E-Voting with Commitments

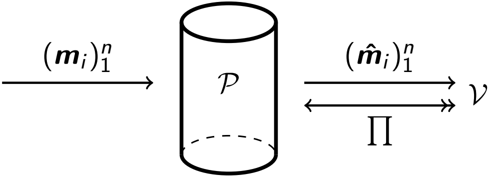
Thor Tunge

May 11, 2019

Introduction

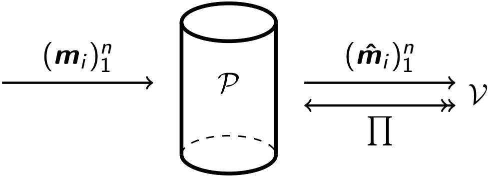
- Voting model
- Shuffling votes
- Proof of shuffling
- Add commitments

Voting Model



- Ordered set of messages as input
- (Permuted) ordered set as output
- Proof that $\hat{m}_i = m_{\pi(i)}$

Voting Model

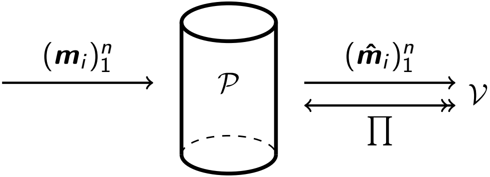


- Ordered set of messages as input
- (Permuted) ordered set as output
- Proof that $\hat{m}_i = m_{\pi(i)}$

A cheating $\mathcal{P}$ wants to change a message m_k such that no permutation exists

$$\pi : (m_i)_1^n \to (\hat{m}_i)_1^n$$

Voting Model



- Public: $(\hat{m}_i)_1^n$ and $(m_i)_1^n$
- Secret: Permutation π
- Will create an interactive protocol between $\mathcal{P}$ and $\mathcal{V}$

Intermezzo: Polynomials

- Pick two polynomials $f, \hat{f}$ at random (of degree n)
- Pick a random number ρ
- How likely is $f(\rho) = \hat{f}(\rho)$

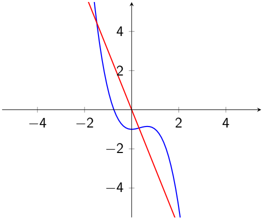


Figure: f and $\hat{f}$

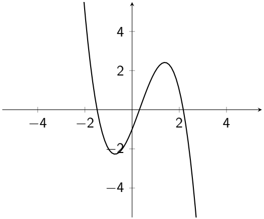


Figure: $f - \hat{f}$

$$f(\rho) - \hat{f}(\rho) = 0 \Rightarrow f = \hat{f}$$

Scetch

- Define $f(X) = \prod(m_i - X), \quad \hat{f}(X) = \prod(\hat{m}_i - X)$
- Construct linear system where solution exists if

$$f(\rho) - \hat{f}(\rho) = 0$$

- Require that $\mathcal{P}$ provides a solution

Scetch

- Define $f(X) = \prod(m_i - X), \quad \hat{f}(X) = \prod(\hat{m}_i - X)$
- Construct linear system where solution exists if

$$f(\rho) - \hat{f}(\rho) = 0$$

- Require that $\mathcal{P}$ provides a solution
- $\mathcal{P}$ proves that they know π to a verifier $\mathcal{V}$

Notation: $M_i = m_i - \rho$ and $\hat{M}_i = \hat{m}_i - \rho$

Interactive Protocol

1. $\mathcal{V}$ picks a random ρ
2. Both compute M_i and $\hat{M}_i$
3. $\mathcal{P}$ picks random $(\theta_i)_1^{n-1}$ and computes $\theta_{k-1}M_k + \theta_k\hat{M}_k$
4. $\mathcal{P}$ sends $\theta_{k-1}M_k + \theta_k\hat{M}_k$ to $\mathcal{V}$
5. $\mathcal{V}$ sends a challenge β
6. $\mathcal{P}$ has to determine s_i

