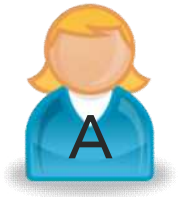
セキュリティプロトコル

コイントス

- ネットワークを介して2者間で公平なコイントスを実現したい



（準備）安全なハッシュ関数*h()*を決める

（1）乱数*x*を決める

（2）*y=h(x)* を計算する

（3） *y* を送る

（4）*x*が偶数か奇数かを予想する

（5）予想を送る

（6）*x*の値と共に予想が正しいか否か送る

（7）*h(x)=y* を確認する

秘密分散プロトコル（分散フェーズ）

- 秘密情報sをN人で分散保管する。N人中t人以上が協力すれば秘密情報の復元が可能。

1. 大きな素数pを選ぶ。秘密情報$s(<p)$とする。
2. 乱数　$a_1, a_2, \cdots, a_{t-1} \in F_p$　を選び、多項式
$$f(x) = s + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$$
を作る。
3. i番目（$i=1,2,\ldots,N$）の参加者に
$$v_i = f(i) \pmod p$$
を配布する。

秘密分散プロトコル（復元フェーズ）

1. 復元に協力する参加者（*t*人）の番号を $i_j (j = 1,2,\cdots,t)$ とする。
2. 各人の保管情報 v_{i_j} を用いて $f(i_j) = v_{i_j} (j = 1,2,\cdots,t)$ を満たす*f(x)*をラグランジェの補間公式等を用いて求める。
3. *s=f(0)* により秘密情報を復元する

ラグランジェの補間公式

- 点 $(x_i, y_i)(i = 1,2,\cdots,t)$ を通る高々*t-1*次の多項式*f(x)*を求める。

$$f(x) = \sum_{i=1}^{t} y_i \lambda_i(x)$$

$$\lambda_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{x - x_j}{x_i - x_j}$$

*t=3*の場合の例

$$f(x) = \frac{y_1(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + \frac{y_2(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + \frac{y_3(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

