ゲーム理論と暗号理論

安永憲司

九州先端科学技術研究所 (ISIT)

暗号理論秋学校@河口湖セントビレッヂ 2012/9/24 - 27

ゲーム理論とは何か

- 複数の意思決定者が相互作用する状況（ゲーム的状況）を研究する理論
  - 自分の利益が他者の行動に依存する状況
  - 一人での意思決定は（あまり）考えない
  - 意思決定を行うとき、
    相手がどう行動するかを考えないといけない

ゲームの例（秋学校ゲーム）

ゲームの例（秋学校ゲーム）

- 秋学校 A と秋学校 B が同時期の開催だと判明

ゲームの例（秋学校ゲーム）

- 秋学校 A と秋学校 B が同時期の開催だと判明
  - 例えば A は 9/24 – 27, B は 9/24 – 26

ゲームの例（秋学校ゲーム）

- 秋学校 A と秋学校 B が同時期の開催だと判明
  - 例えば A は 9/24 – 27, B は 9/24 – 26
- 日程が重ならなければともに参加者が増加

ゲームの例（秋学校ゲーム）

- 秋学校 A と秋学校 B が同時期の開催だと判明
  - 例えば A は 9/24 – 27, B は 9/24 – 26
- 日程が重ならなければともに参加者が増加
- しかし、一度決めた日程を変更するには
講師達の都合・場所の確保等の再調整が必要

ゲームの例（秋学校ゲーム）

- 秋学校 A と秋学校 B が同時期の開催だと判明
  - 例えば A は 9/24 – 27, B は 9/24 – 26
- 日程が重ならなければともに参加者が増加
- しかし、一度決めた日程を変更するには
講師達の都合・場所の確保等の再調整が必要
- 主催者として、自分たちだけの変更は不満

ゲームの例（秋学校ゲーム）

- 秋学校 A と秋学校 B が同時期の開催だと判明
  - 例えば A は 9/24 – 27, B は 9/24 – 26
- 日程が重ならなければともに参加者が増加
- しかし、一度決めた日程を変更するには
  講師達の都合・場所の確保等の再調整が必要
- 主催者として、自分たちだけの変更は不満

⇒ 日程の変更は行われるだろうか？

ゲームの例（秋学校ゲーム）

- 利得
  - 現状のまま → 10
  - 参加者増加 → + 4
  - 調整コスト → - 3
  - 自分たちだけ変更 → - 2

ゲームの例（秋学校ゲーム）

- 利得
  - 現状のまま → 10
  - 参加者増加 → + 4
  - 調整コスト → - 3
  - 自分たちだけ変更 → - 2

秋学校 A ＼ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(14, 9)
変更する	(9, 14)	(11, 11)

ゲームの例（秋学校ゲーム）

秋学校 A \ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(14, 9)
変更する	(9, 14)	(11, 11)

- 行動分析

ゲームの例（秋学校ゲーム）

秋学校 A ＼ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(14, 9)
変更する	(9, 14)	(11, 11)

- 行動分析
  - 秋学校 A は、秋学校 B の行動によらず、「変更しない」の方が高利得（B も同様）

ゲームの例（秋学校ゲーム）

秋学校Ａ ＼ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(14, 9)
変更する	(9, 14)	(11, 11)

- 行動分析
  - 秋学校 A は、秋学校 B の行動によらず、「変更しない」の方が高利得（B も同様）
  - したがって、ともに「変更しない」を選択

ゲームの例（秋学校ゲーム）

秋学校 A \ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(14, 9)
変更する	(9, 14)	(11, 11)

- 行動分析
  - 秋学校 A は、秋学校 B の行動によらず、「変更しない」の方が高利得（B も同様）
  - したがって、ともに「変更しない」を選択
  - ともに「変更する」の方が高利得だがそれを選択しない → 囚人のジレンマ

ゲームの例（秋学校ゲーム）

秋学校 A \ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(10 + x, 10 + x - y - z)
変更する	(10 + x - y - z, 10 + x)	(10 + x - y, 10 + x - y)

- 利得を一般化
  - 参加者増加 → + x, 調整コスト → - y, 不満 → - z

ゲームの例（秋学校ゲーム）

秋学校 A \ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(10 + x, 10 + x - y - z)
変更する	(10 + x - y - z, 10 + x)	(10 + x - y, 10 + x - y)

- 利得を一般化
  - 参加者増加 → + x，調整コスト → - y, 不満 → - z
- x - y < z であれば、同じ結果
  - 調整して参加者を増やすこと（x - y）よりも不満（z）が大きいとき

ゲームの例（秋学校ゲーム）

秋学校 A \ 秋学校B	変更しない	変更する
変更しない	(10, 10)	(10 + x, 10 + x - y - z)
変更する	(10 + x - y - z, 10 + x)	(10 + x - y, 10 + x - y)

- 利得を一般化
  - 参加者増加 → + x, 調整コスト → - y, 不満 → - z
- x - y < z であれば、同じ結果
  - 調整して参加者を増やすこと（x - y）よりも
    不満（z）が大きいとき

→ 日程が重なった秋学校があれば、
それは主催者の不満が大きかったと考えられる（？）

ゲーム理論の用語

- プレイヤー：意思決定を行う主体
- 行動：プレイヤーがもつ選択肢
- 戦略：行動計画
- 利得：ゲームを実行した結果として得られる数値（大きい方が望ましい）
- 利得関数：ゲームの結果を数値に対応させる関数
- ゲームの解：ゲームにおいて予想される結果

ゲームのバリエーション

- 戦略型ゲームと展開型ゲーム
  - 戦略型：すべてのプレイヤーが同時に行動
  - 展開型：それ以外
- 完備情報ゲームと不完備情報ゲーム
  - 完備情報：ゲームの情報（プレイヤー・利得・行動の候補）に不確実性がないもの
- 完全情報ゲームと不完全情報ゲーム
  - 完全情報：自分以前のプレイヤーの行動選択がわかるとき（戦略型は不完全情報ゲーム）

解の見つけ方

解の見つけ方

- 支配戦略を探す
  - 支配戦略：他のプレイヤーがどの戦略をとっても、自分の他の戦略よりも良い戦略
    - σ_i が支配戦略 $\Leftrightarrow \forall \rho_i \neq \sigma_i, \forall \rho_{-i}, U_i(\sigma_i, \rho_{-i}) > U_i(\rho_i, \rho_{-i})$

解の見つけ方

- 支配戦略を探す
  - 支配戦略：他のプレイヤーがどの戦略をとっても、自分の他の戦略よりも良い戦略
    - σ_i が支配戦略 ⇔ $\forall \rho_i \neq \sigma_i, \forall \rho_{-i}, U_i(\sigma_i, \rho_{-i}) > U_i(\rho_i, \rho_{-i})$
- 最適反応戦略を考える
  - 最適反応戦略：他のプレイヤーの戦略に対し、自分の利得を最大化する戦略
    - σ_i が σ_{-i} の最適反応 ⇔ $\forall \rho_i \neq \sigma_i, U_i(\sigma_i, \sigma_{-i}) \geq U_i(\rho_i, \sigma_{-i})$

ゲームの例（合理的な豚）

ゲームの例（合理的な豚）

- 檻の中に大きな豚と小さな豚

ゲームの例（合理的な豚）

- 檻の中に大きな豚と小さな豚
- 離れた場所のボタンを押すとエサが出てくる

ゲームの例（合理的な豚）

- 檻の中に大きな豚と小さな豚
- 離れた場所のボタンを押すとエサが出てくる
- 豚は餌を食べたいがなるべく動きたくない

ゲームの例（合理的な豚）

- 檻の中に大きな豚と小さな豚
- 離れた場所のボタンを押すとエサが出てくる
- 豚は餌を食べたいがなるべく動きたくない
- ２匹とも押しに行くと、大豚がエサを全部食べる

ゲームの例（合理的な豚）

- 檻の中に大きな豚と小さな豚
- 離れた場所のボタンを押すとエサが出てくる
- 豚は餌を食べたいがなるべく動きたくない
- ２匹とも押しに行くと、大豚がエサを全部食べる
- 大豚だけ押しに行くと、戻る間に小豚が半分食べる

ゲームの例（合理的な豚）

- 檻の中に大きな豚と小さな豚
- 離れた場所のボタンを押すとエサが出てくる
- 豚は餌を食べたいがなるべく動きたくない
- ２匹とも押しに行くと、大豚がエサを全部食べる
- 大豚だけ押しに行くと、戻る間に小豚が半分食べる
- 利得
  - エサを全部食べる → 10
  - ボタンを押しに行く → - 2

ゲームの例（合理的な豚）

- 檻の中に大きな豚と小さな豚
- 離れた場所のボタンを押すとエサが出てくる
- 豚は餌を食べたいがなるべく動きたくない
- ２匹とも押しに行くと、大豚がエサを全部食べる
- 大豚だけ押しに行くと、戻る間に小豚が半分食べる
- 利得
  - エサを全部食べる → 10
  - ボタンを押しに行く → - 2

どのような結果になるだろうか？

ゲームの例（合理的な豚）

大きな豚 \ 小さな豚	ボタンを押しに行く	エサ場で待つ
ボタンを押しに行く	(8, -2)	(3, 5)
エサ場で待つ	(10, -2)	(0, 0)

ゲームの例（合理的な豚）

大きな豚 ＼ 小さな豚	ボタンを押しに行く	エサ場で待つ
ボタンを押しに行く	(8, -2)	(3, 5)
エサ場で待つ	(10, -2)	(0, 0)

- 小さな豚にとって「エサ場で待つ」が支配戦略

ゲームの例（合理的な豚）

大きな豚 ＼ 小さな豚	ボタンを押しに行く	エサ場で待つ
ボタンを押しに行く	(8, -2)	(3, 5)
エサ場で待つ	(10, -2)	(0, 0)

- 小さな豚にとって「エサ場で待つ」が支配戦略
- 小さな豚が「エサ場で待つ」とき、
  大きな豚は「ボタンを押しに行く」が最適反応

ゲームの例（合理的な豚）

大きな豚 \ 小さな豚	ボタンを押しに行く	エサ場で待つ
ボタンを押しに行く	(8, -2)	(3, 5)
エサ場で待つ	(10, -2)	(0, 0)

- 小さな豚にとって「エサ場で待つ」が支配戦略
- 小さな豚が「エサ場で待つ」とき、
  大きな豚は「ボタンを押しに行く」が最適反応

小さな豚がエサ場で待っていれば、
大きな豚がボタンを押しに行く

Nash 均衡

Nash 均衡

- すべてのプレイヤーの戦略が
最適反応戦略である戦略の組
    - $\sigma = (\sigma_1, \ldots, \sigma_n)$ が Nash 均衡
$\Leftrightarrow \forall i, \forall \rho_i, U_i(\sigma_i, \sigma_{-i}) \geq U_i(\rho_i, \sigma_{-i})$

Nash 均衡

- すべてのプレイヤーの戦略が
最適反応戦略である戦略の組
  - $\sigma = (\sigma_1, \ldots, \sigma_n)$ が Nash 均衡
$\Leftrightarrow \forall i, \forall \rho_i, U_i(\sigma_i, \sigma_{-i}) \geq U_i(\rho_i, \sigma_{-i})$
  - 他のプレイヤーがその戦略に従うとき、
どのような他の戦略をとっても、
利得を高くできないとき

Nash 均衡

- すべてのプレイヤーの戦略が最適反応戦略である戦略の組
  - $\sigma = (\sigma_1, \ldots, \sigma_n)$ が Nash 均衡
    $\Leftrightarrow \forall\, i, \forall\, \rho_i, U_i(\sigma_i, \sigma_{-i}) \geq U_i(\rho_i, \sigma_{-i})$
  - 他のプレイヤーがその戦略に従うとき、どのような他の戦略をとっても、利得を高くできないとき

戦略型ゲームの解は Nash 均衡であるべき
（ただし、十分であるとは考えられていない）

戦略の弱支配関係

戦略の弱支配関係

- 戦略 σ_i が戦略 ρ_i を弱支配

  ⇔　他のプレイヤーがどの戦略をとっても
  σ_i が ρ_i より悪くなることはなく、かつ
  他のプレイヤーのある戦略において、
  σ_i が ρ_i より真に良い

戦略の弱支配関係

- 戦略 σ_i が戦略 ρ_i を弱支配

  ⇔　他のプレイヤーがどの戦略をとっても
  σ_i が ρ_i より悪くなることはなく、かつ
  他のプレイヤーのある戦略において、
  σ_i が ρ_i より真に良い

合理的なプレイヤーは
弱支配される戦略を選択しないと考えられる

Nash 均衡に関する事実

Nash 均衡に関する事実

- Nash 均衡は複数存在することがある

Nash 均衡に関する事実

- Nash 均衡は複数存在することがある
- Nash 均衡は弱支配されることがある

Nash 均衡に関する事実

- Nash 均衡は複数存在することがある
- Nash 均衡は弱支配されることがある

→ 弱支配されない Nash 均衡が解であるべき

Nash 均衡に関する事実

- Nash 均衡は複数存在することがある
- Nash 均衡は弱支配されることがある

→ 弱支配されない Nash 均衡が解であるべき

1 \ 2	x	y
a	(5, 2)	(10, 0)
b	(2, 0)	(10, 2)

Nash 均衡に関する事実

- Nash 均衡は複数存在することがある
- Nash 均衡は弱支配されることがある

→ 弱支配されない Nash 均衡が解であるべき

1 \ 2	x	y
a	(5, 2)	(10, 0)
b	(2, 0)	(10, 2)

- (a, x) と (b, y) が Nash 均衡

Nash 均衡に関する事実

- Nash 均衡は複数存在することがある
- Nash 均衡は弱支配されることがある

→ 弱支配されない Nash 均衡が解であるべき

1 \ 2	x	y
a	(5, 2)	(10, 0)
b	(2, 0)	(10, 2)

- (a, x) と (b, y) が Nash 均衡
- しかし、戦略 b は戦略 a に弱支配
  → (b, y) は解でないと考えられる

Nash 均衡に関する事実（続き）

Nash 均衡に関する事実（続き）

- 純粋戦略 Nash 均衡は存在するとは限らない
  - 純粋戦略：行動が確定的
  - 混合戦略：行動が確率的

Nash 均衡に関する事実（続き）

- 純粋戦略 Nash 均衡は存在するとは限らない
  - 純粋戦略：行動が確定的
  - 混合戦略：行動が確率的

1 ＼ 2	表	裏
表	(1, -1)	(-1, 1)
裏	(-1, 1)	(1, -1)

Nash 均衡に関する事実（続き）

- 純粋戦略 Nash 均衡は存在するとは限らない
  - 純粋戦略：行動が確定的
  - 混合戦略：行動が確率的

1 \ 2	表	裏
表	(1, -1)	(-1, 1)
裏	(-1, 1)	(1, -1)

- 任意の有限ゲームにおいて、
  混合戦略を含めれば Nash 均衡は存在

展開型ゲーム

展開型ゲーム

- すべてのプレイヤーが同時に行動するとは限らないゲーム
  - ゲームは逐次的に行われる

展開型ゲーム

- すべてのプレイヤーが同時に行動するとは限らないゲーム
  - ゲームは逐次的に行われる
- プレイヤーの戦略は、履歴を行動に対応させる関数
  - 戦略型では、一度決めるだけ