Interactive Protocol

1. $\mathcal{V}$ picks a random ρ
2. Both compute M_i and $\hat{M}_i$
3. $\mathcal{P}$ picks random $(\theta_i)_1^{n-1}$ and computes $\theta_{k-1}M_k + \theta_k\hat{M}_k$
4. $\mathcal{P}$ sends $\theta_{k-1}M_k + \theta_k\hat{M}_k$ to $\mathcal{V}$
5. $\mathcal{V}$ sends a challenge β
6. $\mathcal{P}$ has to determine s_i

$$
\begin{aligned}
\beta M_1 + s_1\hat{M}_1 &= \theta_1\hat{M}_1 \\
s_1M_2 + s_2\hat{M}_2 &= \theta_1M_2 + \theta_2\hat{M}_2 \\
&\vdots \\
s_{n-2}M_{n-1} + s_{n-1}\hat{M}_{n-1} &= \theta_{n-2}M_{n-2} + \theta_{n-1}\hat{M}_{n-1} \\
(-1)^n\beta\hat{M}_n + s_{n-1}M_n &= \theta_{n-1}M_n
\end{aligned}
$$

Interactive Protocol

1. $\mathcal{V}$ picks a random ρ
2. Both compute M_i and $\hat{M}_i$
3. $\mathcal{P}$ picks random $(\theta_i)_1^{n-1}$ and computes $\theta_{k-1} M_k + \theta_k \hat{M}_k$
4. $\mathcal{P}$ sends $\theta_{k-1} M_k + \theta_k \hat{M}_k$ to $\mathcal{V}$
5. $\mathcal{V}$ sends a challenge β
6. $\mathcal{P}$ has to determine s_i

$$
\begin{aligned}
s_1 \hat{M}_1 &= \theta_1 \hat{M}_1 \\
s_1 M_2 + s_2 \hat{M}_2 &= \theta_1 M_2 + \theta_2 \hat{M}_2 \\
&\vdots \\
s_{n-2} M_{n-1} + s_{n-1} \hat{M}_{n-1} &= \theta_{n-2} M_{n-2} + \theta_{n-1} \hat{M}_{n-1} \\
s_{n-1} M_n &= \theta_{n-1} M_n
\end{aligned}
$$

Interactive Protocol

1. $\mathcal{V}$ picks a random ρ
2. Both compute M_i and $\hat{M}_i$
3. $\mathcal{P}$ picks random $(\theta_i)_1^{n-1}$ and computes $\theta_{k-1}M_k + \theta_k\hat{M}_k$
4. $\mathcal{P}$ sends $\theta_{k-1}M_k + \theta_k\hat{M}_k$ to $\mathcal{V}$
5. $\mathcal{V}$ sends a challenge β
6. $\mathcal{P}$ has to determine s_i

$$\begin{aligned}
\beta M_1 + s_1\hat{M}_1 &= \theta_1\hat{M}_1 \\
s_1 M_2 + s_2\hat{M}_2 &= \theta_1 M_2 + \theta_2\hat{M}_2 \\
&\vdots \\
s_{n-2}M_{n-1} + s_{n-1}\hat{M}_{n-1} &= \theta_{n-2}M_{n-2} + \theta_{n-1}\hat{M}_{n-1} \\
(-1)^n\beta\hat{M}_n + s_{n-1}M_n &= \theta_{n-1}M_n
\end{aligned}$$

The linear system

$$\underbrace{\begin{pmatrix} \hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\ M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\ 0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\ 0 & 0 & 0 & \dots & 0 & M_n \end{pmatrix}}_{\boldsymbol{M}} \underbrace{\begin{pmatrix} s_1 - \theta_1 \\ s_2 - \theta_2 \\ s_3 - \theta_3 \\ \vdots \\ s_{n-2} - \theta_{n-2} \\ s_{n-1} - \theta_{n-1} \end{pmatrix}}_{\boldsymbol{s}} = \underbrace{\begin{pmatrix} -\beta M_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ (-1)^{n-1}\beta \hat{M}_n \end{pmatrix}}_{\boldsymbol{b}}$$

$$\boldsymbol{Ms} = \boldsymbol{b}$$

is a $n \times (n-1)$ system of linear equations. Over-determined.

Want to show that $\boldsymbol{b}$ is in the span of $\boldsymbol{M}$ if shuffle is done correctly.

To show that $\boldsymbol{b}$ is in the span of $\boldsymbol{M}$, append column $\boldsymbol{b}$ to $\boldsymbol{M}$

$$\underbrace{\begin{pmatrix} \beta M_1 & \hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\ \beta(-1)^n \hat{M}_n & 0 & 0 & 0 & \dots & 0 & M_n \end{pmatrix}}_{\boldsymbol{b}||\boldsymbol{M}}.$$

- Square $n \times n$ matrix

To show that $\boldsymbol{b}$ is in the span of $\boldsymbol{M}$, append column $\boldsymbol{b}$ to $\boldsymbol{M}$

$$\underbrace{\begin{pmatrix} \beta M_1 & \hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\ \beta(-1)^n \hat{M}_n & 0 & 0 & 0 & \dots & 0 & M_n \end{pmatrix}}_{\boldsymbol{b}||\boldsymbol{M}}.$$

