[I486S]
暗号プロトコル理論

藤崎 英一郎

北陸先端科学技術大学院大学

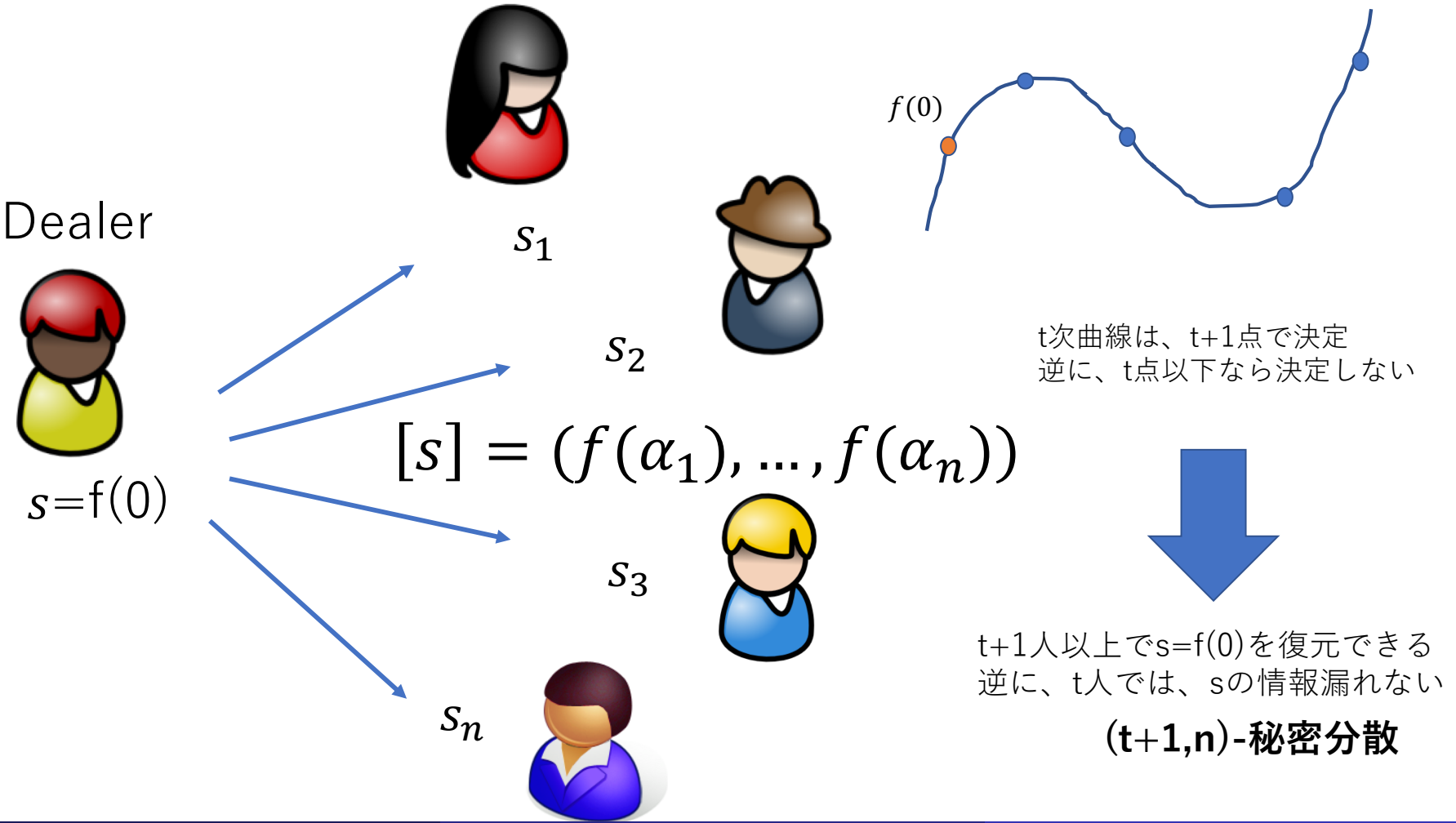
2020 年 5 月 26 日

本日の講義の内容

1. Shamir 秘密分散（復習）
2. 足し算と掛け算（復習）
3. 耐受動的攻撃安全なマルチパーティ計算（$t < n/2$ の場合）
4. 証明（$t < n/2$ の場合）
5. $t \geq n/2$ の場合

Shamir 秘密分散

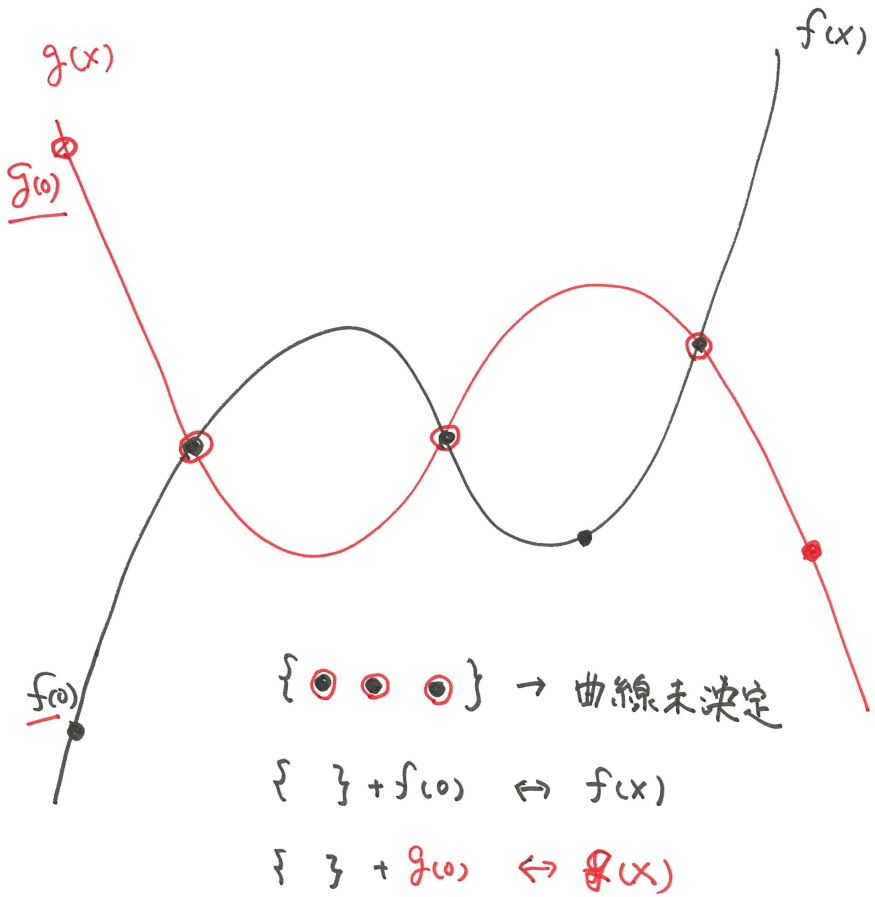
Dealer は、多項式 f を $s = f(0)$ かつ $\deg(f) = t$ となる条件でランダムに選ぶ。秘密を分割して参加者間で保持。t 人では秘密が全く漏れない。$t+1$ 人以上で秘密を復元できる。



多項式の決定から

$f(X)$ の決定 $\Longleftrightarrow$ 曲線の $\deg(f)+1$ 点の決定

- （$f(0)$ 以外の）$\deg(f)$ 点が漏れる　$\Longrightarrow$ 秘密 $f(0)$ の情報全くなし
- $\deg(f)+1$ 点を公開　$\Longrightarrow$ $f(X)$ が決定するので $f(0)$ が決定



Shamir 秘密分散 (formal)

Shamir 秘密分散 SS $= (\mathsf{Share}, \mathsf{Recon}, \Gamma_{t+1,n})$. $\alpha_1, \ldots, \alpha_n \in K$ はシステムで予め定められた異なる値。

Shamir SS

- Share(s):
  - $a_0 := s$ の条件のもと K 係数のランダムな t 次多項式 $f(X) = \sum_{i=0}^{t} a_i X^i$ を選ぶ。

