Almost-surely Terminating Asynchronous Byzantine Agreement Revisited

(Extended abstract published in **PODC 2018**)

Ashish Choudhury

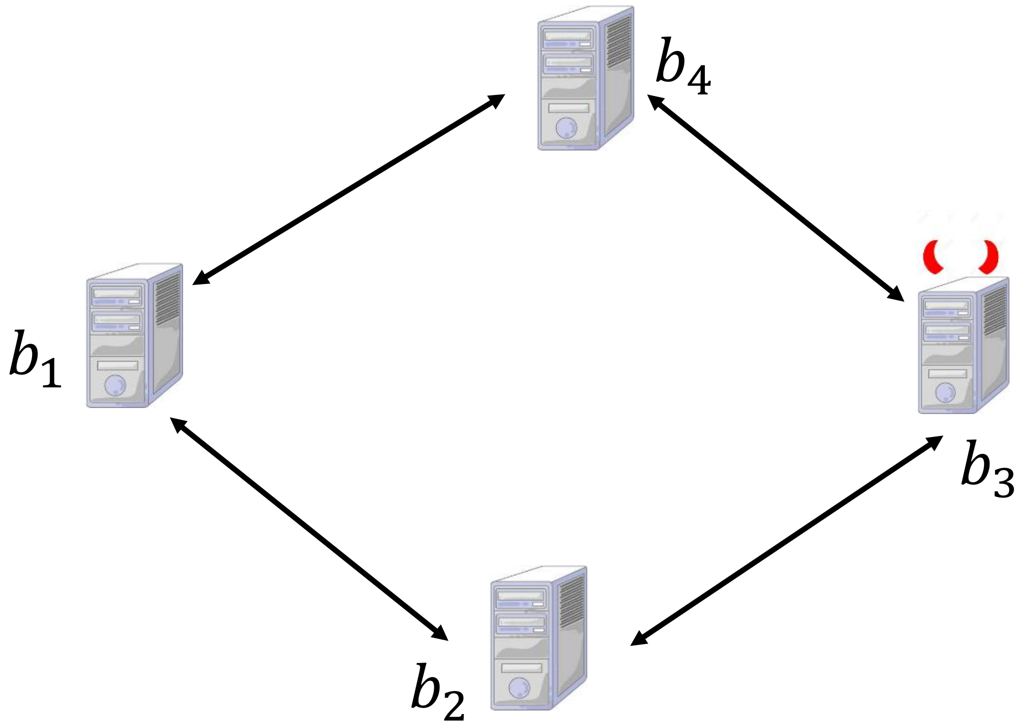
International Institute of Information Technology (IIIT) Bangalore

Joint work with:

B. Laasya (University of Rochester) --- **acknowledgement for the slides**

Arpita Patra (Indian Institute of Science)

Byzantine Agreement : Problem Definition



- n **mutually-distrusting parties**
- Up to t **corruptions**
- **Goal**: to design a **distributed protocol**, allowing the **honest parties** to agree on a **common output**

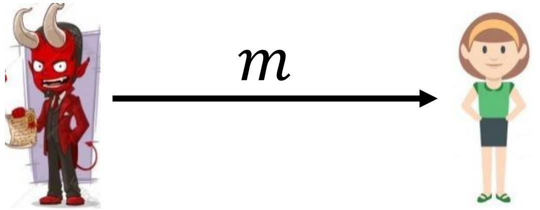
$b_1 \quad b_2 \quad b_3 \quad b_4 \quad \cdots \quad b_{n-1} \quad b_n$ ----> b **Agreement**

$b \quad b \quad b_3 \quad b \quad \cdots \quad b_{n-1} \quad b$ ----> b **Validity**

Asynchronous Communication Model

Asynchronous Network

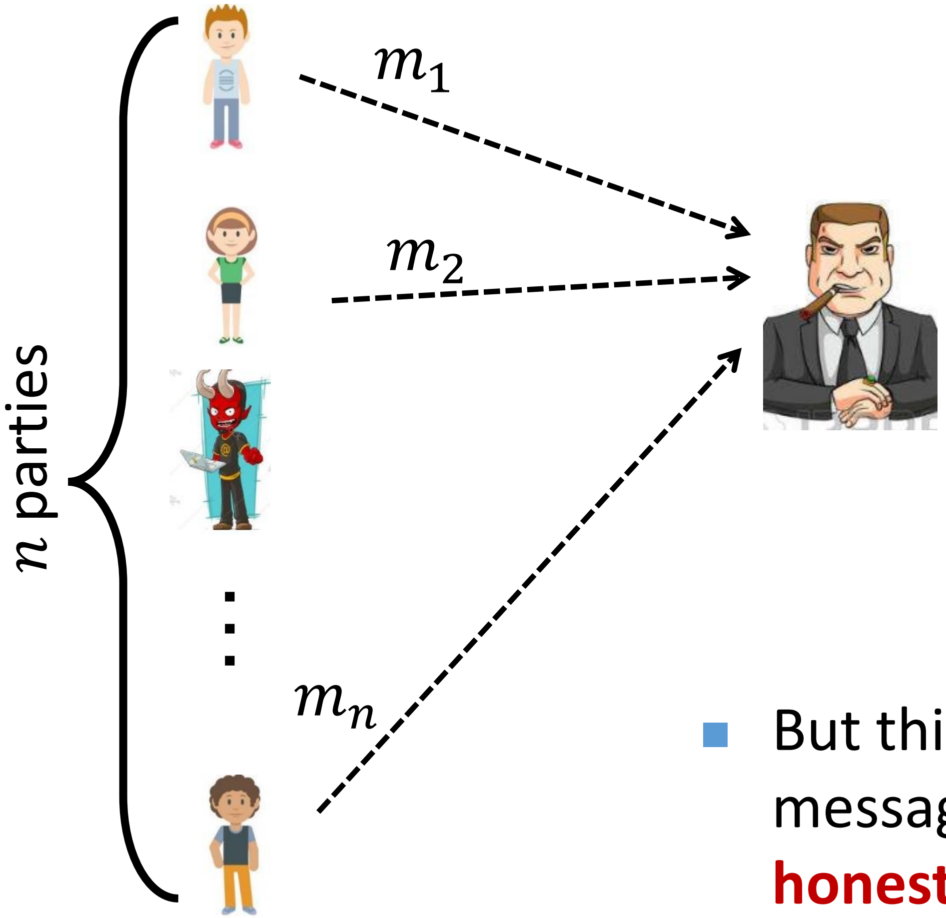
- No Global Clock
- Channels unbounded delay
- Waiting time is not known



Wait to receive m

How long??

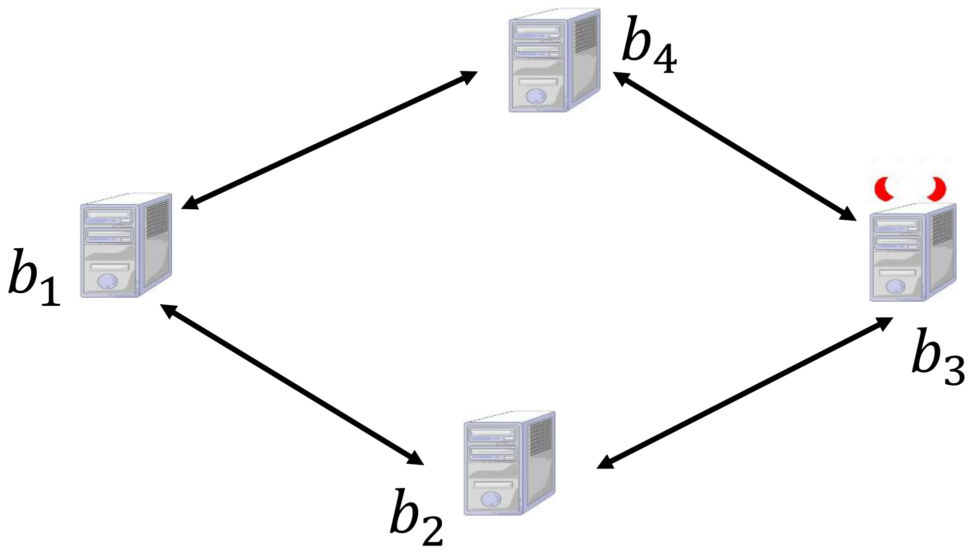
No distinction between a **slow** (but honest) **sender** and a **corrupt sender**



- Waiting for all results in **endless waiting**!
- Can afford to wait for $(n-t)$ parties
- But this can lead to **ignoring** messages of t **potentially honest parties**

In the asynchronous setting, the network itself is the adversary

BA Problem in the Asynchronous Setting : ABA



- n **mutually-distrusting parties,** up to t **corruptions**
- Completely **asynchronous network**
- **Goal:** to design a **distributed protocol**, allowing the **honest parties** to agree on a **common output**

$b_1 \quad b_2 \quad b_3 \quad b_4 \quad \cdots \quad b_{n-1} \quad b_n$ ----> b **Agreement**

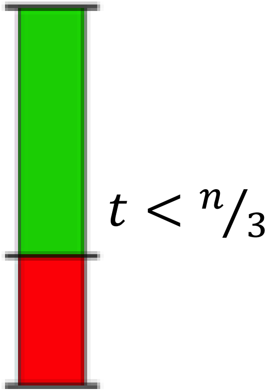
$b \quad b \quad b_3 \quad b \quad \cdots \quad b_{n-1} \quad b$ ----> b **Validity**