- Square $n \times n$ matrix
- $\boldsymbol{M}$ has linearly independent vectors

To show that $\boldsymbol{b}$ is in the span of $\boldsymbol{M}$, append column $\boldsymbol{b}$ to $\boldsymbol{M}$

$$\underbrace{\begin{pmatrix} \beta M_1 & \hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\ \beta(-1)^n \hat{M}_n & 0 & 0 & 0 & \dots & 0 & M_n \end{pmatrix}}_{\boldsymbol{b}||\boldsymbol{M}}.$$

- Square $n \times n$ matrix
- $\boldsymbol{M}$ has linearly independent vectors
- $\det(\boldsymbol{b}||\boldsymbol{M}) = \beta \left(\prod_{i=1}^{n} M_i - \prod_{i=1}^{n} \hat{M}_i \right) = \beta(f(\rho) - \hat{f}(\rho))$

To show that $\boldsymbol{b}$ is in the span of $\boldsymbol{M}$, append column $\boldsymbol{b}$ to $\boldsymbol{M}$

$$\underbrace{\begin{pmatrix} \beta M_1 & \hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\ \beta(-1)^n \hat{M}_n & 0 & 0 & 0 & \dots & 0 & M_n \end{pmatrix}}_{\boldsymbol{b}||\boldsymbol{M}}.$$

- Square $n \times n$ matrix
- $\boldsymbol{M}$ has linearly independent vectors
- $\det(\boldsymbol{b}||\boldsymbol{M}) = \beta\left(\prod_{i=1}^{n} M_i - \prod_{i=1}^{n} \hat{M}_i\right) = \beta(f(\rho) - \hat{f}(\rho))$
- $\det(\boldsymbol{b}||\boldsymbol{M}) = 0 \iff \boldsymbol{b}$ in the span of $\boldsymbol{M}$

The linear system

$$
\underbrace{\begin{pmatrix}
\hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\
M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\
0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\
0 & 0 & 0 & \dots & 0 & M_n
\end{pmatrix}}_{\boldsymbol{M}}
\underbrace{\begin{pmatrix}
s_1 - \theta_1 \\
s_2 - \theta_2 \\
s_3 - \theta_3 \\
\vdots \\
s_{n-2} - \theta_{n-2} \\
s_{n-1} - \theta_{n-1}
\end{pmatrix}}_{\boldsymbol{s}}
=
\underbrace{\begin{pmatrix}
-\beta M_1 \\
0 \\
0 \\
\vdots \\
0 \\
(-1)^{n-1}\beta \hat{M}_n
\end{pmatrix}}_{\boldsymbol{b}}
$$

- This system has a solution iff $f(\rho) = \hat{f}(\rho)$

The linear system

$$\underbrace{\begin{pmatrix} \hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\ M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\ 0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\ 0 & 0 & 0 & \dots & 0 & M_n \end{pmatrix}}_{\boldsymbol{M}} \underbrace{\begin{pmatrix} s_1 - \theta_1 \\ s_2 - \theta_2 \\ s_3 - \theta_3 \\ \vdots \\ s_{n-2} - \theta_{n-2} \\ s_{n-1} - \theta_{n-1} \end{pmatrix}}_{\boldsymbol{s}} = \underbrace{\begin{pmatrix} -\beta M_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ (-1)^{n-1}\beta \hat{M}_n \end{pmatrix}}_{\boldsymbol{b}}$$

- This system has a solution iff $f(\rho) = \hat{f}(\rho)$
- Shuffle done honestly $\Rightarrow f(X) = \hat{f}(X) \Rightarrow f(\rho) = \hat{f}(\rho)$

