CS 594 – Advanced Crypto (Spring 2026)

Alex Block

Lecture 9

February 16, 2026

Verifiable Secret Sharing

The Story So Far

The Story So Far

Dealer D

The Story So Far

Dealer D

P_5

P_1

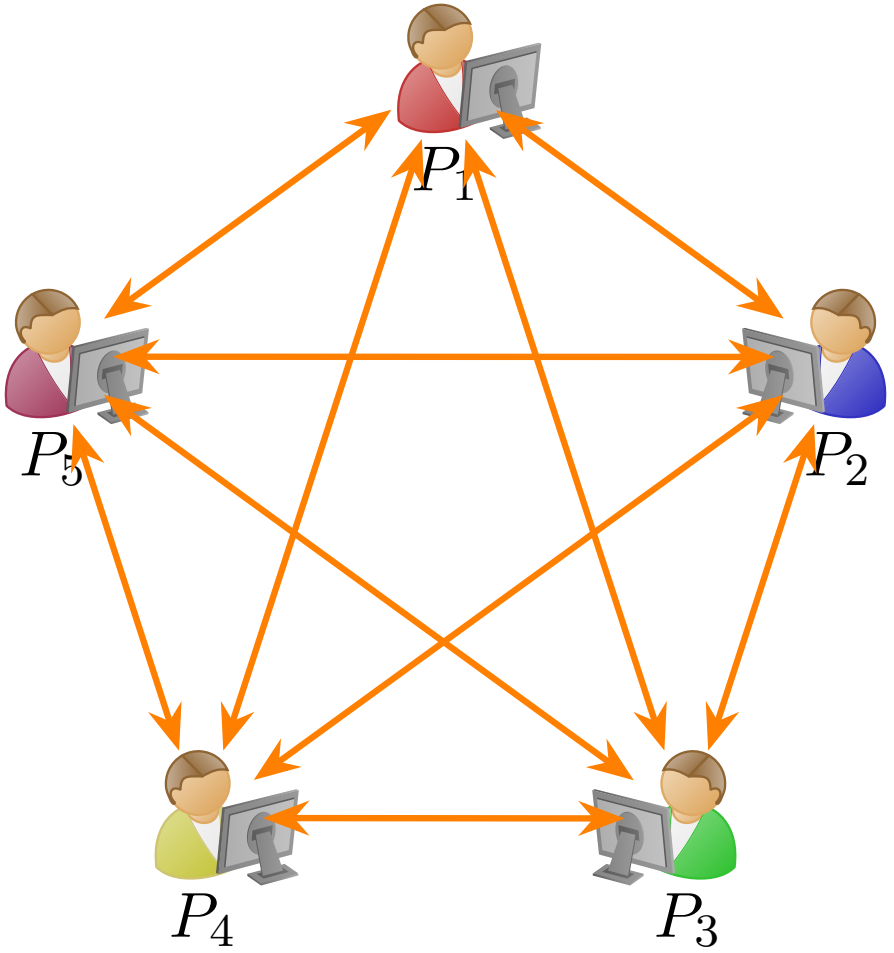
P_2

P_4

P_3

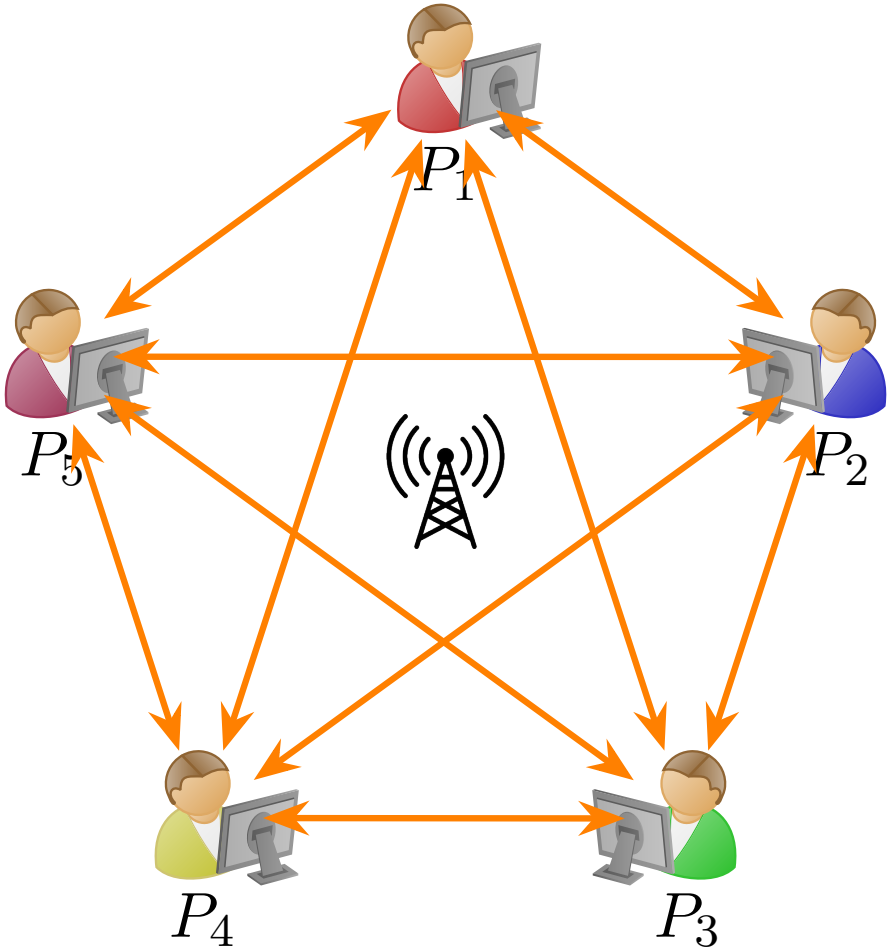
The Story So Far

Dealer D

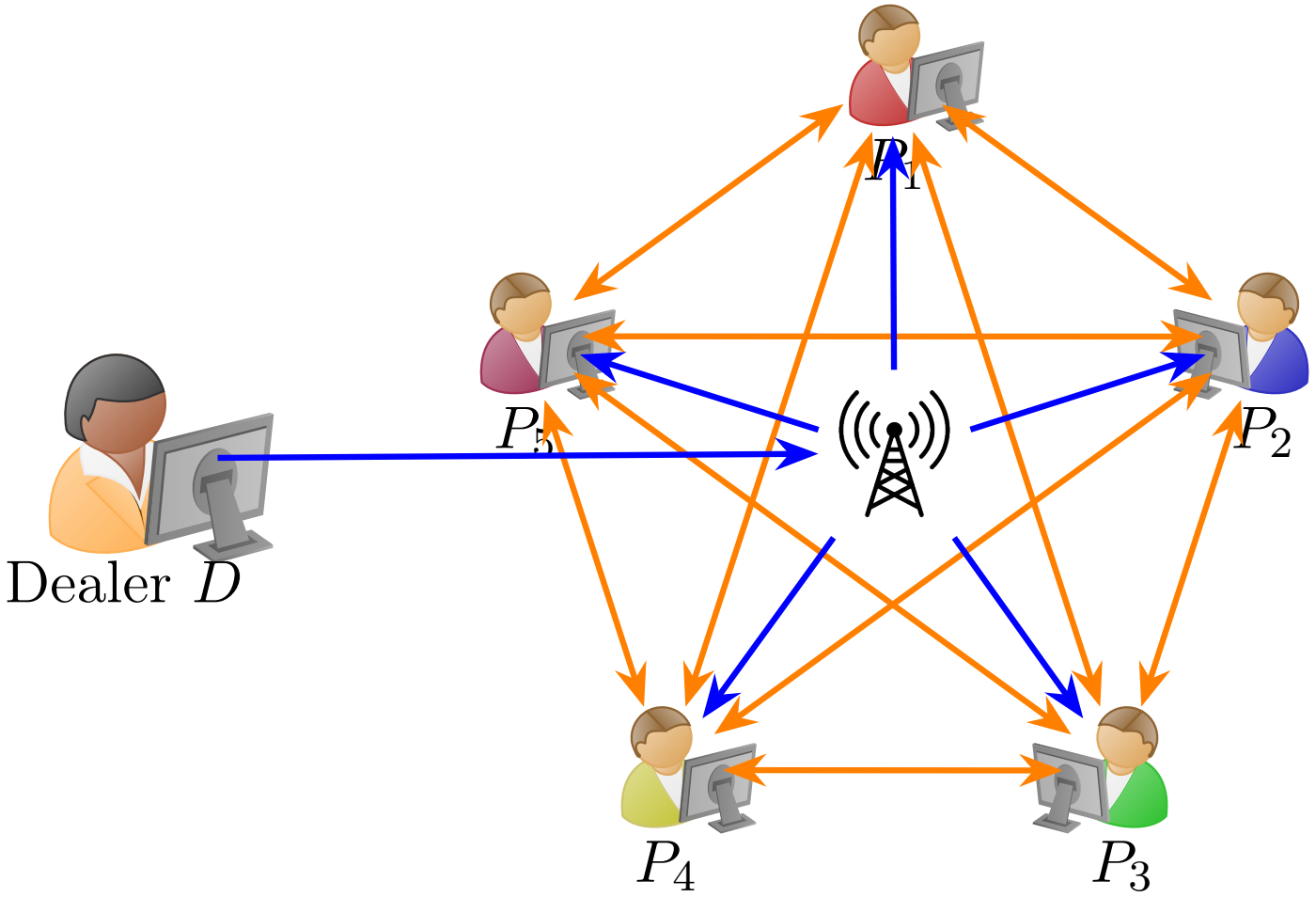


The Story So Far

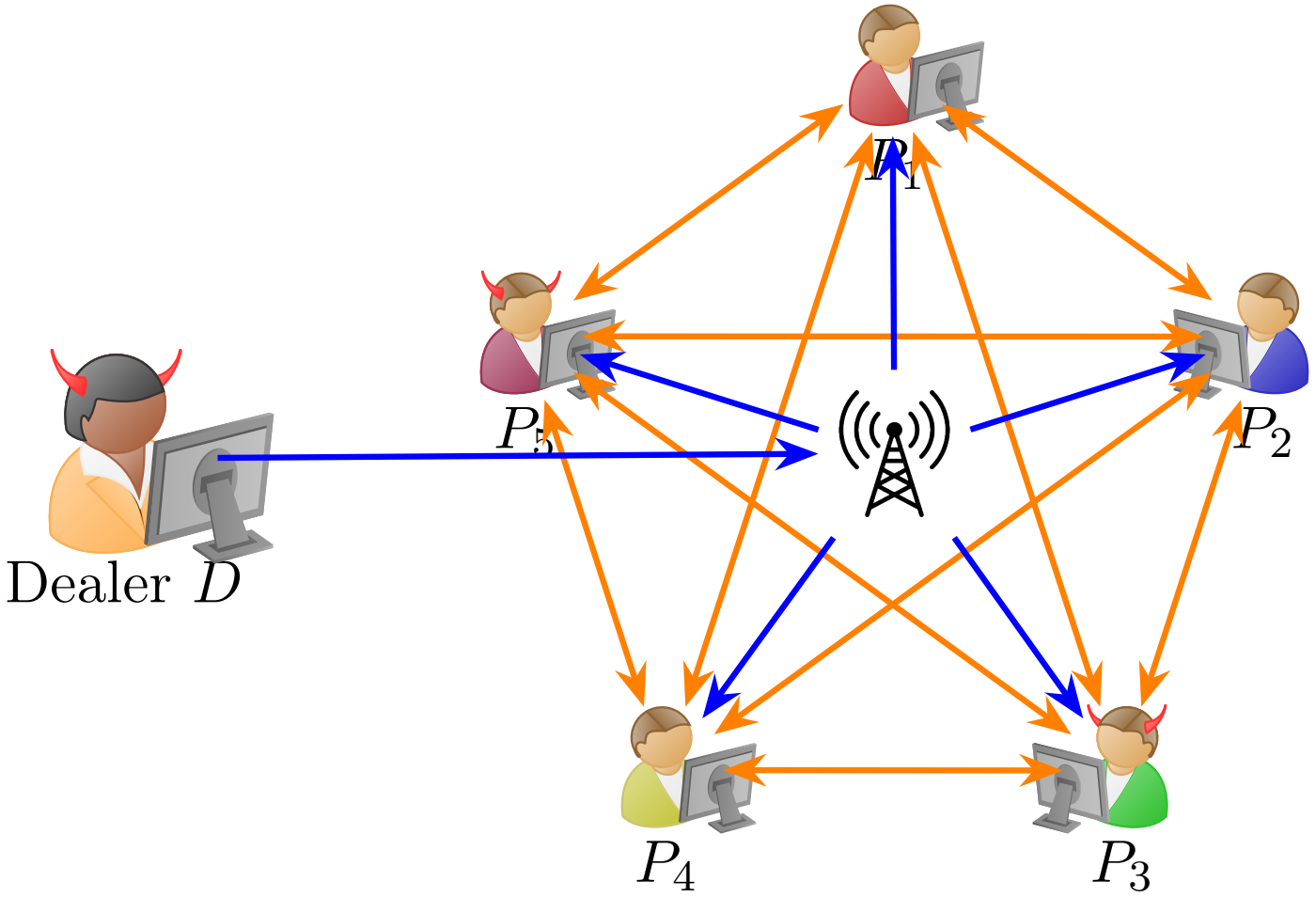
Dealer D



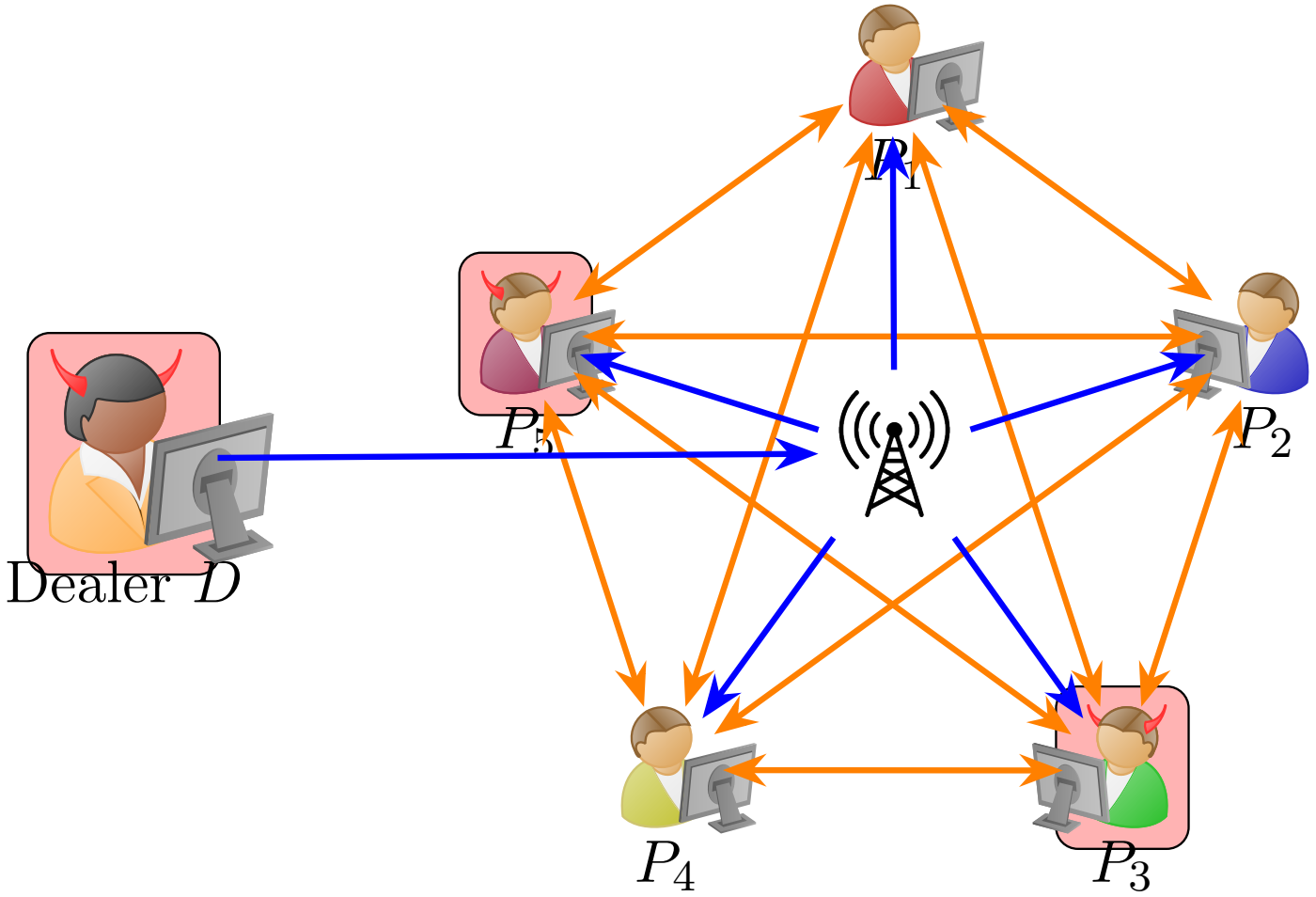
The Story So Far



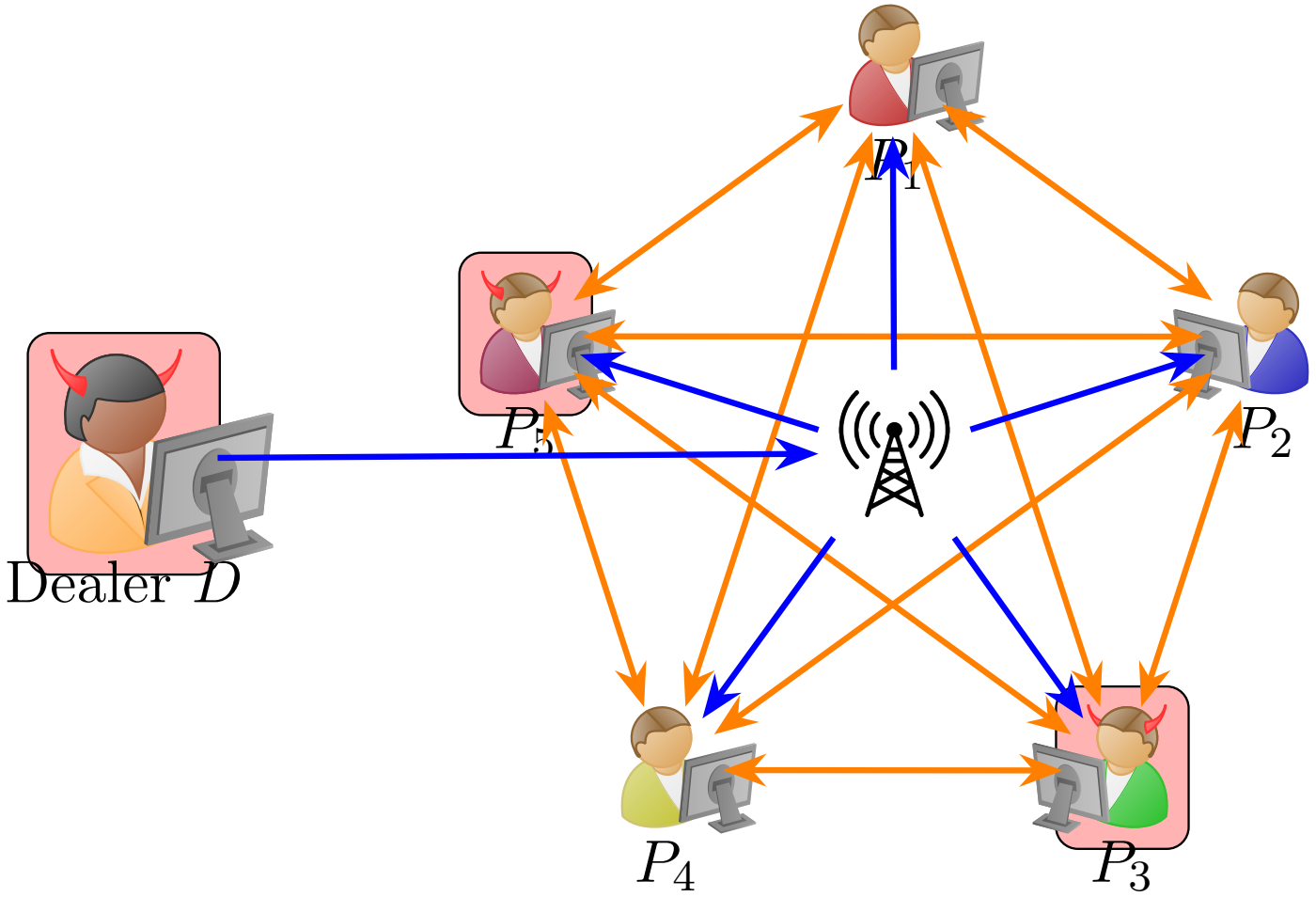
The Story So Far



The Story So Far



The Story So Far



How do we know the dealer has given honest parties *valid* shares?

Verifiable Secret Sharing

- *Verifiable Secret Sharing* (VSS) is an attempt to solve this problem.

Verifiable Secret Sharing

- *Verifiable Secret Sharing* (VSS) is an attempt to solve this problem.
- VSS extend (t, n) secret sharing to protect against:

Verifiable Secret Sharing

- *Verifiable Secret Sharing* (VSS) is an attempt to solve this problem.

- VSS extend (t, n) secret sharing to protect against:
  - Corrupt dealers colluding with other parties; and

Verifiable Secret Sharing

- *Verifiable Secret Sharing* (VSS) is an attempt to solve this problem.
- VSS extend (t, n) secret sharing to protect against:
  - Corrupt dealers colluding with other parties; and
  - Malicious and active adversaries.
- VSS guarantees that:

Verifiable Secret Sharing

- *Verifiable Secret Sharing* (VSS) is an attempt to solve this problem.
- VSS extend (t, n) secret sharing to protect against:
  - Corrupt dealers colluding with other parties; and
  - Malicious and active adversaries.
- VSS guarantees that:
  - If the dealer is honest, then honest parties reconstruct as in normal secret sharing; and

Verifiable Secret Sharing

- *Verifiable Secret Sharing* (VSS) is an attempt to solve this problem.
- VSS extend (t, n) secret sharing to protect against:
  - Corrupt dealers colluding with other parties; and
  - Malicious and active adversaries.
- VSS guarantees that:
  - If the dealer is honest, then honest parties reconstruct as in normal secret sharing; and
  - If the dealer is *corrupt*, then honest parties can still reconstruct *some secret*.

Verifiable Secret Sharing

- *Verifiable Secret Sharing* (VSS) is an attempt to solve this problem.
- VSS extend (t, n) secret sharing to protect against:
  - Corrupt dealers colluding with other parties; and
  - Malicious and active adversaries.
- VSS guarantees that:
  - If the dealer is honest, then honest parties reconstruct as in normal secret sharing; and
  - If the dealer is *corrupt*, then honest parties can still reconstruct *some secret*.
- VSS also extends Sharing and Reconstruction to *protocols*.