$$f(X) \leftarrow_R K[X] \quad \text{such that} \quad \deg(f) = t \text{ and } a_0 = s.$$

  - $[s] = (f(\alpha_1), \ldots, f(\alpha_n))$ を出力する。
- $\mathsf{Recon}(S_Q)$ ($S_Q = \{f(\alpha_i) | i \in Q \text{ s.t. } Q \in \Gamma_{t+1,n}\}$): s を出力。

$$s = \sum_{i \in Q} \lambda_{i,Q} f(\alpha_i) \text{ where } \lambda_{i,Q} = \prod_{j \in Q \setminus \{i\}} \left(\frac{\alpha_j}{\alpha_j - \alpha_i} \right).$$

Reconstruction vector $(\lambda_{1,Q}, \ldots, \lambda_{\#Q,Q})$ は、S_Q のみで決定することに注意

本日の講義の内容

1. Shamir 秘密分散（復習）
2. 足し算と掛け算（復習）
3. 耐受動的攻撃安全なマルチパーティ計算（$t < n/2$ の場合）
4. 証明（$t < n/2$ の場合）
5. $t \geq n/2$ の場合

Shamir 秘密分散は線型

$f, g \in K[X]$ $(\deg(f), \deg(g) \leq t)$ に対して次のことは明らか。

線型性

$$[\alpha f(0) + \beta g(0)] = \alpha[f(0)] + \beta[g(0)] \qquad \text{where } \alpha, \beta \in K.$$

$$f(X) = a_0 + a_1 X + \ldots a_t X^t \text{ および } g(X) = b_0 + b_1 X + \ldots b_t X^t$$

