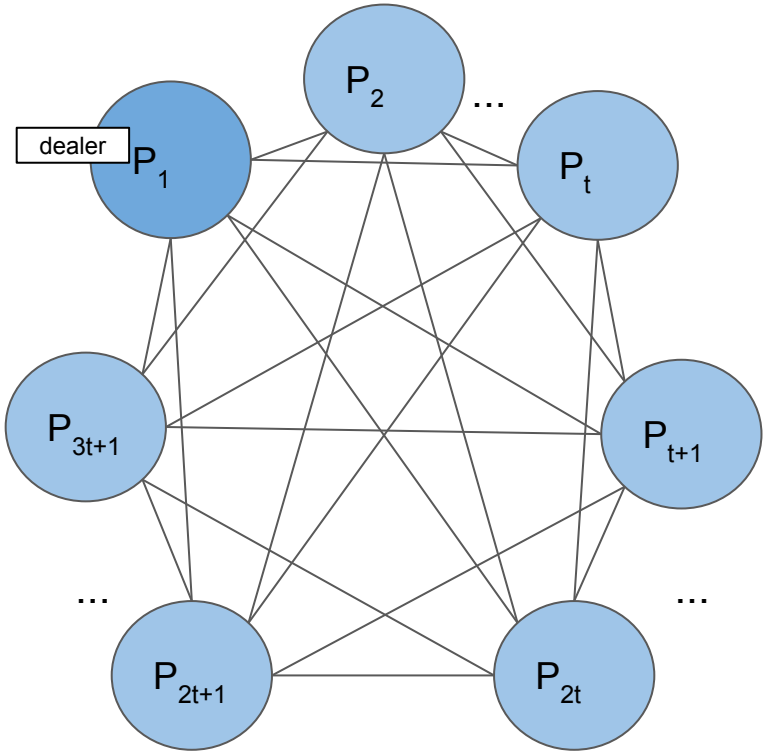
High-Threshold Asynchronous Verifiable Secret Sharing with Optimal Communication Complexity

Nicolas Alhaddad, Mayank Varia, and Haibin Zhang
Financial Crypto 2021

Broadcast

~~High-Threshold~~ Asynchronous ~~Verifiable Secret Sharing~~



Setting:

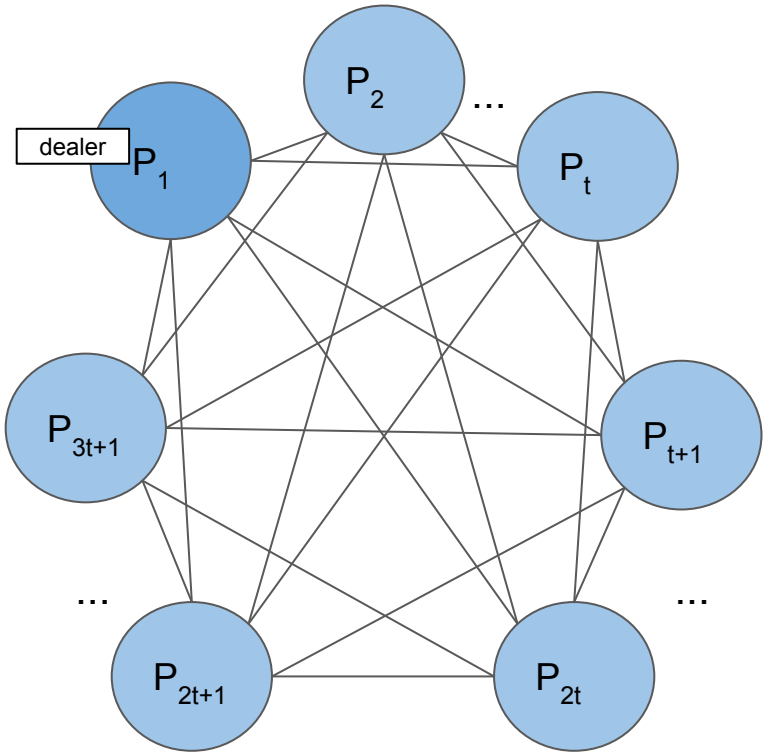
- T parties are malicious
- N = 3T + 1 total parties

Security goal

- Agreement over the broadcasted message

Note: In the following slides, we will only consider the case when n = 3t+1. However, everything would still work for any n >= 3t+1