VSS: Definition

VSS: Definition

Definition 1 ((Type-II) Verifiable Secret Sharing)

VSS: Definition

Definition 1 ((Type-II) Verifiable Secret Sharing)

Let $\mathsf{SS} = (\mathsf{Share}, \mathsf{Recon})$ be a (t, n) secret sharing scheme and let $\mathcal{P}$ be the set of parties.

VSS: Definition

Definition 1 ((Type-II) Verifiable Secret Sharing)

Let $\mathsf{SS} = (\mathsf{Share}, \mathsf{Recon})$ be a (t, n) secret sharing scheme and let $\mathcal{P}$ be the set of parties. Then, a pair of protocols $\Pi = (\Pi_{\mathsf{S}}, \Pi_{\mathsf{R}})$ for parties $\mathcal{P}$, with a designated dealer $\mathsf{D} \in \mathcal{P}$ with some private input s, is a *(Type-II) perfectly secure VSS scheme* if the following hold.

VSS: Definition

Definition 1 ((Type-II) Verifiable Secret Sharing)

Let $\mathsf{SS} = (\mathsf{Share}, \mathsf{Recon})$ be a (t, n) secret sharing scheme and let $\mathcal{P}$ be the set of parties. Then, a pair of protocols $\Pi = (\Pi_{\mathsf{S}}, \Pi_{\mathsf{R}})$ for parties $\mathcal{P}$, with a designated dealer $\mathsf{D} \in \mathcal{P}$ with some private input s, is a *(Type-II) perfectly secure VSS scheme* if the following hold.

- **Correctness**: if D is honest, then at the end of Π_{S}, s is shared with $\mathcal{P}$ according to Share; moreover, all honest parties output s at the end of Π_{R}.

VSS: Definition

Definition 1 ((Type-II) Verifiable Secret Sharing)

Let $\mathsf{SS} = (\mathsf{Share}, \mathsf{Recon})$ be a (t, n) secret sharing scheme and let $\mathcal{P}$ be the set of parties. Then, a pair of protocols $\Pi = (\Pi_{\mathsf{S}}, \Pi_{\mathsf{R}})$ for parties $\mathcal{P}$, with a designated dealer $\mathsf{D} \in \mathcal{P}$ with some private input s, is a *(Type-II) perfectly secure VSS scheme* if the following hold.

- **Correctness**: if D is honest, then at the end of Π_{S}, s is shared with $\mathcal{P}$ according to Share; moreover, all honest parties output s at the end of Π_{R}.
- **Privacy**: if D is honest, corrupt parties learn *nothing* about s during Π_{S}.

VSS: Definition

Type-I

Definition 1 ((Type-II) Verifiable Secret Sharing)

Let $\mathsf{SS} = (\mathsf{Share}, \mathsf{Recon})$ be a (t, n) secret sharing scheme and let $\mathcal{P}$ be the set of parties. Then, a pair of protocols $\Pi = (\Pi_{\mathsf{S}}, \Pi_{\mathsf{R}})$ for parties $\mathcal{P}$, with a designated dealer $\mathsf{D} \in \mathcal{P}$ with some private input s, is a *(Type-II) perfectly secure VSS scheme* if the following hold.

- **Correctness**: if D is honest, then at the end of Π_{S}, s is shared with $\mathcal{P}$ according to Share; moreover, all honest parties output s at the end of Π_{R}.
- **Privacy**: if D is honest, corrupt parties learn *nothing* about s during Π_{S}.