Termination

If all honest parties participate in the protocol, then all honest parties eventually terminate the protocol with an output

ABA Problem : Known Results

- ABA tolerating t **Byzantine faults** possible only if $t < n/3$
  - Holds, **even if a PKI setup is available** and parties are allowed to use cryptography



ABA: with or without cryptography

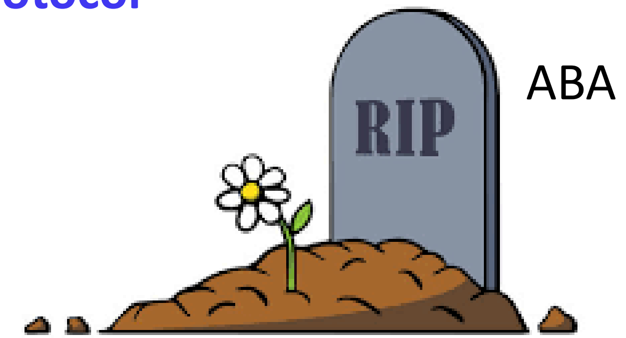
- **FLP Impossibility results for ABA**: Don't even dare to design a **deterministic** ABA protocol

[M. J. Fischer, N. A. Lynch and M. S. Paterson, JACM 1985]

Any **deterministic ABA** protocol will have **non-terminating runs**, even if one party crashes

How to Circumvent FLP Impossibility Result ?

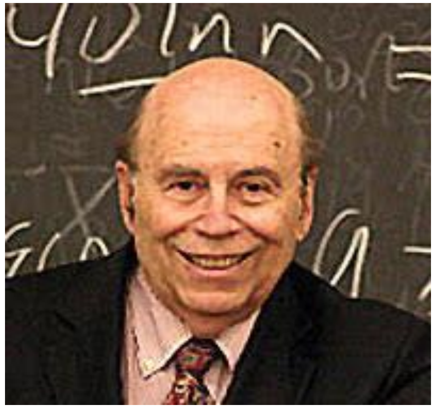
[M. J. Fischer, N. A. Lynch and M. S. Paterson, JACM 1985]: any **deterministic ABA protocol** will have **non-terminating runs**, even if one party crashes



- Does FLP impossibility result mean the end of ABA ?
  - No

- A common approach to circumvent FLP impossibility result --- “embrace” **randomness**

[M. Ben-Or, PODC 1983]



[M. Rabin, FOCS 1983]

- **$(1 - \lambda)$-terminating ABA**: honest parties terminate, with probability $(1 - \lambda)$
- **Almost-surely terminating ABA**: honest parties terminate, with probability 1

Relevant Results for Almost-surely Terminating ABA

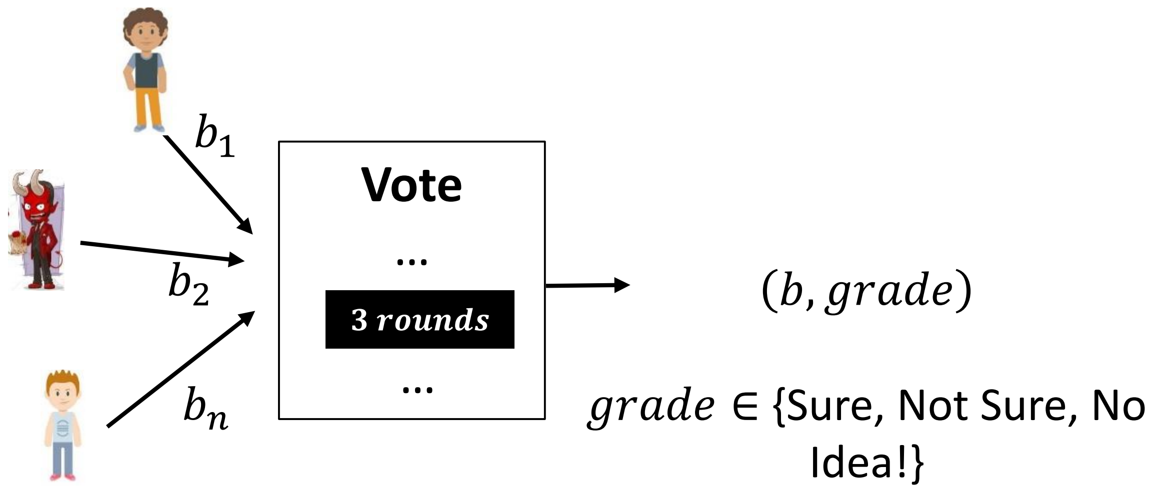
Reference	Resilience	Expected Rounds	Expected Communication Complexity	Expected Computation
Feldman-Micali, STOC 1988	$t < n/4$	$\mathcal{O}(1)$	$\mathcal{O}(n^6 \log(n) \log(\lvert\mathbb{F}\rvert))$	Polynomial
Abraham-Dolev-Halpern, PODC 2008	$t < n/3$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{10} \log(\lvert\mathbb{F}\rvert))$	Polynomial
Wang, CoRR 2015	$t < n/3$	$\mathcal{O}(n)$	$\mathcal{O}(n^7 \log(\lvert\mathbb{F}\rvert))$	Exponential

Our Results

$t < n/3$	$\mathcal{O}(n)$	$\mathcal{O}(n^6 \log(\lvert\mathbb{F}\rvert))$	Polynomial
$t < n/(3+\epsilon)$	$\mathcal{O}(1)$	$\mathcal{O}(n^5 \log(\lvert\mathbb{F}\rvert))$	Polynomial

Common Framework for Randomized BA (Rabin, Ben-Or)

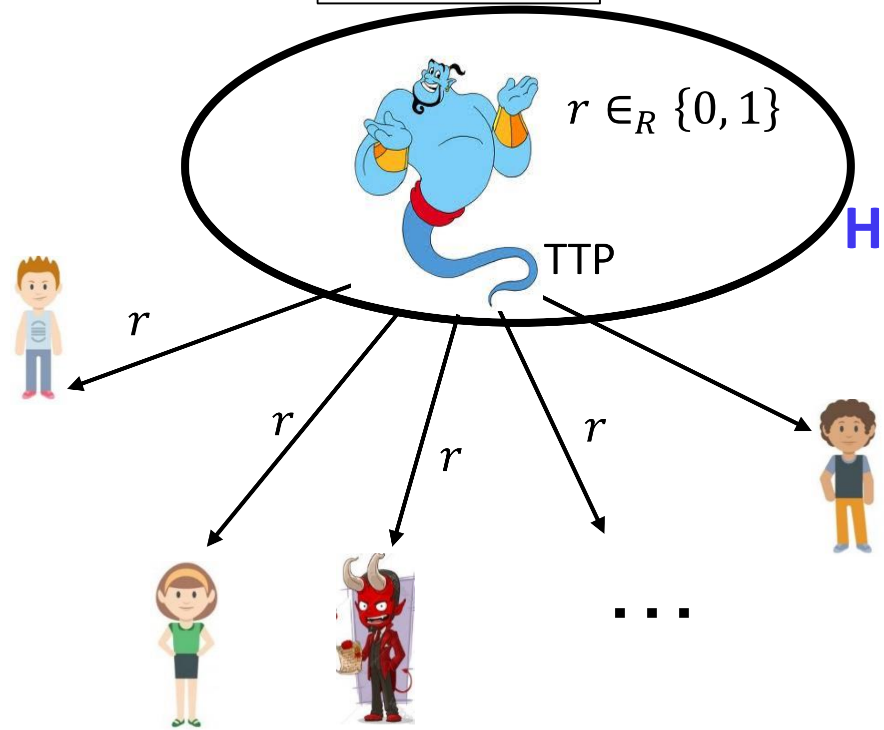
Vote



"whatever can be done deterministically"

- If **all honest parties have the same input bit** $\Rightarrow$ all honest parties output that common bit and **grade = Sure**

