コンテンツセキュリティ　　2020 年 6 月 8 日

秘匿計算

執筆者: 安永憲司

友達同士でテストの点数を比べたい．でも点数は教えたくない．どうすればよいだろうか．点数をつけた先生に教えてもらうのが一つだが，余計なことを言われそうで嫌である．自分たちでなんとかしたい．情報を隠すといえば暗号化だが，点数を暗号化したとして，それをどう使えばよいかわからない．このように，入力を隠して，計算結果だけを知ることを，秘匿計算 (secure computation) やマルチパーティ計算 (multiparty computation) という．

この例は，ひとりの点数が x，もうひとりの点数が y のとき，以下の関数

$$f(x,y) = \begin{cases} 1 & x > y \\ 0 & x \leq y \end{cases}$$

の値を，相手に入力を知られずに計算することに対応する．秘匿計算では，入力を秘密にすること以外にも，相手と同じタイミングで結果を知ることや，必ず計算結果を得られることなど，様々な安全性を考えることができる．まずは，入力 x および y を相手に秘匿して計算結果 $f(x,y)$ を得ること考えよう．

1　目隠し関数表

目標を少し簡単にして，点数の比較ではなく，点数がともに 50 点以上あるかどうかを知りたいことにしよう．ともに 50 点以上はほぼ明らかかもしれないが，その後，60 点，70 点と点数を変えれば，いずれは大小が明らかになるのでこれで十分である．この計算は，2 ビット入力 1 ビット出力の関数で表現できる．入力は，50 点以上あるとき 1，なければ 0 とし，出力は，ともに 50 点以上あれば 1，そうでなければ 0 とする．この関数を表に表すと次のとおりである．

入力 1	入力 2	出力
0	0	0
0	1	0
1	0	0
1	1	1

図 1　関数表

この表は，論理積 (logical conjunction, AND) の真理値表に対応する．ここではまず，論理積を秘匿計算する方法を紹介する．

実現のためのアイディアを簡単にまとめると，ひとりが関数表の「情報」を隠した「目隠し関数表」(garbled table) をつくり，もうひとりが目隠し関数表から，必要な部分，つまり，計算結果だけを取り出せるようにすることである．

はじめに，表にある値の情報を隠すため，表中の入力 1 の 0 と 1，入力 2 の 0 と 1，出力の 0 と 1，それぞれをランダムに選んだラベル $x_0, x_1, y_0, y_1, z_0, z_1 \in \{0,1\}^\ell$ で置き換える．（論理積計算のためには出力ラベルは必要なく，$z_0 = 0, z_1 =$ としてよいが，後の拡張のためラベルで表す．）次に，出力ラベルを暗号化し，復号を許された人だけが取り出せるようにする．最後に，行をランダムに入れ替えることで，並び順に関する情報を隠す．ただし，行の入れ替えは，すべての行でランダムに入れ替える必要はない．入力情報を隠せばよいため，入力 1 については，x_0 を含む行をそれに対応する x_1 を含む行に入れ替えるか否か，入力 2 についても，y_0 を含む行をそれに対応する y_1 を含む行に入れ替えるか否かをランダムに決めればよい．入力 1, 2 について，入れ替えるか否かをそれぞれ $a, b \in \{0,1\}$ で表すと，目隠し関数表は図 2 のようにつくられる．ここでは，$a = 1, b = 0$ であり，x_0 と x_1 に関する行が入れ替わっており，y_0 と y_1 に関する行はそのままである．

最終的なラベルは，入力 1 については，$X_0 = (x_a, 0)$ と $X_1 = (x_{1-a}, 1)$，入力 2 は，$Y_0 = (y_b, 0)$ と $Y_1 = (y_{1-b}, 1)$ とする．図 2 では，$X_0 = (x_1, 0)$, $X_1 = (x_0, 1)$, $Y_0 = (y_0, 0)$, $Y_1 = (y_1, 1)$ である．このようにすれば，入力ラベ

