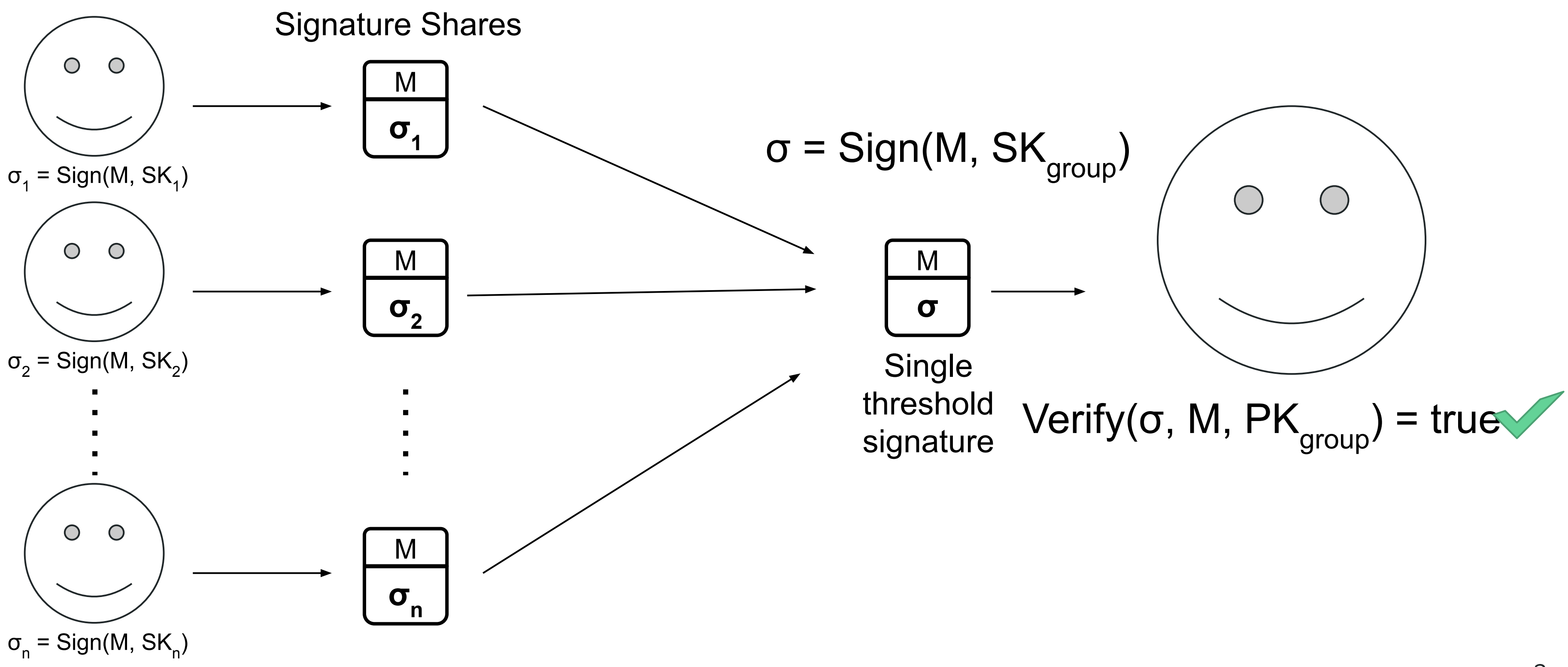
Background: Digital Signatures



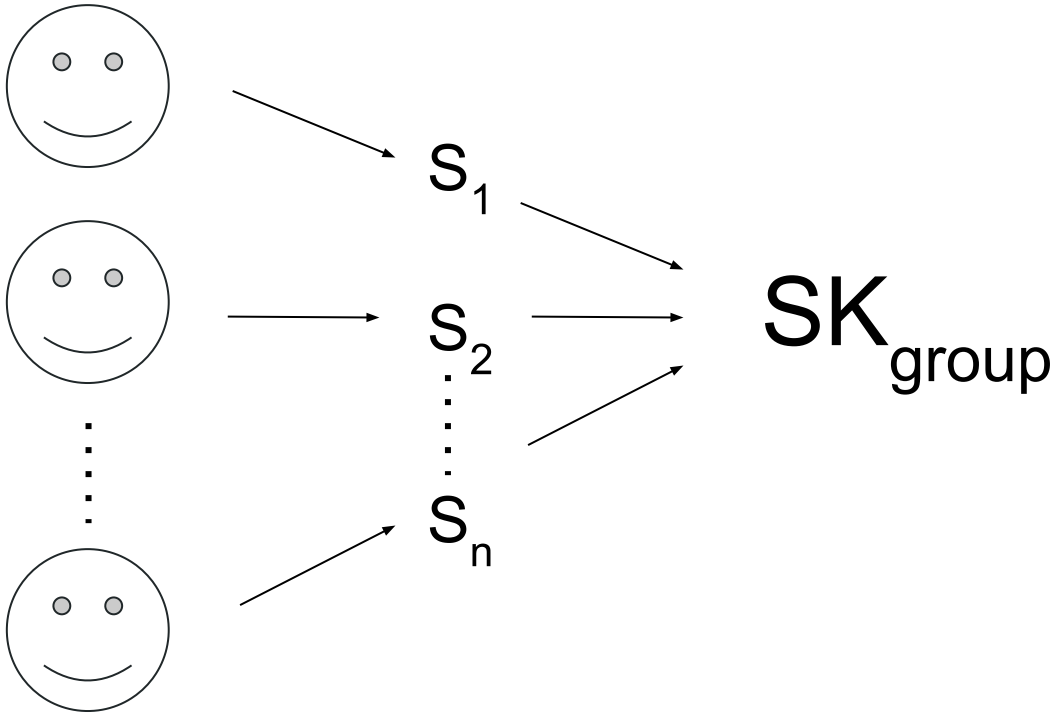
M = “Hello.”
$\sigma = \text{Sign}(M, SK_{\text{Signer}})$

$\text{Verify}(\sigma, M, PK_{\text{Signer}}) = \text{true}$ ✓

Background: Threshold Signatures



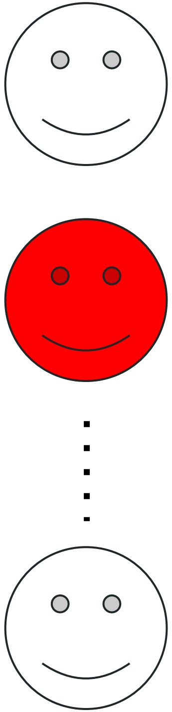
Distributed Key Generation (DKG)