The linear system

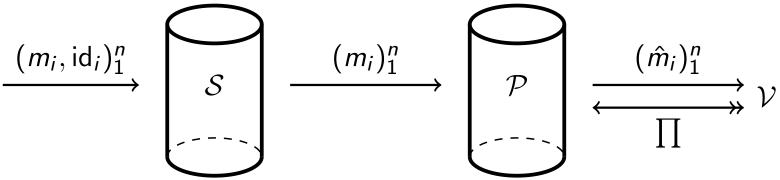
$$\underbrace{\begin{pmatrix} \hat{M}_1 & 0 & 0 & \dots & 0 & 0 \\ M_2 & \hat{M}_2 & 0 & \dots & 0 & 0 \\ 0 & M_3 & \hat{M}_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & M_{n-1} & \hat{M}_{n-1} \\ 0 & 0 & 0 & \dots & 0 & M_n \end{pmatrix}}_{\boldsymbol{M}} \underbrace{\begin{pmatrix} s_1 - \theta_1 \\ s_2 - \theta_2 \\ s_3 - \theta_3 \\ \vdots \\ s_{n-2} - \theta_{n-2} \\ s_{n-1} - \theta_{n-1} \end{pmatrix}}_{\boldsymbol{s}} = \underbrace{\begin{pmatrix} -\beta M_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ (-1)^{n-1}\beta \hat{M}_n \end{pmatrix}}_{\boldsymbol{b}}$$

- This system has a solution iff $f(\rho) = \hat{f}(\rho)$
- Shuffle done honestly $\Rightarrow f(X) = \hat{f}(X) \Rightarrow f(\rho) = \hat{f}(\rho)$
- If permutation does not exist $\Rightarrow f(\rho) = \hat{f}(\rho)$ negligible

Recap: Shuffling Proof

- $\mathcal{V}$ picks a random $\rho \rightarrow$
- Both compute $\hat{M}_i = \hat{m}_i - \rho$, $\mathcal{P}$ computes $M_i = m_i - \rho$
- $\mathcal{P}$ picks θ_i and computes $\theta_{k-1} M_k + \theta_k \hat{M}_k \rightarrow$
- $\mathcal{V}$ picks a challenge $\beta \rightarrow$
- $\mathcal{P}$ solves linear system $\rightarrow$
- $\mathcal{V}$ verifies the linear system

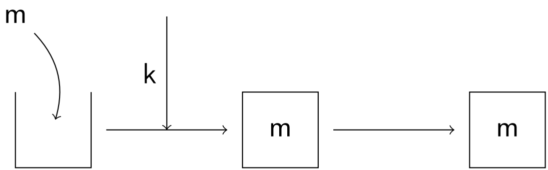
Problem: The Whole Picture



Server $\mathcal{S}$ can recover the permutation π!

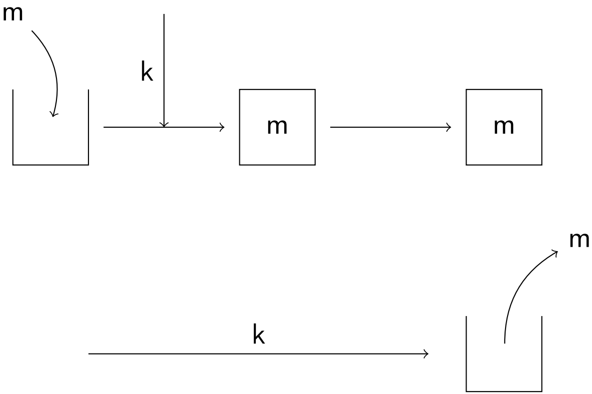
Commitments

We want to commit to a value, and later reveal which value we commited to



Commitments

We want to commit to a value, and later reveal which value we commited to



Commitments

- 3 algorithms: $\texttt{Keygen}, \texttt{Commit}, \texttt{Verify}$
- Commit to a value m using randomness r:

$$\texttt{Commit}(m, r) := [m; r]$$

- Send commitment $[m; r]$.
- Later, reveal (m, r) by sending it
- Verifier can check that $[m; r] = \texttt{Commit}(m, r)$

Commitments

- 3 algorithms: $\texttt{Keygen}, \texttt{Commit}, \texttt{Verify}$
- Commit to a value m using randomness r:

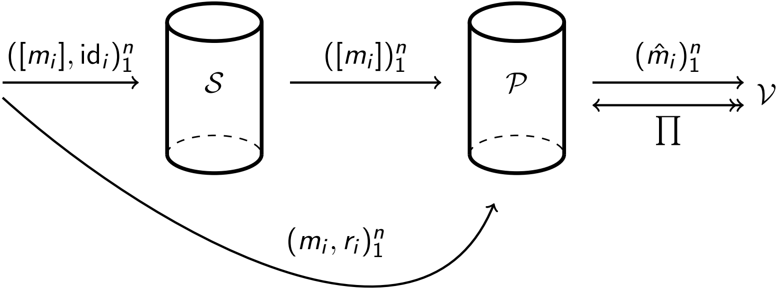
$$\texttt{Commit}(m, r) := [m; r]$$

- Send commitment $[m; r]$.
- Later, reveal (m, r) by sending it
- Verifier can check that $[m; r] = \texttt{Commit}(m, r)$
- Hiding - Verifier cannot open before (m, r) is sent.
- Binding - Prover cannot send different (m', r') such that

$$\texttt{Commit}(m', r') = \texttt{Commit}(m, r) \quad m \neq m'$$

Solution: Commitments!

