MODULE *ShamirSecretSharing*

Sepcification for simple *Shamir* Secret Sharing. This is not a veriable secret sharing scheme.

We specify that dealer first sends shares to all players, and once all players have received their shares the can eventually reconstruct the secret.

We do not deal with the communication protocol between players to send their shares to each other before reconstructing the secret.

We use a trick from https://*github.com*/tlaplus/Examples/blob/master/specifications/*ewd*840/*SyncTerminationDetection.tla* to detect that all players have reconstructed the secret and we have detected it

EXTENDS *Integers*, *Sequences*, *Reals*, *TLC*

CONSTANT

- *Dealer*, The dealer sharing the secret with the players
- *Players*, Set of all players
- *Coefficients* The coefficient of the polynomial. These are provided by the model

VARIABLES

- *shares*, Function mapping Player to computed shares
- *shares_sent*, Function mapping Player to shares received
- *shares_received*, Function mapping Player to received shares
- *reconstructed*, Function mapping Player to flag if secret has been successfully constructed
- *allReconstructDetected* We detected all reconstructions and can therefore terminate

$vars \triangleq \langle shares,\ shares\_sent,\ shares\_received,\ reconstructed,\ allReconstructDetected \rangle$

$NoValue \triangleq -1$

$Init \triangleq$

Compute shares as a $+ bx + cx\hat{\ }2$

$\land\ shares = [p \in Players \mapsto Coefficients[1] + Coefficients[2] * p + Coefficients[3] * p^2]$

$\land\ shares\_sent = [p \in Players \mapsto NoValue]$

$\land\ shares\_received = [p \in Players \mapsto NoValue]$

$\land\ reconstructed = [p \in Players \mapsto \text{FALSE}]$

$\land\ allReconstructDetected = \text{FALSE}$

The type invariant for all variables.

$TypeOK \triangleq$

$\land\ shares \in [Players \rightarrow Int]$

$\land\ shares\_sent \in [Players \rightarrow Int]$

$\land\ shares\_received \in [Players \rightarrow Int]$

$\land\ reconstructed \in [Players \rightarrow \text{BOOLEAN}]$

$\land\ allReconstructDetected \in \text{BOOLEAN}$

$allReconstructed \triangleq \forall\, p \in Players : reconstructed[p]$

Send the share to Player p.

```
$SendShare(p) \triangleq$
    $\land\ shares\_sent[p] = NoValue$
     Send a share that has not been sent to anyone
    $\land\ shares\_sent' = [shares\_sent \text{ EXCEPT } ![p] = shares[p]]$
    $\land\ \text{UNCHANGED } \langle shares, shares\_received, reconstructed, allReconstructDetected \rangle$
```

Receive the share at Player p. It should have been sent before.

```
$ReceiveShare(p) \triangleq$
    $\land\ shares\_received[p] = NoValue$
    $\land\ shares\_sent[p] \neq NoValue$
    $\land\ shares\_received' = [shares\_received \text{ EXCEPT } ![p] = shares\_sent[p]]$
    $\land\ \text{UNCHANGED } \langle shares, shares\_sent, reconstructed, allReconstructDetected \rangle$
```

Reconstruct secret with *Players* p and q. The payers should have receieved share.

```
$Reconstruct(p, q) \triangleq$
    $\land\ \ \forall t \in Players : shares\_received[t] \neq NoValue$
    $\land\ \ p \neq q$
    $\land\ \ shares\_received[p] \neq NoValue$
    $\land\ \ shares\_received[q] \neq NoValue$
    $\land\ \ reconstructed[p] = \text{FALSE}$
     We don't specify how the secret is reconstructed, just that it is
     reconstructed using shares of all two player combinations
    $\land\ reconstructed' = [reconstructed \text{ EXCEPT } ![p] = \text{TRUE}]$
    $\land\ allReconstructDetected' \in \{allReconstructDetected, allReconstructed'\}$
    $\land\ \text{UNCHANGED } \langle shares, shares\_sent, shares\_received \rangle$
```

```
$DetectReconstructed \triangleq$
    $\land\ allReconstructed$
    $\land\ allReconstructDetected' = \text{TRUE}$
    $\land\ \text{UNCHANGED } \langle shares, shares\_sent, shares\_received, reconstructed \rangle$
```

The next step either sends shares, receieves them or reconstructs the secret.

```
$Next \triangleq \exists p, q \in Players :$
        $\lor\ SendShare(p)$
        $\lor\ ReceiveShare(p)$
        $\lor\ Reconstruct(p, q)$
        $\lor\ DetectReconstructed$

$Spec \triangleq$
    $\land\ Init$
```

$$\land \Box[\mathit{Next}]_{\mathit{vars}}$$

Liveness states that eventually all players reconstruct the secret.

$$\begin{aligned}&\mathit{Liveness} \triangleq \forall\, p,\ q \in \mathit{Players} :\\ &\qquad \mathrm{WF}_{\mathit{vars}}(\mathit{ReceiveShare}(p) \land \mathit{Reconstruct}(p,\ q) \land \mathit{DetectReconstructed})\end{aligned}$$

Stability - once all reconstructions are detected, all *Players′* secrets
remain reconstructed.

$$\mathit{Stable} \triangleq \Box(\mathit{allReconstructDetected} \Rightarrow \Box\mathit{allReconstructed})$$

For a fair specification, we assure the spec takes next steps and liveness is guaranteed.

$$\mathit{FairSpec} \triangleq \mathit{Spec} \land \mathit{Liveness}$$