入力 1	入力 2	出力		出力暗号文		行入れ替え暗号文
x_0	y_0	z_0		$\mathsf{Enc}_{(x_0,y_0)}(z_0)$		$\mathsf{Enc}_{(x_1,y_0)}(z_0)$
x_0	y_1	z_0	$\Rightarrow$	$\mathsf{Enc}_{(x_0,y_1)}(z_0)$	$\Rightarrow$	$\mathsf{Enc}_{(x_1,y_1)}(z_1)$
x_1	y_0	z_0		$\mathsf{Enc}_{(x_1,y_0)}(z_0)$		$\mathsf{Enc}_{(x_0,y_0)}(z_0)$
x_1	y_1	z_1		$\mathsf{Enc}_{(x_1,y_1)}(z_1)$		$\mathsf{Enc}_{(x_0,y_1)}(z_0)$

図 2 目隠し関数表のつくり方

ルの最後のビットを見ることで，対応する行がわかる．つまり，1 行目は (X_0, Y_0)，2 行目は (X_0, Y_1)，3 行目は (X_1, Y_0)，4 行目は (X_1, Y_1) が対応する．一方で，入力値の情報はラベルで隠され，行の入れ替えにより暗号文の並び方からも情報はもれない．

入力 1 に対応する人が目隠し関数表をつくるとする．入力 2 に対応する人は関数表を受けとり，あとはその人が計算結果を取り出せばよい．これはどのように実現すればよいだろうか．その前に，暗号化関数 $\mathsf{Enc}_{(x_i,y_j)}(\cdot)$ について説明をしていなかった．ここでは，ラベル (x_i, y_j) を知っている人だけが暗号化および復号ができるようにする．関数表をつくる人はラベルを選んでいるので問題なく暗号化できる．その後，入力に対応するラベル (x_i, y_j) を復号する人に知らせればよい．自身の入力である入力 1 のラベル x_i については，それをそのまま伝えればよい．問題は，復号側の入力である入力 2 のラベル y_j をどのように伝えるかである．これは，紛失通信 (oblivious transfer) プロトコルで実現できる．3 節を参照しよう．復号する人は，y_j を受けとり，対応する行の暗号文を復号することで計算結果を得る．

あとは暗号化関数 $\mathsf{Enc}_{(x_i,y_j)}(\cdot)$ のつくり方である．ラベル (x_i, y_j) を知っていれば暗号化と復号ができ，知らない人は $\mathsf{Enc}_{(x_i,y_j)}(z)$ から z の情報がもれないようにしたい．色々な方法があるが，擬似ランダム関数 $f : \{0,1\}^\ell \times \{0,1\}^{|t|} \to \{0,1\}^\ell$ を使って $\mathsf{Enc}^t_{(x_i,y_j)}(z) = f_{x_i}(t) \oplus f_{y_j}(t) \oplus z$ とするのがひとつである．ここで，$t \in \{0,1\}^{|t|}$ はタグであり，隠す必要はなく適当な値にすればよいが，あとで回路に拡張する場合は，素子の通し番号などを指定することになる．擬似ランダム関数の性質より，鍵 x_i, y_j を知らなければ，入力 z は隠されている．他にも，ランダムオラクル $H : \{0,1\}^{2\ell+|t|} \to \{0,1\}^\ell$ を使って $\mathsf{Enc}^t_{(x_i,y_j)}(z) = H(x_i, y_i, t) \oplus z$ としてもよい．擬似ランダム置換であるブロック暗号 E による二重暗号化 $\mathsf{Enc}^t_{(x_i,y_j)}(z) = (t, E_{x_i}(E_{y_j}(z)))$ でもよい．

