Applied Cryptography

Lecture 1

Applied Cryptography

Lecture 1

Our first encounter with secrecy:
Secret-Sharing

Secrecy

Secrecy

- Cryptography is all about "controlling access to information"
  - Access to learning and/or influencing information

Secrecy

- Cryptography is all about "controlling access to information"
  - Access to learning and/or influencing information
- One of the aspects of access control is secrecy

A Game

A Game

- A “dealer” and two “players” Alice and Bob

A Game

- A "dealer" and two "players" Alice and Bob
- Dealer has a message, say two bits m_1m_2

A Game

- A "dealer" and two "players" Alice and Bob
- Dealer has a message, say two bits $m_1 m_2$
- She wants to "share" it among the two players so that neither player by itself learns anything about the message, but together they can find it

A Game

- A “dealer” and two “players” Alice and Bob
- Dealer has a message, say two bits m_1m_2
- She wants to “share” it among the two players so that neither player by itself learns anything about the message, but together they can find it
- Bad idea: Give m_1 to Alice and m_2 to Bob

A Game

- A "dealer" and two "players" Alice and Bob
- Dealer has a message, say two bits m_1m_2
- She wants to "share" it among the two players so that neither player by itself learns anything about the message, but together they can find it
- Bad idea: Give m_1 to Alice and m_2 to Bob
- Other ideas?

Sharing a bit

Sharing a bit

- To share a bit m, Dealer picks a uniformly random bit b and gives $a := m \oplus b$ to Alice and b to Bob

Sharing a bit

- To share a bit m, Dealer picks a uniformly random bit b and gives a := m⊕b to Alice and b to Bob
  - Bob learns nothing (b is a random bit)

Sharing a bit

- To share a bit m, Dealer picks a uniformly <u>random</u> bit b and gives a := m⊕b to Alice and b to Bob
  - Bob learns nothing (b is a random bit)
  - Alice learns nothing either: for each possible value of m (0 or 1), a is a random bit (0 w.p. ½, 1 w.p. ½)

Sharing a bit

- To share a bit m, Dealer picks a uniformly random bit b and gives $a := m \oplus b$ to Alice and b to Bob
  - Bob learns nothing (b is a random bit)
  - Alice learns nothing either: for each possible value of m (0 or 1), a is a random bit (0 w.p. ½, 1 w.p. ½)

m = 0 → (a,b) = (0,0) or (1,1)
m = 1 → (a,b) = (1,0) or (0,1)

Sharing a bit

- To share a bit m, Dealer picks a uniformly random bit b and gives a := m⊕b to Alice and b to Bob
  - Bob learns nothing (b is a random bit)
  - Alice learns nothing either: for each possible value of m (0 or 1), a is a random bit (0 w.p. ½, 1 w.p. ½)
    - Her view is independent of the message

m = 0 → (a,b) = (0,0) or (1,1)
m = 1 → (a,b) = (1,0) or (0,1)

Sharing a bit

- To share a bit m, Dealer picks a uniformly <u>random</u> bit b and gives a := m⊕b to Alice and b to Bob
  - Bob learns nothing (b is a random bit)
  - Alice learns nothing either: for each possible value of m (0 or 1), a is a random bit (0 w.p. ½, 1 w.p. ½)
    - Her view is <u>independent</u> of the message
  - Together they can recover m as a⊕b

m = 0 → (a,b) = (0,0) or (1,1)
m = 1 → (a,b) = (1,0) or (0,1)

Sharing a bit

- To share a bit m, Dealer picks a uniformly random bit b and gives $a := m \oplus b$ to Alice and b to Bob
  - Bob learns nothing (b is a random bit)
  - Alice learns nothing either: for each possible value of m (0 or 1), a is a random bit (0 w.p. ½, 1 w.p. ½)
    - Her view is independent of the message
  - Together they can recover m as $a \oplus b$
- Multiple bits can be shared independently: as, $m_1m_2 = a_1a_2 \oplus b_1b_2$

m = 0 → (a,b) = (0,0) or (1,1)
m = 1 → (a,b) = (1,0) or (0,1)

Sharing a bit

- To share a bit m, Dealer picks a uniformly random bit b and gives $a := m \oplus b$ to Alice and b to Bob
  - Bob learns nothing (b is a random bit)
  - Alice learns nothing either: for each possible value of m (0 or 1), a is a random bit (0 w.p. ½, 1 w.p. ½)
    - Her view is independent of the message
  - Together they can recover m as $a \oplus b$
- Multiple bits can be shared independently: as, $\underline{m_1 m_2} = \underline{a_1 a_2} \oplus \underline{b_1 b_2}$
- Note: any one share can be chosen before knowing the message [why?]

$m = 0 \rightarrow (a,b) = (0,0)$ or $(1,1)$
$m = 1 \rightarrow (a,b) = (1,0)$ or $(0,1)$

Secrecy

Secrecy

- Is the message m really secret?

Secrecy

- Is the message m really secret?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing

Secrecy

- Is the message m really <u>secret</u>?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing
  - Worse, if they already know something about m, they can do better (Note: we didn't say m is random!)

Secrecy

- Is the message m really <u>secret</u>?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing
  - Worse, if they already know something about m, they can do better (Note: we didn't say m is random!)
- But this they could have done without obtaining the shares

Secrecy

- Is the message m really <u>secret</u>?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing
  - Worse, if they already know something about m, they can do better (Note: we didn't say m is random!)
- But this they could have done without obtaining the shares
- The shares did not leak any <u>additional</u> information to either party

Secrecy

- Is the message m really <u>secret</u>?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing
  - Worse, if they already know something about m, they can do better (Note: we didn't say m is random!)
- But this they could have done without obtaining the shares
- The shares did not leak any <u>additional</u> information to either party
- Crypto goal: **<u>preserving</u>** secrecy

Secrecy

- Is the message m really <u>secret</u>?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing
  - Worse, if they already know something about m, they can do better (Note: we didn't say m is random!)
- But this they could have done without obtaining the shares
- The shares did not leak any <u>additional</u> information to either party
- Crypto goal: **<u>preserving</u>** secrecy
  - View is <u>independent</u> of the message

Secrecy

- Is the message m really <u>secret</u>?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing
  - Worse, if they already know something about m, they can do better (Note: we didn't say m is random!)
- But this they could have done without obtaining the shares
- The shares did not leak any <u>additional</u> information to either party
- Crypto goal: <u>**preserving**</u> secrecy
  - View is <u>independent</u> of the message
    - i.e., for all possible values of the message, the view is distributed the same way

Secret-Sharing

Secret-Sharing

- More general secret-sharing

Secret-Sharing

- More general secret-sharing
    - Allow more than two parties (how?)

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged <u>subsets</u> of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged <u>subsets</u> of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)
- Very useful

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged <u>subsets</u> of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)
- Very useful
  - Direct applications (distributed storage of data or keys)

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged <u>subsets</u> of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)
- Very useful
  - Direct applications (distributed storage of data or keys)
  - Important component in other cryptographic constructions

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged subsets of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)
- Very useful
  - Direct applications (distributed storage of data or keys)
  - Important component in other cryptographic constructions
    - Amplifying secrecy of various primitives

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged subsets of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)
- Very useful
  - Direct applications (distributed storage of data or keys)
  - Important component in other cryptographic constructions
    - Amplifying secrecy of various primitives
    - Secure multi-party computation

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged subsets of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)
- Very useful
  - Direct applications (distributed storage of data or keys)
  - Important component in other cryptographic constructions
    - Amplifying secrecy of various primitives
    - Secure multi-party computation
    - Attribute-Based Encryption

Secret-Sharing

- More general secret-sharing
  - Allow more than two parties (how?)
  - Privileged subsets of parties should be able to reconstruct the secret (not necessarily just the entire set of parties)
- Very useful
  - Direct applications (distributed storage of data or keys)
  - Important component in other cryptographic constructions
    - Amplifying secrecy of various primitives
    - Secure multi-party computation
    - Attribute-Based Encryption
    - Leakage resilience ...

Threshold Secret-Sharing

Threshold Secret-Sharing

- (n,t)-secret-sharing

Threshold Secret-Sharing

- (n,t)-secret-sharing
  - Divide a message m into n shares $s_1,...,s_n$, such that any t shares are enough to reconstruct the secret

Threshold Secret-Sharing

- (n,t)-secret-sharing
  - Divide a message m into n shares $s_1,\ldots,s_n$, such that any t shares are enough to reconstruct the secret
  - Up to t-1 shares should have no information about the secret

Threshold Secret-Sharing

- (n,t)-secret-sharing
  - Divide a message m into n shares $s_1,\ldots,s_n$, such that any t shares are enough to reconstruct the secret
  - Up to t-1 shares should have no information about the secret
    - i.e., say, $(s_1,\ldots,s_{t-1})$ identically distributed for every m in the message space

Threshold Secret-Sharing

- (n,t)-secret-sharing
  - Divide a message m into n shares $s_1,\ldots,s_n$, such that any t shares are enough to reconstruct the secret
  - Up to t-1 shares should have no information about the secret
    - i.e., say, $(s_1,\ldots,s_{t-1})$ identically distributed for every m in the message space
    - our previous example: (2,2) secret-sharing

Threshold Secret-Sharing

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. G = $\mathbb{Z}_2$ (group of bits, with xor as the group operation)

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. G = $\mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, G = $\mathbb{Z}_2{}^d$ (group of d-bit strings)

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. G = $\mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, G = $\mathbb{Z}_2{}^d$ (group of d-bit strings)
    - or, G = $\mathbb{Z}_p$ (group of integers mod p)

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. G = $\mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, G = $\mathbb{Z}_2{}^d$ (group of d-bit strings)
    - or, G = $\mathbb{Z}_p$ (group of integers mod p)
  - Share(M):

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. G = $\mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, G = $\mathbb{Z}_2{}^d$ (group of d-bit strings)
    - or, G = $\mathbb{Z}_p$ (group of integers mod p)
  - Share(M):
    - Pick $s_1,\ldots,s_{n-1}$ uniformly at random from G

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. G = $\mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, G = $\mathbb{Z}_2{}^d$ (group of d-bit strings)
    - or, G = $\mathbb{Z}_p$ (group of integers mod p)
  - Share(M):
    - Pick $s_1,\dots,s_{n-1}$ uniformly at random from G
    - Let $s_n = M - (s_1 + \dots + s_{n-1})$

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. G = $\mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, G = $\mathbb{Z}_2^{\ d}$ (group of d-bit strings)
    - or, G = $\mathbb{Z}_p$ (group of integers mod p)
  - Share(M):
    - Pick $s_1,\ldots,s_{n-1}$ uniformly at random from G
    - Let $s_n = M - (s_1 + \ldots + s_{n-1})$
  - Reconstruct($s_1,\ldots,s_n$): $M = s_1 + \ldots + s_n$

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. $G = \mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, $G = \mathbb{Z}_2{}^d$ (group of d-bit strings)
    - or, $G = \mathbb{Z}_p$ (group of integers mod p)
  - Share(M):
    - Pick $s_1,\ldots,s_{n-1}$ uniformly at random from G
    - Let $s_n = M - (s_1 + \ldots + s_{n-1})$
  - Reconstruct($s_1,\ldots,s_n$): $M = s_1 + \ldots + s_n$
  - Claim: This is an (n,n) secret-sharing scheme [Why?]

Threshold Secret-Sharing

- Construction: (n,n) secret-sharing
  - Message-space = share-space = G, a group
    - e.g. $G = \mathbb{Z}_2$ (group of bits, with xor as the group operation)
    - or, $G = \mathbb{Z}_2{}^d$ (group of d-bit strings)
    - or, $G = \mathbb{Z}_p$ (group of integers mod p)
  - Share(M):
    - Pick $s_1,\ldots,s_{n-1}$ uniformly at random from G
    - Let $s_n = M - (s_1 + \ldots + s_{n-1})$
  - Reconstruct($s_1,\ldots,s_n$): $M = s_1 + \ldots + s_n$
  - Claim: This is an (n,n) secret-sharing scheme [Why?]

Additive Secret-Sharing

PROOF

Additive Secret-Sharing: Proof

- Share(M):
  - Pick $s_1,\ldots,s_{n-1}$ uniformly at random from G
  - Let $s_n = M - (s_1 + \ldots + s_{n-1})$
- Reconstruct($s_1,\ldots,s_n$): $M = s_1 + \ldots + s_n$
- Claim: Upto n-1 shares give no information about M
- Proof: Let $T \subseteq \{1,\ldots,n\}$, $|T| = n-1$. We shall show that $\{ s_i \}_{i\in T}$ is distributed the same way (in fact, uniformly) irrespective of what M is.
  - For concreteness consider $T = \{2,\ldots,n\}$. Fix any (n-1)-tuple of elements in G, $(g_1,\ldots,g_{n-1}) \in G^{n-1}$. To prove $\Pr[(s_2,\ldots,s_n)=(g_1,\ldots,g_{n-1})]$ is independent of M.
  - Fix any M.
  - $(s_2,\ldots,s_n) = (g_1,\ldots,g_{n-1}) \Leftrightarrow (s_2,\ldots,s_{n-1}) = (g_1,\ldots,g_{n-2})$ and $s_1 = M-(g_1+\ldots+g_{n-1})$.
  - So $\Pr[(s_2,\ldots,s_n)=(g_1,\ldots,g_{n-1})] = \Pr[(s_1,\ldots,s_{n-1})=(M-(g_1+\ldots+g_{n-1}), g_1,\ldots,g_{n-2})]$
  - But $\Pr[(s_1,\ldots,s_{n-1})=(M-(g_1+\ldots+g_{n-1}), g_1,\ldots,g_{n-2})] = 1/|G|^{n-1}$, since $(s_1,\ldots,s_{n-1})$ are picked uniformly at random
  - Hence $\Pr[(s_2,\ldots,s_n)=(g_1,\ldots,g_{n-1})] = 1/|G|^{n-1}$, irrespective of M. □

Threshold Secret-Sharing

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)

n distinct, non-0 field elements

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for $i=1,\ldots,n < |F|$)

n distinct, non-0 field elements

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - <u>Share</u>(M): pick random r. Let $s_i = r \cdot i + M$ (for i=1,...,n < |F|)

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for $i=1,\ldots,n < |F|$)
  - Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\, i$

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for $i=1,\ldots,n < |F|$)
  - Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\, i$
  - Each s_i by itself is uniformly distributed, irrespective of M [Why?]