とする。$h(X) \triangleq \alpha f(X) + \beta g(X)$ と定義すると、$\deg(h) \leq t$ で、

$$\begin{aligned}
[h(0)] &= (h(\alpha_1), \ldots, h(\alpha_n)) \\
&= (\alpha f(\alpha_1) + \beta g(\alpha_1), \ldots, \alpha f(\alpha_n) + \beta g(\alpha_n)) \\
&= \alpha[f(0)] + \beta[g(0)]
\end{aligned}$$

Shamir SS の足し算

線形性から簡単に計算できる。

addition

$[a]$, $[b]$: $a, b \in K$ が $P_1, \ldots, P_n$ 間でシェアされた状態

$$[a] + [b] = [a + b]$$

f, g をそれぞれ a, b をシェアする時の t 次多項式とする。$f_0 = a$, $g_0 = b$ で、

$$f(X) = f_0 + f_1 X + \ldots + f_t X^t, \text{ and } g(X) = g_0 + g_1 X + \ldots + g_t X^t.$$

$h(X) = f(X) + g(X)$ と置くと、$h(0) = a + b$ と $\deg(h) = t$ より

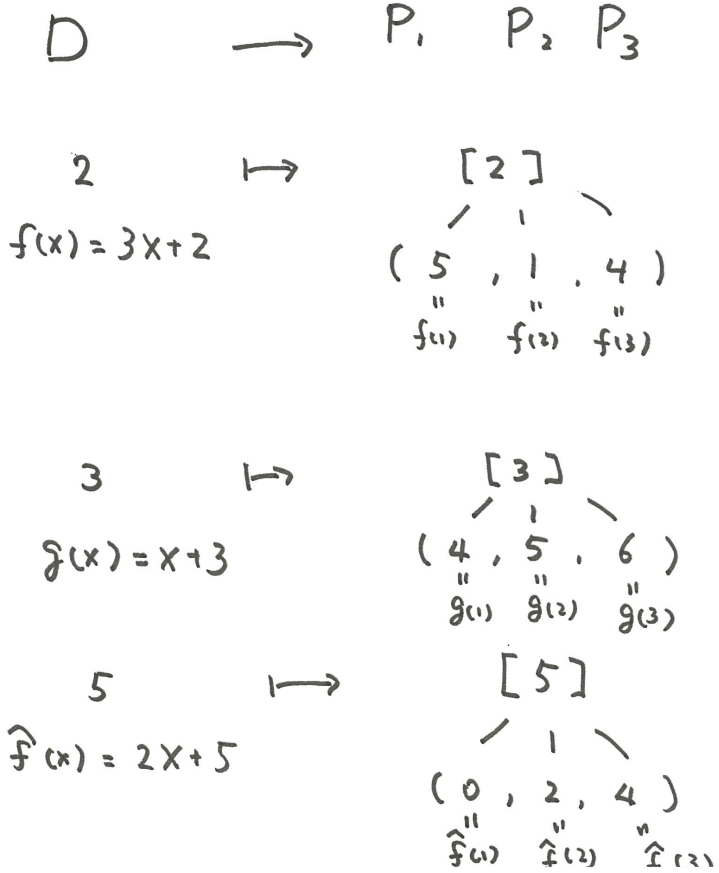
$$[a + b] = (h(\alpha_1), \ldots, h(\alpha_n))$$

$h(\alpha_i) = f(\alpha_i) + g(\alpha_i)$ であるから、各 P_i がローカルに自分のシェアを足し算すれば、$[a + b]$ と言う状態になる。

線型性

例) Shamir SS での線型性の確認

$K = \mathbb{Z}/7\mathbb{Z}$



[2]+[3]

def (5+4, 1+5, 4+6)

(f(1)+g(1), f(2)+g(2), f(3)+g(3))

(2, 6, 3)

= [5] ← 秘密を復元した時に同じという意味

= (0, 2, 4)

Shamir SS の掛け算（試み）

掛け算は少し工夫がいる。

multiplication

$[a]_{(t)}$, $[b]_{(t)}$: $a, b \in K$ が $P_1, \ldots, P_n$ 間で $(t+1, n)$-SS でシェアされた状態.

$$[a]_{(t)} \cdot [b]_{(t)} = [ab]_{(2t)}$$

が成り立つ。ただし、$[a] \cdot [b] := (a_1 b_1, \ldots, a_n b_n)$ と定義（各 P_i がローカルに自分のシェアを掛け合わせた状態）。

f, g をそれぞれ a, b をシェアする時の t 次多項式とする。$f_0 = a$, $g_0 = b$ で、

$$f(X) = f_0 + f_1 X + \ldots + f_t X^t, \text{ and } g(X) = g_0 + g_1 X + \ldots + g_t X^t.$$

