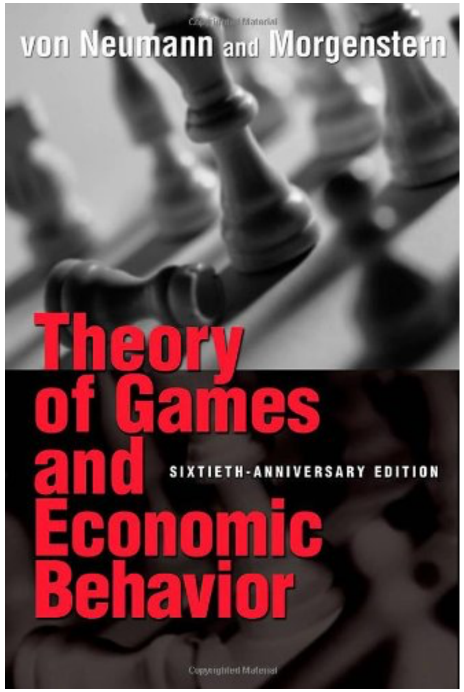
暗号技術に対する
ゲーム理論的なアプローチ

安永 憲司（金沢大学）

2017.12.22
第9回 暗号及び情報セキュリティと数学の相関ワークショップ (CRISMATH 2017)

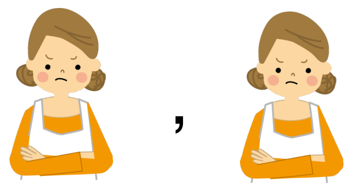
ゲーム理論

- 戦略的状況における意思決定のための理論
  - 戦略的状況：自身の損得が他者の行動に依存
  - 相手がどのように行動するかを考える必要



ゲームの例（囚人のジレンマ）

- Ａちゃんと B 君がお父さんの部屋で遊んでいて、クリスマスプレゼントのような箱の袋を破ってしまった。遊んではダメなのに。
- お母さんから「遊んでいたの？正直に言えば怒らないから」と別々の部屋で聞かれた場合、どう答えるべきか
  - ２人とも嘘をつく → お母さんイライラ
  - ２人とも認める → 少し怒られる
  - １人が認め、１人が嘘をつく → 嘘をついた方はすごく怒られる（認めた方は怒られない）

Ａ＼Ｂ	嘘をつく	認める
嘘をつく	()	()
認める	()	()

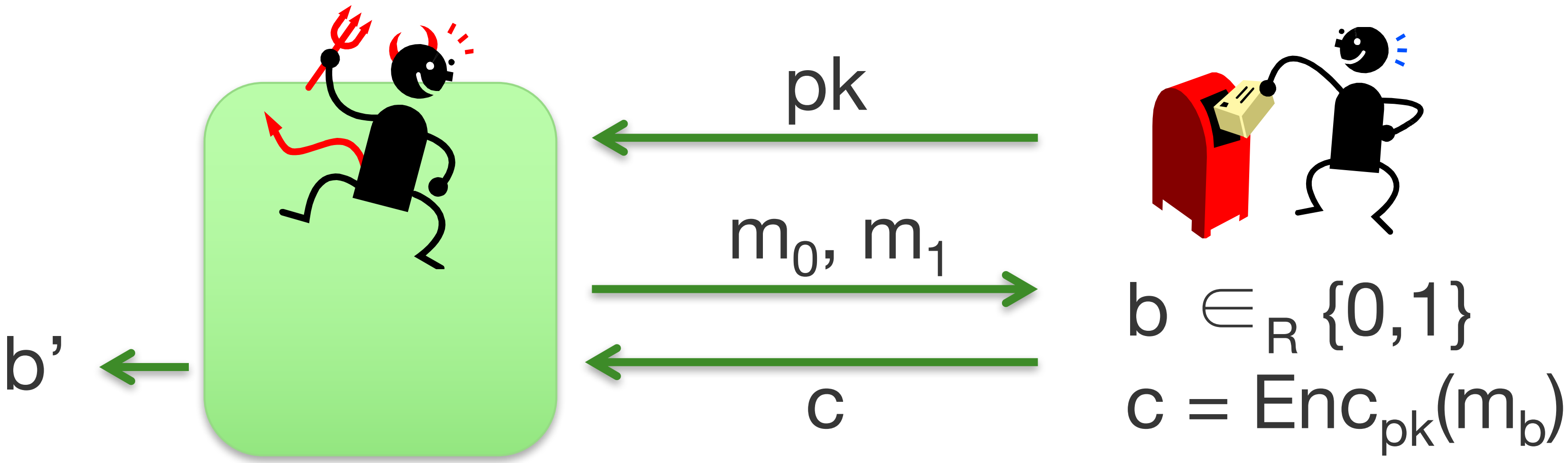
ゲームの定式化

- プレイヤー集合：意思決定を行う主体
- 行動（戦略）：プレイヤーのとりうる選択肢
- 利得（効用）：ゲームの結果に対する好みを数値で表したもの（大きいほうが望ましい）
  - 利得関数：結果を利得に対応させる関数
- ゲームの解：ゲームにおいて予想される結果
  - 支配戦略・ナッシュ均衡などの解概念が存在

A ＼ B	嘘をつく	認める
嘘をつく	(-1 , -1)	(-10 , 0)
認める	(0 , -10)	(-3 , -3)

暗号理論におけるゲーム

- 安全性ゲーム
  - 攻撃者 vs チャレンジャー



- 攻撃者による１人ゲーム（戦略的状況ではない）
  - チャレンジャーの行動は規定通り
  - 攻撃者は攻撃成功確率を最大化すればよい

暗号理論とゲーム理論に関する研究 (1/2)

- 暗号理論の登場人物を「合理的」に

与えられた環境下で自身の利得の最大化を目指す

  - 正直者を合理的に
    - 秘密分散（秘匿計算）[HT04, ADGH06, GK06, KN08a, KN08b, MS09, OPRV09, FKN10, NS12, KOTY17, etc.]
    - リーダー選出・コイン投げ・コンセンサス [Gra10, BCZ12, ADH13, AGLS14, HV16]
    - 公開鍵暗号 [Y16, YY17]
  - 敵対者を合理的に
    - ビザンチン合意 [GKTZ12]
    - プロトコル設計 [GKMTZ13, GKTZ15]
    - セキュアメッセージ転送 [FK17]

暗号理論とゲーム理論に関する研究 (2/2)

- ゲーム理論に暗号技術を活用
  - 信頼できる仲介者（相関均衡）を暗号技術で実現 [DHR00, LMS05, ILM05, ILM08]
  - ナッシュ均衡解発見の困難性 [BPR15, GPS16, RSS17]
  - 繰り返しゲームにおける均衡解の発見 [BC+10,HPS14a, HPS14b]

- ゲーム理論と暗号理論の概念間の関係
  - 暗号理論向けの均衡概念の導入 [HP10, GLV10, PS11, HPS16]
  - 均衡概念による安全性の特徴づけ [ACH11, GK12, HTYY12]

- 暗号技術に金銭（報酬・罰金）を活用
  - ビットコイン [Nak08, Ros11, LJG15, CK+16, SB+16, FPS17]
  - 報酬つき委託計算 [AM13, GHRV14, CG15, GHRV16, IY17]
  - 罰金つき秘匿計算 [AD+14, BK14, KB14, KK+16, KB16]

