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.

EXTENDS *Integers*, *Sequences*

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

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

$NoValue \triangleq -1$

$Init \triangleq$

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

$\land\ shares = [p \in Players \mapsto Coefficients[1] + Coefficients[2] * p + Coefficients[3] * p^2]$
$\land\ shares\_sent = [p \in Players \mapsto NoValue]$
$\land\ shares\_received = [p \in Players \mapsto NoValue]$
$\land\ reconstructed = [p \in Players \mapsto \text{FALSE}]$

The type invariant for all variables.

$TypeOK \triangleq$

$\land\ shares \in [Players \to Int]$
$\land\ shares\_sent \in [Players \to Int]$
$\land\ shares\_received \in [Players \to Int]$
$\land\ reconstructed \in [Players \to \text{BOOLEAN }]$

Send the share to Player p.

$SendShare(p) \triangleq$

$\land\ shares\_sent[p] = NoValue$

Send a share that has not been sent to anyone

$\land\ shares\_sent' = [shares\_sent \text{ EXCEPT } ![p] = shares[p]]$
$\land\ \text{UNCHANGED } \langle shares, shares\_received, reconstructed \rangle$

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

$$
\begin{aligned}
&\mathit{ReceiveShare}(p) \triangleq \\
&\quad \land \mathit{shares\_received}[p] = \mathit{NoValue} \\
&\quad \land \mathit{shares\_sent}[p] \neq \mathit{NoValue} \\
&\quad \land \mathit{shares\_received}' = [\mathit{shares\_received} \text{ EXCEPT } ![p] = \mathit{shares\_sent}[p]] \\
&\quad \land \text{UNCHANGED } \langle \mathit{shares}, \mathit{shares\_sent}, \mathit{reconstructed} \rangle
\end{aligned}
$$

Reconstruct secret at Player p. It should have been received

$$
\begin{aligned}
&\mathit{Reconstruct}(p) \triangleq \\
&\quad \land \quad \mathit{shares\_received}[p] \neq \mathit{NoValue} \\
&\quad \land \quad \mathit{reconstructed}' = [\mathit{reconstructed} \text{ EXCEPT } ![p] = \text{TRUE}] \\
&\quad \land \quad \text{UNCHANGED } \langle \mathit{shares}, \mathit{shares\_sent}, \mathit{shares\_received} \rangle
\end{aligned}
$$

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

$$
\begin{aligned}
&\mathit{Next} \triangleq \\
&\quad \lor \exists\, p \in \mathit{Players} : \\
&\qquad \mathit{SendShare}(p) \lor \mathit{ReceiveShare}(p) \lor \mathit{Reconstruct}(p)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{Spec} \triangleq \\
&\quad \land \mathit{Init} \\
&\quad \land \Box[\mathit{Next}]_{\mathit{vars}}
\end{aligned}
$$

Liveness states that eventually all players reconstruct the secret.

$$\mathit{Liveness} \triangleq \forall\, p \in \mathit{Players} : \mathrm{WF}_{\mathit{vars}}(\mathit{Reconstruct}(p))$$

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

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

\ * Modification History
\ * Last modified *Tue Jun* 13 21:26:31 *CEST* 2023 by *kulpreet*
\ * Created *Fri Jun* 09 17:03:07 *CEST* 2023 by *kulpreet*