2 プロトコルの安全性

上記のような複数人による計算のやり取りをプロトコルと呼ぶ．プロトコルのような対話計算は，対話型 Turing 機械でモデル化できる．二者間のプロトコルは，対話型 Turing 機械の組 (A, B) で表すことができる．通常の Turing 機械と同様，A と B は入力をもつことができ，コイン投げもできる．異なるのは，通信内容を書いたり，それを読んだりすることができる点である．これらは，一度書くと書き換えることはできない．対話計算では，ラウンドが $1, 2, ...$ とあり，各ラウンドでは片方の機械しか動作できないものとする．通常の計算と同様，出力や停止もできる．プロトコル (A, B) において，A は入力 x_A を受けとり，乱数列 r_A を使って計算をし，B は同様に，x_B を受けとり，乱数列 r_B を使って計算を行ったとする．プロトコル実行時に，A が受け取ったメッセージの列を順に $m_A = (m^1_A, m^2_A, \dots)$，$B$ が受けとったものを $m_B = (m^1_B, m^2_B, \dots)$ とする．このとき，$v_A = (x_A, r_A, m_A)$ を A の視野と呼び，$\mathsf{view}_A(A(x_A), B(x_B))$ で表す．これは A がプロトコル実行時に見ることができた情報すべてを表している．また，A の出力は，視野から決まるため，$\mathrm{out}_A(v_A)$ と表す．B の視野と出力も同様に定める．

プロトコルの安全性を定義するときに役に立つのが，意味的な安全性の定義である．模倣できれば情報が漏れていないと考える．ここでは，プロトコルに従いながら情報（相手の入力）を読み取ろうとする**準正直な** *(semi-honest, honest-but-curious)* 敵対者に対する安全性を考える．とても弱い安全性に思えるが，これを満たすプロトコルを，**悪意のある** *(malicious)* 敵対者に対して安全なものに変換する方法もあり，まず考えるべき安全性である．

定義 1（準正直な敵対者に対する安全性） プロトコル (A, B) が関数 $f(x_A, x_B) = (f_A(x_A, x_B), f_B(x_A, x_B))$ を準正直な敵対者に対し安全に計算するとは，非一様確率的多項式時間アルゴリズム S_A, S_B が存在し，任意の入力

$x_A, x_B \in \{0,1\}^n$ に対し，

$$\{v_A, \mathrm{out}_A(v_A), \mathrm{out}_B(v_B)\}_n \approx_c \{S_A(x_A, f_A(x_A, x_B))), f(x_A, x_B)\}_n,$$
$$\{v_B, \mathrm{out}_A(v_A), \mathrm{out}_B(v_B)\}_n \approx_c \{S_B(x_B, f_B(x_A, x_B)), f(x_A, x_B)\}_n$$

を満たすときである．[*1]ただし，$v_A = \mathsf{view}_A(A(x_A), B(x_B))$, $v_B = \mathsf{view}_B(A(x_A), B(x_B))$ である．

アルゴリズム S_A は，A の入力と出力の組 $(x_A, f_A(x_A, x_B))$ から，A の視野 v_A を模倣することができている．これは，A が見た情報には，入出力の組 $(x_A, f_A(x_A, x_B))$ からわかること以上に x_B に関する情報は含まれていないことを保証する．また，比較する確率変数として $(\mathrm{out}_A(v_A), \mathrm{out}_B(v_B))$ と $f(x_A, x_B)$ も加えていることから，プロトコルの出力 $(\mathrm{out}_A(v_A), \mathrm{out}_B(v_B))$ が正しい出力 $f(x_A, x_B)$ に一致するという正当性要件も同時に要求している安全性である．

左辺は，現実のプロトコルの実行で得られる情報を，右辺は，入力と出力しか手に入らないという理想的な状況で得られる情報を表しているといえる．そのため，このような枠組みによって与えられる安全性を**理想・現実** *(ideal/real)* **パラダイム**，もしくは，模倣による安全性であることから**模倣** *(simulation)* **パラダイム**と呼ぶ．