- **Strong Commitment**: if D is *corrupt*, the execution of Π_{S} with honest parties defines some secret value $\widetilde{s}$ such that (1) $\widetilde{s}$ is shared among $\mathcal{P}$ according to Share, and (2) all honest parties output $\widetilde{s}$ at the end of Π_{R}.

(2) ✗

Weak (Type-I) : doesn't hold w.r.t SS

VSS: Definition Flexibility

- VSS has many other considerations in the definition (versus standard SS):

VSS: Definition Flexibility

- VSS has many other considerations in the definition (versus standard SS):
  - Privacy: perfect, statistical, or computational.

VSS: Definition Flexibility

- VSS has many other considerations in the definition (versus standard SS):
  - Privacy: perfect, statistical, or computational.
  - Adversary: adaptive or static, computationally bounded or unbounded.

VSS: Definition Flexibility

- VSS has many other considerations in the definition (versus standard SS):
  - Privacy: perfect, statistical, or computational.
  - Adversary: adaptive or static, computationally bounded or unbounded.
  - Commitment: perfect or statistical.

VSS: Definition Flexibility

- VSS has many other considerations in the definition (versus standard SS):
  - Privacy: perfect, statistical, or computational.
  - Adversary: adaptive or static, computationally bounded or unbounded.
  - Commitment: perfect or statistical.
  - Communication: point-to-point or broadcast; synchronous or asynchronous.

VSS: Definition Flexibility

- VSS has many other considerations in the definition (versus standard SS):
  - Privacy: perfect, statistical, or computational.
  - Adversary: adaptive or static, computationally bounded or unbounded.
  - Commitment: perfect or statistical.
  - Communication: point-to-point or broadcast; synchronous or asynchronous.
  - Corruption threshold t.

VSS: Definition Flexibility

- VSS has many other considerations in the definition (versus standard SS):
  - Privacy: perfect, statistical, or computational.
  - Adversary: adaptive or static, computationally bounded or unbounded.
  - Commitment: perfect or statistical.
  - Communication: point-to-point or broadcast; synchronous or asynchronous.
  - Corruption threshold t.
  - Security with abort or guaranteed output delivery.

VSS Setup for this Lecture

VSS Setup for this Lecture

- *Perfect* VSS: honest parties *always* reconstruct some (consistent) value.

VSS Setup for this Lecture

- *Perfect* VSS: honest parties *always* reconstruct some (consistent) value.

- Corruption Threshold: $3t < n$. $t < n/3$

VSS Setup for this Lecture

- *Perfect* VSS: honest parties *always* reconstruct some (consistent) value.
- Corruption Threshold: $3t < n$.
- Synchronous Communication.