~~High-Threshold~~ Asynchronous Verifiable Secret Sharing



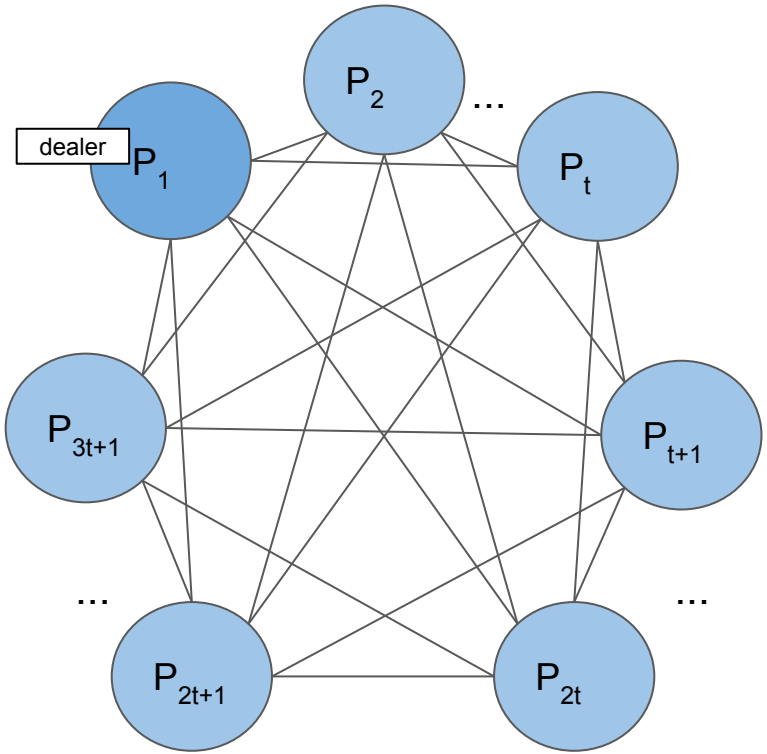
Setting:

- T parties are malicious
- N = 3T + 1 total parties
- P = T + 1 can reconstruct the secret

Security goal

- Agreement.
- Privacy.

High-Threshold Asynchronous Verifiable Secret Sharing



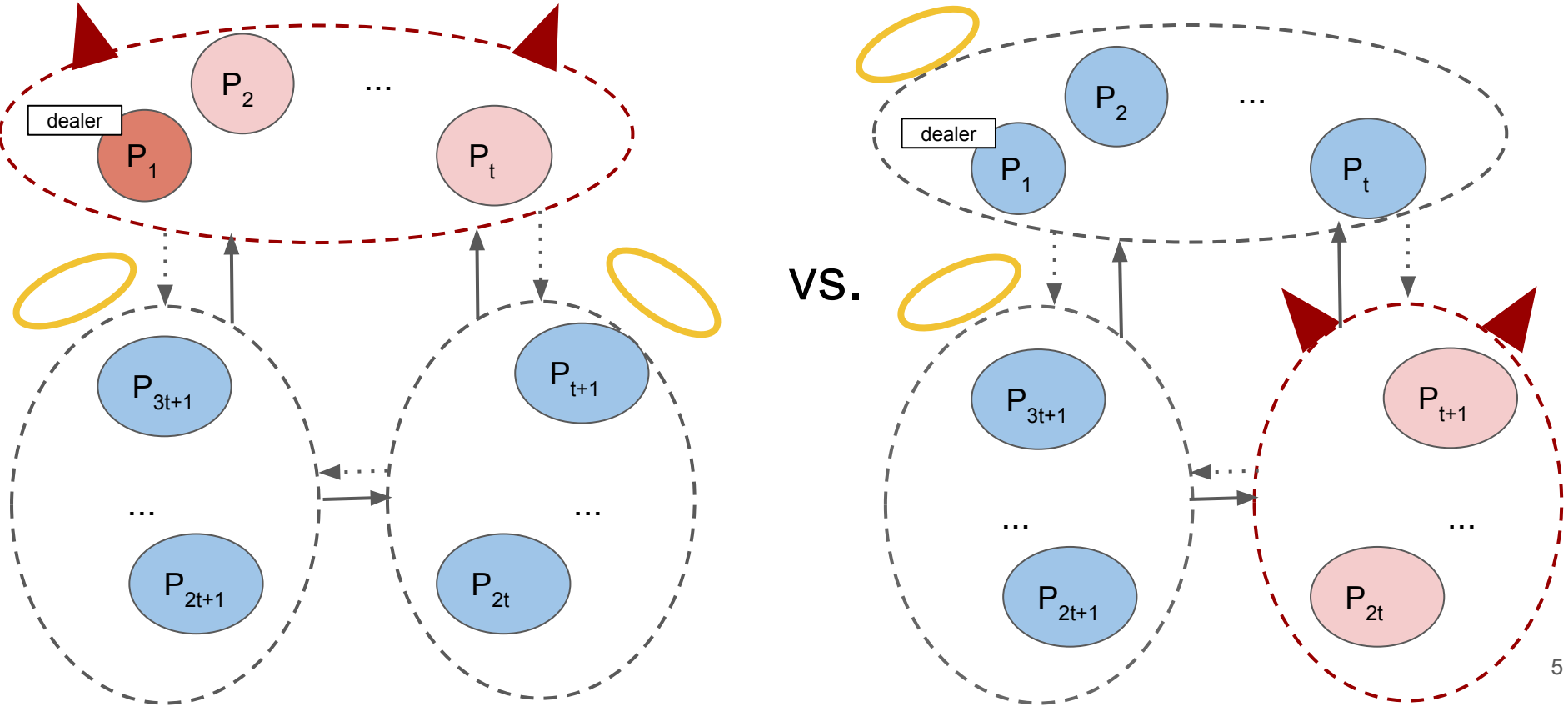
Setting:

- T parties are malicious
- N = 3T + 1 total parties
- P = N - T parties can reconstruct the secret

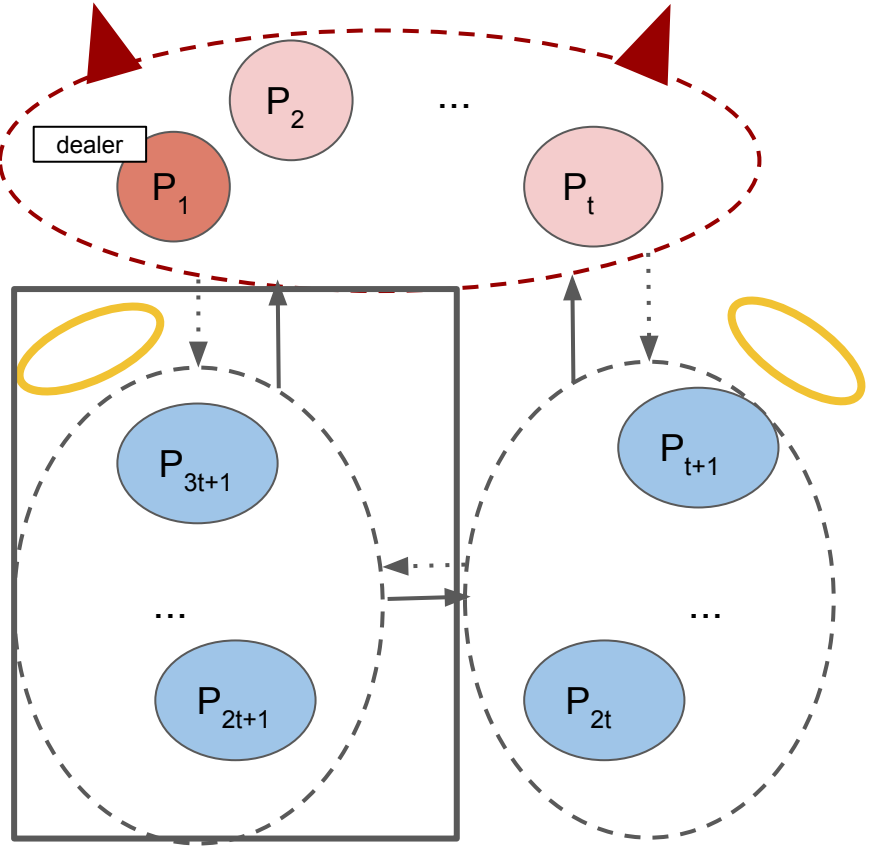
Security goal

- Agreement: any p honest parties should be able to reconstruct the same secret.
- Privacy: any p - 1 shares should not reveal anything about the secret.