展開型ゲーム

- すべてのプレイヤーが同時に行動するとは限らないゲーム
  - ゲームは逐次的に行われる
- プレイヤーの戦略は、履歴を行動に対応させる関数
  - 戦略型では、一度決めるだけ
- 利得関数は、終着履歴（ゲームの結果）から数値への関数
  - 戦略型でも、ゲームの結果から数値への関数

ゲームの例（鍋の争い）

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催
- もつ鍋のテーブルと水炊きのテーブルが用意

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催
- もつ鍋のテーブルと水炊きのテーブルが用意
- N 君と K 君はたくさん食べることで有名

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催
- もつ鍋のテーブルと水炊きのテーブルが用意
- N 君と K 君はたくさん食べることで有名
- ２人ともその日はもつ鍋が食べたい気分

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催
- もつ鍋のテーブルと水炊きのテーブルが用意
- N 君と K 君はたくさん食べることで有名
- ２人ともその日はもつ鍋が食べたい気分
- しかし、同じテーブルだとたくさん食べられない

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催
- もつ鍋のテーブルと水炊きのテーブルが用意
- N 君と K 君はたくさん食べることで有名
- ２人ともその日はもつ鍋が食べたい気分
- しかし、同じテーブルだとたくさん食べられない
  - 同じテーブルの時 K 君は先輩の N 君に遠慮する

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催
- もつ鍋のテーブルと水炊きのテーブルが用意
- N 君と K 君はたくさん食べることで有名
- ２人ともその日はもつ鍋が食べたい気分
- しかし、同じテーブルだとたくさん食べられない
  - 同じテーブルの時 K 君は先輩の N 君に遠慮する
- ２人が店に到着

ゲームの例（鍋の争い）

- T 研究室の打ち上げが鍋のおいしい店で開催
- もつ鍋のテーブルと水炊きのテーブルが用意
- N 君と K 君はたくさん食べることで有名
- ２人ともその日はもつ鍋が食べたい気分
- しかし、同じテーブルだとたくさん食べられない
  - 同じテーブルの時 K 君は先輩の N 君に遠慮する
- ２人が店に到着

→ ２人はどちらのテーブルに着席すべきか？

ゲームの例（鍋の争い）

- 利得
  - 別々のテーブルでもつ鍋 → 100
  - 別々のテーブルで水炊き → 60
  - 同じテーブルでもつ鍋 → K 君 30, N 君 70
  - 同じテーブルで水炊き → K 君 20, N 君 40

ゲームの例（鍋の争い）

- 利得
  - 別々のテーブルでもつ鍋 → 100
  - 別々のテーブルで水炊き → 60
  - 同じテーブルでもつ鍋 → K 君 30, N 君 70
  - 同じテーブルで水炊き → K 君 20, N 君 40
- ２人が同時に着席する場合（戦略型ゲーム）

ゲームの例（鍋の争い）

- 利得
  - 別々のテーブルでもつ鍋 → 100
  - 別々のテーブルで水炊き → 60
  - 同じテーブルでもつ鍋 → K 君 30, N 君 70
  - 同じテーブルで水炊き → K 君 20, N 君 40
- ２人が同時に着席する場合（戦略型ゲーム）

K 君 ＼ N 君	もつ鍋	水炊き
もつ鍋	(30, 70)	(100, 60)
水炊き	(60, 100)	(20, 40)

ゲームの例（鍋の争い）

- 利得
  - 別々のテーブルでもつ鍋 → 100
  - 別々のテーブルで水炊き → 60
  - 同じテーブルでもつ鍋 → K 君 30, N 君 70
  - 同じテーブルで水炊き → K 君 20, N 君 40
- ２人が同時に着席する場合（戦略型ゲーム）
  - (もつ鍋, 水炊き), (水炊き, もつ鍋) が Nash 均衡

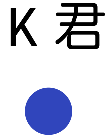
K 君 ＼ N 君	もつ鍋	水炊き
もつ鍋	(30, 70)	(100, 60)
水炊き	(60, 100)	(20, 40)

