Asynchronous Verifiable Secret Sharing in Optimal Communication Complexity

Michael Backes
MPI-SWS

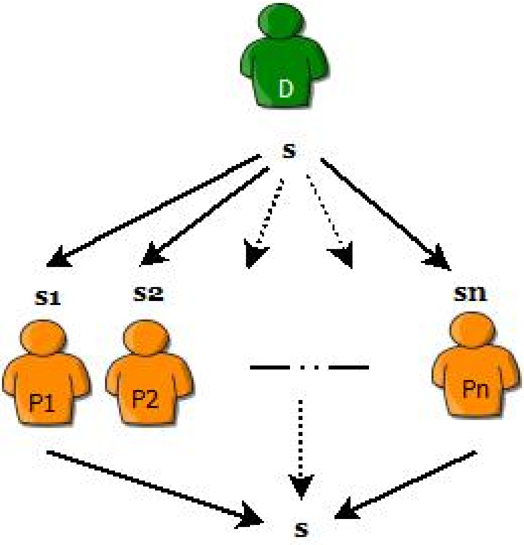
Amit Datta
IIT, Kharagpur

Aniket Kate
MPI-SWS

Outline

- Background: Verifiable Secret Sharing (VSS)
- Asynchronous Verifiable Secret Sharing (AVSS)
- State-of-the-art Protocols
- What we want to Achieve and How
- Our Protocols

An (n,t)-VSS: Sharing and Reconstruction



Asynchronous System Model

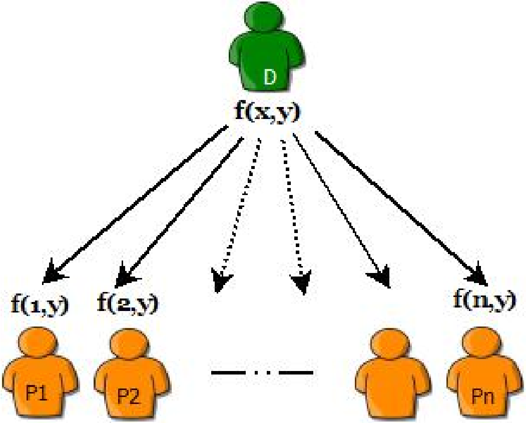
The Adversary:

- Controls the network and may delay messages between any two honest parties
- Cannot read or modify these messages
- Has to eventually deliver all the messages by honest parties
- Can corrupt at most t parties, out of n

In this setting, the optimal resiliency bound is $n \geq 3t + 1$

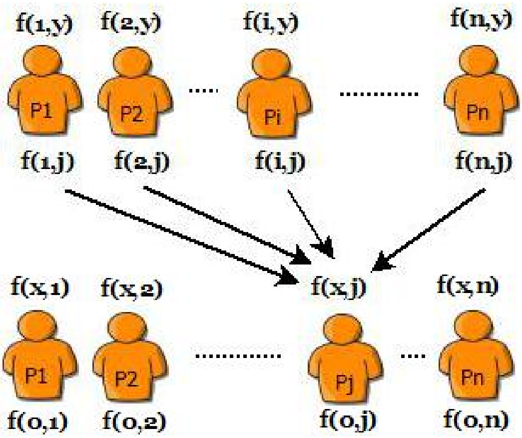
State-of-the-art Protocol for AVSS

Sharing Phase:



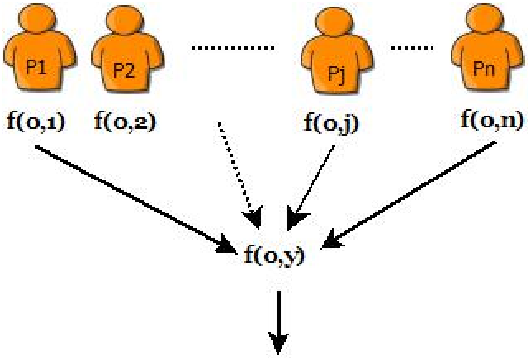
State-of-the-art Protocol for AVSS

Sharing Phase:



State-of-the-art Protocol for AVSS

Reconstruction Phase:



Compute the secret $s = f(0, 0)$

State-of-the-art Protocol for AVSS

For verification:

Commitment matrix $\mathbf{C} = \{C_{jl}\} = g^{f_{jl}}$

State-of-the-art Protocol for AVSS

For verification:

Commitment matrix $\mathbf{C} = \{C_{jl}\} = g^{f_{jl}}$

Reduced from $O(n^2)$ to $O(n)$ with hash functions.