Add commitments to the picture!



Public: $([m])_1^n$ and $(\hat{m})_1^n$
Secret: permutation π

What did we break?

- ✓ $\mathcal{V}$ picks a random $\rho \rightarrow$
- ✗ Both compute $\hat{M}_i$ and $\hat{M}_i$
- ✗ $\mathcal{P}$ picks θ_i and computes $\theta_{k-1} M_k + \theta_k \hat{M}_k \rightarrow$
- ✓ $\mathcal{V}$ picks a challenge $\beta \rightarrow$
- ✓ $\mathcal{P}$ solves linear system by determining $s_i \rightarrow$
- ✗ $\mathcal{V}$ verifies the linear system

What did we break?

- ✓ $\mathcal{V}$ picks a random $\rho \rightarrow$
- ✗ Both compute $\hat{M}_i$ and $\hat{M}_i$
- ✗ $\mathcal{P}$ picks θ_i and computes $\theta_{k-1} M_k + \theta_k \hat{M}_k \rightarrow$
- ✓ $\mathcal{V}$ picks a challenge $\beta \rightarrow$
- ✓ $\mathcal{P}$ solves linear system by determining $s_i \rightarrow$
- ✗ $\mathcal{V}$ verifies the linear system

What did we break?

- ✓ $\mathcal{V}$ picks a random $\rho \longrightarrow$
- ✓ Both compute $\hat{M}_i$ and $\hat{M}_i$
- ✗ $\mathcal{P}$ picks θ_i and computes $\theta_{k-1} M_k + \theta_k \hat{M}_k \longrightarrow$
- ✓ $\mathcal{V}$ picks a challenge $\beta \longrightarrow$
- ✓ $\mathcal{P}$ solves linear system by determining $s_i \longrightarrow$
- ✗ $\mathcal{V}$ verifies the linear system

What did we break?

- ✓ $\mathcal{V}$ picks a random $\rho \rightarrow$
- ✓ Both compute $\hat{M}_i$ and $\hat{M}_i$
- ✗ $\mathcal{P}$ picks θ_i and computes $\theta_{k-1} M_k + \theta_k \hat{M}_k \rightarrow$
- ✓ $\mathcal{V}$ picks a challenge $\beta \rightarrow$
- ✓ $\mathcal{P}$ solves linear system by determining $s_i \rightarrow$
- ✗ $\mathcal{V}$ verifies the linear system

What did we break?

- ✓ $\mathcal{V}$ picks a random $\rho \rightarrow$
- ✓ Both compute $\hat{M}_i$ and $\hat{M}_i$
- ✓ $\mathcal{P}$ picks θ_i and computes $[\theta_{k-1} M_k + \theta_k \hat{M}_k] \rightarrow$
- ✓ $\mathcal{V}$ picks a challenge $\beta \rightarrow$
- ✓ $\mathcal{P}$ solves linear system by determining $s_i \rightarrow$
- ✗ $\mathcal{V}$ verifies the linear system

What did we break?

- ✓ $\mathcal{V}$ picks a random $\rho \rightarrow$
- ✓ Both compute $\hat{M}_i$ and $\hat{M}_i$
- ✓ $\mathcal{P}$ picks θ_i and computes $[\theta_{k-1} M_k + \theta_k \hat{M}_k] \rightarrow$
- ✓ $\mathcal{V}$ picks a challenge $\beta \rightarrow$
- ✓ $\mathcal{P}$ solves linear system by determining $s_i \rightarrow$
- ✗ $\mathcal{V}$ verifies the linear system

How to Verify

$\mathcal{V}$ is supposed to verfy that

$$\begin{aligned}
\beta M_1 + s_1 \hat{M}_1 &= \theta_1 \hat{M}_1 \\
s_1 M_2 + s_2 \hat{M}_2 &= \theta_1 M_2 + \theta_2 \hat{M}_2 \\
&\vdots \\
s_{n-2} M_{n-1} + s_{n-1} \hat{M}_{n-1} &= \theta_{n-2} M_{n-2} + \theta_{n-1} \hat{M}_{n-1} \\
(-1)^n \beta \hat{M}_n + s_{n-1} M_n &= \theta_{n-1} M_n
\end{aligned}$$

is satisfied.