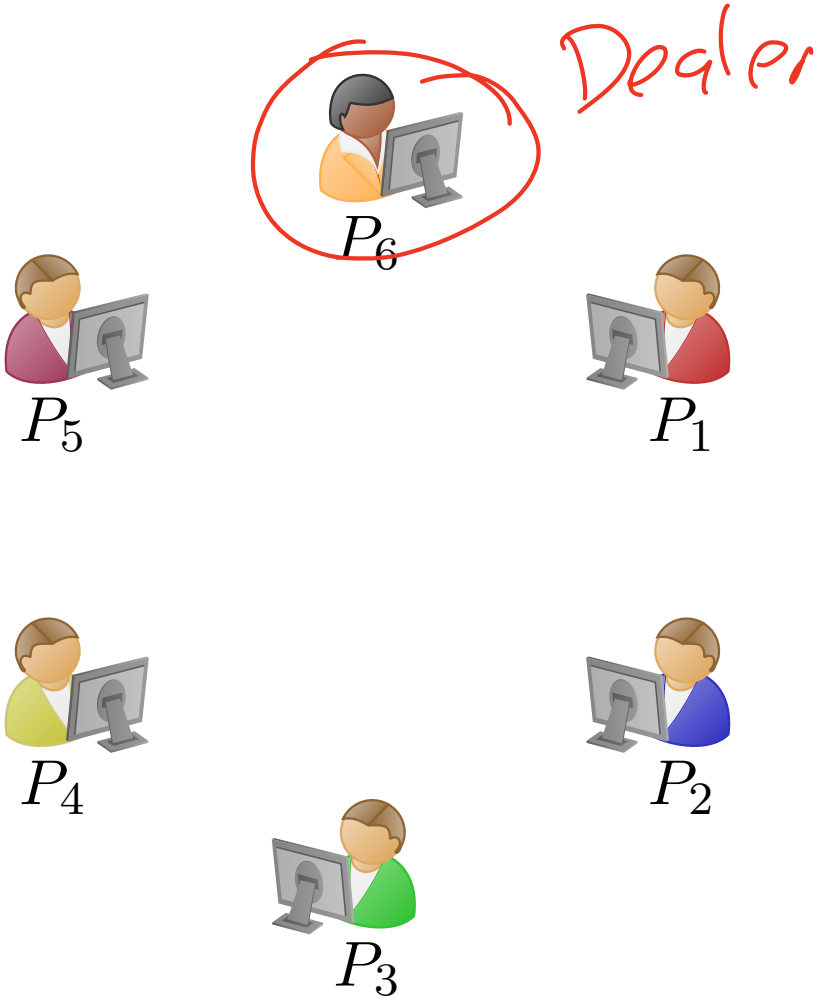
VSS Setup for this Lecture

- *Perfect* VSS: honest parties *always* reconstruct some (consistent) value.
- Corruption Threshold: $3t < n$.
- Synchronous Communication.
- Secure Broadcast and Secure Point-to-point Channels.

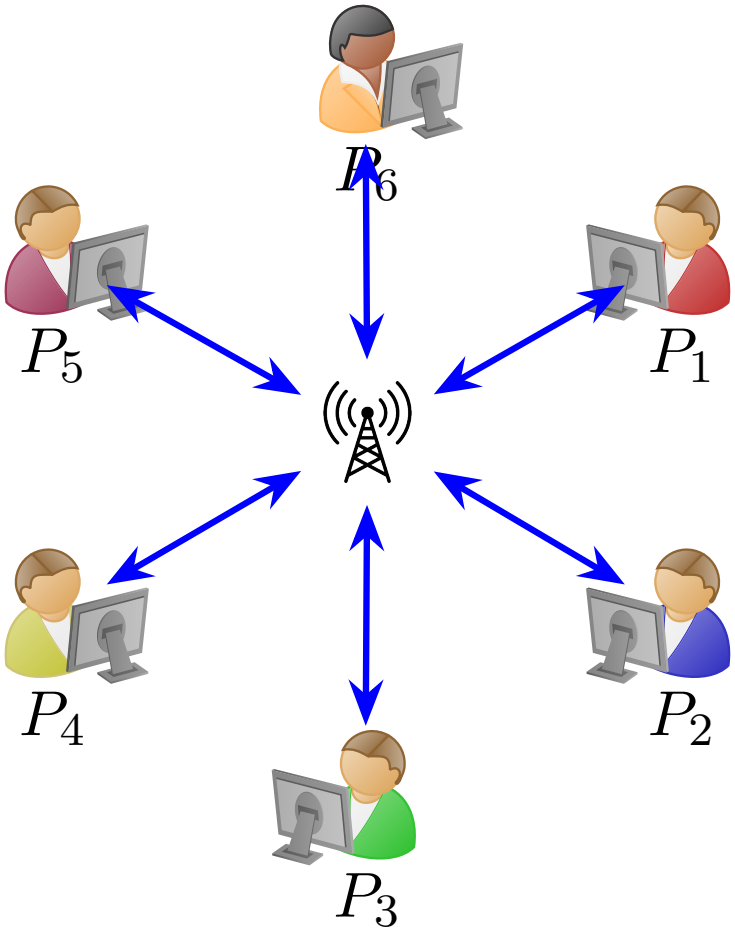
VSS Setup for this Lecture

- *Perfect* VSS: honest parties *always* reconstruct some (consistent) value.
- Corruption Threshold: $3t < n$.
- Synchronous Communication.
- Secure Broadcast and Secure Point-to-point Channels.
- Adversary: static, malicious, computationally (un)bounded.