(p_0, p_1)-CC



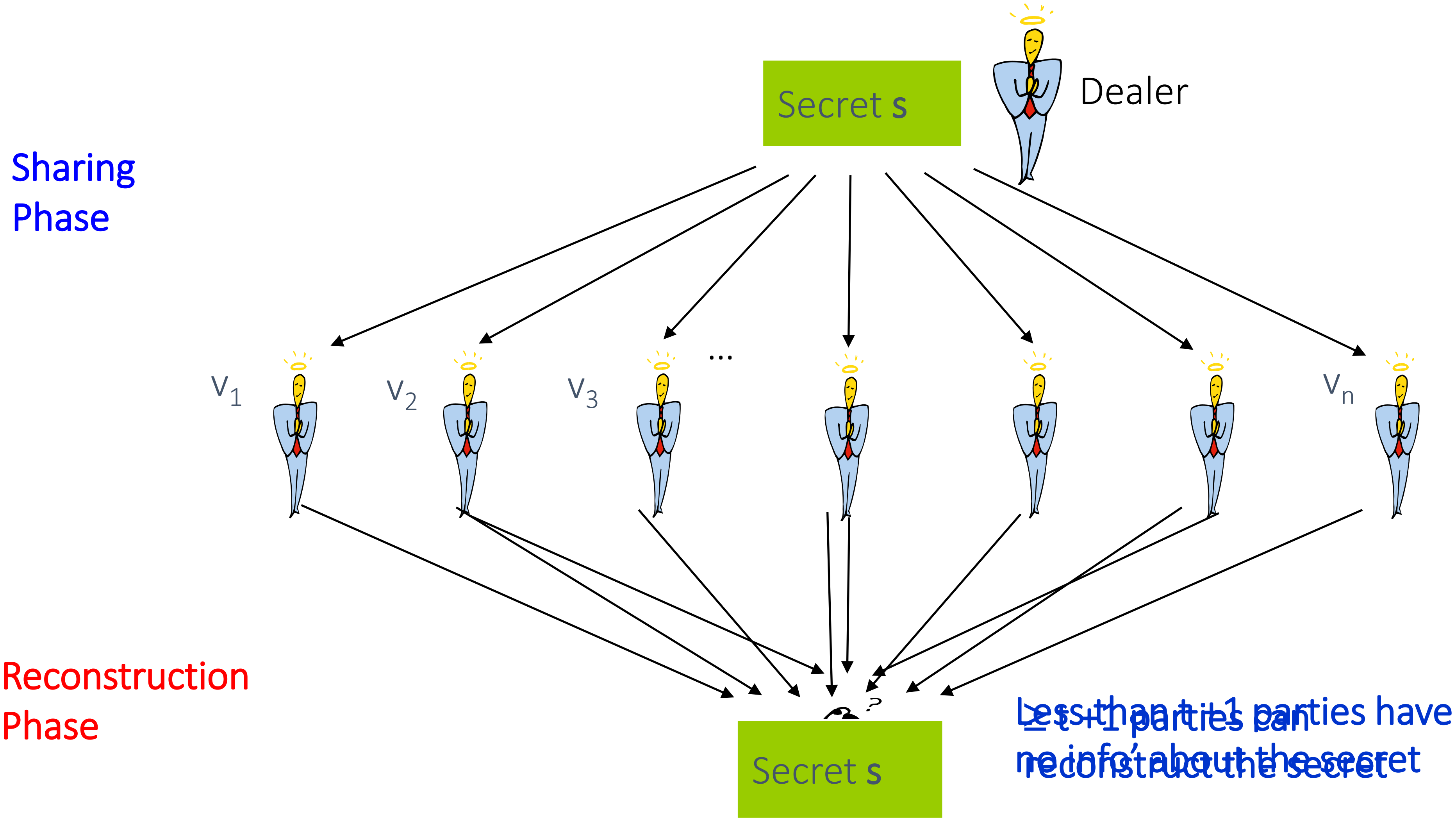
How to emulate the TTP ?

Common-ness: with probability p_r, the common bit for all honest parties will be r

If p_0 **and** p_1 **are constant**, then **expected constant number of iterations of Vote + CC** → ABA

(n,t) Secret Sharing

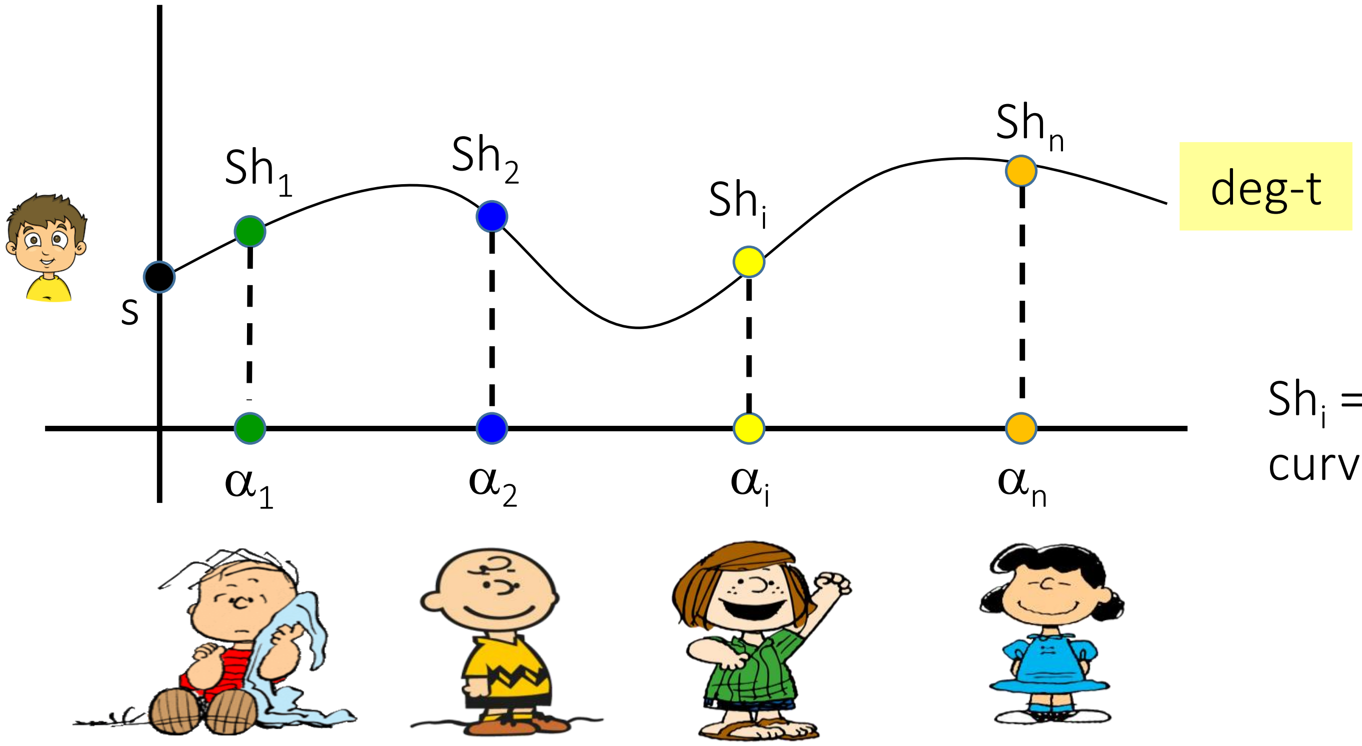
*Slide acknowledgement: Juan Garay



(n,t) Secret Sharing [Shamir79]

*Slide acknowledgement: Arpita Patra

Sharing Phase

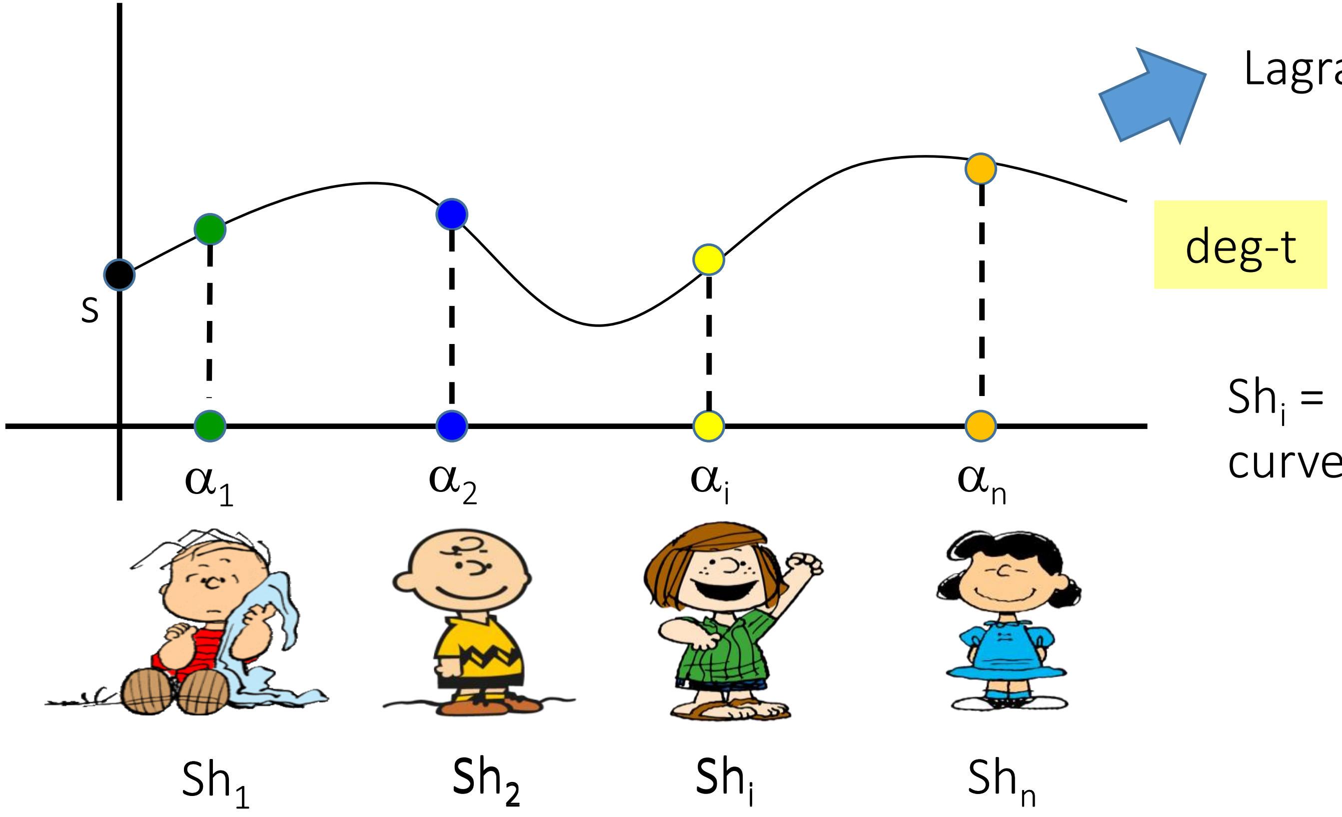


Sh_i = Evaluation of the curve at $x = \alpha_i$

(n,t) Secret Sharing [Shamir79]