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for i=1,...,n < |F|)
  - Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\ i$
  - Each s_i by itself is uniformly distributed, irrespective of M [Why?]
  - "Geometric" interpretation

n distinct, non-0 field elements

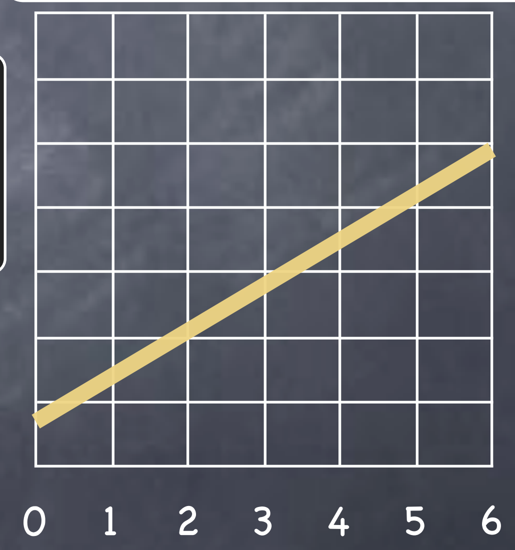
Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d

Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for i=1,...,n < |F|)
  - Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\,i$
  - Each s_i by itself is uniformly distributed, irrespective of M [Why?]
  - "Geometric" interpretation

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d

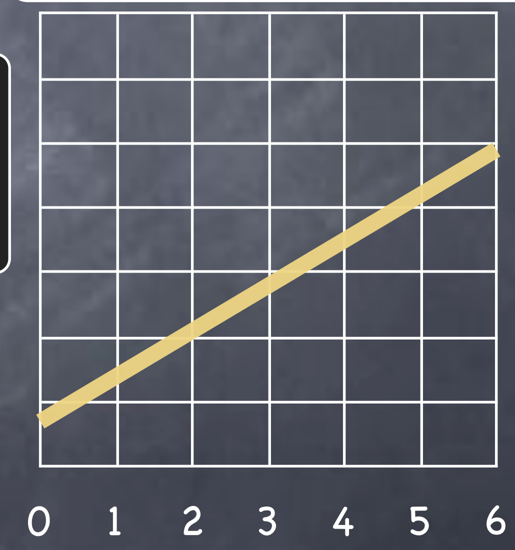


Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for $i=1,\ldots,n < |F|$)
  - Reconstruct(s_i, s_j): $r = (s_i-s_j)/(i-j)$; $M = s_i - r\, i$
  - Each s_i by itself is uniformly distributed, irrespective of M [Why?]
  - "Geometric" interpretation
    - Sharing picks a random "line" $y = f(x)$, such that $f(0)=M$. Shares $s_i = f(i)$.

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i+M=d$, for every value of d

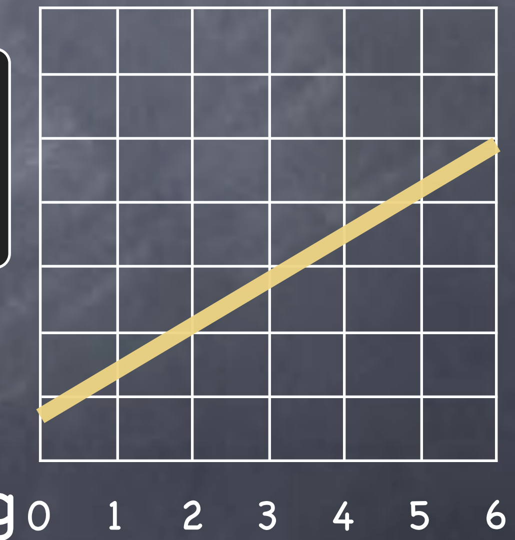


Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for $i=1,\ldots,n < |F|$)
  - Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\,i$
  - Each s_i by itself is uniformly distributed, irrespective of M [Why?]
  - "Geometric" interpretation
    - Sharing picks a random "line" $y = f(x)$, such that $f(0)=M$. Shares $s_i = f(i)$.
    - s_i is independent of M: exactly one line passing through (i,s_i) and $(0,M')$ for each secret M'

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d

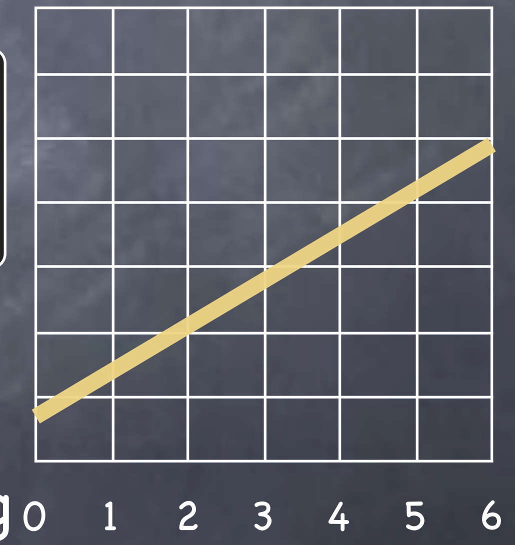


Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for $i=1,\ldots,n < |F|$)
  - Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\,i$
  - Each s_i by itself is uniformly distributed, irrespective of M [Why?]
  - "Geometric" interpretation
    - Sharing picks a random "line" $y = f(x)$, such that $f(0)=M$. Shares $s_i = f(i)$.
    - s_i is independent of M: exactly one line passing through (i,s_i) and $(0,M')$ for each secret M'
    - But can reconstruct the line from two points!

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d

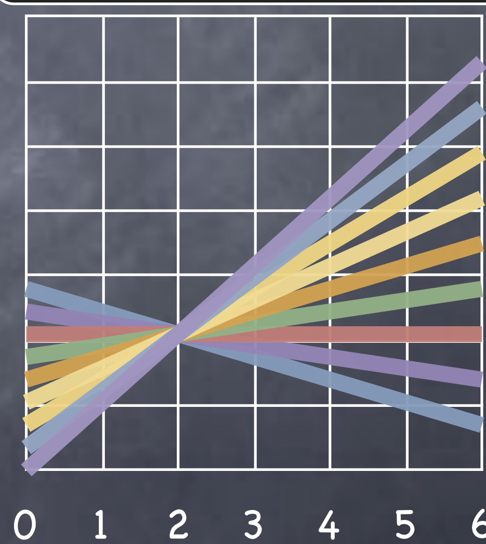


Threshold Secret-Sharing

- Construction: (n,2) secret-sharing
- Message-space = share-space = F, a field (e.g. integers mod a prime, $\mathbb{F}_p$)
  - Share(M): pick random r. Let $s_i = r \cdot i + M$ (for i=1,...,n < |F|)
  - Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\,i$
  - Each s_i by itself is uniformly distributed, irrespective of M [Why?]
  - "Geometric" interpretation
    - Sharing picks a random "line" y = f(x), such that f(0)=M. Shares s_i = f(i).
    - s_i is independent of M: exactly one line passing through (i,s_i) and (0,M′) for each secret M′
    - But can reconstruct the line from two points!

n distinct, non-0 field elements

Since i^{-1} exists, exactly one solution for $r \cdot i + M = d$, for every value of d



PROOF

(n,2) Secret-Sharing: Proof

- Share(M): pick random $r \leftarrow F$. Let $s_i = r \cdot i + M$ (for $i=1,\ldots,n < |F|$)
- Reconstruct(s_i, s_j): $r = (s_i - s_j)/(i-j)$; $M = s_i - r\,i$
- Claim: Any one share gives no information about M
- Proof: For any $i \in \{1,\ldots,n\}$ we shall show that s_i is distributed the same way (in fact, uniformly) irrespective of what M is.
- Consider any $g \in F$. We shall show that $\Pr[\ s_i = g\ ]$ is independent of M.
- Fix any M.
- For any $g \in F$, $s_i = g \Leftrightarrow r \cdot i + M = g \Leftrightarrow r = (g-M) \cdot i^{-1}$ (since $i \neq 0$)
- So, $\Pr[\ s_i = g\ ] = \Pr[\ r = (g-M) \cdot i^{-1}\ ] = 1/|F|$, since r is chosen uniformly at random □

Threshold Secret-Sharing

Threshold Secret-Sharing

- (n,t) secret-sharing in a field

Threshold Secret-Sharing

- (n,t) secret-sharing in a field
  - Generalizing the geometric/algebraic view: instead of lines, use polynomials

Threshold Secret-Sharing

Shamir Secret-Sharing

- (n,t) secret-sharing in a field
  - Generalizing the geometric/algebraic view: instead of lines, use polynomials

Threshold Secret-Sharing

Shamir Secret-Sharing

- (n,t) secret-sharing in a field
  - Generalizing the geometric/algebraic view: instead of lines, use polynomials
    - Share(m): Pick a random degree t-1 polynomial f(X), such that f(0)=M. Shares are s_i = f(i).

Threshold Secret-Sharing

Shamir Secret-Sharing

- (n,t) secret-sharing in a field
  - Generalizing the geometric/algebraic view: instead of lines, use polynomials
    - Share(m): Pick a random degree t-1 polynomial f(X), such that f(0)=M. Shares are $s_i = f(i)$.
      - Random polynomial with f(0)=M: $c_0 + c_1X + c_2X^2 + \ldots + c_{t-1}X^{t-1}$ by picking c_0=M and $c_1,\ldots,c_{t-1}$ at random.

Threshold Secret-Sharing

Shamir Secret-Sharing

- (n,t) secret-sharing in a field
  - Generalizing the geometric/algebraic view: instead of lines, use polynomials
    - Share(m): Pick a random degree t-1 polynomial f(X), such that f(0)=M. Shares are $s_i = f(i)$.
      - Random polynomial with f(0)=M: $c_0 + c_1X + c_2X^2 + \ldots + c_{t-1}X^{t-1}$ by picking c_0=M and $c_1,\ldots,c_{t-1}$ at random.
    - Reconstruct($s_1,\ldots,s_t$): Lagrange interpolation to find M=c_0

Threshold Secret-Sharing

Shamir Secret-Sharing

- (n,t) secret-sharing in a field
  - Generalizing the geometric/algebraic view: instead of lines, use polynomials
    - Share(m): Pick a random degree t-1 polynomial f(X), such that f(0)=M. Shares are $s_i = f(i)$.
      - Random polynomial with f(0)=M: $c_0 + c_1X + c_2X^2 + \ldots + c_{t-1}X^{t-1}$ by picking c_0=M and $c_1,\ldots,c_{t-1}$ at random.
    - Reconstruct($s_1,\ldots,s_t$): Lagrange interpolation to find $M=c_0$
      - Need t points to reconstruct the polynomial. Given t-1 points, there is exactly one polynomial passing through (0,M') for each M'

Lagrange Interpolation

Lagrange Interpolation

- Given t distinct points on a degree t-1 polynomial (univariate, over some field of more than t elements), reconstruct the entire polynomial (i.e., find all t co-efficients)

Lagrange Interpolation

- Given t distinct points on a degree t-1 polynomial (univariate, over some field of more than t elements), reconstruct the entire polynomial (i.e., find all t co-efficients)
  - t variables: $c_0,\ldots,c_{t-1}$. t equations: $1.c_0 + i.c_1 + i^2.c_2 + \ldots\ i^{t-1}.c_{t-1} = s_i$

Lagrange Interpolation

- Given t distinct points on a degree t-1 polynomial (univariate, over some field of more than t elements), reconstruct the entire polynomial (i.e., find all t co-efficients)
  - t variables: $c_0,\ldots,c_{t-1}$. t equations: $1.c_0 + i.c_1 + i^2.c_2 + \ldots i^{t-1}.c_{t-1} = s_i$
  - A linear system: $W\mathbf{c}=\mathbf{s}$, where W a txt matrix with $W_i= (1\ i\ i^2 \ldots i^{t-1})$

Lagrange Interpolation

- Given t distinct points on a degree t-1 polynomial (univariate, over some field of more than t elements), reconstruct the entire polynomial (i.e., find all t co-efficients)
  - t variables: $c_0,\ldots,c_{t-1}$. t equations: $1.c_0 + i.c_1 + i^2.c_2 + \ldots i^{t-1}.c_{t-1} = s_i$
  - A linear system: $W\mathbf{c}=\mathbf{s}$, where W a txt matrix with $W_i= (1\ i\ i^2 \ldots i^{t-1})$
  - W is a Vandermonde matrix: invertible

Lagrange Interpolation

- Given t distinct points on a degree t-1 polynomial (univariate, over some field of more than t elements), reconstruct the entire polynomial (i.e., find all t co-efficients)
  - t variables: $c_0,\ldots,c_{t-1}$. t equations: $1.c_0 + i.c_1 + i^2.c_2 + \ldots\ i^{t-1}.c_{t-1} = s_i$
  - A linear system: W**c**=**s**, where W a txt matrix with $W_i = (1\ i\ i^2 \ldots i^{t-1})$
  - W is a Vandermonde matrix: invertible
    - $\mathbf{c} = W^{-1}\mathbf{s}$

More General Access Structures

More General Access Structures

- (n,t)-secret-sharing allowed any t (or more) parties to reconstruct the secret

More General Access Structures

- (n,t)-secret-sharing allowed any t (or more) parties to reconstruct the secret
  - i.e., "access structure" $\mathcal{A} = \{S: |S| \geq t\}$, is the set of all subsets of parties who can reconstruct the secret

More General Access Structures

- (n,t)-secret-sharing allowed any t (or more) parties to reconstruct the secret
    - i.e., "access structure" $\mathcal{A} = \{S: |S| \geq t\}$, is the set of all subsets of parties who can reconstruct the secret
    - In general access structure could be any monotonic set of subsets

More General Access Structures

- (n,t)-secret-sharing allowed any t (or more) parties to reconstruct the secret
  - i.e., "access structure" $\mathcal{A} = \{S: |S| \geq t\}$, is the set of all subsets of parties who can reconstruct the secret
- In general access structure could be any monotonic set of subsets

If $S^* \in \mathcal{A}$, then for all $S \supseteq S^*$, $S \in \mathcal{A}$.

