Efficient Perfectly Secure Computation with Optimal Resilience

Gilad Asharov
Bar-Ilan University

Ittai Abraham, Avishay Yanai
VMWare Research

TCC 2021

BIU
Center for Research in Applied Cryptography and Cyber Security

Secure Multiparty Computation

- A set of mutually distrustful parties
- Each has some **private** input
- **Goal**: To compute some **joint** function of their inputs
- Requirements:
  - Privacy, Correctness, more..

Seminal Results

- Any function can be securely computed [Yao'86,GMW'87,BGW'88,CCD'88,RB'89,BMR'90]

- **[BenOr-Goldwasser-Widgerson'88]:**
  - Semi-honest, perfect security, assuming $n \geq 2t + 1$
  - Malicious, perfect security, assuming $n \geq 3t + 1$

t: number of corrupted parties n: number of parties

The Communication Complexity

- For computing an arithmetic circuit C:

BGW:

- **Semi-honest:** $O(n^2 \cdot |C|)$
- **Malicious:** $O(n^4 \cdot |C|)$ optimistic case, $O(n^6 \cdot |C|)$ pessimistic case

Our Result:

- **Malicious**: $O(n^3 \cdot |C|)$ optimistic case, $O(n^4 \cdot |C|)$ pessimistic case

Counting Secret Shares Per Multiplication

BGW:

- **Semi-honest:** $O(n \cdot \text{comm}(SS))$
- **Malicious:** $O(n^2 \cdot \text{comm}(VSS))$

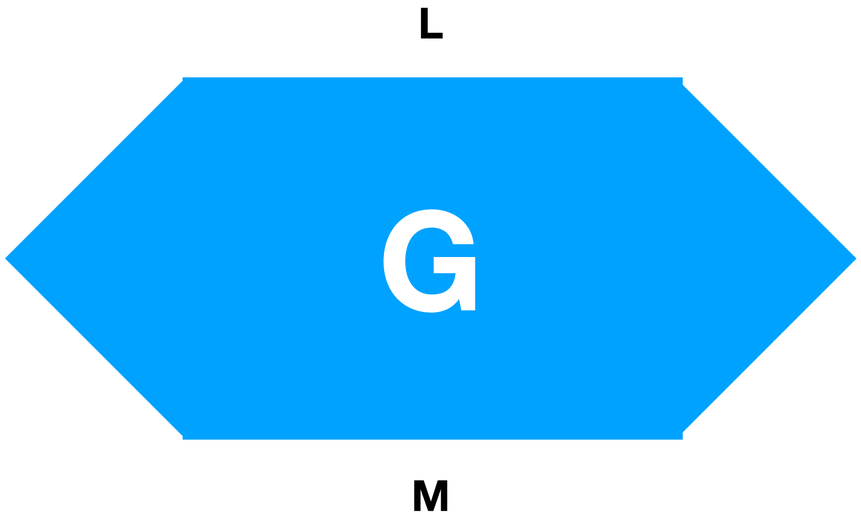
Our Result:

- **Malicious:** $O(n \cdot \text{comm}(VSS))$

VSS = verifiable secret sharing

The Bonus

- Consider a circuit $G : \{0,1\}^L \rightarrow \{0,1\}^M$ of multiplication-depth 1:
  - **BGW:** $O((L+M)n + |G|n^2)$ VSSes
  - **Our work**: $O((L+M)n)$ VSSes



Sub-linear CC in the size of the circuit !!

Example: Matrix Multiplication

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mm} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11}b_{11} + \dots + a_{1m}b_{m1} & a_{11}b_{12} + \dots + a_{1m}b_{m2} & \dots & a_{11}b_{1m} + \dots + a_{1m}b_{mm} \\ a_{21}b_{11} + \dots + a_{2m}b_{m1} & a_{21}b_{12} + \dots + a_{2m}b_{m2} & \dots & a_{21}b_{1m} + \dots + a_{2m}b_{mm} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \dots + a_{mm}b_{m1} & a_{m1}b_{12} + \dots + a_{mm}b_{m2} & \dots & a_{m1}b_{1m} + \dots + a_{mm}b_{mm} \end{pmatrix}$$