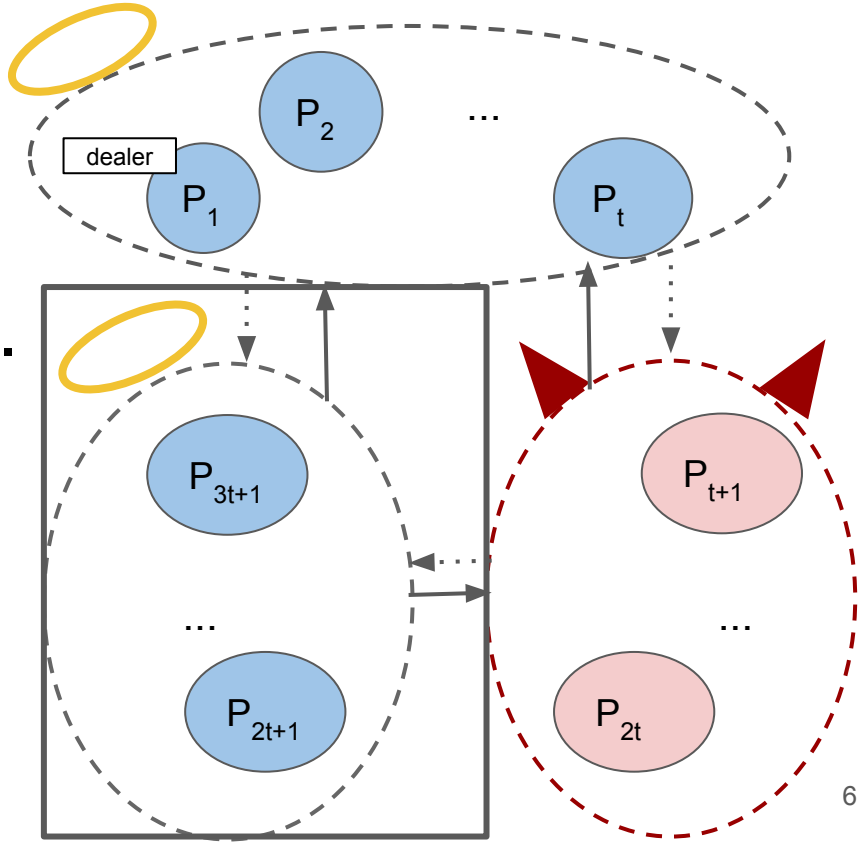
Why high threshold AVSS is challenging



Why high threshold AVSS is challenging



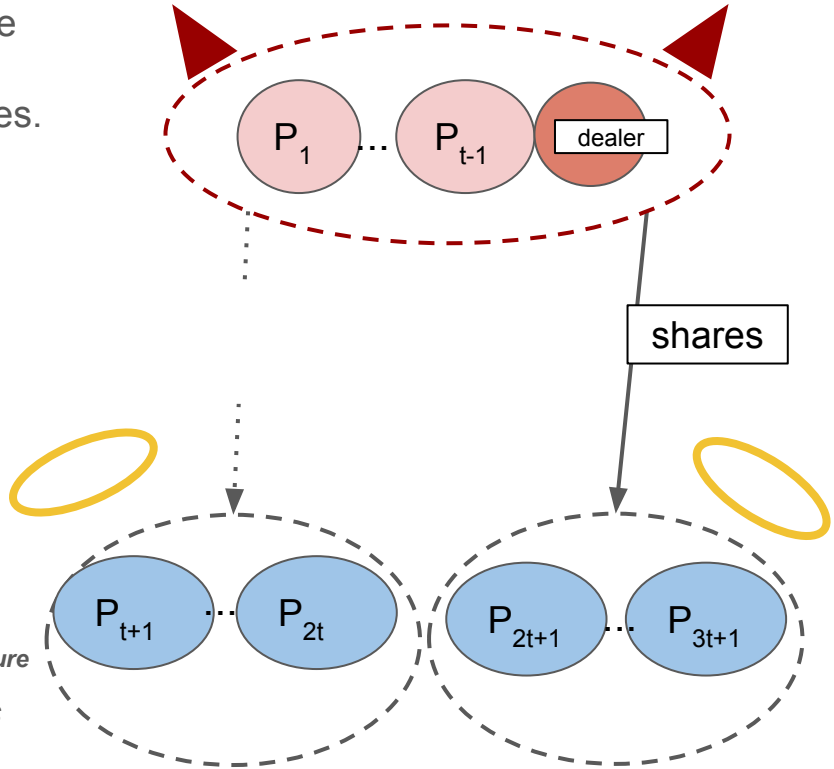
vs.



Related Work - Dual AVSS

Reliable Broadcast of a bivariate polynomial commitment

Previous work. ♣Bivariate polynomial of different degrees + digital signatures. The recovery polynomials was made of a degree t sharing, while the share polynomials are made of degree 2t.



a_{t0}	a_{t1}	...	a_{ti}	...	$a_{t(2t+1)}$
...	...	...	...	...	...
a_{i0}	a_{i1}	...	a_{ii}	...	$a_{i(2t+1)}$
...	...	...	...	...	...
a_{10}	a_{11}	...	a_{1i}	...	$a_{1(2t+1)}$
$a_{00}=s$	a_{10}	...	a_{0i}	...	$a_{0(2t+1)}$

Polynomial that can recover the secret s

♣ ***Asynchronous Distributed Key Generation for Computationally- Secure Randomness, Consensus, and Threshold Signatures.*** *ELEFTHERIOS KOKORIS-KOGIAS, DAHLIA MALKHI, ALEXANDER SPIEGELMAN. (2020)*