More General Access Structures

- (n,t)-secret-sharing allowed any t (or more) parties to reconstruct the secret
  - i.e., "access structure" $\mathcal{A}$ = {S: |S| ≥ t }, is the set of all subsets of parties who can reconstruct the secret
  - In general access structure could be any monotonic set of subsets

If $S^* \in \mathcal{A}$, then for all $S \supseteq S^*$, $S \in \mathcal{A}$.

- Shamir's secret-sharing solves threshold secret-sharing. How about the others?

More General Access Structures

More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.

More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.
  - Works, but very "inefficient"

More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.
  - Works, but very "inefficient"
    - How big is $\mathcal{B}$? (Say when $\mathcal{A}$ is a threshold access structure)

More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.
  - Works, but very "inefficient"
    - How big is $\mathcal{B}$? (Say when $\mathcal{A}$ is a threshold access structure)

$|\mathcal{B}|$ = (n choose t)

More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.
  - Works, but very "inefficient"
    - How big is $\mathcal{B}$? (Say when $\mathcal{A}$ is a threshold access structure)

      $|\mathcal{B}|$ = (n choose t)
    - Total share complexity = $\Sigma_{S \in \mathcal{B}} |S|$ field elements. (Compare with Shamir's scheme: n field elements in all.)

More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.
  - Works, but very "inefficient"
    - How big is $\mathcal{B}$? (Say when $\mathcal{A}$ is a threshold access structure)
    - Total share complexity = $\Sigma_{S\in\mathcal{B}}$ |S| field elements. (Compare with Shamir's scheme: n field elements in all.)

$|\mathcal{B}|$ = (n choose t)

t·(n choose t)

More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.
  - Works, but very "inefficient"
    - How big is $\mathcal{B}$? (Say when $\mathcal{A}$ is a threshold access structure)

      $|\mathcal{B}|$ = (n choose t)
    - Total share complexity = $\Sigma_{S\in\mathcal{B}}$ |S| field elements. (Compare with Shamir's scheme: n field elements in all.)

      $t\cdot$(n choose t)
  - More efficient schemes known for large classes of access structures

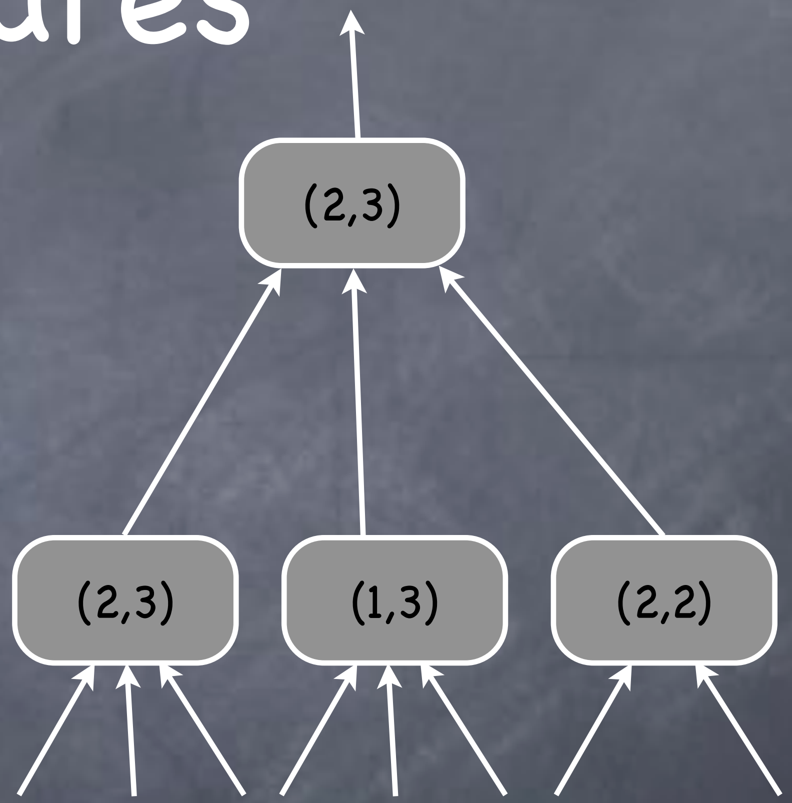
More General Access Structures

More General Access Structures

- A simple generalization of threshold access structures

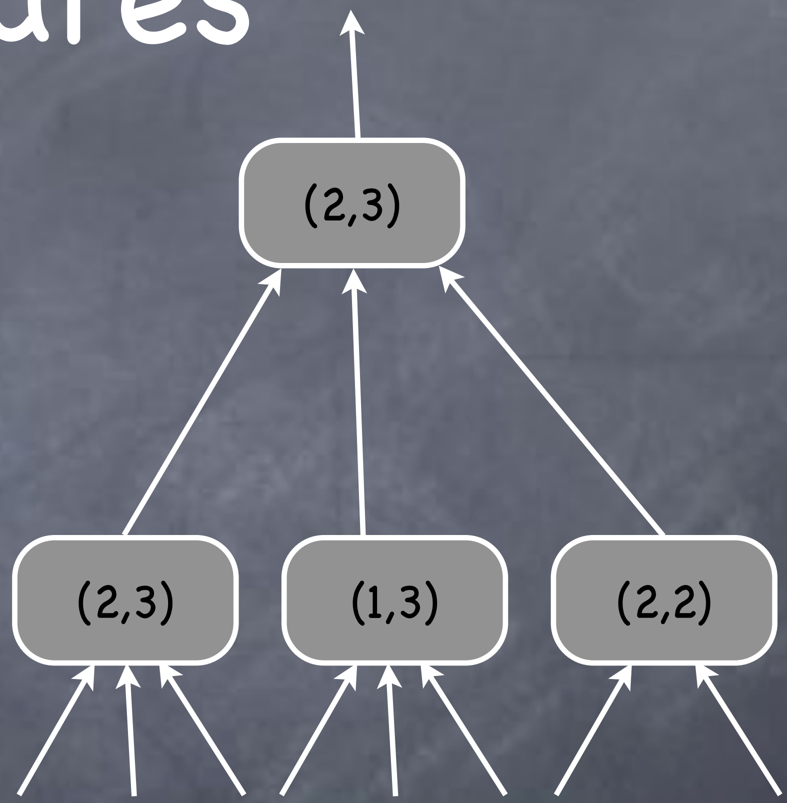
More General Access Structures

- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure



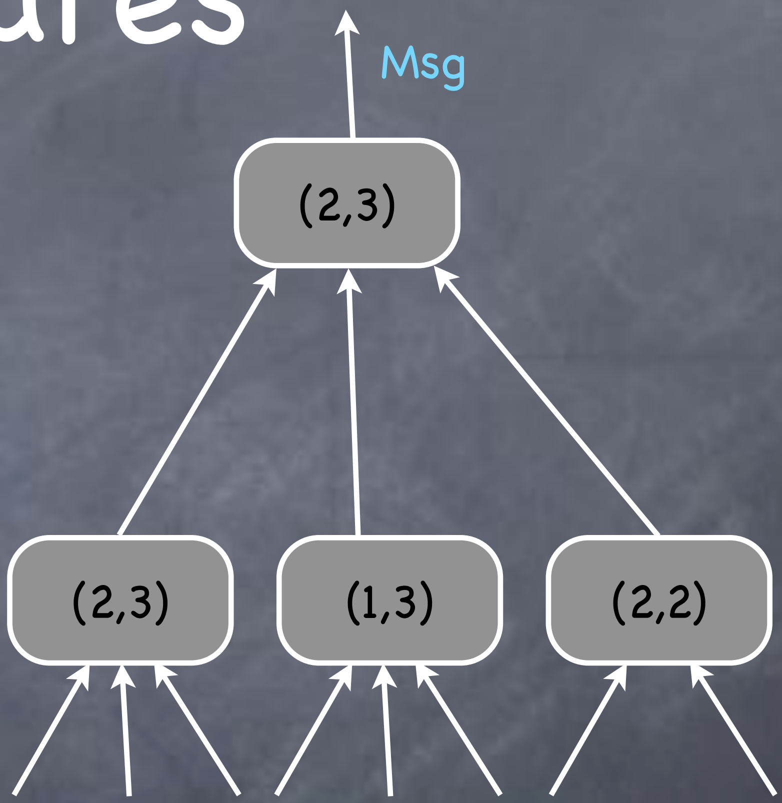
More General Access Structures

- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure
  - Can realize by recursively threshold secret-sharing the shares



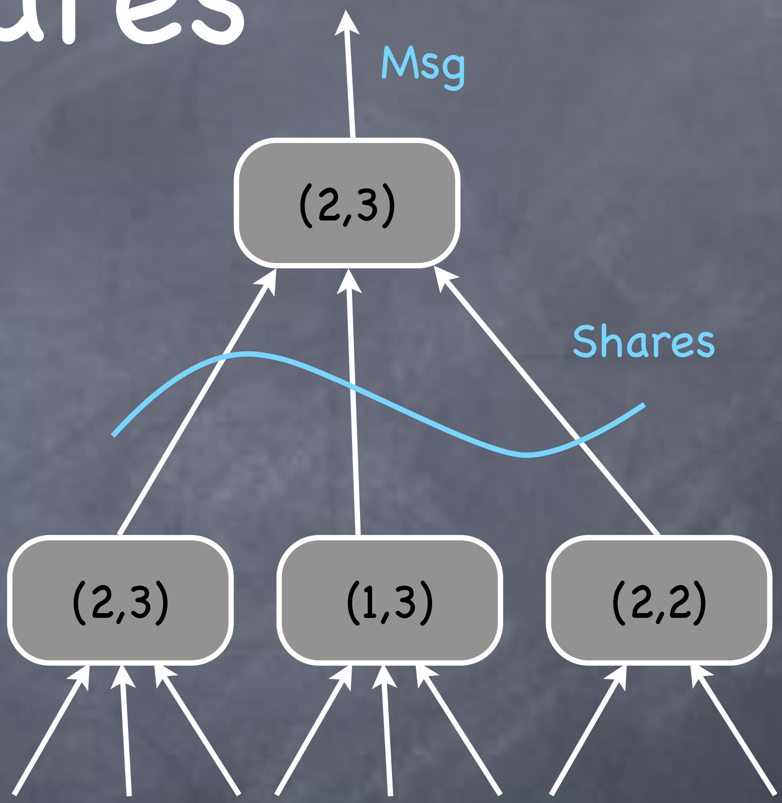
More General Access Structures

- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure
  - Can realize by recursively threshold secret-sharing the shares



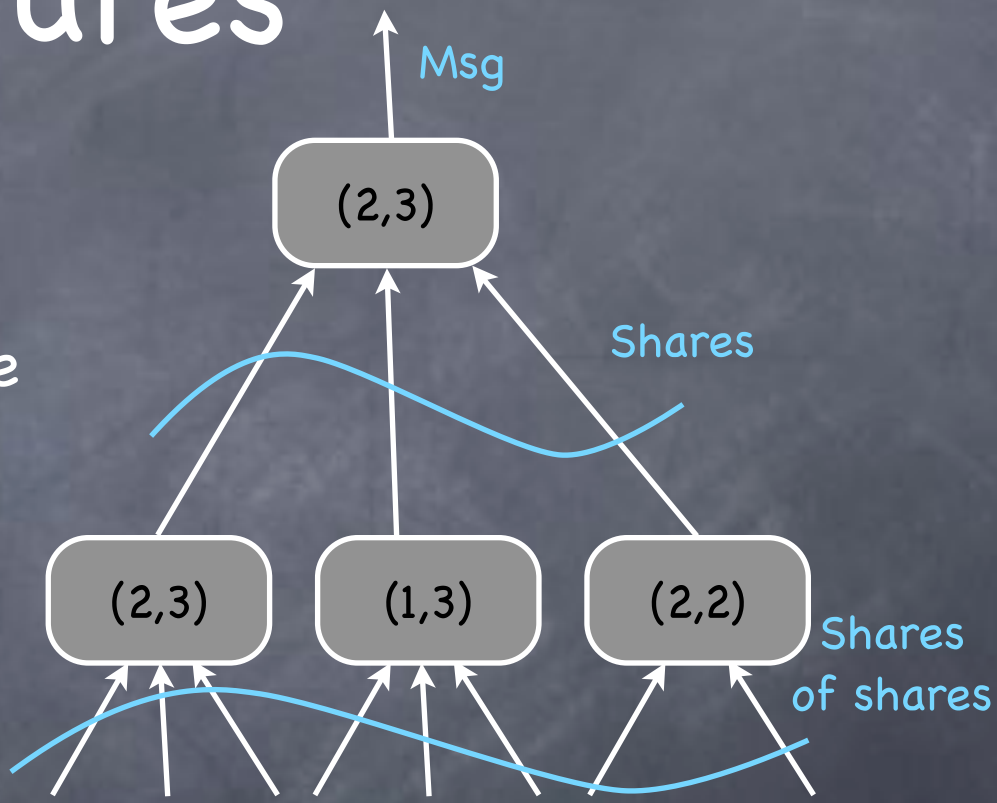
More General Access Structures

- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure
  - Can realize by recursively threshold secret-sharing the shares