3 紛失通信

紛失通信 (oblivious transfer) は二者間プロトコルであり，入力として送り手は $x_0, x_1 \in \{0,1\}^\ell$，受け手は $b \in \{0,1\}$ をもっている．プロトコルを実行後，受け手は x_b だけを手に入れるというものである．つまり，送り手は x_0, x_1 の 2 つを入力したにも関わらず，受け手が選んでいない x_{1-b} の方は紛失してしまう．関数としては，二つの出力を持つ関数 $f_{\mathsf{OT}}((x_0, x_1), b) = (\perp, x_b)$ で表すことができる．

準正直な敵対者に対して安全な紛失通信プロトコルを，公開鍵暗号方式 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ をもとにつくる．この方式には，通常の安全性に加え，秘密鍵なしで正当な公開鍵をランダムにサンプルできる性質が必要である．

プロトコル $\Pi_{\mathsf{OT}} = (S, R)$

送り手 S は入力 $x_0, x_1 \in \{0,1\}^\ell$，受け手 R は入力 $b \in \{0,1\}$ をもつ．

1. R は，$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$ を実行し，公開鍵空間 $\mathcal{PK}$ から一様ランダムに pk' を選ぶ．$b = 0$ のとき (pk, pk') を，$b = 1$ のとき (pk', pk) を S へ送る．
2. S は，(pk_0, pk_1) を受けとり，暗号文 $c_0 \leftarrow \mathsf{Enc}_{pk_0}(x_0)$ と $c_1 \leftarrow \mathsf{Enc}_{pk_1}(x_1)$ を R へ送る．
3. R は，(c_0, c_1) を受けとり，復号結果 $\mathsf{Dec}_{sk}(c_b)$ を出力する．

図 3 公開鍵暗号にもとづく紛失通信プロトコル

命題 2 公開鍵暗号方式 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ が選択平文安全であるとき，プロトコル Π_{OT} は，準正直な敵対者に対し安全な紛失通信である．

証明: プロトコルに従えば，R は x_b を出力することが簡単に確認できるため，$\mathrm{out}_R(S(x_0, x_1), R(b)) = x_b$ である．残るは，S と R の視野 v_S, v_R をそれぞれ模倣するシミュレータ S_S, S_R の存在を示すことである．

送り手 S の視野 v_S は，入力 (x_0, x_1)，乱数 r_S のとき，受け手 R から鍵対 (pk, pk') を受けとるため，$v_S = (x_0, x_1, r_S, pk, pk')$ である．正当な公開鍵は，秘密鍵なしでサンプルできるという性質があるため，v_S は (x_0, x_1) から簡単に模倣できる．

[*1] ここで，$\{P_n\}_n$ は，確率分布の族 $\{P_i : i \in \mathbb{N}\}$ を表し，確率分布の族 $\{P_n\}_n, \{Q_n\}_n$ に対し，$\{P_n\}_n \approx_c \{Q_n\}_n$ とは，任意の確率的多項式時間アルゴリズム D に対し，$|\Pr[D(P_n) = 1] - \Pr[D(Q_n) = 1]| \leq \mathsf{negl}(n)$ を満たすことであり，$\{P_n\}_n$ と $\{Q_n\}_n$ は計算量的に識別できないという．また，関数 $\mathsf{negl}(n)$ は，無視できる (negligible) 関数のことであり，任意の多項式 $p(\cdot)$ について，十分大きな $n > 0$ に対し，$\epsilon(n) < 1/p(n)$ を満たすとき，$\epsilon(\cdot)$ は無視できるという．

受け手 R の視野 v_R は，入力 b，乱数 r_R のとき，$v_R = (b, r_R, \mathsf{Enc}_{pk_0}(x_0), \mathsf{Enc}_{pk_1}(x_1))$ である．受け手のシミュレータ S_R は，(b, x_b) から視野 v_R を模倣すればよい．まず，受け手の手順とまったく同様に，(pk_0, pk_1) を用意する．固定した平文 x' を二つの鍵により $(\mathsf{Enc}_{pk_0}(x'), \mathsf{Enc}_{pk_1}(x'))$ と暗号化し，ランダムなビット列 r を使って $(b, r, \mathsf{Enc}_{pk_0}(x'), \mathsf{Enc}_{pk_1}(x'))$ を出力すればよい．鍵のつくり方は同じであり，暗号方式の安全性より暗号文は識別できないため，視野 v_R を模倣できている． □