HAVEN

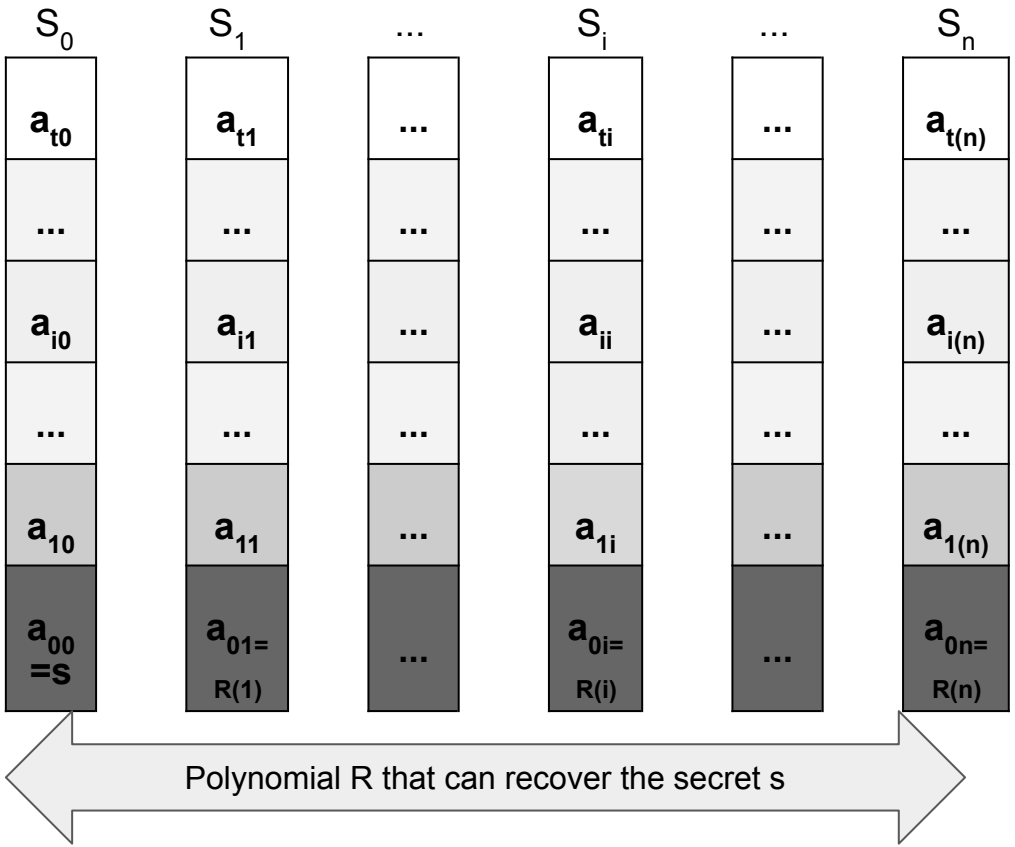
HAVEN is a customizable Dual AVSS that supports high thresholds of reconstruction. HAVEN bridges asynchronous reliable broadcast with secret sharing using additively homomorphic polynomial commitments. As a result, based on the polynomial commitment that is used with HAVEN we achieve different properties that outperform the best AVSS.

We include a comparison of HAVEN equipped with KZG commitments (option1) and Bullet proofs (option 2) with the state of the art AVSS.

Works	threshold		complexity				avoiding setup		crypto
	dual	high	message	comm.	amortized	rounds	no trust?	no PKI?	assumption
Cachin et al. [15]	✓	✗	$O(n^2)$	$O(\kappa n^3)$	$O(\kappa n^2)$	3	✓	✓	DL
Backes et al. [2]	✗	✗	$O(n^2)$	$O(\kappa n^2)$	$O(\kappa n^2)$	3	✗	✓	t-SDH
Kate et al. [25]	✗	✗	$O(n^2)$	$O(\kappa n^3)$	$O(\kappa n)$	> 4	✗	✗	t-SDH
Kokoris-Kogias et al. [28]	✓	✓	$O(n^2)$	$O(\kappa n^4)$	$O(\kappa n^3)$	4	✓	✗	DL
HAVEN option 1	✓	✓	$O(n^2)$	$O(\kappa n^2)$	$O(\kappa n)$	3	✗	✓	t-SDH
HAVEN option 2	✓	✓	$O(n^2)$	$\tilde{O}(\kappa n^2)$	$O(\kappa n)$	3	✓	✓	DL + ROM

Where k is the security parameter that reflects the size of the element in the group, and n is the total number of parties in the protocol.