ゲームの例（鍋の争い）

- K 君が先に着席する場合

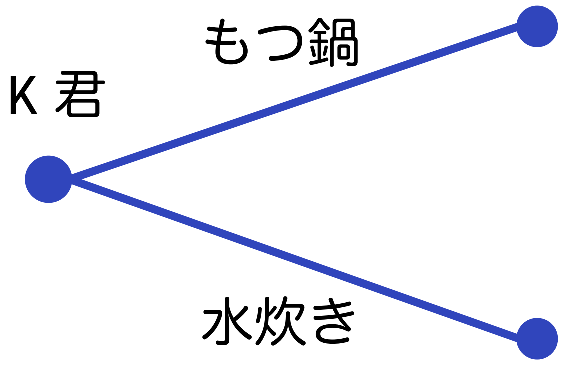
ゲームの例（鍋の争い）

- K 君が先に着席する場合



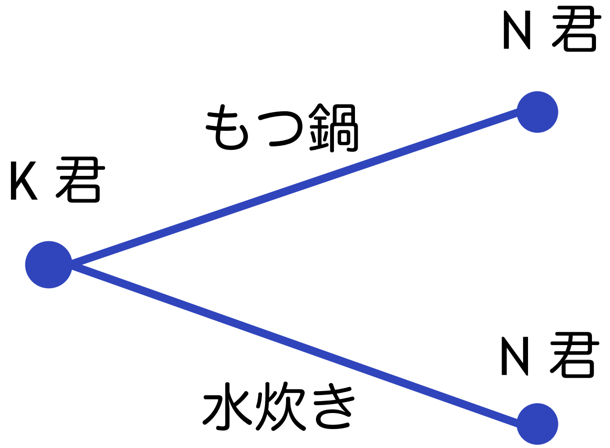
ゲームの例（鍋の争い）

- K 君が先に着席する場合



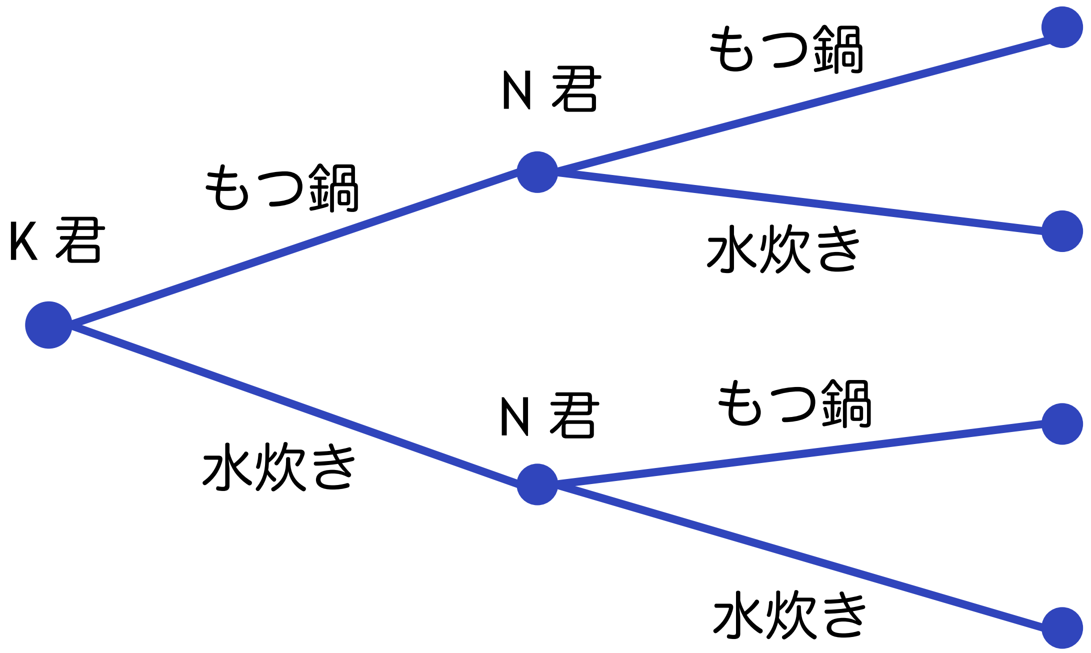
ゲームの例（鍋の争い）

- K 君が先に着席する場合



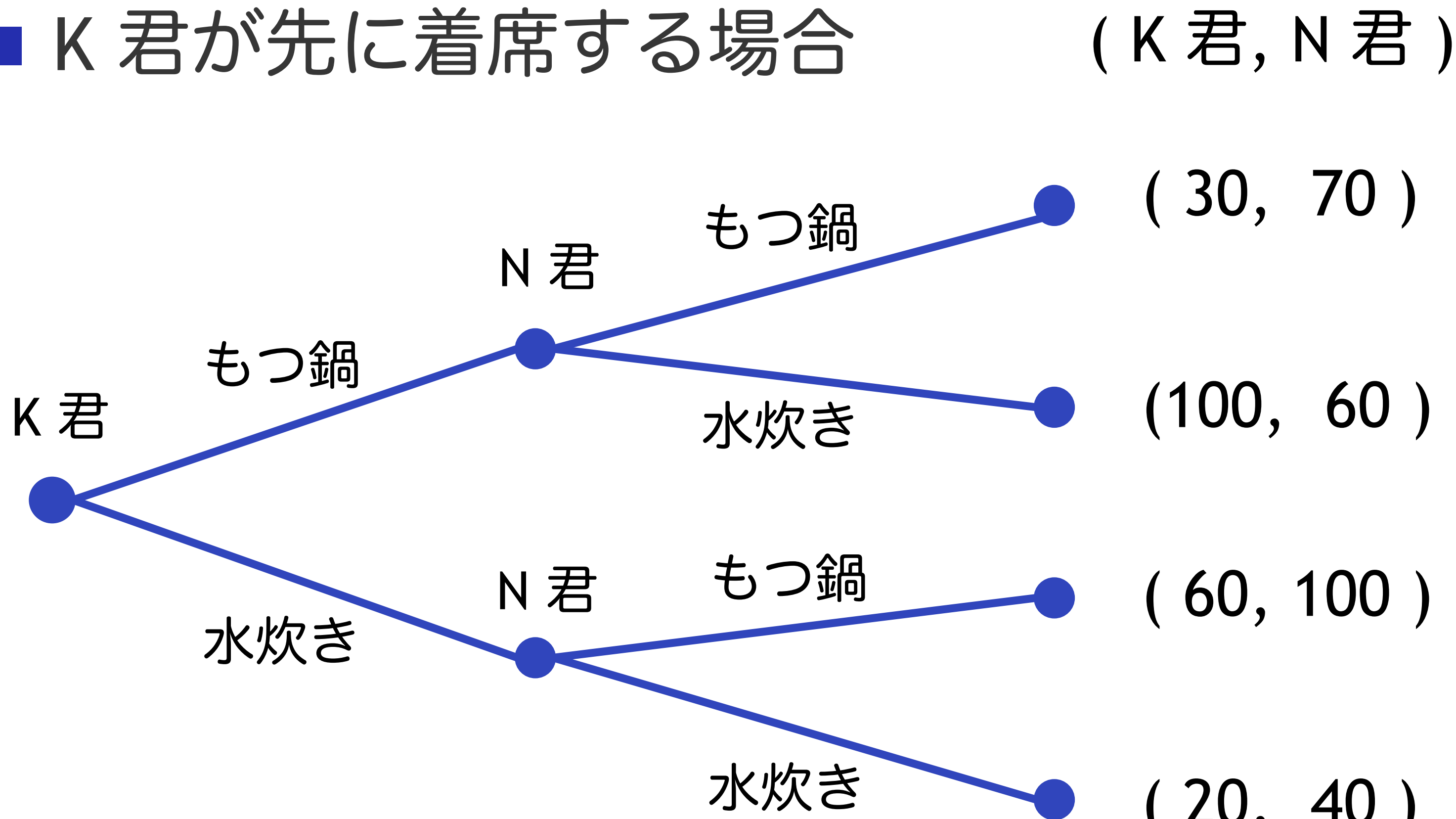
ゲームの例（鍋の争い）

■ K 君が先に着席する場合



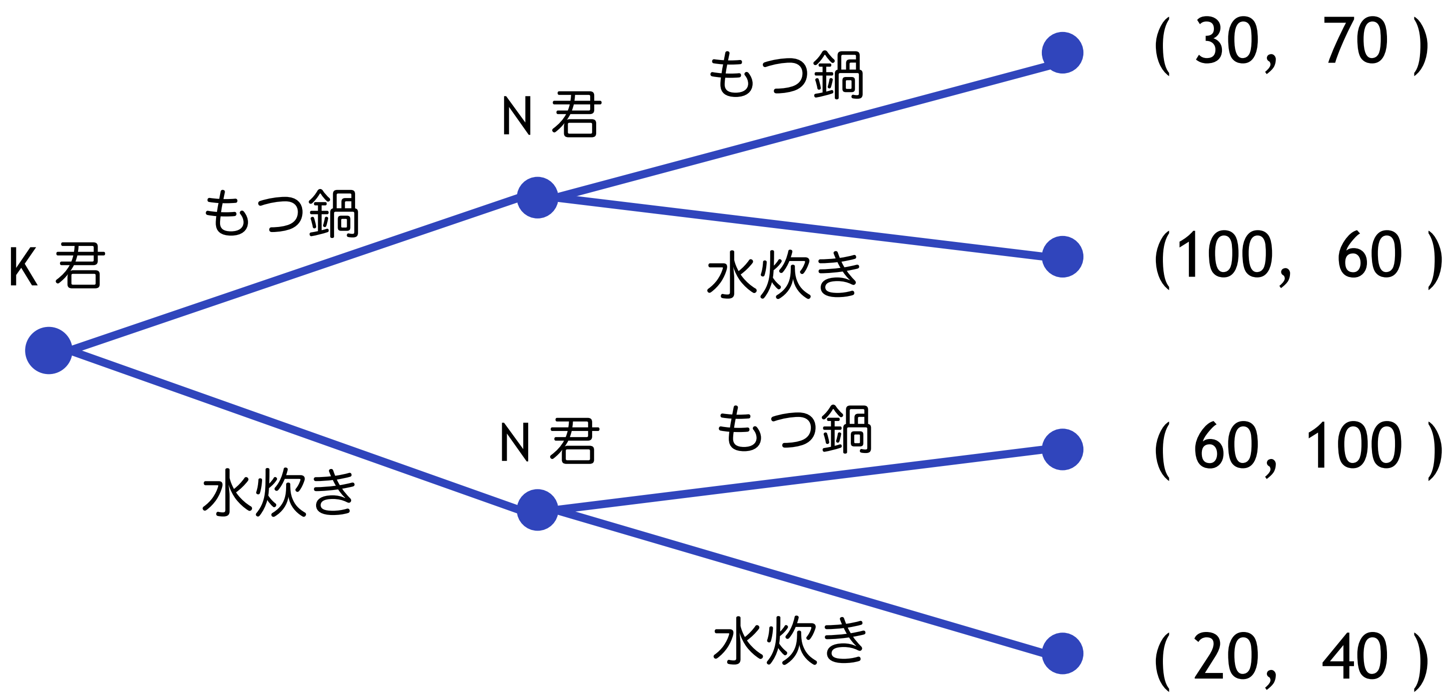
ゲームの例（鍋の争い）

- K 君が先に着席する場合



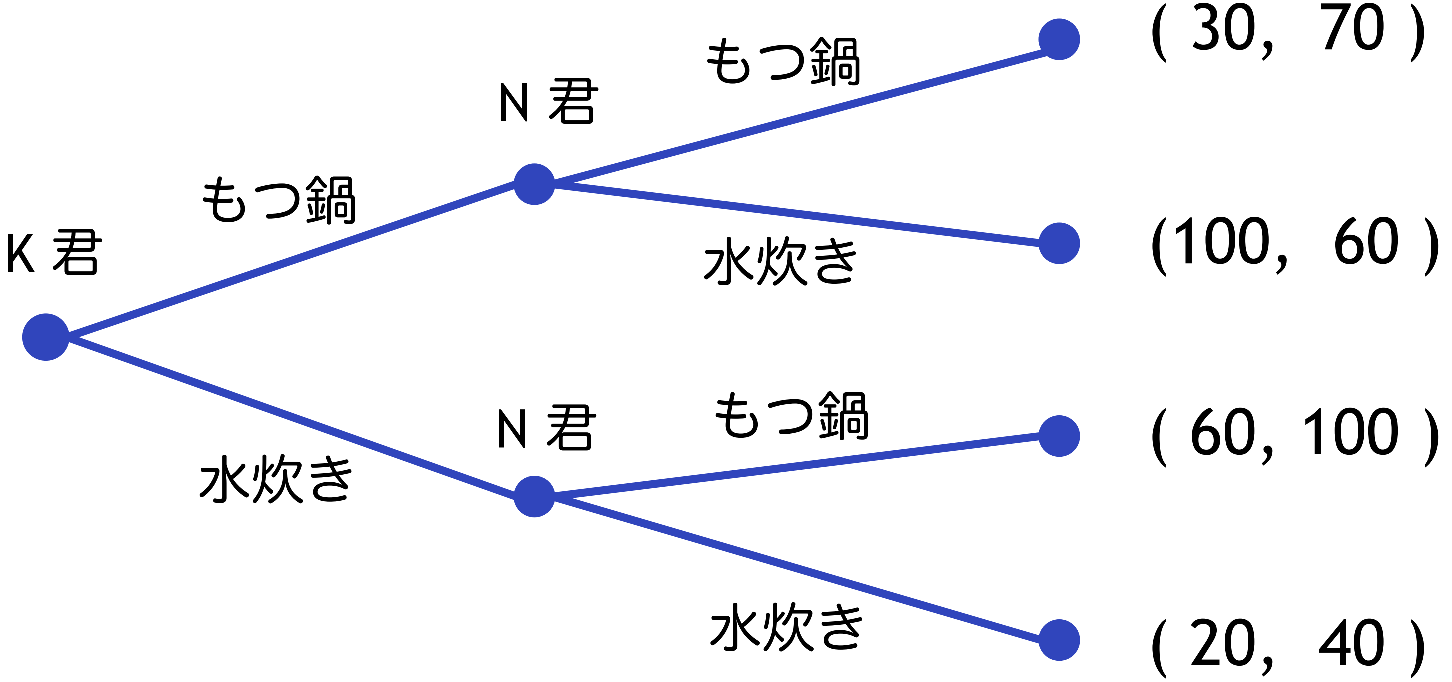
ゲームの例（鍋の争い）

■ K 君が先に着席する場合　　(K 君, N 君)