PK_{group}

SK_{group}

Distributed Key Generation (DKG)



malicious < t

PK_{group}

SK_{group}

Distributed Key Generation (DKG): Applications

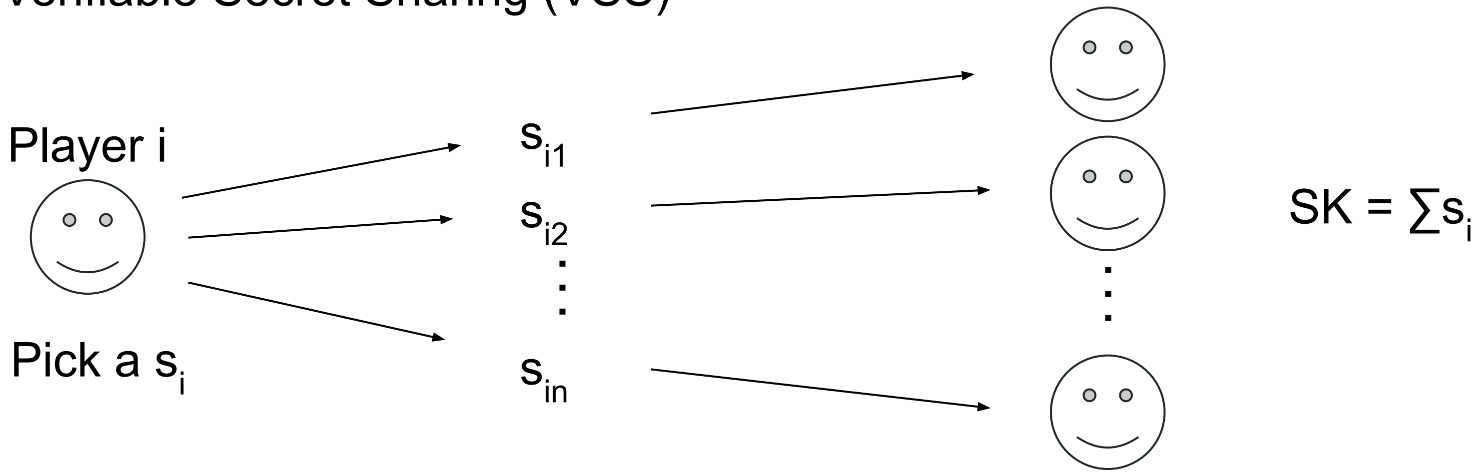
- Generating secret keys for threshold signature schemes
- Generating random nonces for Schnorr threshold signatures
- Random beacons
- Proactive Secret Sharing