以降の内容

- 使える！ゲーム理論風テクニック
- ゲーム理論的に自然な安全性とは？
- まとめ

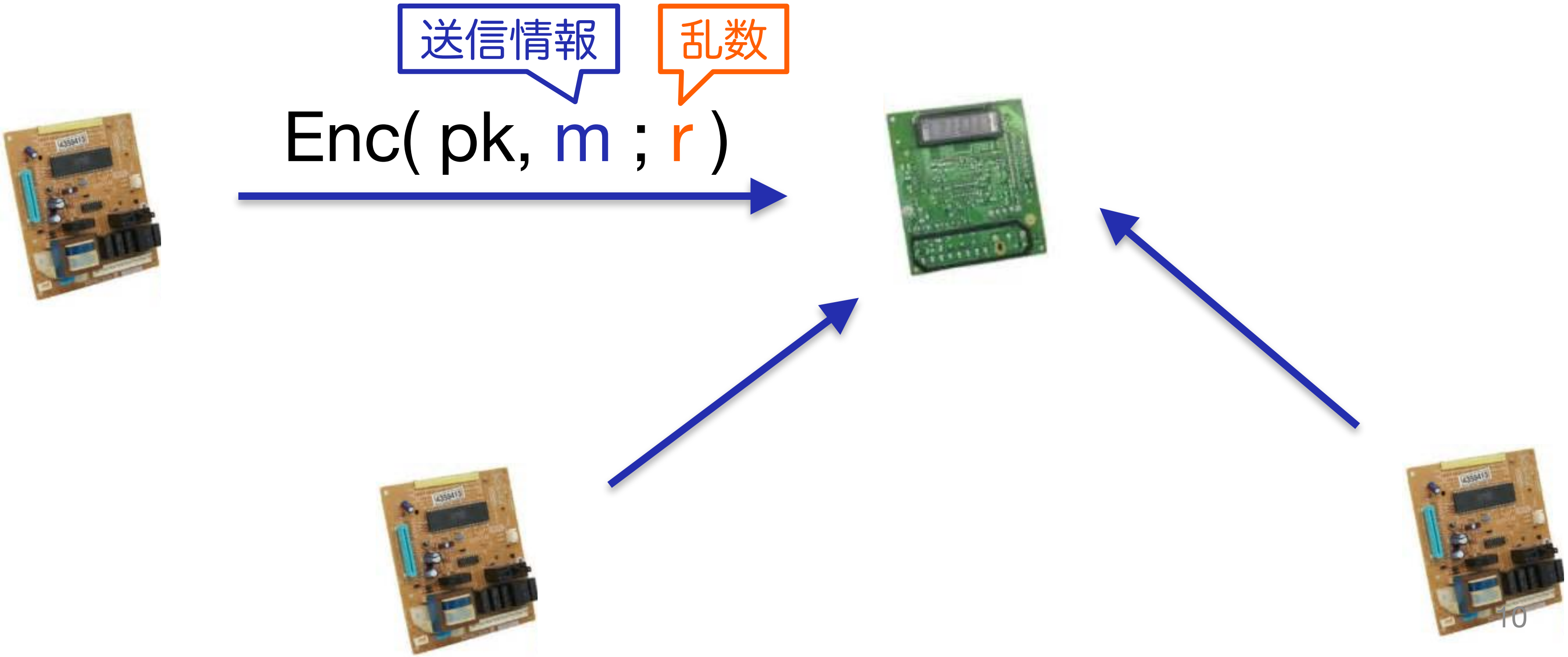
使える！ゲーム理論風テクニック

- 繰り返しゲーム（題材：公開鍵暗号）
- スコアリングルール（題材：委託計算）

[Y16] Yasunaga. Public-key encryption with lazy parties. IEICE Trans. Fund. (2016)
[YY17] Yasunaga, Yuzawa. Repeated games for generating randomness in encryption. Cryptology ePrint Archive: 2017/218
[IY17] Inasawa, Yasunaga. Rational proofs against rational verifiers. IEICE Trans. Fund. (2017)

背景：IoT における省電力デバイス間暗号化通信

- デバイス間の暗号化通信 → 公開鍵暗号で実現
  - 安全な暗号化のためには「乱数生成」が必要
  - 省電力デバイスには「乱数生成」は高コスト



背景：IoT における省電力デバイス間暗号化通信

■ ここで問題が・・・

一様なランダムな r
の生成は高コスト

送信情報　乱数

Enc(pk, m ; r)

