特集
Special Feature

［カードベース暗号とその展開（前編）
～情報セキュリティ教育にも応用可能な身近な道具を利用した暗号技術～］

4 カードベース暗号の計算モデル

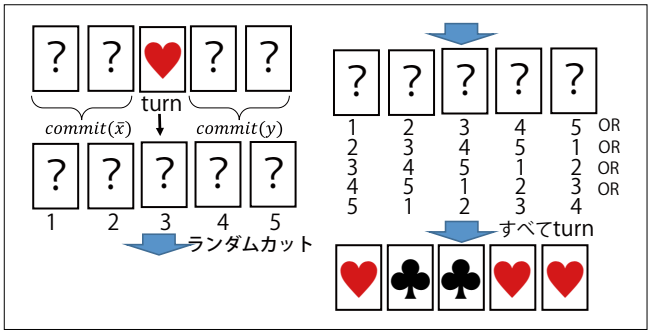
真鍋義文　工学院大学

本稿が対象とするプロトコル

入力値を明かすことなく計算結果を得る秘密計算を，トランプのような物理的なカードを用いて実現するカードベース暗号プロトコルが数多く考えられている．本稿では Five-card trick[Boe90] ☆1 に始まる，秘密計算を行うプロトコルの計算モデルの違いについて述べる．モデルの分類の基準は，使用するカードと値の表現法，入出力，実行する操作の種類，演算関数の秘匿，プレイヤーのモデル，カード以外の補助ツールである．これらのモデルの違いとプロトコルに与える影響について述べる．

入力値 $x_1, x_2, \ldots, x_n$ を明かすことなく，関数値 $f(x_1, x_2, \ldots, x_n)$ を計算することを秘密計算と呼ぶ．本稿では特に断らない限り，計算を行うプレイヤーはアリスとボブの 2 名とする．一般に，カードベース暗号プロトコルの効率として使用カードの枚数とシャッフルの回数が用いられる．使用カードの枚数は計算量理論における空間計算量に相当し，シャッフルの回数は時間計算量の最大の要因を与える．関数 f として多くのものが考えられているが，本稿では 1 ビットの入力値 $x, y \in \{0,1\}$ に対する論理積 $f = x \wedge y$ の計算を主な例とする．

☆1 参考文献のうち，本稿の末尾に示したもの以外は，宮原，水木，品川：カードベース暗号の bib ファイル，https://www.ns.kogakuin.ac.jp/manabe/card.html を参照のこと．

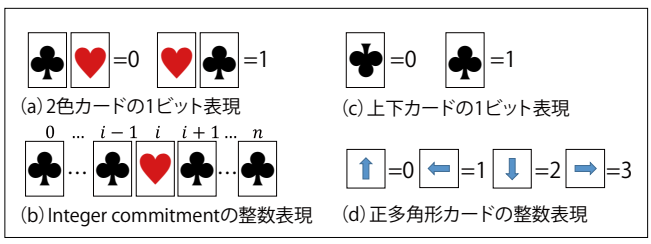


■図 -1　Five-card trick

使用するカードと値の表現法

図 -1 に示す，論理積 $f = x \wedge y$ を計算する Five-card trick[Boe90] においては♥と♣のカードが用いられた．♥のカード同士，♣のカード同士は区別がつかない．カードの裏面はすべて ? であり，区別不能である．2 枚のカード♥♣の並びで 1 ビットの値 1(true) を，♣♥の並びで 0(false) を表現する（図 -2 (a)）．このとき，論理の否定は 2 枚のカードの左右を入れ替えることで計算できる．1 ビットの値 x を表す 2 枚のカードを裏向きにしたカード対を $commit(x)$ と呼ぶ．Five-card trick を以下に示す．

1. $commit(x)$ の左右を入れ替えた $commit(\bar{x})$，♥，$commit(y)$ をこの順に並べて♥を裏向きにした



■図 -2　カードと値の表現法

5枚のカード列を作成する．

2. このカード列に対してアリスとボブが公開の場でランダムカットを行う．ランダムカットとは，図-1の右の5行の数字で示したように，結果のカードの並びが元のままの（12345）であるか，（23451），（34512），（45123），（51234）のいずれかであり，その生起確率がすべて等しい（1/5の）シャッフルである．
3. すべてのカードを表にする．♥が3枚並ぶと$f=1$，そうでないと$f=0$である．並びの判定の際に左右の両端はつながったものと見なす．図-1の右下の結果例は$f=1$である．