Contributions

DKG scheme	**Per-player bandwidth**	**Per-player computation time (deal + verify)**
Feldman DKG	O(nt)	O(nt)
Kate DKG	O(n)	O(nt)
AMT DKG	O(n log(n))	O(n log(n))

DKG Outline

- Each player **i** acts as a **dealer** and "shares" a secret $\mathbf{s_i}$ with all other players via Verifiable Secret Sharing (VSS)



- **Our contribution:** We show how to do VSS in O(n log n) time rather than O(nt) time, which helps scale DKG

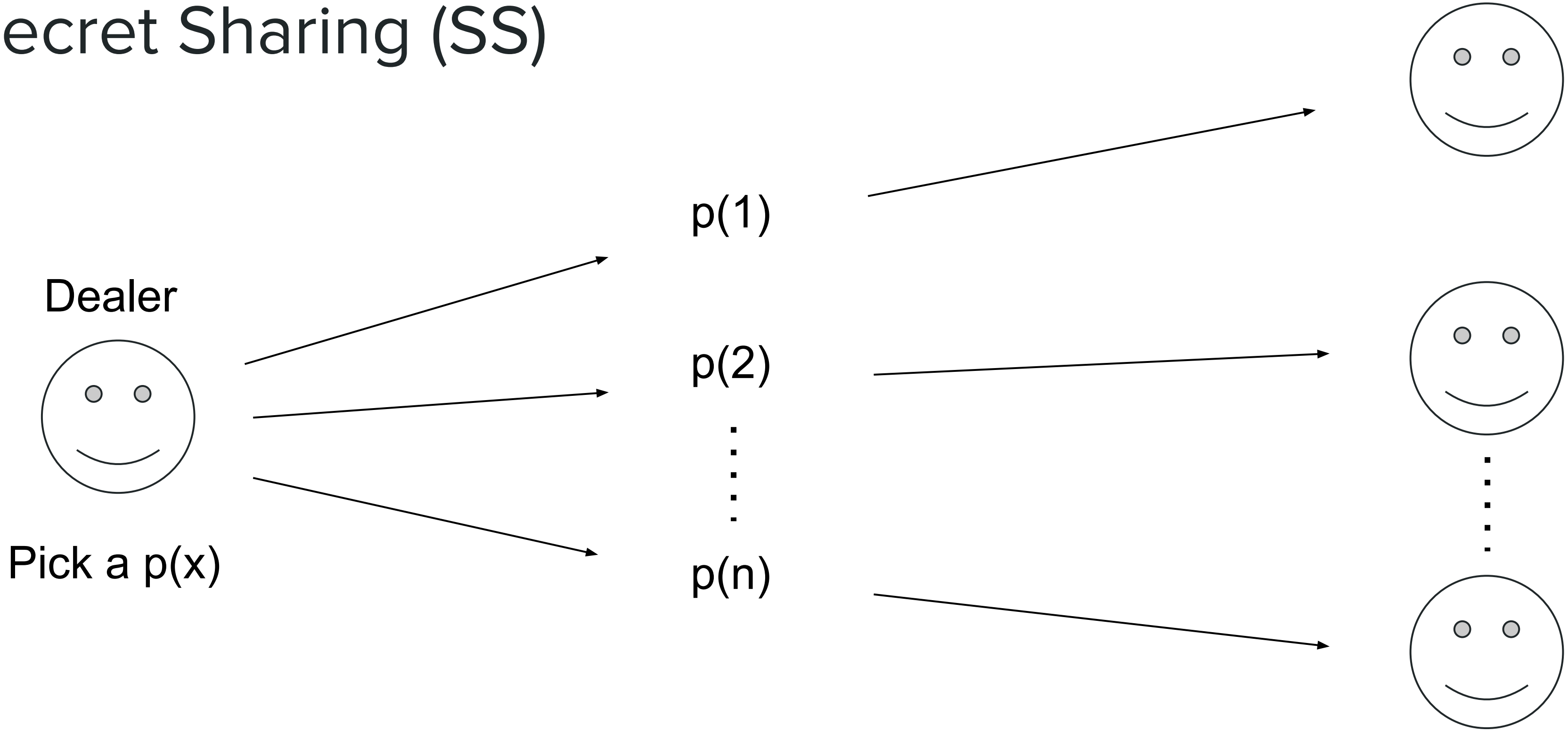
Secret Sharing (SS)