We have m^3 multiplications of the same $2m^2$ values

BGW: $O(n^2 \cdot m^3 \cdot \text{comm}(VSS))$

Our Protocol: $O(n \cdot m^2 \cdot \text{comm}(VSS))$

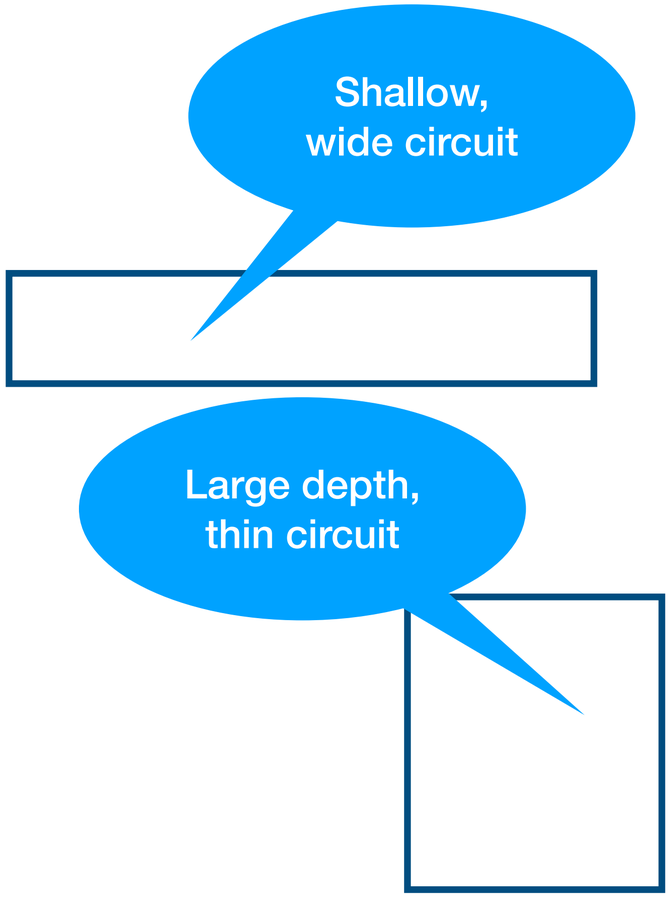
Related Work

- **Our setting:** Constant round per multiplication, perfect security, malicious, optimal resilience
  - [BGW88,GRR98,CDM00,**A**LR11]
  - **Our work**:
    $O(n^3 \cdot |C|)$ optimistic,
    $O(n^4 \cdot |C|)$ pessimistic,
    $O(depth(C))$ round complexity

  - **[BH08,GSZ19]:**
    $O(n \cdot |C|)$ communication,
    $O(n + depth(C))$
    round complexity

Example:
Matrix Multiplication

$O(n^4 \cdot m^2)$	Comm
$O(1)$	Rounds
$O(n \cdot m^3)$	Comm
$O(n)$	Rounds

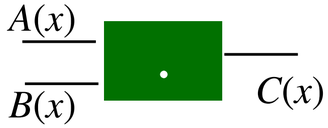


Techniques

- Assume:
  - Familiarity with **semi-honest BGW protocol**
    (with simplification of [GRR98])
    https://www.youtube.com/watch?v=XA_4dzs1Zys

  - Familiarity with **verifiable secret sharing and bivariate sharing**
    https://www.youtube.com/watch?v=Qm4EgaNDLK4

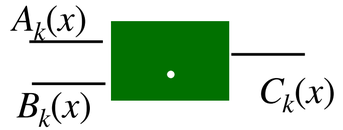
The BGW Multiplication



- **Input**:
  - Each party P_i holds $A(\alpha_i),\ B(\alpha_i)$ for some degree-t polynomials $A(x), B(x)$

- **Goal**: each party P_i should hold some share $C(\alpha_i)$ for some degree-t polynomial satisfying $C(0) = A(0) \cdot B(0)$

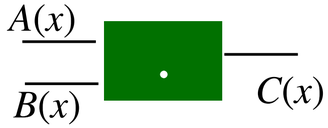
Multiplication with A Dealer



- **Input**:
  - Each party P_i holds $A_k(\alpha_i),\ B_k(\alpha_i)$ for some degree-t polynomials $A_k(x), B_k(x)$
  - **The dealer knows** $A_k(x), B_k(x)$

- **Goal**: each party P_i should hold some share $C_k(\alpha_i)$ for some degree-t polynomial satisfying $C_k(0) = A_k(0) \cdot B_k(0)$
  (The dealer knows $C_k(x)$)

Multiplication with A Dealer



- **Input**:
  - Each party P_i holds $A(\alpha_i),\ B(\alpha_i)$ for some degree-t polynomials $A(x), B(x)$
  - **The dealer knows** $A(x), B(x)$

- **Goal**: each party P_i should hold some share $C(\alpha_i)$ for some degree-t polynomial satisfying $C(0) = A(0) \cdot B(0)$
  (The dealer knows $C(x)$)

If the dealer fails - the parties simply recover $A(0), B(0)$ and use $C(x) = A(0) \cdot B(0)$

Multiplication with a Dealer [BGW]

- **The dealer** holds polynomials $A(x), B(x)$
- **The dealer** shares $C(x)$ satisfying $C(0) = A(0) \cdot B(0)$
- Each party P_j holds $A(\alpha_j), B(\alpha_j), C(\alpha_j)$
- **The dealer** defines t polynomials $D_1(x), \ldots, D_t(x)$

$$C(x) = A(x) \cdot B(x) - \sum_{\ell=1}^{t} x^{\ell} \cdot D_{\ell}(x)$$

- (Note that $C(0) = A(0) \cdot B(0)$)
- **The dealer** shares $D_1(x), \ldots, D_t(x)$, and each party verifies that:

$$C(\alpha_i) = A(\alpha_i) \cdot B(\alpha_i) - \sum_{\ell=1}^{t} \alpha_i^{\ell} \cdot D_{\ell}(\alpha_i)$$

- If not - broadcast a complain…

$O(n)$ VSSs

O(n) VSSs -> O(1) VSSs

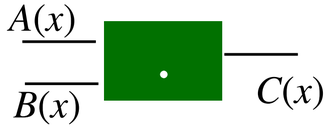
You cannot do better!
Each polynomial gives you just one degree of freedom!