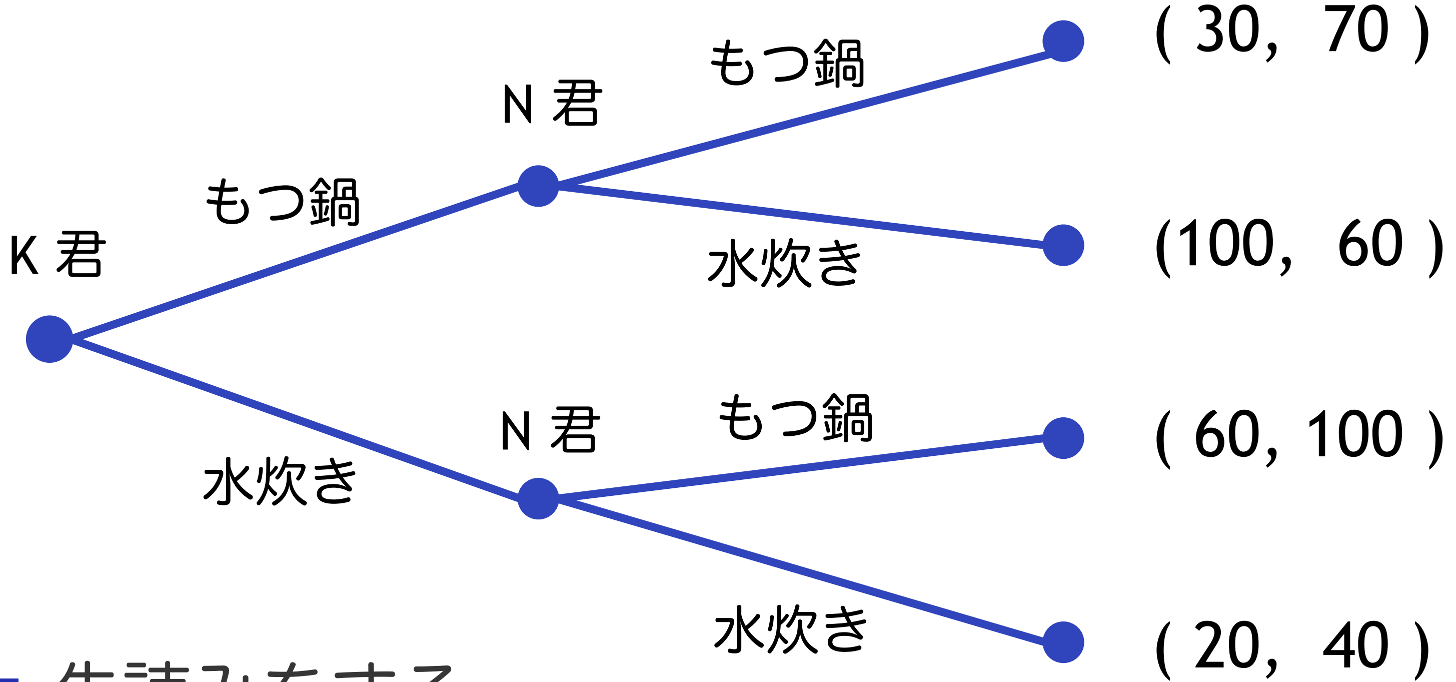


ゲームの解は何か？

展開型ゲームにおける解の見つけ方

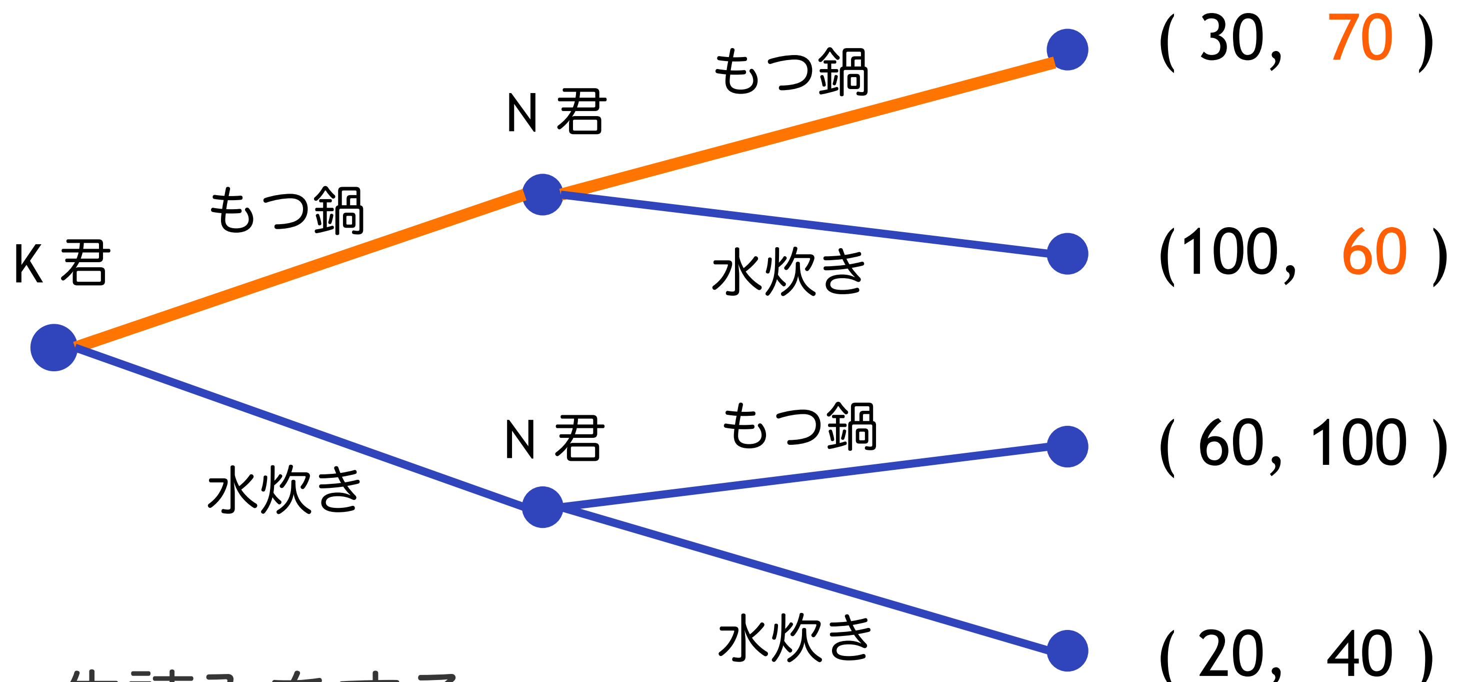


展開型ゲームにおける解の見つけ方



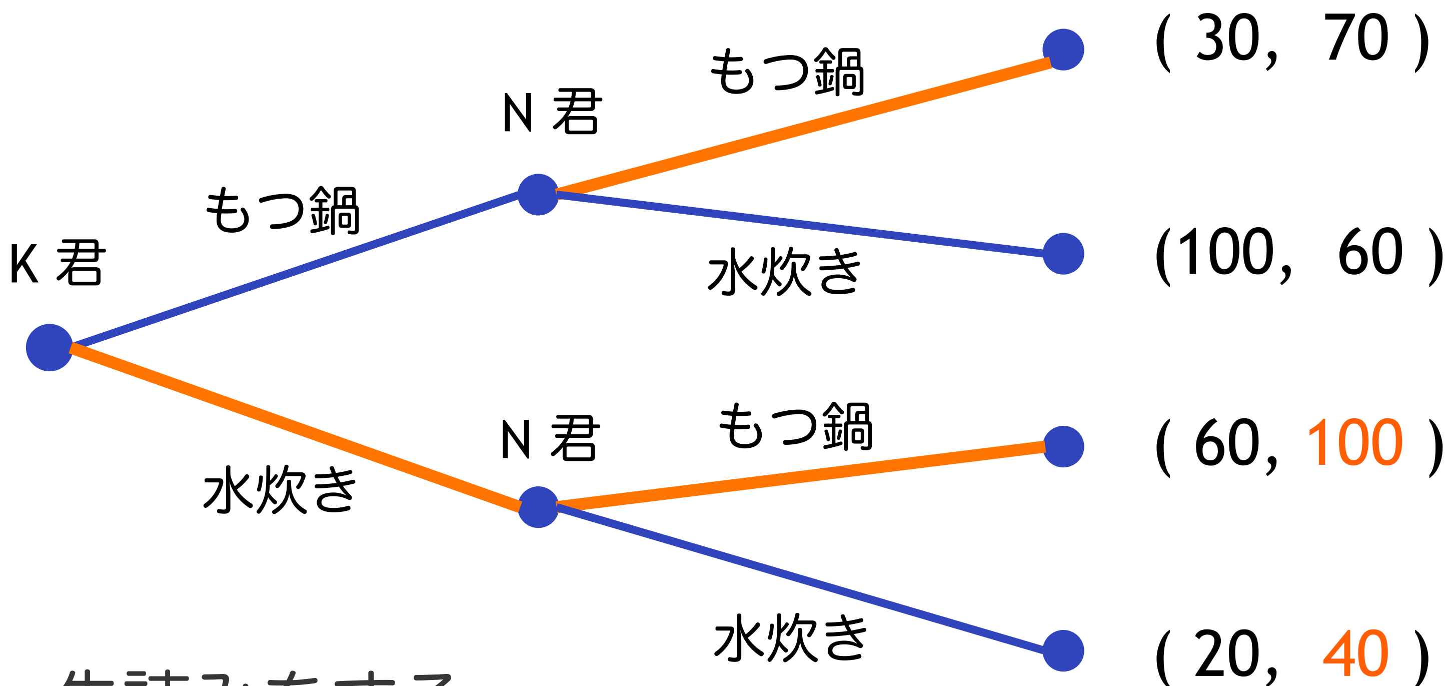
- 先読みをする

展開型ゲームにおける解の見つけ方



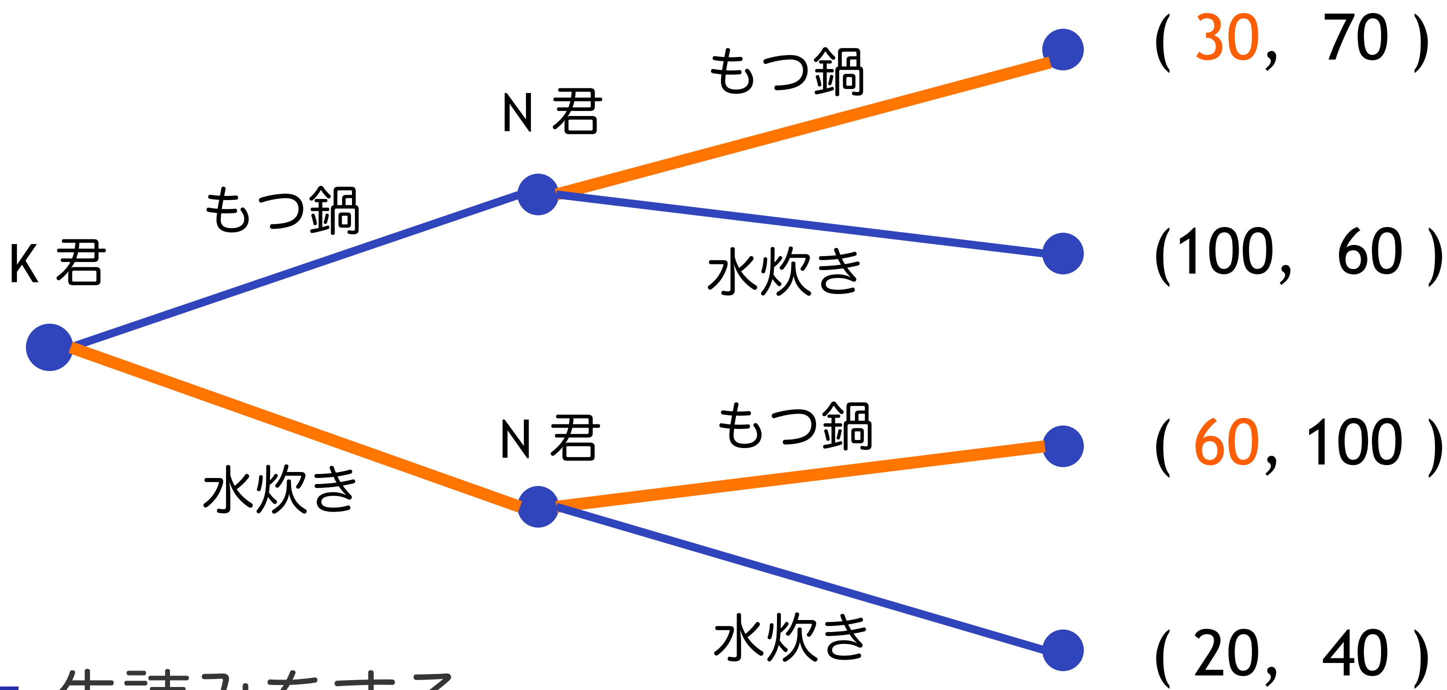
- 先読みをする
  - K 君が「もつ鍋」のとき、N 君は「もつ鍋」

展開型ゲームにおける解の見つけ方



- 先読みをする
  - K 君が「もつ鍋」のとき、N 君は「もつ鍋」
  - K 君が「水炊き」のとき、N 君は「もつ鍋」

展開型ゲームにおける解の見つけ方



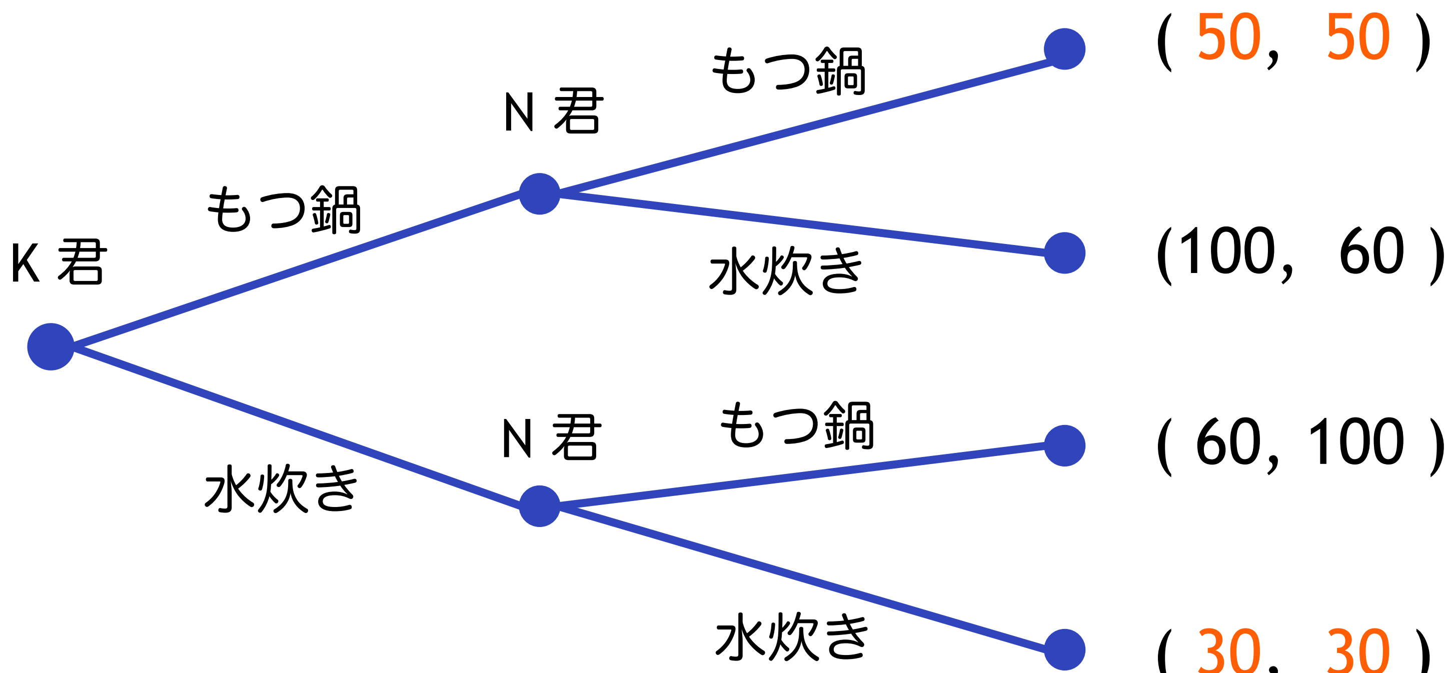
- 先読みをする
  - K 君が「もつ鍋」のとき、N 君は「もつ鍋」
  - K 君が「水炊き」のとき、N 君は「もつ鍋」
  - N 君はいずれにしても「もつ鍋」なので、K 君は「水炊き」を選ぶ

展開型ゲームにおける用語

- 手番：ゲーム木における（終点以外の）点
  - プレイヤーの意思決定が行われる
- 終点：ゲームの結果が判明する点
- 行動戦略：各プレイヤーが「すべての」手番でどのような行動をとるかを表すもの
- 行動戦略における Nash 均衡：
  他のプレイヤーがその行動戦略に従うとき、
  どのような他の行動戦略をとっても、
  利得を高くできないとき