- Dealer picks a secret **s** and “shares” it with all other players such that **t** out of **n** can reconstruct it

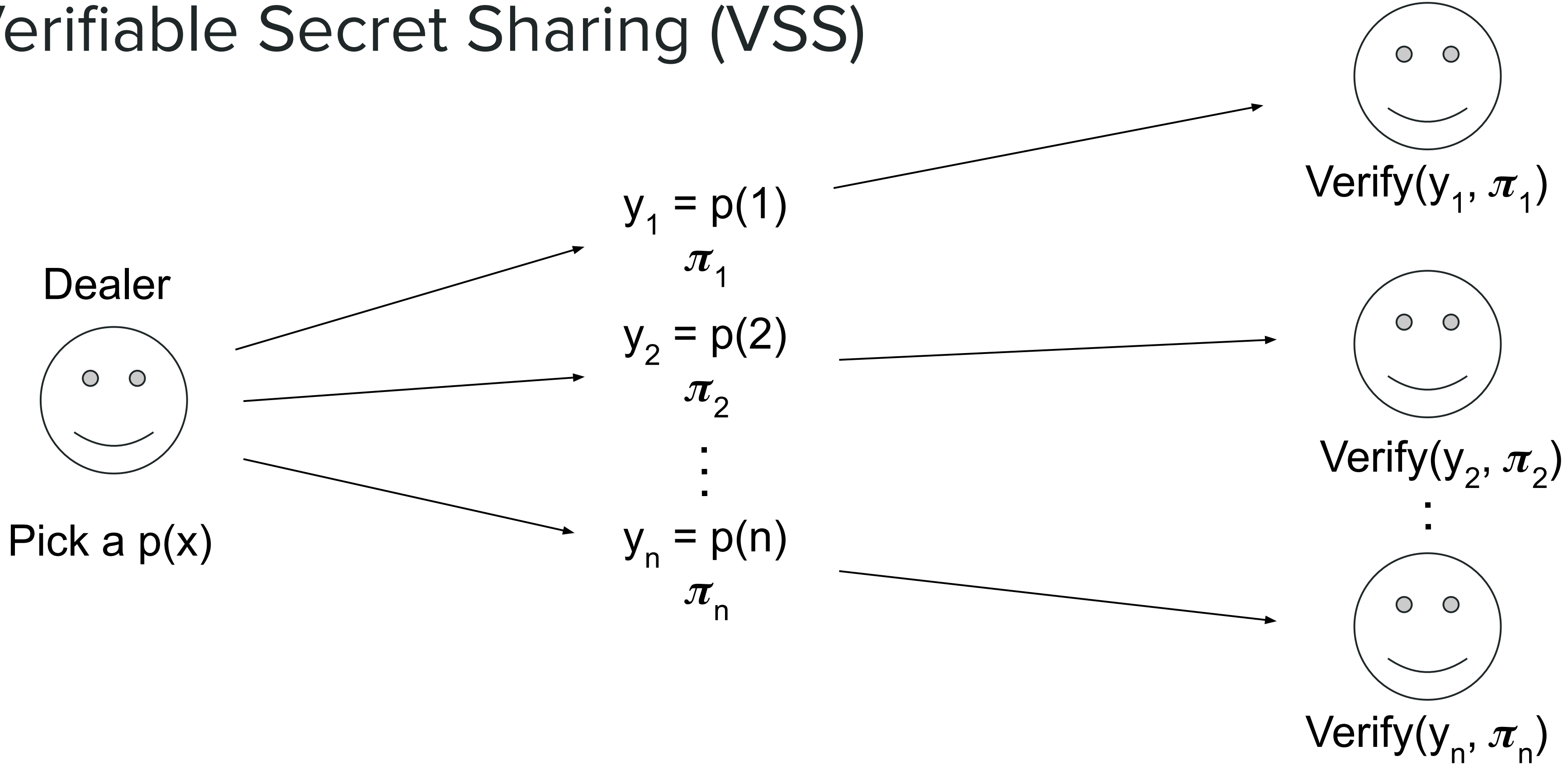
$$p(x) = c_0 + c_1x + c_2x^2 + \dots + c_{t-1}x^{t-1}$$