*Slide acknowledgement: Arpita Patra

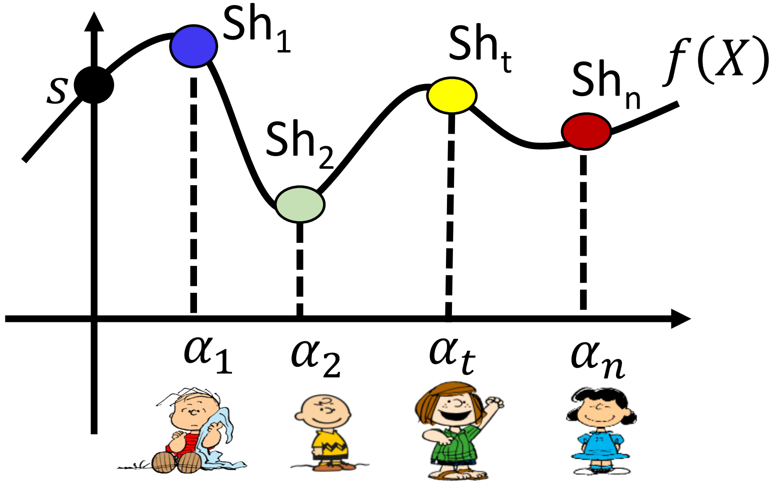
Reconstruction Phase



Lagrange Interpolation

Sh_i = Evaluation of the curve at x = α_i

(n, t)-Secret Sharing and Verifiable Secret-Sharing (VSS)



- $(t+1)$ **distinct values** of an unknown t-degree polynomial $f(X)$ are **sufficient** to uniquely reconstruct $f(X)$
- t **distinct values** of an unknown t-degree polynomial $f(X)$ are **not sufficient** to uniquely recover $f(X)$

- Shamir's secret-sharing is **insecure against a malicious adversary**
  - **Case I**: Honest dealer, but **corrupt share-holders**
    - Taken care by using **Reed-Solomon (RS) error-correction**
  - **Case II**: Corrupt dealer AND corrupt share-holders
    - Honest parties need to **verify** that **Dealer is committed** to a **single** t-degree polynomial

Provided $\frac{\text{Bad Shares}}{\text{Good Shares}} < \frac{1}{2}$

- **Asynchronous VSS** (AVSS) requirements
  - Secrecy
  - Correctness
  - Termination

Reducing Common Coin (CC) to Asynchronous Verifiable Secret-sharing (AVSS)

$t < n/4$

- [Feldman-Micali, STOC, 1988]

ABA → CC → AVSS

Asynchronous Byzantine Agreement | Common Coin | Asynchronous Verifiable Secret Sharing

---

$t < n/3$

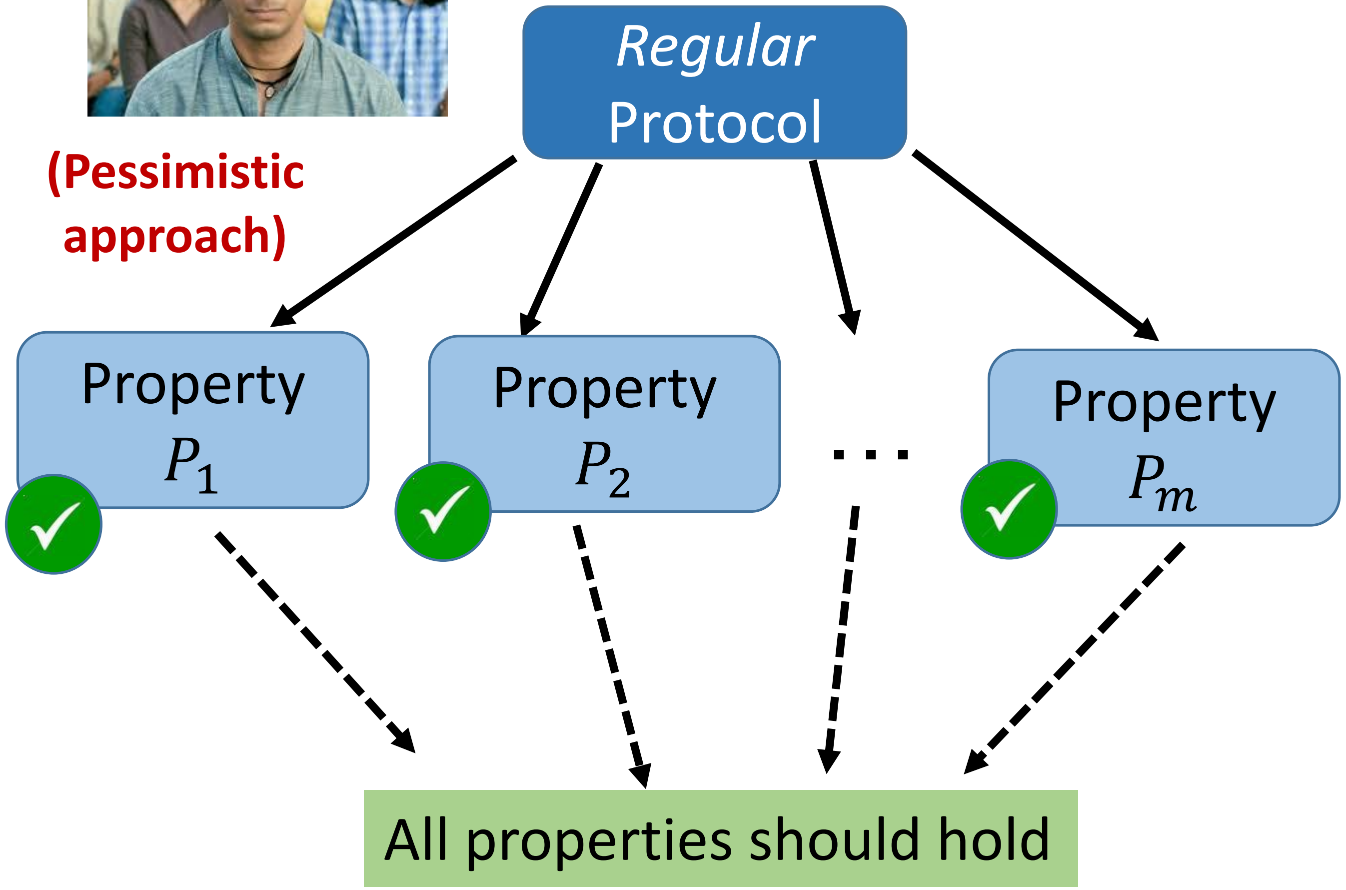
- [Abraham et. al., PODC, 2008]
- [Wang, CoRR, 2015]
- Our Protocol