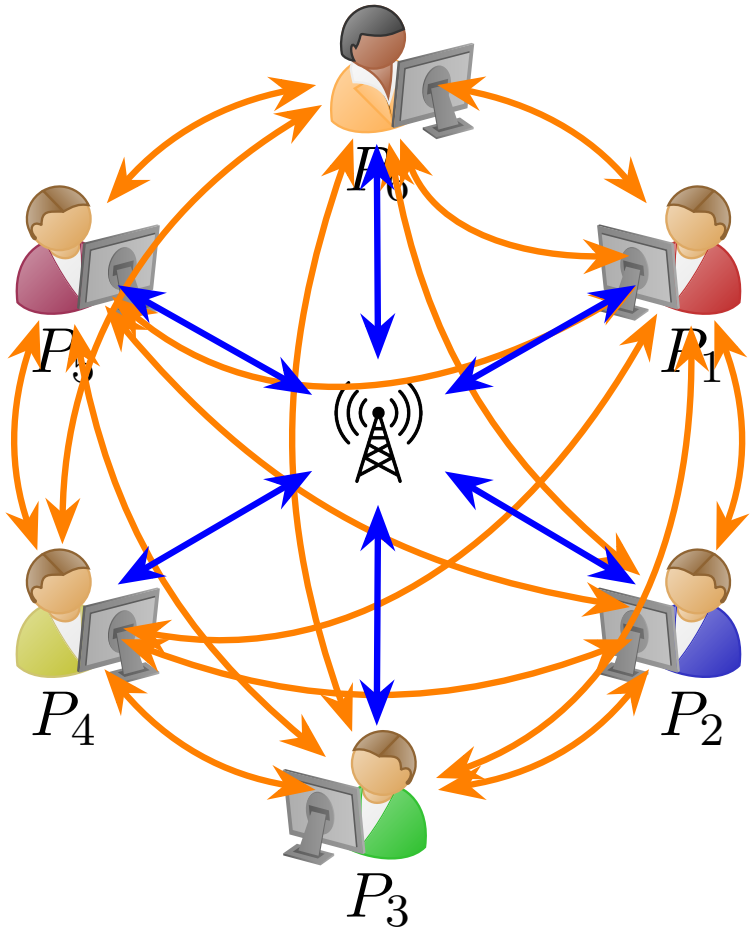
VSS Adversary and Communication Model



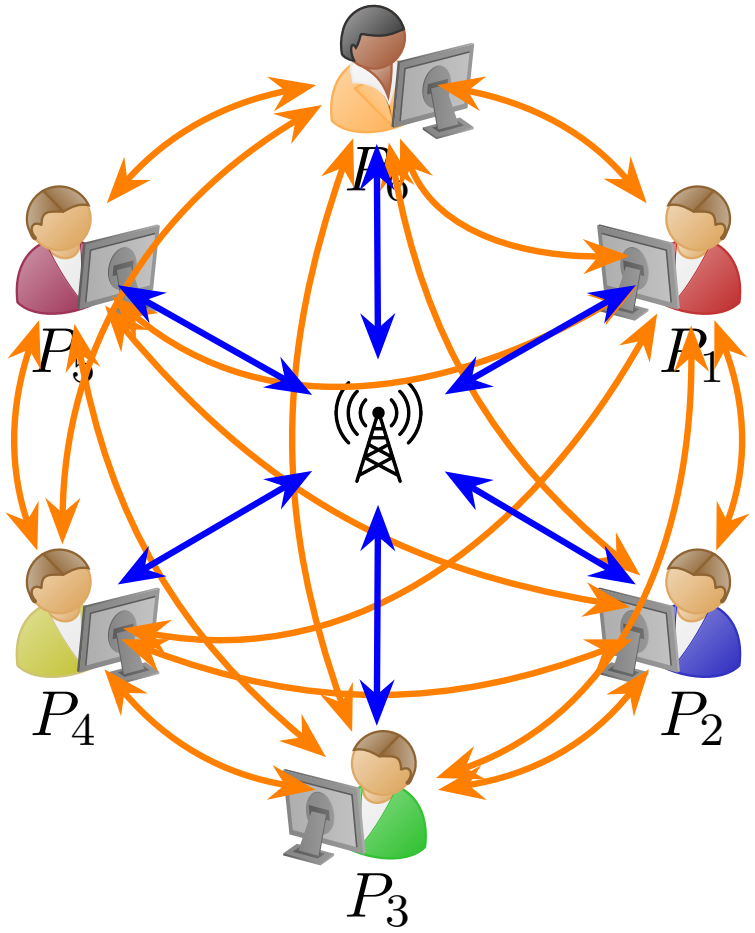
VSS Adversary and Communication Model



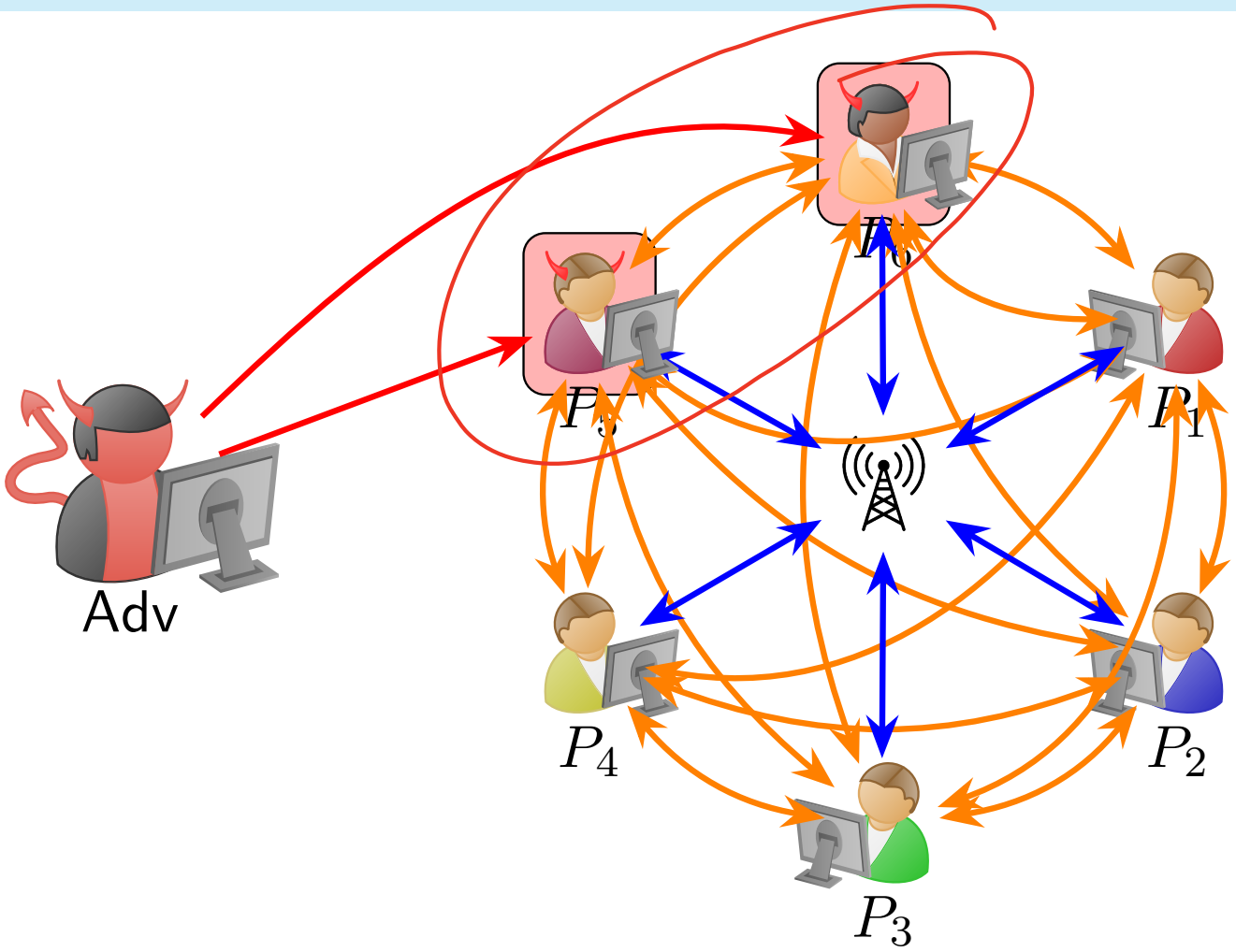
VSS Adversary and Communication Model



VSS Adversary and Communication Model

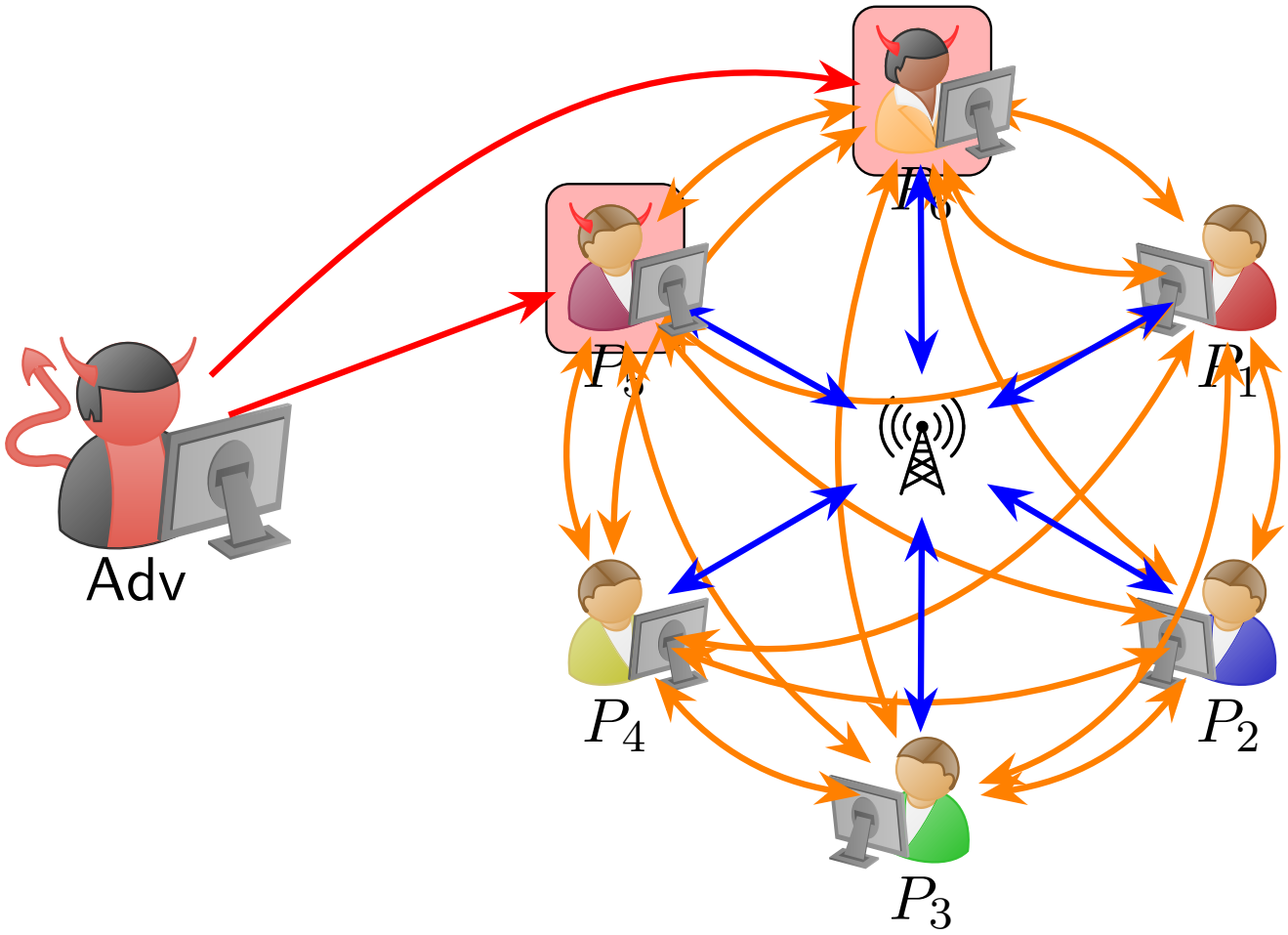


VSS Adversary and Communication Model



$3t < n$

VSS Adversary and Communication Model



(R, R') VSS

Both Π_{S} and Π_{R} proceed for at most R rounds, where the broadcast channel is used for at most R' rounds.

Feldman's Non-interactive VSS

Feldman's Non-interactive VSS

- Feldman gave a simple *non-interactive* VSS scheme.

Feldman's Non-interactive VSS

- Feldman gave a simple *non-interactive* VSS scheme.
- Builds on Shamir, uses discrete-log based commitments.

Feldman's Non-interactive VSS

- Feldman gave a simple *non-interactive* VSS scheme.
- Builds on Shamir, uses discrete-log based commitments.
- Let p, q be large enough primes such that $p|(q-1)$.

Feldman's Non-interactive VSS

- Feldman gave a simple *non-interactive* VSS scheme.
- Builds on Shamir, uses discrete-log based commitments.
- Let p, q be primes such that $p|(q-1)$.
- Let $\mathbb{G}$ be a cyclic group of order q where discrete-log is hard.

g^a

$(g, g \cdot a)$ for rand. a, then $\approx_c (g, h)$ $h \xleftarrow{\$} \mathbb{G}$

Feldman's Non-interactive VSS

- Feldman gave a simple *non-interactive* VSS scheme.
- Builds on Shamir, uses discrete-log based commitments.
- Let p, q be primes such that $p|(q-1)$.
- Let $\mathbb{G}$ be a cyclic group of order q where discrete-log is hard.
- Let $g \in \mathbb{G}$ be an element of order p: $g^p = 1$ (in $\mathbb{G}$).

g·p in additive group

Feldman's Non-interactive VSS

- Feldman gave a simple *non-interactive* VSS scheme.
- Builds on Shamir, uses discrete-log based commitments.
- Let p, q be primes such that $p|(q-1)$.
- Let $\mathbb{G}$ be a cyclic group of order q where discrete-log is hard.
- Let $g \in \mathbb{G}$ be an element of order p: $g^p = 1$ (in $\mathbb{G}$).
- Idea: Perform Shamir over $\mathbb{Z}_p$, commit to the coefficients using g.