$$s = c_0$$

Secret Sharing (SS)



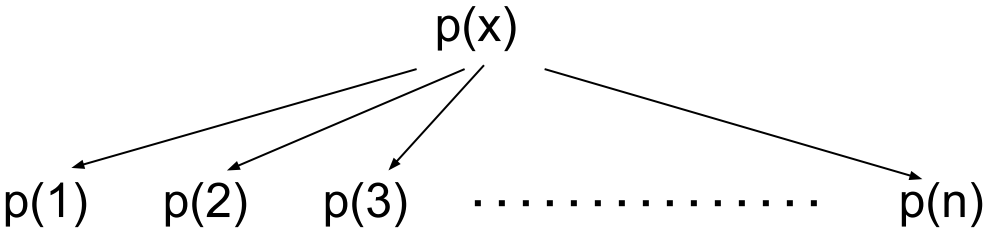
Verifiable Secret Sharing (VSS)



Polynomial Commitments

- Polynomial commitment to p(x) is $\mathbf{g^{p(\alpha)}}$
- How do we provide **evaluation proofs** $\boldsymbol{\pi}_{\mathbf{i}}$ that a value **p(i) = y** and verify proof against commitment $g^{p(\alpha)}$?
- Polynomial remainder theorem:
  p(x) - y = q(x)(x-i) if and only if p(i) = y
- **Proof: Commitment to quotient $\mathbf{g^{q(\alpha)}}$**
- Verify? Check using magic! (bilinear pairings)
- Dealer: O(nt) time to compute evaluation proofs

Solution: Multipoint Evaluation



O(n $\log^2$n) rather than O(nt)

- Need to build evaluation proofs that p(i) = y
- Key idea: multipoint evaluation is just a tree of polynomials. We commit to some of them and obtain proofs too.