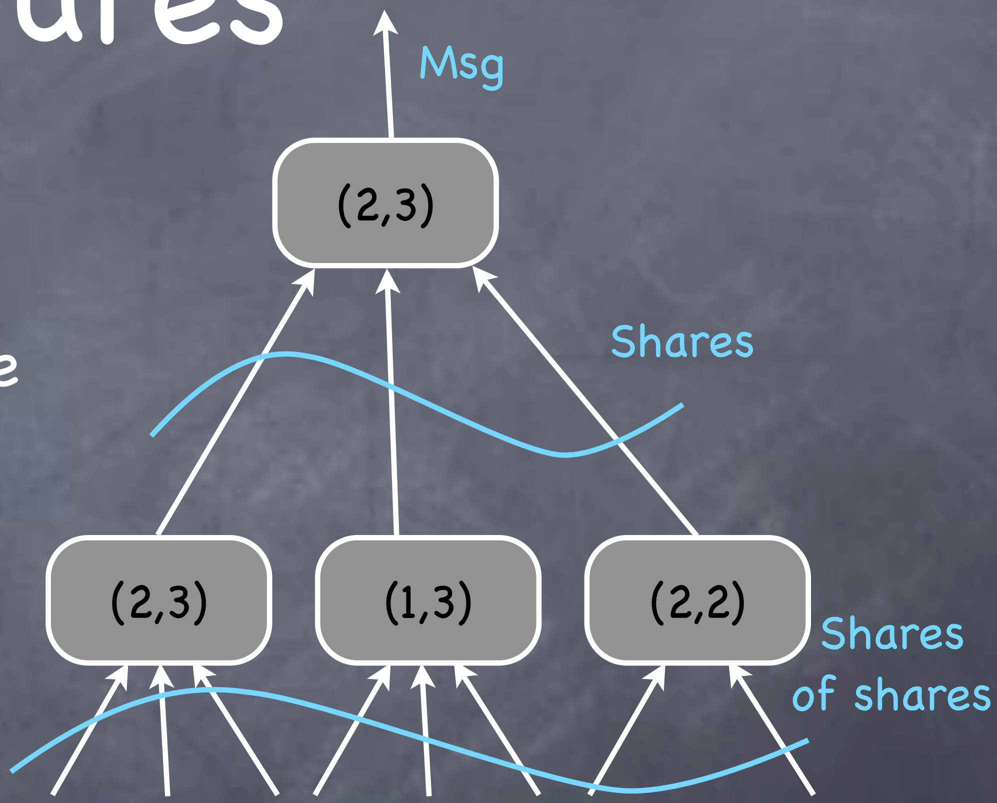
More General Access Structures

- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure
  - Can realize by recursively threshold secret-sharing the shares



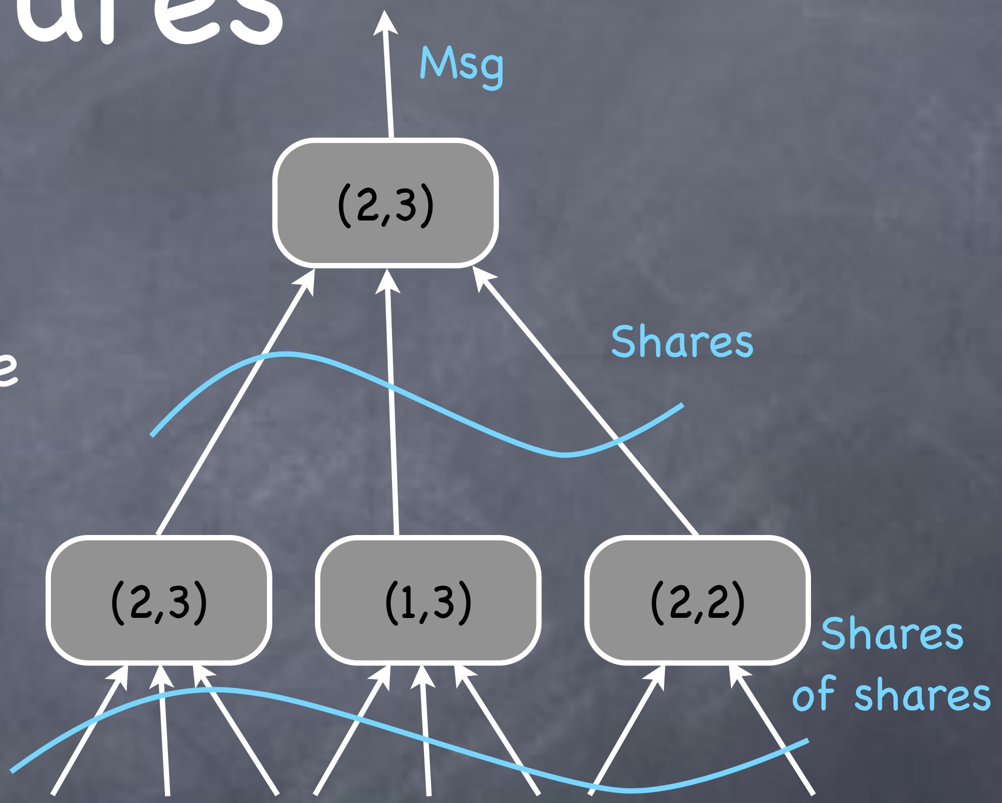
More General Access Structures

- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure
  - Can realize by recursively threshold secret-sharing the shares
- A special case of access structures that can be specified using "monotone span programs"



More General Access Structures

- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure
  - Can realize by recursively threshold secret-sharing the shares
- A special case of access structures that can be specified using "monotone span programs"
  - Admits linear secret-sharing



Linear Secret-Sharing

Linear Secret-Sharing

- Share(M): For some fixed $n \times t$ matrix W, let $\overline{\mathbf{s}} = W \cdot \overline{\mathbf{c}}$, where $c_0 = M$ and other t-1 coordinates are random.

Linear Secret-Sharing

- Share(M): For some fixed $n \times t$ matrix W, let $\overline{\mathbf{s}} = W \cdot \overline{\mathbf{c}}$, where $c_0 = M$ and other t-1 coordinates are random.
    - The shares are subsets of coordinates of $\overline{\mathbf{s}}$

Linear Secret-Sharing

- Share(M): For some fixed $n \times t$ matrix W, let $\overline{\mathbf{s}} = W \cdot \overline{\mathbf{c}}$, where $c_0 = M$ and other t-1 coordinates are random.
  - The shares are subsets of coordinates of $\overline{\mathbf{s}}$

Shamir Secret-Sharing is of this form

Linear Secret-Sharing

- Share(M): For some fixed $n \times t$ matrix W, let $\overline{\mathbf{s}} = W \cdot \overline{\mathbf{c}}$, where $c_0 = M$ and other t-1 coordinates are random.
  - The shares are subsets of coordinates of $\overline{\mathbf{s}}$

Shamir Secret-Sharing is of this form

- Reconstruction: pool together all the available coordinates of $\overline{\mathbf{s}}$; can reconstruct if there are enough equations to solve for c_0

Linear Secret-Sharing

- Share(M): For some fixed $n \times t$ matrix W, let $\bar{s} = W \cdot \bar{c}$, where $c_0 = M$ and other t-1 coordinates are random.
  - The shares are subsets of coordinates of $\bar{s}$

Shamir Secret-Sharing is of this form

- Reconstruction: pool together all the available coordinates of $\bar{s}$; can reconstruct if there are enough equations to solve for c_0
  - If not reconstructible, shares independent of secret

Linear Secret-Sharing

- Share(M): For some fixed $n \times t$ matrix W, let $\overline{\mathbf{s}} = W \cdot \overline{\mathbf{c}}$, where $c_0 = M$ and other t-1 coordinates are random.
  - The shares are subsets of coordinates of $\overline{\mathbf{s}}$

Shamir Secret-Sharing is of this form

- Reconstruction: pool together all the available coordinates of $\overline{\mathbf{s}}$; can reconstruct if there are enough equations to solve for c_0
  - If not reconstructible, shares independent of secret
- May not correspond to a threshold access structure

Linear Secret-Sharing

- Share(M): For some fixed $n \times t$ matrix W, let $\bar{s} = W \cdot \bar{c}$, where c_0 = M and other t-1 coordinates are random.
  - The shares are subsets of coordinates of $\bar{s}$

- Reconstruction: pool together all the available coordinates of $\bar{s}$; can reconstruct if there are enough equations to solve for c_0
  - If not reconstructible, shares independent of secret
- May not correspond to a threshold access structure
- Reconstruction too is a linear combination of available shares (coefficients depending on which subset of shares available)

Linear Secret-Sharing

Linear Secret-Sharing

- Linearity of linear secret-sharing:

Linear Secret-Sharing

- Linearity of linear secret-sharing:
  - If two secrets m_1, $m_2 \in \mathbb{F}$ have been shared and parties get shares $\{x_i\}$ and $\{y_i\}$ (also $\mathbb{F}$ elements) as shares, then each party can locally obtain sharing $\{z_i\}$ of am_1+bm_2

Linear Secret-Sharing

- Linearity of linear secret-sharing:
  - If two secrets $m_1, m_2 \in \mathbb{F}$ have been shared and parties get shares $\{x_i\}$ and $\{y_i\}$ (also $\mathbb{F}$ elements) as shares, then each party can locally obtain sharing $\{z_i\}$ of am_1+bm_2
    - $z_i = ax_i + by_i$

Linear Secret-Sharing

- Linearity of linear secret-sharing:
  - If two secrets $m_1, m_2 \in \mathbb{F}$ have been shared and parties get shares $\{x_i\}$ and $\{y_i\}$ (also $\mathbb{F}$ elements) as shares, then each party can locally obtain sharing $\{z_i\}$ of am_1+bm_2
    - $z_i = ax_i + by_i$

$\overline{\mathbf{x}} = W \cdot \overline{\mathbf{c}}_1$

$\overline{\mathbf{y}} = W \cdot \overline{\mathbf{c}}_2$

$\overline{\mathbf{z}} = W \cdot (a\overline{\mathbf{c}}_1 + b\overline{\mathbf{c}}_2)$

Linear Secret-Sharing

- Linearity of linear secret-sharing:
  - If two secrets $m_1, m_2 \in \mathbb{F}$ have been shared and parties get shares $\{x_i\}$ and $\{y_i\}$ (also $\mathbb{F}$ elements) as shares, then each party can locally obtain sharing $\{z_i\}$ of am_1+bm_2
    - $z_i = ax_i + by_i$
  - Useful in secure multiparty computation (later)

$$\overline{\mathbf{x}} = W \cdot \overline{\mathbf{c}}_1$$
$$\overline{\mathbf{y}} = W \cdot \overline{\mathbf{c}}_2$$
$$\overline{\mathbf{z}} = W \cdot (a\overline{\mathbf{c}}_1 + b\overline{\mathbf{c}}_2)$$

Linear Secret-Sharing

- Linearity of linear secret-sharing:
  - If two secrets $m_1, m_2 \in \mathbb{F}$ have been shared and parties get shares $\{x_i\}$ and $\{y_i\}$ (also $\mathbb{F}$ elements) as shares, then each party can locally obtain sharing $\{z_i\}$ of am_1+bm_2
    - $z_i = ax_i + by_i$
  - Useful in secure multiparty computation (later)

$$\overline{\mathbf{x}} = W \cdot \overline{\mathbf{c}}_1$$
$$\overline{\mathbf{y}} = W \cdot \overline{\mathbf{c}}_2$$
$$\overline{\mathbf{z}} = W \cdot (a\overline{\mathbf{c}}_1 + b\overline{\mathbf{c}}_2)$$

- Simple(st) example: from <u>additive</u> shares for two bits m_1 and m_2, n parties can locally obtain an additive sharing of $m_1 \oplus m_2$

Linear Secret-Sharing

- Linearity of linear secret-sharing:
  - If two secrets $m_1, m_2 \in \mathbb{F}$ have been shared and parties get shares $\{x_i\}$ and $\{y_i\}$ (also $\mathbb{F}$ elements) as shares, then each party can locally obtain sharing $\{z_i\}$ of am_1+bm_2
    - $z_i = ax_i + by_i$
  - Useful in secure multiparty computation (later)
- Simple(st) example: from additive shares for two bits m_1 and m_2, n parties can locally obtain an additive sharing of $m_1 \oplus m_2$
  - Gives a "private summation" protocol

$$\overline{\mathbf{x}} = W \cdot \overline{\mathbf{c}}_1$$
$$\overline{\mathbf{y}} = W \cdot \overline{\mathbf{c}}_2$$
$$\overline{\mathbf{z}} = W \cdot (a\overline{\mathbf{c}}_1 + b\overline{\mathbf{c}}_2)$$