Root commitment doesn't have to be a bivariate polynomial!



Claim:

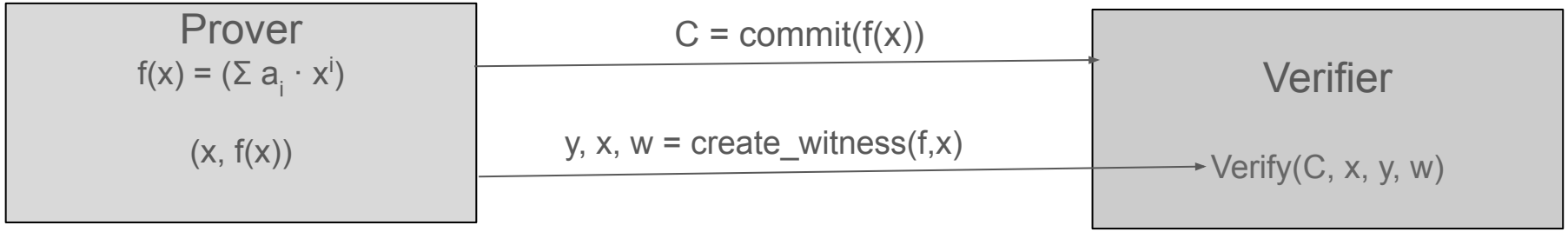
We can commit to this root commitment in O(1)

Problem:

Every party has to check in zero-knowledge that:

1. The share (column) polynomials are consistent with the recovery polynomial ($\forall i\ S_i(0) = R(i)$)
2. $\deg(R) <= p$
3. $\deg(S_i) <= t$

Polynomial commitments



Let g, h be elements of Z_p of order q such that $g^q = h^q = 1 \bmod p$

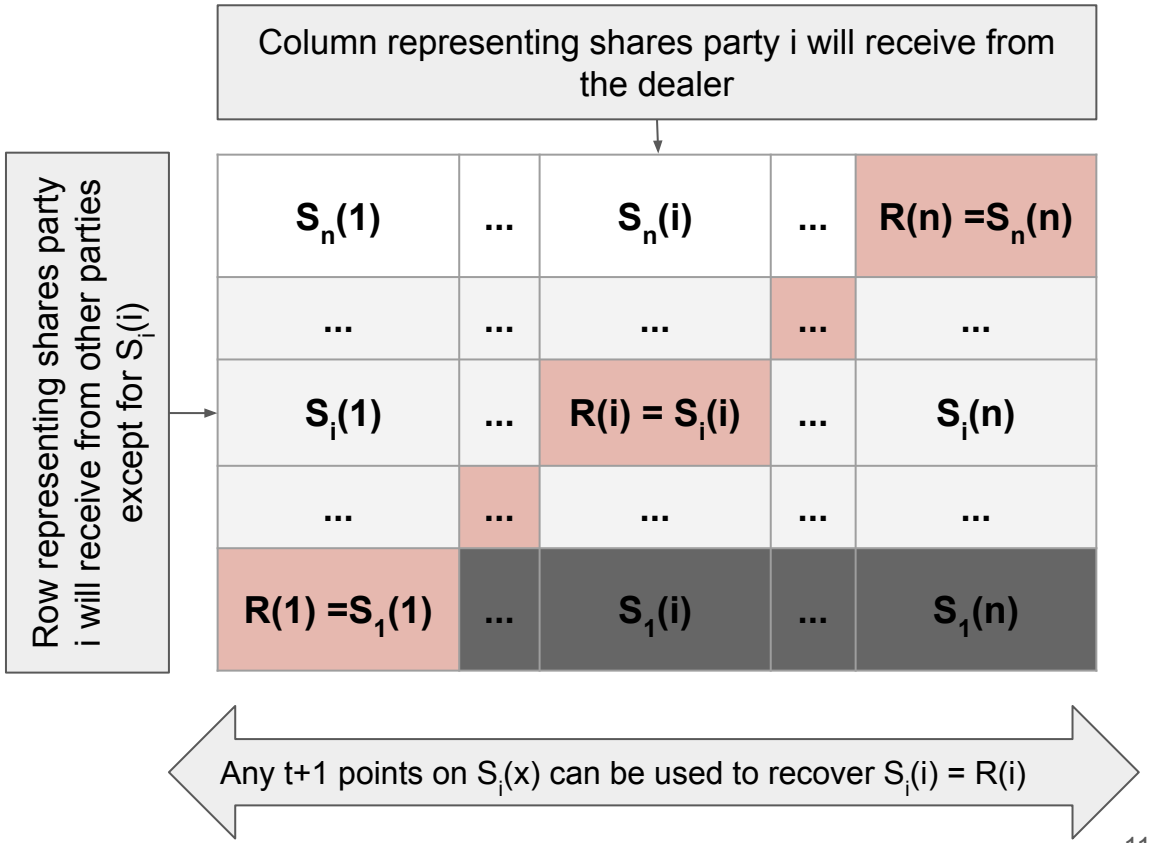
Name	Polynomial Commitment	Size		Additively Homomorphic $f_i + f_2 = f_3 \rightarrow$ (C1 operator C2) = C3
		Commitment	Witness	
Feldman style commitment	[g^a_0,g^a_1 … g^a_d]	linear in d	NULL	Yes / C1 * C2 = C3
Pedersen style	[g^a_0*h^r_0, … g^a_d*h^r_d]	linear in d	NULL	Yes / C1 * C2 = C3
KZG commitments	(g^f(α)) or (g^f(α) * h^f'(α))	constant	constant	Yes / C1 * C2 = C3
Bullet proofs	(Π(g$_i$^ a$_i$)) or ((Π(g$_i$^ a$_i$)) * h^r)	constant	log(d)	Yes / C1 * C2 = C3