自身のプライバシに関係ない送信情報 m に対しては
r = 0 … 0 を使うことでコスト削減！

いや、困る

このようなデバイス間でも
安全に通信できる仕組みを作りたい

問題の設定

- 送信者 S と受信者 R が合理的なプレイヤーであり

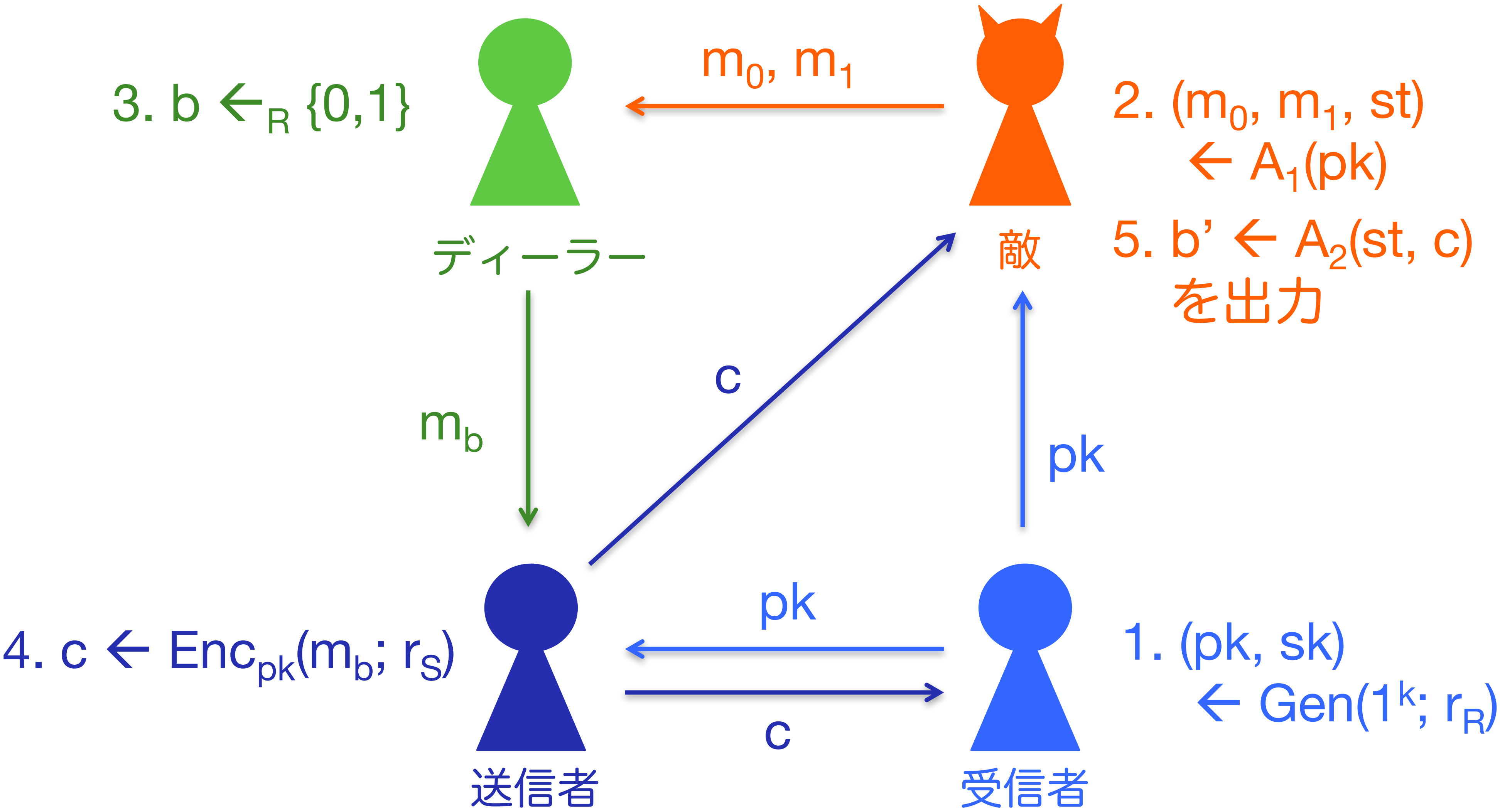
  (1) 安全性に関心のあるメッセージ集合 M_S, M_R をもち

  (2) 乱数生成を高コストだと考える

  - 以下を選択可能
    1. 生成コストありの一様乱数系列（Good 乱数）
    2. 生成コストなしのオールゼロ系列（Bad 乱数）

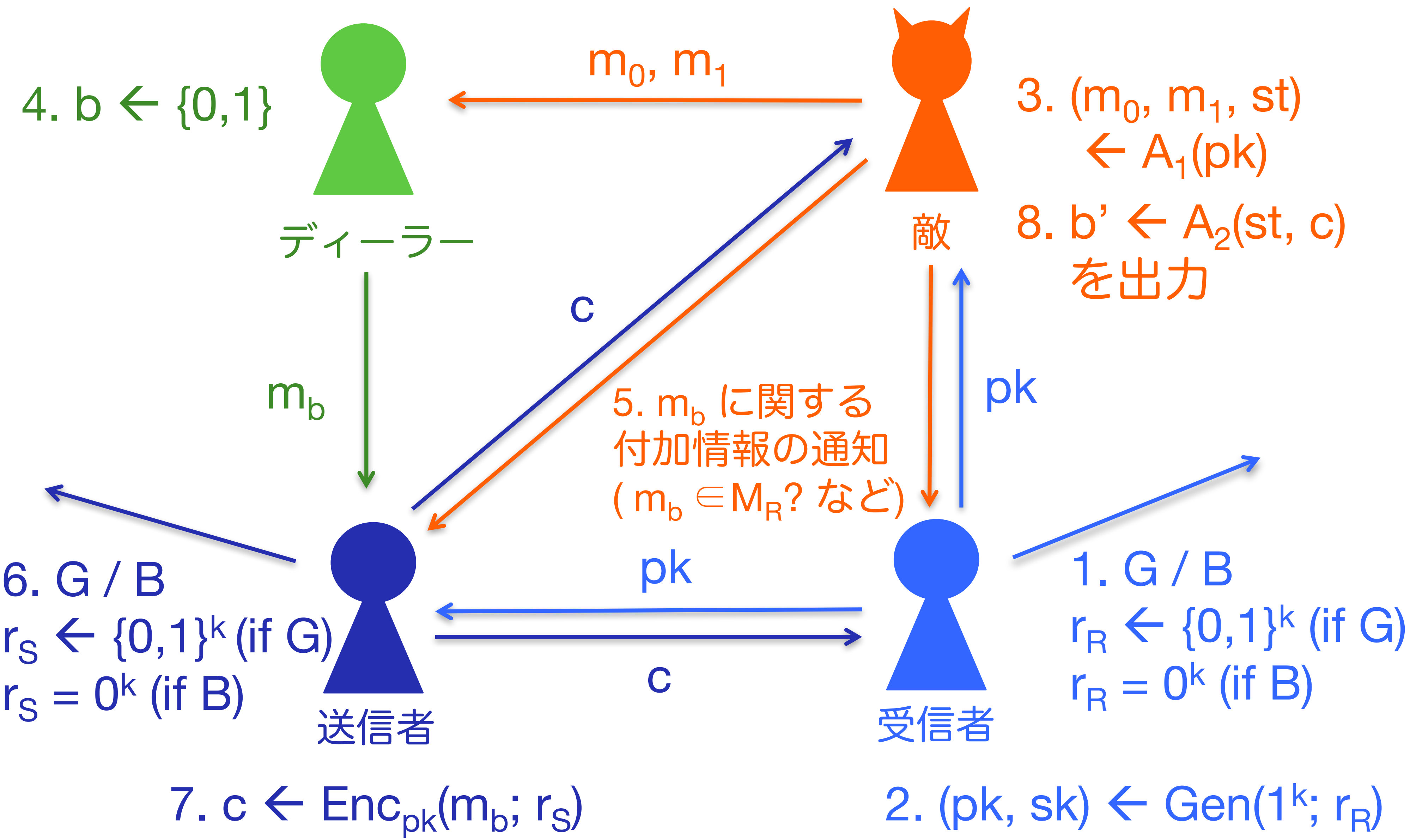
- S, R, および攻撃者 A による安全性ゲームを定義
  - S と R は利得を最大化するために行動
  - $\forall m \in M_S \cup M_R$ を安全にする方式を設計したい

通常の安全性ゲーム



3. b ←R {0,1}
m0, m1
2. (m0, m1, st)
← A1(pk)
ディーラー
敵
5. b' ← A2(st, c)
を出力
c
mb
pk
pk
4. c ← Encpk(mb; rS)
1. (pk, sk)
← Gen(1k; rR)
c
送信者
受信者

今回の安全性ゲーム



4. b ← {0,1}
ディーラー
m_0, m_1
敵
3. $(m_0, m_1, st) \leftarrow A_1(pk)$
8. $b' \leftarrow A_2(st, c)$ を出力
m_b
c
5. m_b に関する付加情報の通知 ($m_b \in M_R$? など)
pk
6. G / B
$r_S \leftarrow \{0,1\}^k$ (if G)
$r_S = 0^k$ (if B)
送信者
pk
c
受信者
1. G / B
$r_R \leftarrow \{0,1\}^k$ (if G)
$r_R = 0^k$ (if B)
7. $c \leftarrow Enc_{pk}(m_b; r_S)$
2. $(pk, sk) \leftarrow Gen(1^k; r_R)$