Feldman's Non-interactive VSS

$p > n$

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g)$:

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g)$:

- The dealer does the following:

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i$.

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i$.
  - Broadcast $(C_0, \ldots, C_{t-1})$ to all parties.

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_S(s, g)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i$.
  - Broadcast $(C_0, \ldots, C_{t-1})$ to all parties.
  - Compute $s_i = f(i)$ for $i \in [n]$.

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_S(s, g)$:

(1,1) VSS

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i$.
  - Broadcast $(C_0, \ldots, C_{t-1})$ to all parties.
  - Compute $s_i = f(i)$ for $i \in [n]$.
  - Output secret shares $(s_1, \ldots, s_n)$.

<- P2P channel

$(s_i, (C_0, \ldots, C_{t-1})) \longrightarrow P_i$

each party i checks $g^{s_i} = \prod_{k=0}^{t-1} C_k^{i^k}$

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(s_1, \ldots, s_n)$:

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(s_1, \ldots, s_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(s_1, \ldots, s_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$
  - Compute $\widetilde{C}_i = g \cdot s_i$.

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(s_1, \ldots, s_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$
  - Compute $\widetilde{C}_i = g \cdot s_i$.
  - Check $\widetilde{C}_i \stackrel{?}{=} \sum_{k=0}^{t-1} (C_k) \cdot i^k$, kicking i out of the protocol if it fails.

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(s_1, \ldots, s_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$
  - Compute $\widetilde{C}_i = g \cdot s_i$.
  - Check $\widetilde{C}_i \stackrel{?}{=} \sum_{k=0}^{t-1} (C_k) \cdot i^k$, kicking i out of the protocol if it fails.
- Let $s_{i_1}, \ldots, s_{i_t}$ be the (first) t shares that correctly authenticated.

Feldman's Non-interactive VSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(s_1, \ldots, s_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$
  - Compute $\widetilde{C}_i = g \cdot s_i$.
  - Check $\widetilde{C}_i \stackrel{?}{=} \sum_{k=0}^{t-1} (C_k) \cdot i^k$, kicking i out of the protocol if it fails.
- Let $s_{i_1}, \ldots, s_{i_t}$ be the (first) t shares that correctly authenticated.
- Return $\mathsf{Shamir.Reconstruct}(s_{i_1}, \ldots, s_{i_t})$.

$s_i = f(i) = \sum_{k=0}^{t-1} f_k \cdot i^k$

$g \cdot s_i = \sum_{k=0}^{t-1} g \cdot f_k \cdot i^k$ → C_k if D_i is honest

Feldman's NIVSS: Correctness

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

Feldman's NIVSS: Correctness

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

Proof.

Feldman's NIVSS: Correctness

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

Proof.

- For (1), if D is honest, it honestly shares s as $f(i)$ for $i \in [n]$, where $f(X) = \sum_{j=0}^{t-1} f_j X^j$, $f_0 = s$, and $f_j \xleftarrow{\$} \mathbb{Z}_p$ for $j \in [t-1]$.

(Shamir, done!

Feldman's NIVSS: Correctness

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

Proof.

- For (1), if D is honest, it honestly shares s as $f(i)$ for $i \in [n]$, where $f(X) = \sum_{j=0}^{t-1} f_j X^j$, $f_0 = s$, and $f_j \xleftarrow{\$} \mathbb{Z}_p$ for $j \in [t-1]$.
- For (2), if one party is dishonest, then checking consistency with $C_0, \ldots, C_{t-1}$ will fail (unless Adv can break the discrete-log assumption in $\mathbb{G}$). $\tilde{s}_i$ s.t. $g \cdot \tilde{s}_i = \sum_{k=0}^{t-1} C_k \cdot i^k$ but $\tilde{s}_i \neq f(i)$

Feldman's NIVSS: Correctness

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

Proof.

- For (1), if D is honest, it honestly shares s as $f(i)$ for $i \in [n]$, where $f(X) = \sum_{j=0}^{t-1} f_j X^j$, $f_0 = s$, and $f_j \xleftarrow{\$} \mathbb{Z}_p$ for $j \in [t-1]$.
- For (2), if one party is dishonest, then checking consistency with $C_0, \ldots, C_{t-1}$ will fail (unless Adv can break the discrete-log assumption in $\mathbb{G}$).
  - If a share is not authenticated, kick that party out of the protocol.

Feldman's NIVSS: Correctness

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

Proof.

- For (1), if D is honest, it honestly shares s as $f(i)$ for $i \in [n]$, where $f(X) = \sum_{j=0}^{t-1} f_j X^j$, $f_0 = s$, and $f_j \xleftarrow{\$} \mathbb{Z}_p$ for $j \in [t-1]$.
- For (2), if one party is dishonest, then checking consistency with $C_0, \ldots, C_{t-1}$ will fail (unless Adv can break the discrete-log assumption in $\mathbb{G}$).
  - If a share is not authenticated, kick that party out of the protocol.
  - Collect the first t shares which authenticate, output $\mathsf{Shamir.Reconstruct}$ of those values.

Feldman's NIVSS: Correctness

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

Proof.

- For (1), if D is honest, it honestly shares s as $f(i)$ for $i \in [n]$, where $f(X) = \sum_{j=0}^{t-1} f_j X^j$, $f_0 = s$, and $f_j \xleftarrow{\$} \mathbb{Z}_p$ for $j \in [t-1]$.
- For (2), if one party is dishonest, then checking consistency with $C_0, \ldots, C_{t-1}$ will fail (unless Adv can break the discrete-log assumption in $\mathbb{G}$).
  - If a share is not authenticated, kick that party out of the protocol.
  - Collect the first t shares which authenticate, output $\mathsf{Shamir.Reconstruct}$ of those values.
- Honesty of D implies the honest parties output s. □

Feldman's VSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Feldman's VSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof*.

Feldman's VSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof*.

- Feldman's doesn't satisfy a strong notion of privacy.

Feldman's VSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof*.

- Feldman's doesn't satisfy a strong notion of privacy.
- If D is honest and $s \xleftarrow{\$} \mathbb{Z}_p$, then $g \cdot s$ is indistinguishable from a random group element by hardness of discrete-log.

Feldman's VSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof*.

- Feldman's doesn't satisfy a strong notion of privacy.
- If D is honest and $s \xleftarrow{\$} \mathbb{Z}_p$, then $g \cdot s$ is indistinguishable from a random group element by hardness of discrete-log.
- If s is *not random*, then Adv learns some information about s.

$C_0 = g \cdot s$

Feldman's VSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof*.

- Feldman's doesn't satisfy a strong notion of privacy.
- If D is honest and $s \xleftarrow{\$} \mathbb{Z}_p$, then $g \cdot s$ is indistinguishable from a random group element by hardness of discrete-log.
- If s is *not random*, then Adv learns some information about s.
  - E.g., if $s \equiv 0 \bmod p$, then $g \cdot s = 1$.

Feldman's VSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof*.

- Feldman's doesn't satisfy a strong notion of privacy.
- If D is honest and $s \xleftarrow{\$} \mathbb{Z}_p$, then $g \cdot s$ is indistinguishable from a random group element by hardness of discrete-log.
- If s is *not random*, then Adv learns some information about s.
  - E.g., if $s \equiv 0 \bmod p$, then $g \cdot s = 1$.
- In the context of many uses of VSS (and Feldman's), this is just fine, so we ignore it for now. □

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to Shamir.Share and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to Shamir.Share and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to $\mathsf{Shamir.Share}$ and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

- Without loss of generality, we can examine Π_{R} from the view of all honest parties P_i for $i \in H := [n] \setminus T$, where $T \subseteq [n]$ of size $|T| \leqslant t-1$ is the set of corrupt parties.

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to $\mathsf{Shamir.Share}$ and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

- Without loss of generality, we can examine Π_{R} from the view of all honest parties P_i for $i \in H := [n] \setminus T$, where $T \subseteq [n]$ of size $|T| \leqslant t-1$ is the set of corrupt parties.
- During Π_{R}, each P_i for $i \in H$ authenticates the shares s_i.

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to $\mathsf{Shamir.Share}$ and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

- Without loss of generality, we can examine Π_{R} from the view of all honest parties P_i for $i \in H := [n] \setminus T$, where $T \subseteq [n]$ of size $|T| \leqslant t-1$ is the set of corrupt parties.
- During Π_{R}, each P_i for $i \in H$ authenticates the shares s_i.
  - By construction of C_k, authentication will only pass if Adv:

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to Shamir.Share and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

- Without loss of generality, we can examine Π_{R} from the view of all honest parties P_i for $i \in H := [n] \setminus T$, where $T \subseteq [n]$ of size $|T| \leqslant t-1$ is the set of corrupt parties.
- During Π_{R}, each P_i for $i \in H$ authenticates the shares s_i.
  - By construction of C_k, authentication will only pass if Adv:
    1. shares some secret $\widetilde{s}$ according to Shamir.Share and correctly constructs C_k for $k \in \{0, \ldots, t-1\}$; or

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to Shamir.Share and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

- Without loss of generality, we can examine Π_{R} from the view of all honest parties P_i for $i \in H := [n] \setminus T$, where $T \subseteq [n]$ of size $|T| \leqslant t-1$ is the set of corrupt parties.
- During Π_{R}, each P_i for $i \in H$ authenticates the shares s_i.
  - By construction of C_k, authentication will only pass if Adv:
    1. shares some secret $\widetilde{s}$ according to Shamir.Share and correctly constructs C_k for $k \in \{0, \ldots, t-1\}$; or
    2. can break the discrete-log assumption in $\mathbb{G}$.