$h(X) = f(X)g(X)$ と置くと、$h(0) = ab$ と $\deg(h) = 2t$ より

$$[ab]_{(2t)} = (f(\alpha_1)g(\alpha_1), \ldots, f(\alpha_n)g(\alpha_n))$$

よって、$P_1, \ldots, P_n$ 間で ab をシェアした状態になるが、$2t+1$ 個のシェアが集まらないと ab が復元できない。

Shamir SS の掛け算（アイデア）

multiplication

Lagrange と線型性により

$$[ab] = [h(0)] = [\sum_{i=1}^{n} \lambda_{i,n} h(\alpha_i)] = \sum_{i=1}^{n} \lambda_{i,n} [h(\alpha_i)]$$

$(\lambda_{1,n}, \ldots, \lambda_{n,n})$ は $\{\alpha_1, \ldots, \alpha_n\}$ のみから決まる。

$h(X) = f(X)g(X)$ であるから、$h(\alpha_i) = f(\alpha_i)g(\alpha_i)$. よって、$[a]$, $[b]$ の状態から、$[ab]$ を作るには各 P_i がローカルに $a_i b_i = f(\alpha_i)g(\alpha_i)$ を計算した後、その値のシェアを他の参加者に $(t+1, n)$-線型秘密分散で配り $[f(\alpha_i)g(\alpha_i)]$ という状態を作り出せば良い。後は線型性からローカルな計算で $[ab]$ の状態に持っていく。

