On the Optimization of Secret Sharing Schemes for General Access Structures

Carles Padró

Universitat Politècnica de Catalunya

Applications of Matroid Theory and Combinatorial Optimization to Information and Coding Theory
BIRS, August 2009

The Simplest Way to Share a Secret

How to share a secret value $s \in G$ (a finite group)
among a set of n players

Take random elements $s_1, \ldots, s_n \in G$ with

$$s = s_1 + \cdots + s_n$$

and give the value s_i to the i-th player.

The full set of n players can reconstruct
the secret value s from their shares

Any $n - 1$ players get no information about the value of s

Secure Multiparty Computation

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Secure Multiparty Computation

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

$$\begin{array}{c|cccc|c}
 & P_1 & P_2 & \cdots & P_n & \\
\hline
P_1 & & & & & x_1 \\
P_2 & & & & & x_2 \\
\vdots & & & & & \vdots \\
P_n & & & & & x_n \\
\hline
 & & & & & \\
\end{array}$$

Secure Multiparty Computation

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Share

	P_1	P_2	$\cdots$	P_n	
P_1	s_{11}	s_{12}	$\cdots$	s_{1n}	x_1
P_2					x_2
$\vdots$					$\vdots$
P_n					x_n

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Share

	P_1	P_2	$\cdots$	P_n	
P_1	s_{11}	s_{12}	$\cdots$	s_{1n}	x_1
P_2	s_{21}	s_{22}	$\cdots$	s_{2n}	x_2
$\vdots$					$\vdots$
P_n					x_n

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Share

	P_1	P_2	$\cdots$	P_n	
P_1	s_{11}	s_{12}	$\cdots$	s_{1n}	x_1
P_2	s_{21}	s_{22}	$\cdots$	s_{2n}	x_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
P_n	s_{n1}	s_{n2}	$\cdots$	s_{nn}	x_n

Secure Multiparty Computation

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Compute

$$
\begin{array}{c|cccc|c}
 & P_1 & P_2 & \cdots & P_n & \\
\hline
P_1 & s_{11} & s_{12} & \cdots & s_{1n} & x_1 \\
P_2 & s_{21} & s_{22} & \cdots & s_{2n} & x_2 \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
P_n & s_{n1} & s_{n2} & \cdots & s_{nn} & x_n \\
\hline
 & y_1 & & & &
\end{array}
$$

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Compute

	P_1	P_2	$\cdots$	P_n	
P_1	s_{11}	s_{12}	$\cdots$	s_{1n}	x_1
P_2	s_{21}	s_{22}	$\cdots$	s_{2n}	x_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
P_n	s_{n1}	s_{n2}	$\cdots$	s_{nn}	x_n
	y_1	y_2			

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Compute

	P_1	P_2	$\cdots$	P_n	
P_1	s_{11}	s_{12}	$\cdots$	s_{1n}	x_1
P_2	s_{21}	s_{22}	$\cdots$	s_{2n}	x_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
P_n	s_{n1}	s_{n2}	$\cdots$	s_{nn}	x_n
	y_1	y_2	$\cdots$	y_n	

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \dots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Reconstruct

$$\begin{array}{c|cccc|c} & P_1 & P_2 & \cdots & P_n & \\ \hline P_1 & s_{11} & s_{12} & \cdots & s_{1n} & x_1 \\ P_2 & s_{21} & s_{22} & \cdots & s_{2n} & x_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ P_n & s_{n1} & s_{n2} & \cdots & s_{nn} & x_n \\ \hline & y_1 & y_2 & \cdots & y_n & \mathbf{S} \end{array}$$

Secure multiparty computation: Some players want to compute an agreed function of their private inputs

A toy example: n players compute $F(x_1, \ldots, x_n) = x_1 + \cdots + x_n$
They proceed in three steps: share, compute, and reconstruct

Reconstruct

	P_1	P_2	$\cdots$	P_n	
P_1	s_{11}	s_{12}	$\cdots$	s_{1n}	x_1
P_2	s_{21}	s_{22}	$\cdots$	s_{2n}	x_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
P_n	s_{n1}	s_{n2}	$\cdots$	s_{nn}	x_n
	y_1	y_2	$\cdots$	y_n	**S**

Of course, we want to compute any function in a more secure way

How to Share a Secret

How to share a secret is such a way that $t \leq n$ players can reconstruct it but $t - 1$ players get no information?