How to Verify

But $\mathcal{V}$ can only see this mess

$$
\begin{aligned}
\beta[M_1] + s_1\hat{M}_1 &\neq [\theta_1\hat{M}_1] \\
s_1[M_2] + s_2\hat{M}_2 &\neq [\theta_1 M_2 + \theta_2\hat{M}_2] \\
&\vdots \\
s_{n-2}[M_{n-1}] + s_{n-1}\hat{M}_{n-1} &\neq [\theta_{n-2}M_{n-2} + \theta_{n-1}\hat{M}_{n-1}] \\
(-1)^n\beta\hat{M}_n + s_{n-1}[M_n] &\neq [\theta_{n-1}M_n]
\end{aligned}
$$

How to Verify

Consider the second equation

$$s_1[M_2] + s_2 \ \hat{M}_2 \ \neq [\theta_1 M_2 + \theta_2 \hat{M}_2]$$

How to Verify

Consider the second equation

$$s_1[M_2] + s_2[\hat{M}_2] \neq [\theta_1 M_2 + \theta_2 \hat{M}_2]$$

How to Verify

Consider the second equation

$$s_1[M_2] + s_2[\hat{M}_2] \neq [\theta_1 M_2 + \theta_2 \hat{M}_2]$$

$[\hat{M}_2] = \hat{M}_2$ trivial commitment (not hiding).

How to Verify

Consider the second equation

$$s_1[M_2] + s_2[\hat{M}_2] \neq [\theta_1 M_2 + \theta_2 \hat{M}_2]$$

$[\hat{M}_2] = \hat{M}_2$ trivial commitment (not hiding).
Use ZKPOK to show that the commitments

$$[M_2], [\hat{M}_2], [\theta_1 M_2 + \theta_2 \hat{M}_2]$$

are such that

$$s_1 M_2 + s_2 \hat{M}_2 = \theta_1 M_2 + \theta_2 \hat{M}_2$$

where s_1, s_2 are known to $\mathcal{V}$.

How to Verify

Consider the second equation

$$s_1[M_2] + s_2[\hat{M}_2] \neq [\theta_1 M_2 + \theta_2 \hat{M}_2]$$

$[\hat{M}_2] = \hat{M}_2$ trivial commitment (not hiding).
Use ZKPOK to show that the commitments

$$[M_2], [\hat{M}_2], [\theta_1 M_2 + \theta_2 \hat{M}_2]$$

are such that

$$s_1 M_2 + s_2 \hat{M}_2 = \theta_1 M_2 + \theta_2 \hat{M}_2$$

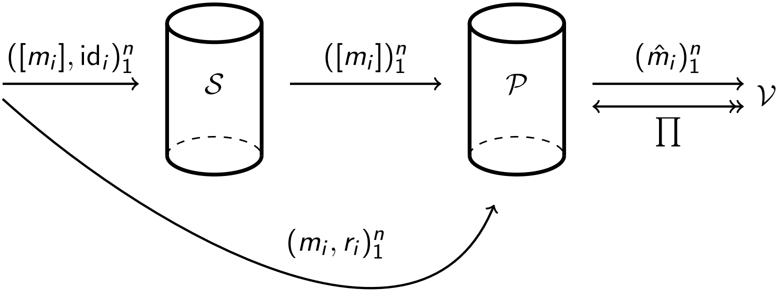
where s_1, s_2 are known to $\mathcal{V}$.

This is exactly the equation $\mathcal{V}$ wants to verify!

What did we fix

- ✓ $\mathcal{V}$ picks a random $\rho \rightarrow \mathcal{P}$
- ✓ Both compute $\hat{M}_i = \hat{m}_i - \rho$, $\mathcal{P}$ computes $M_i = m_i - \rho$
- ✓ $\mathcal{P}$ picks θ_i and computes $[\theta_{k-1} M_k + \theta_k \hat{M}_k] \rightarrow \mathcal{V}$
- ✓ $\mathcal{V}$ picks a challenge $\beta \rightarrow \mathcal{P}$
- ✓ $\mathcal{P}$ solves linear system by determining $s_i \rightarrow \mathcal{V}$
- ✓ $\mathcal{V}$ verifies the linear system *using ZKPOK* for each equation

Conclusion and Additional Work



- Proof of shuffling using commitments
- Verifiable encryption of (m, r)
- Multiple intermediate servers $\mathcal{S}_i$