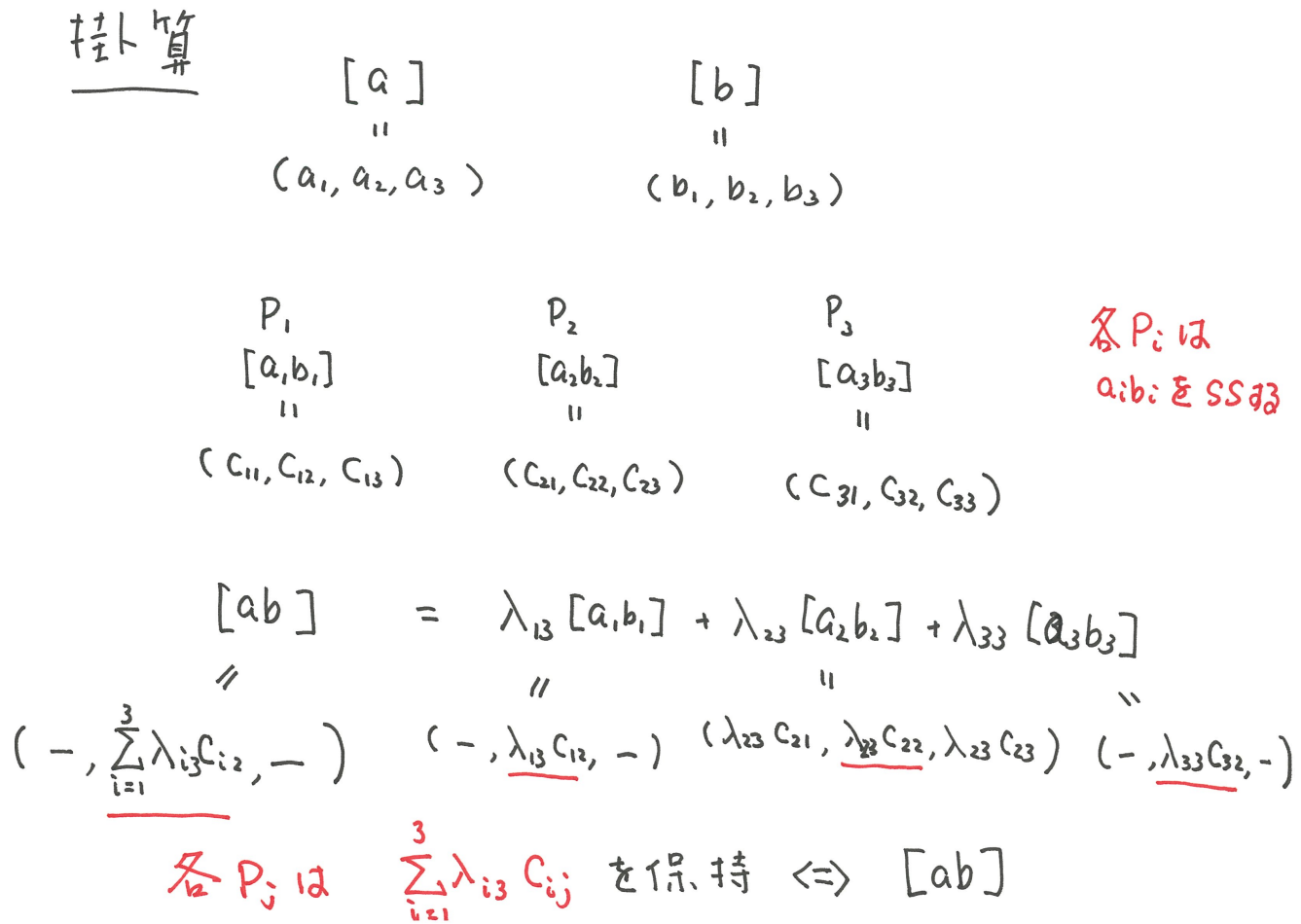
Shamir SS の掛け算（まとめ）

- Input: $[a], [b]$: a, b がそれぞれ $P_1, \ldots, , P_n$ 間でシェアされている
- Output: $[ab]$: ab が $P_1, \ldots, , P_n$ 間でシェアされる

1. 各 P_i が $c_i = a_i b_i$ をローカルに計算
2. 各 P_i が c_i を $P_1, \ldots, P_n$ 間で線型秘密分散。$[c_1], \ldots, [c_n]$ の状態になる。P_i は、$c_{1,i}, \ldots, c_{i,i}, \ldots, c_{n,i}$ を、$[c_1], \ldots, [c_n]$ の自分のシェアとして持つ。
3. $i = 1, \ldots, n$ に対して、$\lambda_{i,n} = \prod_{j \neq i} \frac{\alpha_i}{\alpha_j - \alpha_i}$ を計算する。
4. 各 P_i が $d_i = \sum_{j=1} \lambda_{j,n} c_{j,i}$ を計算する。
5. $[ab] = (d_1, \ldots, d_n)$ なので、$[ab]$ の状態になる。

例

$[a]$, $[b]$ の状態から


掛算
[a]
=
(a1, a2, a3)
[b]
=
(b1, b2, b3)
P1
[a1b1]
=
(c11, c12, c13)
P2
[a2b2]
=
(c21, c22, c23)
P3
[a3b3]
=
(c31, c32, c33)
各Piは
aibiをSSBJ
[ab] = λ13[a1b1] + λ23[a2b2] + λ33[a3b3]
=
(-, Σ_{i=1}^{3} λi3 ci2, -)
(-, λ13 c12, -)
(λ23 c21, λ23 c22, λ23 c23)
(-, λ33 c32, -)
各Pjは Σ_{i=1}^{3} λi3 cij を保持 <=> [ab]

Shamir SS による MPC の線型和と積

初期状態 $[a] = (a_1, \ldots, a_n)$, $[b] = (b_1, \ldots, b_n)$: すなわち $a, b \in K$ が $P_1, \ldots, P_n$ 間でシェアされた状態.

線形和と積

$$[a] + [b] = [a + b], \qquad -[a] = [-a]$$

線型性から、各 P_i がローカルに $a_i + b_i$ を計算すれば、$[a + b]$ という状態になる。また各 P_i が $-a_i$ を計算すれば $[-a]$ になる。

$$[ab] = [\sum_{i=1}^{n} \lambda_{i,n} a_i b_i] = \sum_{i=1}^{n} \lambda_{i,n} [a_i b_i].$$

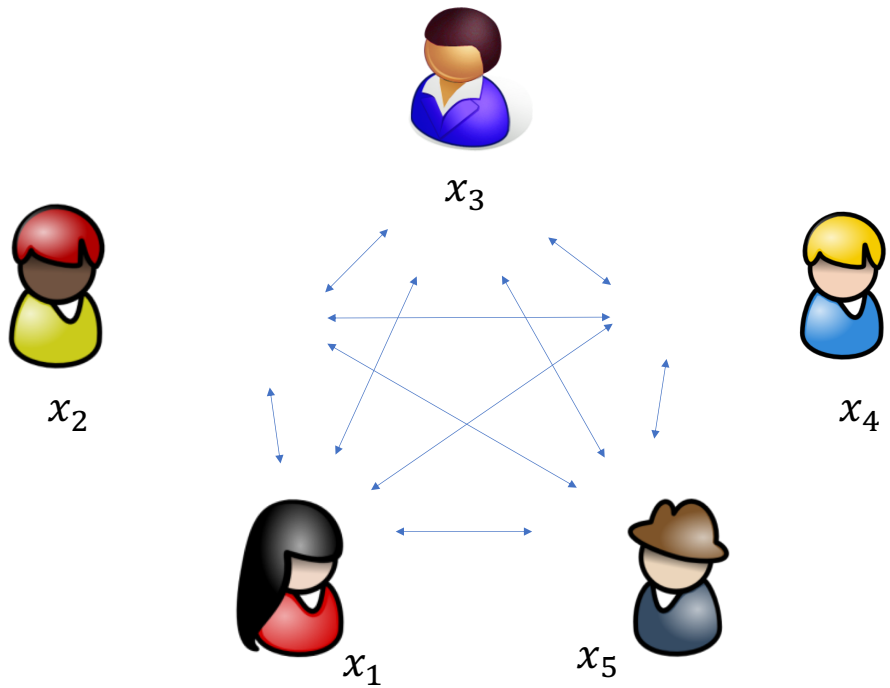
$(\lambda_{1,n}, \ldots, \lambda_{n,n})$ は $\{\alpha_1, \ldots, \alpha_n\}$ のみから決まる。各 P_i がローカルに $a_i b_i$ を計算後、その値を $(t+1, n)$-線型秘密分散で $[a_i b_i]$ を行う。後は線型性からローカルな計算で $[ab]$ という状態にできる.