安全性ゲームについて

- 実際は、一般化して定義
  - 送信者も Gen を実行
  - Enc は対話も可
  - m_b に関する付加情報は重要

- プレイヤーの戦略は、Gen, Enc に対する G/B の選択

- ゲームの出力は
  $\mathrm{Out} = (\mathrm{Win}, \mathrm{Val}_S, \mathrm{Val}_R, \mathrm{Num}_S, \mathrm{Num}_R)$
  - $\mathrm{Win} \in \{0,1\}$, $\mathrm{Win} = 1 \Leftrightarrow b = b'$
  - $\mathrm{Val}_w \in \{0,1\}$, $\mathrm{Val}_w = 1 \Leftrightarrow m \in M_w$
  - $\mathrm{Num}_w : w \in \{S, R\}$ が Good を選択した回数

利得関数

- 出力 Out = (Win, Val_S, Val_R, Num_S, Num_R) のとき

$$u_w(Out) = u_w^{sec} \cdot (-\,Win) \cdot Val_w + (-\,c_w^{rand}) \cdot Num_w$$

  - u_w^{sec}, $c_w^{rand} > 0$ はある固定実数値
  - $u_w^{sec}/2 > q_w \cdot c_w^{rand}$ と仮定（ q_w : Num_w の最大値 ）
    - Good のコストで $u_w^{sec}/2$ の利得（安全性）を得る価値あり

- 戦略の組 (σ_S, σ_R) に従ったときの利得

$$U_w(\sigma_S, \sigma_R) = \min E[u_w(Out)]$$

  - min はすべての敵, メッセージ空間 M_S, M_R でとる

安全な方式の構成アイディア

- 注意点：R が m $\in$ M_R であるか否かを知らなければ R に乱数生成させればいいが、知っているかもしれない
  - R がそれを知っているか否かも S は知らない

- 3 ラウンド暗号方式 Π_3 の構成アイディア
  - 暗号化フェーズで鍵共有
    - どちらかが Good を使えば共有鍵も Good
    - 共有鍵を暗号化の乱数とする
  - 鍵生成でも Good を使う必要があるように

3ラウンド暗号方式 Π_3

送信者 / 受信者

鍵生成

受信者: $(pk_R, sk_R) \leftarrow Gen(1^k; \boxed{r_1^R})$

$\xleftarrow{pk_R}$

送信者: $(pk_S, sk_S) \leftarrow Gen(1^k; \boxed{r_1^S})$

$\xrightarrow{pk_S}$

暗号化

受信者: $\boxed{r_2^R \leftarrow_R U}$

受信者: $c_1 \leftarrow Enc(pk_S, r_2^R; \boxed{r_3^R})$

$\xleftarrow{c_1}$

送信者: $r_2^R \leftarrow Dec(sk_S, c_1)$

送信者: $\boxed{r_2^S \leftarrow_R U}$

送信者: $r = r_2^R \oplus r_2^S \ (= r_L \circ r_R)$

送信者: $c_2 \leftarrow Enc(pk_R, r_2^S; \boxed{r_3^S})$

送信者: $c_3 \leftarrow Enc(pk_R, m; r_L)$

$\xrightarrow{c_2,\, c_3}$

受信者: $r_2^S \leftarrow Dec(sk_R, c_2)$

受信者: $r = r_2^R \oplus r_2^S \ (= r_L \circ r_R)$

受信者: $m \leftarrow Dec(sk_R, c_3)$

受信者: $c_4 \leftarrow Enc(pk_S, m; r_R)$

$\xleftarrow{c_4}$

3 ラウンド方式 Π_3 の安全性

定理 1

(1) Π_3 に従えば $m \in M_S \cup M_R$ は安全

(2) Π_3 に従うことは狭義ナッシュ均衡

逸脱すると損する

■ 証明概要

	S の鍵生成	R の鍵生成	S の暗号化	R の暗号化	秘匿性
(1)	Good	Good	Good	-	✓
(2)	Good	Good	-	Good	✓
(3)	-	Bad	-	-	x
(4)	Bad	-	-	-	x

- (1), (2) の秘匿性達成は簡単に確認可能
- (3) のとき $m \in M_R \setminus M_S$ は c_2, c_3 から破られる
- (4) のとき $m \in M_S \setminus M_R$ は c_4 から破られる

（無限）繰り返しゲーム

- プレイヤー間の長期的関係による振る舞いを説明可能
  - ステージゲームを繰り返し実行
  - 前ステージの結果を観測後、次ステージ実行

- 囚人のジレンマでは１回きりゲームと異なる均衡を達成

A ＼ B	嘘をつく	認める
嘘をつく	(-1 , -1)	(-10 , 0)
認める	(0 , -10)	(-3 , -3)

裏切ると罰を与える仕組みを導入。裏切ると長期的には損

繰り返しゲームにもとづいた問題設定

- 鍵生成は１回限り、
暗号文の送受信を複数回（無限回）繰り返す
と考えて利得を計算

- 利得は無限回の合計

$\delta \in (0,1)$: 割引因子

$$u_w(\mathrm{Out}) = (-\, c_w^{\mathrm{rand}}) \cdot \mathrm{Num}_w^{\mathrm{Gen}} + \textstyle\sum_{i=1,2,\ldots} \delta^{i-1}\, u_w[i]$$

$$u_w[i] = u_w^{\mathrm{sec}} \cdot (-\, \mathrm{Win}) \cdot \mathrm{Val}_w^{i} + (-\, c_w^{\mathrm{rand}}) \cdot \mathrm{Num}_w^{i}$$

$$U_w(\sigma_S, \sigma_R) = \min E[u_w(\mathrm{Out})]$$

繰り返しゲーム設定における安全な方式の構成アイディア

- 2 ラウンド方式 Π_2^{Repeat} の構成アイディア：
  - Π_3 : Enc で S または R が Good を使う限り安全
  - S, R ともに Good のときだけ安全な方式へ変更

  → 無限繰り返しゲームではトリガー戦略が均衡
    - トリガー戦略：基本的には Good を選択し、相手が Bad を選択した場合、次回以降 Bad を選択
    - Bad を選ぶと、１回は利得が増えるが、その後は安全性を達成できず、長期的には下がる

繰り返しゲームによってラウンド数を削減

繰り返しゲーム向け 2 ラウンド方式 Π_2^{Repeat}

送信者 S		受信者 R
鍵生成フェーズ	$\xleftarrow{pk_R}$	$(pk_R, sk_R) \leftarrow \mathrm{Gen}(1^k; \boxed{r_1^R})$
$(pk_S, sk_S) \leftarrow \mathrm{Gen}(1^k; \boxed{r_1^S})$	$\xrightarrow{pk_S}$	
暗号化フェーズ		$\boxed{r_2^R \leftarrow_R U}$
$r_2^R \leftarrow \mathrm{Dec}(sk_S, c_1)$	$\xleftarrow{c_1}$	$c_1 \leftarrow \mathrm{Enc}(pk_S, r_2^R; \boxed{r_3^R})$
$\boxed{r_2^S \leftarrow_R U}$		
$r = r_2^R \circ r_2^S$		
$c_2 \leftarrow \mathrm{Enc}(pk_R, r_2^S; \boxed{r_3^S})$	$\xrightarrow{c_2,\ c_3}$	$r_2^S \leftarrow \mathrm{Dec}(sk_R, c_2)$
$c_3 \leftarrow m \oplus r$		$r = r_2^R \circ r_2^S$
		$m \leftarrow c_3 \oplus r$