Linear Secret-Sharing

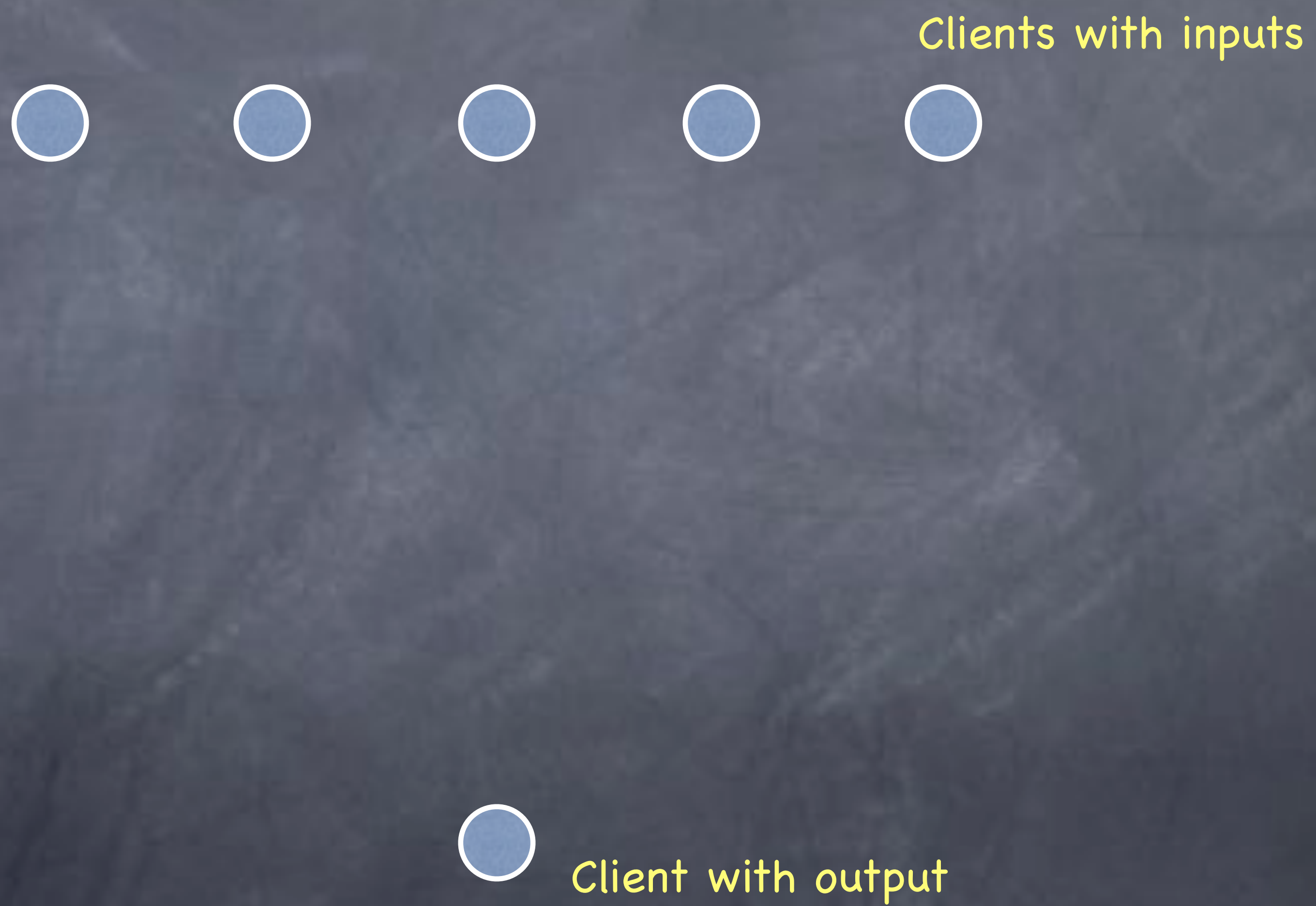
- Gives a “private summation” protocol

Linear Secret-Sharing

- Gives a "private summation" protocol

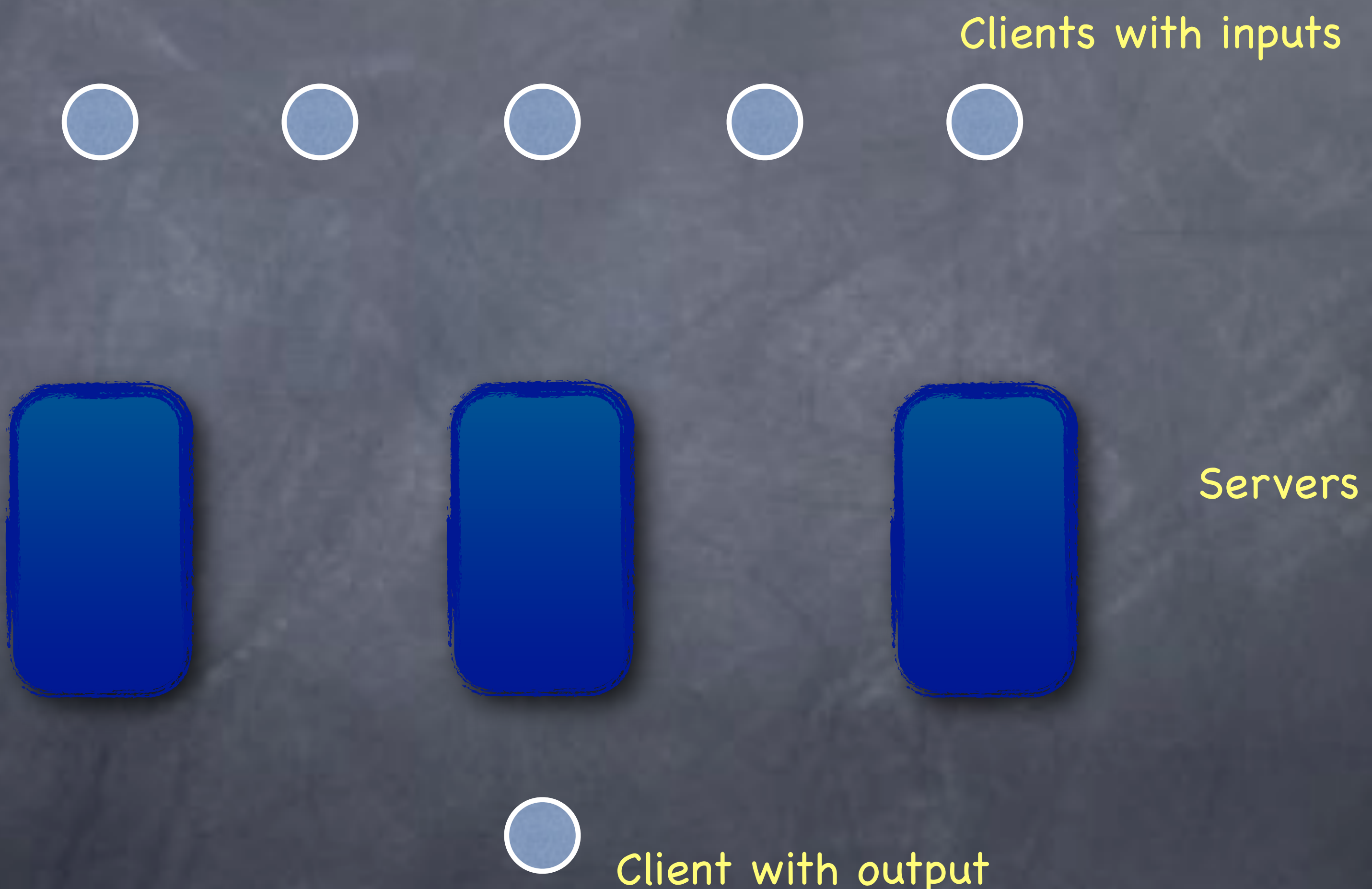
Linear Secret-Sharing

- Gives a "private summation" protocol



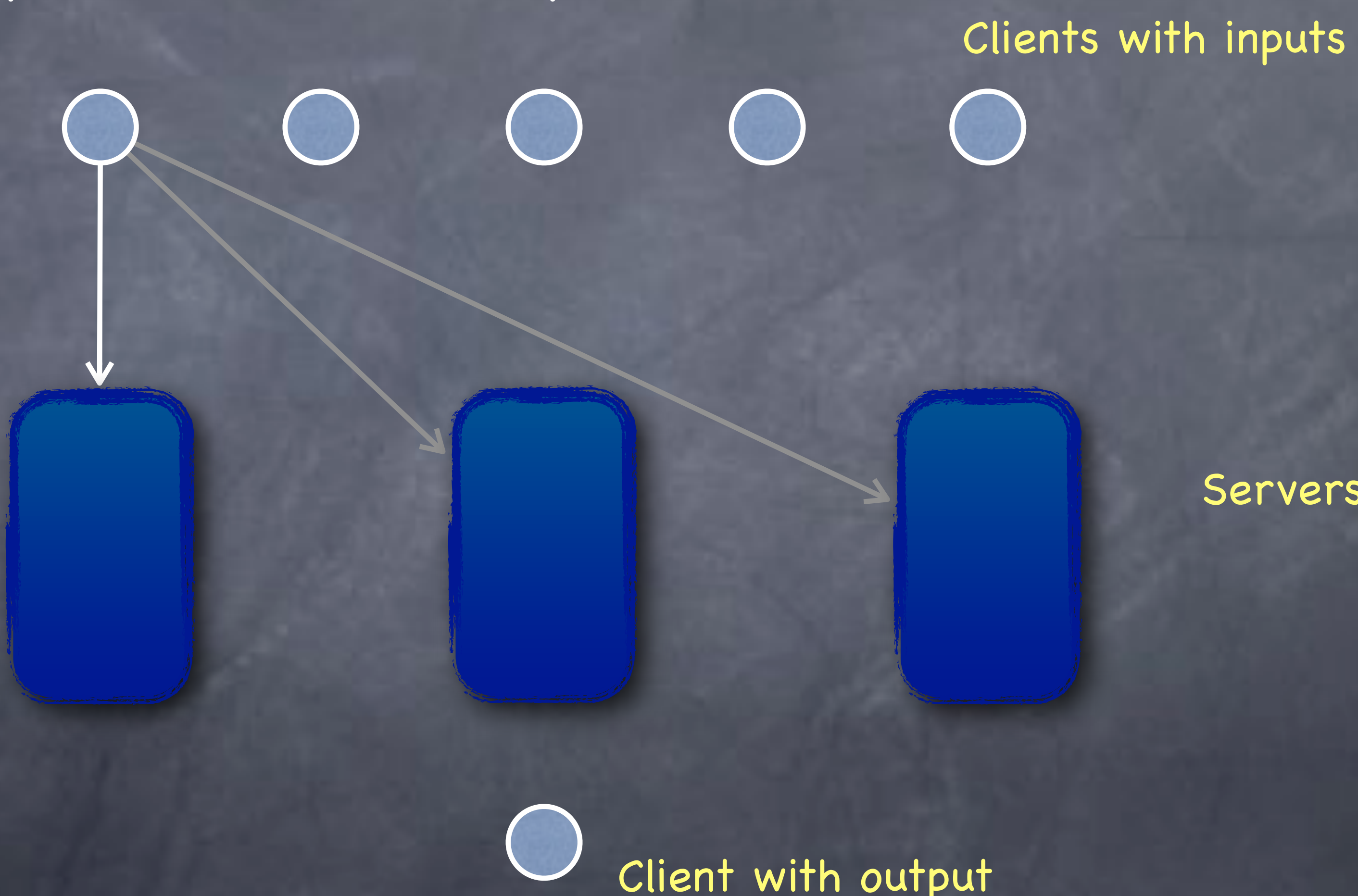
Linear Secret-Sharing

- Gives a "private summation" protocol



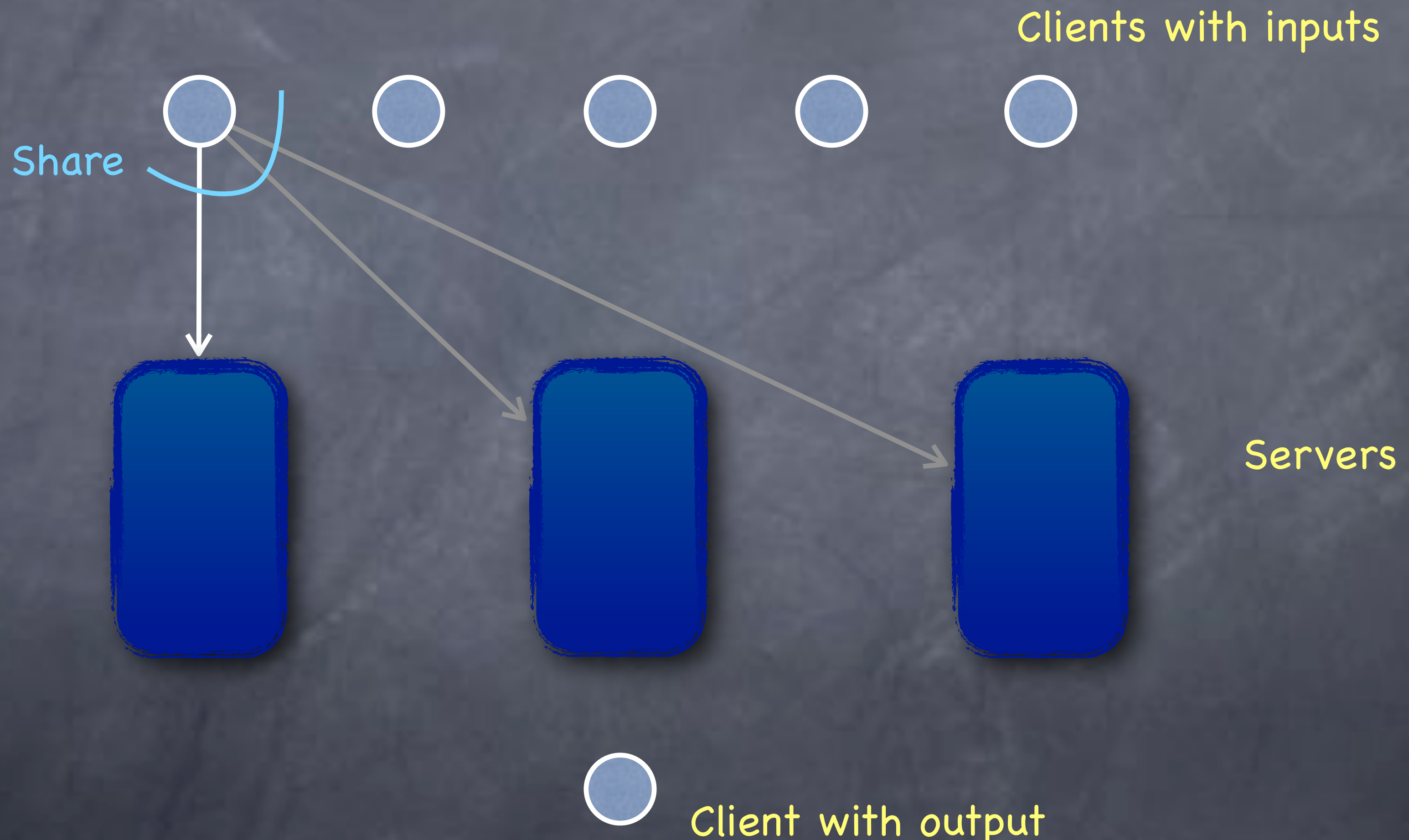
Linear Secret-Sharing

- Gives a “private summation” protocol



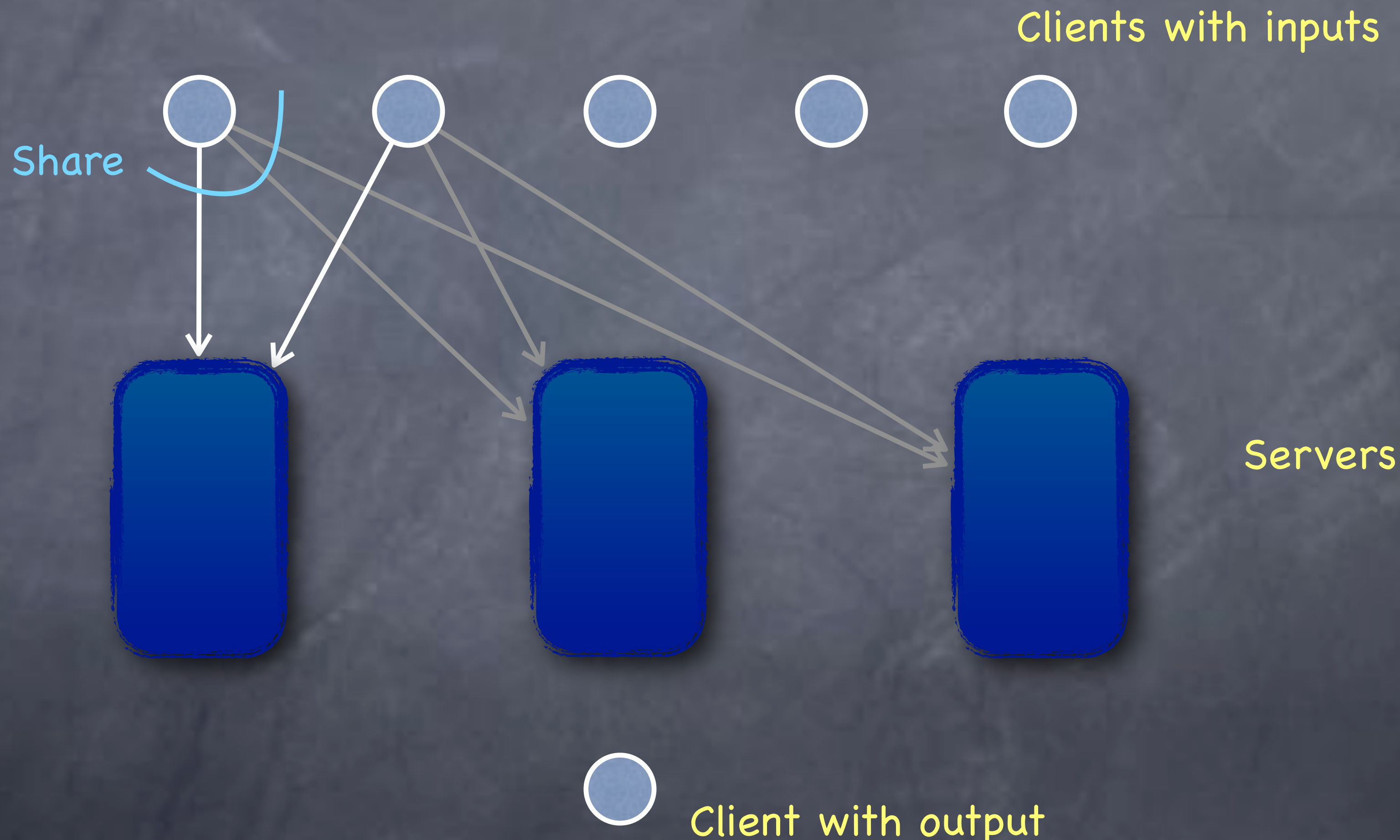
Linear Secret-Sharing

- Gives a "private summation" protocol



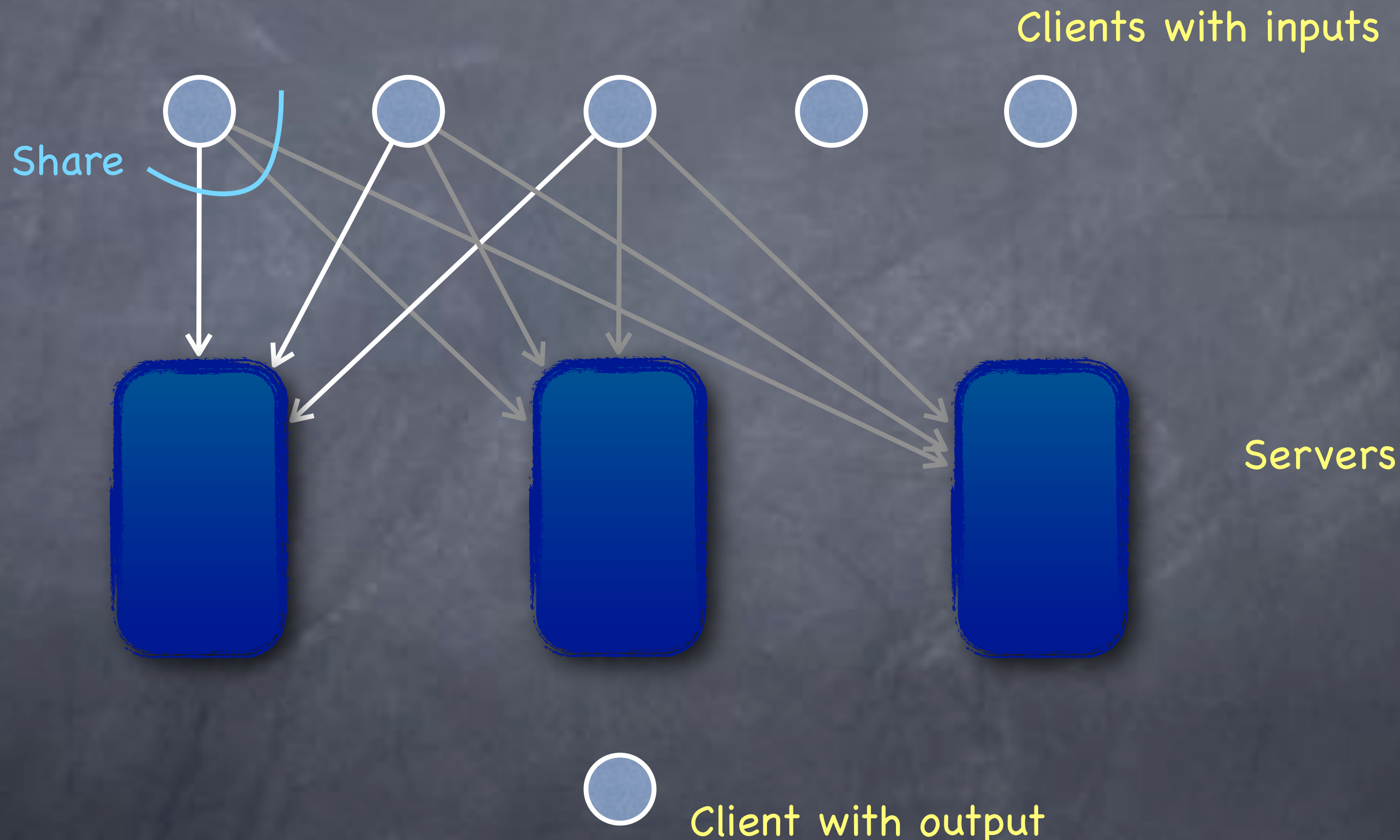
Linear Secret-Sharing

- Gives a "private summation" protocol



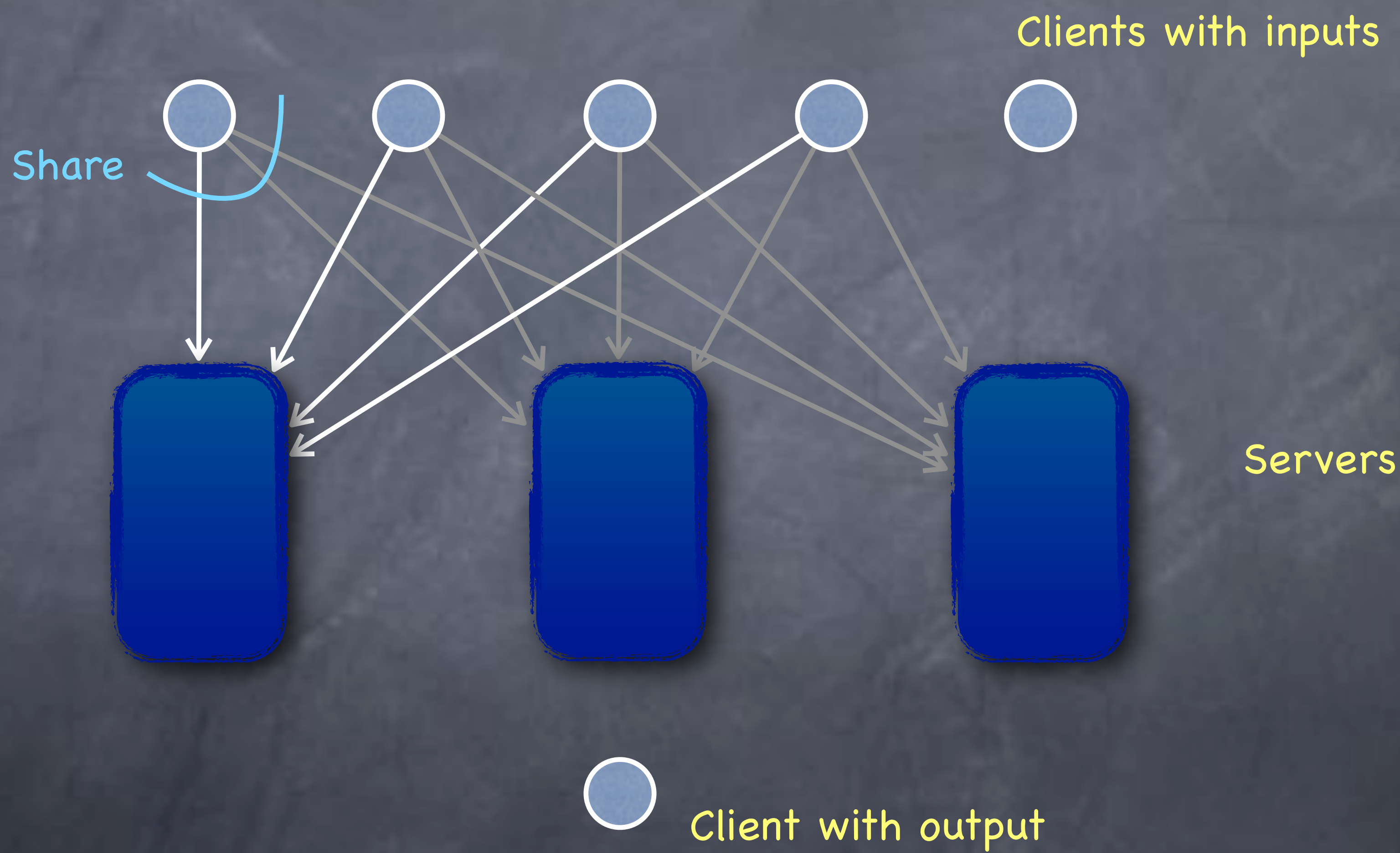
Linear Secret-Sharing

- Gives a "private summation" protocol



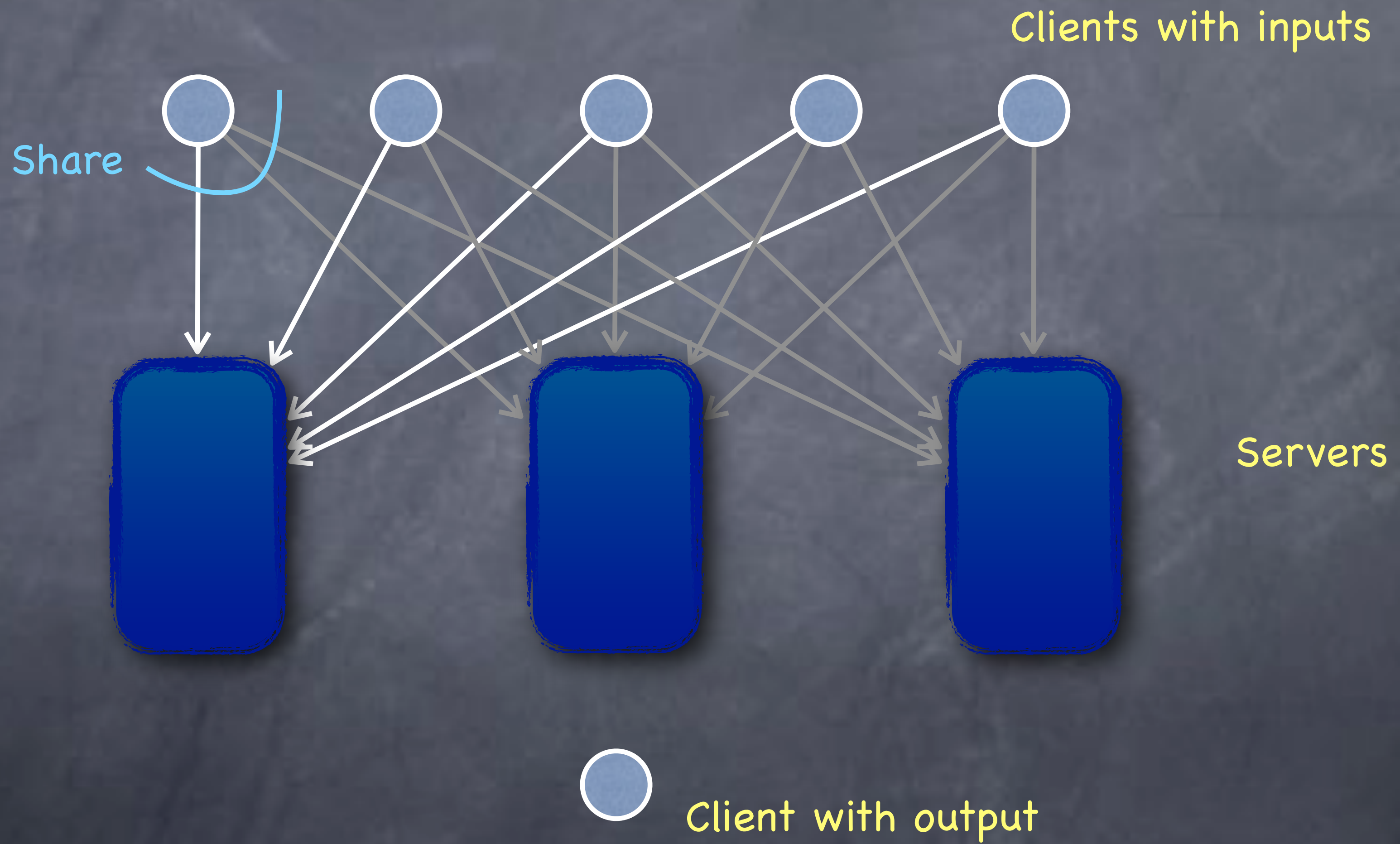
Linear Secret-Sharing

- Gives a "private summation" protocol



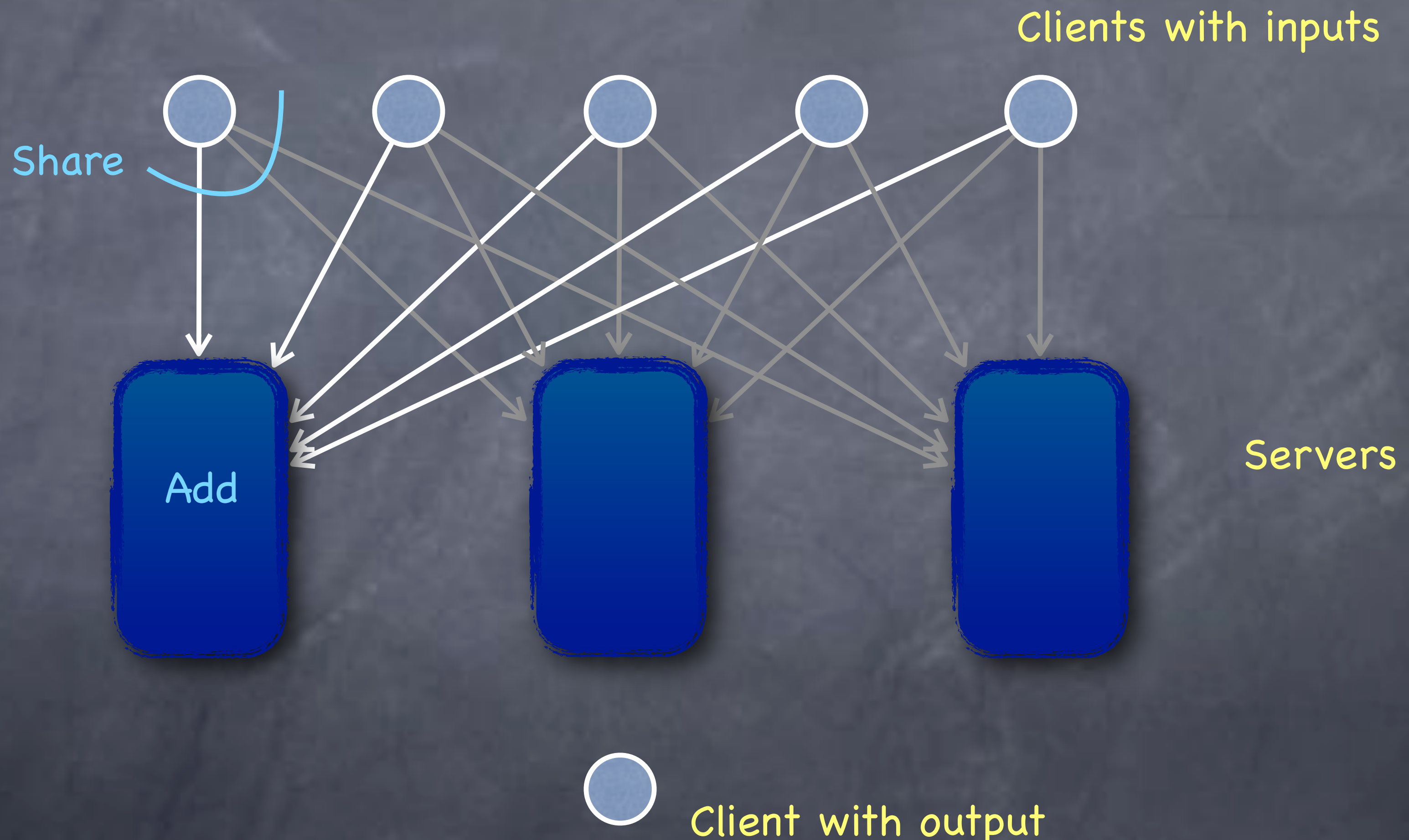
Linear Secret-Sharing

- Gives a "private summation" protocol



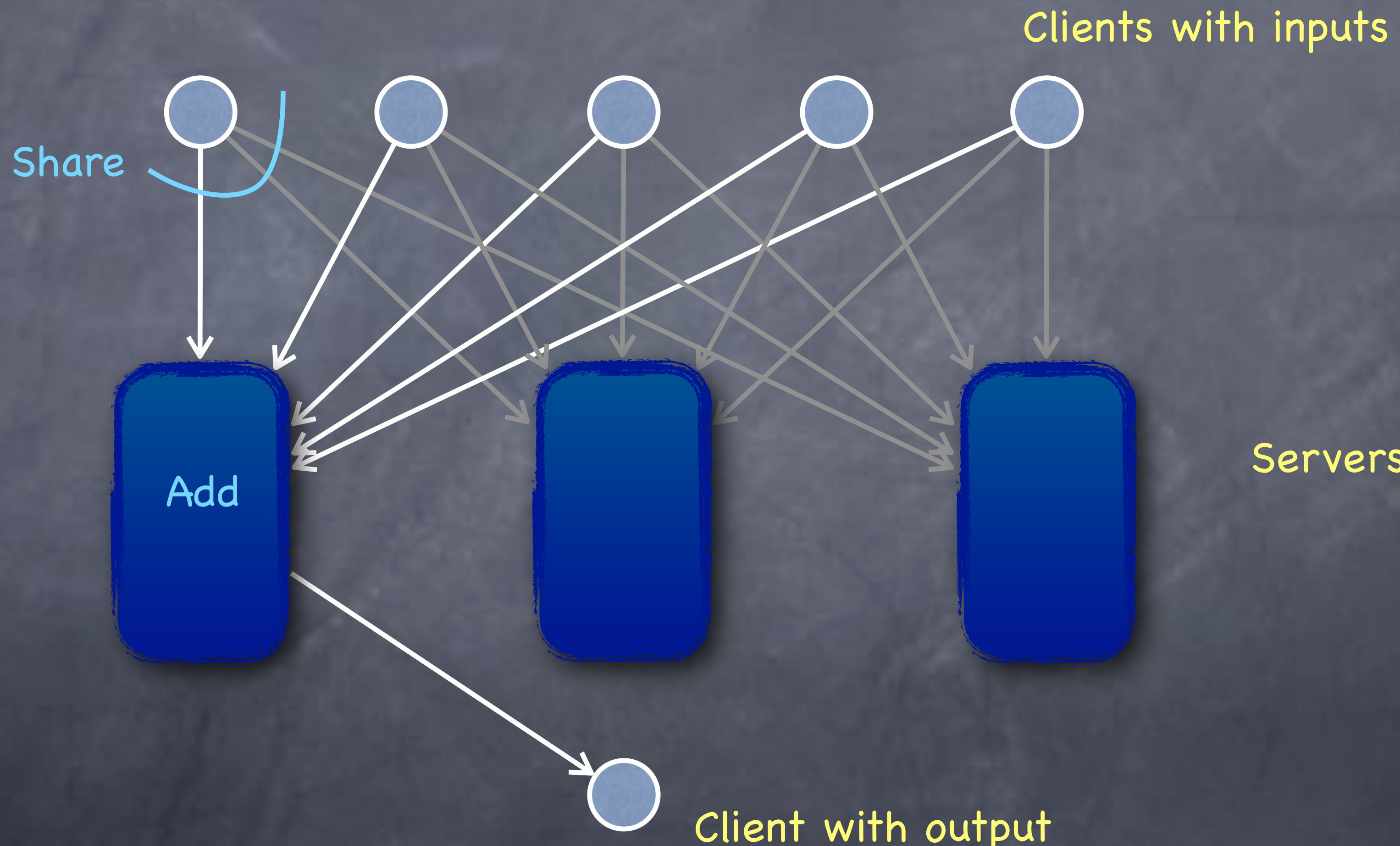
Linear Secret-Sharing

- Gives a "private summation" protocol



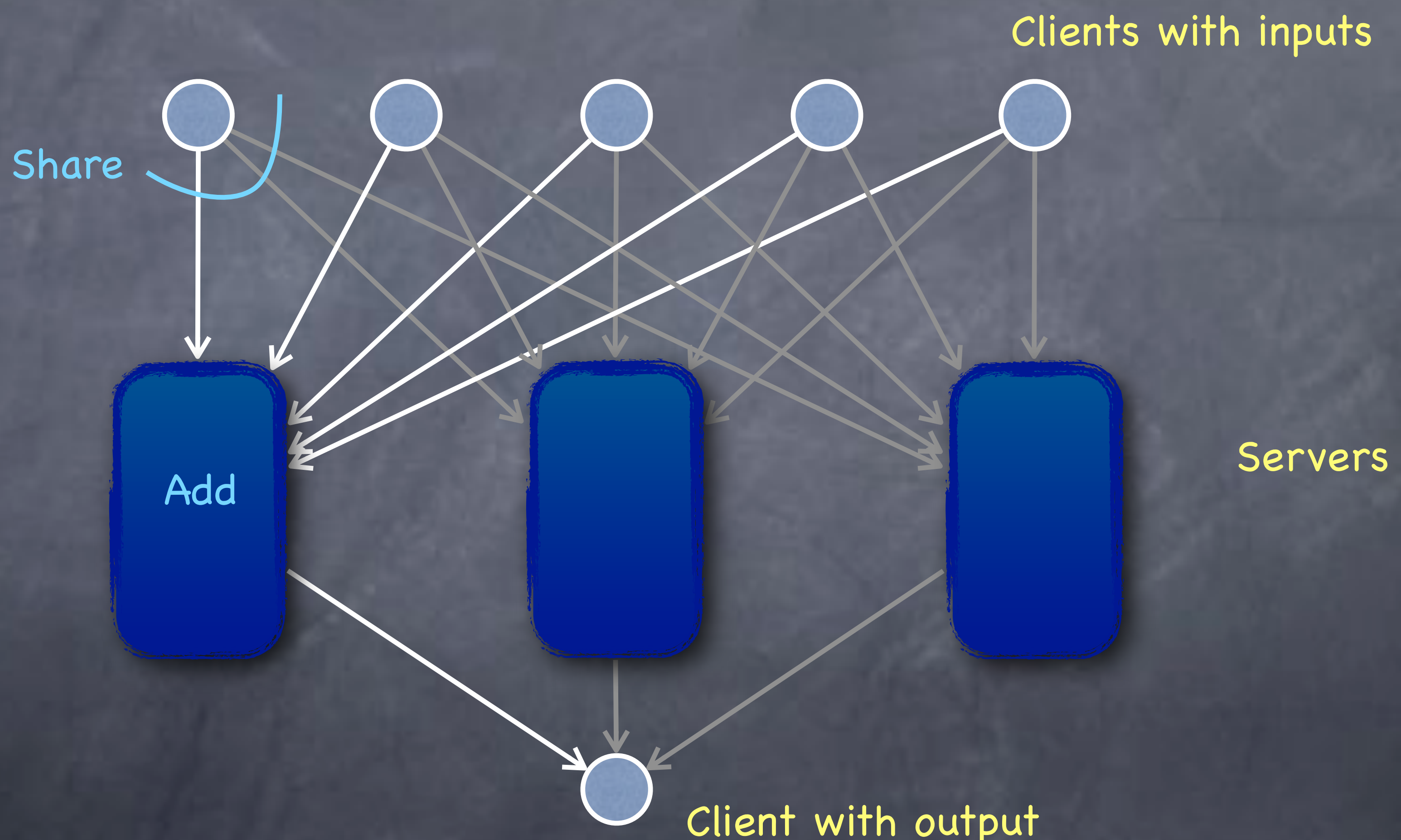
Linear Secret-Sharing

- Gives a "private summation" protocol



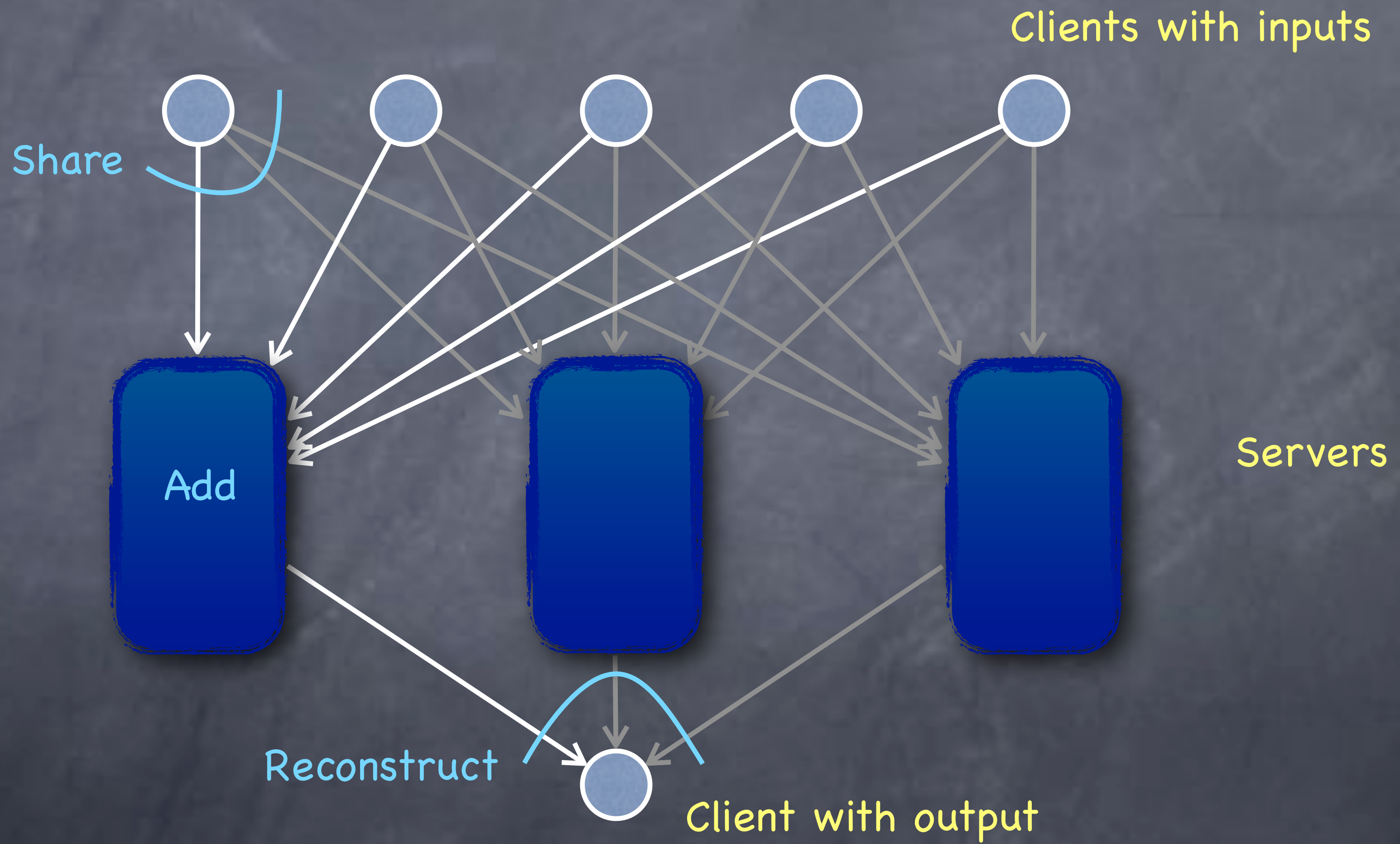
Linear Secret-Sharing

- Gives a "private summation" protocol



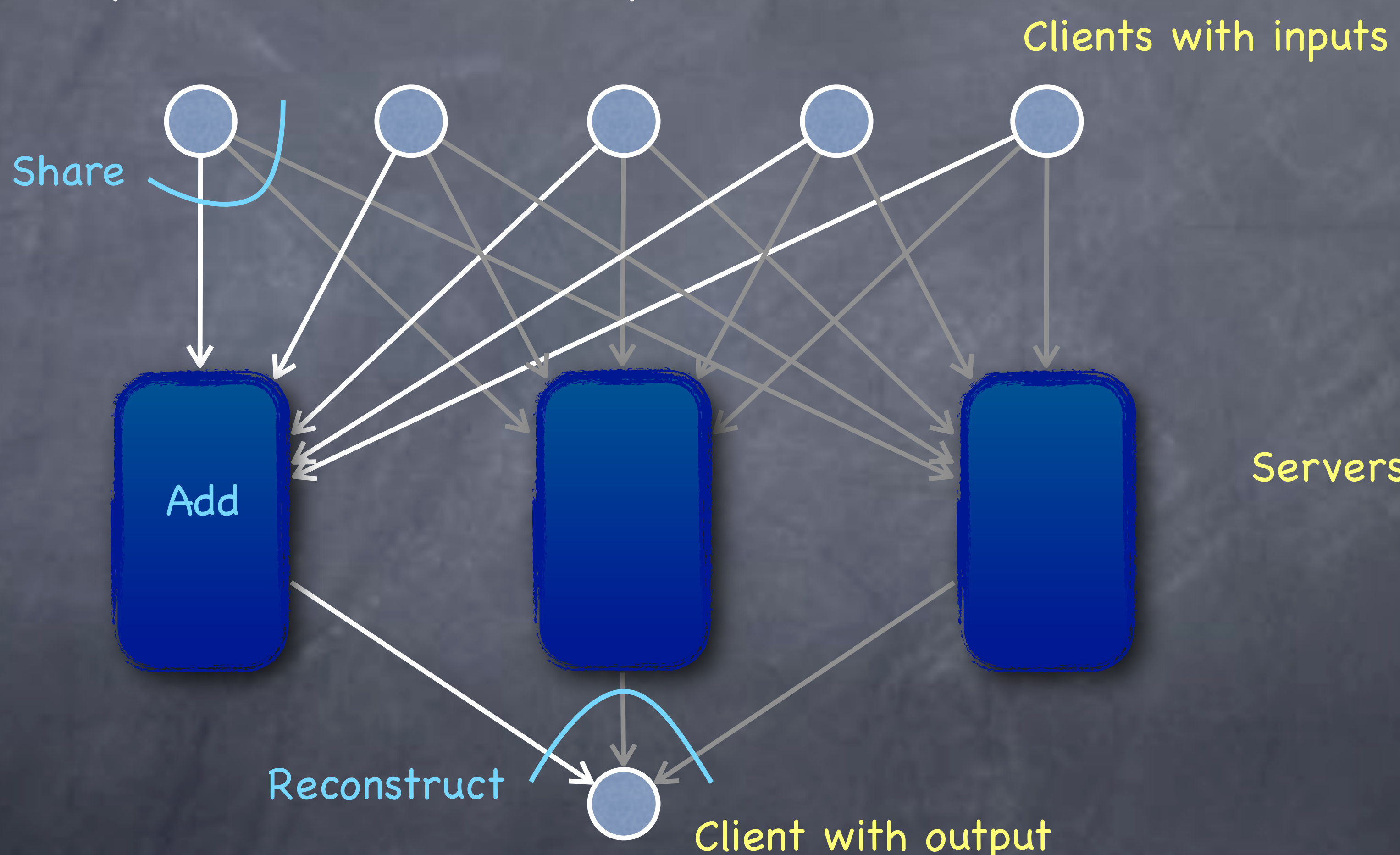
Linear Secret-Sharing

- Gives a "private summation" protocol



Linear Secret-Sharing

- Gives a "private summation" protocol



- Secure against passive corruption (no set of parties learn more than what they must) if at least one server is uncorrupted

Efficiency

Efficiency

- Main measure: size of the shares (say, total of all shares)

Efficiency

- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)

Efficiency

- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - Naive scheme for arbitrary monotonic access structure: if a party is in N sets in $\mathcal{B}$, N basic shares

Efficiency

- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - Naive scheme for arbitrary monotonic access structure: if a party is in N sets in $\mathcal{B}$, N basic shares
    - N can be exponential in n (as $\mathcal{B}$ can have exponentially many sets)

Efficiency

- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - Naive scheme for arbitrary monotonic access structure: if a party is in N sets in $\mathcal{B}$, N basic shares
    - N can be exponential in n (as $\mathcal{B}$ can have exponentially many sets)
  - Share size must be at least as big as the secret: "last share" in a minimal authorized set should contain all the information about the secret