本日の講義の内容

1. Shamir 秘密分散（復習）
2. 足し算と掛け算（復習）
3. 耐受動的攻撃安全なマルチパーティ計算（$t < n/2$ の場合）
4. 証明（$t < n/2$ の場合）
5. $t \geq n/2$ の場合

マルチパーティ計算

- 参加者: $P_1, \ldots, P_n$.
- 各 P_i への秘密の入力: $x_i \in \{0,1\}^\lambda$
- 全参加者への入力（公開情報）: 関数 $F : \{0,1\}^{n\lambda} \to \{0,1\}^*$.
- 各 P_i への出力：$F(x_1, \ldots, x_n)$. より一般的には、参加者ごとに違う出力をすることも許す.
- ネットワーク: Pair-wise private & synchronized.



Secure MPC against Passive Adversaries

- 参加者: $P_1, \ldots, P_n$.
- 各 P_i への秘密の入力: $x_i \in \{0,1\}^\lambda$
- 全参加者への入力: 関数 $F : \{0,1\}^{n\lambda} \to \{0,1\}^*$.
- 各 P_i への出力：$F(x_1, \ldots, x_n)$.
- パラメータ：t, n $(t < n/2)$
- 不正者：passive, $\mathcal{A}_{t,n}(\triangleq \{A \subset \{1, \ldots, n\} \mid \#A \leq t\})$.

Theorem 1

There is an efficient MPC protocol to evaluate any efficiently computable function F such that the following conditions hold:

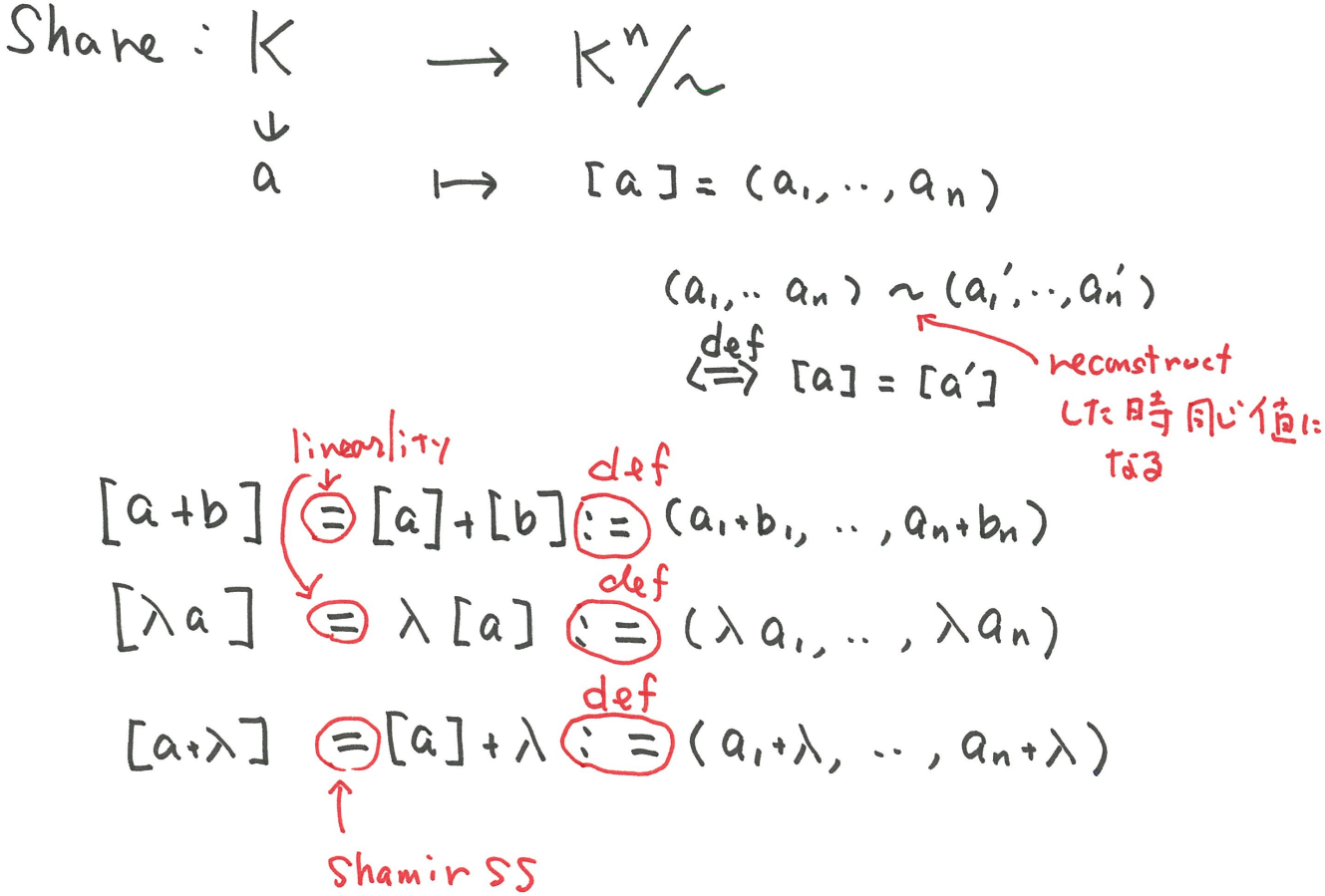
- *(Perfect Correctness) All players receive $F(x_1, \ldots, x_n)$ with prob. 1.*
- *(Perfect Privacy) Any passive $\mathcal{A}_{t,n}$-adversary with $t < n/2$ learns no information beyond $\{x_i\}_{i \in \mathcal{A}_{t,n}}$ and $F(x_1, \ldots, x_n)$ from executing the protocol regardless of their computing power and memory.*