State-of-the-art Protocol for AVSS

For verification:

Commitment matrix $\mathbf{C} = \{C_{jl}\} = g^{f_{jl}}$

Reduced from $O(n^2)$ to $O(n)$ with hash functions.

Message complexity: $O(n^2)$,

Communication complexity: $O(\kappa n^3)$,

where κ is the security parameter

Reference : C. Cachin, K. Kursawe, A.Lysyanskaya, and R. Strobl. Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems, ACM CCS'02.

What We Want to Achieve

Message complexity: $O(n^2)$,

Communication complexity: $O(\kappa n^2)$,

where κ is the security parameter

What We Want to Achieve

Message complexity: $O(n^2)$,

Communication complexity: $O(\kappa n^2)$,

where κ is the security parameter

Solution : Polynomial Commitments

Helps commit to a vector by publishing just one value

Reference : A.Kate, G. M. Zaverucha, and I. Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In Proceedings of ASIACRYPT'10.

A PolyCommit scheme

Setup$(1^\kappa, t)$ generates an appropriate algebraic structure $\mathcal{G} = \langle e, \mathbb{G}, \mathbb{G}_T \rangle$ and the system parameter PK

A PolyCommit scheme

Setup$(1^{\kappa}, t)$ generates an appropriate algebraic structure $\mathcal{G} = \langle e, \mathbb{G}, \mathbb{G}_T \rangle$ and the system parameter PK

Commit$(PK, \phi(x))$ outputs a commitment $\mathcal{C}$ to a polynomial $\phi(x)$

A PolyCommit scheme

Setup$(1^\kappa, t)$ generates an appropriate algebraic structure $\mathcal{G} = \langle e, \mathbb{G}, \mathbb{G}_T \rangle$ and the system parameter PK

Commit$(PK, \phi(x))$ outputs a commitment $\mathcal{C}$ to a polynomial $\phi(x)$

CreateWitness$(PK, \phi(x), i)$ outputs $\langle i, \phi(i), w_i \rangle$, where w_i is a witness for the evaluation $\phi(i)$ of $\phi(x)$

A PolyCommit scheme

Setup$(1^\kappa, t)$ generates an appropriate algebraic structure $\mathcal{G} = \langle e, \mathbb{G}, \mathbb{G}_T \rangle$ and the system parameter PK

Commit$(PK, \phi(x))$ outputs a commitment $\mathcal{C}$ to a polynomial $\phi(x)$

CreateWitness$(PK, \phi(x), i)$ outputs $\langle i, \phi(i), w_i \rangle$, where w_i is a witness for the evaluation $\phi(i)$ of $\phi(x)$

VerifyEval$(PK, \mathcal{C}, i, \phi(i), w_i)$ verifies that $\phi(i)$ is indeed the evaluation of the polynomial committed in $\mathcal{C}$

Our Protocol

Protocol for AVSS - 1

Dealer

- D selects a polynomial $\phi(x)$, such that $\phi(0) = s$.
- $\mathcal{C} = \text{Commit}(PK, \phi(x))$, $w_i = \text{CreateWitness}((PK, \phi(x), i))$
- D sends ($\mathcal{C}$, w_i, $\phi(i)$) to every party P_i.

Protocol for AVSS - 1

Dealer

- D selects a polynomial $\phi(x)$, such that $\phi(0) = s$.
- $\mathcal{C} =$ Commit$(PK, \phi(x))$, $w_i =$ CreateWitness$((PK, \phi(x), i))$
- D sends ($\mathcal{C}$, w_i, $\phi(i)$) to every party P_i.