How to Share a Secret

How to share a secret is such a way that $t \leq n$ players can reconstruct it but $t-1$ players get no information?

A simple and brilliant idea by Shamir, 1979

Let $\mathbb{K}$ be a finite field with $|\mathbb{K}| \geq n+1$

To share a secret value $k \in \mathbb{K}$, take a random polynomial

$$f(x) = k + a_1 x + \cdots + a_{t-1} x^{t-1} \in \mathbb{K}[x]$$

and distribute the shares

$$f(x_1),\ f(x_2),\ \ldots,\ f(x_n)$$

where $x_i \in \mathbb{K} - \{0\}$ is a public value associated to player p_i

Independently, Blakley proposed in 1979
a geometric secret sharing scheme

Properties of Shamir's Secret Sharing Scheme

1. It is a threshold scheme
2. It is perfect
3. It is ideal
4. It is linear
5. It is multiplicative

Properties of Shamir's Secret Sharing Scheme

1. It is a threshold scheme
   Every set of t players can reconstruct the secret value $k = f(0)$
   from their shares $f(x_1), \dots, f(x_t)$
   by using Lagrange interpolation
2. It is perfect
3. It is ideal
4. It is linear
5. It is multiplicative

Properties of Shamir's Secret Sharing Scheme

1. It is a threshold scheme
   Every set of t players can reconstruct the secret value $k = f(0)$
   from their shares $f(x_1), \ldots, f(x_t)$
   by using Lagrange interpolation
2. It is perfect
   The shares of any $t - 1$ players contain no information
   about the value of the secret
3. It is ideal
4. It is linear
5. It is multiplicative

Properties of Shamir's Secret Sharing Scheme

1. It is a threshold scheme
2. It is perfect
3. It is ideal
   Every share has the same length as the secret:
   all are elements in a finite field
   This is the best possible situation
4. It is linear
5. It is multiplicative

Properties of Shamir's Secret Sharing Scheme

1. It is a threshold scheme
2. It is perfect
3. It is ideal
4. It is linear
   Shares are a linear function of the secret and random values.
   The secret can be recovered by a linear function of the shares.
   Shares for a linear combination of two secrets
   can be obtained from the linear combination of the shares

   $$\lambda_1 k_1 + \lambda_2 k_2 = (\lambda_1 f_1 + \lambda_2 f_2)(0) \qquad \lambda_1 s_{1i} + \lambda_2 s_{2i} = (\lambda_1 f_1 + \lambda_2 f_2)(x_i)$$

5. It is multiplicative

Properties of Shamir's Secret Sharing Scheme

1. It is a threshold scheme
2. It is perfect
3. It is ideal
4. It is linear
5. It is multiplicative
   If $n \geq 2t - 1$, shares for the product of two secrets can be obtained from the products of the shares

$$k_1 k_2 = f_1 f_2(0) \qquad s_{1i} s_{2i} = f_1 f_2(x_i)$$

Properties of Shamir's Secret Sharing Scheme

1. It is a threshold scheme
2. It is perfect
3. It is ideal
4. It is linear
5. It is multiplicative

To which extent these properties can be generalized to secret sharing schemes with other access structures?

The access structure Γ is the family of qualified subsets

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure?

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

- From now on, we deal only with perfect schemes

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

- From now on, we deal only with perfect schemes

Does there exist a linear SSS for every access structure?

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

- From now on, we deal only with perfect schemes

Does there exist a linear SSS for every access structure? YES

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

- From now on, we deal only with perfect schemes

Does there exist a linear SSS for every access structure? YES

Does there exist an ideal SSS for every access structure?