	x	x^2	x^3	$\ldots$	x^t	x^{t+1}	x^{t+2}	$\ldots$	x^{2t-2}	x^{2t-1}	x^{2t}
D_t					$r_{t,0}$	$r_{t,1}$	$r_{t,2}$	$\ldots$	$r_{t,t-2}$	$r_{t,t-1}$	$R_{t,t}$
D_{t-1}				$\ldots$	$r_{t-1,1}$	$r_{t-1,2}$	$r_{t-1,3}$	$\ldots$	$r_{t-1,t-1}$	$R_{t-1,t}$	
D_{t-2}				$\ldots$	$r_{t-2,2}$	$r_{t-2,3}$	$r_{t-2,4}$	$\ldots$	$R_{t-2,t}$		
$\vdots$				$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$			
D_3			$r_{3,0}$	$\ldots$	$r_{3,t-3}$	$r_{3,t-2}$	$r_{3,t-1}$	$\ldots$			
D_2		$r_{2,0}$	$r_{2,1}$	$\ldots$	$r_{2,t-2}$	$r_{2,t-1}$	$R_{2,t}$				
D_1	$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	$\ldots$	$r_{1,t-1}$	$R_{1,t}$					

Table 1: Coefficients of the polynomial $\sum_{\ell=1}^{t} x^{\ell} \cdot D_{\ell}(x)$.

Maybe we can have secret sharing of a degree-2t polynomial?

Multiplication with A Dealer
Bivariate Sharing



- **Input**:
  - Each party P_i holds $A(\alpha_i),\ B(\alpha_i)$ for some degree-t polynomials $A(x), B(x)$
  - **The dealer knows** $A(x), B(x)$

- **Goal**: each party P_i should hold some share $C(x, \alpha_i),\ C(\alpha_i, y)$ for some degree-t polynomial satisfying $C(0,0) = A(0) \cdot B(0)$
  (The dealer knows $C(x, y)$**)**

If the dealer fails - the parties simply recover $A(0), B(0)$ and use $C(x, y)) = A(0) \cdot B(0)$

Bivariate sharing makes the sub-sharing verification protocol redundant [**A**LR11]

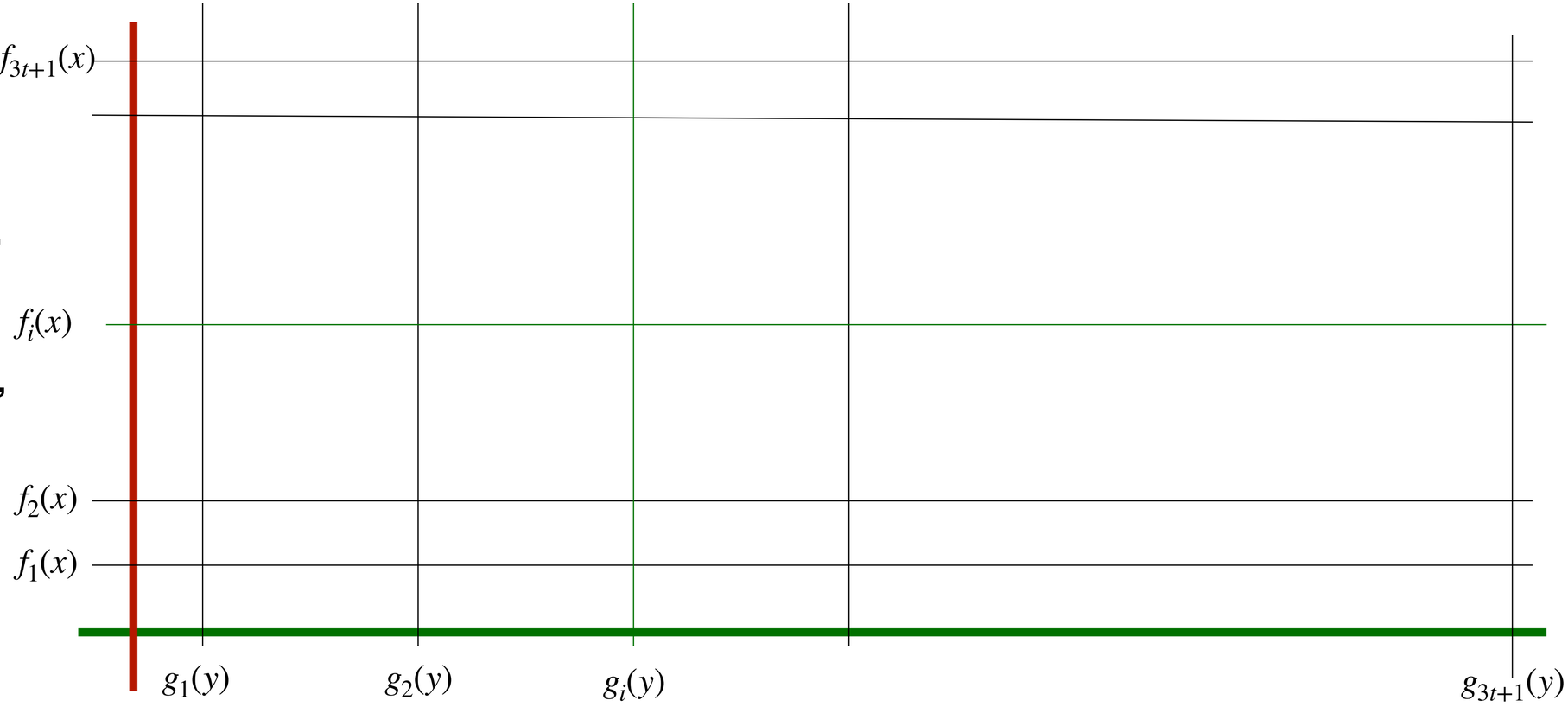
Sharing a Degree-2t Polynomial?