Solving the consistency problem

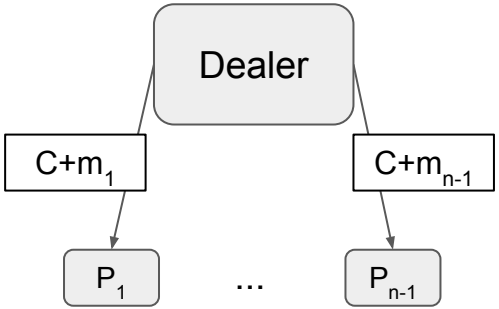
High Level Idea. Every party Checks that the row polynomial is a secret sharing of a share on R ($\forall i\ S_i(i)=R(i)$)

Dealer gives every party access to the polynomial commitments of S_i and R_i and a witness that $(S_i - R)(i) = 0$

Reminder. *Any p+1 points on the pink diagonal can reconstruct R(0) = s !*



Dealing Stage



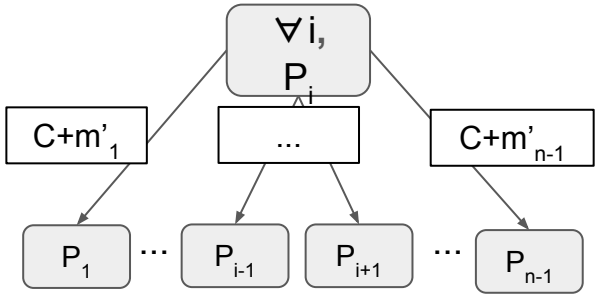
Protocol.

1. Create a polynomial R with degree p+1 such that f(0) = s.
2. Produce n points using f, secret share every point and produce the row columns.
3. Commit to every row polynomial S_i and to the diagonal R_i
4. Create witnesses that $(R- S_i)(i) = 0$
5. Commit to all S_i and R, we are going to call this the root commitment C
6. Send C and all S commits to everyone, and for every P_i the proper shares and witnesses

Shares party i will receive from the dealer

$S_n(1)$	...	$S_n(i)$	...	$R(n) = S_n(n)$
...	...	...	...	...
$S_i(1)$	...	$R(i) = S_i(i)$	...	$S_i(n)$
...	...	...	...	...
$R(1) = S_1(1)$	...	$S_1(i)$	...	$S_1(n)$