MPC の構成（準備）

準備

- $F : \{0,1\}^{n\lambda} \to \{0,1\}^*$: 多項式時間計算可能関数
- C_F: F を計算する多項式サイズの AND, OR, NOT で構成された論理回路
- K: $n < \#K$ な有限体
- C_F の論理演算子への入力 $b \in \{0,1\}$ を $b \in \{0,1\} \subset K$ と K の元とみなす。
- AND, OR, NOT の論理ゲートを K 上の演算に置き換える。
  - $b \wedge b' \iff b \cdot b' \in K$.
  - $b \vee b' \iff b + b - b \cdot b' \in K$.
  - $\neg b \iff 1 - b \in K$.

線形性と ShamirSS の性質があると

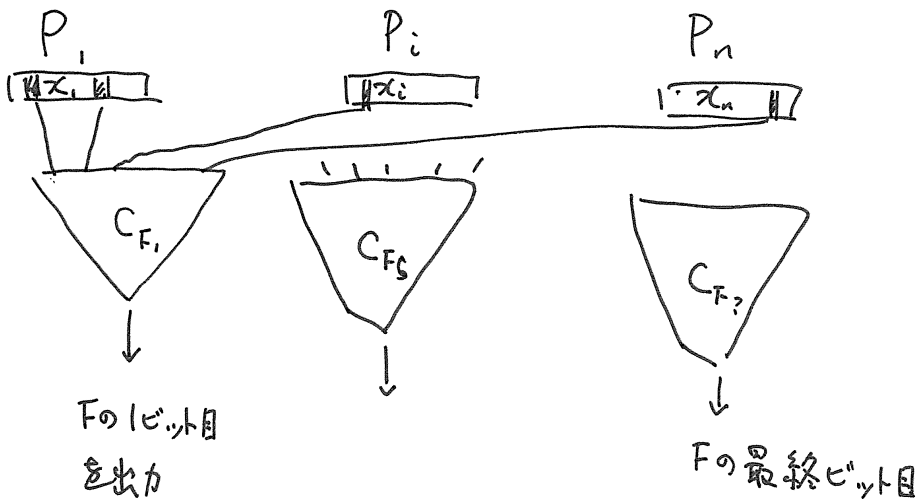


関数 F の回路

$F : \{0,1\}^{\lambda} \times \cdots \times \{0,1\}^{\lambda} \to \{0,1\}^{*}$ ← 任意の有限ビット列の集合

(n)

回路に分解



MPC の構成

- Input sharing phase: 各参加者 P_i は、x_i の各ビット b を $(t+1, n)$-Shamir SS を使い $[b]$ の状態にする。この結果、全ての参加者の秘密の全てのビットが（線型）秘密分散される。
- Computation phase: 各論理ゲートを秘密分散した形で実行
  - $[a] + [b] = [a + b]$
  - $1 - [a] = [1 - a]$
  - $[ab] = \sum_{i=1}^{n} \lambda_{i,n} [a_i b_i]$
- Output reconstruction phase: 出力ゲート結果が秘密分散された状態になっているので、各 P_i は自分の出力ゲート結果に関するシェアを他の参加者に公開（全員に送る）。各 P_i は、出力結果を復元する。