2 ラウンド方式 Π_2^{Repeat} の安全性

定理 2

繰り返しゲーム設定において

(a) $\Pr[\, m \in M_S \,] > c_S^{\mathrm{Enc}} / (\delta\, u_S^{\mathrm{Sec}})$

(b) $\Pr[\, m \in M_R \,] > c_R^{\mathrm{Enc}} / (\delta\, u_R^{\mathrm{Sec}})$

を満たすとき

(1) Π_2^{Repeat} に従えば $m \in M_S \cup M_R$ は安全

(2) Π_2^{Repeat} に従うことはナッシュ均衡

c_S^{Enc} : S の暗号化時 Good 乱数のコスト

u_S^{Sec} : 安全に送信できたときに S が得る利得

δ : 利得の割引因子, c_R^{Enc}, u_R^{Sec} は同様に定義

繰り返しゲームのまとめ

- 長期的関係性を考慮することで
  １回きりゲームとは異なる均衡を達成可能
- 乱数生成が高コストのプレイヤーに対し
  ２ラウンド方式を実現（ラウンド数 2 は最適）
  - 以下の仮定が必要
    (1) ステージゲーム毎に結果を観測可能
    (2) メッセージ出現確率がある値以上
- 他の暗号技術への適用は？

スコアリングルール

- 天気予報士から正しい予報を聞き出すための道具
  - 確率空間 $\Omega = \{$ ,  $\}$
  - P：正しい確率分布
  - Q：予測した確率分布
  - S：スコアリングルール

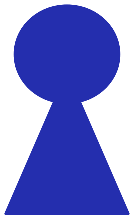
Brierルール

$$S^B(Q,\omega) = 2Q(\omega) - \sum_{\omega' \in \Omega} Q(\omega')^2 - 1$$

$$= -\sum_{\omega' \in \Omega} (T(\omega') - Q(\omega'))^2$$

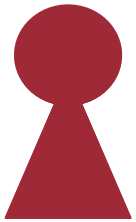
$P($  $) = 0.6$

$P($  $) = 0.4$



予報 Q →

← 報酬 $= S(Q,\omega)$



予報者

検証者

実際の天気

$\omega \leftarrow P$

$$T(\omega') = \begin{cases} 1 & \text{if } \omega' = \omega \\ 0 & \text{otherwise} \end{cases}$$

$\forall Q \neq P$ に対して

$$\underbrace{\sum_{\omega \in \Omega} P(\omega) S(P,\omega)}_{P\text{を予測時の期待スコア}} > \underbrace{\sum_{\omega \in \Omega} P(\omega) S(Q,\omega)}_{Q\text{を予測時の期待スコア}}$$

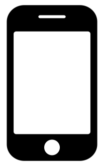
（検証可能）委託計算

- クラウドサーバに関数 f の計算を委託したい

クラウドサーバ

f の委託計算
プロトコル

本当に f(x) だろうか？



ユーザ

入力 x

出力 f(x)

- サーバが正しく計算しているか検証したいが、
そのコストは小さくしたい

➔「ユーザの検証コスト << f(x) の計算コスト」の実現

報酬つき委託計算 [Azar, Micali (2013), Guo ら (2014)]

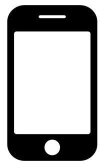
- サーバとの対話履歴から報酬を決定

不正しても儲からん

クラウドサーバ

f の委託計算
プロトコル

T



ユーザ

入力 x

出力 f(x)

報酬 R(T, x)

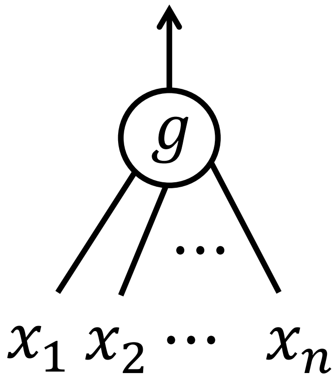
- 正しく計算すれば報酬が最大化（不正すると減額）

➔ サーバは、報酬最大化のため正しく計算

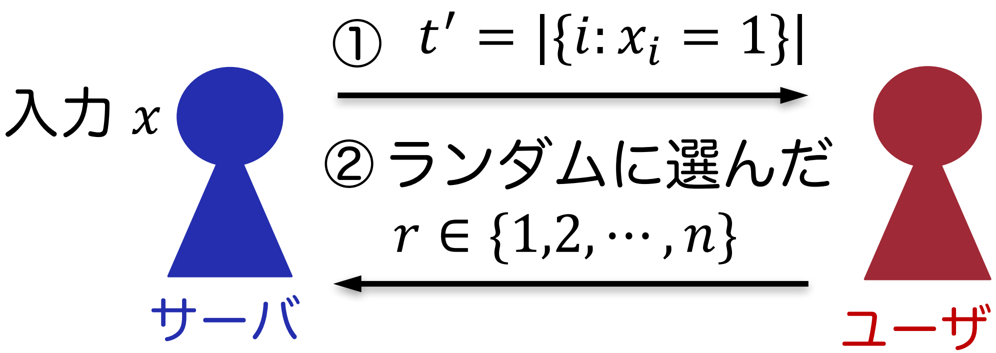
サーバの合理性を信じて検証の手間を省略

しきい値素子に対する委託計算

- しきい値 t のしきい値素子
  - $g(x) = \begin{cases} 1 & x_1 + \cdots + x_n \geq t \\ 0 & x_1 + \cdots + x_n < t \end{cases}$