$f=1$の場合に♥が3枚並ぶことは自明である．$f=0$となる残り3通り$(x, y)=(0,0), (0,1), (1,0)$のカードの並びはランダムカットを行うと区別不能となることより，入力値xとyの秘密は保たれる．ここで複数の♥同士および♣同士が区別不能である性質が用いられている．

♥と♣のほか，プロトコルによっては♢や空白のカードなどが，位置を記憶するマーキングなどのために用いられることがある[TSSM24]．

♥1枚で1，♣1枚で0を表現する方式も存在するが，否定の演算の実現が困難であるため，使用例は少ない[NR98], [Shi25]．

2枚で1ビットを表現する方式は論理演算に向いているが，数値演算のプロトコルは複雑になる場合が多い．このため，2色カードで整数値を表現するinteger commitment[CK94]が用いられる．n枚の♣と1枚の♥を用いて整数値$i(0 \le i \le n)$を，左から$i+1$枚目が♥で，それ以外が♣である$n+1$枚のカード列で表現する（図-2（b））．Integer commitmentを用いた数値計算法については，本特集の別の記事で詳しく述べられている．

2色カード以外にも多数の種類のカードが用いられている．裏面は対称であるが表面は上下が非対称な上下カード♣1枚を用いて♣で1を，✤で0で表現するモデル（図-2（c））も考えられている[MS14a]．上下カードを用いたプロトコルについては，次号に掲載される本特集の後編の記事で詳しく述べられている．

上下カードの一般化として正多角形カードも考えられている．正n角形カード1枚で0から$n-1$の値を表現することができ（図-2（d）），投票などのプロトコルで用いられている[SMSN+17], [TS24]．

上記は特別なカードを用意する必要がある．これに対して，家庭で容易に入手可能なトランプやUNOのカードによる秘密計算も考えられている[NR99], [Miz16b], [SM19]．Five-card trickのように，同一のマークのカードが区別不可能であることで秘密値の安全性を実現することが多いため，区別がつくトランプカードでは手続きが複雑になる．

さらには，トランプでマークが描かれる位置に着目し，カードの一部分のみを見ることができる穴の開いた封筒にカードを入れることによる計算も考えられている．カードAとBに対して，使う封筒によってAとBとの区別不可能／区別可能を変化させることによる効率的な計算が考えられている[MM23], [HS24], [HS25]．

入力，出力のモデル

入力については秘密値入力と秘匿入力の2種類がある．前者は全プレイヤーが知らない秘密値を入力として使用する．Five-card trickのように，入力値xを表現するカードが，$commit(x)$という形で与えられる．

後者は各プレイヤーが持つ秘密の値をプロトコルに入力する．たとえばアリスはボブに知られたくない値xを持ち，ボブはアリスに知られたくない値yを持つ状況で$f=x \wedge y$を計算する場合である．

秘匿入力の場合にも，アリス（ボブ）が持つ値$x(y)$に対し，ほかのプレイヤーに見えない所で$commit(x)(commit(y))$を作成することで秘密値入力のプロトコルを実行可能である．したがって秘密値入力のプロトコルの方が汎用性が高い．しかし，秘匿入力の場合は，後述の背面処理を行うことで効率的なプロトコルが得られる場合がある．

出力についても公開出力と秘匿出力の2種類がある．Five-card trickはカードをすべて公開することで

結果を得る公開出力である．それに対し，結果 f を $commit(f)$ の形で得る秘匿出力プロトコルも存在する．結果を $commit(f)$ の形で得ることができれば，結果を用いてさらなる秘密演算を行うことが可能である．もしも結果を出力する必要があれば，$commit(f)$ のカードを表にすればよい．したがって秘匿出力プロトコルの方が汎用性が高いが，プロトコルは複雑になりがちである．Five-card trick と同じ秘密値入力，公開出力の使用カード 4 枚（最小）の論理積プロトコルが示されている [MKS12]．秘密値入力，秘匿出力の [MS09] が，使用カード数 6 枚，シャッフル 1 回の非常に効率的な論理積プロトコルである．