4 目隠し回路

1 節では，AND 素子の計算を目隠し関数表により計算する方法を示した．この方法は，OR（論理和）素子や XOR（排他的論理和）素子，NAND（否定論理積）素子などの 2 ビット入力 1 ビット出力の任意の素子に適用できる．これらの論理素子を組み合せて論理回路として表すことで，任意の関数 $f : \{0,1\}^n \to \{0,1\}^m$ の計算にも適用できる．もちろん，大きくないサイズの回路で関数を表現しないと，効率的に計算することはできない．

基本的な手順は，1 節と同様である．二人のうち一人が目隠し回路 (garbled circuit) をつくる．与えられた論理回路の各ワイヤ i に，ランダムに選んだ二つのラベル $\ell_i^0, \ell_i^1 \in \{0,1\}^\ell$ を割り当て，それぞれ値が 0 の場合と 1 の場合に対応させる．また，ワイヤ i を入力とする素子において，その入力に対応する関数表の行を入れ替えるか否かをランダムに選んだ $b_i \in \{0,1\}$ で表し，$L_i^0 = (\ell_i^{b_i}, 0), L_i^1 = (\ell_i^{1-b_i}, 1)$ を最終的なラベルとする．これは，$x \in \{0,1\}$ を使って，$L_i^x = (\ell_i^{x \oplus b_i}, x)$ と書ける．つまり，$b_i = 0$ であれば，ℓ_i^0, ℓ_i^1 はそれぞれ L_i^0, L_i^1 に含まれるが，$b_i = 1$ のとき，その場所が入れ替わり，ℓ_i^0, ℓ_i^1 はそれぞれ L_i^1, L_i^0 に含まれる．各素子 t では，関数表の各行を暗号化し，目隠し関数表をつくる．入力ワイヤが i と j，出力ワイヤが k であるとする．ランダムオラクル H を使って暗号化するならば，$x, y \in \{0,1\}$ に対し，

$$\mathsf{Enc}^t_{(L_i^x, L_j^y)}(L_k^{g_t(x,y) \oplus b_k}) = H(\ell_i^{x \oplus b_i}, \ell_j^{y \oplus b_j}, t) \oplus L_k^{g_t(x \oplus b_i, y \oplus b_j) \oplus b_k}$$

とすればよい．ここで，関数 $g_t : \{0,1\}^2 \to \{0,1\}$ は，素子 t の演算を表している．出力ワイヤ k のラベルも，b_k により含まれる場所が入れ替わるため，$L_k^{g_t(x \oplus b_i, y \oplus b_j) \oplus b_k}$ を暗号化している．

最後に，回路の出力ラベルから出力値を復号できるようにすればよい．これは同様に，ラベル L_i^z を知っていれば値 $z \in \{0,1\}$ を復号できるように暗号化した表を用意すればよい．

定理 3 関数 H, H' をランダムオラクルとしたとき，図 4 のプロトコル Π_{GC} は関数 $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ を準正直な敵対者に対し安全に計算する．

図 5 は，二つの AND 素子 t_1, t_2 を組み合わせた 3 ビット入力 1 ビット出力の関数を計算する目隠し回路である．各ワイヤ $i \in \{1, ..., 5\}$ について，関数表における行の入れ替えを表す $b_i \in \{0,1\}$ は，$b_1 = 1, b_2 = 0, b_3 = 0, b_4 = 1, b_5 = 1$ である．

プロトコル $\Pi_{\mathsf{GC}} = (A, B)$

A は入力 $x_A \in \{0,1\}^n$，B は入力 $x_B \in \{0,1\}^n$ をもち，関数 $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ を計算する論理回路 C が与えられている．関数 H, H' はハッシュ関数.