Feldman's VSS: Strong Commitment

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to Shamir.Share and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

- Without loss of generality, we can examine Π_{R} from the view of all honest parties P_i for $i \in H := [n] \setminus T$, where $T \subseteq [n]$ of size $|T| \leqslant t-1$ is the set of corrupt parties.
- During Π_{R}, each P_i for $i \in H$ authenticates the shares s_i.
  - By construction of C_k, authentication will only pass if Adv:
    1. shares some secret $\widetilde{s}$ according to Shamir.Share and correctly constructs C_k for $k \in \{0, \ldots, t-1\}$; or
    2. can break the discrete-log assumption in $\mathbb{G}$.
- The statement follows. □

Adv. D:

Broadcast $(\tilde{c}_0, \ldots, \tilde{c}_{t-1}) \in G^t$

- P_i gets $\tilde{s}_i$

Honest parties $h \in H$:

$$\tilde{s}_h \cdot g = \sum_{k=0}^{t-1} \tilde{c}_k \cdot h^k \in G_q \qquad = g \cdot \sum_{k=0}^{t-1} \tilde{\alpha}_k \cdot h^k$$

$$g \cdot \tilde{\alpha}_k \to g(\tilde{\alpha}_k \cdot h^k)$$

Pedersen's NIVSS

Pedersen's NIVSS

- A relatively straightforward modification of Feldman's VSS.

Pedersen's NIVSS

- A relatively straightforward modification of Feldman's VSS.
- Hides the secret s from even a *computationally unbounded* adversary Adv.

Pedersen's NIVSS

- A relatively straightforward modification of Feldman's VSS.
- Hides the secret s from even a *computationally unbounded* adversary Adv.
- Adversary Adv still must be computationally bounded when the dealer is corrupt, as ~~security~~ relies on hardness of discrete-log.

Strong Comm.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_S(s, g, h)$:

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g, h)$:

- The dealer does the following:

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g, h)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g, h)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - Sample $f'_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in \{0, 1, \ldots, t-1\}$; set $f'(X) = \sum_{i=0}^{t-1} f'_i X^i$.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g, h)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - Sample $f'_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in \{0, 1, \ldots, t-1\}$; set $f'(X) = \sum_{i=0}^{t-1} f'_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i + h \cdot f'_i$.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g, h)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - Sample $f'_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in \{0, 1, \ldots, t-1\}$; set $f'(X) = \sum_{i=0}^{t-1} f'_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i + h \cdot f'_i$.
  - Broadcast $(C_0, \ldots, C_{t-1})$ to all parties.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g, h)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - Sample $f'_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in \{0, 1, \ldots, t-1\}$; set $f'(X) = \sum_{i=0}^{t-1} f'_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i + h \cdot f'_i$.
  - Broadcast $(C_0, \ldots, C_{t-1})$ to all parties.
  - Compute $S_i = (s_i, s'_i) = (f(i), f'(i))$ for $i \in [n]$.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{S}}(s, g, h)$:

- The dealer does the following:
  - Sample $f_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [t-1]$; set $f_0 = s$; set $f(X) = \sum_{i=0}^{t-1} f_i X^i$.
  - Sample $f'_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in \{0, 1, \ldots, t-1\}$; set $f'(X) = \sum_{i=0}^{t-1} f'_i X^i$.
  - For $i \in \{0, 1, \ldots, t-1\}$, compute $C_i = g \cdot f_i + h \cdot f'_i$.
  - Broadcast $(C_0, \ldots, C_{t-1})$ to all parties.
  - Compute $S_i = (s_i, s'_i) = (f(i), f'(i))$ for $i \in [n]$.
  - Output secret shares $(s_1, \ldots, s_n)$.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(S_1, \ldots, S_n)$:

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(S_1, \ldots, S_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$, parse $S_i = (s_i, s_i')$

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(S_1, \ldots, S_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$, parse $S_i = (s_i, s_i')$
  - Compute $\widetilde{C}_i = g \cdot s_i + h \cdot s_i'$.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(S_1, \ldots, S_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$, parse $S_i = (s_i, s_i')$
  - Compute $\widetilde{C}_i = g \cdot s_i + h \cdot s_i'$.
  - Check $\widetilde{C}_i \stackrel{?}{=} \sum_{k=0}^{t-1} (C_k) \cdot i^k$, kicking i out of the protocol if it fails.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(S_1, \ldots, S_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$, parse $S_i = (s_i, s_i')$
  - Compute $\widetilde{C}_i = g \cdot s_i + h \cdot s_i'$.
  - Check $\widetilde{C}_i \stackrel{?}{=} \sum_{k=0}^{t-1} (C_k) \cdot i^k$, kicking i out of the protocol if it fails.
- Let $s_{i_1}, \ldots, s_{i_t}$ be the (first) t shares that correctly authenticated.

Pedersen's NIVSS

(n, t) fixed, $\mathcal{M} = \mathbb{Z}_p$, $\mathcal{S} = \mathbb{Z}_p$, $g, h \in \mathbb{G}$ of order p

$\Pi_{\mathsf{R}}(S_1, \ldots, S_n)$:

- All parties authenticate the shares as follows: for each $i \in [n]$, parse $S_i = (s_i, s_i')$
  - Compute $\widetilde{C}_i = g \cdot s_i + h \cdot s_i'$.
  - Check $\widetilde{C}_i \stackrel{?}{=} \sum_{k=0}^{t-1} (C_k) \cdot i^k$, kicking i out of the protocol if it fails.
- Let $s_{i_1}, \ldots, s_{i_t}$ be the (first) t shares that correctly authenticated.
- Return $\mathsf{Shamir.Reconstruct}(s_{i_1}, \ldots, s_{i_t})$.

Pedersen's NIVSS: Correctness and Strong Commitment

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to Shamir.Share and (2) all honest parties output s at the end of Π_{R}.

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to Shamir.Share and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

By Feldman's scheme

Pedersen's NIVSS: Correctness and Strong Commitment

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to $\mathsf{Shamir.Share}$ and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

□

Pedersen's NIVSS: Correctness and Strong Commitment

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to $\mathsf{Shamir.Share}$ and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

Follows directly from Feldman's VSS.

□

Pedersen's NIVSS: Correctness and Strong Commitment

VSS Correctness

If D is honest, then (1) $s \in \mathbb{Z}_p$ is shared according to $\mathsf{Shamir.Share}$ and (2) all honest parties output s at the end of Π_{R}.

VSS Strong Commitment

If D is *corrupt*, then the execution of Π_{S} uniquely defines some secret value $\widetilde{s}$ such that $\widetilde{s}$ is shared among $\mathcal{P}$ according to $\mathsf{Shamir.Share}$ and all honest parties output $\widetilde{s}$ at the end of Π_{R}.

Proof.

Follows directly from Feldman's VSS. Crucially, for Strong Commitment, Adv must be computationally bounded so they cannot break the discrete-log assumption in $\mathbb{G}$. □

Pedersen's NIVSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Pedersen's NIVSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof.

Pedersen's NIVSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof.

- We argue that even if Adv is computationally *unbounded*, Adv learns nothing about secret s shared by honest dealer D.

Pedersen's NIVSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof.

- We argue that even if Adv is computationally *unbounded*, Adv learns nothing about secret s shared by honest dealer D.
- The values broadcast to all parties: $C_k = g \cdot f_k + h \cdot f'_k$ for $k \in \{0, 1, \ldots, t-1\}$, where $f_0 = s$ and all other f_i, f'_j sampled uniformly at random.

$k > 0 \quad C_k \equiv U_G$

Pedersen's NIVSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof.

- We argue that even if Adv is computationally *unbounded*, Adv learns nothing about secret s shared by honest dealer D.
- The values broadcast to all parties: $C_k = g \cdot f_k + h \cdot f'_k$ for $k \in \{0, 1, \ldots, t-1\}$, where $f_0 = s$ and all other f_i, f'_j sampled uniformly at random.
- Each $C_k \equiv U_{\mathbb{G}}$ by construction. C_0 : fixed + random = random

Pedersen's NIVSS: Privacy

VSS Privacy

If D is honest, then Adv (i.e., all corrupt parties) learn nothing about s during Π_{S}.

Proof.

- We argue that even if Adv is computationally *unbounded*, Adv learns nothing about secret s shared by honest dealer D.
- The values broadcast to all parties: $C_k = g \cdot f_k + h \cdot f'_k$ for $k \in \{0, 1, \ldots, t-1\}$, where $f_0 = s$ and all other f_i, f'_j sampled uniformly at random.
- Each $C_k \equiv U_{\mathbb{G}}$ by construction.
- Moreover, since Adv corrupts at most $t-1$ parties, the shares (s_i, s'_i) information-theoretically hide the polynomial interpolating s. □

Upper and Lower Bounds for VSS

Upper and Lower Bounds for VSS

- We've given a brief overview of VSS.

Upper and Lower Bounds for VSS

- We've given a brief overview of VSS.
- The field is much larger and very deeply related to other parts of crypto, especially

Upper and Lower Bounds for VSS

- We've given a brief overview of VSS.
- The field is much larger and very deeply related to other parts of crypto, especially
  - MPC; and

Upper and Lower Bounds for VSS

- We've given a brief overview of VSS.

- The field is much larger and very deeply related to other parts of crypto, especially
  - MPC; and
  - Secure Broadcast.

Upper and Lower Bounds for VSS

- We've given a brief overview of VSS.
- The field is much larger and very deeply related to other parts of crypto, especially
  - MPC; and
  - Secure Broadcast.
- I'll give you a brief overview of known upper and lower bounds, and some open questions related to VSS.

Lower Bounds for VSS

Lower Bounds for VSS

Theorem 1

Any perfectly secure VSS scheme satisfies $n > 3t$.

Lower Bounds for VSS

Theorem 1

Any perfectly secure VSS scheme satisfies $n > 3t$.

- Proved by Dolev, Dwork, Waarts, and Yung by relating perfectly secure VSS to *1-way perfectly-secure message transmission*.