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

- From now on, we deal only with perfect schemes

Does there exist a linear SSS for every access structure? YES

Does there exist an ideal SSS for every access structure? NO

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

- From now on, we deal only with perfect schemes

Does there exist a linear SSS for every access structure? YES

Does there exist an ideal SSS for every access structure? NO

Problem

What access structures admit an ideal secret sharing scheme?

Existential Questions & Optimization Problems

Does there exist a perfect SSS for every access structure? YES

- From now on, we deal only with perfect schemes

Does there exist a linear SSS for every access structure? YES

Does there exist an ideal SSS for every access structure? NO

Problem

What access structures admit an ideal secret sharing scheme?

Problem

Find the most efficient (linear) secret sharing scheme for every access structure

Brickell's Ideal Secret Sharing Scheme

The geometric schemes by Blakley (1979) were transformed by Brickell (1989) into a linear construction

Every linear code defines a vector space secret sharing scheme

$$(x_1, \ldots, x_d) \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \pi_0 & \pi_1 & \cdots & \pi_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = (k, s_1, \ldots, s_n)$$

It is perfect, ideal, and linear,
and it can have non-threshold access structure

Brickell's Ideal Secret Sharing Scheme

The geometric schemes by Blakley (1979) were transformed by Brickell (1989) into a linear construction

Every linear code defines a vector space secret sharing scheme

$$(x_1, \ldots, x_d) \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \pi_0 & \pi_1 & \cdots & \pi_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = (k, s_1, \ldots, s_n)$$

It is perfect, ideal, and linear,
and it can have non-threshold access structure

$A \in \Gamma$ if and only if $\mathrm{rank}(\pi_0, (\pi_i)_{i \in A}) = \mathrm{rank}((\pi_i)_{i \in A})$

$k = \pi_0(x) = \sum_{i \in A} \lambda_{i,A} \pi_i(x) = \sum_{i \in A} \lambda_{i,A} s_i$

Ideal Access Structures: a Sufficient Condition

$$(x_1, \ldots, x_d) \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \pi_0 & \pi_1 & \cdots & \pi_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = (k, s_1, \ldots, s_n)$$

$P = \{p_1, \ldots, p_n\}$, $Q = P \cup \{p_0\}$

Ideal Access Structures: a Sufficient Condition

$$(x_1, \ldots, x_d) \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \pi_0 & \pi_1 & \cdots & \pi_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = (k, s_1, \ldots, s_n)$$

$P = \{p_1, \ldots, p_n\}$, $Q = P \cup \{p_0\}$

If $\mathcal{M} = (Q, r)$ is the representable matroid associated to the code,

$$\Gamma = \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P \, : \, r(A \cup \{p_0\}) = r(A)\}$$

Equivalently,

$$\min \Gamma = \min \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P \, : \, A \cup \{p_0\} \text{ is a circuit of } \mathcal{M}\}$$

Ideal Access Structures: a Sufficient Condition

$$(x_1, \ldots, x_d) \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \pi_0 & \pi_1 & \cdots & \pi_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = (k, s_1, \ldots, s_n)$$

$P = \{p_1, \ldots, p_n\}$, $Q = P \cup \{p_0\}$

If $\mathcal{M} = (Q, r)$ is the representable matroid associated to the code,

$$\Gamma = \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P : r(A \cup \{p_0\}) = r(A)\}$$

Equivalently,

$$\min \Gamma = \min \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P : A \cup \{p_0\} \text{ is a circuit of } \mathcal{M}\}$$

That is, Γ is the port of the matroid $\mathcal{M}$ at the point p_0

Matroid ports were introduced by Lehman 1976 to solve the Shannon switching game

Ideal Access Structures: a Sufficient Condition

$$(x_1, \ldots, x_d) \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \pi_0 & \pi_1 & \cdots & \pi_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = (k, s_1, \ldots, s_n)$$

$P = \{p_1, \ldots, p_n\}$, $Q = P \cup \{p_0\}$

If $\mathcal{M} = (Q, r)$ is the representable matroid associated to the code,

$$\Gamma = \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P \,:\, r(A \cup \{p_0\}) = r(A)\}$$