ABA → SCC → SAVSS

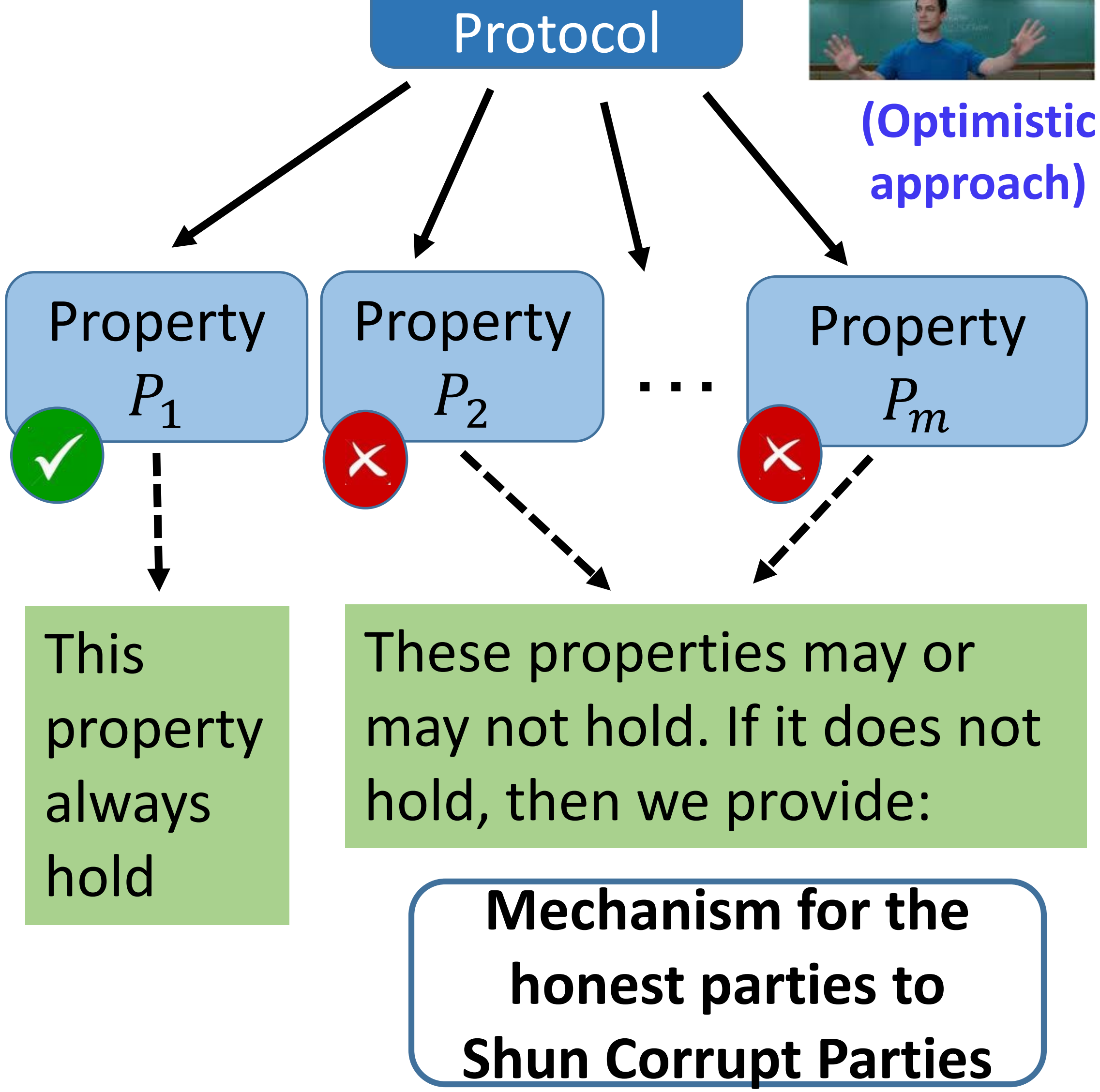
Asynchronous Byzantine Agreement | Shunning Common Coin | Shunning Asynchronous Verifiable Secret Sharing

The Notion of Shunning

(Pessimistic approach)



(Optimistic approach)

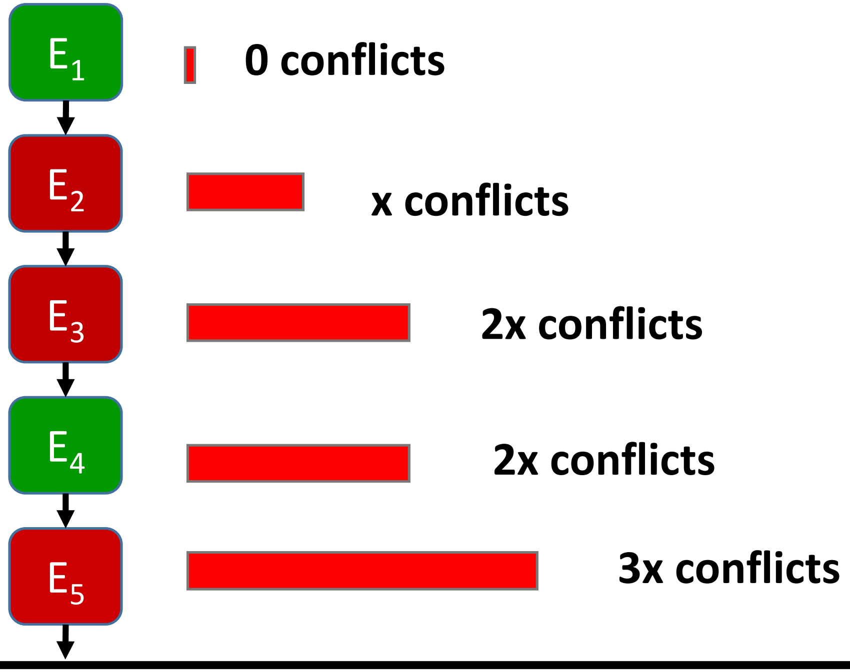


What does a Shunning Protocol Achieve ?

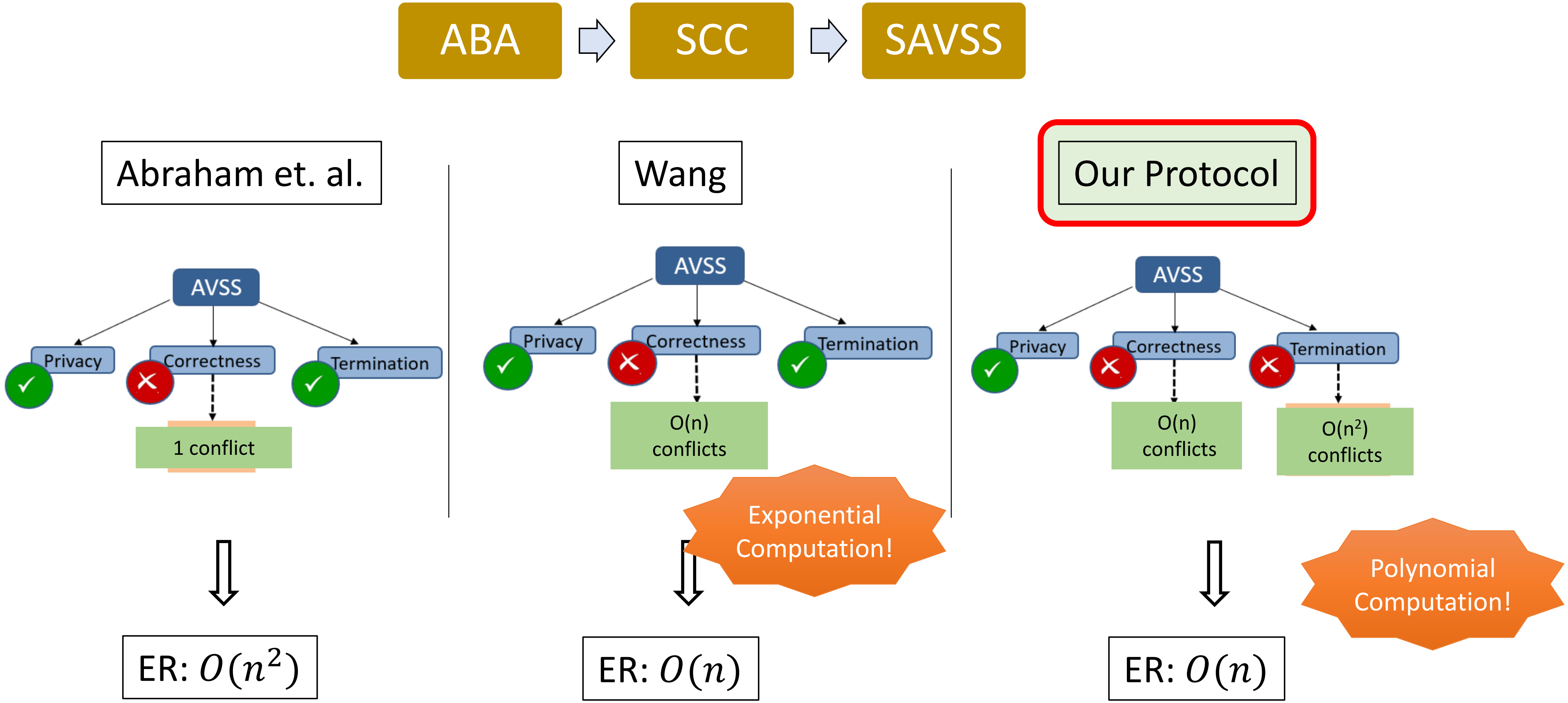
- Creates a win-win situation with the adversary
  - Either, the adversary let all the properties of the protocol being satisfied
  - Else, it exposes its identity to a subset of honest parties --- shunning/conflict