プレイヤーの実行する操作の種類

Five-card trick などの秘密値入力の多くのプロトコルですべての操作は公開で行われる．すべての操作が決定的であれば，入力 $commit(x)$ のカードと出力カードとの位置関係の情報から，出力値 f を公開したときに秘密の入力値（の一部）が漏洩する．したがって非決定的な操作が必要であり，カードベース暗号ではシャッフルという操作が行われる．シャッフルはカードの配置を非決定的に変更する操作であり，操作結果はどのプレイヤーにも分からない．Five-card trick ではランダムカットがそれにあたる．公開操作のプロトコルのモデル化は [MS17] で行われ，操作として turn（カードの表裏を反転），perm（カードの配置を公開で変更），shuf（シャッフル）が示されている．

秘匿と処理効率に大きくかかわるシャッフル操作としてさまざまなものが考案されてきた．n 枚のカード列 $(c_1, c_2, \ldots, c_n)$ に対するシャッフルは $S=(\Pi, F)$ で定義される．S_n を，置換 $\{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ 全体の集合とする．$\Pi \subseteq S_n$ は，S による並べ替えの集合を表す．F は Π 上の確率分布であり，$\forall \pi \in \Pi, F(\pi) > 0$ かつ $\Sigma_{\pi \in \Pi} F(\pi) = 1$ を満足する．シャッフル S を実行すると並べ替え $\pi \in \Pi$ が確率 $F(\pi)$ で行われ，どの π が実行されたかはプレイヤーには分からない．

S が閉じたシャッフルであるとは，S を 2 回実行した結果と S が等しいことである．閉じたシャッフルは以下の場合に有効である．アリスによるシャッフル S の実行時に，実行した π をアリスが不正に覚える可能性がある．たとえばランダムカットの実行時，器用なアリスはカードの移動枚数を覚えることができるかもしれない．可能であれば秘密の入力値がアリスに漏洩する．このとき，結果に対してボブがさらにシャッフル S を実行することで情報漏洩を防止できる．同様にボブが自分の実行した π を覚えられてもアリスのシャッフル結果が不明なのでボブに対する情報漏洩も起きない．

代表的なシャッフルとして，n 枚のカードを完全にランダムに並べ替えるランダムシャッフルや，前述のランダムカットがある．これらは閉じたシャッフルである．変則的なシャッフルとして，使用カード枚数 4 枚（最小）の秘密値入力，秘匿出力の論理積プロトコル [KWH15] では $F(\pi_1)=1/3$，$F(\pi_2)=2/3$ であるシャッフルが用いられている．このシャッフルをプレイヤーが手で実行するのは困難で，特殊なケースを用いた実現法が考えられている [NHMS16]．

さらに，シャッフルをカードではなくカードの束（p_1, p_2, …, p_n）に対して実行することも考えられている．各 p_i は同数のカードからなる束で，シャッフルによって p_i 内のカードの並びは変化しない．カードの束に対するランダムカットはパイルシフティングシャッフル，ランダムシャッフルはパイルスクランブルシャッフルと呼ばれる．また，偶数枚のカード列を左右の 2 つの束に分けて左右を入れ替えるか否かを確率 1/2 で実行するランダム二等分割カットは論理積の計算において使用された [MS09]．これ以外にも多数のシャッフルが提案されたが，追加カードを用いることにより，任意のシャッフルをカードだけで実現できることが示されている [SMAM+20]．

上記のモデルでは秘匿入力が考慮されていない．$commit(x)$ 以外の手段で入力を行うために背面処理が導入された．背面処理は，ほかのプレイヤーに見えない所で行う処理である．たとえばアリスがボブに

背を向けて，あるいは机の下のボブから見えない所で処理を行う．背面処理により入力と出力の位置関係を秘匿することができるので，シャッフル操作なしのプロトコルも可能となる．

背面処理が許された場合にはアリスが x，ボブが y を持つ場合に論理積 $f=x \wedge y$ を以下の単純なプロトコルで求めることができる [1), 2)]．

1. アリスは $commit(x)$ を背面処理で作成し，ボブに渡す．
2. ボブは背面処理で $y=1$ の場合はアリスのカードを出力し，$y=0$ の場合は $commit(0)$ を出力する．結果は $commit(f)$ となる．

$$x \wedge y = \begin{cases} 0 & \text{if } y = 0 \\ x & \text{if } y = 1 \end{cases}$$