Efficiency

- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - Naive scheme for arbitrary monotonic access structure: if a party is in N sets in $\mathcal{B}$, N basic shares
    - N can be exponential in n (as $\mathcal{B}$ can have exponentially many sets)
  - Share size must be at least as big as the secret: "last share" in a minimal authorized set should contain all the information about the secret
    - Ideal: if all shares are only this big (e.g. Shamir's scheme)

Efficiency

- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - Naive scheme for arbitrary monotonic access structure: if a party is in N sets in $\mathcal{B}$, N basic shares
    - N can be exponential in n (as $\mathcal{B}$ can have exponentially many sets)
  - Share size must be at least as big as the secret: "last share" in a minimal authorized set should contain all the information about the secret
    - <u>Ideal</u>: if all shares are only this big (e.g. Shamir's scheme)
    - Not all access structures have ideal schemes

Efficiency

- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - Naive scheme for arbitrary monotonic access structure: if a party is in N sets in $\mathcal{B}$, N basic shares
    - N can be exponential in n (as $\mathcal{B}$ can have exponentially many sets)
  - Share size must be at least as big as the secret: "last share" in a minimal authorized set should contain all the information about the secret
    - <u>Ideal</u>: if all shares are only this big (e.g. Shamir's scheme)
    - Not all access structures have ideal schemes
  - Non-linear schemes can be more efficient than linear schemes

Verifiable Secret-Sharing

Verifiable Secret-Sharing

- Guarding against possible malicious behavior by participants

Verifiable Secret-Sharing

- Guarding against possible malicious behavior by participants
  - Bad players: may substitute their shares to change the outcome (e.g., in additive sharing, can add to the outcome by adding to one's share)

Verifiable Secret-Sharing

- Guarding against possible malicious behavior by participants
  - Bad players: may substitute their shares to change the outcome (e.g., in additive sharing, can add to the outcome by adding to one's share)
  - Bad dealer (plus some bad players): may distribute shares which do not have a consistent secret (e.g., in Shamir's, if dealer uses a higher degree polynomial); if participating in reconstruction, may be able to fix the secret at that time, or, even if enough good players get together, deny them ability to reconstruct

Verifiable Secret-Sharing

- Guarding against possible malicious behavior by participants