Party P_i

- If VerifyEval(PK,$\mathcal{C}$, i, $\phi(i)$, w_i) succeeds, send (`echo`, $\mathcal{C}$)
- On receiving $(n - t)$ (`echo`, $\mathcal{C}$), send (`ready`, `holder`, $\mathcal{C}$)
- Otherwise:
  - (a) On receiving $(n - 2t)$ (`ready`,*, $\mathcal{C}$) signals, send (`ready`, `holder`, $\mathcal{C}$) to every party P_j.
  - (b) On receiving $(n - 2t)$ (`ready`,*, $\mathcal{C}'$) signals, send (`ready`, `non-holder`, $\mathcal{C}'$) to every party P_j.
- On receiving $(n - t)$ (`ready`, $\mathcal{C}$) signals, and at least $(n - 2t)$ contain `holder`, terminate.

Salient Points

- There are at least $n - 2t \geq 3t + 1 - 2t = t + 1$ honest parties with correct shares
- There are at most n `send`, n^2 `echo` and n^2 `ready` messages

Properties of AVSS

Liveness. If the dealer D is honest, then all honest parties complete sharing.

Secrecy. If D is honest, then the adversary has no information about s.

Agreement. If some honest party completes the sharing phase, then all honest parties will eventually complete the sharing phase.

Properties of AVSS

Liveness. If the dealer D is honest, then all honest parties complete sharing.

Secrecy. If D is honest, then the adversary has no information about s.

Agreement. If some honest party completes the sharing phase, then all honest parties will eventually complete the sharing phase.

Correctness. Once all honest parties complete sharing, there exists a fixed value $z \in \mathbb{Z}_p$, such that the following holds:

(a) If an honest dealer has shared the secret s, then $s = z$.

(b) If each of the honest servers P_i reconstructs some z_i, then $z_i = z$

Properties of AVSS

Liveness. If the dealer D is honest, then all honest parties complete sharing.

Secrecy. If D is honest, then the adversary has no information about s.

Agreement. If some honest party completes the sharing phase, then all honest parties will eventually complete the sharing phase.

Strong Correctness. Once $t+1$ honest parties complete sharing, there exists a fixed value $z \in \mathbb{Z}_p$, such that the following holds:

(a) If an honest dealer has shared the secret s, then $s = z$.

(b) If each of the honest servers P_i reconstructs some z_i, then $z_i = z$

Protocol for AVSS-SC

- Dealer `sends` polynomials $\phi^0(x), \phi^1(x), ...\phi^n(x)$, with $\phi^k(x) = F(x, k)$, $F(x, y)$ is of degree $\leq t$. Commitments: $\mathcal{C}^0, \mathcal{C}^1, ..., \mathcal{C}^n$.

Protocol for AVSS-SC

- Dealer `sends` polynomials $\phi^0(x), \phi^1(x), ...\phi^n(x)$, with $\phi^k(x) = F(x, k)$, $F(x, y)$ is of degree $\leq t$. Commitments: $\mathcal{C}^0, \mathcal{C}^1, ..., \mathcal{C}^n$.

- There will be at least $t + 1$ honest parties with correct polynomials $\phi^k(x) = F(x, k)$, and they compute their shares $s_k = \phi^k(0) = F(0, k) = F(k, 0) = \phi^0(k)$

Protocol for AVSS-SC

- Dealer `sends` polynomials $\phi^0(x), \phi^1(x), ...\phi^n(x)$, with $\phi^k(x) = F(x, k)$, $F(x, y)$ is of degree $\leq t$. Commitments: $\mathcal{C}^0, \mathcal{C}^1, ..., \mathcal{C}^n$.

- There will be at least $t + 1$ honest parties with correct polynomials $\phi^k(x) = F(x, k)$, and they compute their shares $s_k = \phi^k(0) = F(0, k) = F(k, 0) = \phi^0(k)$

- These $t + 1$ parties can enable any P_i to reconstruct its polynomial $\phi^i(x)$ by sending $\phi^k(i) = \phi^i(k)$

Protocol for AVSS-SC

- Dealer `sends` polynomials $\phi^0(x), \phi^1(x), ...\phi^n(x)$, with $\phi^k(x) = F(x,k)$, $F(x,y)$ is of degree $\leq t$. Commitments: $\mathcal{C}^0, \mathcal{C}^1, ..., \mathcal{C}^n$.