より結果は正しい．公開出力の場合には使用カード枚数がさらに少ないプロトコルが存在する．

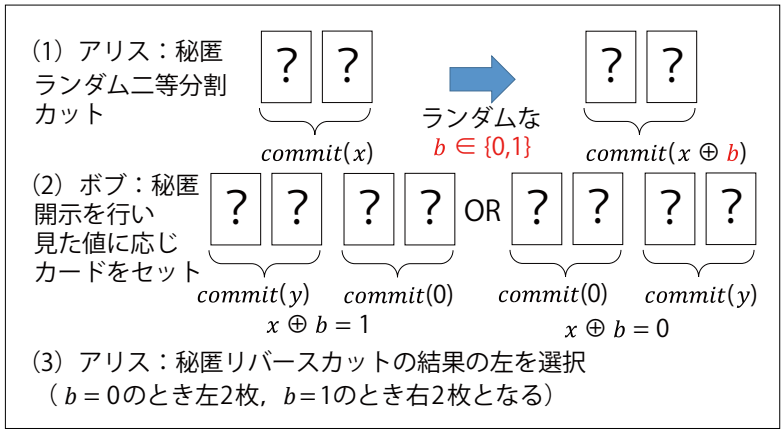
背面処理は強力なため，秘匿入力以外の背面処理を許した場合に効率的なプロトコルを求めることができるかが課題となった．まず金持ち比べのプロトコル [3)] が示された．その後，以下の 3 種類の背面処理操作を利用する論理演算プロトコル [4)] が示された．

- 秘匿ランダム二等分割カット：偶数枚のカード列に対して，秘密の乱数値 $b \in \{0,1\}$ を用いてランダム二等分割カット [MS09] を背面処理で実行する．$commit(x)$ に対して実行すると，左右を入れ替えるか否かをランダムに行い，結果は x と b との排他的論理和 $commit(x \oplus b)$ となる．
- 秘匿リバースカット：上記で選んだ乱数 b を再度用いて，偶数枚のカード列に対してランダム二等分カットを背面処理で実行する．秘匿ランダム二等分割を行ったカード列に対して実行すると，カード列を元に戻す処理となる．
- 秘匿開示：裏向けのカードを表にして自分だけ値を知る．アリスが $commit(x)$ に対して秘匿ランダム二等分割カットを行った結果のカードをボブが秘匿開示した場合，見る値は $x \oplus b$ であり，b を知らないボブは x に関する情報を得ることができない．

これらの操作を用いて $x \wedge y$ を計算するプロトコルを**図 -3** に示す．

1. アリスは $commit(x)$ に対して秘匿ランダム二等分割カットを用いて $commit(x \oplus b)$ を作成し，ボブに渡す．
2. ボブは秘匿開示を行い，$x \oplus b$ の値を知る．この値が 1 のときはカード列 ($commit(y)$，$commit(0)$) を，0 のときはカード列 ($commit(0)$，$commit(y)$) を作成してアリスに渡す．ここで，$commit(0)$ の作成には $x \oplus b$ のカードを再利用するので入力以外のカードは不要である．
3. アリスは b を用いて秘匿リバースカットを行った後，左側のカード組を出力する．

秘匿入力かつ秘匿出力の場合には turn を実行しない，すなわち実行中にカードの表面を見ることがないプロトコルを構成することができる．この性質を無開示性と呼ぶ．無開示性を持つプロトコルの安全性の証明は自明である．前述の秘匿入力論理積プロトコルはその一例である．一般にカードを表にする誤動作等により秘密の値が漏洩することを防止するには，カードを封筒などに入れて封をすることが考えられるが，turn を実行してカードの表面を見るのにも手間がかかるという副作用を持つためあまり実際的ではない．しかし無開示性を持つプロトコルではこの手間が発生しないので情報漏洩の防止が容易である．無開示性を持つプロトコルは