  - Bad players: may substitute their shares to change the outcome (e.g., in additive sharing, can add to the outcome by adding to one's share)
  - Bad dealer (plus some bad players): may distribute shares which do not have a consistent secret (e.g., in Shamir's, if dealer uses a higher degree polynomial); if participating in reconstruction, may be able to fix the secret at that time, or, even if enough good players get together, deny them ability to reconstruct
- Privacy: if dealer is honest, adversary (who does not control an authorized set) learns nothing of the secret

Verifiable Secret-Sharing

- Guarding against possible malicious behavior by participants
  - Bad players: may substitute their shares to change the outcome (e.g., in additive sharing, can add to the outcome by adding to one's share)
  - Bad dealer (plus some bad players): may distribute shares which do not have a consistent secret (e.g., in Shamir's, if dealer uses a higher degree polynomial); if participating in reconstruction, may be able to fix the secret at that time, or, even if enough good players get together, deny them ability to reconstruct
- Privacy: if dealer is honest, adversary (who does not control an authorized set) learns nothing of the secret
- Correctness: if dealer honest, reconstruction correct; even if dealer corrupt, a fixed consistent secret at the end of sharing

Verifiable Secret-Sharing

Verifiable Secret-Sharing

- Access structure and "Adversary Structure"

Verifiable Secret-Sharing

- Access structure and "Adversary Structure"
  - Latter saying who all can be malicious

Verifiable Secret-Sharing

- Access structure and "Adversary Structure"
  - Latter saying who all can be malicious
  - VSS not possible unless some restrictions on the adversary structure (e.g., at most a minority of the parties can be corrupted)

Verifiable Secret-Sharing

- Access structure and "Adversary Structure"
  - Latter saying who all can be malicious
  - VSS not possible unless some restrictions on the adversary structure (e.g., at most a minority of the parties can be corrupted)
- Typically require that for admissible adversary structures, if dealer honest, honest players in an authorized set will reconstruct the secret (even if malicious players in the set try to sabotage)

Verifiable Secret-Sharing

- Access structure and "Adversary Structure"
  - Latter saying who all can be malicious
  - VSS not possible unless some restrictions on the adversary structure (e.g., at most a minority of the parties can be corrupted)
- Typically require that for admissible adversary structures, if dealer honest, honest players in an authorized set will reconstruct the secret (even if malicious players in the set try to sabotage)