$$S(x,y) = \sum_{i=0}^{2t} \sum_{j=0}^{t} a_{i,j} x^i y^j$$

$$f_i(x) := S(x, \alpha_i)$$

$$g_i(y) := S(\alpha_i, y)$$

I: Dealer send shares
II: Exchange sub-shares, Complain
III: Dealer Resolve complaints
IV: Complaint Resolution, Vote

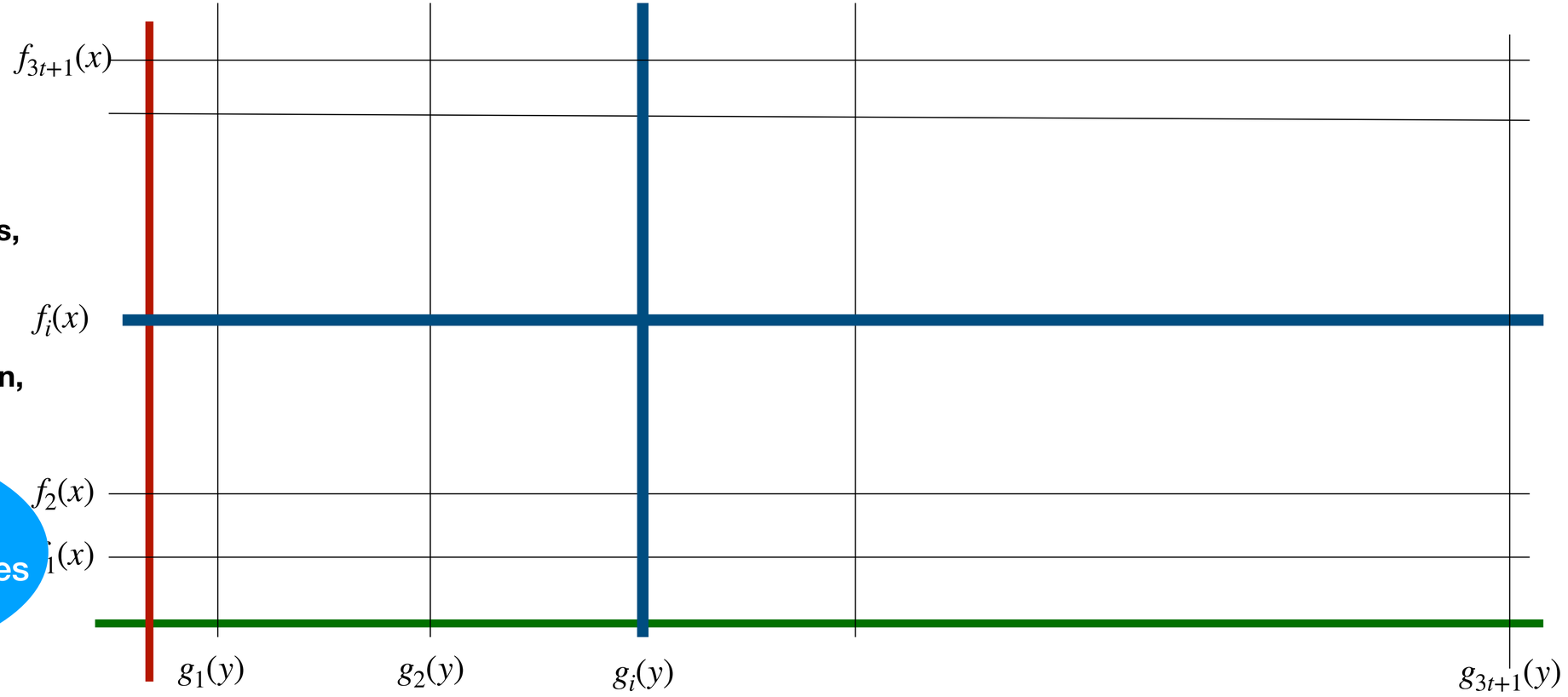


Sharing a Degree-2t Polynomial?

$$S(x,y) = \sum_{i=0}^{2t} \sum_{j=0}^{t} a_{i,j} x^i y^j$$

I: Dealer send shares
II: Exchange sub-shares, Complain
III: Dealer Resolve complaints
IV: Complaint Resolution, Vote