No honest party shunning another honest party, if any property is violated

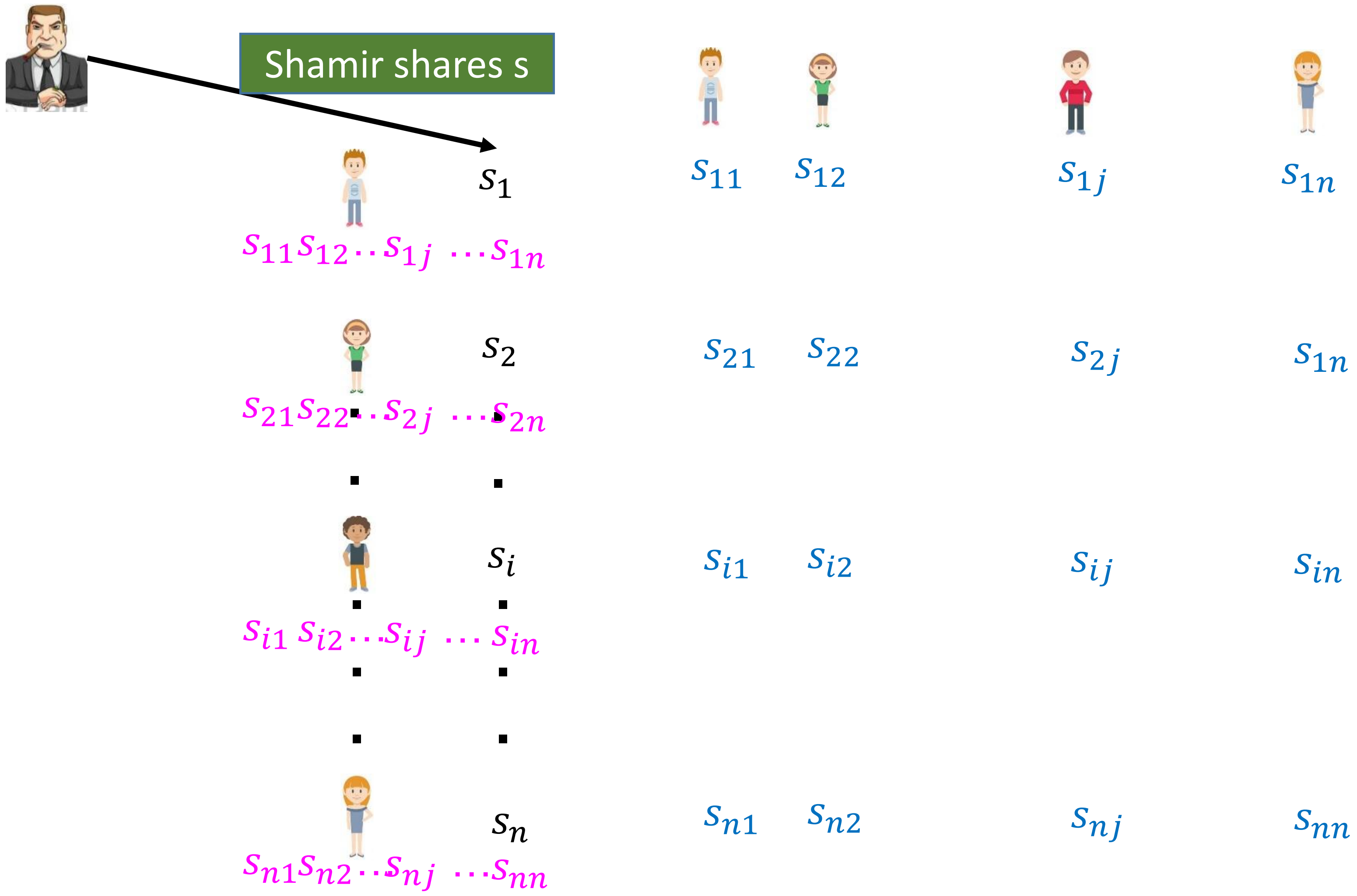
- Suppose X pairs of shunning happen whenever some property fails
  - At most $(n-t)t = \mathcal{O}(n^2)$ pairs of conflicts possible
  - After $\mathcal{O}(n^2/X)$ failed execution, all executions will be clean executions
    - If $X = \Omega(n)$ -> at most $\mathcal{O}(n)$ failed executions
    - If $X = \Omega(n^2)$ -> at most $\mathcal{O}(1)$ failed executions



Almost-Surely Terminating ABA from Shunning Common Coin



SAVSS: Dealer Shares the Secret



- D shamir shares its secret s
- Further, each share s_i is shamir-shared by D.

SAVSS: Pair-wise Consistency Check

s_{11} s_{12} s_{1j} s_{1n}

$s_{11}s_{12}\cdots s_{1j}\ \cdots s_{1n}$

s_{21} s_{22} s_{2j} s_{2n}

$s_{21}s_{22}\cdots s_{2j}\ \cdots s_{2n}$

s_{i1} s_{i2} s_{ij} s_{in}

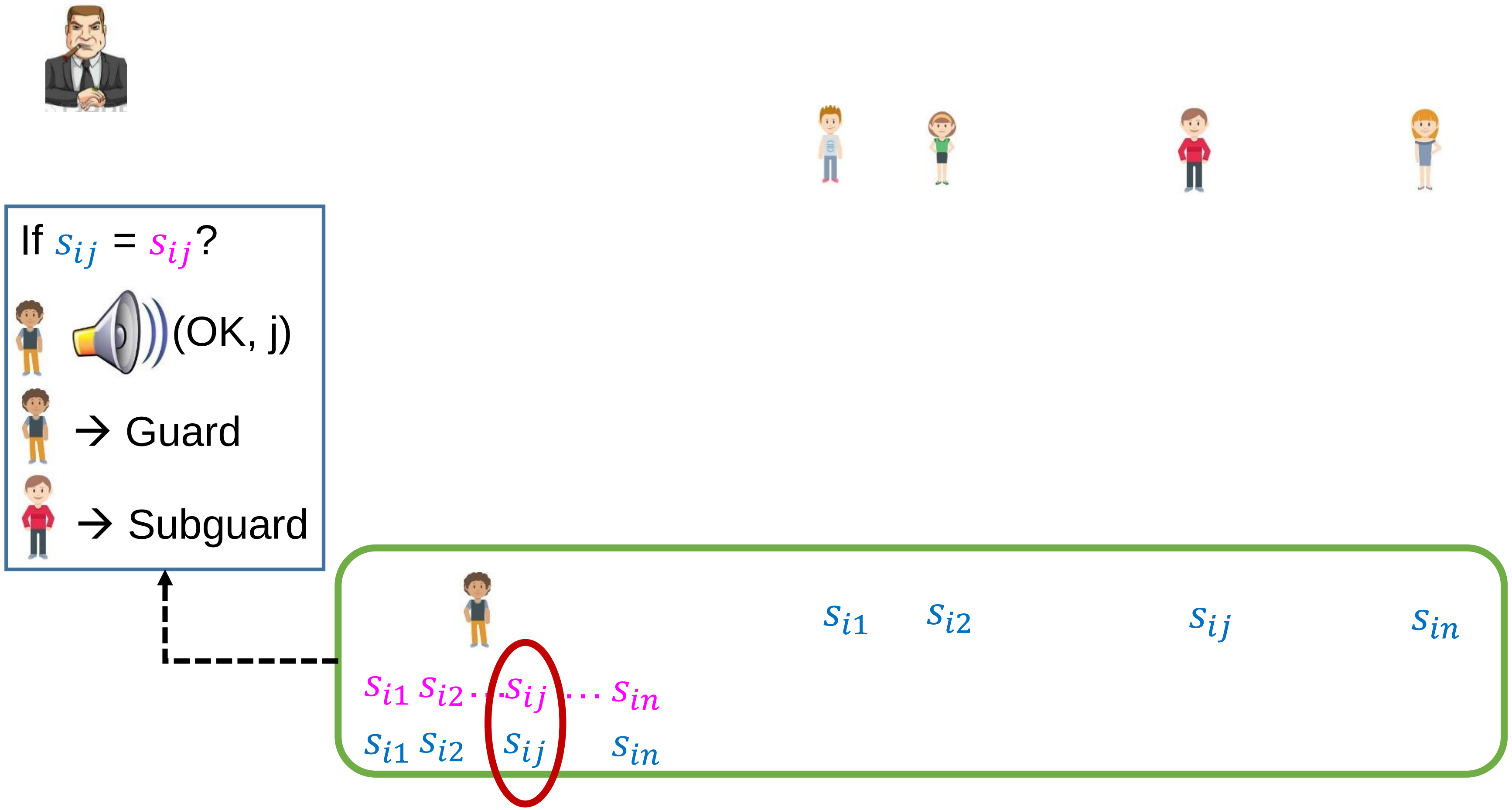
$s_{i1}\ s_{i2}\cdots s_{ij}\ \cdots s_{in}$