- 委託計算プロトコル



③ 報酬 = $S^B(Q, x_r)$ を計算

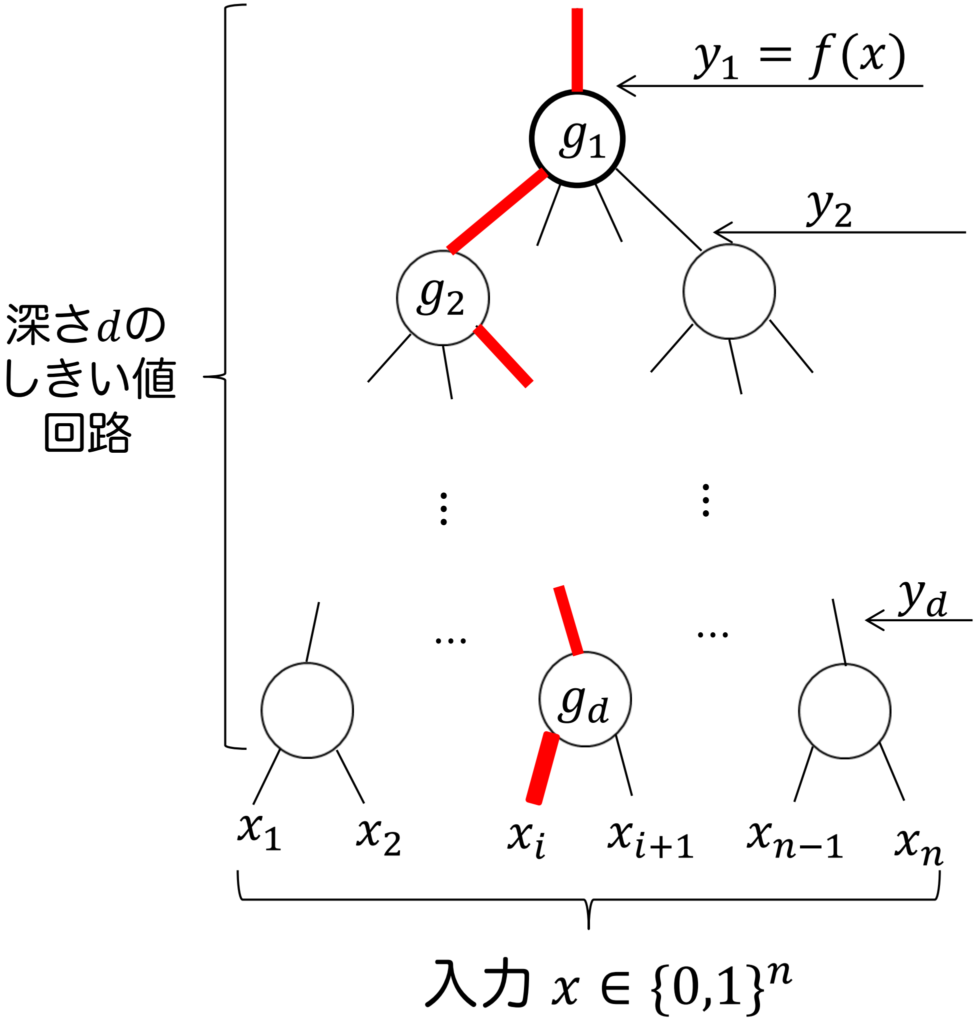
Q は $\{0,1\}$ 上の分布であり

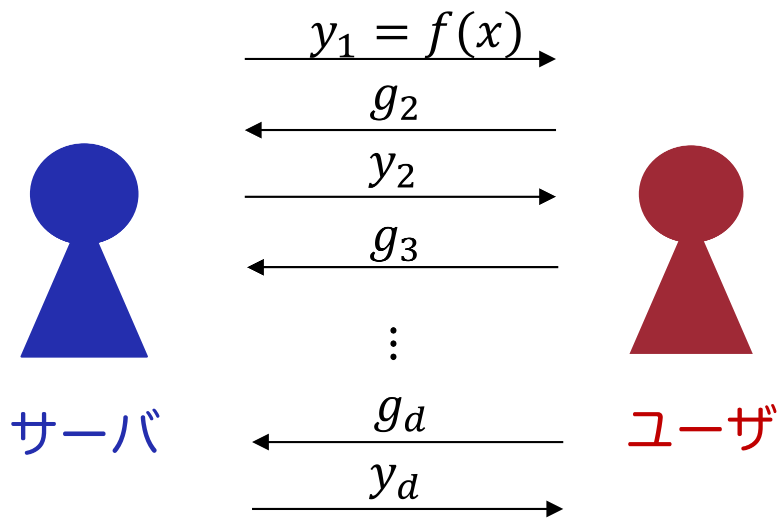
$$\Pr[Q = 1] = \frac{t'}{n}$$

④ 出力は $t' \geq t$ のとき 1, それ以外 0

- サーバの予報は Q
- 実際の天気は x_r ➔ 正しい分布 P : $\Pr[P = 1] = \frac{|\{i : x_i = 1\}|}{n}$
- スコアリングルールの性質より $Q = P$ のとき期待報酬最大
→ 報酬最大化のため正しく答えると、正しい出力を得る
- 検証時間 O(log n) の委託計算を実現

Guoら (2014) の委託計算プロトコル





深さ d, サイズ S しきい値回路を持つ f に対し、
検証時間 O(d･polylog(S)) を実現

GuoらのプロトコルΘ問題点とその対処法 [IY17]

- 検証者は意図的な報酬減額が可能

合理的な検証者

1. 入力へ早く接続することで報酬を減らす
2. 期待報酬の小さい素子を選ぶ
   - $t < n/2$ → 値 1 の入力が多い方が期待値小
   - $t \geq n/2$ → 値 0 の入力が多いほうが期待値小

ナッシュ均衡の実現

- 成果：検証者が不正できないプロトコルを提案
  - 検証者の乱数を、二者間コイン投げで決定
  - ランダムオラクルを仮定すれば
    検証時間 polylog(n) の 3 ラウンドプロトコル
  - 標準モデルにおける構成・最適ラウンド数は未解決