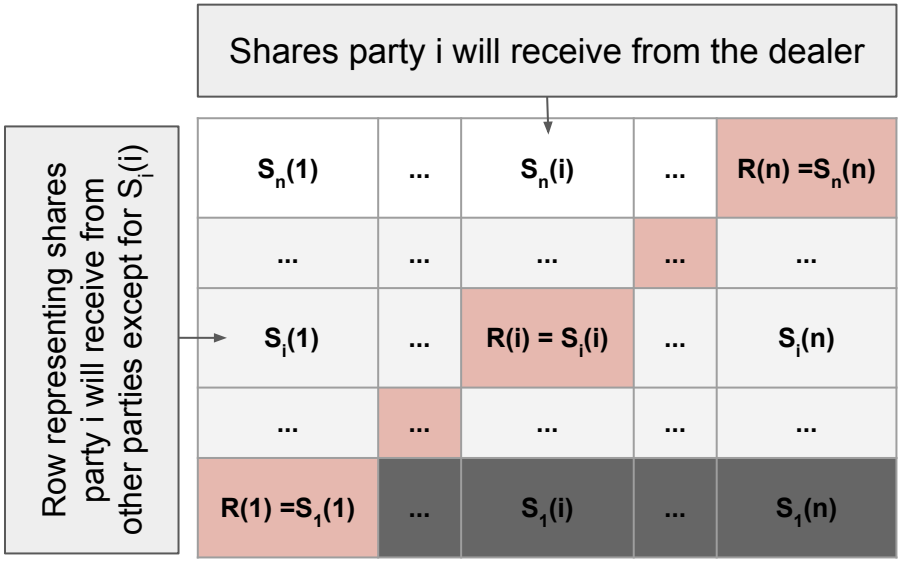
Echo Stage



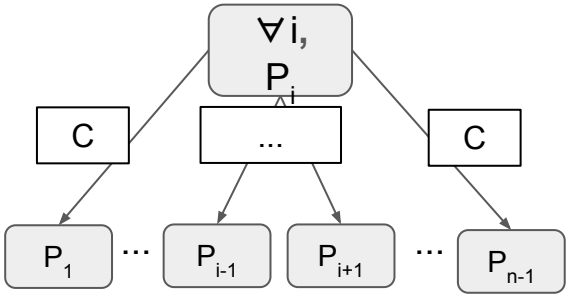
Protocol.

1. Each party p_i will perform checks to see that the C is produced consistently with the data provided by dealer.

2. For every party j: Send C, party j's share and what it thinks is the party's polynomial commitment. Along with an argument that it's linked to C, C -> S_i -> share

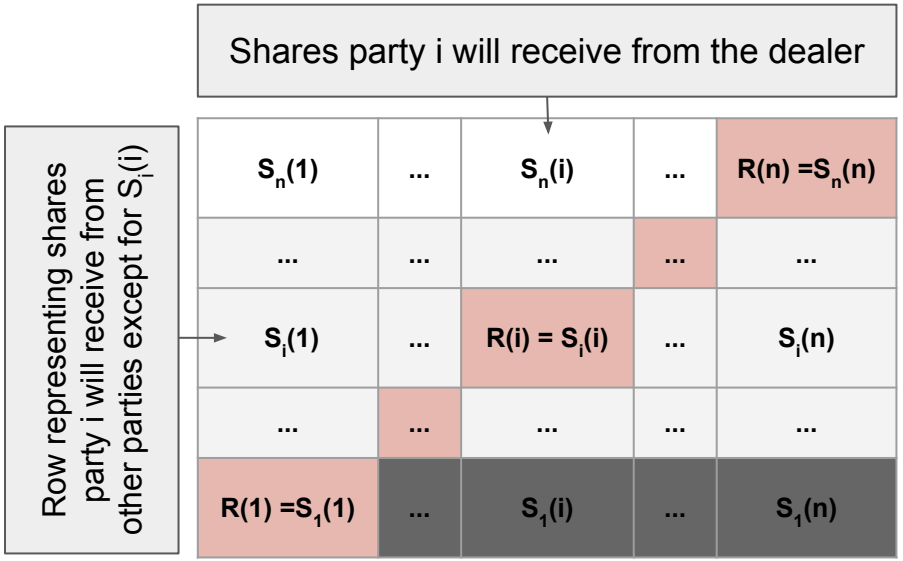


Ready Stage

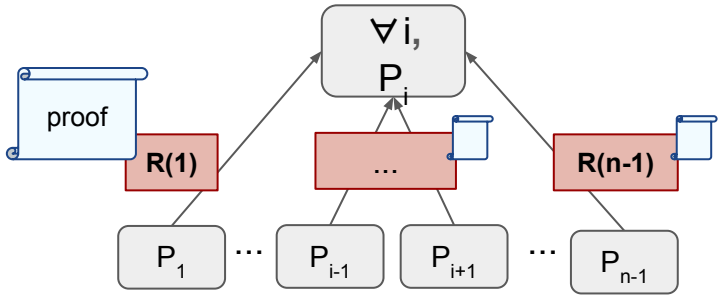


Protocol.

1. Each party Pi will send a ready message C, in only two cases:
2. If 2t+1 echo with the same message C and are “good echo message”
3. Or t+1 ready messages



Reconstruction



$S_n(1)$	…	$S_n(i)$	…	$R(n) = S_n(n)$
…	…	…	…	…
$S_i(1)$	…	$R(i) = S_i(i)$	…	$S_i(n)$
…	…	…	…	…
$R(1) = S_1(1)$	…	$S_1(i)$	…	$S_1(n)$

Reconstruct R(0) = s from diagonal

s=R(0)	R(1)	…	R(i)	…	R(n)