Multipoint Evaluation

$O(n \log^2 n)$

$p \,/\, (x-1)(x-2)(x-3)(x-4)$

$q_{1,4}, r_{1,4}$

$r_{1,4} \,/\, (x-1)(x-2)$

$q_{1,2}, r_{1,2}$

$r_{1,4} \,/\, (x-3)(x-4)$

$q_{3,4}, r_{3,4}$

$r_{1,2} \,/\, (x-1)$

q_1, r_1

$r_{1,2} \,/\, (x-2)$

q_2, r_2

$r_{3,4} \,/\, (x-3)$

q_3, r_3

$r_{3,4} \,/\, (x-4)$

q_4, r_4

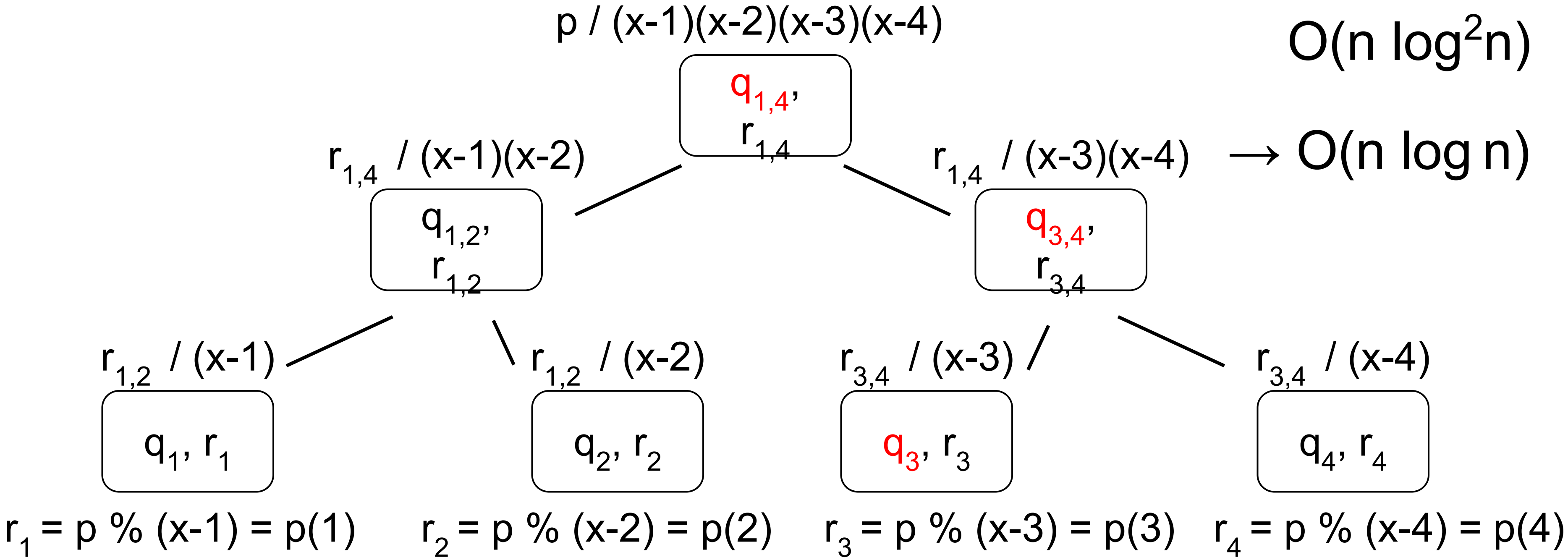
$r_1 = p \,\%\, (x-1) = p(1)$

$r_2 = p \,\%\, (x-2) = p(2)$

$r_3 = p \,\%\, (x-3) = p(3)$

$r_4 = p \,\%\, (x-4) = p(4)$

Authenticated Multipoint Evaluation Trees (AMT)