2t+1 good votes ->
Only t+1 honest parties

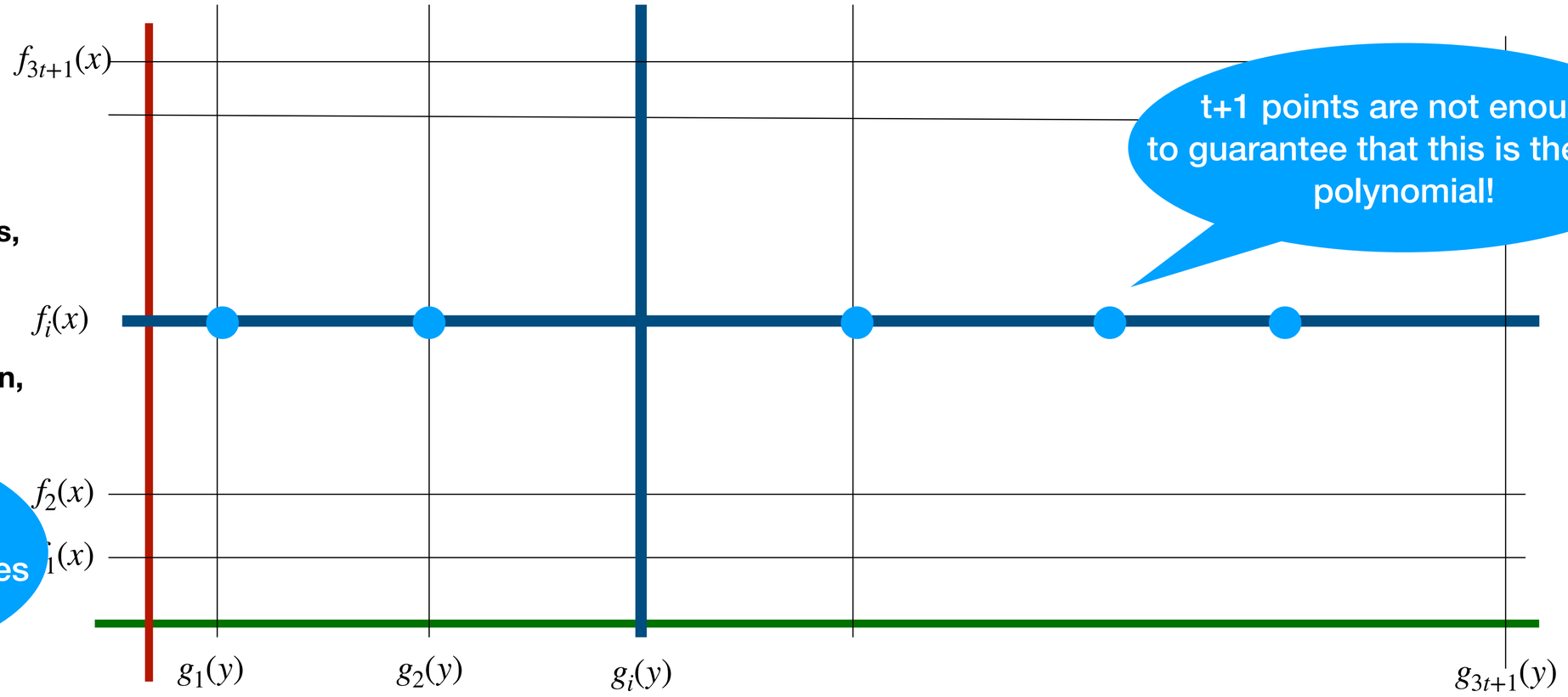


Sharing a Degree-2t Polynomial?

$$S(x,y) = \sum_{i=0}^{2t} \sum_{j=0}^{t} a_{i,j} x^i y^j$$

I: Dealer send shares
II: Exchange sub-shares, Complain
III: Dealer Resolve complaints
IV: Complaint Resolution, Vote

2t+1 good votes -> Only t+1 honest parties

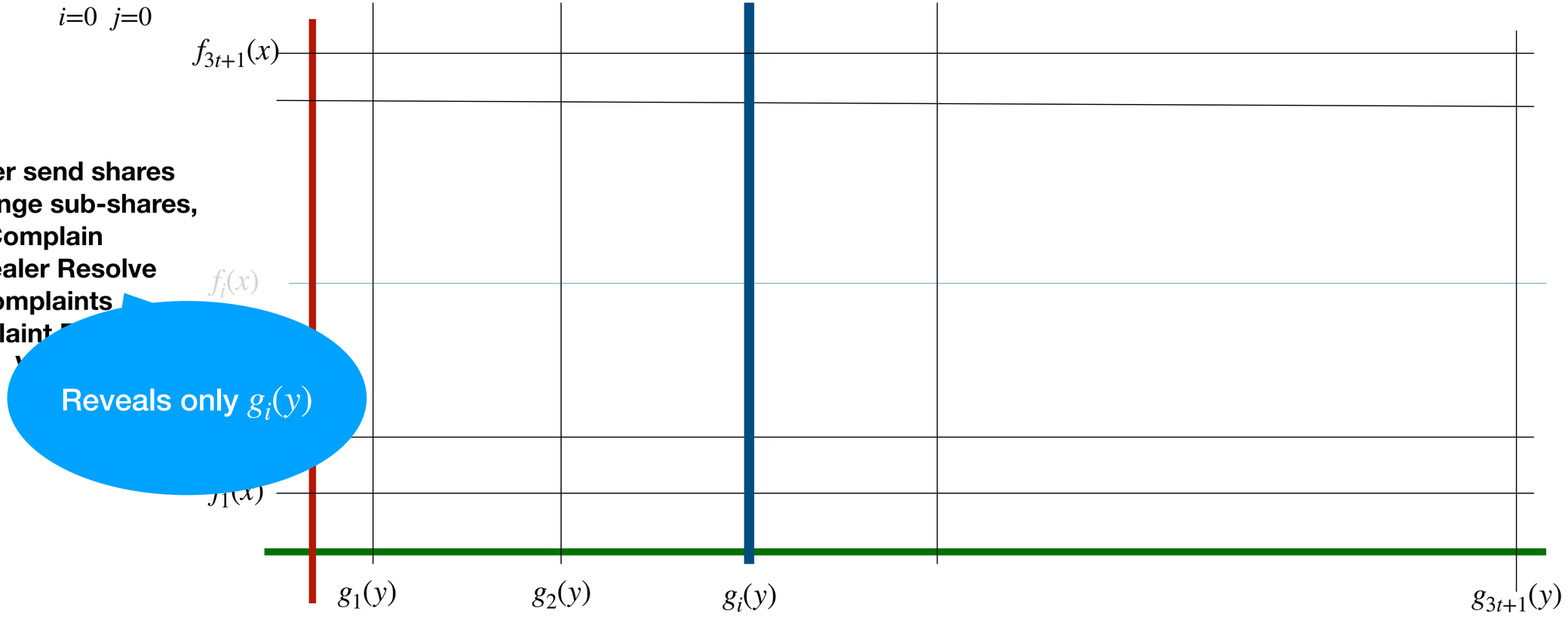


t+1 points are not enough to guarantee that this is the right polynomial!