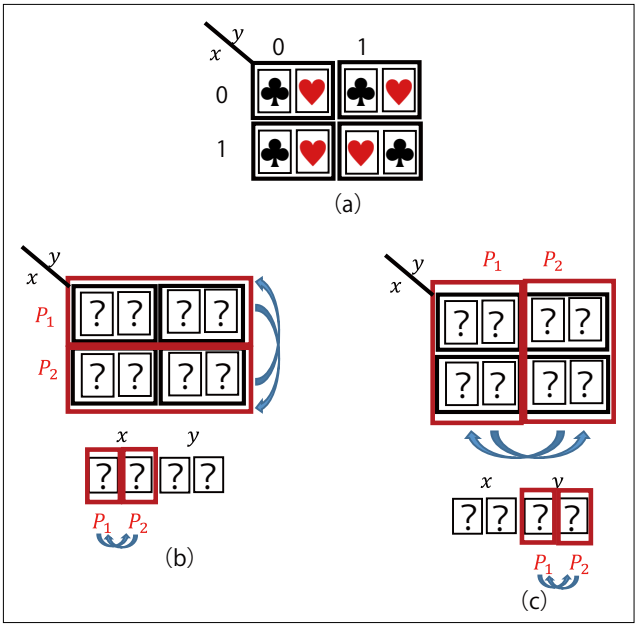
■図 -3　背面処理を用いた論理積プロトコル

多数決 [ANWI+23]，金持ち比べ [OM18a] などの問題で示されている．

関数の秘匿

Five-card trick においてプレイヤーは，論理積を計算することを知っている．計算する関数 f は公開情報とするものが多い．しかし，関数 f を秘匿したい場合もある．暗号理論分野におけるガーブルド回路の考えを用いることで，アリスが関数 f を持ち，ボブが入力値 $(x_1, x_2, \ldots, x_n)$ を持つ状況で，互いの値を明かすことなく $f(x_1, x_2, \ldots, x_n)$ を求めることができる．$f = x \wedge y$ を求める場合のプロトコル [TMM25] の概要を図 -4 に示す．

1. アリスが論理ゲートに対する真理値表を作成し，裏向けにして値を秘匿する（図 -4（a））．ボブが入力値 $commit(x)$，$commit(y)$ を与える（ここで x が 0(1) である場合に 1(2) 行目を選択し，y が 0(1) である場合に 1(2) 列目を選択すると，$x \wedge y$ の値を得る）．
2. アリスとボブは 2 つの束 P_1，P_2 を作る．$P_1(P_2)$ は真理値表の 1(2) 行目と $commit(x)$ の左（右）のカードで構成する．そして 2 人で P_1 と P_2 をランダムに入れ替えるパイルスクランブルシャッフルを行う（図 -4（b））．シャッフル後のカードを真理値表の各行と x の位置に配置する．P_1 と P_2 が入れ替わった場合には x の値が反転するが，その場合には真理値表の行も反転されるので x を用いて真理値表の正しい行を選択することができる．
3. アリスとボブは 2 つの束 P_1，P_2 を作る．$P_1(P_2)$ は真理値表の 1(2) 列目と $commit(y)$ の左（右）のカードで構成する．そして 2 人で P_1 と P_2 をランダムに入れ替えるパイルスクランブルシャッフルを行う（図 -4（c））．シャッフル後のカードを真理値表の各列と y の位置に配置する．この操作が演算結果を変えないことは上記と同様である．
4. x と y に turn を行いその値に応じた真理値表の値を出力とする．結果は $commit(x \wedge y)$ となる．



■図 -4　ガーブルド回路による論理積プロトコル

turn したカードはランダム化されているので x，y の値の秘密は保たれる．ここでは論理ゲート g が 1 つの場合を示したが，g の出力がさらに別の論理ゲート h の入力となる場合には，g の真理値表の結果の値を表す（4 組の）2 枚のカードの左右と h の真理値表の行（または列）で 2 つの束を作ってランダムに入れ替えることで複雑な論理回路に対しても同様の処理を行うことができる．ガーブルド回路の構成法は文献 5）で提案され，その後使用カード枚数を 1 ゲートあたり 8 枚 [TMM25] および 6 枚 [OSN+24] に削減したプロトコルが考案された．ほかに決定木の評価プロトコルも示されている [KM25b]．

関数秘匿の別の実現手法として，暗号理論における難読化の手法を用いるプロトコルも考案されている [KW22]．

プレイヤーのモデル

Five-card trick をはじめとする多くのプロトコルでプレイヤーは semi-honest，すなわち，プロトコルを正しく実行するが，秘密の値を知ろうとするプレイヤー

であると仮定している．多くのプロトコルはすべての操作を公開の場で行うので，プロトコルに従わない不正な操作はほかのプレイヤーによって検出可能である．ただし，背面処理を行うプロトコルでは，アリスはボブの見ていない所での不正な操作，たとえば伏せられたカードを表にして秘密の値を知るなどの不正操作が考えられる．この不正をボブは検出することはできない．これは背面処理の大きな問題点である．アリスによる背面処理による配置変更の正しさをボブが検証できるプロトコル [KW20] や，封筒を用いることで背面処理の不正を防止するプロトコル [MO22] が考案されている．