スコアリングルールのまとめ

- 天気予報士（専門家）から正しい予報（予測）を聞き出すための道具

- 報酬つき委託計算に応用可能
  - 深さ d, サイズ S しきい値回路を持つ f に対し、検証時間 O(d・polylog(S)) を実現

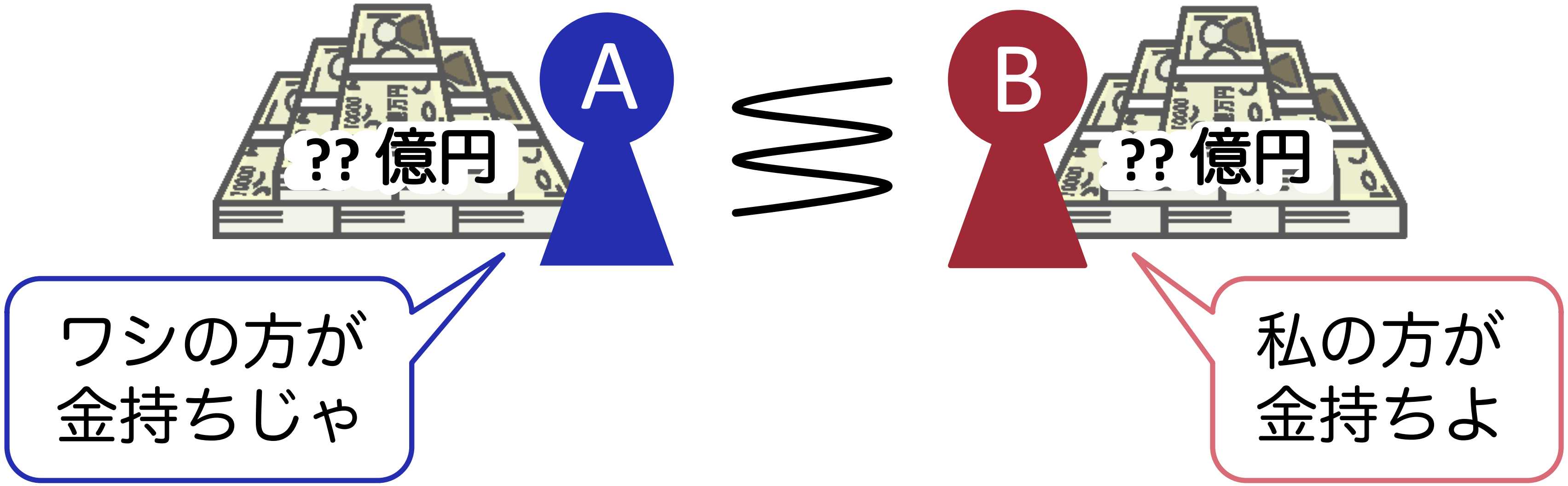
- 報酬によって効率化可能なその他の暗号技術・プロトコルは？

- スコアリングルールのその他の応用先は？

ゲーム理論的に自然な安全性とは？

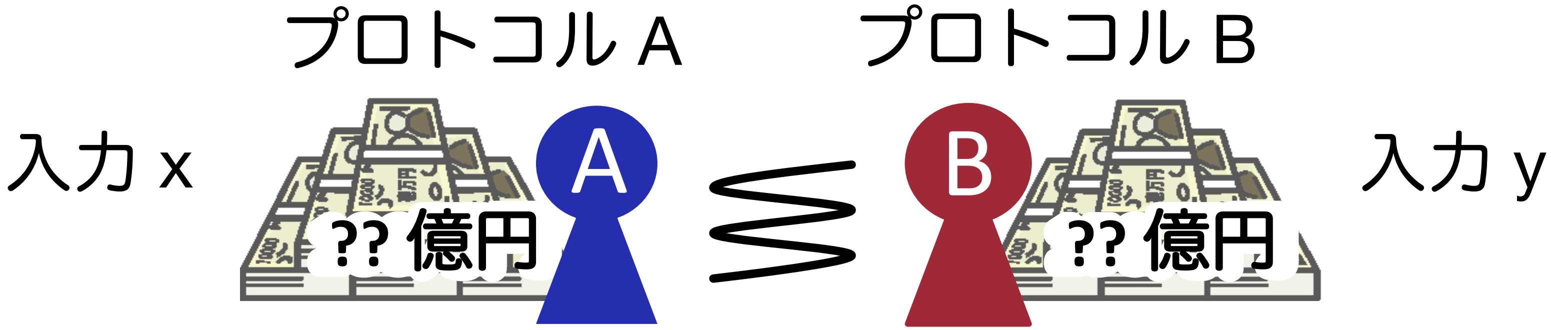
[HTYY17] Higo, Tanaka, Yamada, Yasunaga.
Game-theoretic security for two-party protocols.
Cryptology ePrint Archive: 2016/1072

億万長者問題



- どちらが金持ちか知りたい
- 自分の資産額を知られたくない

億万長者プロトコルに対する暗号理論的な安全性



- 両者プロトコルに従えば、どちらが金持ちかがわかる（正当性）
  - $\forall$ x, y,   output((A(x), B(y))) = ($\mathbf{1}$(x > y), $\mathbf{1}$(x > y))

- B がどのように振る舞っても、A がプロトコルに従う限り、A の資産額を知ることができない（A の安全性）
  - $\forall\ x_0, x_1, y$ s.t. $\mathbf{1}(x_0 > y) = \mathbf{1}(x_1 > y)$, PPT B^*, D_B,
    $\Pr[\,D_B(\mathrm{view}_{B^*}(A(x_a), B^*(y))) = 1\,] \approx 1/2$ （ただし $a \in_R \{0,1\}$ ）

- A がどのように振る舞っても、B がプロトコルに従う限り、B の資産額を知ることができない（B の安全性）
  - $\forall\ x, y_0, y_1$ s.t. $\mathbf{1}(x > y_0) = \mathbf{1}(x > y_1)$, PPT A^*, D_A,
    $\Pr[\,D_A(\mathrm{view}_{A^*}(A^*(x), B(y_b))) = 1\,] \approx 1/2$ （ただし $b \in_R \{0,1\}$ ）

暗号理論的な安全性を ゲーム理論の言葉で解釈してみよう

- （自然な）ゲームは？
  - 入力は $\forall\ x_0, x_1, y_0, y_1$ s.t. $\mathbf{1}(x_0 > y_0) = \mathbf{1}(x_1 > y_0) = \mathbf{1}(x_0 > y_1) = \mathbf{1}(x_1 > x_1)$
  - ランダム $a, b \in \{0,1\}$ に対し入力を (x_a, y_b) として、プロトコルを実行。D_A が b を、D_B が a を推測
  - ゲームの出力 $(suc_A, suc_B, guess_A, guess_B)$：
    - $suc_A = 1 \Leftrightarrow$ A が $\mathbf{1}(x_a > y_b)$ を出力 or プロトコルが中断
    - $suc_B = 1 \Leftrightarrow$ B が $\mathbf{1}(x_a > y_b)$ を出力 or プロトコルが中断
    - $guess_A = 1 \Leftrightarrow D_A$ が b を出力
    - $guess_B = 1 \Leftrightarrow D_B$ が a を出力

暗号理論的な安全性を
ゲーム理論の言葉で解釈してみよう

- （自然な）利得関数は？
  - A は以下の選好をもつ：
    (1) 中断しない限り $\mathbf{1}(x_a > y_b)$ を知りたい
    (2) B の入力 y_b を当てたい
    (3) 自身の入力 x_a を B に当てられたくない
  - B も同様の選好をもつ

  - $u_A((A, D_A), (B, D_B)) = suc_A + guess_A - guess_B$
  - $u_B((A, D_A), (B, D_B)) = suc_B + guess_B - guess_A$

暗号理論的な安全性のゲーム理論による特徴付け

定理 3

プロトコル (A, B) が暗号理論的に安全

⇔ (A, B) が先ほどのゲームで「ナッシュ均衡もどき」

$\overset{\text{定義}}{\Leftrightarrow}$ $\forall$ PPT A^*, D_A, D_B, 有効な入力 x_0, x_1, y_0, y_1に対し

$$E[\, u_A(\, (A^*, D_A), (B, D_B)\,)\,] \leq E[\, u_A(\, (A, D_A), (B, D_B)\,)\,]$$

かつ

$\forall$ PPT B^*, D_A, D_B, 有効な入力 x_0, x_1, y_0, y_1に対し

$$E[\, u_B(\, (A, D_A), (B^*, D_B)\,)\,] \leq E[\, u_B(\, (A, D_A), (B, D_B)\,)\,]$$

戦略の組 (A, B) がナッシュ均衡

⇔ $\forall\, A^*$, $E[\, u_A(\, A^*, B\,)\,] \leq E[\, u_A(\, A, B\,)\,]$

かつ $\forall\, B^*$, $E[\, u_B(\, A, B^*\,)\,] \leq E[\, u_B(\, A, B\,)\,]$

定理３の証明の概要

- 「暗号理論的に安全 → ナッシュ均衡もどき」
  - ナッシュ均衡でないとする
  - u_A において A → A* によって
    - suc_A 増： (A, B) は正当性を満たさない
    - $guess_A$ 増： A or A* で攻撃可 → B 安全性を満たさない
    - $guess_B$ 減： B or B* で攻撃可 → A 安全性を満たさない
  - u_B も同様に証明可能
- 「ナッシュ均衡もどき → 暗号理論的に安全」
  - 正当性を満たさない：
$D_A = D_B = D^{rand}$ で A → A^{abort} とすれば u_A が増
  - 正直者 B が A 安全性を破る：存在する D_B に対し、
$D_A = D^{rand}$ で A → A^{abort} とすれば u_A が増
  - $B^* \neq B$ が A 安全性を破る：存在する B^*, D_B に対し、
$D_A = D^{rand}$ で B が B^* 使用後に abort とすれば u_B が増
  - B 安全性を満たさない場合も同様に証明可能