1. A は以下を行う．
   (a) C の各ワイヤ i について，$\ell_i^0, \ell_i^1 \in \{0,1\}^\ell$ および $b_i \in \{0,1\}$ をランダムに選び，ラベル $L_i^0 = (\ell_i^{b_i}, 0), L_i^1 = (\ell_i^{1-b_i}, 1)$ を定める．
   (b) 目隠し回路の生成：C の各素子 t について，t の演算が関数 $g_t : \{0,1\}^2 \to \{0,1\}$ で与えられ，入力ワイヤが i と j，出力ワイヤが k とする．素子 t の各入力 $x, y \in \{0,1\}$ に対し，暗号文 $c_{x,y}^t = H(\ell_i^{x \oplus b_i}, \ell_j^{y \oplus b_j}, t) \oplus L_k^{g_t(x \oplus b_i, y \oplus b_j) \oplus b_k}$ を計算し，$c_{0,0}^t, c_{0,1}^t, c_{1,0}^t, c_{1,1}^t$ の順に並べる．
   (c) 出力復号表の生成：C の各出力ワイヤ k について，$z \in \{0,1\}$ に対し，暗号文 $c_z^k = H'(\ell_k^{z \oplus b_k}, k) \oplus (z \oplus b_k)$ を計算し，c_0^k, c_1^k の順に並べる．
   (d) 生成した目隠し回路と出力復号表，入力 x_A に対応するワイヤのラベルを B へ送る．
2. 入力 x_B に対応する各ワイヤ i のラベルを，紛失通信により，A から B へ送る．つまり，A は二つのラベル L_i^0, L_i^1 を入力とした送り手，B は入力 x_B で決まる値 $b \in \{0,1\}$ を入力とした受け手として実行する．
3. B は受けとった目隠し回路を評価する．各素子 t において，関数表 $\{c_{x,y}^t\}_{x,y \in \{0,1\}}$ と入力ワイヤのラベル $L_i^x = (\ell_i^{x \oplus b_i}, x), L_j^y = (\ell_j^{y \oplus b_j}, y)$ から，出力ワイヤのラベル $L_k = H(\ell_i^{x \oplus b_i}, \ell_j^{y \oplus b_j}, t) \oplus c_{x,y}$ を求める．評価後，回路の出力ワイヤのラベルと出力復号表より，回路の出力 $f(x_A, x_B)$ を求め，その値を A へ送る．A, B ともにその値を出力して停止する．

図 4　目隠し回路による秘匿計算

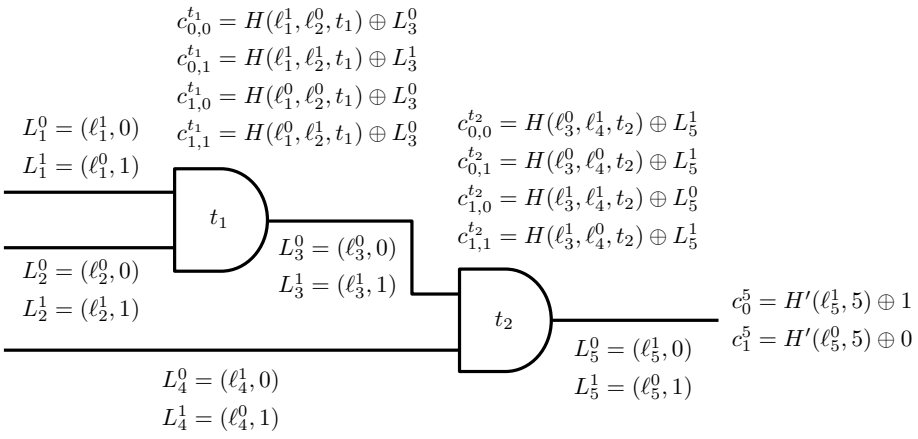

図 5　目隠し回路