Equivalently,

$$\min \Gamma = \min \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P \,:\, A \cup \{p_0\} \text{ is a circuit of } \mathcal{M}\}$$

That is, Γ is the port of the matroid $\mathcal{M}$ at the point p_0

Matroid ports were introduced by Lehman 1976 to solve the Shannon switching game

Theorem

If Γ is the port of a representable matroid, then Γ is ideal

General Secret Sharing

A secret sharing scheme on the set $P = \{p_1, \ldots, p_n\}$ of participants is a mapping

$$\begin{aligned} \Pi\colon\ E &\to E_0 \times E_1 \times \cdots \times E_n \\ x &\mapsto (\pi_0(x)|\pi_1(x), \ldots, \pi_n(x)) \end{aligned}$$

together with a probability distribution on E

A secret sharing scheme is a collection of random variables

- $\pi_0(x) \in E_0$ is the secret value
- $\pi_i(x) \in E_i$ is the share for the player p_i

General Secret Sharing

A secret sharing scheme on the set $P = \{p_1, \ldots, p_n\}$ of participants is a mapping

$$\begin{aligned} \Pi\colon\ E &\rightarrow E_0 \times E_1 \times \cdots \times E_n \\ x &\mapsto (\pi_0(x) | \pi_1(x), \ldots, \pi_n(x)) \end{aligned}$$

together with a probability distribution on E

A secret sharing scheme is a collection of random variables such that

- If $A \subseteq P$ is qualified, $H(E_0 | E_A) = H(E_0 | (E_i)_{p_i \in A}) = 0$
- Otherwise, $H(E_0 | E_A) = H(E_0)$

General Secret Sharing

A secret sharing scheme on the set $P = \{p_1, \dots, p_n\}$ of participants is a mapping

$$\begin{aligned} \Pi\colon E &\to E_0 \times E_1 \times \cdots \times E_n \\ x &\mapsto (\pi_0(x) | \pi_1(x), \dots, \pi_n(x)) \end{aligned}$$

together with a probability distribution on E

A secret sharing scheme is a collection of random variables such that

- If $A \subseteq P$ is qualified, $H(E_0|E_A) = H(E_0|(E_i)_{p_i \in A}) = 0$
- Otherwise, $H(E_0|E_A) = H(E_0)$

The qualified subsets form the access structure Γ of the scheme

If p_i is a non-redundant player, then $H(E_i) \geq H(E_0)$

General Secret Sharing

A secret sharing scheme on the set $P = \{p_1, \ldots, p_n\}$ of participants is a mapping

$$\begin{aligned} \Pi\colon\ E &\to E_0 \times E_1 \times \cdots \times E_n \\ x &\mapsto (\pi_0(x) | \pi_1(x), \ldots, \pi_n(x)) \end{aligned}$$

together with a probability distribution on E

A secret sharing scheme is a collection of random variables such that

- If $A \subseteq P$ is qualified, $H(E_0|E_A) = H(E_0|(E_i)_{p_i \in A}) = 0$
- Otherwise, $H(E_0|E_A) = H(E_0)$

The qualified subsets form the access structure Γ of the scheme

If p_i is a non-redundant player, then $H(E_i) \geq H(E_0)$

There exists a secret sharing scheme for every access structure, but in general the shares are much larger than the secret

Secret Sharing and Polymatroids

Consider as before $P = \{p_1, \ldots, p_n\}$ and $Q = P \cup \{p_0\}$

For an arbitrary secret sharing scheme consider,
for every $A \subseteq Q$

$$h(A) = \frac{H(E_A)}{H(E_0)}$$

Secret Sharing and Polymatroids

Consider as before $P = \{p_1, \ldots, p_n\}$ and $Q = P \cup \{p_0\}$

For an arbitrary secret sharing scheme consider,
for every $A \subseteq Q$

$$h(A) = \frac{H(E_A)}{H(E_0)}$$

Then

1. $h(\emptyset) = 0$
2. $X \subseteq Y \subseteq Q \Rightarrow h(X) \leq h(Y)$
3. $h(X \cup Y) + h(X \cap Y) \leq h(X) + h(Y)$
4. $h(A \cup \{p_0\}) \in \{h(A), h(A) + 1\}$