背面処理での不正に対するほかの対策として，プレイヤー数が 3 以上の場合に相互監視を行う方法が考えられている 4)，[AIO20], [MMS23]．アリス，ボブ，キャロルの 3 名で実行する場合に，アリスの背面処理をボブが監視，ボブの背面処理をキャロルが監視，キャロルの背面処理をアリスが監視する．この場合にもたとえばアリスはボブの背面処理を知ることができないことにより，入力値の秘匿は可能である．

さらには，カードにインクで印をつける不正に関する考察も行われている [SSM23]．

また，プレイヤーが semi-honest であっても操作ミスの可能性がある．公開実行プリミティブの操作ミス，たとえば perm において配置を間違える場合などは，ほかのプレイヤーが操作ミスを指摘することが可能であるが，プレイヤーが気づかない場合も考えられる．操作ミスを行った場合の情報漏洩の評価が [MK22] において行われている．

誤った turn によりカードの表面を見るミスを行った場合には情報漏洩が起きる可能性がある．情報漏洩を防止するために，暗号理論における秘密分散の考えを利用したプロトコルも考えられている [TMMS21]．秘密値 x に対して乱数 r を選び，シェア $x \oplus r$ と r の組で 1 ビットの値を保持する．これにより，どちらか 1 組（2 枚）のカードが誤って表にされても，x の値は漏洩しない．演算はシェアに対して行い，結果も秘密分散された形で得ることにより，結果のカードの一部が誤って表にされても結果の秘密値は漏洩しない．

さらに，秘密値の入力時にミスを行う場合も考えられる．1 ビットの値 0 または 1 を入力する際に誤って♣，♣または♥，♥を入力する場合が考えられる．不正な入力は検出を行い，正しい入力値である場合は入力値に関する情報が得られない検出プロトコルが考えられている [MS14a]．

カード以外の補助ツール

Five-card trick をはじめとする多くのプロトコルではカード以外のツールは用いられない．しかし，補助ツールが必要な場合も多い．たとえば前述のカードの束に対するシャッフルを行う場合には，それぞれの束のカードがばらばらになることを防止するために，クリップや輪ゴムで束ねる，あるいは封筒に束を入れた後にシャッフルを行うことが現実的な処理手順として想定されている．秘匿操作を行う場合にカードを封筒に入れることも必要と考えられる．また，カードを入れる特殊なケースを用いた効率的なプロトコルが示されている [ISSM24]．

参考文献

下記以外の文献は宮原, 水木, 品川：カードベース暗号の bib ファイル，https://www.ns.kogakuin.ac.jp/manabe/card.html を参照のこと（最終参照日：2026 年 3 月 20 日）．

1）黒澤　馨，篠崎友希：コンパクトなカードプロトコル，SCIS2017 1A2-6（2017）．
2）城内聡志，中井雄士，岩本　貢，太田和夫：秘匿操作を用いた効率的カードベース論理演算プロトコル，SCIS2017 1A2-2（2017）．
3）Nakai, T., Shirouchi, S., Tokushige, Y., Iwamoto, M. and Ohta, K.: How to Solve Millionaires' Problem with Two Kinds of Cards, New Generation Computing Vol.39(1), pp.73-96 (2021).
4）Ono, H. and Manabe, Y.: Card-Based Cryptographic Logical Computations Using Private Operations, New Generation Computing, Vol.39(1), pp.19-40 (2021).
5）Shinagawa, K. and Nuida, K.: A Single Shuffe Is Enough for Secure Card-Based Computation of Any Boolean Circuit, Discrete Applied Mathematics, Vol.289, pp.248-261 (2021).

（2026 年 1 月 31 日受付）

■真鍋義文（正会員）　manabe@cc.kogakuin.ac.jp

1983 年大阪大学・基礎工・情報工卒業，1985 年同大学院修士課程修了．1993 年博士（工学）．1985 ～ 2013 年日本電信電話（株）勤務．2001 ～ 2013 年京都大学大学院情報学研究科社会情報学専攻連携助教授（准教授）兼務．2013 年工学院大学情報学部コンピュータ科学科教授．現在，同大情報科学科教授．日本応用数理学会フェロー，電子情報通信学会，IEEE，ACM 会員．