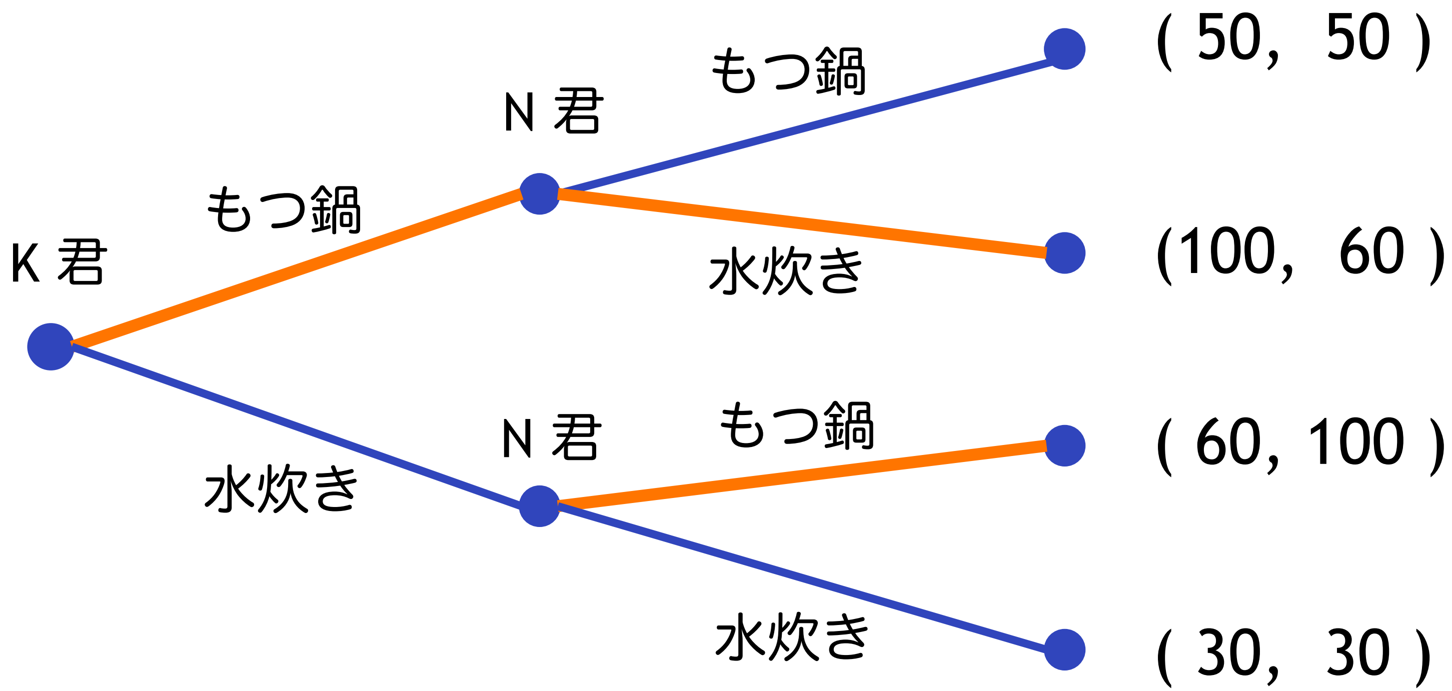
展開型ゲームにおける Nash 均衡の問題点

- K 君が N 君に遠慮しない場合



展開型ゲームにおける Nash 均衡の問題点

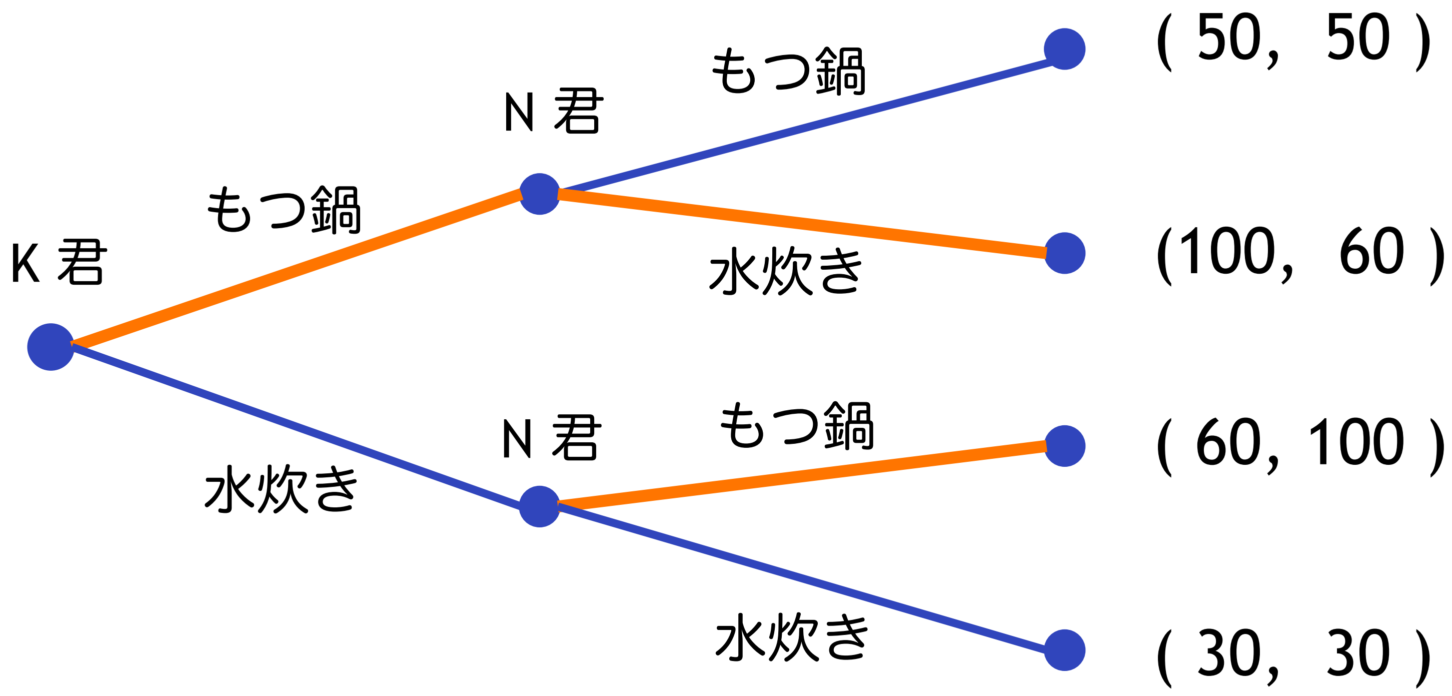
■ K 君が N 君に遠慮しない場合



(K, N) = (もつ, (水炊, もつ)) は Nash 均衡

展開型ゲームにおける Nash 均衡の問題点

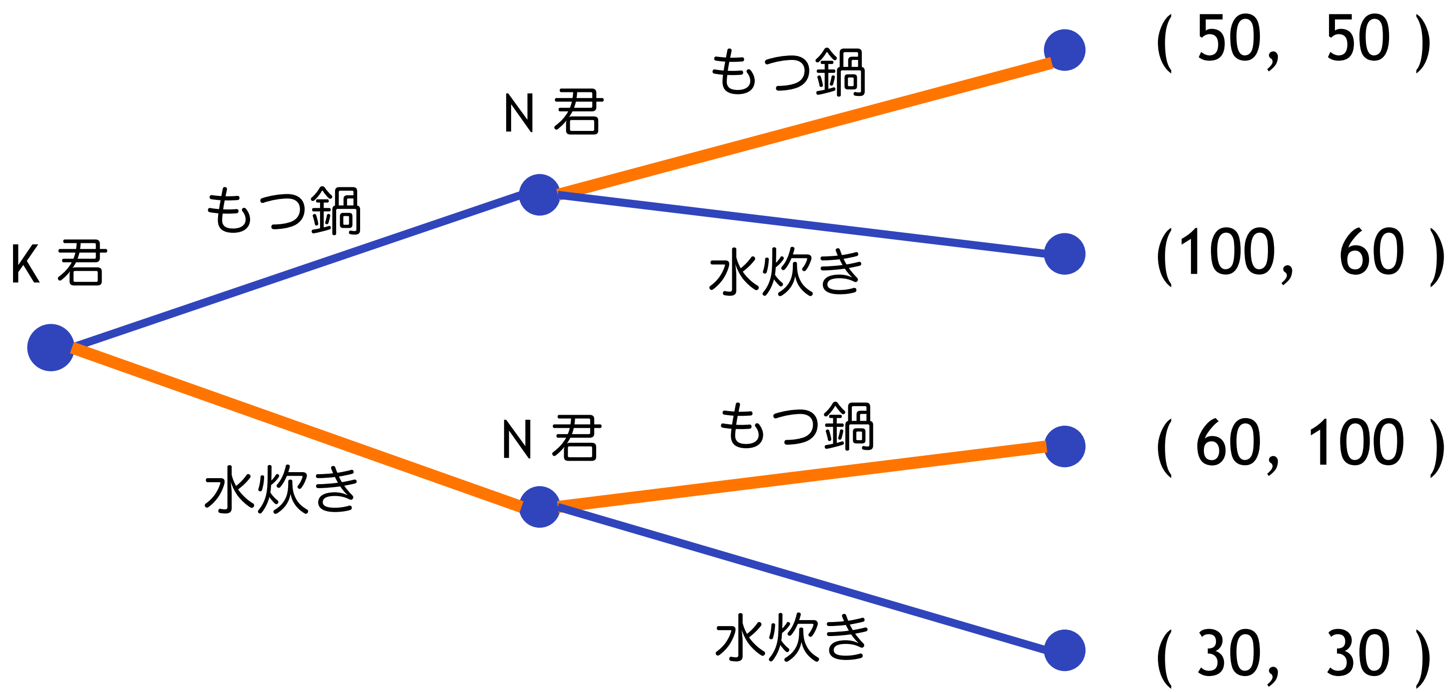
- K 君が N 君に遠慮しない場合



(K, N) = (もつ, (水炊, もつ)) は Nash 均衡

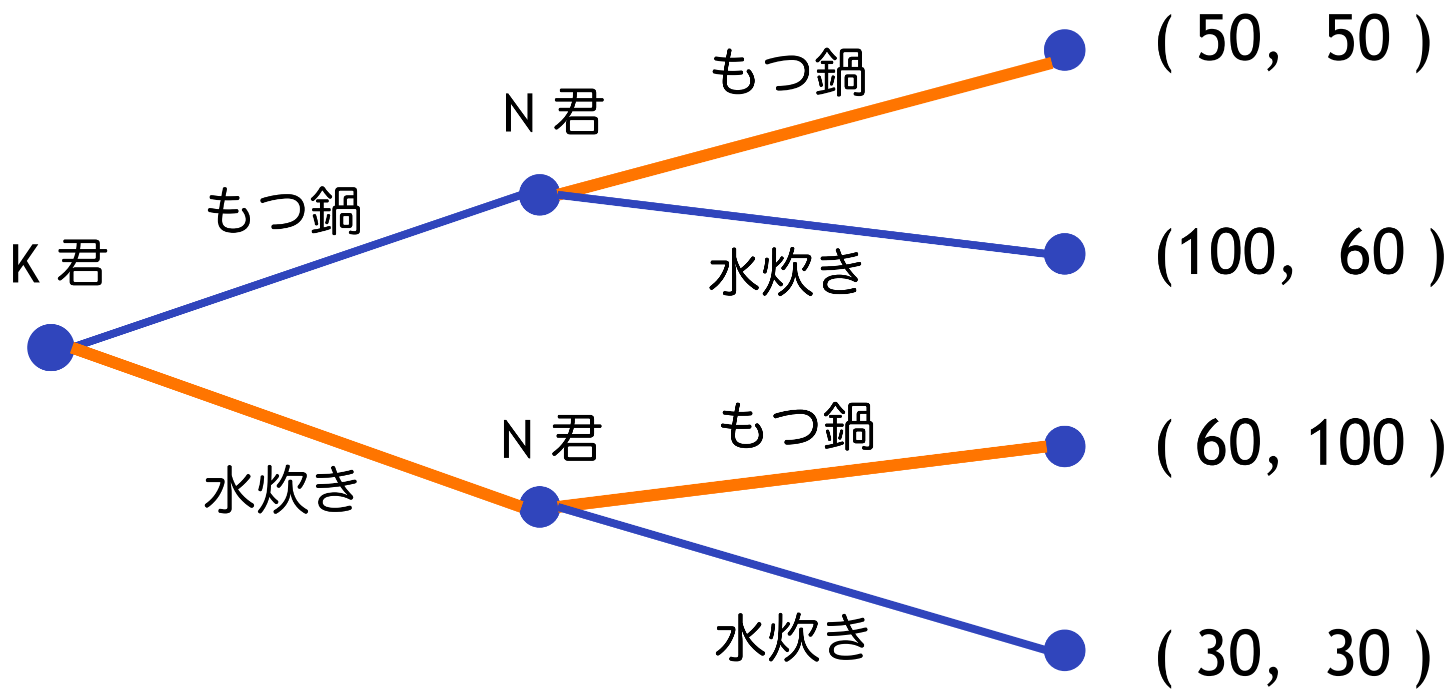
(K, N) = (もつ, (水炊, 水炊)), (水炊, (もつ, もつ)) も Nash 均衡

展開型ゲームにおける Nash 均衡の問題点



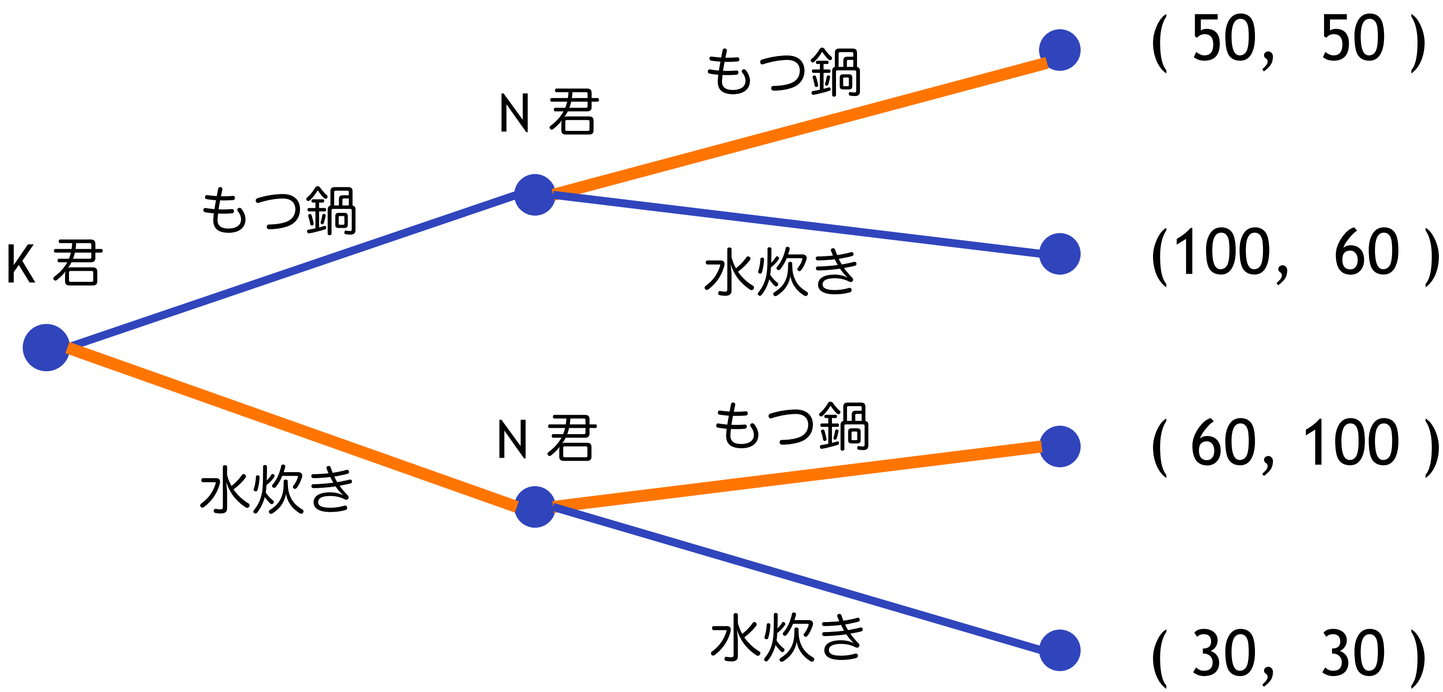
- (K, N) = (水炊, (もつ, もつ)) は Nash 均衡

展開型ゲームにおける Nash 均衡の問題点



- (K, N) = (水炊, (もつ, もつ)) は Nash 均衡
- しかし、K 君が「もつ鍋」を選んだとき、
  N 君が「もつ鍋」を選ぶとは考えにくい

展開型ゲームにおける Nash 均衡の問題点



- (K, N) = (水炊, (もつ, もつ)) は Nash 均衡
- しかし、K 君が「もつ鍋」を選んだとき、N 君が「もつ鍋」を選ぶとは考えにくい

→ 信憑性のない脅し

展開型ゲームにおける Nash 均衡の問題点

- なぜ、信憑性のない脅しが存在するのか？

展開型ゲームにおける Nash 均衡の問題点

- なぜ、信憑性のない脅しが存在するのか？

→ Nash 均衡では、
ゲーム開始前に行動戦略を決めてしまうから

展開型ゲームにおける Nash 均衡の問題点

- なぜ、信憑性のない脅しが存在するのか？

→ Nash 均衡では、
ゲーム開始前に行動戦略を決めてしまうから

- 相手の行動を観察してから行動する
  という要素が抜けている

展開型ゲームにおける Nash 均衡の問題点

- なぜ、信憑性のない脅しが存在するのか？

→ Nash 均衡では、
ゲーム開始前に行動戦略を決めてしまうから

- 相手の行動を観察してから行動する
という要素が抜けている
- 別の見方として、Nash 均衡では
実現パス以外のパスにおける均衡を考えない

展開型ゲームにおける Nash 均衡の問題点

- なぜ、信憑性のない脅しが存在するのか？

→ Nash 均衡では、
ゲーム開始前に行動戦略を決めてしまうから

- 相手の行動を観察してから行動する
という要素が抜けている
- 別の見方として、Nash 均衡では
実現パス以外のパスにおける均衡を考えない

→ 部分ゲーム完全均衡でこの問題を解決

ここまでのまとめ

- ゲーム的状況 = 複数の意思決定者が相互作用する状況
- 戦略型ゲーム
  - すべてのプレイヤーが同時に行動
  - 解の見つけ方
    1. 支配戦略を見つける
    2. 最適反応戦略を考える → Nash 均衡
  - Nash 均衡の問題点: 弱支配される可能性
- 展開型ゲーム
  - プレイヤーの行動が逐次的
  - 解の見つけ方 → 先読みをする
  - Nash 均衡の問題点: 信憑性のない脅しの可能性

暗号理論におけるゲーム理論

暗号理論 vs ゲーム理論

- ともにプレイヤー間の相互作用に関する研究

暗号理論 vs ゲーム理論

- ともにプレイヤー間の相互作用に関する研究

- 暗号理論
  - プレイヤーは正直者 or 悪者
  - 正直者をどのように守るか？

- ゲーム理論
  - プレイヤーは合理的
  - 合理的なプレイヤーはどう振る舞うか？

暗号理論とゲーム理論に関する研究

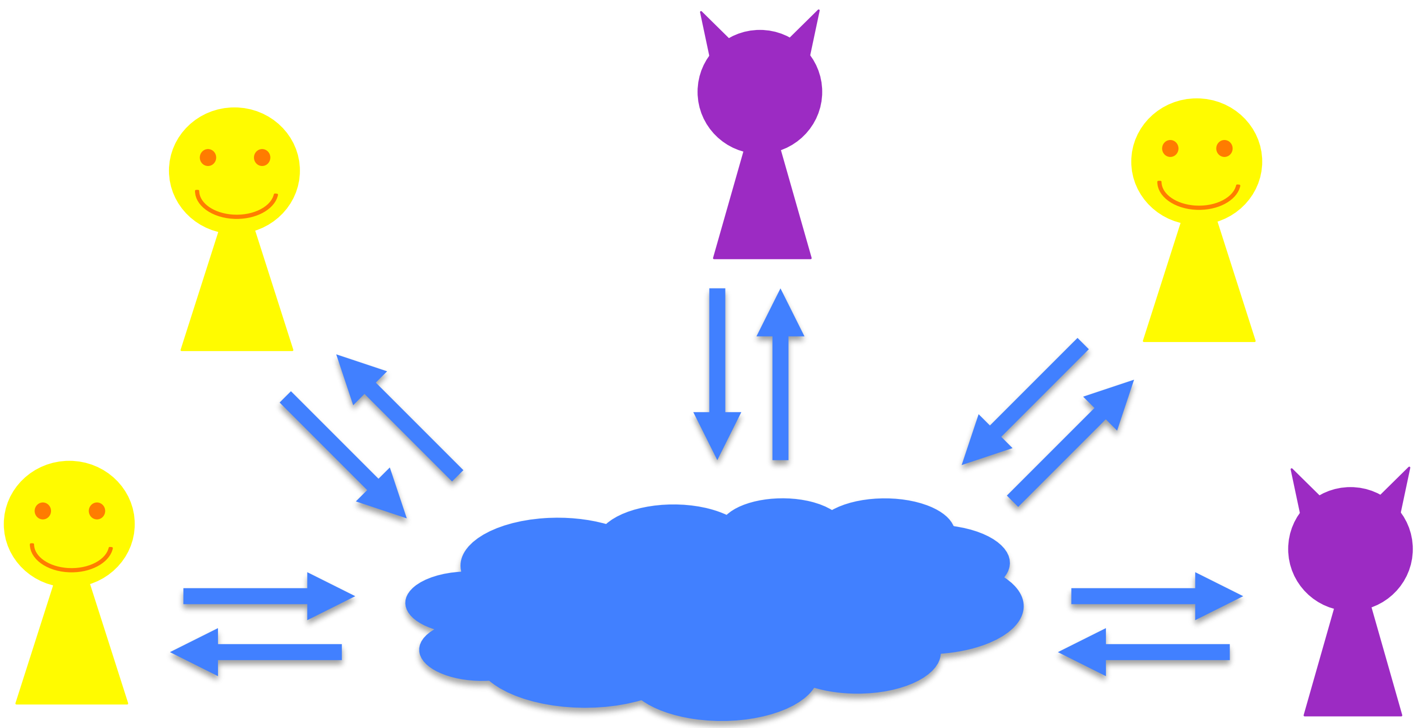
- 暗号理論をゲーム理論に利用
  - 信頼できる仲介者を暗号技術で実現 [DM00, ADGH06, LMPS04, ILM05, IML05, ASV08, ADH08, ILM08, AKL+09, ILM11, AKMZ12, CV12]
- ゲーム理論を暗号理論へ適用
  - 合理的なプレイヤーが暗号プロトコルを実行 [HT04, ADGH06, LT06, GK06, KN08a, KN08b, MS09, OPRV09, AL09, Gra10, FKN10, PS11, GKTZ12, Y12]
- ゲーム理論と暗号理論の概念間の関係
  - 暗号理論向けのゲーム理論の概念 [HP10, GLV10, PS11]
  - ゲーム理論の概念によって安全性を特徴付け [ACH11, GK12, HTYY12]

暗号理論とゲーム理論に関する研究

- 暗号理論をゲーム理論に利用
  - 信頼できる仲介者を暗号技術で実現 [DM00, ADGH06, LMPS04, ILM05, IML05, ASV08, ADH08, ILM08, AKL+09, ILM11, AKMZ12, CV12]
- ゲーム理論を暗号理論へ適用
  - 合理的なプレイヤーが暗号プロトコルを実行 [HT04, ADGH06, LT06, GK06, KN08a, KN08b, MS09, OPRV09, AL09, Gra10, FKN10, PS11, GKTZ12, Y12]
- ゲーム理論と暗号理論の概念間の関係
  - 暗号理論向けのゲーム理論の概念 [HP10, GLV10, PS11]
  - ゲーム理論の概念によって安全性を特徴付け [ACH11, GK12, HTYY12]

暗号プロトコル

- 正直者と悪者が存在
- 悪者がいたとしても、プロトコルに従えば、正直者は目的を達成できる



プレイヤーに合理性があると・・・

プレイヤーに合理性があると・・・

- 正直者は、いつもプロトコルに従うと仮定

プレイヤーに合理性があると・・・

- 正直者は、いつもプロトコルに従うと仮定
  - 自分の利益のためなら、
    プロトコルに従わないかもしれない

プレイヤーに合理性があると・・・

- 正直者は、いつもプロトコルに従うと仮定
  - 自分の利益のためなら、
    プロトコルに従わないかもしれない

→ より強力な暗号プロトコルの必要性

例. 秘密分散における合理的なプレイヤー [HT04]

プレイヤーに合理性があると・・・

- 正直者は、いつもプロトコルに従うと仮定
  - 自分の利益のためなら、
    プロトコルに従わないかもしれない

  → より強力な暗号プロトコルの必要性

  例. 秘密分散における合理的なプレイヤー [HT04]

- 悪者は、可能な限り正直者の邪魔をすると仮定
  - 別の目的をもって攻撃しているかもしれない

プレイヤーに合理性があると・・・

- 正直者は、いつもプロトコルに従うと仮定
  - 自分の利益のためなら、
    プロトコルに従わないかもしれない

  → より強力な暗号プロトコルの必要性

  例. 秘密分散における合理的なプレイヤー [HT04]