Secret Sharing and Polymatroids

Consider as before $P = \{p_1, \dots, p_n\}$ and $Q = P \cup \{p_0\}$

For an arbitrary secret sharing scheme consider,
for every $A \subseteq Q$

$$h(A) = \frac{H(E_A)}{H(E_0)}$$

Then

1. $h(\emptyset) = 0$
2. $X \subseteq Y \subseteq Q \Rightarrow h(X) \leq h(Y)$
3. $h(X \cup Y) + h(X \cap Y) \leq h(X) + h(Y)$
4. $h(A \cup \{p_0\}) \in \{h(A), h(A) + 1\}$

- $\mathcal{S} = (Q, h)$ is a polymatroid
- p_0 is an atomic point of $\mathcal{S}$
- $\Gamma = \Gamma_{p_0}(\mathcal{S}) = \{A \subseteq P \,:\, h(A \cup \{p_0\}) = h(A)\}$

Fujishige 1978, Csirmaz 1997

Ideal Secret Sharing and Matroids

For every ideal secret sharing scheme

$$h(A \cup \{x\}) \in \{h(A), h(A) + 1\} \text{ for all } x \in Q$$

That is, the polymatroid $\mathcal{M} = (Q, h)$ is a matroid

Brickell and Davenport 1991

For every ideal secret sharing scheme

$$h(A \cup \{x\}) \in \{h(A), h(A)+1\} \text{ for all } x \in Q$$

That is, the polymatroid $\mathcal{M} = (Q, h)$ is a matroid

Brickell and Davenport 1991

In this situation we say that $\mathcal{M}$ is ss-representable

Ideal Secret Sharing and Matroids

For every ideal secret sharing scheme

$$h(A \cup \{x\}) \in \{h(A), h(A) + 1\} \text{ for all } x \in Q$$

That is, the polymatroid $\mathcal{M} = (Q, h)$ is a matroid

Brickell and Davenport 1991

In this situation we say that $\mathcal{M}$ is ss-representable

Equivalently, a matroid is ss-representable if its rank function can be defined from the entropy of a family of random variables

For every ideal secret sharing scheme

$$h(A \cup \{x\}) \in \{h(A), h(A) + 1\} \text{ for all } x \in Q$$

That is, the polymatroid $\mathcal{M} = (Q, h)$ is a matroid

Brickell and Davenport 1991

In this situation we say that $\mathcal{M}$ is ss-representable

Equivalently, a matroid is ss-representable if its rank function can be defined from the entropy of a family of random variables

The access structure of an ideal scheme is of the form

$$\Gamma = \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P \, : \, h(A \cup \{p_0\}) = h(A)\}$$

Ideal Secret Sharing and Matroids

For every ideal secret sharing scheme

$$h(A \cup \{x\}) \in \{h(A), h(A) + 1\} \text{ for all } x \in Q$$

That is, the polymatroid $\mathcal{M} = (Q, h)$ is a matroid

Brickell and Davenport 1991

In this situation we say that $\mathcal{M}$ is ss-representable

Equivalently, a matroid is ss-representable if its rank function can be defined from the entropy of a family of random variables

The access structure of an ideal scheme is of the form

$$\Gamma = \Gamma_{p_0}(\mathcal{M}) = \{A \subseteq P \,:\, h(A \cup \{p_0\}) = h(A)\}$$

That is, Γ is a matroid port

At this point, we have a necessary condition

Theorem (Brickell and Davenport 1991)

Every ideal access structure is a matroid port

and a sufficient condition

Theorem (Brickell 1989)

Every port of a representable matroid is an ideal access structure

Problem Solved?

Theorem (Brickell and Davenport 1991)

Every ideal access structure is a matroid port

Theorem (Brickell 1989)

Every port of a representable matroid is an ideal access structure

The necessary condition is not sufficient

Theorem (Seymour 1992)

The Vamos matroid is not ss-representable
There exist non-ideal matroid ports

The sufficient condition is not necessary

Theorem (Simonis and Ashikhmin 1998)

The non-Pappus matroid is not representable
but it is ss-representable

Characterizing Ideal Access Structures

The ideal access structures coincide with the
ports of ss-representable matroids

Problem

Characterize the matroid ports

More later. . .