Lower Bounds for VSS

Theorem 1

Any perfectly secure VSS scheme satisfies $n > 3t$.

- Proved by Dolev, Dwork, Waarts, and Yung by relating perfectly secure VSS to *1-way perfectly-secure message transmission*.
- An alternative (and slightly easier) argument is by the reduction to perfectly-secure *reliable broadcast* over P2P channels.

Lower Bounds for VSS

Theorem 1

Any perfectly secure VSS scheme satisfies $n > 3t$.

- Proved by Dolev, Dwork, Waarts, and Yung by relating perfectly secure VSS to *1-way perfectly-secure message transmission*.
- An alternative (and slightly easier) argument is by the reduction to perfectly-secure *reliable broadcast* over P2P channels.

Theorem 2

Perfectly secure VSS implies the existence of perfectly-secure reliable broadcast over P2P channels.

Lower Bounds for VSS

Theorem 1

Any perfectly secure VSS scheme satisfies $n > 3t$.

- Proved by Dolev, Dwork, Waarts, and Yung by relating perfectly secure VSS to *1-way perfectly-secure message transmission.*
- An alternative (and slightly easier) argument is by the reduction to perfectly-secure *reliable broadcast* over P2P channels.

Theorem 2

Perfectly secure VSS implies the existence of perfectly-secure reliable broadcast over P2P channels.

Theorem 3 (Pease, Shostak, Lamport)

Perfectly secure reliable broadcast over P2P channels requires $n > 3t$.

Lower Bounds for VSS

Lower Bounds for VSS

- Since broadcast is expensive, many results wish to understand the round complexity of VSS.

Lower Bounds for VSS

- Since broadcast is expensive, many results wish to understand the round complexity of VSS.

Theorem 4

Let $R \geqslant 1$ and $R' \leqslant R$ be integers. Then:

Lower Bounds for VSS

- Since broadcast is expensive, many results wish to understand the round complexity of VSS.

Theorem 4

Let $R \geqslant 1$ and $R' \leqslant R$ be integers. Then:

- *If $R = 1$, there does not exist any perfectly secure VSS scheme with (R, R') rounds in the sharing phase if either (a) $t > 1$ (for arbitrary n), or (b) $t = 1$ and $n \leqslant 4$*

Lower Bounds for VSS

- Since broadcast is expensive, many results wish to understand the round complexity of VSS.

Theorem 4

Let $R \geqslant 1$ and $R' \leqslant R$ be integers. Then:

- *If $R = 1$, there does not exist any perfectly secure VSS scheme with (R, R') rounds in the sharing phase if either (a) $t > 1$ (for arbitrary n), or (b) $t = 1$ and $n \leqslant 4$*
- *If $R = 2$, then perfectly secure VSS with (R, R') rounds in the sharing phase exists only if $n > 4t$.*

Lower Bounds for VSS

- Since broadcast is expensive, many results wish to understand the round complexity of VSS.

Theorem 4

Let $R \geqslant 1$ and $R' \leqslant R$ be integers. Then:

- *If $R = 1$, there does not exist any perfectly secure VSS scheme with (R, R') rounds in the sharing phase if either (a) $t > 1$ (for arbitrary n), or (b) $t = 1$ and $n \leqslant 4$*
- *If $R = 2$, then perfectly secure VSS with (R, R') rounds in the sharing phase exists only if $n > 4t$.*
- *If $R \geqslant 3$, then perfectly secure VSS with (R, R') rounds in the sharing phase exists only of $n > 3t$.*

Upper Bounds for VSS

Scheme	n	Round Complexity	Type	Sharing Semantic	Algebraic Structure	Communication Complexity
7BGW-VSS [12]	$n > 3t$	$(7, 5)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
5BGW-VSS [33]	$n > 3t$	$(5, 3)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
4GIKR-VSS [33]	$n > 3t$	$(4, 3)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
3GIKR-VSS [33]	$n > 3t$	$(3, 2)$	Type-II	RSS	$\mathbb{G}$	$\mathcal{O}(n \cdot \binom{n}{t} \log \lvert\mathbb{G}\rvert + \mathcal{BC}(n \cdot \binom{n}{t} \log \lvert\mathbb{G}\rvert))$
3FGGRS-VSS [32]	$n > 3t$	$(3, 2)$	Type-I	Not Applicable	$\mathbb{F}$	$\mathcal{O}(n^3 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^3 \log \lvert\mathbb{F}\rvert))$
3KKK-VSS [39]	$n > 3t$	$(3, 1)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^3 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^3 \log \lvert\mathbb{F}\rvert))$
3AKP-VSS [3]	$n > 3t$	$(3, 2)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^3 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^3 \log \lvert\mathbb{F}\rvert))$
2GIKR-VSS [33]	$n > 4t$	$(2, 1)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
1GIKR-VSS [33]	$n = 5, t = 1$	$(1, 0)$	Type-I	Not Applicable	$\mathbb{F}$	$\mathcal{O}(n \log \lvert\mathbb{F}\rvert)$

Table 1: Summary of the sharing phase of the perfectly-secure VSS schemes, with $\mathcal{BC}$ denoting communication over the broadcast channel.

Upper Bounds for VSS

Scheme	n	Round Complexity	Type	Sharing Semantic	Algebraic Structure	Communication Complexity
7BGW-VSS [12]	$n > 3t$	$(7, 5)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
5BGW-VSS [33]	$n > 3t$	$(5, 3)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
4GIKR-VSS [33]	$n > 3t$	$(4, 3)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
3GIKR-VSS [33]	$n > 3t$	$(3, 2)$	Type-II	RSS	$\mathbb{G}$	$\mathcal{O}(n \cdot \binom{n}{t} \log \lvert\mathbb{G}\rvert + \mathcal{BC}(n \cdot \binom{n}{t} \log \lvert\mathbb{G}\rvert))$
3FGGRS-VSS [32]	$n > 3t$	$(3, 2)$	Type-I	Not Applicable	$\mathbb{F}$	$\mathcal{O}(n^3 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^3 \log \lvert\mathbb{F}\rvert))$
3KKK-VSS [39]	$n > 3t$	$(3, 1)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^3 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^3 \log \lvert\mathbb{F}\rvert))$
3AKP-VSS [3]	$n > 3t$	$(3, 2)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^3 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^3 \log \lvert\mathbb{F}\rvert))$
2GIKR-VSS [33]	$n > 4t$	$(2, 1)$	Type-II	Shamir	$\mathbb{F}$	$\mathcal{O}(n^2 \log \lvert\mathbb{F}\rvert + \mathcal{BC}(n^2 \log \lvert\mathbb{F}\rvert))$
1GIKR-VSS [33]	$n = 5, t = 1$	$(1, 0)$	Type-I	Not Applicable	$\mathbb{F}$	$\mathcal{O}(n \log \lvert\mathbb{F}\rvert)$

Table 1: Summary of the sharing phase of the perfectly-secure VSS schemes, with $\mathcal{BC}$ denoting communication over the broadcast channel.

- Taken from https://eprint.iacr.org/2021/445.pdf

Notes on Upper and Lower Bounds

- All above results assume:

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and
  - a Secure Broadcast Channel.

+ Comp. unbounded adversaries

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and
  - a Secure Broadcast Channel.

- Can replace Secure Broadcast with *reliable broadcast protocols* over P2P channels.

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and
  - a Secure Broadcast Channel.

- Can replace Secure Broadcast with *reliable broadcast protocols* over P2P channels.
  - Increases the actual round complexity and/or number of bits communicated.

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and
  - a Secure Broadcast Channel.

- Can replace Secure Broadcast with *reliable broadcast protocols* over P2P channels.
  - Increases the actual round complexity and/or number of bits communicated.
  - Broadcast delivers messages with high probability, or with many rounds if you want messages guaranteed to be delivered.

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and
  - a Secure Broadcast Channel.

- Can replace Secure Broadcast with *reliable broadcast protocols* over P2P channels.
  - Increases the actual round complexity and/or number of bits communicated.
  - Broadcast delivers messages with high probability, or with many rounds if you want messages guaranteed to be delivered.

- Can also model in *asynchronous networks*.

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and
  - a Secure Broadcast Channel.

- Can replace Secure Broadcast with *reliable broadcast protocols* over P2P channels.
  - Increases the actual round complexity and/or number of bits communicated.
  - Broadcast delivers messages with high probability, or with many rounds if you want messages guaranteed to be delivered.

- Can also model in *asynchronous networks*.
  - Messages reach their recipients *eventually*.

Notes on Upper and Lower Bounds

- All above results assume:
  - Perfectly secure VSS,
  - Synchronous rounds of communication,
  - Secure P2P channels, and
  - a Secure Broadcast Channel.

- Can replace Secure Broadcast with *reliable broadcast protocols* over P2P channels.
  - Increases the actual round complexity and/or number of bits communicated.
  - Broadcast delivers messages with high probability, or with many rounds if you want messages guaranteed to be delivered.

- Can also model in *asynchronous networks*.
  - Messages reach their recipients *eventually*.

- Can also model the network as *hybrid*: async phases and sync phases.

(Some) Open Problems in Perfectly Secure VSS

- *Communication Complexity* of VSS (i.e., number of bits sent during the protocols).
  - 3-round *efficient* VSS schemes have $n\times$ more communication than $\geqslant 4$ round schemes. Can this gap be closed?
  - Better than trivial lower bound on communication complexity of perfectly secure VSS?

- *Broadcast Complexity*: number of bits sent via broadcast.
  - All schemes with $n > 3t$ have broadcast complexity $O(L \cdot n^3 \log |\mathbb{F}|)$, where L is the number of secrets shared by the scheme.
  - For asynchronous VSS, there is a scheme with $O(n^2 \cdot \log(n))$ broadcast complexity (independent of L).
  - Can we get this same bound for the synchronous setting?

Next Time: Function Secret Sharing