s_{n1} s_{n2} s_{nj} s_{nn}

$s_{n1}s_{n2}\cdots s_{nj}\ \cdots s_{nn}$

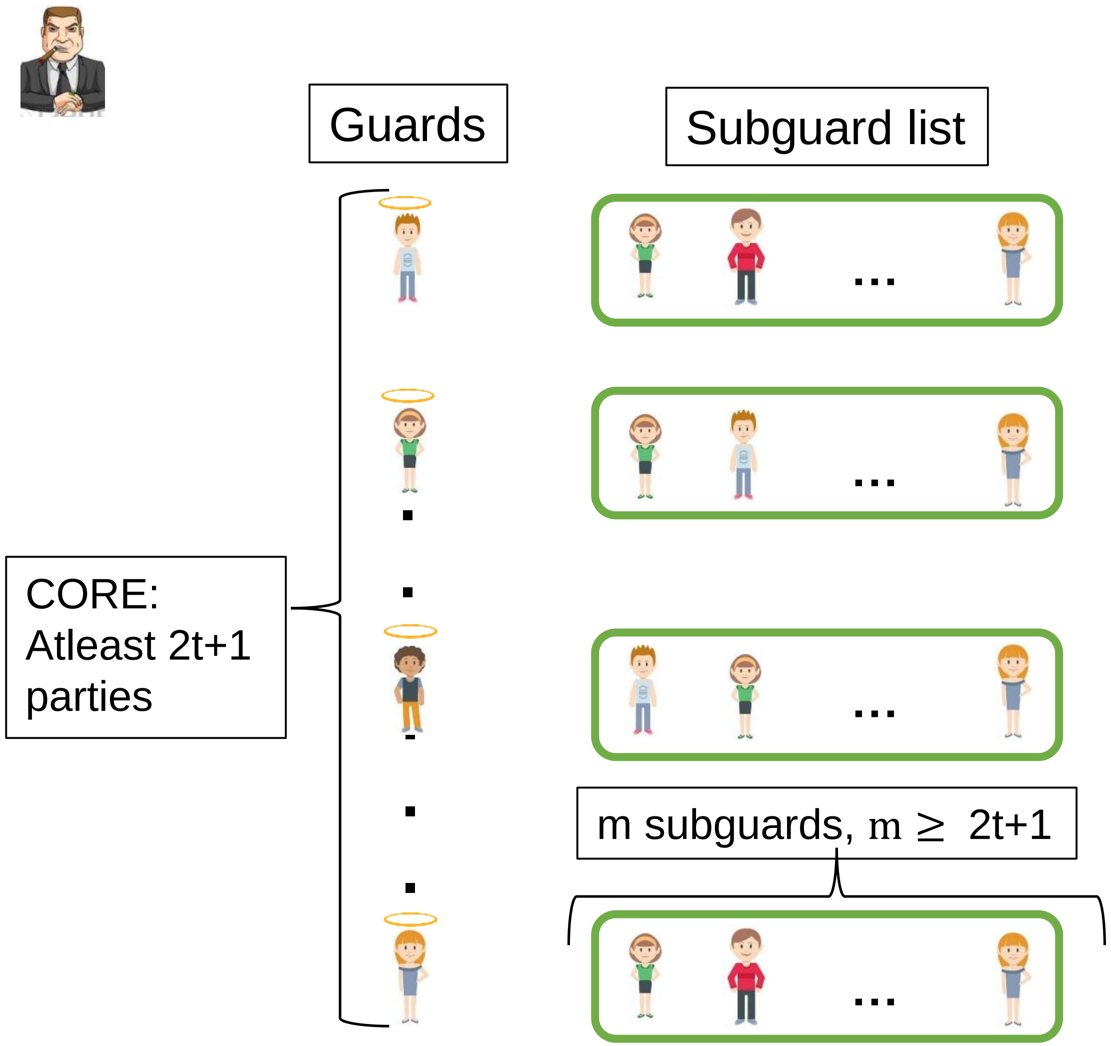
- Each P_i publicly confirms the consistency of its s_i shares
- P_j is a **subguard** for **guard** P_i if P_i broadcasts (OK, j)

SAVSS: Pair-wise Consistency Check



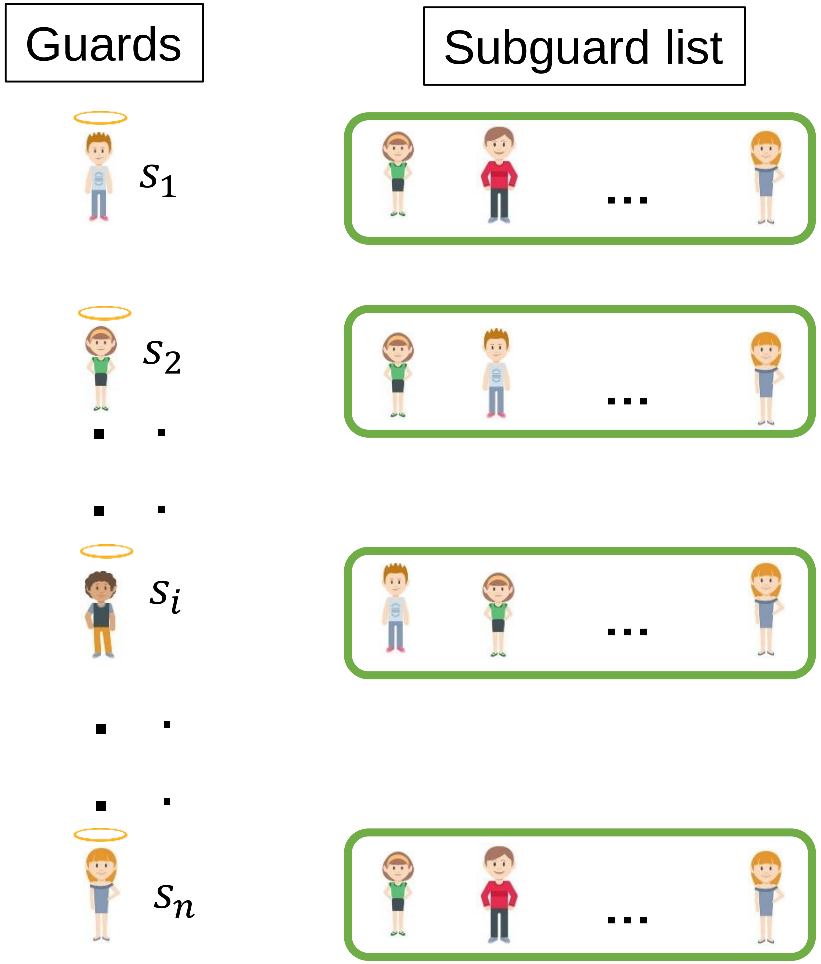
- Each P_i publicly confirms the consistency of its s_i shares
- P_j is **subguard** for **guard** P_i if P_i broadcasts (OK, j)

SAVSS: Identifying the CORE Set



- D identifies a set CORE consisting of at least 2t + 1 guards.
- Each guard in CORE has at least 2t+1 subguards.

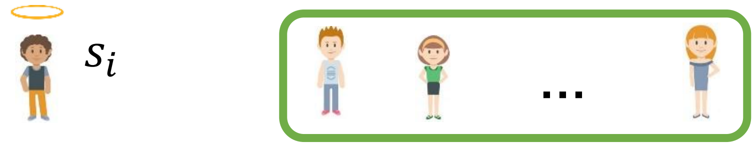
SAVSS: Reconstructing the Secret s



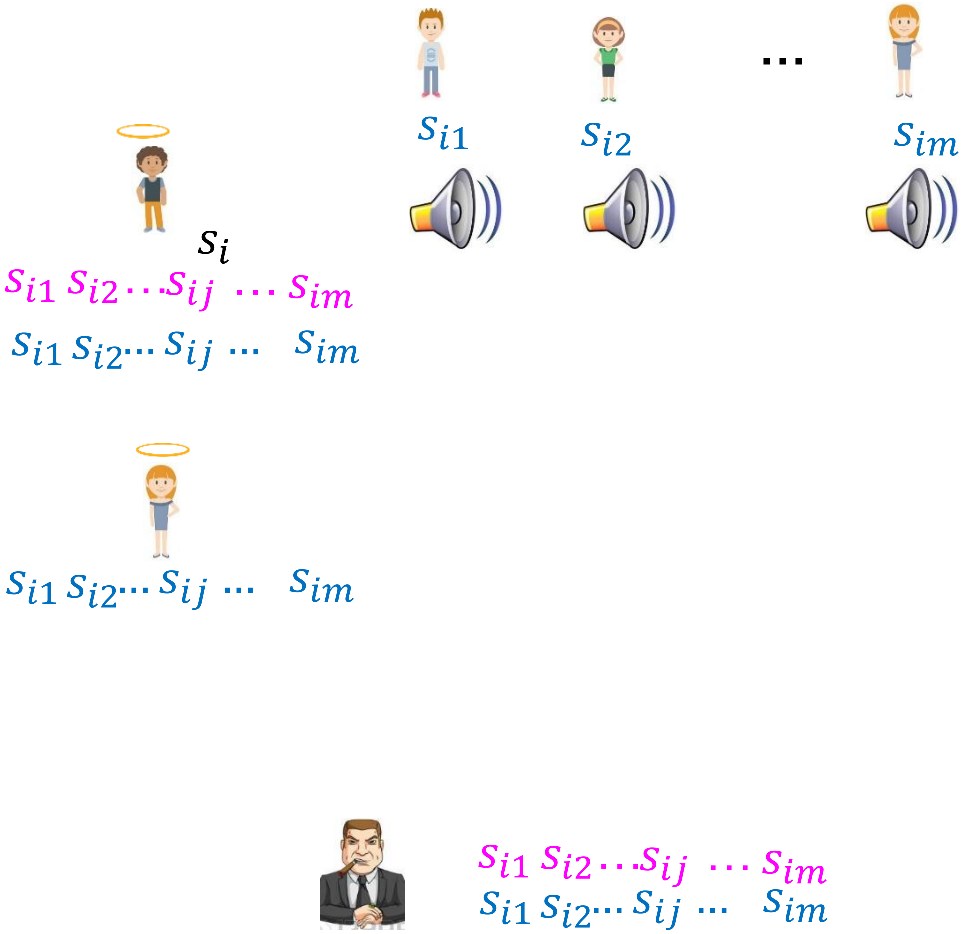
Reconstructing the secret s reduces to reconstructing the shares $s_1, s_2, \ldots, s_n$.

SAVSS: Reconstructing the Share s_i

Reconstructing secret s reduces to reconstructing $s_1, s_2, \ldots, s_n$.