Problem

Characterize the ss-representable matroids

Interesting techniques to attack this problem have been proposed by
Matúš 1999 and Simonis and Ashikhmin 1998

These problems have been studied (and solved)
for several particular families of access structures

Duality and Minors

Dual access structure: $\Gamma^* = \{A \subseteq P : P - A \notin \Gamma\}$

The minors of access structures are defined by the operations

$$\Gamma \setminus Z = \{A \subseteq P - Z : A \in \Gamma\} \qquad \Gamma / Z = \{A \subseteq P - Z : A \cup Z \in \Gamma\}$$

Properties

- $\Gamma_{p_0}(\mathcal{M}^*) = (\Gamma_{p_0}(\mathcal{M}))^*$,
- $\Gamma_{p_0}(\mathcal{M} \setminus Z) = \Gamma_{p_0}(\mathcal{M}) \setminus Z$,
- $\Gamma_{p_0}(\mathcal{M} / Z) = \Gamma_{p_0}(\mathcal{M}) / Z$

Theorem

The following classes of access structures are minor-closed

1. *Ports of representable matroids*
2. *Ideal access structures*
3. *Matroid ports*

But only the first and the third are known to be closed by duality

Ideal Access Structures. Summary

The access structures	are ports of	the matroids
Vector space a.s.	$\longleftrightarrow$	Representable matroids
$\bigcap$		$\bigcap$
Ideal access structures	$\longleftrightarrow$	ss-Representable matroids
$\bigcap$		$\bigcap$
Matroid ports	$\longleftrightarrow$	Matroids

Complexity of Secret Sharing Schemes

We move now to non-ideal secret sharing schemes

Problem

Find the most efficient secret sharing scheme
for every access structure

$\max H(E_i)$, $\sum H(E_i)$, and $H(E)$, compared to $H(E_0)$,
are used to measure the complexity of a secret sharing scheme

Definition (complexity of a secret sharing scheme)

The complexity $\sigma(\Sigma)$ of a secret sharing scheme Σ is defined as

$$\sigma(\Sigma) = \max_{p_i \in P} \frac{H(E_i)}{H(E_0)} \geq 1$$

The Big Problem

Problem

Find the most efficient secret sharing scheme
for every access structure

Definition (optimal complexity of an access structure)

The optimal complexity $\sigma(\Gamma)$ of an access structure Γ is the infimum of the complexities of all secret sharing schemes for Γ

Problem

Determine $\sigma(\Gamma)$ *for every* Γ
At least, determine the asymptotic behavior of this parameter

Very little is known about this problem

It has been studied as well for several particular families of access structures

Upper Bounds from Constructions

Of course, every construction of a secret sharing scheme Σ for Γ provides an upper bound: $\sigma(\Gamma) \leq \sigma(\Sigma)$

Most of the good construction methods used until now provide linear secret sharing schemes

Upper Bounds from Constructions

Of course, every construction of a secret sharing scheme Σ for Γ provides an upper bound: $\sigma(\Gamma) \leq \sigma(\Sigma)$

Most of the good construction methods used until now provide linear secret sharing schemes

That is, the mapping

$$\begin{aligned} \Pi\colon E &\to E_0 \times E_1 \times \cdots \times E_n \\ x &\mapsto (\pi_0(x) | \pi_1(x), \ldots, \pi_n(x)) \end{aligned}$$

is linear and the the uniform probability distribution is taken on E

Definition

For an access structure Γ, we define $\lambda(\Gamma)$ as the infimum of the complexities of all linear secret sharing schemes for Γ

Obviously, $\sigma(\Gamma) \leq \lambda(\Gamma)$

How Good Are Linear Secret Sharing Schemes?