先ほどの特徴付けはゲーム理論的に妥当か？

(A, B) がナッシュ均衡もどきで「ない」

⇔ ∃ PPT A^*, D_A, D_B, 有効な入力 x_0, x_1, y_0, y_1 s.t.

$$E[\, u_A(\,(A^*, D_A), (B, D_B)\,)\,] > E[\, u_A(\,(A, D_A), (B, D_B)\,)\,]$$

または

∃ PPT B^*, D_A, D_B, 有効な入力 x_0, x_1, y_0, y_1 s.t.

$$E[\, u_B(\,(A, D_A), (B^*, D_B)\,)\,] > E[\, u_B(\,(A, D_A), (B, D_B)\,)\,]$$

- A が高い利得を得るために、都合のよい D_B を仮定
  ➔ D_B は B の戦略の一部であり仮定するのはやや不自然
- D_A, D_B のデフォルトアルゴリズムを定めていない
  ➔ 戦略の一部であり、定めるべき（D^{rand} が妥当？）

より自然なゲーム理論的な特徴づけ

(A, B) が「自然な」ナッシュ均衡もどきで「ない」

⇔ $\exists$ PPT A^*, D_A, 有効な入力 x_0, x_1, y_0, y_1 s.t. $\forall$ PPT D_B

$$E[\, u_A(\, (A^*, D_A), (B, D_B)\,)\,] > E[\, u_A(\, (A, D^{rand}), (B, D_B)\,)\,]$$

または

$\exists$ PPT B^*, D_B, 有効な入力 x_0, x_1, y_0, y_1 s.t. $\forall$ PPT D_A

$$E[\, u_B(\, (A, D_A), (B^*, D_B)\,)\,] > E[\, u_B(\, (A, D_A), (B, D^{rand})\,)\,]$$

定義 (A, B) が「適応的ナッシュ均衡もどき」

D_B を適応的に選択

$\overset{定義}{\Leftrightarrow}$ $\forall$ PPT A^*, D_A, 有効な入力 x_0, x_1, y_0, y_1 $\exists$ PPT D_B s.t.

$$E[\, u_A(\, (A^*, D_A), (B, D_B)\,)\,] \le E[\, u_A(\, (A, D^{rand}), (B, D_B)\,)\,]$$

かつ

$\forall$ PPT B^*, D_B, 有効な入力 x_0, x_1, y_0, y_1 $\exists$ PPT D_A s.t.

$$E[\, u_B(\, (A, D_A), (B^*, D_B)\,)\,] \le E[\, u_B(\, (A, D_A), (B, D^{rand})\,)\,]$$

適応的ナッシュ均衡の暗号理論的な意味は？

定義 (A, B) が適応的ナッシュ均衡もどき

$\overset{\text{定義}}{\Leftrightarrow}$ $\forall$ PPT A^*, D_A, 有効な入力 x_0, x_1, y_0, y_1 $\exists$ PPT D_B s.t.

$$E[\, u_A((A^*, D_A), (B, D_B))\,] \leq E[\, u_A((A, D^{rand}), (B, D_B))\,]$$

かつ

$\forall$ PPT B^*, D_B, 有効な入力 x_0, x_1, y_0, y_1 $\exists$ PPT D_A s.t.

$$E[\, u_B((A, D_A), (B^*, D_B))\,] \leq E[\, u_B((A, D_A), (B, D^{rand}))\,]$$

- 逸脱範囲を狭めているため、安全性としては弱まっている
- A は、相手側の D_B がどのようなものであっても
  利得が高まるような逸脱を考える

→ 自身の安全性関わる a ∈ {0,1} が推測されない範囲の逸脱

リスク回避的な攻撃

リスク回避的攻撃に対する安全性

ゲーム理論における不確実性に対する
「リスク回避 vs リスク中立」とは異なる

- 正当性：
  - $\forall$ x, y,　output((A(x), B(y))) = ($\mathbf{1}$(x > y), $\mathbf{1}$(x > y))
- 正直者 B に対する A の安全性：
  - $\forall$ 有効な入力 x_0, x_1, y, PPT D_B,
    $\Pr[\, D_B(\, view_B(\, A(x_a), B(y)\,)\,) = 1\,] \approx 1/2,\ a \in_R \{0,1\}$
- リスク回避的な B に対する A の安全性：
  - $\forall$ 有効な入力 x_0, x_1, y_0, y_1, PPT B^*, D_B s.t.

    b を推測されず
    正当性を保つ
    ような B*

    (1) $\forall$ PPT D_A, $\Pr[\, D_A(view_A(A(x_a), B^*(y_b))) = 1\,] \approx 1/2$

    (2) $\Pr[\, output((A(x_a), B(x_b)) = (\mathbf{1}(x > y), \mathbf{1}(x > y)) \vee abort]$
    $= \Pr[\, output(A(x_a), B^*(x_b)) = (\mathbf{1}(x > y), \mathbf{1}(x > y)) \vee abort]$

    $\Pr[\, D_B(\, view_{B^*}(\, A(x_a), B^*(y_b)\,)\,) = 1\,] \approx 1/2$
- 正直な A に対する B の安全性：（上記と同様の定義）
- リスク回避的な A に対する B の安全性：（上記と同様の定義）

適応的ナッシュ均衡との関係

定理 4

プロトコル (A, B) が適応的ナッシュ均衡もどきであるとき、(A, B) はリスク回避的攻撃に対して安全

- **証明概要：**
  リスク回避的攻撃に対し安全でないと仮定
  - 正当性を満たさない：
    $A \rightarrow A^{abort}$ とすれば ∀D_B に対し u_A が増
  - 正直者 B に対し A の安全性を満たさない：D_B に対し
    $(B, D^{rand}) \rightarrow (B, D_B)$ とすれば ∀D_A に対し u_B が増
  - リスク回避的な B に対し A の安全性を満たさない：
    存在する B^*, D_B に対し、
    $(B, D^{rand}) \rightarrow (B^*, D_B)$ とすれば ∀D_A に対し u_B が増
  - B の安全性を満たさない場合も同様に証明可能

ゲーム理論的に自然な安全性のまとめ

- プロトコルの安全性は、ゲーム理論の言葉で解釈可能（ナッシュ均衡もどき）
- ゲーム理論的に自然な定義（適応的ナッシュ均衡）を考えると、安全性は弱まるが、リスク回避的攻撃に対する安全性を満たす
  - さまざまな二者間計算に適用可能な概念
- リスク回避的攻撃安全性については未解決問題多数
  - 通常の安全性との本質的なギャップは？
  - 効率改善につながるのか？
  - プロトコルを多数組み合わせると更に弱くなる？

全体のまとめ

- 今日のお話
  - 使える！ゲーム理論風テクニック
    - 繰り返しゲーム + 罰 → 効率改善
    - スコアリングルール + 合理性 → 効率改善
  - ゲーム理論的に自然な安全性
    - 適応的ナッシュ均衡 → リスク回避的攻撃安全性

- 今後の展望・期待
  - リスク回避的攻撃安全性の発展
  - シミュレーションベース安全性のゲーム理論的解釈