本日の講義の内容

1. Shamir 秘密分散（復習）
2. 足し算と掛け算（復習）
3. 耐受動的攻撃安全なマルチパーティ計算（$t < n/2$ の場合）
4. 証明（$t < n/2$ の場合）
5. $t \geq n/2$ の場合

安全性

- Perfect Correctness: Passive adversary より、自明。
- Perfect Privacy: 不正者（達）は、次の二種類の情報*を正直な参加者から受け取る。
  - Type1: Input sharing phase と multiplication 計算時の、正直な参加者が持つ秘密 $[b]$ のシェア（$\{b_i\}_{i\in\mathcal{A}_{t,n}}$）
  - Type2: Output reconstruction phase での出力 y の全てのシェア。

  プライバシーが保たれる直感的な説明。
  - Type1 での情報 $\{b_i\}_{i\in\mathcal{A}_{t,n}}$ は、実際の秘密 b と独立。よってなにも秘密は漏れていない。
  - Type2 の場合：ある t 次元多項式 f により、$y = f(0)$ となっており、不正者達は $\{f(\alpha_i)\}_{i\in\mathcal{A}_{t,n}}$ を共有している。今不正者の数がちょうど t とする。すると $\{f(\alpha_i)\}_{i\in\mathcal{A}_{t,n}}$ と $f(0)$ から、多項式 $f(X)$ が復元できるので、正直な参加者から送られてきたシェアは、「余剰」情報にはならない**。

*: 線形和の時は不正者は新たな情報を正直な参加者からもらってないので、なにも情報が増えないことに注意。

**: 不正者数が t 未満のときはどうなるか自分で考えよ。

本日の講義の内容

1. Shamir 秘密分散（復習）
2. 足し算と掛け算（復習）
3. 耐受動的攻撃安全なマルチパーティ計算（$t < n/2$ の場合）
4. 証明（$t < n/2$ の場合）
5. $t \geq n/2$ の場合

$t \geq n/2$ の場合

$t < n/2$ という制限は optimal である。実際、$t \geq n/2$ の場合、実現できない関数が存在する。

Theorem 2

If $t \geq n/2$, there is a function that cannot be securely evaluated, even if the adversary is passive. Here, a function securely evaluated means that it satisfies perfect correctness and perfect privacy at the same time.

- *(Perfect Correctness) All players receive $F(x_1, \ldots, x_n)$ with prob. 1.*
- *(Perfect Privacy) Any passive $\mathcal{A}_{t,n}$-adversary with $t < n/2$ learns no information beyond $\{x_i\}_{i \in \mathcal{A}_{t,n}}$ and $F(x_1, \ldots, x_n)$ from executing the protocol regardless of their computing power and memory.*

実現できない関数

関数として、AND（または OR）は実現できない。 特別なケースとして $n = 2$, $t = 1$ の時を考える。

- P_1 の秘密を $b_1 \in \{0, 1\}$, P_2 の秘密を $b_2 \in \{0, 1\}$ とする。
- $b_1 = 0$ のとき、b_2 がなんであっても、計算結果は $0 = b_1 \wedge b_2$ であるので、もし安全なプロトコルがあるのであれば perfect privacy から、P_1 は b_2 の情報がなにもわからないはず。反対に、P_1 の perfect privacy もあるので、$b_1 = 1$ としても P_1 と P_2 の間でやりとりされた情報 τ にあう P_1 の内部乱数 ρ_1 がなければいけない。
- 計算結果は、τ のみから得られて今計算結果は 0 であったはず（もともと P_1 の入力が 0 だから）。しかし、$b_1 = 1$ でも対応する P_1 の内部乱数 ρ_1 があるので、perfect correctness から P_2 の入力は $b_2 = 0$ でなければ行けなくなって P_2 の perfect privacy がなくなってしまう。よって矛盾。
- 逆に perfect correctness から始めると、$b_1 = 1$ かつ τ にあう ρ_1 が存在しないことになって、P_1 の perfect privacy がないことになってしまう。

実現できない関数（その２）

$n > 2$ の場合は、2 人参加者の場合に帰着する。

- n 人の参加者を、$\#V_1, \#V_2 \leq t$ となる重ならない二つのグループ V_1 と V_2 に分ける。$t \geq n/2$ なのでこれは可能。
- V_1 が P_1 の V_2 が P_2 の役割を果たすことで、2 人参加者のケースに帰着できる。

AND ではなく、OR も実現できないことを示せ。