For some access structures, the optimal schemes must be non-linear

Beimel and Weinreb (2005) Proved a strong separation result:
There exist a family of access structures such that
$\sigma(\Gamma_n)$ grows linearly while
$\lambda(\Gamma_n)$ grows superpolynomially

Lower Bounds from Polymatroids

For a polymatroid $\mathcal{S} = (Q, h)$, we define $\sigma(\mathcal{S}) = \max_{p \in Q} h(\{p\})$

Every polymatroid $\mathcal{S} = (Q, h)$ with an atomic point $p_0 \in Q$ defines an access structure on $P = Q - p_0$

$$\Gamma = \Gamma_{p_0}(\mathcal{S}) = \{A \subseteq P \; : \; h(A \cup \{p_0\}) = h(A)\}$$

In this situation, we say that $\mathcal{S}$ is a Γ-polymatroid

$$\kappa(\Gamma) = \inf\{\sigma(\mathcal{S}) \; : \; \Gamma = \Gamma_{p_0}(\mathcal{S})\}$$

Lower Bounds from Polymatroids

For a polymatroid $\mathcal{S} = (Q, h)$, we define $\sigma(\mathcal{S}) = \max_{p \in Q} h(\{p\})$

Every polymatroid $\mathcal{S} = (Q, h)$ with an atomic point $p_0 \in Q$ defines an access structure on $P = Q - p_0$

$$\Gamma = \Gamma_{p_0}(\mathcal{S}) = \{A \subseteq P \ : \ h(A \cup \{p_0\}) = h(A)\}$$

In this situation, we say that $\mathcal{S}$ is a Γ-polymatroid

$$\kappa(\Gamma) = \inf\{\sigma(\mathcal{S}) \ : \ \Gamma = \Gamma_{p_0}(\mathcal{S})\}$$

A secret sharing scheme Σ for Γ defines a polymatroid $\mathcal{S} = \mathcal{S}(\Sigma)$ such that $\Gamma = \Gamma_{p_0}(\mathcal{S})$ and $\sigma(\Sigma) = \sigma(\mathcal{S})$

Therefore $\kappa(\Gamma) \leq \sigma(\mathcal{S}) = \sigma(\Sigma)$

Lower Bounds from Polymatroids

For a polymatroid $\mathcal{S} = (Q, h)$, we define $\sigma(\mathcal{S}) = \max_{p \in Q} h(\{p\})$

Every polymatroid $\mathcal{S} = (Q, h)$ with an atomic point $p_0 \in Q$ defines an access structure on $P = Q - p_0$

$$\Gamma = \Gamma_{p_0}(\mathcal{S}) = \{A \subseteq P \,:\, h(A \cup \{p_0\}) = h(A)\}$$

In this situation, we say that $\mathcal{S}$ is a Γ-polymatroid

$$\kappa(\Gamma) = \inf\{\sigma(\mathcal{S}) \,:\, \Gamma = \Gamma_{p_0}(\mathcal{S})\}$$

A secret sharing scheme Σ for Γ defines a polymatroid $\mathcal{S} = \mathcal{S}(\Sigma)$ such that $\Gamma = \Gamma_{p_0}(\mathcal{S})$ and $\sigma(\Sigma) = \sigma(\mathcal{S})$

Therefore $\kappa(\Gamma) \leq \sigma(\mathcal{S}) = \sigma(\Sigma)$

Theorem

For every access structure Γ

$$\kappa(\Gamma) \leq \sigma(\Gamma) \leq \lambda(\Gamma)$$

Minors

The minors of access structures are defined by the operations

$$\Gamma \setminus Z = \{A \subseteq P - Z \,:\, A \in \Gamma\} \qquad \Gamma / Z = \{A \subseteq P - Z \,:\, A \cup Z \in \Gamma\}$$

Minors of a polymatroid $\mathcal{S} = (Q, h)$

- $\mathcal{S} \setminus Z = (Q - Z, h_{\setminus Z})$, where $h_{\setminus Z}(A) = h(A)$
- $\mathcal{S} / Z = (Q - Z, h_{/Z})$, where $h_{/Z}(A) = h(A \cup Z) - h(Z)$