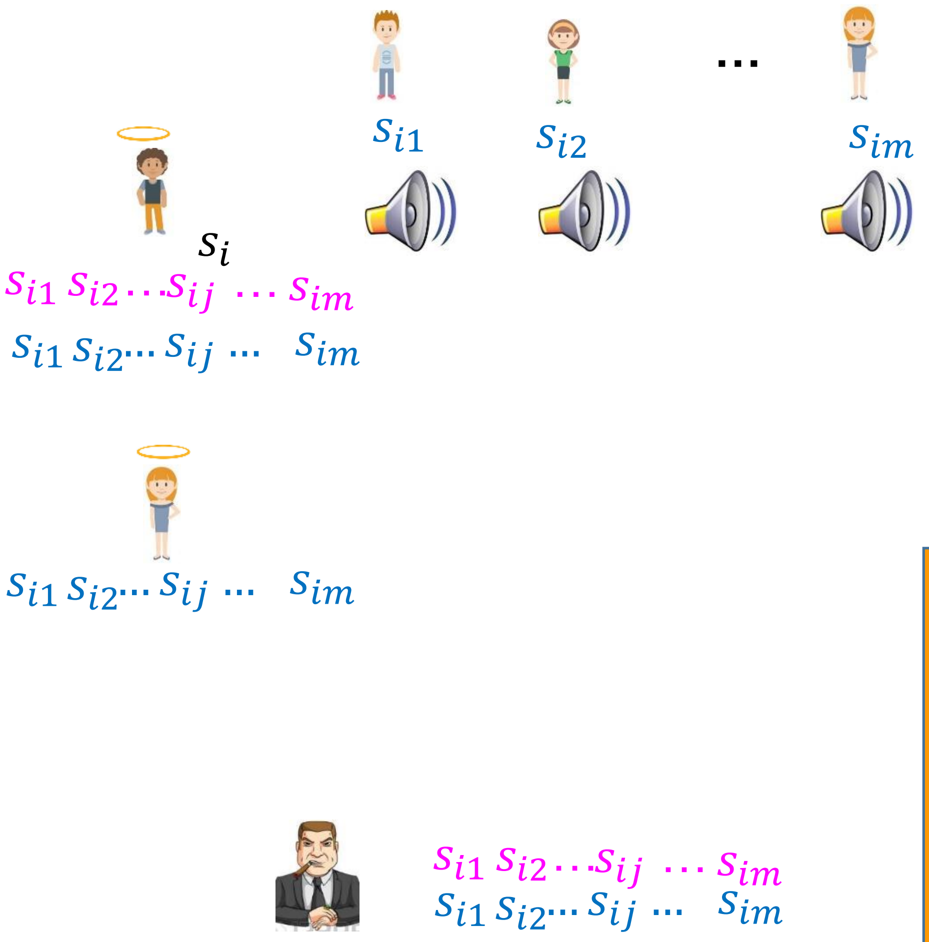
SAVSS: Reconstructing the Share s_i



Goal: To reconstruct the share s_i:

- Wait for any $\frac{3t}{2} + 1$ shares of shares from the subguards.
- Apply Reed Solomon Error Correction on these received shares:
  - Codeword size = $\frac{3t}{2} + 1$
  - t-degree polynomial
  - Corrects upto $\frac{t}{4}$ errors

Intuition Behind Proof : Termination

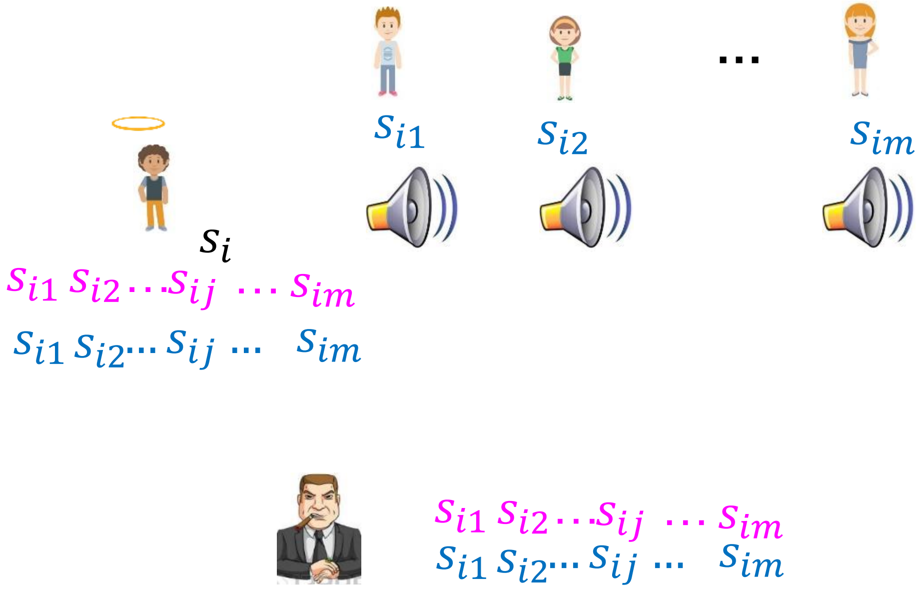


Goal: To reconstruct the share secret s_i:

- Wait for any $\frac{3t}{2} + 1$ shares of shares from the subguards.
- Apply Reed Solomon Error Correction on these

- ❑ Termination fails if more than $\frac{t}{2}$ subguards don’t broadcast their shares
- ❑ Each honest party suspects $\left(\frac{t}{2} + 1\right)$ corrupt parties
- ❑ (n-t)$\left(\frac{t}{2} + 1\right)$ conflicts occur.

Intuition Behind Proof : Correctness

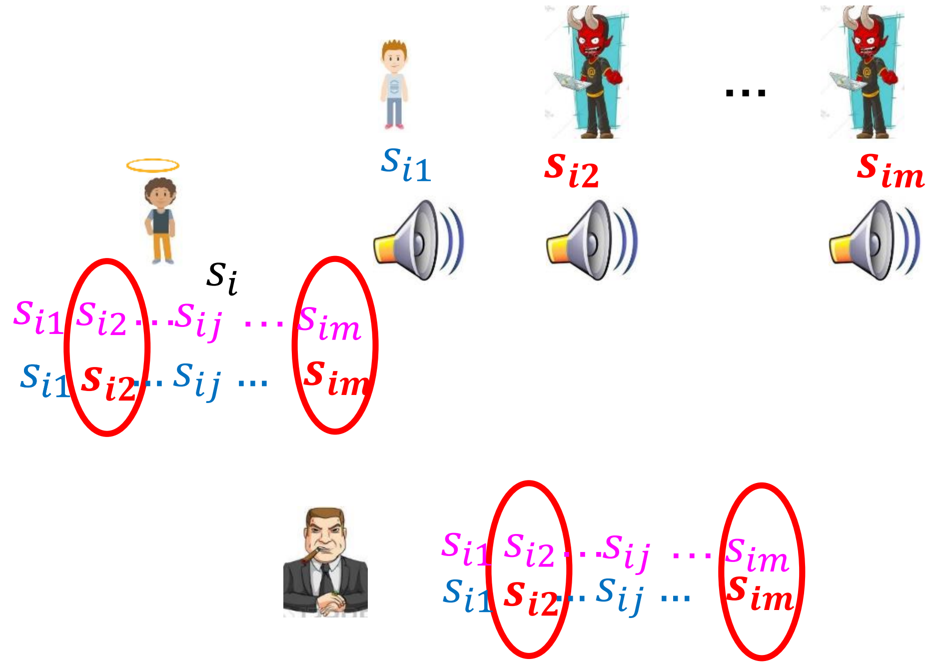


What if $> \frac{t}{4}$ incorrect sub-shares are broadcasted?

Goal: To reconstruct the share s_i:

- Wait for any $\frac{3t}{2} + 1$ shares of shares from the subguards.
- Apply Reed Solomon Error Correction on these received shares:
  - Codeword size = $\frac{3t}{2} + 1$
  - t-degree polynomial
  - Correct upto $\frac{t}{4}$ errors

Intuition Behind Proof : Correctness



Wrong s_i is reconstructed if more than $\frac{t}{4}$+1 subguards broadcast incorrect values.

Honest or → $\frac{t}{4}$+1 conflicts occur

Corrupt and → $\frac{t}{4}$+1 conflicts occur

What if $> \frac{t}{4}$ incorrect sub-shares are broadcasted?

HOW?

Sharing the secret using t-degree Bivariate polynomials
+
Additional requirements on CORE

Conclusion

- A new **optimally resilient** (t < n/3) almost-surely terminating asynchronous Byzantine agreement (ABA) protocol with a **linear** $(\mathcal{O}(n))$ expected rounds
  - [Abraham et. al, PODC, 2008] $\mathcal{O}(n^2)$ expected rounds
  - [Wang, CoRR, 2015] **linear expected rounds**, but **exponential computation complexity**
- **Efficient communication complexity**: Improves over
  - [Abraham et. al, PODC, 2008] by $\mathcal{O}(n^4)$ bits
  - [Wang, CoRR, 2015] by $\mathcal{O}(n)$ bits
- Future work and open problems
  - Almost-surely terminating ABA with a **constant expected running time**
  - Almost-surely terminating ABA with an **improved communication complexity**
  - Almost-surely terminating ABA in the **full-information model**


WAKE UP CALL


Thank you!
Pusheen.com