- There will be at least $t+1$ honest parties with correct polynomials $\phi^k(x) = F(x,k)$, and they compute their shares $s_k = \phi^k(0) = F(0,k) = F(k,0) = \phi^0(k)$

- These $t+1$ parties can enable any P_i to reconstruct its polynomial $\phi^i(x)$ by sending $\phi^k(i) = \phi^i(k)$

Problem: Have to send a vector of commitments in the `echo` and `ready` messages.

Protocol for AVSS-SC

- Dealer `sends` polynomials $\phi^0(x), \phi^1(x), ...\phi^n(x)$, with $\phi^k(x) = F(x, k)$, $F(x, y)$ is of degree $\leq t$. Commitments: $\mathcal{C}^0, \mathcal{C}^1, ..., \mathcal{C}^n$.

- There will be at least $t + 1$ honest parties with correct polynomials $\phi^k(x) = F(x, k)$, and they compute their shares $s_k = \phi^k(0) = F(0, k) = F(k, 0) = \phi^0(k)$

- These $t + 1$ parties can enable any P_i to reconstruct its polynomial $\phi^i(x)$ by sending $\phi^k(i) = \phi^i(k)$

Problem: Have to send a vector of commitments in the `echo` and `ready` messages.

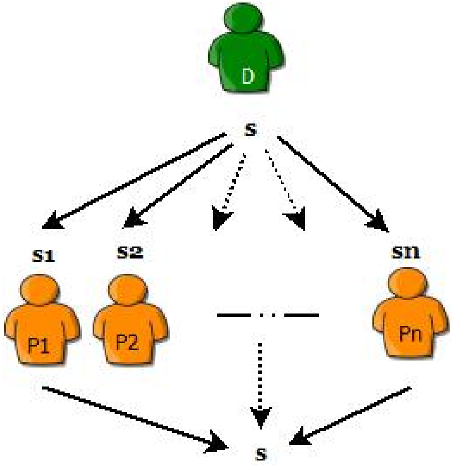
Solution: Perform another round of PolyCommit on hash values of the commitments.

Complexity Analysis

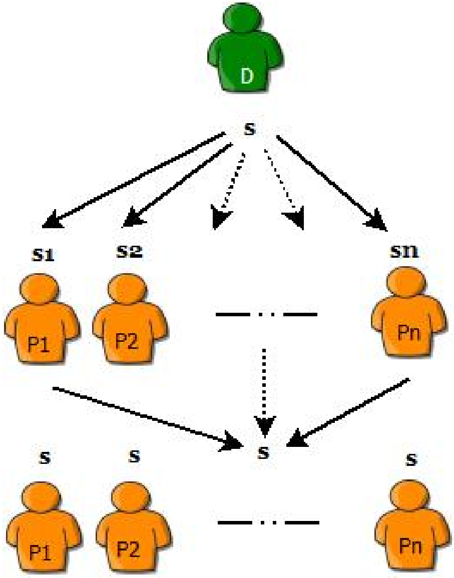
Theorem

A protocol for AVSS is sufficient to generate a protocol for a reliable broadcast.

Complexity Analysis



Complexity Analysis



Complexity Analysis

Theorem

A protocol for AVSS is sufficient to generate a protocol for a reliable broadcast.

Complexity Analysis

Theorem

A protocol for AVSS is sufficient to generate a protocol for a reliable broadcast.

Theorem

If a reliable broadcast protocol terminates, the number of messages exchanged is lower bounded by $max\{(n-t),(1+t/2)^2\}$.

Reference : D. Dolev and R. Reischuk. Bounds on information exchange for byzantine agreement. J. ACM, 1985

Contributions

- Incorporation of Polynomial Commitments to solve AVSS with improved complexity
- This protocol for AVSS achieves optimal complexity

Contributions

- Incorporation of Polynomial Commitments to solve AVSS with improved complexity
- This protocol for AVSS achieves optimal complexity

Thank You