Theorem

If Γ' is a minor of Γ, then

$$\kappa(\Gamma') \le \kappa(\Gamma) \qquad \sigma(\Gamma') \le \sigma(\Gamma) \qquad \lambda(\Gamma') \le \lambda(\Gamma)$$

Duality

Dual access structure: $\Gamma^* = \{A \subseteq P : P - A \notin \Gamma\}$

Since linear secret sharing schemes can be identified to linear codes,

Theorem (Jackson and Martin 1994)

For every access structure Γ,

$$\lambda(\Gamma^*) = \lambda(\Gamma)$$

By considering a suitable definition of dual polymatroid,

Theorem (Martí-Farré and P. 2007)

For every access structure Γ,

$$\kappa(\Gamma^*) = \kappa(\Gamma)$$

The relationship between $\sigma(\Gamma^*)$ and $\sigma(\Gamma)$ is unknown

How Good Are Combinatorial Lower Bounds?

Theorem (Csirmaz 1997)

There exist a family of access structures with

$$\sigma(\Gamma_n) \geq \kappa(\Gamma_n) \geq \frac{n}{\log n}$$

This is the best known general lower bound on σ

But, on the other hand

Theorem (Csirmaz 1997)

For every access structure Γ on n participants, $\kappa(\Gamma) \leq n$

This seems to imply that $\kappa(\Gamma)$
must be in general much smaller than $\sigma(\Gamma)$

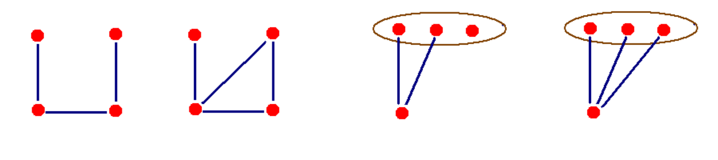
Nevertheless no strong separation result
between these parameters is known

Non-Shannon information inequalities (next talk)

An Old Result on Matroid Ports

Theorem (Seymour 1976)

An access structure is a matroid port if and only if it has no minor isomorphic to Φ, $\widehat{\Phi}$, $\widehat{\Phi}^*$ or Ψ_s with $s \geq 3$.



Since all these forbidden minors satisfy $\sigma(\Gamma) \geq \kappa(\Gamma) \geq 3/2$

Corollary (Martí-Farré and P. 2007)

If $\sigma(\Gamma) < 3/2$, then Γ is a matroid port

In addition, there is no access structure with $1 < \kappa(\Gamma) < 3/2$

Self-Dual Codes and Identically Self-Dual Matroids

Every linear code defines a vector space secret sharing scheme

$$(x_1, \dots, x_d) \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \pi_0 & \pi_1 & \cdots & \pi_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = (k, s_1, \dots, s_n)$$

If the code is self-dual, then the secret sharing scheme is multiplicative because

$$kk' + s_1 s_1' + \cdots + s_n s_n' = 0$$

The access structure is self-dual, $\Gamma^* = \Gamma$
It is the port of a representable identically self-dual matroid

Self-Dual Codes and Identically Self-Dual Matroids

Problem

Can every representable identically self-dual matroid be represented by a self-dual code?

The answer is yes for

- Binary matroids
- Uniform matroids
- Bipartite matroids (Cramer et al. 2005)
- Matroids with up to 8 points (Gracia and P. 2006)

References

Mainly

- J. Martí-Farré, C. Padró
  On secret Sharing Schemes, Matroids and Polymatroids
  TCC 2007, Lect. Notes in Comput. Sci. **4392** (2007) 273–290
  Full version available at my webpage

But also

- R. Cramer, V. Daza, I. Gracia, J. Jiménez Urroz, G. Leander, J. Martí-Farré, C. Padró
  On codes, matroids and secure multi-party computation from linear secret sharing schemes
  IEEE Transactions on Information Theory **54** (2008) 2644–2657
- C. Padró, I. Gracia
  Representing small identically self-dual matroids by self-dual codes
  SIAM Journal on Discrete Mathematics **20** (2006) 1046–1055