- 悪者は、可能な限り正直者の邪魔をすると仮定
  - 別の目的をもって攻撃しているかもしれない

  → より効率的/既存の不可能性を回避した
  暗号プロトコルの可能性

  例. ビザンチン合意における合理的な敵 [GKTZ12]

ゲーム理論を応用する際の難しさ

- 計算能力の制限されたプレイヤー
  - ゲームの例（一方向性置換ゲーム）
    1. P_1 が $x \in \{0,1\}^n$ をランダムに選び $f(x)$ を P_2 に送る
    2. P_2 が $z \in \{0,1\}^n$ を P_1 に送る
    3. P_2 は $z = x$ のときに利得 1, それ以外で - 1
- 漸近的な議論
- （無視できる程度の）誤り確率

秘密分散とゲーム理論

秘密分散

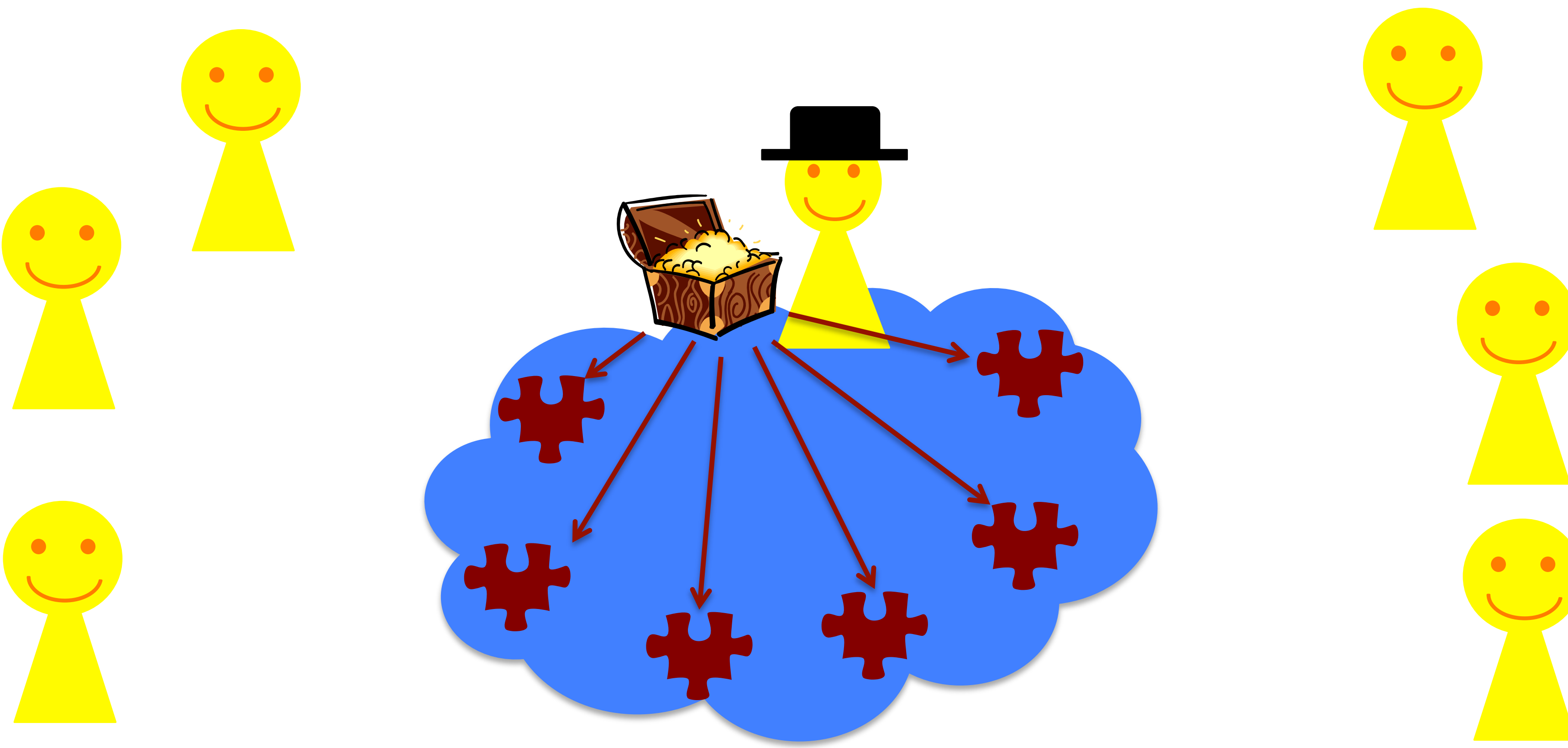
- 参加者： ディーラー１人とプレイヤー n 人

秘密分散

- 分散フェーズ：
  ディーラーは、秘密からシェアを作り、
  各プレイヤーにを配る

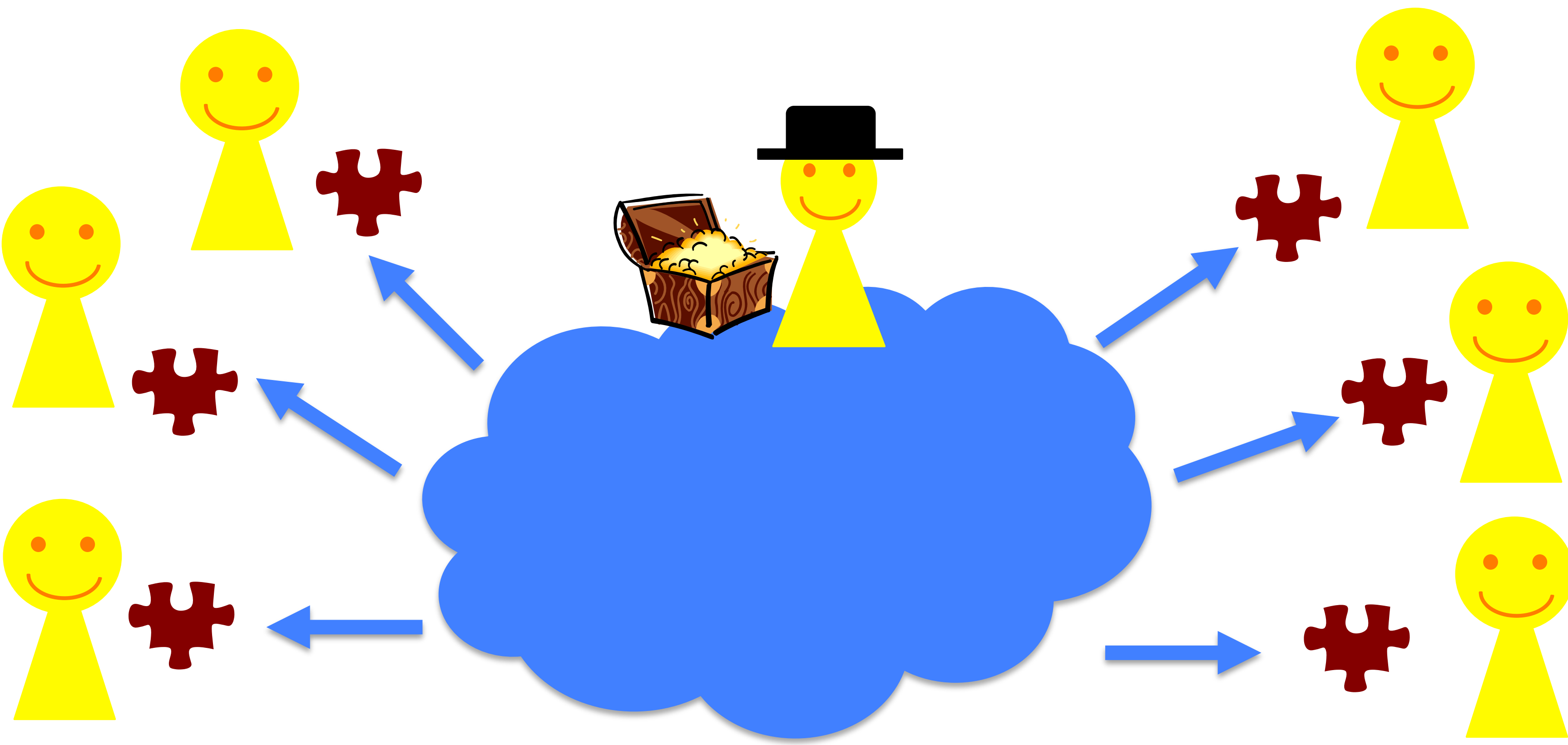
秘密分散

- 分散フェーズ：
  ディーラーは、秘密からシェアを作り、
  各プレイヤーにを配る



秘密分散

- 分散フェーズ：
  ディーラーは、秘密からシェアを作り、
  各プレイヤーにを配る



秘密分散

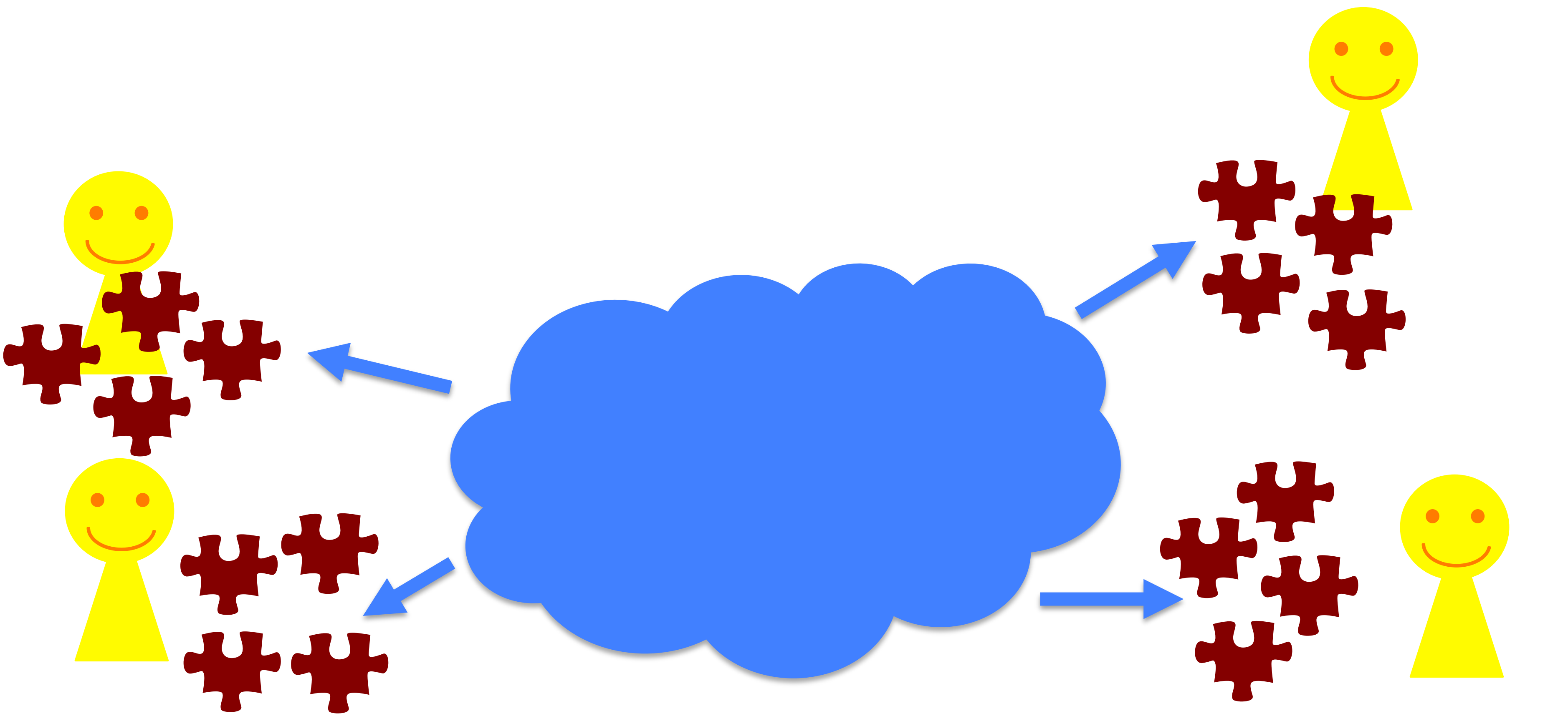
- 復元フェーズ：
一定人数のプレイヤーがそろったとき、
シェアを出し合うことで秘密を復元