Sharing a Degree-2t Polynomial?

$$S(x,y) = \sum_{i=0}^{2t} \sum_{j=0}^{t} a_{i,j} x^i y^j$$

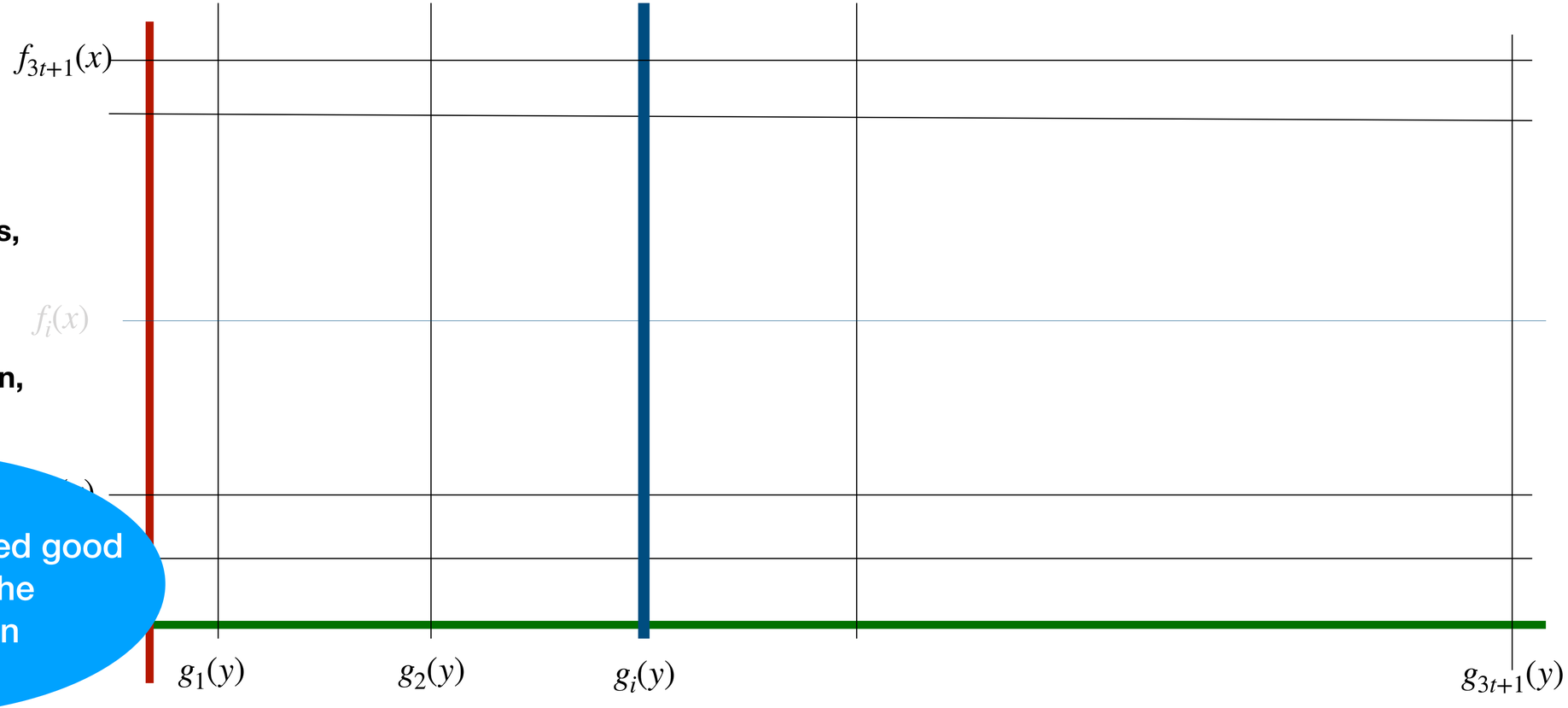
I: Dealer send shares
II: Exchange sub-shares, Complain
III: Dealer Resolve complaints
IV: Complaint



Sharing a Degree-2t Polynomial?

$$S(x,y) = \sum_{i=0}^{2t} \sum_{j=0}^{t} a_{i,j} x^i y^j$$

I: Dealer send shares
II: Exchange sub-shares, Complain
III: Dealer Resolve complaints
IV: Complaint Resolution, Vote