$O(n \log^2 n)$

$\rightarrow O(n \log n)$

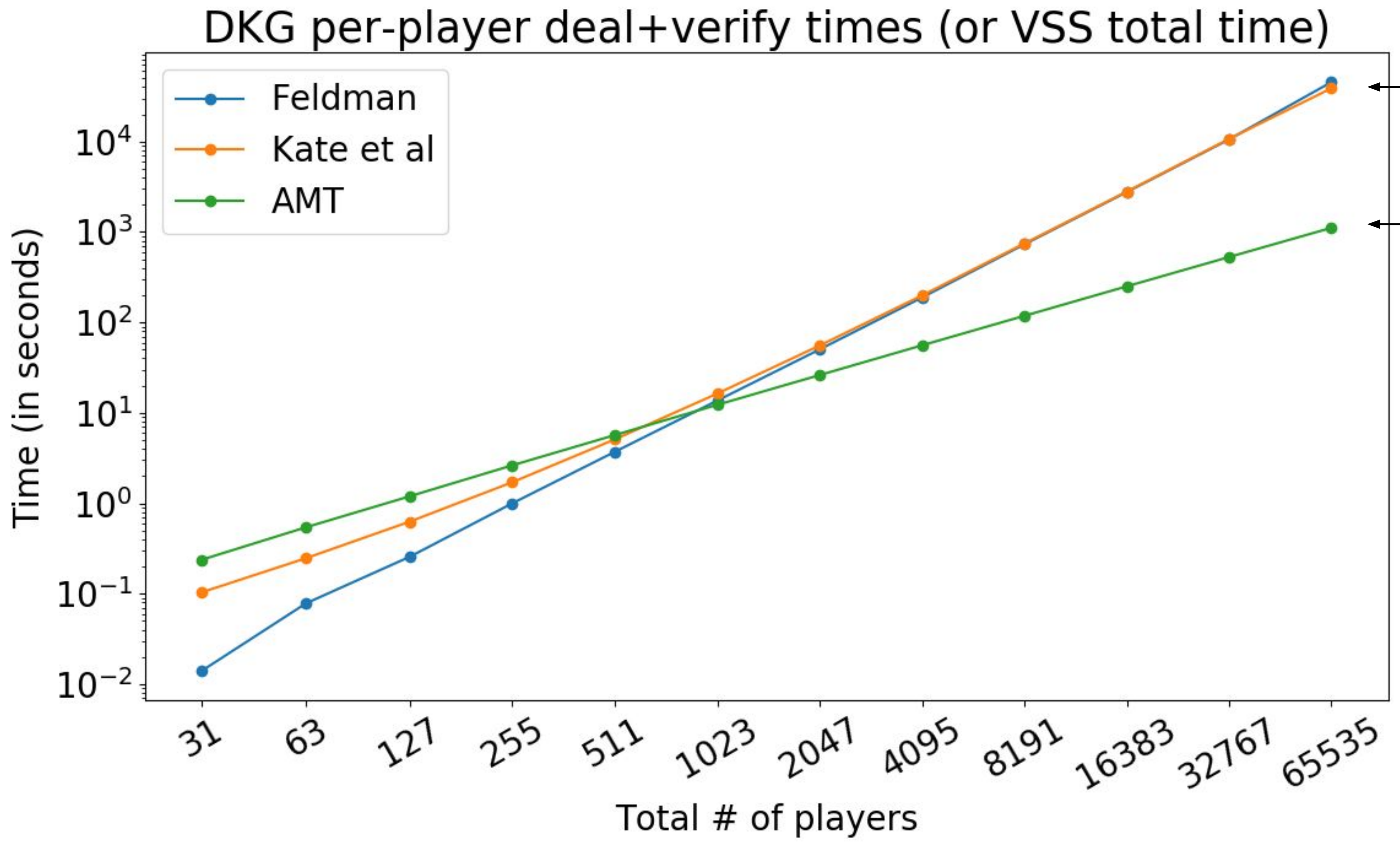
$\pi_3 = (g^{q_{1,4}(\alpha)}, g^{q_{3,4}(\alpha)}, g^{q_3(\alpha)})$

$p(x) = q_{1,4}(x-1)(x-2)(x-3)(x-4) + q_{3,4}(x-3)(x-4) + q_3(x-3) + p(3)$

Recap

- DKG - generate shared SK and PK, requires each player to perform a VSS
- VSS - pick polynomial p and send p(i) to each player i, needs to compute proofs that p(i) is valid using polynomial commitments
- Polynomial commitments - existing schemes like Kate take O(nt) to compute all proofs, AMT provides all proofs in $O(n \log^2 n)$ time
- Result: Faster DKG that scales to tens of thousands of players.

Results



10.78 hours

18.52 minutes

Acknowledgements

I would like to thank:

- My mentor, Alin Tomescu, for his support and guidance
- Srini Devadas, for coordinating CS-PRIMES
- My parents and family
- MIT-PRIMES program

Thank you!

Questions?