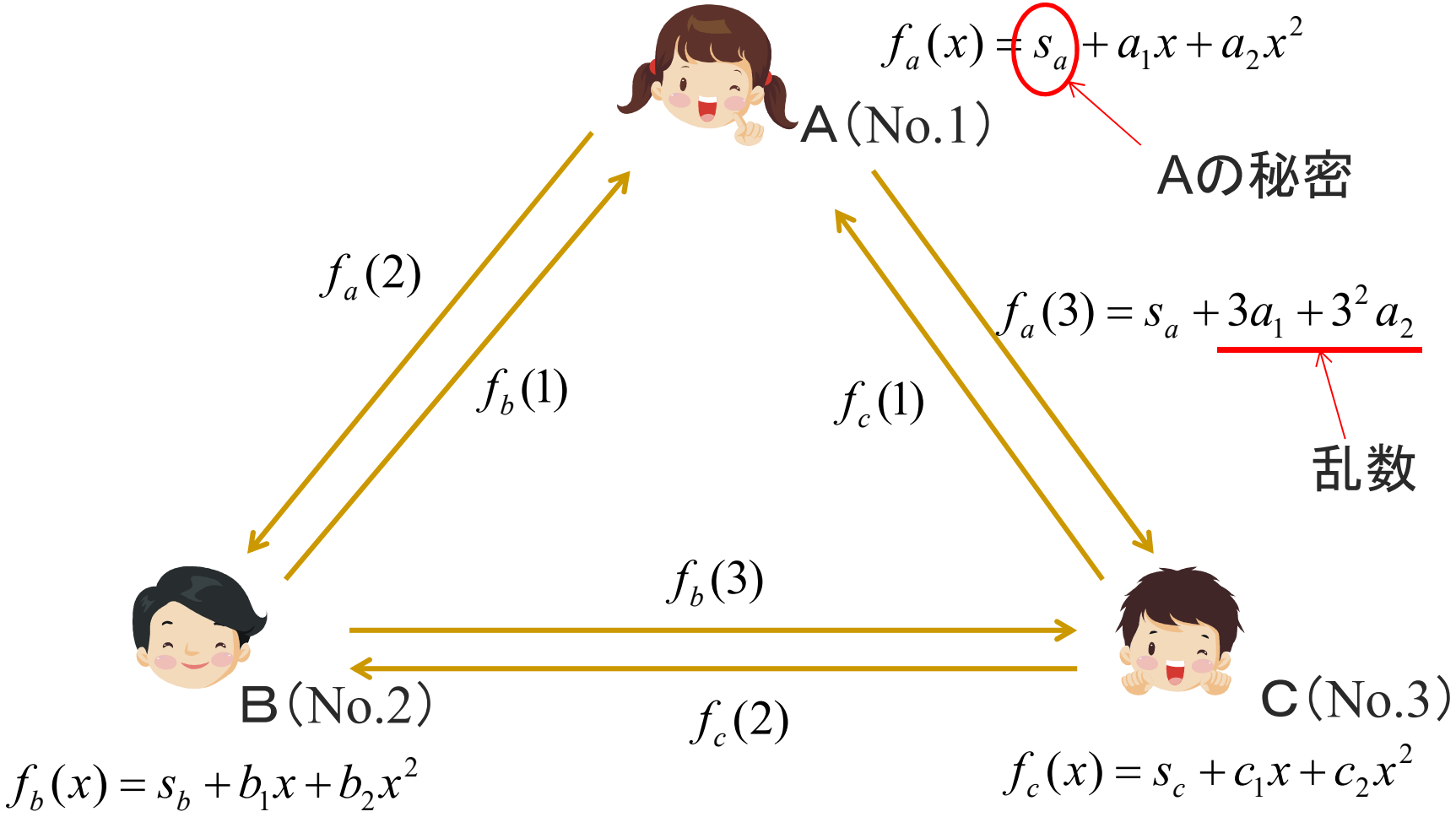
マルチパーティープロトコル

- 複数人（*N*人）の参加者が所有する秘密情報に関して、その秘密情報を明かさずに秘密情報に関する何らかの計算をするプロトコル
- 具体例として、*N*人の参加者の秘密情報 $s_i (i = 1,2,\cdots,N)$ の総和を求めるプロトコルを示す。あらかじめ、総和よりも十分大きい素数*p*を決めておく。

マルチパーティープロトコル（総和）

1. *i*番目(*i=1,2,…,N*)の参加者*Ai*は$f_i(0) = s_i$を満たす高々*N-1*次のランダムな多項式 $f_i(x)$ を作る。
2. $v_{i,j} = f_i(j) (\mathrm{mod}\ p)(j = 1,2,\cdots,N)$を求め、これを参加者*Aj*に送る。
3. 各参加者*Aj*は、$v_j = v_{1,j} + v_{2,j} + \cdots + v_{N,j} (\mathrm{mod}\ p)$を求め、それを参加者全員に公開する
4. 各 $v_j (j = 1,2,\cdots,N)$から $f(j) = v_j (\mathrm{mod}\ p)$を満たす高々*N-1*次の多項式*f(x)*を求める。*S=f(0)*により秘密情報の総和が求まる。

マルチパーティープロトコルの例（N＝3）



（注）全ての演算は、法pで行う

マルチパーティープロトコルの例（N＝３）（続き）

A（No.1）

$$v_a = f_a(1) + f_b(1) + f_c(1)$$
$$= (s_a + s_b + s_c) + (a_1 + b_1 + c_1) + (a_2 + b_2 + c_2)$$

B（No.2）

$$v_b = f_a(2) + f_b(2) + f_c(2)$$
$$= (s_a + s_b + s_c) + 2(a_1 + b_1 + c_1) + 2^2(a_2 + b_2 + c_2)$$