Only those who voted good contribute in the reconstruction

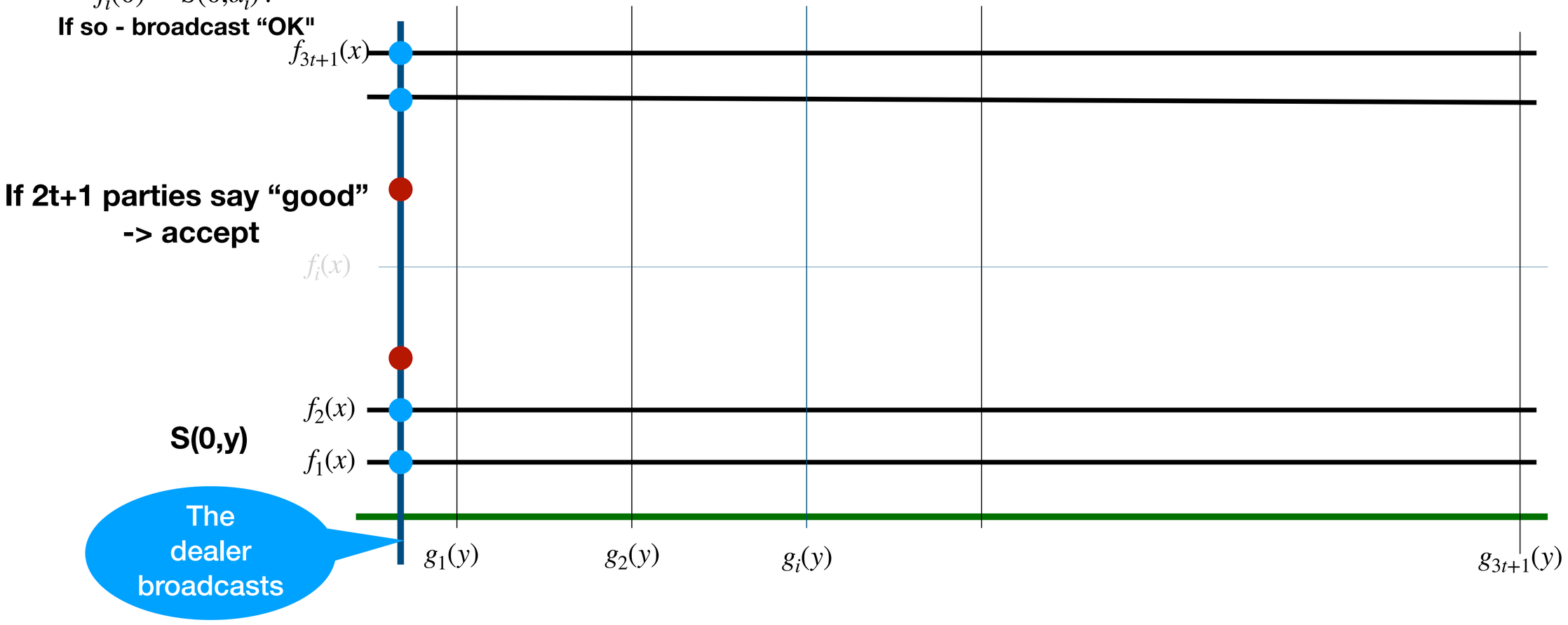
Honest dealer:
2t+1 honest parties have both
$f_i(x), g_i(y)$

Corrupted dealer:
t+1 honest parties have both
$f_i(x), g_i(y)$

Reconstruction

Only those who have $f_i(x)$: check: $f_i(0) = S(0,\alpha_i)$? If so - broadcast "OK"

$f_j(0) = S(0,\alpha_j)$?



If 2t+1 parties say "good" -> accept

S(0,y)

The dealer broadcasts

Honest dealer:
2t+1 honest parties have both
$f_i(x), g_i(y)$

Reconstruction: always succeeds

Corrupted dealer:
t+1 honest parties have both
$f_i(x), g_i(y)$

Reconstruction: Either to the polynomial that the honest parties hold, or to $\perp$

Weak Secret Sharing

Our Multiplication with a Dealer

- **The dealer** holds polynomials $A(x), B(x)$
- **The dealer** shares $C(x, y)$ satisfying $C(0,0) = A(0) \cdot B($
- Each party P_j holds $A(\alpha_j), B(\alpha_j), C(x, \alpha_j), C(\alpha_j, y)$
- **The dealer** defines a single polynomial $D(x, y)$ with degree-$2t$ in x and t in
  $D(x,0) = A(x) \cdot B(x) - C(x,0)$
- Each party verifies that $D(\alpha_i,0) = A(\alpha_i) \cdot B(\alpha_i) - C(\alpha_i,0)$
  - If not - broadcast a complain…
- Publicly reconstruct $D(0,y)$; All verify that $D(0,0) = 0$