- A broadcast channel is very useful (to force each player to tell everyone the same story)

Verifiable Secret-Sharing

- Access structure and "Adversary Structure"
  - Latter saying who all can be malicious
  - VSS not possible unless some restrictions on the adversary structure (e.g., at most a minority of the parties can be corrupted)
- Typically require that for admissible adversary structures, if dealer honest, honest players in an authorized set will reconstruct the secret (even if malicious players in the set try to sabotage)
- A broadcast channel is very useful (to force each player to tell everyone the same story)
  - Broadcast can be achieved on top of point-to-point channels if only a small fraction (<1/3) corrupted

Verifiable Secret-Sharing

- Access structure and "Adversary Structure"
  - Latter saying who all can be malicious
  - VSS not possible unless some restrictions on the adversary structure (e.g., at most a minority of the parties can be corrupted)
- Typically require that for admissible adversary structures, if dealer honest, honest players in an authorized set will reconstruct the secret (even if malicious players in the set try to sabotage)
- A broadcast channel is very useful (to force each player to tell everyone the same story)
  - Broadcast can be achieved on top of point-to-point channels if only a small fraction (<1/3) corrupted
    - Otherwise malicious players can cause denial-of-service

Today

Today

- Secrecy: if view is independent of the message

Today

- Secrecy: if view is independent of the message
  - Does not give unprivileged sets of parties any additional information about the message, than what they already had

Today

- Secrecy: if view is independent of the message
    - Does not give unprivileged sets of parties any additional information about the message, than what they already had
    - Irrespective of their computational power

Today

- Secrecy: if view is independent of the message
  - Does not give unprivileged sets of parties any additional information about the message, than what they already had
  - Irrespective of their computational power
- Such secrecy not always possible (e.g., no public-key encryption)

Today

- Secrecy: if view is independent of the message
  - Does not give unprivileged sets of parties any <u>additional</u> information about the message, than what they already had
  - Irrespective of their computational power
- Such secrecy not always possible (e.g., no public-key encryption)
- Next: secrecy against computationally bounded players