秘密分散

- 復元フェーズ：
  一定人数のプレイヤーがそろったとき、
  シェアを出し合うことで秘密を復元

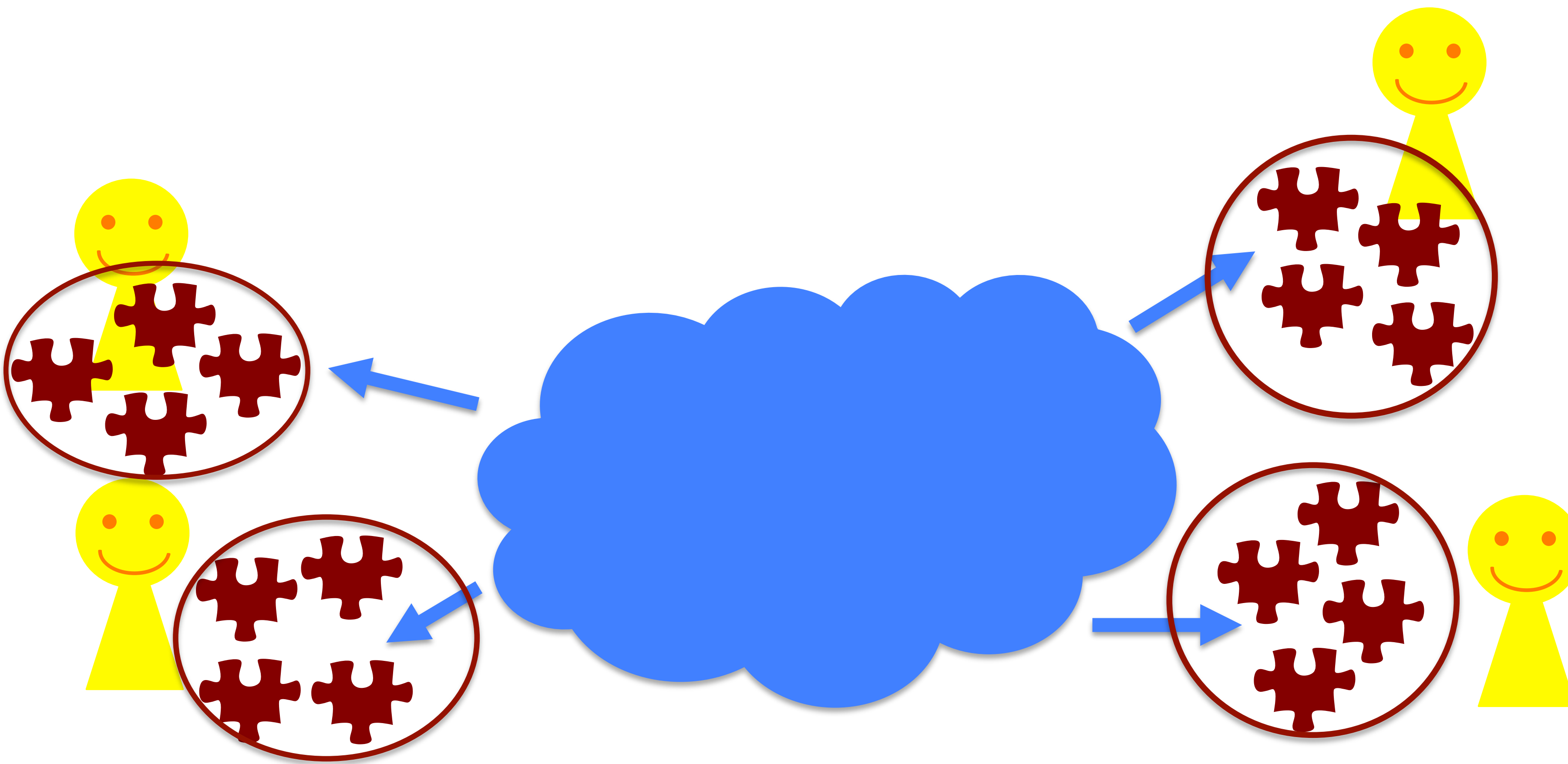
秘密分散

- 復元フェーズ：
一定人数のプレイヤーがそろったとき、
シェアを出し合うことで秘密を復元



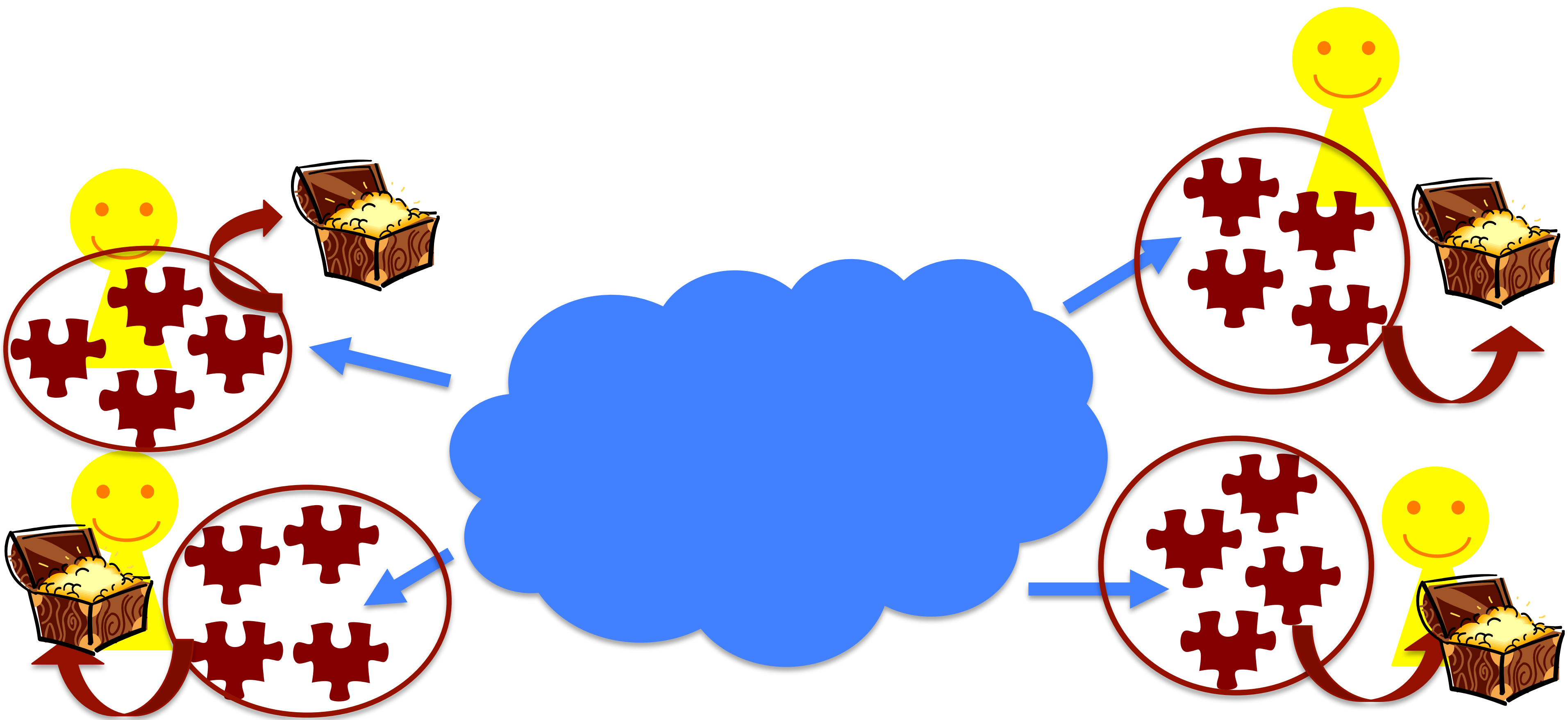
秘密分散

- 復元フェーズ：
  一定人数のプレイヤーがそろったとき、
  シェアを出し合うことで秘密を復元



秘密分散

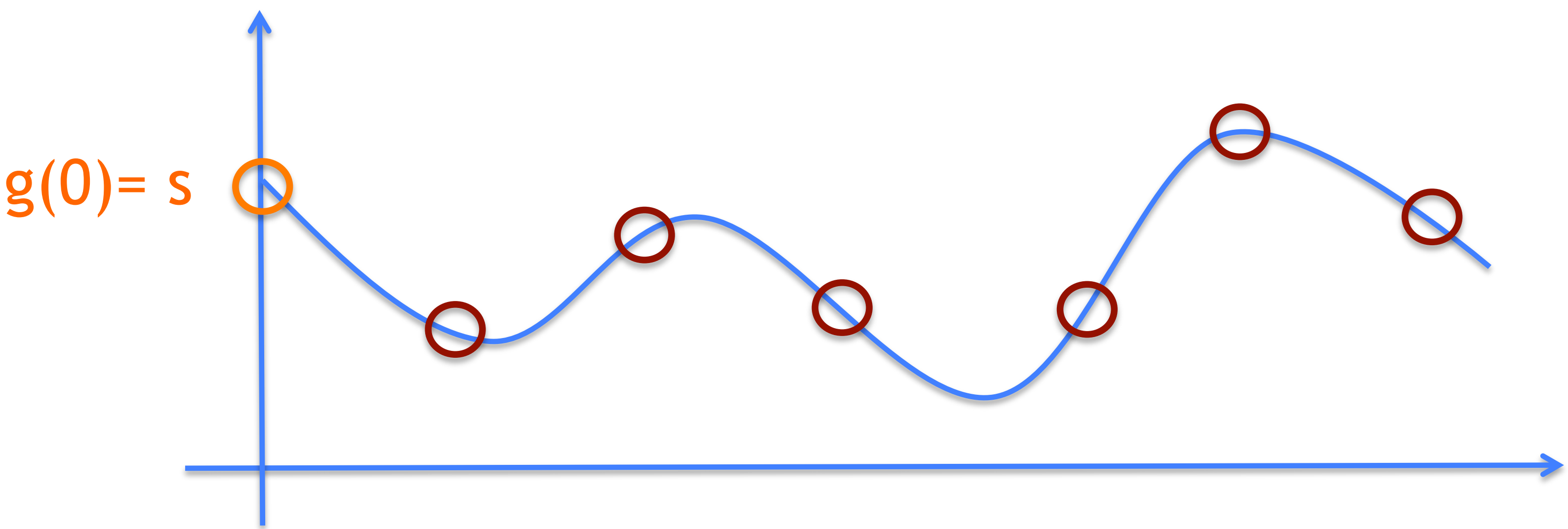
- 復元フェーズ：
一定人数のプレイヤーがそろったとき、
シェアを出し合うことで秘密を復元



- (m, n) しきい値型秘密分散
  m 個以上のシェアから秘密を復元でき、
  m 個未満では秘密についてわからない

- (m, n) しきい値型秘密分散
  m 個以上のシェアから秘密を復元でき、
  m 個未満では秘密についてわからない

- Shamir の秘密分散
  ランダム (m - 1) 次多項式 g s.t. g(0)= s を選び、
  g(1), ..., g(n) をシェアとし、多項式補間で復元



[Halpern, Teague 2004]

[Halpern, Teague 2004]

- プレイヤーの利得

1. 秘密を復元したい
2. より少ない人数で復元したい

[Halpern, Teague 2004]

- プレイヤーの利得

1. 秘密を復元したい
2. より少ない人数で復元したい

Shamir の秘密分散プロトコルは
正しく実行されない

Shamir の (m, n) 秘密分散の問題点

Shamir の (m, n) 秘密分散の問題点

- 復元フェーズで、
  全員がシェアを出すという戦略がよくない

Shamir の (m, n) 秘密分散の問題点

- 復元フェーズで、
全員がシェアを出すという戦略がよくない

- 認証つき秘密分散を仮定すると
プレイヤーの選択肢は実質的に２つ
  - シェアを「出す」
  - シェアを「出さない」

Shamir の (m, n) 秘密分散の問題点

- m = n のとき

- m < n のとき

Shamir の (m, n) 秘密分散の問題点

- m = n のとき
  - 「出す」　→ n 人で復元
  - 「出さない」　→ 1 人で復元
- m < n のとき

Shamir の (m, n) 秘密分散の問題点

- m = n のとき
  - 「出す」　→　n 人で復元
  - 「出さない」　→　1 人で復元

⇒ Nash 均衡ではない

- m < n のとき

Shamir の (m, n) 秘密分散の問題点

- m = n のとき
  - 「出す」　→ n 人で復元
  - 「出さない」　→ 1 人で復元

➡ Nash 均衡ではない

- m < n のとき
  - シェアを出しても出さなくても n 人で復元
  - 「出さない」が「出す」より悪い状況はなく、また、ある状況では真に良い

Shamir の (m, n) 秘密分散の問題点

- m = n のとき
  - 「出す」 → n 人で復元
  - 「出さない」 → 1 人で復元

➡ Nash 均衡ではない

- m < n のとき
  - シェアを出しても出さなくても n 人で復元
  - 「出さない」が「出す」より悪い状況はなく、また、ある状況では真に良い

➡ 弱支配される Nash 均衡

[Gordon, Katz 08] のプロトコル

[Gordon, Katz 08] のプロトコル

- (2, 2) 秘密分散の場合を考える

[Gordon, Katz 08] のプロトコル

- (2, 2) 秘密分散の場合を考える

- プレイヤー P_i の利得
  - P_i だけが復元 → U^+
  - ２人とも復元 → U
  - どちらも復元しない → U^-
  - $U^+ > U > U^-$

GK08 プロトコルのアイディア

- ディーラーは P_1, P_2 それぞれに、
無限個のシェア $(a_1, a_2, \ldots)$, $(b_1, b_2, \ldots)$を用意
  - 各 i について（独立に）
    - 確率 δ で $a_i + b_i = s$（本物の秘密）
    - 確率 1 – δ で $a_i + b_i = \perp$（偽物）
- 各ラウンド i において
  - 両プレイヤーはシェア a_i, b_i を同時に出す
  - $a_i + b_i = s$ なら終了
  - $a_i + b_i = \perp$ なら次のラウンドへ
  - もし一人がシェアを出さなかったら終了

GK08 プロトコルの分析

GK08 プロトコルの分析

- P_1 が逸脱することを考える
  - Nash 均衡を考えるので P_2 は従うと仮定

GK08 プロトコルの分析

- P_1 が逸脱することを考える
  - Nash 均衡を考えるので P_2 は従うと仮定
- P_1 がシェアを出さないとき、
  P_1 は確率 δ で U^+ を、確率 1 – δ で U^- を得る
  → 期待利得は $\delta U^+ + (1 - \delta) U^-$

GK08 プロトコルの分析

- P_1 が逸脱することを考える
  - Nash 均衡を考えるので P_2 は従うと仮定
- P_1 がシェアを出さないとき、
  P_1 は確率 δ で U^+ を、確率 1 – δ で U^- を得る
  → 期待利得は $\delta U^+ + (1 - \delta) U^-$
- P_1 がシェアを出すとき、利得は U

GK08 プロトコルの分析

- P_1 が逸脱することを考える
  - Nash 均衡を考えるので P_2 は従うと仮定
- P_1 がシェアを出さないとき、
  P_1 は確率 δ で U^+ を、確率 1 – δ で U^- を得る
  → 期待利得は $\delta U^+ + (1 - \delta) U^-$
- P_1 がシェアを出すとき、利得は U
- ここで、$\delta U^+ + (1 - \delta) U^- < U$ ならば
  シェアを出すことは、弱支配ではない

GK08 プロトコルの分析

- P_1 が逸脱することを考える
  - Nash 均衡を考えるので P_2 は従うと仮定
- P_1 がシェアを出さないとき、
  P_1 は確率 δ で U^+ を、確率 $1 - \delta$ で U^- を得る
  → 期待利得は $\delta U^+ + (1 - \delta) U^-$
- P_1 がシェアを出すとき、利得は U
- ここで、$\delta U^+ + (1 - \delta) U^- < U$ ならば
  シェアを出すことは、弱支配ではない
- ただし、同時にシェアを出すことに強く依存

実際のプロトコル

- 無限個のシェアを用意することはできない
- ディーラーは a + b = s となるシェアを用意
- 各ラウンド i において
  - P_1 と P_2 は安全なプロトコル(MPC)を利用して a_i と b_i を a と b から生成
  - 残りは同様

[Fuchsbauer, Katz, Naccache 2010] プロトコル

[Fuchsbauer, Katz, Naccache 2010] プロトコル

- GK08 等のプロトコルはシェアを同時に出すことを必要

  → 同時ブロードキャスト通信路を仮定

[Fuchsbauer, Katz, Naccache 2010] プロトコル

- GK08 等のプロトコルはシェアを同時に出すことを必要

  → 同時ブロードキャスト通信路を仮定

- GK08 は MPC を毎ラウンド計算
  - 計算効率はよくない

[Fuchsbauer, Katz, Naccache 2010] プロトコル

- GK08 等のプロトコルはシェアを同時に出すことを必要

  → 同時ブロードキャスト通信路を仮定

- GK08 は MPC を毎ラウンド計算
  - 計算効率はよくない

- FKN10 では上記の問題点を解決し、かつ強い解概念をもつプロトコルを提案

FKN10 プロトコルのアイディア

FKN10 プロトコルのアイディア

- 基本アイディアは同じ：
  - 本物ラウンドと偽物ラウンドが存在
  - 本物である確率が十分小さいので、
プレイヤーは正しくシェアを出し続ける

FKN10 プロトコルのアイディア

- 基本アイディアは同じ：
  - 本物ラウンドと偽物ラウンドが存在
  - 本物である確率が十分小さいので、
プレイヤーは正しくシェアを出し続ける

- 既存プロトコルと異なる点：
  - 既存：本物ラウンドであるかをすぐに認識
  - FKN10：本物ラウンドであるかは後で認識

FKN10 プロトコルのアイディア

- 基本アイディアは同じ：
  - 本物ラウンドと偽物ラウンドが存在
  - 本物である確率が十分小さいので、プレイヤーは正しくシェアを出し続ける

- 既存プロトコルと異なる点：
  - 既存：本物ラウンドであるかをすぐに認識
  - FKN10：本物ラウンドであるかは後で認識

- 検証可能ランダム関数 (VRF) を利用
  - 擬似ランダム関数であり、正しさを証明で検証可能。また、証明は１つしか存在しない

FKN10 プロトコル

- ディーラーは
  - 本物ラウンド r* を選ぶ（幾何分布に従う）
  - VRF の鍵を２種類生成：(pk_i, sk_i), (pk_i', sk_i'), $i \in \{1,2\}$
  - P_1 に以下のシェアを渡す（P_2 も同様）
    $(sk_1, sk_1', pk_2, pk_2', shr_1 = F_{sk2}(r^*) + s, sig_1 = F_{sk2'}(r^*+1))$