C（No.3）

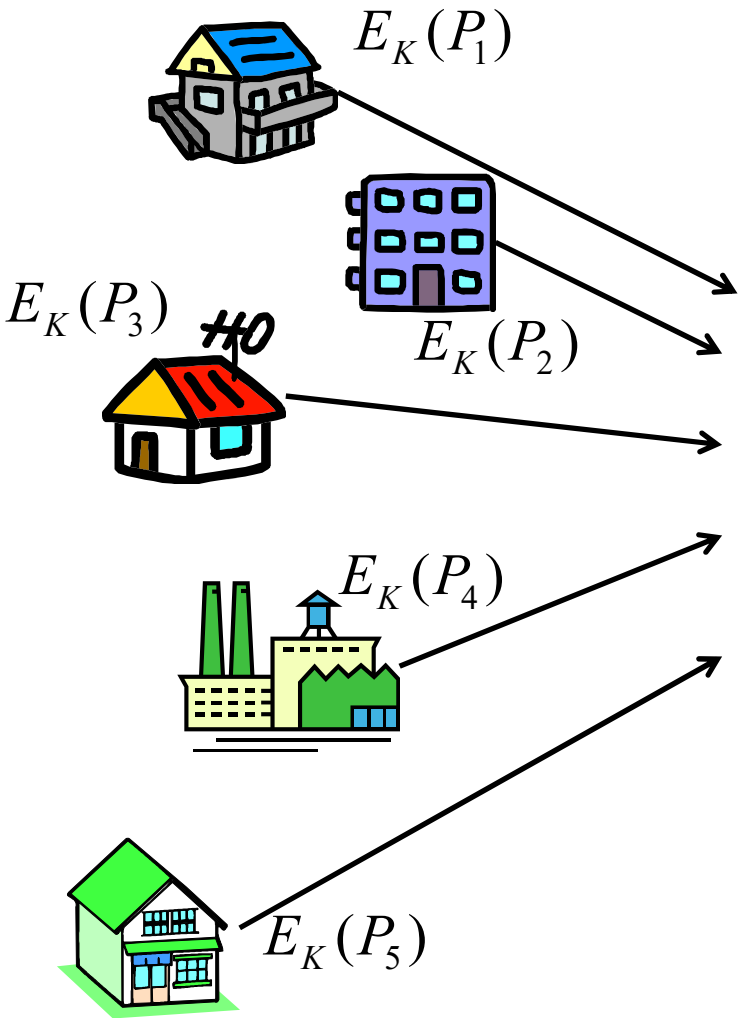
$$v_c = f_a(3) + f_b(3) + f_c(3)$$
$$= (s_a + s_b + s_c) + 3(a_1 + b_1 + c_1) + 3^2(a_2 + b_2 + c_2)$$

$(1, v_a), (2, v_b), (3, v_c)$ を通る高々２次の関数f(x)を求めると:

$$f(x) = (s_a + s_b + s_c) + (a_1 + b_1 + c_1)x + (a_2 + b_2 + c_2)x^2$$

従って、 $f(0) = s_a + s_b + s_c$ ←―― 秘密情報の総和

準同型暗号を用いたプライバシー保護



暗号化したままで
計算処理が可能

MDMS

$E_K(\sum_i P_i)$

EMS

$\sum_i P_i$

$E_K(P_1) \circ E_K(P_2) = E_K(P_1 \bullet P_2)$

演算●が加法の時→加法性準同型暗号
演算●が加法も乗法も可能な時→完全準同型暗号

（例）RSA暗号　$M_1^e \times M_2^e = (M_1 \times M_2)^e \bmod n$

ブラインド署名

メッセージ m,
乱数 r

$x = m \cdot r^e \bmod n$
を計算する

RSA暗号
秘密鍵d

x を送る →

$y = x^d \bmod n$
を計算する

← y を返す

$S = \frac{y}{r} = m^d \bmod n$
を計算する

ユーザは署名者にメッセージ内容を明かさずに署名を計算してもらえる

しきい値署名

- N人中K人が署名をすると、全体として有効な一つの署名になる