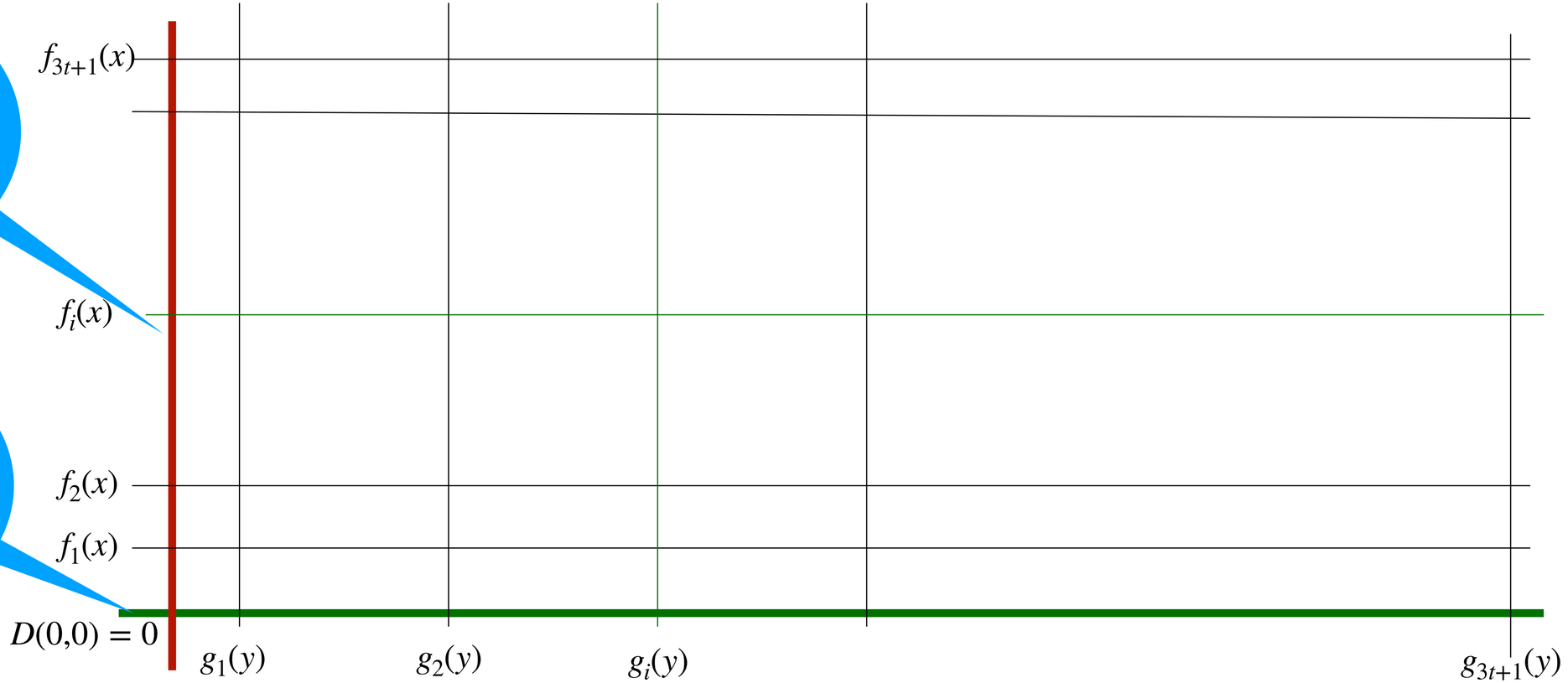
$D(x, y)$ is of degree $2t$

We obtain ***weak secret sharing*** of polynomial of degree $2t$ in x and t in y

Sharing a Degree-2t Polynomial

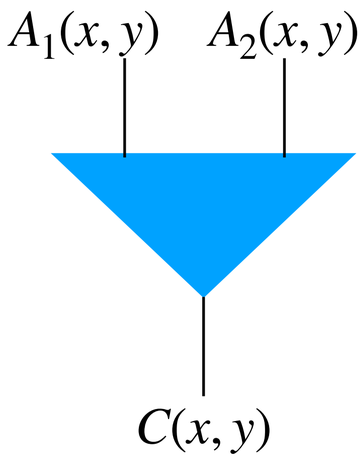
We reveal this polynomial to prove that $D(0,0) = 0$

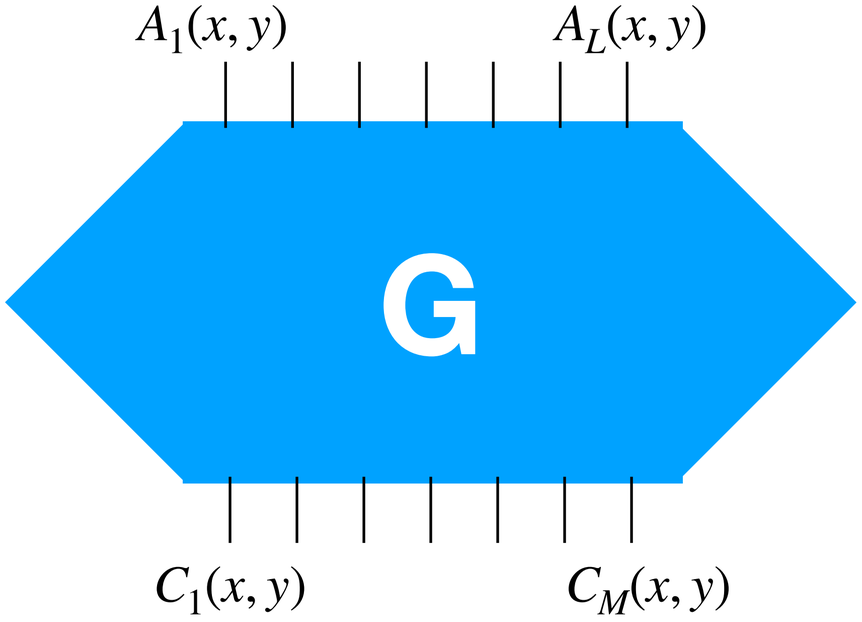
The adversary already knows t shares on this polynomial



$$D(x,0) = A(x) \cdot B(x) - C(x,0)$$

The Bonus





$$D(x,0) = A_1(x) \cdot A_2(x) - C(x,0)$$

$$D_1(x,0) = G(A_1, \ldots, A_L) - C_1(x,0)$$

Conclusions

- Improvements of a classical protocol
- Beats the state-of-the-art protocols for some applications
- Sublinear communication complexity in the perfect setting!

Thank you!