- 各ラウンド r において（P_1 の立場）
  - $F_{sk1}(r)$, $F_{sk1'}(r)$ とその証明を送る
  - $y^{(r)}$ と $z^{(r)}$ を受け取ったとき
    - $sig_1 = z^{(r)}$ なら $s^{(r-1)} = shr_1 + y^{(r-1)}$ を出力して終了
    - 相手が離脱 or 偽証明を送ったら $s^{(r-1)}$ を出力し終了
    - それ以外の場合、次のラウンドへ

FKN10 プロトコルの分析

FKN10 プロトコルの分析

- P_2 が従い、P_1 が逸脱することを考える

FKN10 プロトコルの分析

- P_2 が従い、P_1 が逸脱することを考える
- 逸脱はラウンド $r = r^* + 1$ または $r < r^* + 1$ で可能

FKN10 プロトコルの分析

- P_2 が従い、P_1 が逸脱することを考える
- 逸脱はラウンド $r = r^* + 1$ または $r < r^* + 1$ で可能
  - $r = r^* + 1$ で逸脱
    → P_2 も s を出力するので利得は U のまま

FKN10 プロトコルの分析

- P_2 が従い、P_1 が逸脱することを考える
- 逸脱はラウンド $r = r^* + 1$ または $r < r^* + 1$ で可能
  - $r = r^* + 1$ で逸脱
    → P_2 も s を出力するので利得は U のまま
  - $r < r^* + 1$ で逸脱
    → $r = r^*$ であれば利得は U^+ の可能性があるが、
    本物ラウンドの確率は十分小さく、
    期待利得は U より小さい（ように設定）

FKN10 プロトコルの分析

- P_2 が従い、P_1 が逸脱することを考える
- 逸脱はラウンド $r = r^* + 1$ または $r < r^* + 1$ で可能
  - $r = r^* + 1$ で逸脱
    → P_2 も s を出力するので利得は U のまま
  - $r < r^* + 1$ で逸脱
    → $r = r^*$ であれば利得は U^+ の可能性があるが、
    本物ラウンドの確率は十分小さく、
    期待利得は U より小さい（ように設定）
- $r = r^* + 1$ での逸脱はプロトコル終了の印であり、
  逸脱でないとみなすと、逸脱は真に利得を下げる

FKN10 プロトコルの分析

- P_2 が従い、P_1 が逸脱することを考える
- 逸脱はラウンド $r = r^* + 1$ または $r < r^* + 1$ で可能
  - $r = r^* + 1$ で逸脱
    → P_2 も s を出力するので利得は U のまま
  - $r < r^* + 1$ で逸脱
    → $r = r^*$ であれば利得は U^+ の可能性があるが、
    本物ラウンドの確率は十分小さく、
    期待利得は U より小さい（ように設定）
- $r = r^* + 1$ での逸脱はプロトコル終了の印であり、
  逸脱でないとみなすと、逸脱は真に利得を下げる
  → 狭義 Nash 均衡（強い解概念）

FKN10 プロトコルの特徴

FKN10 プロトコルの特徴

- 同時ブロードキャスト通信路を必要としない
  - P2P ネットワークで十分
- 計算効率がよい
  - VRF の部分は TDP で実現可能

FKN10 プロトコルの特徴

- 同時ブロードキャスト通信路を必要としない
  - P2P ネットワークで十分
- 計算効率がよい
  - VRF の部分は TDP で実現可能

- 秘密を見て秘密であることが確信できると問題
  - 秘密がパスワードで、正しさの確認ができる場合
  - この問題は非同時ブロードキャスト通信路では避けられない [Asharov, Lindell 2010]

まとめ（秘密分散とゲーム理論）

- 正直者に合理性を仮定すると
プロトコルの実現がとても大変になった例
  - 秘密の復元を独占したいと考えるプレイヤーばかりだと、公平に復元することが大変
- 暗号理論として達成が困難（？）
  - 多くのプロトコルで同時ブロードキャスト
  - 非同時ブロードキャストだと
秘密自体にエントロピーが必要

→ 妥当な仮定等をおいて簡単に実現できないか

ビザンチン合意とゲーム理論

ビザンチン合意問題

- 分散計算・暗号理論の代表的な問題
- Lamport, Shostak, Pease が導入 (1980/1982)
- 故障プロセッサが存在する場合の分散計算問題

ビザンチン合意問題の由来

ビザンチン合意問題の由来

- ビザンチン帝国軍の将軍たちが、
  軍隊を率いて敵の都市を囲っている状況

ビザンチン合意問題の由来

- ビザンチン帝国軍の将軍たちが、
  軍隊を率いて敵の都市を囲っている状況
- 将軍たちは離れた場所にいるため、
  使者を使ってメッセージを伝えあう

ビザンチン合意問題の由来

- ビザンチン帝国軍の将軍たちが、
  軍隊を率いて敵の都市を囲っている状況
- 将軍たちは離れた場所にいるため、
  使者を使ってメッセージを伝えあう
- 将軍たちは、攻撃するのか撤退するのか、
  ひとつの計画に同意したい

ビザンチン合意問題の由来

- ビザンチン帝国軍の将軍たちが、
軍隊を率いて敵の都市を囲っている状況
- 将軍たちは離れた場所にいるため、
使者を使ってメッセージを伝えあう
- 将軍たちは、攻撃するのか撤退するのか、
ひとつの計画に同意したい
- 将軍たちの中に反逆者がいるかもしれない
  - それが誰なのかはわからない

ビザンチン合意問題の由来

- ビザンチン帝国軍の将軍たちが、
軍隊を率いて敵の都市を囲っている状況
- 将軍たちは離れた場所にいるため、
使者を使ってメッセージを伝えあう
- 将軍たちは、攻撃するのか撤退するのか、
ひとつの計画に同意したい
- 将軍たちの中に反逆者がいるかもしれない
  - それが誰なのかはわからない
- （反逆者でない）帝国軍の将軍たちが同じ計画を選択する場合、その計画で同意したい

ビザンチン合意プロトコル

- n 人のプレイヤー $P_1, \ldots, P_n$ が存在
- 各プレイヤー P_i は入力 $v_i \in \{0,1\}$ をもつ
- 敵は n 人のうち t 人までを任意にコントロール
  - 残りの n - t 人のプレイヤーを正直者と呼ぶ
- このときプロトコル実行後に以下を満たすこと

1. すべての正直者は同じ値 w を出力
2. すべての正直者の入力が同じ値のとき、
   その値を w として出力

ブロードキャストプロトコル

- n 人のプレイヤー $P_1, \ldots, P_n$ が存在
- （送信者）P_i が入力 $v \in \{0,1\}$ をもつ
- 敵は n 人のうち t 人までを任意にコントロール
- このときプロトコル実行後に以下を満たすこと
  1. すべての正直者が同じ値 w を出力
  2. P_i が正直者だった場合、v を w として出力

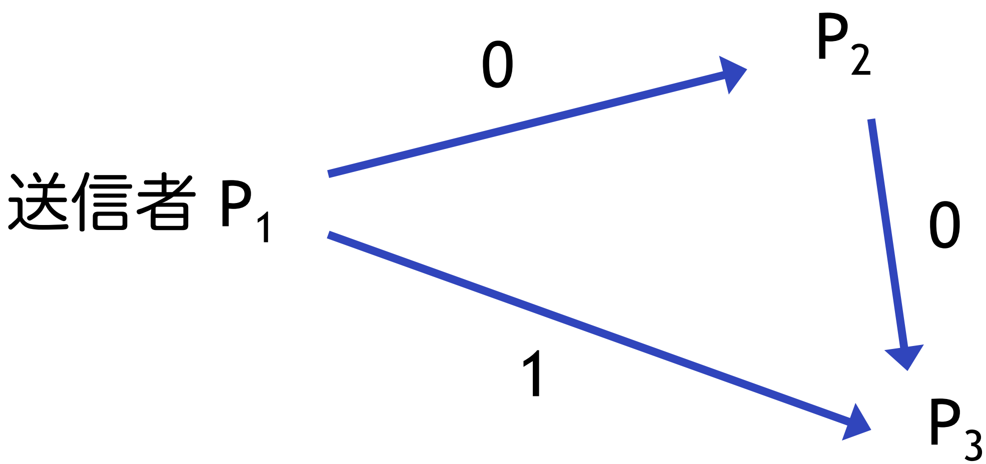
ビザンチン合意とブロードキャストの等価性（ $t < n/2$ の場合）

- ブロードキャストを使ったビザンチン合意
  - 各プレイヤーは自分の入力をブロードキャスト
  - 過半数の値を出力

- ビザンチン合意を使ったブロードキャスト
  - 送信者は自分の入力を他のプレイヤーに送信
  - 各プレイヤーは受信した値でビザンチン合意
  - ビザンチン合意の結果を出力

ビザンチン合意の可能性・不可能性

- ビザンチン合意が可能 ⇔ $t < n/3$
  - $n = 3$, $t = 1$, 送信者が敵対者の場合



P_3 は
P_1 と P_2 のどちらが
正直者かわからない

- 公開鍵暗号系（署名）の存在を仮定すると、任意の $t < n$ でビザンチン合意可能

ビザンチン合意の可能性・不可能性

- $t \geq n/2$ の場合は
  ブロードキャスト通信路を仮定しても不可能

ビザンチン合意の可能性・不可能性

- $t \geq n/2$ の場合は
ブロードキャスト通信路を仮定しても不可能
- 証明：$P_1, \ldots, P_{n/2}$ の入力は 1, それ以外 0 とする

A. 敵が $P_1, \ldots, P_{n/2}$ をコントロール
→ 正直者は 0 を出力

B. 敵が $P_{n/2+1}, \ldots, P_n$ をコントロール
→ 正直者は 1 を出力

C. 敵が誰もコントロールしなかった
→ $P_1, \ldots, P_{n/2}$ は B と区別できず 1 を出力
$P_{n/2+1}, \ldots, P_n$ は A と区別できず 0 を出力
→ 矛盾

ビザンチン合意における合理的な敵
[Groce, Katz, Thiruvengadam, Zikas 2012]

ビザンチン合意における合理的な敵
[Groce, Katz, Thiruvengadam, Zikas 2012]

- 合理的な敵がプレイヤーをコントロール
  - コントロールされないプレイヤーは正直者
    → 合理的な敵１人によるゲーム

ビザンチン合意における合理的な敵
[Groce, Katz, Thiruvengadam, Zikas 2012]

- 合理的な敵がプレイヤーをコントロール
  - コントロールされないプレイヤーは正直者 → 合理的な敵１人によるゲーム
- 敵の利得
  - 0 で合意したとき → u_0
  - 1 で合意したとき → u_1
  - 合意しなかったとき → u_2
  - u_0, u_1, u_2 は異なる実数値と仮定

ビザンチン合意における合理的な敵 [Groce, Katz, Thiruvengadam, Zikas 2012]

- 合理的な敵がプレイヤーをコントロール
  - コントロールされないプレイヤーは正直者 → 合理的な敵１人によるゲーム
- 敵の利得
  - 0 で合意したとき → u_0
  - 1 で合意したとき → u_1
  - 合意しなかったとき → u_2
  - u_0, u_1, u_2 は異なる実数値と仮定
- 敵は正直者の入力値を知っていると仮定

安全性の定義

- 敵は利得関数 U をもち、
  n 人中 t 人までをコントロール
- ビザンチン合意（ブロードキャスト）プロトコルが安全であるとは、任意の敵に対して、ある戦略 S が存在し、以下を満たすこと
  1. S を実行して生じる最終出力分布 D において、安全性は保たれている
  2. 任意の S’ ≠ S を実行して生じる出力分布 D’ に対して、U(D) ≥ U(D’)

既存の不可能性を回避

- ブロードキャスト通信路を仮定したとき、任意の $t < n$ で合理的な敵に対して安全なビザンチン合意プロトコルが存在
  - 利得は $u_2 > u_1 > u_0$ を満たしていると仮定

既存の不可能性を回避

- ブロードキャスト通信路を仮定したとき、任意の $t < n$ で合理的な敵に対して安全なビザンチン合意プロトコルが存在
  - 利得は $u_2 > u_1 > u_0$ を満たしていると仮定
- プロトコル
  1. 各プレイヤーは自分の入力をブロードキャスト
  2. 全プレイヤーが同じ値ならその値を、そうでないときは 0 を出力
- 証明：敵にとって 0 を出力されるくらいならなるべく全員 1 を出力するように振舞った方がよい

利得に関する知識を仮定したプロトコル

- 任意の $t < n$ で、合理的な敵に対して安全なビザンチン合意プロトコルが存在
  - 利得 u_0, u_1, u_2 は知られていると仮定

利得に関する知識を仮定したプロトコル

- 任意の $t < n$ で、合理的な敵に対して安全なビザンチン合意プロトコルが存在
  - 利得 u_0, u_1, u_2 は知られていると仮定

- 証明
  - アイディア：
    先ほどのプロトコルと同様、
    敵が安全性を破ろうとすると
    敵に罰が与えられる仕組みを作る

証明の続き

- u_2 が最大値でないとき
  - プロトコル：
    1. 各プレイヤー P_i は入力値 v_i をすべての P_j に送る
    2. 各 P_j は、すべて同じ値を受け取ったらその値を、そうでないとき、敵が最も好まない値 b’ を出力
  - 安全である理由：
    正直者の入力値が同じときは
    敵はそれを破る動機がなく、
    入力値が異なるときは b’ が出力される

証明の続き

- u_2 が最大値のとき
  - プロトコル：
    1. 各プレイヤー P_i は検出可能ブロードキャストを使って入力値 v_i をブロードキャスト
    2. 離脱もしくは受け取った値の不一致がある場合、敵が最も好まない値 b' を、そうでない場合、全プレイヤーから送られた値を出力
    - 検出可能ブロードキャスト：正直者は離脱 or 受理し 0 or 1 を出力．離脱なしなら、ブロードキャストとして安全．離脱ありなら、敵は送信者の入力はわからない
  - 安全である理由：敵は、正直者に異なる値を出力させることはできない．そして、1 – b' を出力させるように振舞ったほうが良い

まとめ（ビザンチン合意とゲーム理論）

- 敵に合理性を仮定することで
  既存の不可能性を回避できた例

- 敵の合理性に関する知識は少ないほうがよい
  - u_2 が最大という知識だけ
    → 既存の不可能性が適用（$t > n/3$ は不可能）
  - u_2 が最小という知識だけ
    → $t < n/2$ ⇔ 安全なビザンチン合意が存在
    → $t < n$ で安全なブロードキャストが存在

→ 合理性を仮定して不可能性を回避 or 効率改善
となる他の例はないか？

まとめ

- ゲーム理論とは
  - 戦略型ゲーム・展開型ゲーム
  - ゲームの解とその見つけ方
  - 解概念：支配戦略・最適反応戦略・Nash 均衡
  - Nash 均衡の問題点

- 暗号理論におけるゲーム理論
  - 秘密分散とゲーム理論
  - ビザンチン合意とゲーム理論